EPSON
EXCEED YOUR VISION

CMOS 16-BIT SINGLE CHIP MICROCONTROLLER

S1C17M02/M03
Technical Manual

SEIKO EPSON CORPORATION

Rev. 1.0

Evaluation board/kit and Development tool important notice

1. This evaluation board/kit or development tool is designed for use with engineering evaluation, demonstration, or development purposes only. Do not use it for other purposes. It is not intended to meet the design requirements of finished products.
2. This evaluation board/kit or development tool is intended for use by an electronic engineer and is not a consumer product. The user should use it properly and in a safe manner. Seiko Epson does not assume any responsibility or liability of any kind of damage and/or fire caused by its use. The user should cease to use it when any abnormal issue occurs even during proper and safe use.
3. Parts used for this evaluation board/kit or development tool may be changed without any notice.

NOTICE : PLEASE READ CAREFULLY BELOW BEFORE USING THIS DOCUMENT

The contents of this document are subject to change without notice.

1. This document may not be copied, reproduced, or used for any other purpose, in whole or in part, without the consent of the Seiko Epson Corporation (“Epson”).
2. Before purchasing or using Epson products, please contact our sales representative for the latest information and always be sure to check the latest information published on Epson’s official web sites and other sources.
3. Information provided in this document such as application circuits, programs, usage, etc., are for reference purposes only. Using the application circuits, programs, usage, etc. in the design of your equipment or systems is your own responsibility. Epson makes no guarantees against any infringements or damages to any third parties’ intellectual property rights or any other rights resulting from the information. This document does not grant you any licenses, intellectual property rights or any other rights with respect to Epson products owned by Epson or any third parties.
4. Epson is committed to constantly improving quality and reliability, but semiconductor products in general are subject to malfunction and failure. By using Epson products, you shall be responsible for your hardware. Software and systems must be designed well enough to prevent death or injury as well as any property damage even if any of the malfunctions or failures might be caused by Epson products. When designing your products using Epson products, please be sure to check and comply with the latest information regarding Epson products (this document, specifications, data sheets, manuals, Epson’s web site, etc.). When using the information included above materials such as product data, charts, technical contents, programs, algorithms and application circuit examples, you shall evaluate your products both on a stand-alone basis as well as within your overall systems. You shall be solely responsible for deciding whether or not to adopt and use Epson products.
5. Epson has prepared this document and programs provided in this document carefully to be accurate and dependable, but Epson does not guarantee that the information and the programs are always accurate and complete. Epson assumes no responsibility for any damages which you incur due to misinformation in this document and the programs.
6. No dismantling, analysis, reverse engineering, modification, alteration, adaptation, reproduction, etc., of Epson products is allowed.
7. Epson products have been designed, developed and manufactured to be used in general electronic applications (office equipment, communications equipment, measuring instruments, home electronics, etc.) and applications individually listed in this document (“General Purpose”). Epson products are NOT intended for any use beyond the General Purpose uses that requires particular/higher quality or reliability in order to refrain from causing any malfunction or failure leading to death, injury, serious property damage or severe impact on society, including, but not limited to those listed below. Therefore, you are advised to use Epson products only for General Purpose uses. Should you desire to buy and use Epson products for a particular purpose other than a General Purpose use, Epson makes no warranty and disclaims with respect to Epson products, whether express or implied, including without limitation any implied warranty of merchantability or fitness for any particular purpose. Please be sure to contact our sales representative and obtain approval in advance.

   【Particular purpose】

   Space equipment (artificial satellites, rockets, etc.)
   Transportation vehicles and their control equipment (automobiles, aircraft, trains, ships, etc.)
   Medical equipment (other than applications individually listed in this document) / Relay equipment to be placed on ocean floor
   Power station control equipment / Disaster or crime prevention equipment / Traffic control equipment / Financial equipment
   Other applications requiring similar levels of reliability as those listed above
8. Epson products listed in this document and our associated technologies shall not be used in any equipment or systems that laws and regulations in Japan or any other countries prohibit to manufacture, use or sell. Furthermore, Epson products and our associated technologies shall not be used for developing weapons of mass destruction, or any other military purposes or applications. If exporting Epson products or our associated technologies, you shall comply with the Foreign Exchange and Foreign Trade Control Act in Japan, Export Administration Regulations in the U.S.A. (EAR) and other export-related laws and regulations in Japan and any other countries and follow the required procedures as provided by the relevant laws and regulations.
9. Epson assumes no responsibility for any damages (whether direct or indirect) caused by or in relation with your non-compliance with the terms and conditions in this document.
10. Epson assumes no responsibility for any damages (whether direct or indirect) incurred by any third party that you assign, transfer, loan, etc., Epson products to.
11. For more details or other concerns about this document, please contact our sales representative.
12. Company names and product names listed in this document are trademarks or registered trademarks of their respective companies.

(Rev. e1.0, 2021.9)

Preface

This is a technical manual for designers and programmers who develop a product using the S1C17M02/M03. This document describes the functions of the IC, embedded peripheral circuit operations, and their control methods.

For the CPU functions and instructions, refer to the "S1C17 Family S1C17 Core Manual." For the functions and operations of the debugging tools, refer to the respective tool manuals. (Our "Products: Document Downloads" website provides the downloadable manuals.)

Notational conventions and symbols in this manual

Register address

Peripheral circuit chapters do not provide control register addresses. Refer to "Peripheral Circuit Area" in the "Memory and Bus" chapter or "List of Peripheral Circuit Control Registers" in the Appendix.

Register and control bit names

In this manual, the register and control bit names are described as shown below to distinguish from signal and pin names.

XXX register: Represents a register including its all bits.
XXX.YYY bit: Represents the one control bit YYY in the XXX register.
XXX.ZZZ[1:0] bits: Represents the two control bits ZZZ1 and ZZZ0 in the XXX register.

Register table contents and symbols

Initial: Value set at initialization

Reset: Initialization condition. The initialization condition depends on the reset group (H0, H1, or S0). For more information on the reset groups, refer to "Initialization Conditions (Reset Groups)" in the "Power Supply, Reset, and Clocks" chapter.

R/W: R = Read only bit
W = Write only bit
WP = Write only bit with a write protection using the MSCPROT.PROT[15:0] bits
R/W = Read/write bit
R/WP = Read/write bit with a write protection using the MSCPROT.PROT[15:0] bits

Control bit read/write values

This manual describes control bit values in a hexadecimal notation except for one-bit values (and except when decimal or binary notation is required in terms of explanation). The values are described as shown below according to the control bit width.

1 bit: 0 or 1
2 to 4 bits: 0x0 to 0xf
5 to 8 bits: 0x00 to 0xff
9 to 12 bits: 0x000 to 0xfff
13 to 16 bits: 0x0000 to 0xffff

Decimal: 0 to 9999...
Binary: 0b0000... to 0b1111...

Channel number

Multiple channels may be implemented in some peripheral circuits (e.g., 16-bit timer, etc.). The peripheral circuit chapters use '*n*' as the value that represents the channel number in the register and pin names regardless of the number of channel actually implemented. Normally, the descriptions are applied to all channels. If there is a channel that has different functions from others, the channel number is specified clearly.

Example) T16_*n*CTL register of the 16-bit timer
If one channel is implemented (Ch.0 only): T16_*n*CTL = T16_0CTL only
If two channels are implemented (Ch.0 and Ch.1): T16_*n*CTL = T16_0CTL and T16_1CTL

For the number of channels implemented in the peripheral circuits of this IC, refer to "Features" in the "Overview" chapter.

– Contents –

1 Overview

The S1C17M02/M03 is a compact 16-bit MCU specializing in a DMM (Digital Multi-Meter) function that supports C programming. This MCU has implemented the functions (ΣΔ A/D converter, pin switching circuit, and reference voltage generator) required for composing a DMM that is able to measure voltage, current, resistance, capacitance, diode, frequency, and continuity check. Also it is possible to setup optimal measurement conditions with programmable oversampling settings and to reduce a load on software processing with an embedded square root circuit. The S1C17M02/M03 includes a low-power LCD driver, various serial interfaces, a crystal oscillator and various timers as well, thus it is suitable for battery drive measurement equipment not only a DMM. It not only has a Flash memory but also an EEPROM that can be reprogrammed from application software.

1.1 Features

Table 1.1.1 Features

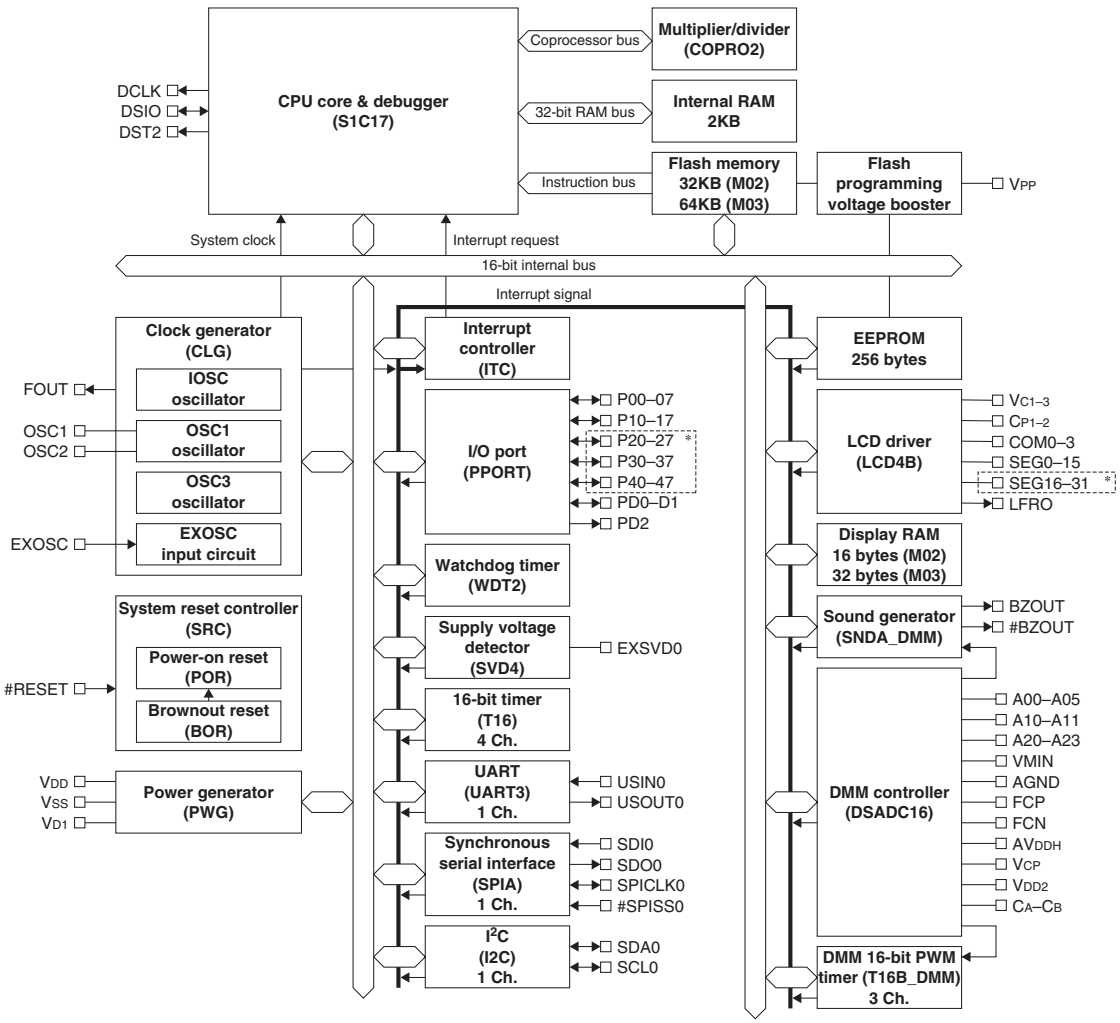
Package type	S1C17M02	S1C17M03
CPU		
CPU core	Seiko Epson original 16-bit RISC CPU core S1C17	
Other	On-chip debugger	
Embedded Flash memory		
Capacity (for both instructions and data)	32K bytes	64K bytes
Erase/program count	1,000 times (min.) * Programming by the debugging tool ICDmini	
Other	Security function to protect from reading/programming by ICDmini	
	On-board programming function using ICDmini	
	Flash programming voltage can be generated internally.	
Embedded EEPROM		
Capacity	256 bytes	
Erase/program count	100,000 times (min.)	
Embedded RAM		
Capacity	2K bytes	
Embedded display RAM		
Capacity	16 bytes	32 bytes
DMM controller (DSADC16)		
DMM measurement mode (range)	DC voltage measuerment (600 mV/6 V/60 V/600 V/1,000 V)	
	AC voltage measurement (600 mV/6 V/60 V/600 V/1,000 V) Simultaneous frequency measurement (5 Hz–100 kHz) is possible.	
	Resistance measurement (600 Ω/6 kΩ/60 kΩ/600 kΩ/6 MΩ/60 MΩ)	
	Continuity check	
	Diode V_F measurement	
	Internal temperature measurement	
	DC current measurement (600 µA/6 mA/60 mA/600 mA/6 A/10 A)	
	AC current measurement (600 µA/6 mA/60 mA/600 mA/6 A/10 A) Simultaneous frequency measurement (5 Hz–100 kHz) is possible.	
	Capacitance measurement (10 nF/100 nF/1 µF/10 µF/100 µF/1,000 µF)	
A/D conversion circuit	ΣΔ type 16-bit A/D converter, averaging circuit, square root circuit, DC/AC peak hold circuits	
Measurement output	DC measurement values: Signed 16-bit values AC measurement values: 32 bits (mean square), 16 bits (mean absolute, true RMS value)	
DMM 16-bit PWM timer (T16B_DMM)	Used for frequency measurement with software.	
Clock generator (CLG)		
System clock source	4 sources (IOSC/OSC1/OSC3/EXOSC)	
System clock frequency (operating frequency)	6.7 MHz (max.)	
IOSC oscillator circuit (boot clock source)	700 kHz (typ.) embedded oscillator (boot clock)	
	23 µs (max.) starting time (time from cancelation of SLEEP state to vector table read by the CPU)	
OSC1 oscillator circuit	32.768 kHz (typ.) crystal oscillator	
	32 kHz (typ.) embedded oscillator	
	Oscillation stop detection circuit included	
OSC3 oscillator circuit	6.4/3.2 MHz (typ.) embedded oscillator	
EXOSC clock input	6.7 MHz (max.) square or sine wave input	
Other	Configurable system clock division ratio	
	Configurable system clock used at wake up from SLEEP state	
	Operating clock frequency for the CPU and all peripheral circuits is selectable.	

Package type		S1C17M02	S1C17M03
I/O port (PPORT)			
Number of general-purpose ports	Input/output port	18 bits (max.)	42 bits (max.)
	Output port	1 bit (max.)	
	Other	Pins are shared with the peripheral I/O.	
Number of input interrupt ports		16 bits (max.)	40 bits (max.)
Number of ports that support universal port multiplexer (UPMUX)		16 bits (8 bits are shared with LCD SEG outputs)	32 bits (24 bits are shared with LCD SEG outputs)
		A peripheral circuit I/O function selected via software can be assigned to each port.	
Timers			
Watchdog timer (WDT2)		Generates NMI or watchdog timer reset.	
		Programmable NMI/reset generation cycle	
16-bit timer (T16)		4 channels	
		Generates the SPIA master clock.	
Supply voltage detector (SVD4)			
Detection voltage		V_{DD} or external voltage (one external voltage input port is provided and an external voltage level can be detected even if it exceeds V_{DD}.)	
Detection level		V_{DD}: 19 levels (1.7 to 3.6 V)/external voltage: 19 levels (1.7 to 3.6 V)	
Other		Intermittent operation mode	
		Generates an interrupt or reset according to the detection level evaluation.	
Serial interfaces			
UART (UART3)		1 channel	
		Baud-rate generator included, IrDA1.0 supported	
		Open drain output, signal polarity, and baud rate division ratio are configurable.	
		Infrared communication carrier modulation output function	
Synchronous serial interface (SPIA)		1 channel	
		2 to 16-bit variable data length	
		The 16-bit timer (T16) can be used for the baud-rate generator in master mode.	
I^2C (I2C) *1		1 channel	
		Baud-rate generator included	
Sound generator (SNDA_DMM)			
Buzzer output function		512 Hz to 16 kHz output frequencies	
		One-shot output function	
Melody generation function		Pitch: 128 Hz to 16 kHz ≈ C3 to C6	
		Duration: 7 notes/rests (Half note/rest to thirty-second note/rest)	
		Tempo: 16 tempos (30 to 480)	
		Tie may be specified.	
Other		Automatic buzzer output function linking with the DSADC16 continuity check mode	
LCD driver (LCD4B)			
LCD output (max value.)		16_{SEG} × 1 to 4_{COM}	32_{SEG} × 1 to 4_{COM}
LCD power supply		1/3 bias power supply included.	
		An external voltage can be applied. (Internal resistors are provided to divide the external source voltage.)	
LCD contrast		32 levels	
Multiplier/divider (COPRO2)			
Arithmetic functions		16-bit × 16-bit multiplier	
		16-bit × 16-bit + 32-bit multiply and accumulation unit	
		32-bit ÷ 32-bit divider	
Reset			
#RESET pin		Reset when the reset pin is set to low.	
Power-on reset		Reset at power on.	
Brownout reset		Reset when the power supply voltage drops.	
Key entry reset		Reset when the P00 to P01/P02/P03 keys are pressed simultaneously (can be enabled/disabled using a register).	
Watchdog timer reset		Reset when the watchdog timer overflows (can be enabled/disabled using a register).	
Supply voltage detector reset		Reset when the supply voltage detector detects the set voltage level (can be enabled/disabled using a register).	
Interrupt			
Non-maskable interrupt		4 systems (Reset, address misaligned interrupt, debug, NMI)	
Programmable interrupt		External interrupt: 1 system (8 levels)	
		Internal interrupt: 17 systems (8 levels)	
Power supply voltage			
V_{DD} operating voltage		2.1 to 3.6 V	
V_{DD} operating voltage for analog operations		2.2 to 3.6 V	
V_{DD} operating voltage for Flash programming		2.2 to 3.6 V (Programming voltage V_{PP}: 7.5 V external voltage or internal boosted voltage)	
V_{DD} operating voltage for EEPROM programming		2.2 to 3.6 V (Programming voltage V_{PP}: internal boosted voltage)	

Package type	S1C17M02	S1C17M03
Operating temperature		
Operating temperature range	-40 to 85 °C	
Current consumption (typ. value)		
SLEEP mode	0.24 µA IOSC = OFF, OSC1 = OFF, OSC3 = OFF	
HALT mode	1.8 µA IOSC = OFF, OSC1 = ON (32 kHz internal oscillator), OSC3 = OFF	
RUN mode	6.0 µA IOSC = OFF, OSC1 = ON (32 kHz internal oscillator), OSC3 = OFF, f_{CPU} = OSC1	
	825 µA IOSC = OFF, OSC1 = ON (32.768 kHz), OSC3 = ON (3.2 MHz internal oscillator), f_{CPU} = OSC3	
Shipping form		
Package *2	QFP13-64PIN (P-LQFP064-1010-0.50, 10 × 10 mm, t = 1.7 mm, 0.5 mm pitch)	QFP15-100PIN (P-LQFP100-1414-0.50, 14 × 14 mm, t = 1.7 mm, 0.5 mm pitch)

*1 The input filter in I2C (SDA and SCL inputs) does not comply with the standard for removing noise spikes less than 50 ns.
*2 Shown in parentheses are JEITA package names.

1.2 Block Diagram



* The pin configuration depends on the package. For detailed information, refer to Section 1.3, "Pins."

Figure 1.2.1 S1C17M02/M03 Block Diagram

1.3 Pins

1.3.1 S1C17M02 Pin Configuration Diagram (QFP13-64PIN)

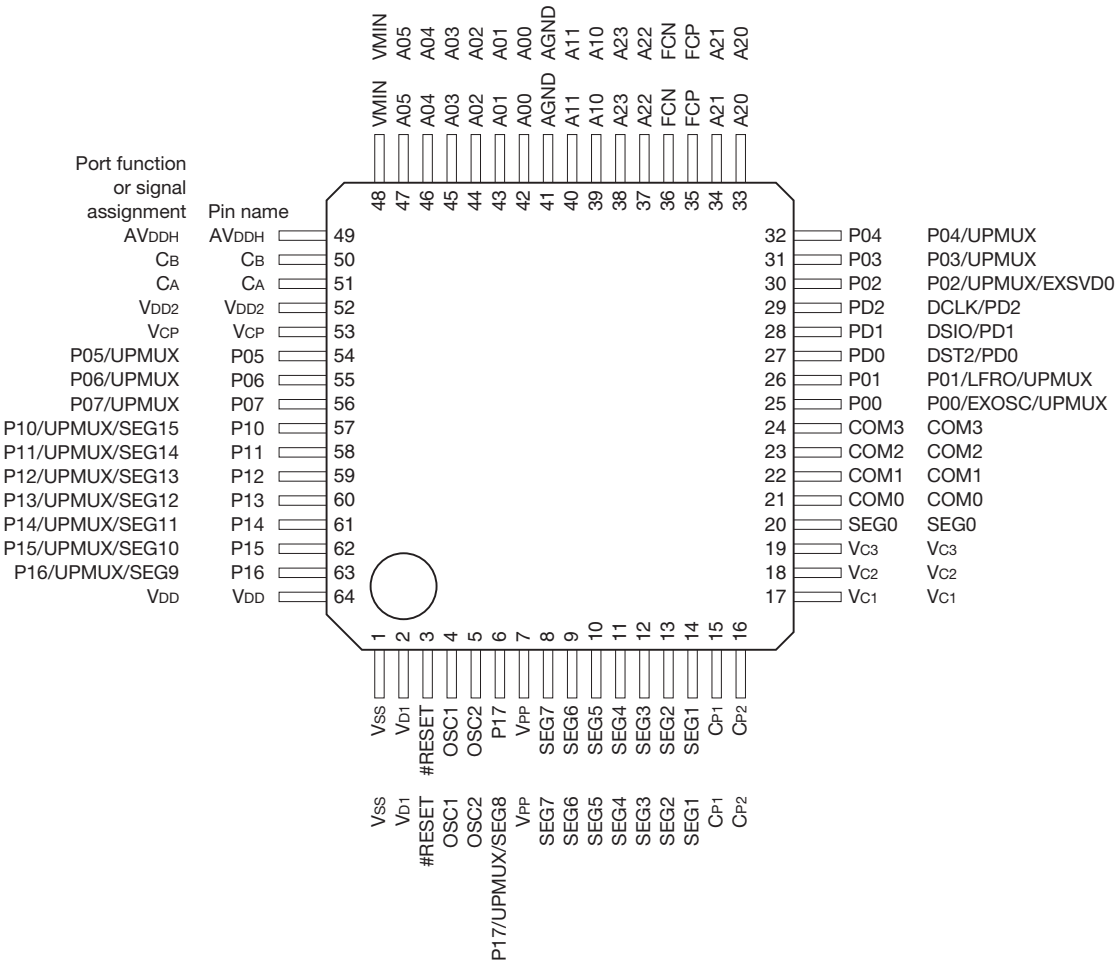


Figure 1.3.1.1 S1C17M02 Pin Configuration Diagram (QFP13-64PIN)

1.3.2 S1C17M03 Pin Configuration Diagram (QFP15-100PIN)

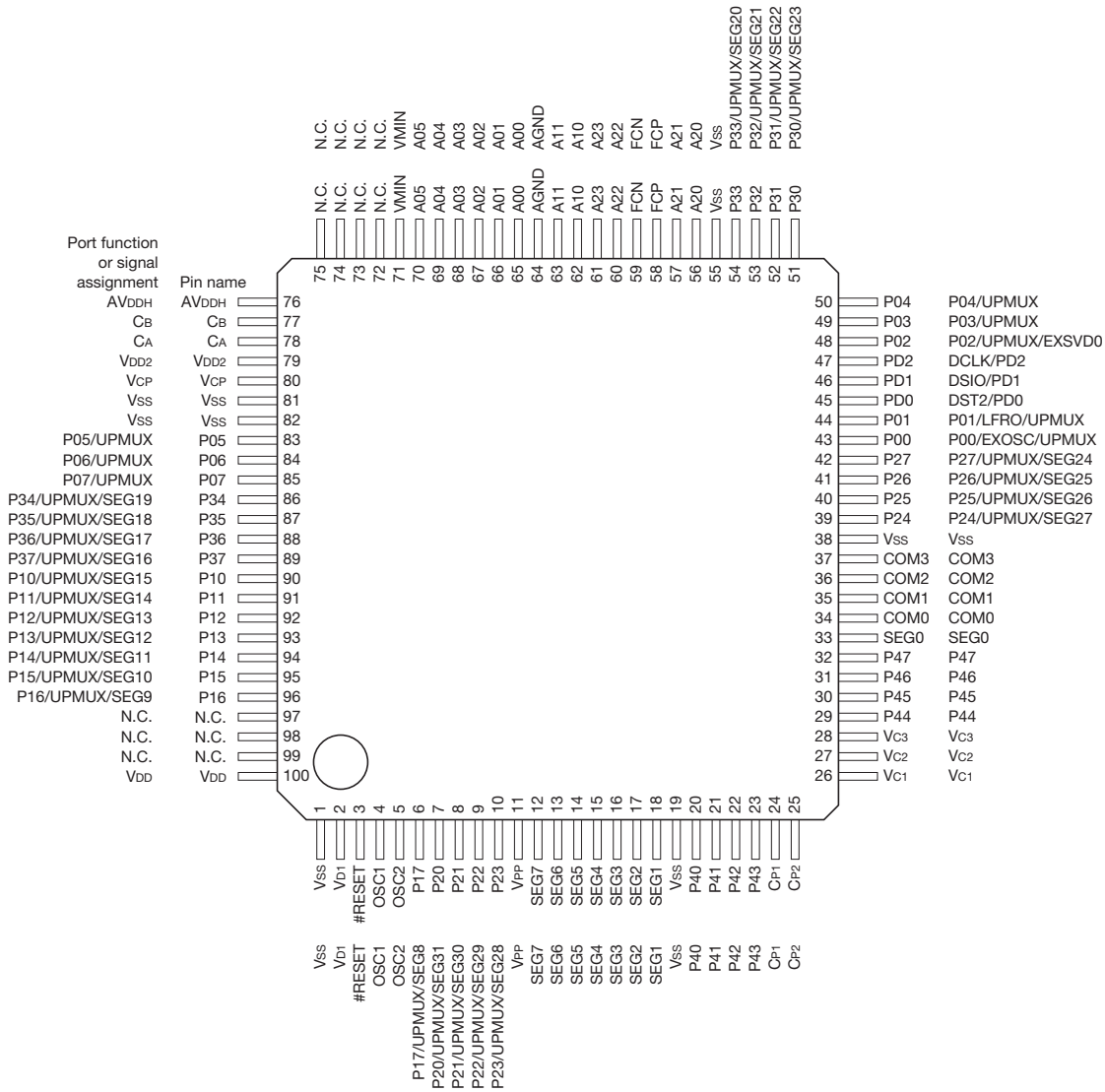


Figure 1.3.2.1 S1C17M03 Pin Configuration Diagram (QFP15-100PIN)

1.3.3 Pin Descriptions

Symbol meanings

Assigned signal: The signal listed at the top of each pin is assigned in the initial state. The pin function must be switched via software to assign another signal (see the "I/O Ports" chapter).

I/O:
- I = Input
- O = Output
- I/O = Input/output
- P = Power supply
- A = Analog signal
- Hi-Z = High impedance state

Initial state:
- I (Pull-up) = Input with pulled up
- I (Pull-down) = Input with pulled down
- Hi-Z = High impedance state
- O (H) = High level output
- O (L) = Low level output

Tolerant fail-safe structure:
- ✓ = Over voltage tolerant fail-safe type I/O cell included (see the "I/O Ports" chapter)

Table 1.3.3.1 Pin description

Pin name	Assigned signal	I/O	Initial state	Tolerant fail-safe structure	Function	M02	M03
V_{DD}	V_{DD}	P	–	–	Power supply (+)	✓	✓
V_{SS}	V_{SS}	P	–	–	GND	✓	✓
V_{D1}	V_{D1}	A	–	–	V_{D1} regulator output	✓	✓
V_{PP}	V_{PP}	A	–	–	Flash programming power supply	✓	✓
AV_{DDH}	AV_{DDH}	P	–	–	Regulator output (3.6 V output, power supply for analog network unit)	✓	✓
V_{CP}	V_{CP}	P	–	–	Regulator output (2.1 V output, power supply for charge pump)	✓	✓
AGND	AGND	P	–	–	Analog GND	✓	✓
V_{DD2}	V_{DD2}	P	–	–	3.6 V regulator power supply	✓	✓
$C_{A–B}$	$C_{A–B}$	A	–	–	Power voltage boost capacitor connect pins	✓	✓
VMIN	VMIN	A	–	–	DMM measurement pin	✓	✓
A00–05	A00–05	A	–	–	DMM measurement pins	✓	✓
A10–11	A10–11	A	–	–	DMM measurement pins	✓	✓
A20–23	A20–23	A	–	–	DMM measurement pins	✓	✓
FCP	FCP	A	–	–	DMM filter capacitor connect pin	✓	✓
FCN	FCN	A	–	–	DMM filter capacitor connect pin	✓	✓
$V_{C1–3}$	$V_{C1–3}$	P	–	–	LCD panel drive power supply	✓	✓
$C_{P1–P2}$	$C_{P1–P2}$	A	–	–	LCD voltage boost capacitor connect pins	✓	✓
OSC1	OSC1	A	–	–	OSC1 oscillator circuit input	✓	✓
OSC2	OSC2	A	–	–	OSC1 oscillator circuit output	✓	✓
#RESET	#RESET	I	I (Pull-up)	–	Reset input	✓	✓
P00	P00	I/O	Hi-Z	✓	I/O port	✓	✓
	EXOSC	I			Clock generator external clock input	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓
P01	P01	I/O	Hi-Z	✓	I/O port	✓	✓
	LFRO	O			LCD frame signal monitor output	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓
P02	P02	I/O	Hi-Z	✓	I/O port	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓
	EXSVD0	A			External power supply voltage detection input	✓	✓
P03	P03	I/O	Hi-Z	–	I/O port	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓
P04	P04	I/O	Hi-Z	–	I/O port	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓
P05	P05	I/O	Hi-Z	–	I/O port	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓
P06	P06	I/O	Hi-Z	–	I/O port	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓
P07	P07	I/O	Hi-Z	–	I/O port	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓

Pin name	Assigned signal	I/O	Initial state	Tolerant fail-safe structure	Function	M02	M03
P10	P10	I/O	Hi-Z	✓	I/O port	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓
	SEG15	A			LCD segment output	✓	✓
P11	P11	I/O	Hi-Z	✓	I/O port	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓
	SEG14	A			LCD segment output	✓	✓
P12	P12	I/O	Hi-Z	✓	I/O port	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓
	SEG13	A			LCD segment output	✓	✓
P13	P13	I/O	Hi-Z	✓	I/O port	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓
	SEG12	A			LCD segment output	✓	✓
P14	P14	I/O	Hi-Z	✓	I/O port	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓
	SEG11	A			LCD segment output	✓	✓
P15	P15	I/O	Hi-Z	✓	I/O port	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓
	SEG10	A			LCD segment output	✓	✓
P16	P16	I/O	Hi-Z	✓	I/O port	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓
	SEG9	A			LCD segment output	✓	✓
P17	P17	I/O	Hi-Z	✓	I/O port	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓
	SEG8	A			LCD segment output	✓	✓
P20	P20	I/O	Hi-Z	✓	I/O port	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓
	SEG31	A			LCD segment output	–	✓
P21	P21	I/O	Hi-Z	✓	I/O port	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓
	SEG30	A			LCD segment output	–	✓
P22	P22	I/O	Hi-Z	✓	I/O port	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓
	SEG29	A			LCD segment output	–	✓
P23	P23	I/O	Hi-Z	✓	I/O port	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓
	SEG28	A			LCD segment output	–	✓
P24	P24	I/O	Hi-Z	✓	I/O port	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓
	SEG27	A			LCD segment output	–	✓
P25	P25	I/O	Hi-Z	✓	I/O port	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓
	SEG26	A			LCD segment output	–	✓
P26	P26	I/O	Hi-Z	✓	I/O port	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓
	SEG25	A			LCD segment output	–	✓
P27	P27	I/O	Hi-Z	✓	I/O port	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓
	SEG24	A			LCD segment output	–	✓
P30	P30	I/O	Hi-Z	✓	I/O port	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓
	SEG23	A			LCD segment output	–	✓
P31	P31	I/O	Hi-Z	✓	I/O port	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓
	SEG22	A			LCD segment output	–	✓
P32	P32	I/O	Hi-Z	✓	I/O port	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓
	SEG21	A			LCD segment output	–	✓
P33	P33	I/O	Hi-Z	✓	I/O port	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓
	SEG20	A			LCD segment output	–	✓
P34	P34	I/O	Hi-Z	✓	I/O port	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓
	SEG19	A			LCD segment output	–	✓

Pin name	Assigned signal	I/O	Initial state	Tolerant fail-safe structure	Function	M02	M03
P35	P35	I/O	Hi-Z	✓	I/O port	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓
	SEG18	A			LCD segment output	–	✓
P36	P36	I/O	Hi-Z	✓	I/O port	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓
	SEG17	A			LCD segment output	–	✓
P37	P37	I/O	Hi-Z	✓	I/O port	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓
	SEG16	A			LCD segment output	–	✓
P40	P40	I/O	Hi-Z	✓	I/O port	–	✓
P41	P41	I/O	Hi-Z	✓	I/O port	–	✓
P42	P42	I/O	Hi-Z	✓	I/O port	–	✓
P43	P43	I/O	Hi-Z	✓	I/O port	–	✓
P44	P44	I/O	Hi-Z	✓	I/O port	–	✓
P45	P45	I/O	Hi-Z	✓	I/O port	–	✓
P46	P46	I/O	Hi-Z	✓	I/O port	–	✓
P47	P47	I/O	Hi-Z	✓	I/O port	–	✓
PD0	DST2	O	O (L)	–	On-chip debugger status output	✓	✓
	PD0	I/O			I/O port	✓	✓
PD1	DSIO	I/O	I (Pull-up)	–	On-chip debugger data input/output	✓	✓
	PD1	I/O			I/O port	✓	✓
PD2	DCLK	O	O (H)	–	On-chip debugger clock output	✓	✓
	PD2	O			Output port	✓	✓
SEG0–7	SEG0–7	A	Hi-Z	–	LCD segment outputs	✓	✓
COM0–3	COM0–3	A	Hi-Z	–	LCD common outputs	✓	✓

Note: In the peripheral circuit descriptions, the assigned signal name is used as the pin name.

Universal port multiplexer (UPMUX)

The universal port multiplexer (UPMUX) allows software to select the peripheral circuit input/output function to be assigned to each pin from those listed below.

Table 1.3.3.2 Peripheral Circuit Input/output Function Selectable by UPMUX

Peripheral circuit	Signal to be assigned	I/O	Channel number *n*	Function
Synchronous serial interface (SPIA)	SDI*n*	I	*n* = 0	SPIA Ch.*n* data input
	SDO*n*	O		SPIA Ch.*n* data output
	SPICLK*n*	I/O		SPIA Ch.*n* clock input/output
	#SPISS*n*	I		SPIA Ch.*n* slave-select input
I²C (I2C)	SCL*n*	I/O	*n* = 0	I2C Ch.*n* clock input/output
	SDA*n*	I/O		I2C Ch.*n* data input/output
UART (UART3)	USIN*n*	I	*n* = 0	UART3 Ch.*n* data input
	USOUT*n*	O		UART3 Ch.*n* data output
Sound generator (SNDA_DMM)	BZOUT	O	*n* = 0	SNDA_DMM buzzer output
	#BZOUT	O		SNDA_DMM inverted buzzer output
Clock generator (CLG)	FOUT	O	*n* = 0	CLG clock external output

Note: Do not assign a function to two or more pins simultaneously.

2 Power Supply, Reset, and Clocks

The power supply, reset, and clocks in this IC are managed by the embedded power generator, system reset controller, and clock generator, respectively.

2.1 Power Generator (PWG)

2.1.1 Overview

PWG is the power generator that controls the internal power supply system to drive this IC with stability and low power. The main features of PWG are outlined below.

- Embedded V_{D1} regulator
  - The V_{D1} regulator generates the V_{D1} voltage to drive internal circuits, this makes it possible to keep current consumption constant independent of the V_{DD} voltage level.
  - The V_{D1} regulator supports two operation modes, normal mode and economy mode, and setting the V_{D1} regulator into economy mode at light loads helps achieve low-power operations.

Figure 2.1.1.1 shows the PWG configuration.

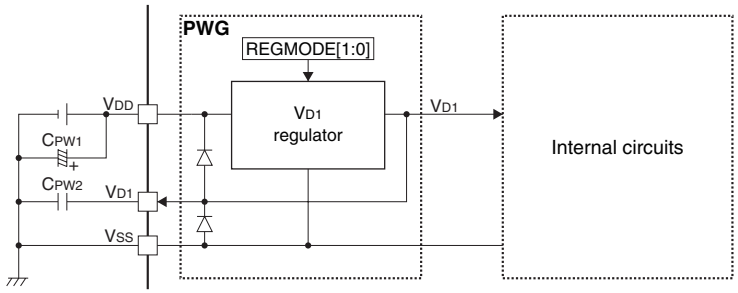


Figure 2.1.1.1 PWG Configuration

2.1.2 Pins

Table 2.1.2.1 lists the PWG pins.

Table 2.1.2.1 List of PWG Pins

Pin name	I/O	Initial status	Function
V_{DD}	P	–	Power supply (+)
V_{SS}	P	–	GND
V_{D1}	A	–	Embedded regulator output pin

For the V_{DD} operating voltage range and recommended external parts, refer to "Recommended Operating Conditions, Power supply voltage V_{DD}" in the "Electrical Characteristics" chapter and the "Basic External Connection Diagram" chapter, respectively.

2.1.3 V_{D1} Regulator Operation Mode

The V_{D1} regulator supports two operation modes, normal mode and economy mode. Setting the V_{D1} regulator into economy mode at light loads helps achieve low-power operations. Table 2.1.3.1 lists examples of light load conditions in which economy mode can be set.

Table 2.1.3.1 Examples of Light Load Conditions in which Economy Mode Can be Set

Light load condition	Exceptions
SLEEP mode (when all oscillators are stopped, or OSC1 only is active)	When a clock source except for OSC1 is active
HALT mode (when OSC1 only is active)	
RUN mode (when OSC1 only is active)	

The V_{D1} regulator also supports automatic mode in which the hardware detects a light load condition and automatically switches between normal mode and economy mode. Use the V_{D1} regulator in automatic mode when no special control is required.

2.2 System Reset Controller (SRC)

2.2.1 Overview

SRC is the system reset controller that resets the internal circuits according to the requests from the reset sources to archive steady IC operations. The main features of SRC are outlined below.

- Embedded reset hold circuit maintains reset state to boot the system safely while the internal power supply is unstable after power on or the oscillation frequency is unstable after the clock source is initiated.
- Supports reset requests from multiple reset sources.
  - #RESET pin
  - POR and BOR
  - Key-entry reset
  - Watchdog timer reset
  - Supply voltage detector reset
  - Peripheral circuit software reset (supports some peripheral circuits only)
- The CPU registers and peripheral circuit control bits will be reset with an appropriate initialization condition according to changes in status.

Figure 2.2.1.1 shows the SRC configuration.

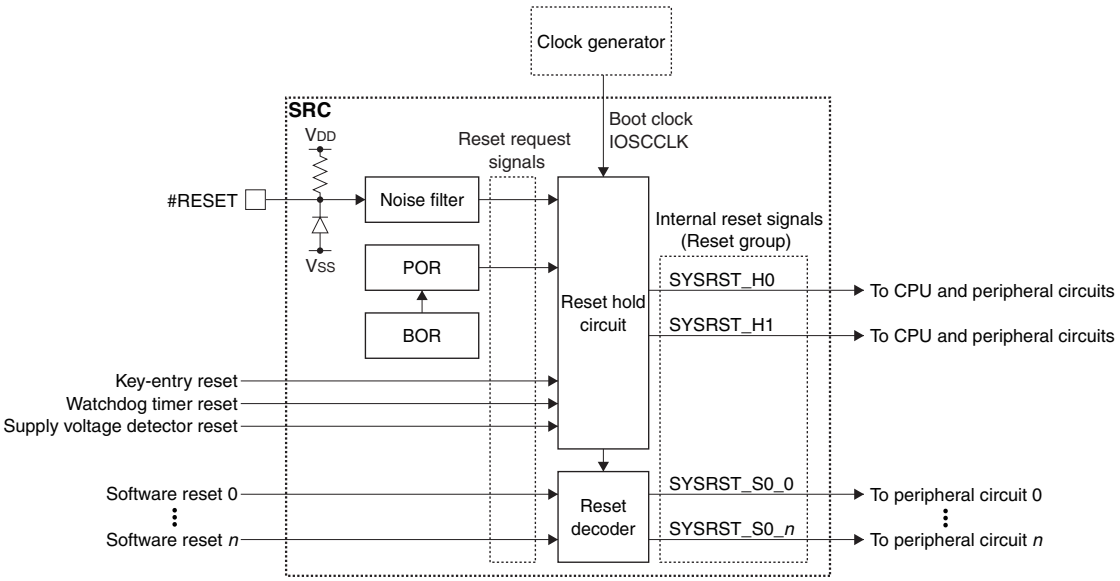


Figure 2.2.1.1 SRC Configuration

2.2.2 Input Pin

Table 2.2.2.1 shows the SRC pin.

Table 2.2.2.1 SRC Pin

Pin name	I/O	Initial status	Function
#RESET	I	I (Pull-up)	Reset input

The #RESET pin is connected to the noise filter that removes pulses not conforming to the requirements. An internal pull-up resistor is connected to the #RESET pin, so the pin can be left open. For the #RESET pin characteristics, refer to "#RESET pin characteristics" in the "Electrical Characteristics" chapter.

2.2.3 Reset Sources

The reset source refers to causes that request system initialization. The following shows the reset sources.

#RESET pin

Inputting a reset signal with a certain low level period to the #RESET pin issues a reset request.

POR and BOR

POR (Power On Reset) issues a reset request when the rise of VDD is detected. BOR (Brownout Reset) issues a reset request when a certain VDD voltage level is detected. Reset requests from these circuits ensure that the system will be reset properly when the power is turned on and the supply voltage is out of the operating voltage range. Figure 2.2.3.1 shows an example of POR and BOR internal reset operation according to variations in VDD.

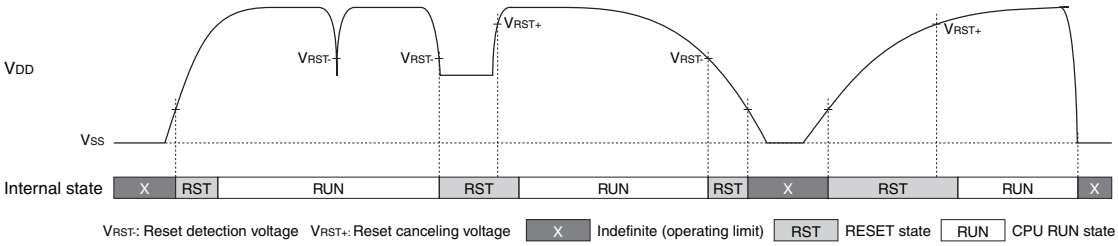


Figure 2.2.3.1 Example of Internal Reset by POR and BOR

For the POR and BOR electrical specifications, refer to "POR/BOR characteristics" in the "Electrical Characteristics" chapter.

Key-entry reset

Inputting a low level signal of a certain period to the I/O port pins configured to a reset input issues a reset request. This function must be enabled using an I/O port register. For more information, refer to the "I/O Ports" chapter.

Watchdog timer reset

Setting the watchdog timer into reset mode will issue a reset request when the counter overflows. This helps return the runaway CPU to a normal operating state. For more information, refer to the "Watchdog timer" chapter.

Supply voltage detector reset

By enabling the low power supply voltage detection reset function, the supply voltage detector will issue a reset request when a drop in the power supply voltage is detected. This makes it possible to put the system into reset state if the IC must be stopped under a low voltage condition. For more information, refer to the "Supply Voltage Detector" chapter.

Peripheral circuit software reset

Some peripheral circuits provide a control bit for software reset (MODEN or SFTRST). Setting this bit initializes the peripheral circuit control bits. Note, however, that the software reset operations depend on the peripheral circuit. For more information, refer to "Control Registers" in each peripheral circuit chapter.

Note: The MODEN bit of some peripheral circuits does not issue software reset.

2.2.4 Initialization Conditions (Reset Groups)

A different initialization condition is set for the CPU registers and peripheral circuit control bits, individually. The reset group refers to an initialization condition. Initialization is performed when a reset source included in a reset group issues a reset request. Table 2.2.4.1 lists the reset groups. For the reset group to initialize the registers and control bits, refer to the "CPU and Debugger" chapter or "Control Registers" in each peripheral circuit chapter.

Table 2.2.4.1 List of Reset Groups

Reset group	Reset source	Reset cancelation timing
H0	#RESET pin POR and BOR Key-entry reset Supply voltage detector reset Watchdog timer reset	Reset state is maintained for the reset hold time t_{RSTR} after the reset request is canceled.
H1	#RESET pin POR and BOR	
S0	Peripheral circuit software reset (MODEN and SFTRST bits. The software reset operations depend on the peripheral circuit.	Reset state is canceled immediately after the reset request is canceled.

2.3 Clock Generator (CLG)

2.3.1 Overview

CLG is the clock generator that controls the clock sources and manages clock supply to the CPU and the peripheral circuits. The main features of CLG are outlined below.

- Supports multiple clock sources.
  - IOSC oscillator circuit that oscillates with a fast startup and no external parts required
  - Low-power OSC1 oscillator circuit in which the oscillator type can be specified from high-precision 32.768 kHz crystal oscillator (an external resonator is required) and internal oscillator
  - High-speed OSC3 oscillator circuit with no external parts required
  - EXOSC clock input circuit that allows input of square wave and sine wave clock signals
- The system clock (SYSCLK), which is used as the operating clock for the CPU and bus, and the peripheral circuit operating clocks can be configured individually by selecting the suitable clock source and division ratio.
- IOSCCLK output from the IOSC oscillator circuit is used as the boot clock for fast booting.
- Controls the oscillator and clock input circuits to enable/disable according to the operating mode, RUN or SLEEP mode.
- Provides a flexible system clock switching function at SLEEP mode cancelation.
  - The clock sources to be stopped in SLEEP mode can be selected.
  - SYSCLK to be used at SLEEP mode cancelation can be selected from all clock sources.
  - The oscillator and clock input circuit on/off state can be maintained or changed at SLEEP mode cancelation.
- Provides the FOUT function to output an internal clock for driving external ICs or for monitoring the internal state.

Figure 2.3.1.1 shows the CLG configuration.

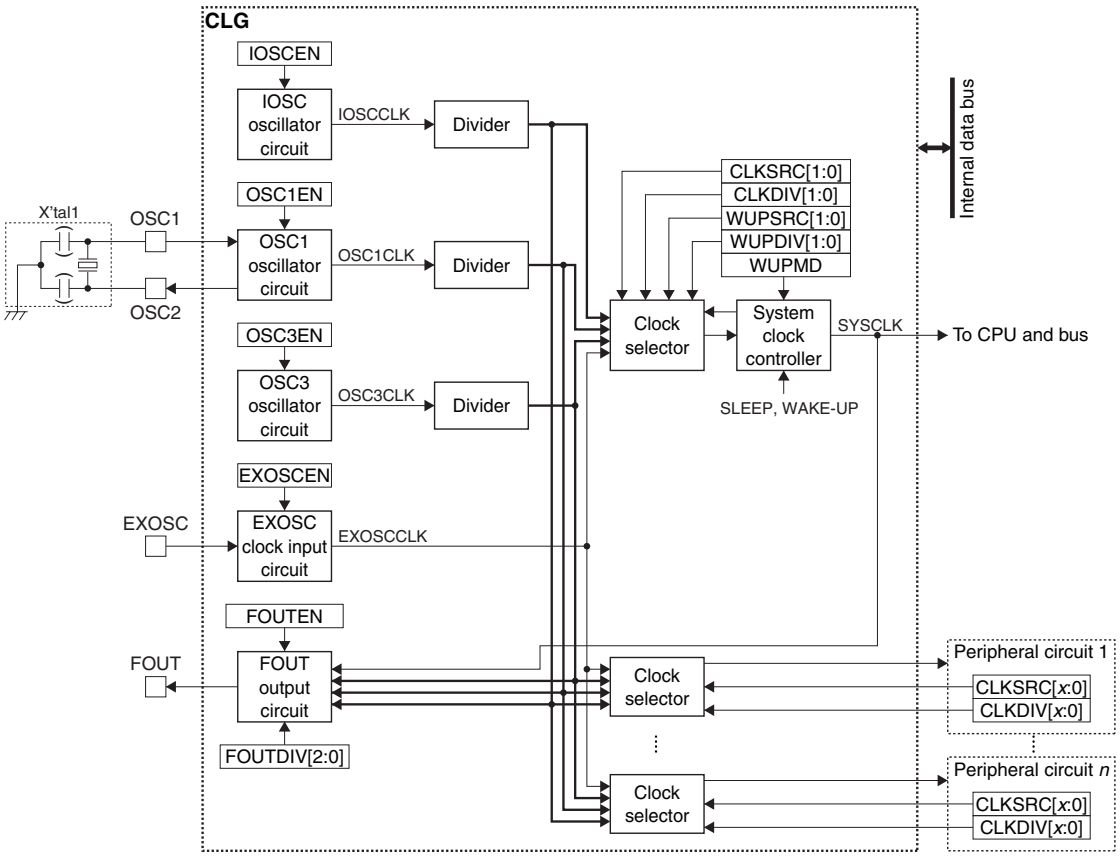


Figure 2.3.1.1 CLG Configuration

2.3.2 Input/Output Pins

Table 2.3.2.1 lists the CLG pins.

Table 2.3.2.1 List of CLG Pins

Pin name	I/O*	Initial status*	Function
OSC1	A	–	OSC1 oscillator circuit input
OSC2	A	–	OSC1 oscillator circuit output
EXOSC	I	I	EXOSC clock input
FOUT	O	O (L)	FOUT clock output

* Indicates the status when the pin is configured for CLG.

If the port is shared with the CLG input/output function and other functions, the CLG function must be assigned to the port. For more information, refer to the "I/O Ports" and "Universal Port Multiplexer" chapters.

2.3.3 Clock Sources

IOSC oscillator circuit

The IOSC oscillator circuit features a fast startup and no external parts are required for oscillating. Figure 2.3.3.1 shows the configuration of the IOSC oscillator circuit.

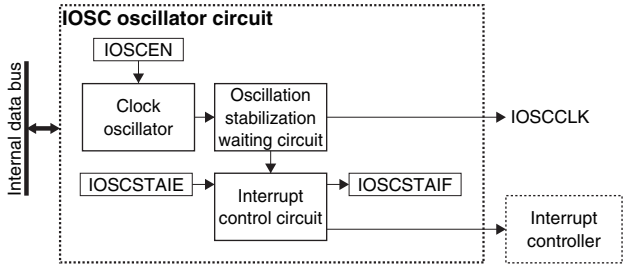


Figure 2.3.3.1 IOSC Oscillator Circuit Configuration

The IOSC oscillator circuit output clock IOSCCLK is used as SYSCLK at booting. For the oscillation characteristics, refer to "IOSC oscillator circuit characteristics" in the "Electrical Characteristics" chapter.

OSC1 oscillator circuit

The OSC1 oscillator circuit is a low-power oscillator circuit that allows software to select the oscillator type from two different types shown below. Figure 2.3.3.2 shows the configuration of the OSC1 oscillator circuit.

Crystal oscillator

This oscillator circuit includes a gain-controlled oscillation inverter and a variable gate capacitor allowing use of various crystal resonators (32.768 kHz typ.) with ranges from cylinder type through surface-mount type. The oscillator circuit also includes a feedback resistor and a drain resistor, so no external parts are required except for a crystal resonator. The embedded oscillation stop detector, which detects oscillation stop and restarts the oscillator, allows the system to operate in safety under adverse environments that may stop the oscillation. The oscillation startup control circuit operates for a set period of time after the oscillation is enabled to assist the oscillator in initiating, this makes it possible to use a low-power resonator that is difficult to start up.

Note: Depending on the circuit board or the crystal resonator type used, an external gate capacitor C_{G1} and a drain capacitor C_{D1} may be required.

Internal oscillator

This 32 kHz oscillator circuit operates without any external parts.

When the internal oscillator circuit is used, the OSC1 and OSC3 pins must be left open.

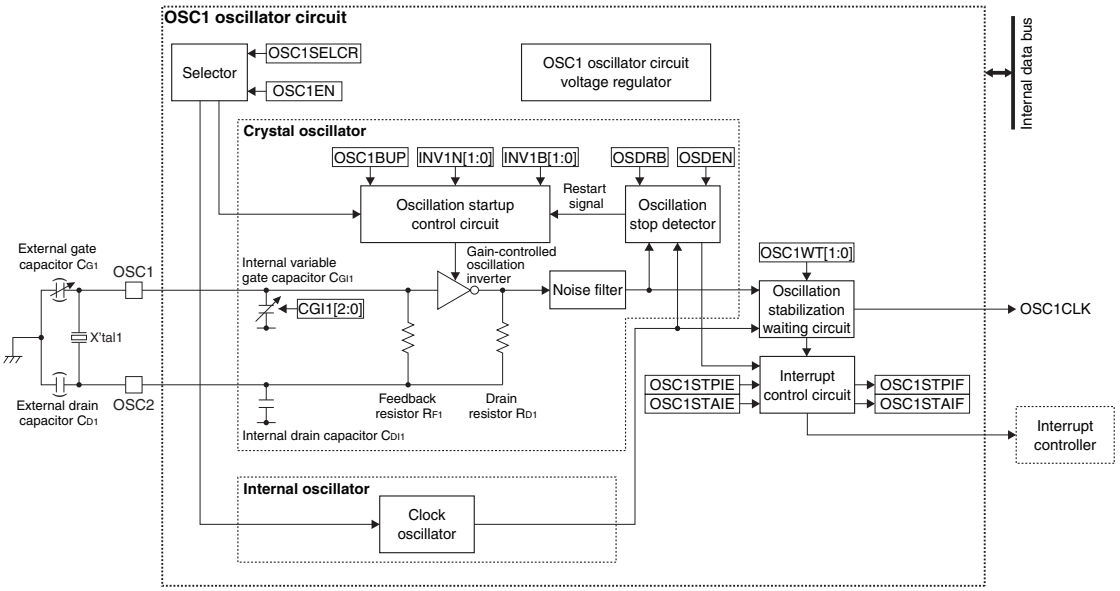


Figure 2.3.3.2 OSC1 Oscillator Circuit Configuration

For the recommended parts and the oscillation characteristics, refer to the "Basic External Connection Diagram" chapter and "OSC1 oscillator circuit characteristics" in the "Electrical Characteristics" chapter, respectively.

OSC3 oscillator circuit

The OSC3 oscillator circuit features a fast startup and no external parts are required for oscillating. Figure 2.3.3.3shows the configuration of the OSC3 oscillator circuit.

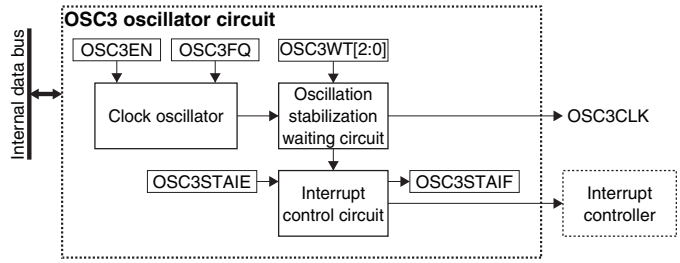


Figure 2.3.3.3 OSC3 Oscillator Circuit Configuration

The OSC3CLK frequency can be selected using the CLGOSC3.OSC3FQ bit. For more information on the oscillation characteristics, refer to "OSC3 oscillator circuit characteristics" in the "Electrical Characteristics" chapter.

EXOSC clock input

EXOSC is an external clock input circuit that supports square wave and sine wave clocks. Figure 2.3.3.4 shows the configuration of the EXOSC clock input circuit.

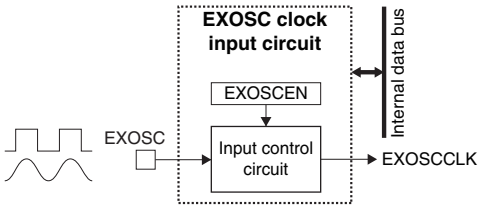


Figure 2.3.3.4 EXOSC Clock Input Circuit

EXOSC has no oscillation stabilization waiting circuit included, therefore, it must be enabled when a stabilized clock is being supplied. For the input clock characteristics, refer to "EXOSC external clock input characteristics" in the "Electrical Characteristics" chapter.

2.3.4 Operations

Oscillation start time and oscillation stabilization waiting time

The oscillation start time refers to the time after the oscillator circuit is enabled until the oscillation signal is actually sent to the internal circuits. The oscillation stabilization waiting time refers to the time it takes the clock to stabilize after the oscillation starts. To avoid malfunctions of the internal circuits due to an unstable clock during this period, the oscillator circuit includes an oscillation stabilization waiting circuit that can disable supplying the clock to the system until the designated time has elapsed. Figure 2.3.4.1 shows the relationship between the oscillation start time and the oscillation stabilization waiting time.

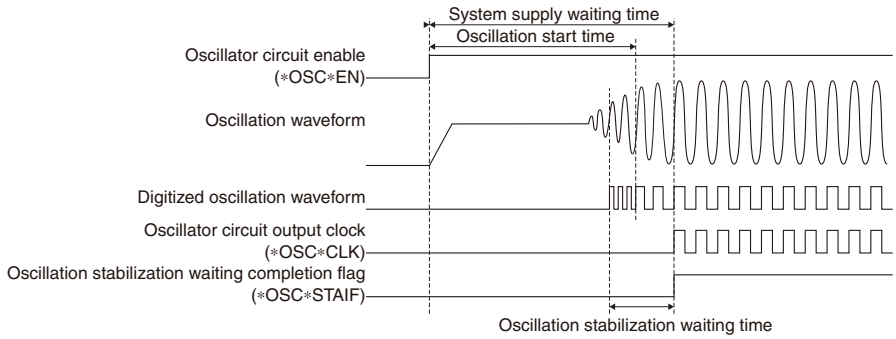


Figure 2.3.4.1 Oscillation Start Time and Oscillation Stabilization Waiting Time

The oscillation stabilization waiting times for the OSC1 and OSC3 oscillator circuits can be set using the CLGOSC1.OSC1WT[1:0] bits and CLGOSC3.OSC3WT[2:0] bits, respectively. To check whether the oscillation stabilization waiting time is set properly and the clock is stabilized immediately after the oscillation starts or not, monitor the oscillation clock using the FOUT output function. The oscillation stabilization waiting time for the IOSC oscillator circuit is fixed at 16 IOSCCLK clocks. The oscillation stabilization waiting time for the OSC1 oscillator circuit should be set to 16,384 OSC1CLK clocks or more when crystal oscillator is selected, or 4,096 OSC1CLK clocks or more when internal oscillator is selected. The oscillation stabilization waiting time for the OSC3 oscillator circuit should be set to 4 OSC3CLK clocks or more.
When the oscillation stabilization waiting operation has completed, the oscillator circuit sets the oscillation stabilization waiting completion flag and starts clock supply to the internal circuits.

Note: The oscillation stabilization waiting time is always expended at start of oscillation even if the oscillation stabilization waiting completion flag has not be cleared to 0.

When the oscillation startup control circuit in the OSC1 crystal oscillator circuit is enabled by setting the CLGOSC1.OSC1BUP bit to 1, it uses the high-gain oscillation inverter for a set period of time (startup boosting operation) after the oscillator circuit is enabled (by setting the CLGOSC.OSC1EN bit to 1) to reduce oscillation start time. Note, however, that the oscillation operation may become unstable if there is a large gain differential between normal operation and startup boosting operation. Furthermore, the oscillation start time being actually reduced depends on the characteristics of the resonator used. Figure 2.3.4.2 shows an operation example when the oscillation startup control circuit is used.

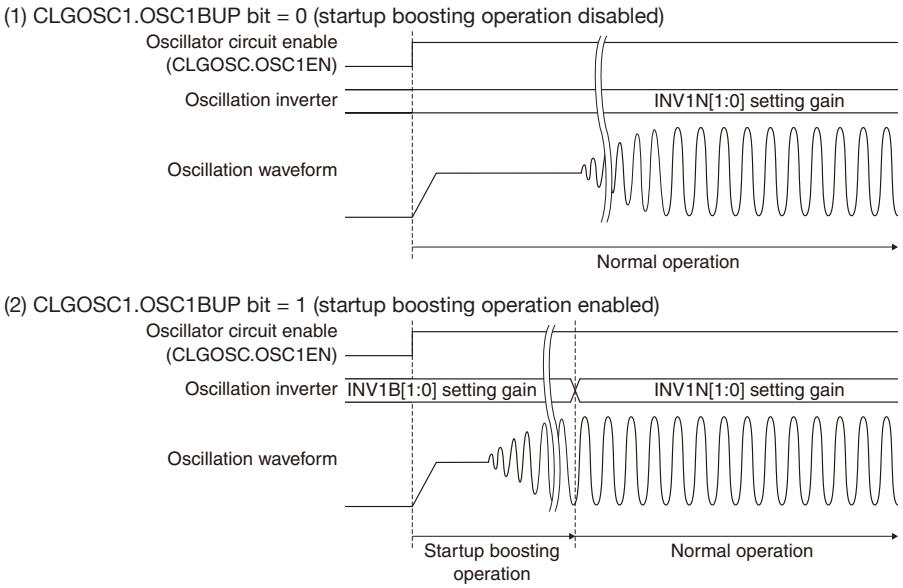


Figure 2.3.4.2 Operation Example when the OSC1 Crystal Oscillation Startup Control Circuit is Used

Oscillation start procedure for the IOSC oscillator circuit

Follow the procedure shown below to start oscillation of the IOSC oscillator circuit.

1. Write 1 to the CLGINTF.IOSCSTAIF bit. (Clear interrupt flag)
2. Write 1 to the CLGINTE.IOSCSTAIE bit. (Enable interrupt)
3. Write 1 to the CLGOSC.IOSCEN bit. (Start oscillation)
4. IOSCCLK can be used if the CLGINTF.IOSCSTAIF bit = 1 after an interrupt occurs.

Oscillation start procedure for the OSC1 oscillator circuit

Follow the procedure shown below to start oscillation of the OSC1 oscillator circuit.

1. Write 1 to the CLGINTF.OSC1STAIF bit. (Clear interrupt flag)
2. Write 1 to the CLGINTE.OSC1STAIE bit. (Enable interrupt)
3. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)

4. Configure the following CLGOSC1 register bits:
   - CLGOSC1.OSC1SELCR bit (Select oscillator type)
   - CLGOSC1.OSC1WT[1:0] bits (Set oscillation stabilization waiting time)

   In addition to the above, configure the following bits when using the crystal oscillator:
   - CLGOSC1.INV1N[1:0] bits (Set oscillation inverter gain)
   - CLGOSC1.CGI1[2:0] bits (Set internal gate capacitor)
   - CLGOSC1.INV1B[1:0] bits (Set oscillation inverter gain for startup boosting period)
   - CLGOSC1.OSC1BUP bit (Enable/disable oscillation startup control circuit)
5. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)
6. Write 1 to the CLGOSC.OSC1EN bit. (Start oscillation)
7. OSC1CLK can be used if the CLGINTF.OSC1STAIF bit = 1 after an interrupt occurs.

The setting values of the CLGOSC1.INV1N[1:0], CLGOSC1.CGI1[2:0], CLGOSC1.OSC1WT[1:0], and CLGOSC1.INV1B[1:0] bits should be determined after performing evaluation using the populated circuit board.

Note: Make sure the CLGOSC.OSC1EN bit is set to 0 (while the OSC3 oscillation is halted) when switching the oscillator within two types.

Oscillation start procedure for the OSC3 oscillator circuit

Follow the procedure shown below to start oscillation of the OSC3 oscillator circuit.

1. Write 1 to the CLGINTF.OSC3STAIF bit. (Clear interrupt flag)
2. Write 1 to the CLGINTE.OSC3STAIE bit. (Enable interrupt)
3. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
4. Configure the following CLGOSC3 register bits:
   - CLGOSC3.OSC3FQ bit (Select oscillation frequency)
   - CLGOSC3.OSC3WT[2:0] bits (Set oscillation stabilization waiting time)
5. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)
6. Write 1 to the CLGOSC.OSC3EN bit. (Start oscillation)
7. OSC3CLK can be used if the CLGINTF.OSC3STAIF bit = 1 after an interrupt occurs.

Note: Before changing the frequency when OSC3 has been selected as the system clock, make sure that the system clock source is switched to a clock other than OSC3 and the CLGOSC.OSC3EN bit is set to 0 (oscillation stop status).

System clock switching

The CPU boots using IOSCCLK as SYSCLK. After booting, the clock source of SYSCLK can be switched according to the processing speed required. The SYSCLK frequency can also be set by selecting the clock source division ratio, this makes it possible to run the CPU at the most suitable performance for the process to be executed. The CLGSCLK.CLKSRC[1:0] and CLGSCLK.CLKDIV[1:0] bits are used for this control.
The CLGSCLK register bits are protected against writings by the system protect function, therefore, the system protection must be removed by writing 0x0096 to the MSCPROT.PROT[15:0] bits before the register setting can be altered. For the transition between the operating modes including the system clock switching, refer to "Operating Mode."

Clock control in SLEEP mode

The CPU enters SLEEP mode when it executes the slp instruction. Whether the clock sources being operated are stopped or not at this point can be selected in each source individually. This allows the CPU to fast switch between SLEEP mode and RUN mode, and the peripheral circuits to continue operating without disabling the clock in SLEEP mode. The CLGOSC.IOSCSLPC, CLGOSC.OSC1SLPC, CLGOSC.OSC3SLPC, and CLGOSC.EXOSCSLPC bits are used for this control. Figure 2.3.4.3 shows a control example.

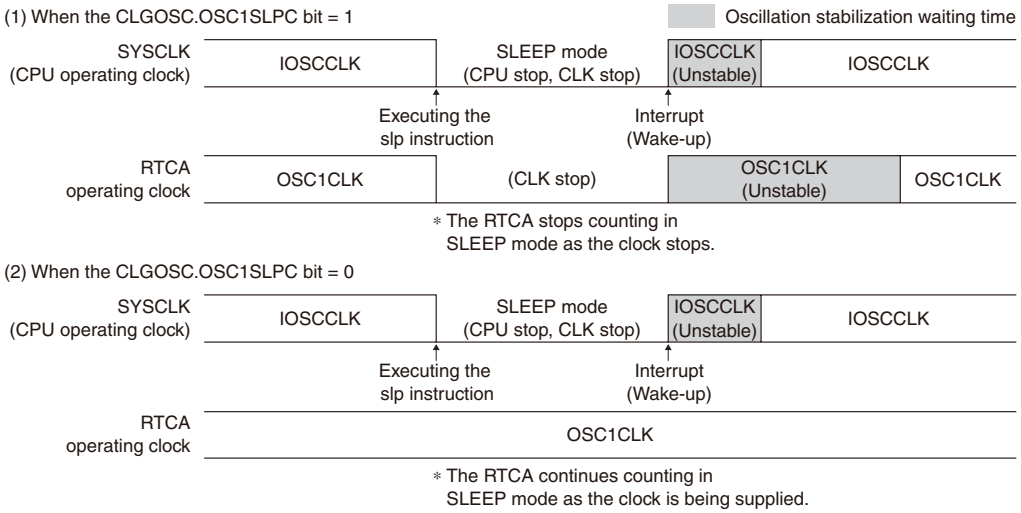


Figure 2.3.4.3 Clock Control Example in SLEEP Mode

The SYSCLK condition (clock source and division ratio) at wake-up from SLEEP mode to RUN mode can also be configured. This allows flexible clock control according to the wake-up process. Configure the clock using the CLGSCLK.WUPSRC[1:0] and CLGSCLK.WUPDIV[1:0] bits, and write 1 to the CLGSCLK.WUPMD bit to enable this function.

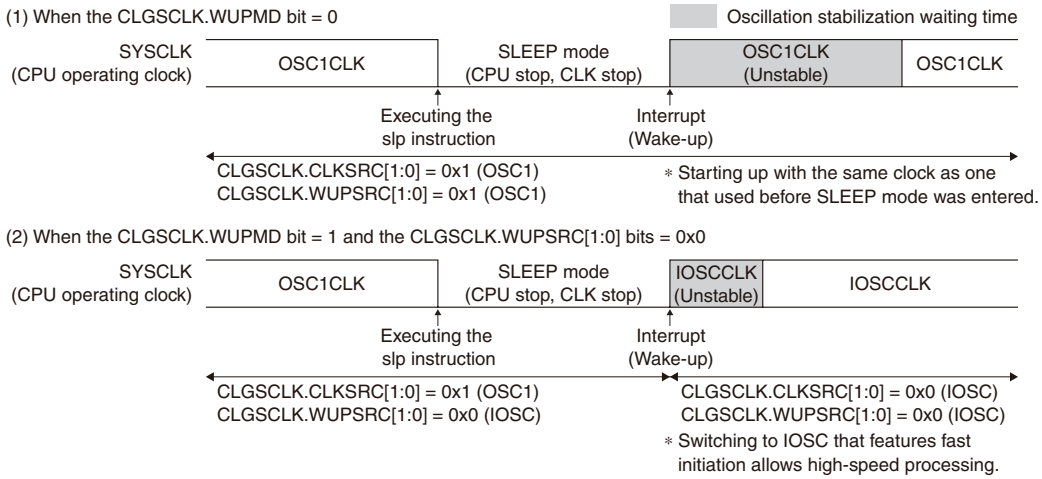


Figure 2.3.4.4 Clock Control Example at SLEEP Cancelation

Clock external output (FOUT)

The FOUT pin can output the clock generated by a clock source or its divided clock to outside the IC. This allows monitoring the oscillation frequency of the oscillator circuit or supplying an operating clock to external ICs. Follow the procedure shown below to start clock external output.

1. Assign the FOUT function to the port (refer to the "I/O Ports" and "Universal Port Multiplexer" chapters).
2. Configure the following CLGFOUT register bits:
   - CLGFOUT.FOUTSRC[1:0] bits (Select clock source)
   - CLGFOUT.FOUTDIV[2:0] bits (Set clock division ratio)
   - Set the CLGFOUT.FOUTEN bit to 1. (Enable clock external output)

OSC1 oscillation stop detection function

The oscillation stop detection function restarts the OSC1 oscillator circuit when it detects oscillation stop under adverse environments that may stop the oscillation. Follow the procedure shown below to enable the oscillation stop detection function.

1. After enabling the OSC1 oscillation, check if the stabilized clock is supplied (CLGINTF.OSC1STAIF bit = 1).
2. Write 1 to the CLGINTF.OSC1STPIF bit. (Clear interrupt flag)
3. Write 1 to the CLGINTE.OSC1STPIE bit. (Enable interrupt)
4. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
5. Set the following CLGOSC1 register bits:
   - Set the CLGOSC1.OSDRB bit to 1. (Enable OSC1 restart function)
   - Set the CLGOSC1.OSDEN bit to 1. (Enable oscillation stop detection function)
6. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)
7. The OSC1 oscillation stops if the CLGINTF.OSC1STPIF bit = 1 after an interrupt occurs. If the CLGOSC1.OSDRB bit = 1, the hardware restarts the OSC1 oscillator circuit.

Note: Enabling the oscillation stop detection function increase the oscillation stop detector current (I_{OSD1}).

2.4 Operating Mode

2.4.1 Initial Boot Sequence

Figure 2.4.1.1 shows the initial boot sequence after power is turned on.

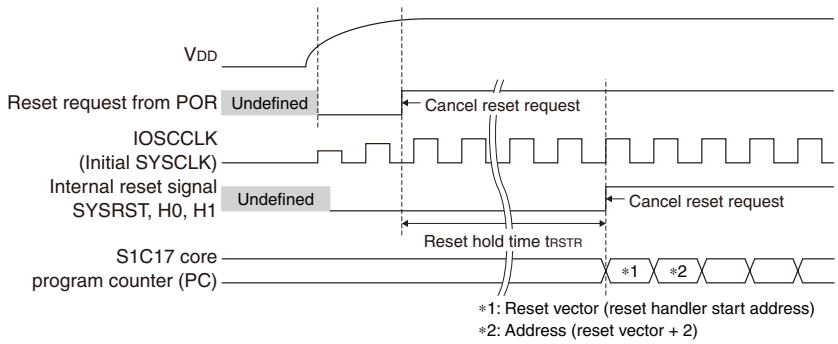


Figure 2.4.1.1 Initial Boot Sequence

Note: The reset cancelation time at power-on varies according to the power rise time and reset request cancelation time.

For the reset hold time t_{RSTR}, refer to "Reset hold circuit characteristics" in the "Electrical Characteristics" chapter.

2.4.2 Transition between Operating Modes

State transitions between operating modes shown in Figure 2.4.2.1 take place in this IC.

RUN mode

RUN mode refers to the state in which the CPU is executing the program. A transition to this mode takes place when the system reset request from the system reset controller is canceled. RUN mode is classified into "IOSC RUN," "OSC1 RUN," "OSC3 RUN," and "EXOSC RUN" by the SYSCLK clock source.

HALT mode

When the CPU executes the halt instruction, it suspends program execution and stops operating. This state is HALT mode. In this mode, the clock sources and peripheral circuits keep operating. This mode can be set while no software processing is required and it reduces power consumption as compared with RUN mode. HALT mode is classified into "IOSC HALT," "OSC1 HALT," "OSC3 HALT," and "EXOSC HALT" by the SYSCLK clock source.

SLEEP mode

When the CPU executes the slp instruction, it suspends program execution and stops operating. This state is SLEEP mode. In this mode, the clock sources stop operating as well. However, the clock source in which the CLGOSC.IOSCSLPC/OSC1SLPC/OSC3SLPC/EXOSCSLPC bit is set to 0 keeps operating, so the peripheral circuits with the clock being supplied can also operate. By setting this mode when no software processing and peripheral circuit operations are required, power consumption can be less than HALT mode.

Note: The current consumption when a clock source is active in SLEEP mode by setting the CLGOSC.IOSCSLPC/OSC1SLPC/OSC3SLPC/EXOSCSLPC bit to 0 is equivalent to the value in HALT mode with the same clock source condition (refer to "Current Consumption, Current consumption in HALT mode I_{HALT1}, I_{HALT2}, and I_{HALT3}" in the "Electrical Characteristics" chapter).

DEBUG mode

When a debug interrupt occurs, the CPU enters DEBUG mode. DEBUG mode is canceled when the retd instruction is executed. For more information on DEBUG mode, refer to "Debugger" in the "CPU and Debugger" chapter.

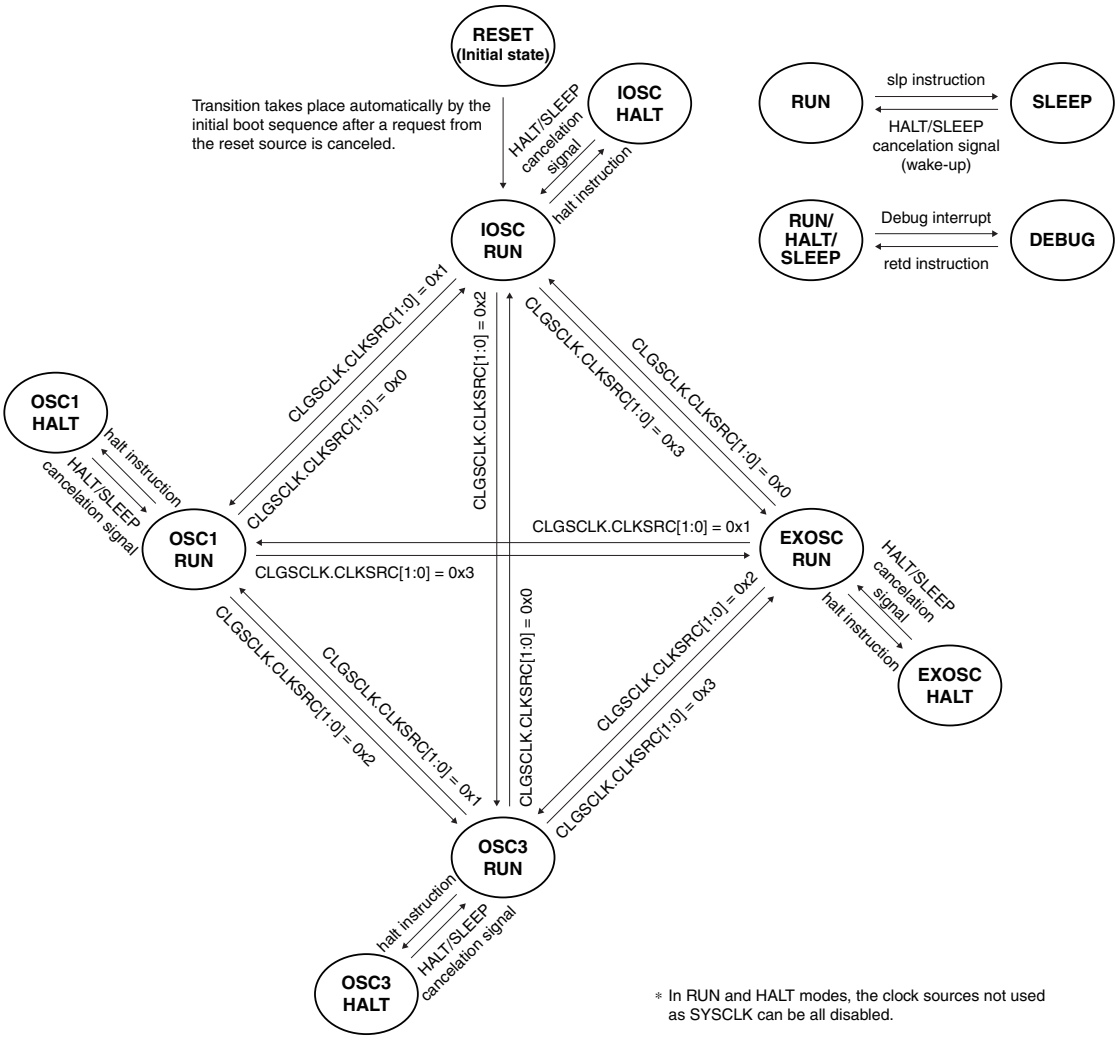


Figure 2.4.2.1 Operating Mode-to-Mode State Transition Diagram

Canceling HALT or SLEEP mode

The conditions listed below generate the HALT/SLEEP cancelation signal to cancel HALT or SLEEP mode and put the CPU into RUN mode. This transition is executed even if the CPU does not accept the interrupt request.

- Interrupt request from a peripheral circuit
- NMI from the watchdog timer
- Debug interrupt
- Reset request

2.5 Interrupts

CLG has a function to generate the interrupts shown in Table 2.5.1.

Table 2.5.1 CLG Interrupt Functions

Interrupt	Interrupt flag	Set condition	Clear condition
IOSC oscillation stabilization waiting completion	CLGINTF.IOSCSTAIF	When the IOSC oscillation stabilization waiting operation has completed after the oscillation starts	Writing 1
OSC1 oscillation stabilization waiting completion	CLGINTF.OSC1STAIF	When the OSC1 oscillation stabilization waiting operation has completed after the oscillation starts	Writing 1
OSC3 oscillation stabilization waiting completion	CLGINTF.OSC3STAIF	When the OSC3 oscillation stabilization waiting operation has completed after the oscillation starts	Writing 1
OSC1 oscillation stop	CLGINTF.OSC1STPIF	When OSC1CLK is stopped, or when the CLGOSC.OSC1EN or CLGOSC1.OSDEN bit setting is altered from 1 to 0.	Writing 1

CLG provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the "Interrupt Controller" chapter.

2.6 Control Registers

PWG V_{D1} Regulator Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PWGVD1CTL	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1–0	REGMODE[1:0]	0x0	H0	R/WP	

Bits 15–2 Reserved

Bits 1–0 REGMODE[1:0]

These bits control the internal regulator operating mode.

Table 2.6.1 Internal Regulator Operating Mode

PWGVD1CTL.REGMODE[1:0] bits	Operating mode
0x3	Economy mode
0x2	Normal mode
0x1	Reserved
0x0	Automatic mode

CLG System Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGSCLK	15	WUPMD	0	H0	R/WP	–
	14	–	0	–	R	
	13–12	WUPDIV[1:0]	0x0	H0	R/WP	
	11–10	–	0x0	–	R	
	9–8	WUPSRC[1:0]	0x0	H0	R/WP	
	7–6	–	0x0	–	R	
	5–4	CLKDIV[1:0]	0x0	H0	R/WP	
	3–2	–	0x0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/WP	

Bit 15 WUPMD

This bit enables the SYSCLK switching function at wake-up.
1 (R/WP): Enable
0 (R/WP): Disable

When the CLGSCLK.WUPMD bit = 1, setting values of the CLGSCLK.WUPSRC[1:0] bits and the CLGSCLK.WUPDIV[1:0] bits are loaded to the CLGSCLK.CLKSRC[1:0] bits and the CLGSCLK.CLKDIV[1:0] bits, respectively, at wake-up from SLEEP mode to switch SYSCLK. When the CLGSCLK.WUPMD bit = 0, the CLGSCLK.CLKSRC[1:0] and CLGSCLK.CLKDIV[1:0] bits are not altered at wake-up.

Note: When the CLGSCLK.WUPMD bit = 1, the clock source enable bits (CLGOSC.EXOSCEN, CLGOSC.OSC1EN, CLGOSC.OSC3EN, CLGOSC.IOSCEN) except for the SYSCLK source selected by the CLGSCLK.CLKSRC[1:0] bits will be cleared to 0 to stop the clocks after a system wake-up. However, the enable bit of the clock source being operated during SLEEP mode by setting the CLGOSC.****SLPC bit retains 1 after a wake-up.

Bit 14 Reserved

Bits 13–12 WUPDIV[1:0]

These bits select the SYSCLK division ratio for resetting the CLGSCLK.CLKDIV[1:0] bits at wake-up. This setting is ineffective when the CLGSCLK.WUPMD bit = 0.

Bits 11–10 Reserved

Bits 9–8 WUPSRC[1:0]

These bits select the SYSCLK clock source for resetting the CLGSCLK.CLKSRC[1:0] bits at wake-up. When a currently stopped clock source is selected, it will automatically start oscillating or clock input at wake-up. However, this setting is ineffective when the CLGSCLK.WUPMD bit = 0.

Table 2.6.2 SYSCLK Clock Source and Division Ratio Settings at Wake-up

CLGSCLK. WUPDIV[1:0] bits	CLGSCLK.WUPSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSCCLK	OSC1CLK	OSC3CLK	EXOSCCLK
0x3	1/8	Reserved	1/8	Reserved
0x2	1/4	Reserved	1/4	Reserved
0x1	1/2	1/2	1/2	Reserved
0x0	1/1	1/1	1/1	1/1

Bits 7–6 Reserved

Bits 5–4 CLKDIV[1:0]

These bits set the division ratio of the clock source to determine the SYSCLK frequency.

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the SYSCLK clock source.
When a currently stopped clock source is selected, it will automatically start oscillating or clock input.

Table 2.6.3 SYSCLK Clock Source and Division Ratio Settings

CLGSCLK. CLKDIV[1:0] bits	CLGSCLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSCCLK	OSC1CLK	OSC3CLK	EXOSCCLK
0x3	1/8	Reserved	1/8	Reserved
0x2	1/4	Reserved	1/4	Reserved
0x1	1/2	1/2	1/2	Reserved
0x0	1/1	1/1	1/1	1/1

CLG Oscillation Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGOSC	15–12	–	0x0	–	R	–
	11	EXOSCSLPC	1	H0	R/W	
	10	OSC3SLPC	1	H0	R/W	
	9	OSC1SLPC	1	H0	R/W	
	8	IOSCSLPC	1	H0	R/W	
	7–4	–	0x0	–	R	
	3	EXOSCEN	0	H0	R/W	
	2	OSC3EN	0	H0	R/W	
	1	OSC1EN	0	H0	R/W	
	0	IOSCEN	1	H0	R/W	

Bits 15–12 Reserved

Bit 11 EXOSCSLPC
Bit 10 OSC3SLPC
Bit 9 OSC1SLPC
Bit 8 IOSCSLPC

These bits control the clock source operations in SLEEP mode.
1 (R/W): Stop clock source in SLEEP mode
0 (R/W): Continue operation state before SLEEP

Each bit corresponds to the clock source as follows:
CLGOSC.EXOSCSLPC bit: EXOSC clock input
CLGOSC.OSC3SLPC bit: OSC3 oscillator circuit
CLGOSC.OSC1SLPC bit: OSC1 oscillator circuit
CLGOSC.IOSCSLPC bit: IOSC oscillator circuit

Bits 7–4 Reserved

Bit 3 EXOSCEN
Bit 2 OSC3EN
Bit 1 OSC1EN
Bit 0 IOSCEN

These bits control the clock source operation.
1(R/W): Start oscillating or clock input
0(R/W): Stop oscillating or clock input

Each bit corresponds to the clock source as follows:
CLGOSC.EXOSCEN bit: EXOSC clock input
CLGOSC.OSC3EN bit: OSC3 oscillator circuit
CLGOSC.OSC1EN bit: OSC1 oscillator circuit
CLGOSC.IOSCEN bit: IOSC oscillator circuit

CLG OSC1 Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGOSC1	15	–	0	–	R	–
	14	OSDRB	1	H0	R/WP	
	13	OSDEN	0	H0	R/WP	
	12	OSC1BUP	1	H0	R/WP	
	11	OSC1SELCR	0	H0	R/WP	
	10–8	CGI1[2:0]	0x0	H0	R/WP	
	7–6	INV1B[1:0]	0x2	H0	R/WP	
	5–4	INV1N[1:0]	0x1	H0	R/WP	
	3–2	–	0x0	–	R	
	1–0	OSC1WT[1:0]	0x2	H0	R/WP	

Bit 15 Reserved

Bit 14 OSDRB

This bit enables the OSC1 oscillator circuit restart function by the oscillation stop detector when OSC1 crystal oscillation stop is detected.
1 (R/WP): Enable (Restart the OSC1 oscillator circuit when oscillation stop is detected.)
0 (R/WP): Disable

Bit 13 OSDEN

This bit controls the oscillation stop detector in the OSC1 oscillator circuit.
1 (R/WP): OSC1 oscillation stop detector on
0 (R/WP): OSC1 oscillation stop detector off

Note: Do not write 1 to the CLGOSC1.OSDEN bit before stabilized OSC1CLK is supplied. Furthermore, the CLGOSC1.OSDEN bit should be set to 0 when the CLGOSC.OSC1EN bit is set to 0.

Bit 12 OSC1BUP

This bit enables the oscillation startup control circuit in the OSC1 crystal oscillator circuit.
1 (R/WP): Enable (Activate booster operation at startup.)
0 (R/WP): Disable

Bit 11 OSC1SELCR

This bit selects an oscillator type of the OSC1 oscillator circuit.
1 (R/WP): Internal oscillator
0 (R/WP): Crystal oscillator

Bits 10–8 CGI1[2:0]

These bits set the internal gate capacitance in the OSC1 crystal oscillator circuit.

Table 2.6.4 OSC1 Internal Gate Capacitance Setting

CLGOSC1.CGI1[2:0] bits	Capacitance
0x7	Max.
0x6	↑
0x5	
0x4	
0x3	
0x2	
0x1	↓
0x0	Min.

For more information, refer to “OSC1 oscillator circuit characteristics, Crystal oscillator internal gate capacitance C_{GI1C}” in the “Electrical Characteristics” chapter.

Bits 7–6 INV1B[1:0]

These bits set the oscillation inverter gain that will be applied at boost startup of the OSC1 crystal oscillator circuit.

Table 2.6.5 Setting Oscillation Inverter Gain at OSC1 Boost Startup

CLGOSC1.INV1B[1:0] bits	Inverter gain
0x3	Max.
0x2	↑
0x1	↓
0x0	Min.

Note: The CLGOSC1.INV1B[1:0] bits must be set to a value equal to or larger than the CLGOSC1.INV1N[1:0] bits.

Bits 5–4 INV1N[1:0]

These bits set the oscillation inverter gain applied at normal operation of the OSC1 crystal oscillator circuit.

Table 2.6.6 Setting Oscillation Inverter Gain at OSC1 Normal Operation

CLGOSC1.INV1N[1:0] bits	Inverter gain
0x3	Max.
0x2	↑
0x1	↓
0x0	Min.

Bits 3–2 Reserved

Bits 1–0 OSC1WT[1:0]

These bits set the oscillation stabilization waiting time for the OSC1 oscillator circuit.

Table 2.6.7 OSC1 Oscillation Stabilization Waiting Time Setting

CLGOSC1.OSC1WT[1:0] bits	Oscillation stabilization waiting time
0x3	65,536 clocks
0x2	16,384 clocks
0x1	4,096 clocks
0x0	Reserved

CLG OSC3 Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGOSC3	15–11	–	0x00	–	R	–
	10	OSC3FQ	0	–	R/WP	
	9-8	–	0x0	–	R	
	7–3	–	0x00	–	R	
	2–0	OSC3WT[2:0]	0x2	H0	R/WP	

Bits 15–11 Reserved

Bit 10 OSC3FQ

This bit sets the oscillation frequency of the OSC3 internal oscillator circuit.
1 (R/WP): 6.4 MHz
0 (R/WP): 3.2 MHz

Bits 9–3 Reserved

Bits 2–0 OSC3WT[2:0]

These bits set the oscillation stabilization waiting time for the OSC3 oscillator circuit.

Table 2.6.8 OSC3 Oscillation Stabilization Waiting Time Setting

CLGOSC3.OSC3WT[2:0] bits	Oscillation stabilization waiting time
0x7	65,536 clocks
0x6	16,384 clocks
0x5	4,096 clocks
0x4	1,024 clocks
0x3	256 clocks
0x2	64 clocks
0x1	16 clocks
0x0	4 clocks

CLG Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGINTF	15–8	–	0x00	–	R	–
	7–6	–	0x0	H0	R	
	5	OSC1STPIF	0	H0	R/W	Cleared by writing 1.
	4–3	–	0x0	–	R	–
	2	OSC3STAIF	0	H0	R/W	Cleared by writing 1.
	1	OSC1STAIF	0	H0	R/W	
	0	IOSCSTAIF	0	H0	R/W	

Bits 15–6, 4–3 Reserved

Bit 5 OSC1STPIF
Bit 2 OSC3STAIF
Bit 1 OSC1STAIF
Bit 0 IOSCSTAIF

These bits indicate the CLG interrupt cause occurrence statuses.
1 (R): Cause of interrupt occurred
0 (R): No cause of interrupt occurred
1 (W): Clear flag
0 (W): Ineffective

Each bit corresponds to the interrupt as follows:
CLGINTF.OSC1STPIF bit: OSC1 oscillation stop interrupt
CLGINTF.OSC3STAIF bit: OSC3 oscillation stabilization waiting completion interrupt
CLGINTF.OSC1STAIF bit: OSC1 oscillation stabilization waiting completion interrupt
CLGINTF.IOSCSTAIF bit: IOSC oscillation stabilization waiting completion interrupt

Note: The CLGINTF.IOSCSTAIF bit is 0 after system reset is canceled, but IOSCCLK has already been stabilized.

CLG Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGINTE	15–8	–	0x00	–	R	–
	7–6	–	0x0	–	R	
	5	OSC1STPIE	0	H0	R/W	
	4–3	–	0x0	–	R	
	2	OSC3STAIE	0	H0	R/W	
	1	OSC1STAIE	0	H0	R/W	
	0	IOSCSTAIE	0	H0	R/W	

Bits 15–6, 4–3 Reserved

Bit 5 OSC1STPIE
Bit 2 OSC3STAIE
Bit 1 OSC1STAIE
Bit 0 IOSCSTAIE

These bits enable the CLG interrupts.
1 (R/W): Enable interrupts
0 (R/W): Disable interrupts

Each bit corresponds to the interrupt as follows:
CLGINTE.OSC1STPIE bit: OSC1 oscillation stop interrupt
CLGINTE.OSC3STAIE bit: OSC3 oscillation stabilization waiting completion interrupt
CLGINTE.OSC1STAIE bit: OSC1 oscillation stabilization waiting completion interrupt
CLGINTE.IOSCSTAIE bit: IOSC oscillation stabilization waiting completion interrupt

CLG FOUT Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGFOUT	15–8	–	0x00	–	R	–
	7	–	0	–	R	
	6–4	FOUTDIV[2:0]	0x0	H0	R/W	
	3–2	FOUTSRC[1:0]	0x0	H0	R/W	
	1	–	0	–	R	
	0	FOUTEN	0	H0	R/W	

Bits 15–7 Reserved

Bits 6–4 FOUTDIV[2:0]

These bits set the FOUT clock division ratios.

Bits 3–2 FOUTSRC[1:0]

These bits select the FOUT clock sources.

Table 2.6.9 FOUT Clock Source and Division Ratio Settings

CLGFOUT. FOUTDIV[2:0] bits	CLGFOUT.FOUTSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSCCLK	OSC1CLK	OSC3CLK	SYSCLK
0x7	1/128	1/32,768	1/128	Reserved
0x6	1/64	1/4,096	1/64	Reserved
0x5	1/32	1/1,024	1/32	Reserved
0x4	1/16	1/256	1/16	Reserved
0x3	1/8	1/8	1/8	Reserved
0x2	1/4	1/4	1/4	Reserved
0x1	1/2	1/2	1/2	Reserved
0x0	1/1	1/1	1/1	1/1

Note: When the CLGFOUT.FOUTSRC[1:0] bits are set to 0x3, the FOUT output will be stopped in SLEEP/HALT mode as SYSCLK is stopped.

Bit 1 Reserved

Bit 0 FOUTEN

This bit controls the FOUT clock external output.

1 (R/W): Enable external output

0 (R/W): Disable external output

Note: Since the FOUT signal generated is out of sync with writings to the CLGFOUT.FOUTEN bit, a glitch may occur when the FOUT output is enabled or disabled.

CLG Oscillation Frequency Trimming Register 1

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGTRIM1	15–14	–	0x0	–	R	–
	13–8	OSC3AJ[5:0]	*	H0	R/WP	* Determined by factory adjustment.
	7–6	–	0x0	–	R	–
	5–0	IOSCAJ[5:0]	*	H0	R/WP	* Determined by factory adjustment.

Bits 15–14 Reserved

Bits 13–8 OSC3AJ[5:0]

These bits set the frequency trimming value for the OSC3 internal oscillator circuit.

Table 2.6.10 Oscillation Frequency Trimming Setting of OSC3 Internal Oscillator Circuit

CLGTRIM1.OSC3AJ[5:0] bits	OSC3 internal oscillator frequency
0x3f	High
:	:
0x00	Low

Note: The initial value of the CLGTRIM1.OSC3AJ[5:0] bits was adjusted so that the OSC3 oscillator circuit characteristics described in the "Electrical Characteristics" chapter can be guaranteed. Be aware that the frequency characteristic may not be satisfied when this setting is altered. When altering this setting, always make sure that the OSC3 oscillator circuit is inactive.

Bits 7–6 Reserved

Bits 5–0 IOSCAJ[5:0]

These bits set the frequency trimming value for the IOSC internal oscillator circuit.

Table 2.6.11 Oscillation Frequency Trimming Setting of IOSC Internal Oscillator Circuit

CLGTRIM1.IOSCAJ[5:0] bits	IOSC oscillation frequency
0x3f	High
:	:
0x00	Low

Note: The initial value of the CLGTRIM1.IOSCAJ[5:0] bits was adjusted so that the IOSC oscillator circuit characteristics described in the "Electrical Characteristics" chapter can be guaranteed. Be aware that the frequency characteristics may not be satisfied when this setting is altered. When altering this setting, always make sure that the IOSC oscillator circuit is inactive.

CLG Oscillation Frequency Trimming Register 2

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGTRIM2	15–8	–	0x00	–	R	–
	7–6	–	0x0	–	R	
	5–0	OSC1AJ[5:0]	*	H0	R/WP	* Determined by factory adjustment.

Bits 15–6 Reserved

Bits 5–0 OSC1AJ[5:0]

These bits set the frequency trimming value for the OSC1 internal oscillator circuit.
This setting does not affect the OSC1 crystal oscillation frequency.

Table 2.6.12 Oscillation Frequency Trimming Setting of OSC1 Internal Oscillator Circuit

CLGTRIM2.OSC1AJ[5:0] bits	OSC1 internal oscillator frequency
0x3f	High
:	:
0x00	Low

Note: The initial value of the CLGTRIM2.OSC1AJ[5:0] bits was adjusted so that the OSC1 oscillator circuit characteristics described in the "Electrical Characteristics" chapter can be guaranteed. Be aware that the frequency characteristic may not be satisfied when this setting is altered. When altering this setting, always make sure that the OSC1 oscillator circuit is inactive.

3 CPU and Debugger

3.1 Overview

This IC incorporates the Seiko Epson original 16-bit CPU core (S1C17) with a debugger. The main features of the CPU core are listed below.

- Seiko Epson original 16-bit RISC processor
  - 24-bit general-purpose registers: 8
  - 24-bit special registers: 2
  - 8-bit special register: 1
  - Up to 16M bytes of memory space (24-bit address)
  - Harvard architecture using separated instruction bus and data bus
- Compact and fast instruction set optimized for development in C language
  - Code length: 16-bit fixed length
  - Number of instructions: 111 basic instructions (184 including variations)
  - Execution cycle: Main instructions are executed in one cycle.
  - Extended immediate instructions: Immediate data can be extended up to 24 bits.
- Supports reset, NMI, address misaligned, debug, and external interrupts.
  - Reads a vector from the vector table and branches to the interrupt handler routine directly.
  - Can generate software interrupts with a vector number specified (all vector numbers specifiable).
- HALT mode (halt instruction) and SLEEP mode (slp instruction) are provided as the standby function.
- Incorporates a debugger with three-wire communication interface to assist in software development.

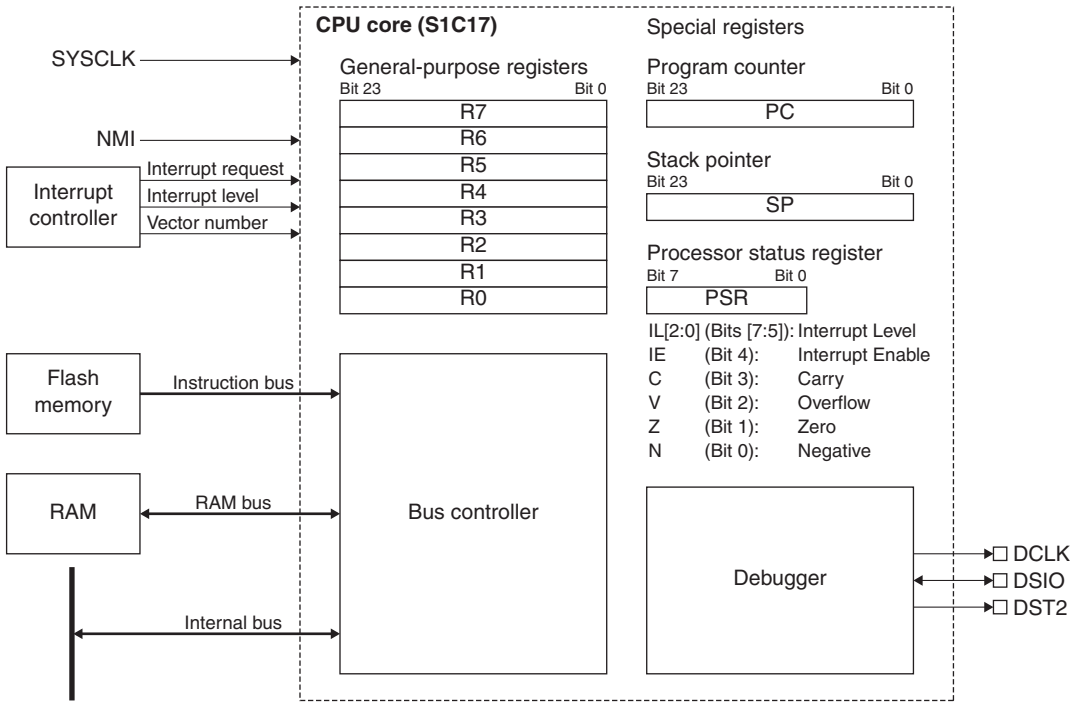


Figure 3.1.1 S1C17 Configuration

3.2 CPU Core

3.2.1 CPU Registers

The CPU includes eight general-purpose registers and three special registers (Table 3.2.1.1).

Table 3.2.1.1 Initialization of CPU Registers

CPU register name			Initial	Reset
General-purpose registers		R0 to R7	0x000000	H0
Special registers	Program counter	PC	The reset vector is automatically loaded.	H0
	Stack pointer	SP	0x000000	H0
	Processor status register	PSR	0x00	H0

For details on the CPU registers, refer to the "S1C17 Family S1C17 Core Manual." For more information on the reset vector, refer to the "Interrupt Controller" chapter.

3.2.2 Instruction Set

The CPU instruction codes are all fixed to 16 bits in length which, combined with pipelined processing, allows the most important instructions to be executed in one cycle. For details on the instructions, refer to the "S1C17 Family S1C17 Core Manual."

3.2.3 Reading PSR

The PSR contents can be read through the MSCPSR register. Note, however, that data cannot be written to PSR through the MSCPSR register.

3.2.4 I/O Area Reserved for the S1C17 Core

The address range from 0xfffc00 to 0xffffff is the I/O area reserved for the S1C17 core. Do not access this area except when it is required.

3.3 Debugger

3.3.1 Debugging Functions

The debugger provides the following functions:

- Instruction break: A debug interrupt is generated immediately before the set instruction address is executed. An instruction break can be set at up to four addresses.
- Single step: A debug interrupt is generated after each instruction has been executed.
- Forcible break: A debug interrupt is generated using an external input signal.
- Software break: A debug interrupt is generated when the brk instruction is executed.

When a debug interrupt occurs, the CPU enters DEBUG mode. The peripheral circuit operations in DEBUG mode depend on the setting of the DBRUN bit provided in the clock control register of each peripheral circuit. For more information on the DBRUN bit, refer to "Clock Supply in DEBUG Mode" in each peripheral circuit chapter. DEBUG mode continues until a cancel command is sent from the personal computer or the CPU executes the retd instruction. Neither hardware interrupts nor NMI are accepted during DEBUG mode.

3.3.2 Resource Requirements and Debugging Tools

Debugging work area

Debugging requires a 64-byte debugging work area. For more information on the work area location, refer to the "Memory and Bus" chapter. The start address of this debugging work area can be read from the DBRAM register.

Debugging tools

To perform debugging, connect ICDmini (S5U1C17001H) to the input/output pin for the debugger embedded in this IC and control it from the personal computer. This requires the tools shown below.

- S1C17 Family In-Circuit Debugger ICDmini (S5U1C17001H)
- S1C17 Family C Compiler Package (e.g., S5U1C17001C)

3.3.3 List of Debugger Input/Output Pins

Table 3.3.3.1 lists the debug pins.

Table 3.3.3.1 List of Debug Pins

Pin name	I/O	Initial state	Function
DCLK	O	O	On-chip debugger clock output pin Outputs a clock to the ICDmini (S5U1C17001H).
DSIO	I/O	I	On-chip debugger data input/output pin Used to input/output debugging data and input the break signal.
DST2	O	O	On-chip debugger status output pin Outputs the processor status during debugging.

The debugger input/output pins are shared with general-purpose I/O ports and are initially set as the debug pins. If the debugging function is not used, these pins can be switched to general-purpose I/O port pins. For details, refer to the "I/O Ports" chapter.

Notes: • Do not drive the DCLK pin with a high level from outside (e.g. pulling up with a resistor). Also, do not connect (short-circuit) between the DCLK pin and another GPIO port. In the both cases, the IC may not start up normally due to unstable pin input/output status at power on.

• Do not drive the DSIO pin with a low level from outside, as it generates a debug interrupt that puts the CPU into DEBUG mode.

3.3.4 External Connection

Figure 3.3.4.1 shows a connection example between this IC and ICDmini when performing debugging.

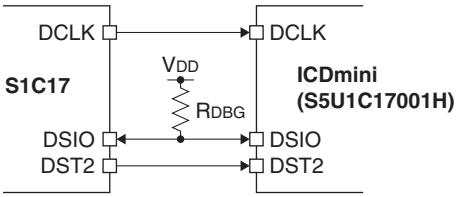


Figure 3.3.4.1 External Connection

For the recommended pull-up resistor value, refer to "Recommended Operating Conditions, DSIO pull-up resistor R_{DBG}" in the "Electrical Characteristics" chapter. R_{DBG} is not required when using the DSIO pin as a general-purpose I/O port pin.

3.3.5 Flash Security Function

This IC provides a security function to protect the internal Flash memory from unauthorized reading and tampering by using the debugger through ICDmini. Figure 3.3.5.1 shows a Flash security function setting flow.

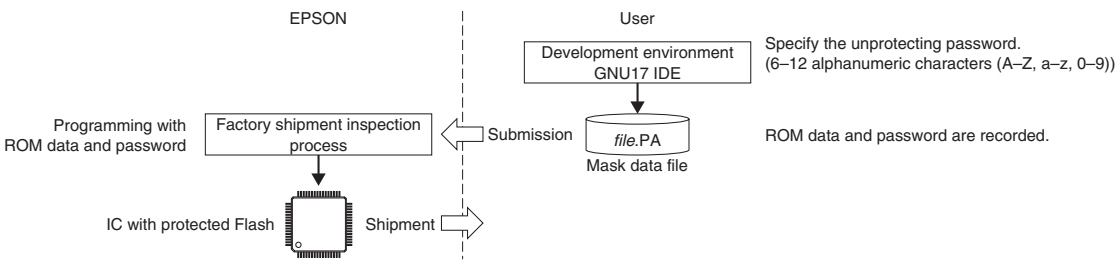


Figure 3.3.5.1 Shipment of IC with ROM Data Programmed and Flash Security Function Setting Flow

The following shows the status of the IC with protected Flash:

- The Flash memory data is undefined if it is read from the debugger.
- An error occurs if an attempt is made to program the Flash memory through ICDmini.

However, the Flash security function can be disabled by entering the unprotecting password predefined to GNU17 IDE (the security function will take effect again after a reset). For setting the password, refer to the "(S1C17 Family C Compiler Package) S5U1C17001C Manual."

Note: Disable the Flash security function before debugging an IC with protected Flash via ICDmini. The debugging functions may not run normally if the Flash security function is enabled.

3.4 Control Register

MISC PSR Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
MSCPSR	15–8	–	0x00	–	R	–
	7–5	PSRIL[2:0]	0x0	H0	R	
	4	PSRIE	0	H0	R	
	3	PSRC	0	H0	R	
	2	PSRV	0	H0	R	
	1	PSRZ	0	H0	R	
	0	PSRN	0	H0	R	

Bits 15–8 Reserved

Bits 7–5 PSRIL[2:0]

The value (0 to 7) of the PSR IL[2:0] (interrupt level) bits can be read out with these bits.

Bit 4 PSRIE

The value (0 or 1) of the PSR IE (interrupt enable) bit can be read out with this bit.

Bit 3 PSRC

The value (0 or 1) of the PSR C (carry) flag can be read out with this bit.

Bit 2 PSRV

The value (0 or 1) of the PSR V (overflow) flag can be read out with this bit.

Bit 1 PSRZ

The value (0 or 1) of the PSR Z (zero) flag can be read out with this bit.

Bit 0 PSRN

The value (0 or 1) of the PSR N (negative) flag can be read out with this bit.

Debug RAM Base Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DBRAM	31–24	–	0x00	–	R	–
	23–0	DBRAM[23:0]	*1	H0	R	

*1 Debugging work area start address

Bits 31–24 Reserved

Bits 23–0 DBRAM[23:0]

The start address of the debugging work area (64 bytes) can be read out with these bits.

4 Memory and Bus

4.1 Overview

This IC supports up to 16M bytes of accessible memory space for both instructions and data.
The features are listed below.

- Embedded Flash memory that supports on-board programming
- All memory and control registers are accessible in 16-bit width and one cycle (except for EEPROM).
- Write-protect function to protect system control registers

Figure 4.1.1 shows the memory map.

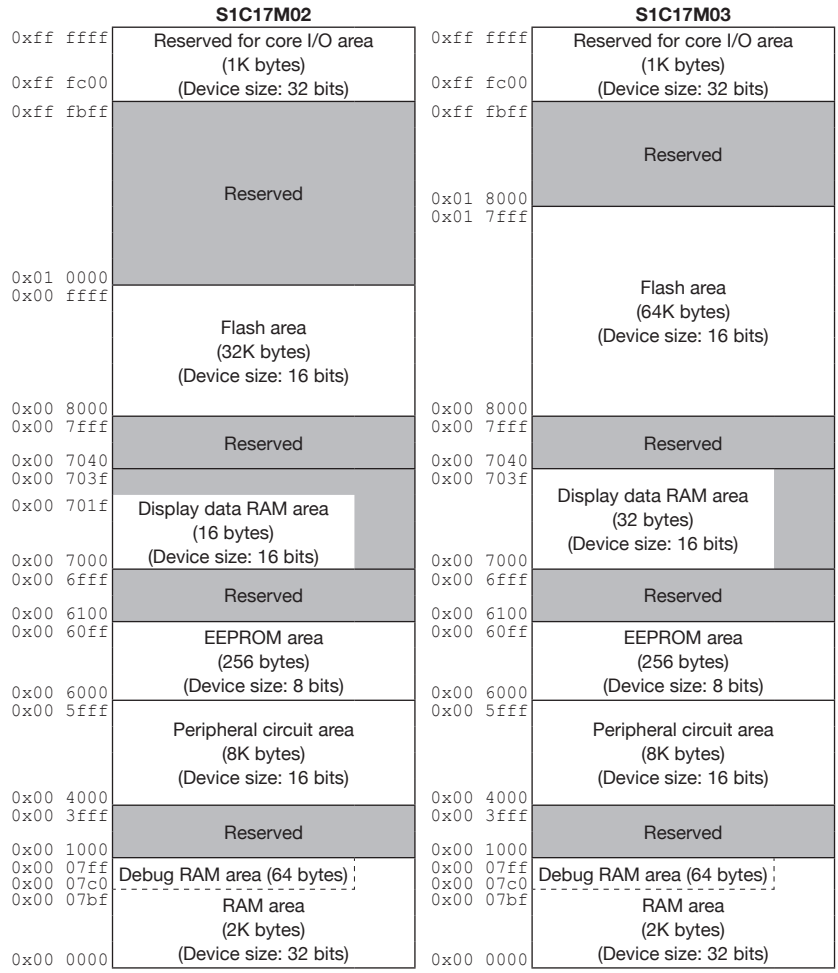


Figure 4.1.1 Memory Map

4.2 Bus Access Cycle

The CPU uses the system clock for bus access operations. First, "Bus access cycle," "Device size," and "Access size" are defined as follows:

- Bus access cycle: One system clock period = 1 cycle
- Device size: Bit width of the memory and peripheral circuits that can be accessed in one cycle
- Access size: Access size designated by the CPU instructions (e.g., ld %rd, [%rb] → 16-bit data transfer)

Table 4.2.1 lists numbers of bus access cycles by different device size and access size. The peripheral circuits can be accessed with an 8-bit, 16-bit, or 32-bit instruction.

Table 4.2.1 Number of Bus Access Cycles

Device size	Access size	Number of bus access cycles
8 bits	8 bits	1
	16 bits	2
	32 bits	4
16 bits	8 bits	1
	16 bits	1
	32 bits	2
32 bits	8 bits	1
	16 bits	1
	32 bits	1

Note: When data is transferred to a memory in 32-bit access, the eight high-order bits are written to the memory as 0x00 since the bit width of the S1C17 core general-purpose registers is 24 bits. Conversely when sending from a memory to a register, the eight high-order bits are ignored.
The CPU performs 32-bit access for stack operations in an interrupt handling. In this case, the CPU read/write 32-bit data that consists of the PSR value as the eight high-order bits and the return address as the 24 low-order bits. For more information, refer to the "S1C17 Family S1C17 Core Manual."

The CPU adopts Harvard architecture that allows simultaneous processing of an instruction fetch and a data access. However, they are not performed simultaneously under one of the conditions listed below. This prolongs the instruction fetch cycle for the number of data area bus cycles.

- When the CPU executes an instruction stored in the Flash area and accesses data in the Flash area
- When the CPU executes an instruction stored in the Flash area and accesses data in the display data RAM/EEPROM area
- When the CPU executes an instruction stored in the internal RAM/display data RAM area and accesses data in the internal RAM/display data RAM/EEPROM area

4.3 Flash Memory

The Flash memory is used to store application programs and data. Address 0x8000 in the Flash area is defined as the vector table base address by default, therefore a vector table must be located beginning from this address. For more information on the vector table, refer to "Vector Table" in the "Interrupt Controller" chapter.

4.3.1 Flash Memory Pin

Table 4.3.1.1 shows the Flash memory pin.

Table 4.3.1.1 Flash Memory Pin

Pin name	I/O	Initial status	Function
VPP	P	–	Flash programming power supply

For the VPP voltage, refer to "Recommended Operating Conditions, Flash programming voltage VPP" in the "Electrical Characteristics" chapter.

Note: Always leave the VPP pin open except when programming the Flash memory.

4.3.2 Flash Bus Access Cycle Setting

There is a limit of frequency to access the Flash memory with no wait cycle, therefore, the number of bus access cycles for reading must be changed according to the system clock frequency. The number of bus access cycles for reading can be configured using the FLASHCWAIT.RDWAIT[1:0] bits. Select a setting for higher frequency than the system clock.

4.3.3 Flash Programming

The Flash memory supports on-board programming, so it can be programmed with the ROM data by using the debugger through an ICDmini. Figure 4.3.3.1 shows connection diagrams for on-board programming.

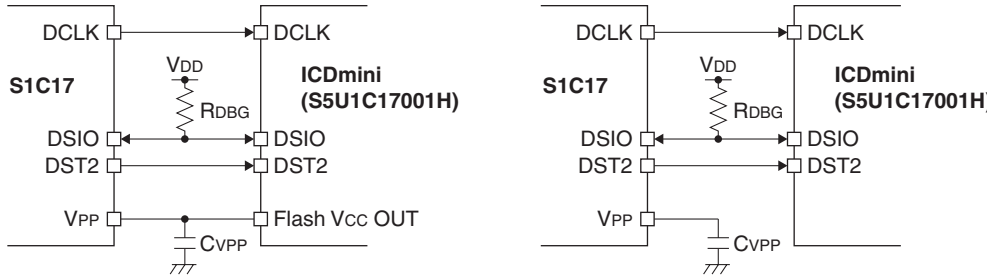


Figure 4.3.3.1 External Connection

The VPP pin must be left open except when programming the Flash memory. However, it is not necessary to disconnect the wire when using ICDmini to supply the VPP voltage, as ICDmini controls the power supply so that it will be supplied during Flash programming only. The VPP voltage can also be generated by the internal power supply for generating the Flash programming voltage. Be sure to connect CVPP for stabilizing the voltage when the VPP voltage is supplied externally or for generating the voltage when the internal power supply is used.
For detailed information on ROM data programming method, refer to the "(S1C17 Family C Compiler Package) S5U1C17001C Manual." The IC can also be shipped after being programmed in the factory with the ROM data developed. Should you desire to ship the IC with ROM data programmed from the factory, please contact our customer support.

Notes: • When programming the Flash memory, 2.2 V or more VDD voltage is required.

• Be sure to avoid using the VPP pin output for driving external circuits when the VPP voltage is generated internally.

4.4 EEPROM

This MCU includes an EEPROM that can be reprogrammed in one-byte units. This EEPROM supports 8-bit reading only, therefore, no instruction code can be stored.

4.4.1 EEPROM Pin

Table 4.4.1.1 shows the EEPROM pin.

Table 4.4.1.1 EEPROM Pin

Pin name	I/O	Initial status	Function
VPP	P	–	EEPROM programming power supply

When reprogramming the EEPROM, the EEPROM controller (EEPROMC) uses the VPP voltage generated internally. Connect CVPP to the VPP pin as shown in "(2) When VPP is generated internally" of Figure 4.3.3.1. The notes described in Section 4.3.3 are also applied to EEPROM reprogramming.

4.4.2 Operations of EEPROM

Reprogramming EEPROM data

Follow the procedure below to reprogram the EEPROM.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Set the following EPRCCTL0 register bits:
   - Set the EPRCCTL0.EP_WMODE bit to 1. (Start reprogramming mode)
   - Set the EPRCCTL0.EP_PWRSET bit to 1. (Turn programming power supply on)

3. Wait for the programming power supply to stabilize (for the wait time, refer to "EEPROM Characteristics" in the "Electrical Characteristics" chapter.).
4. Write 1 to the EPRCINTF.RXBIF bit. (Clear interrupt flag)
5. Set the EPRCINTE.RXBIF bit to 1. (Enable interrupt)
6. Set the EPRCADR.EP_ADDR[7:0] bits. (Set reprogramming address)
7. Set the EPRCWDAT.EP_WDAT[7:0] bits. (Set programming data)
   The programming data should be stored in the RAM for the verification to be performed later.
8. Write 1 to the EPRCCTL1.EP_CK bit. (Output clock pulse)
9. Wait for an interrupt.
   When the reprogramming has completed, the EPRCINTF.RXBIF bit is set to 1.
10. Repeat Steps 4 to 9 for the addresses to be programmed.
11. Set the following EPRCCTL0 register bits:
    - Set the EPRCCTL0.EP_WMODE bit to 0. (Stop reprogramming mode)
    - Set the EPRCCTL0.EP_PWRSET bit to 0. (Turn programming power supply off)
12. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)
13. Read EEPROM data and verify it with the programming data.

Reading EEPROM data

EEPROM data can be read from the memory area (logical address) where the EEPROM has been assigned using a standard 8-bit or 16-bit memory read instruction. If a 32-bit memory read instruction is used, only 16 bits are read from the EEPROM and the high-order bits are all set to 0. Note that EEPROM data is indefinite if it is read while the EPRCCTL0.EP_WMODE bit = 1 (reprogramming mode). When a ECC interrupt has occurred during reading data, the EPRCINTF.ECCERIF bit is set to 1.

Note: If an ECC interrupt has occurred, the reprogramming count may reach its limit. In this case, copy the data to another address and the address that generates an ECC interrupt should not be used in the subsequent reprogramming.

4.4.3 Interrupts

EEPROMC has a function to generate the interrupts shown in Table 4.4.3.1.

Table 4.4.3.1 EEPROMC Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Reprogram/read completion	EPRCINTF.RXBIF	When reprogramming/reading of the EEPROM has completed	Writing 1
ECC	EPRCINTF.ECCERIF	When data has been corrected via ECC during data reading	Writing 1

The EEPROMC provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the "Interrupt Controller" chapter.

4.5 RAM

The RAM can be used to execute the instruction codes copied from another memory as well as storing variables or other data. This allows higher speed processing and lower power consumption than Flash memory.

Note: The 64 bytes at the end of the RAM is reserved as the debug RAM area. When using the debug functions under application development, do not access this area from the application program. This area can be used for applications of mass-produced devices that do not need debugging.

The RAM size used by the application can be configured to equal or less than the implemented size using the MSCIRAMSZ.IRAMSZ[2:0] bits. For example, this function can be used to prevent creating programs that seek to access areas outside the RAM area of the target model when developing an application for a model in which the RAM size is smaller than this IC. After the limitation is applied, accessing an address outside the RAM area results in the same operation (undefined value is read out) as when a reserved area is accessed.

4.6 Display Data RAM

The embedded display data RAM is used to store display data for the LCD driver. Areas unused for display data in the display data RAM can be used as a general-purpose RAM. For specific information on the display data RAM, refer to "Display Data RAM" in the "LCD Driver" chapter.

4.7 Peripheral Circuit Control Registers

The control registers for the peripheral circuits are located in the 8K-byte area beginning with address 0x4000. Table 4.7.1 shows the control register map. For details of each control register, refer to "List of Peripheral Circuit Registers" in the appendix or "Control Registers" in each peripheral circuit chapter.

Table 4.7.1 Peripheral Circuit Control Register Map

Peripheral circuit	Address	Register name	
MISC registers (MISC)	0x4000	MSCPROT	MISC System Protect Register
	0x4002	MSCIRAMSZ	MISC IRAM Size Register
	0x4004	MSCTTBRL	MISC Vector Table Address Low Register
	0x4006	MSCTTBRH	MISC Vector Table Address High Register
	0x4008	MSCPSR	MISC PSR Register
Power generator (PWG)	0x4020	PWGVD1CTL	PWG V_{D1} Regulator Control Register
Clock generator (CLG)	0x4040	CLGSCLK	CLG System Clock Control Register
	0x4042	CLGOSC	CLG Oscillation Control Register
	0x4046	CLGOSC1	CLG OSC1 Control Register
	0x4048	CLGOSC3	CLG OSC3 Control Register
	0x404c	CLGINTF	CLG Interrupt Flag Register
	0x404e	CLGINTE	CLG Interrupt Enable Register
	0x4050	CLGFOUT	CLG FOUT Control Register
	0x4052	CLGTRIM1	CLG Oscillation Frequency Trimming Register 1
	0x4054	CLGTRIM2	CLG Oscillation Frequency Trimming Register 2
Interrupt controller (ITC)	0x4080	ITCLV0	ITC Interrupt Level Setup Register 0
	0x4082	ITCLV1	ITC Interrupt Level Setup Register 1
	0x4084	ITCLV2	ITC Interrupt Level Setup Register 2
	0x4086	ITCLV3	ITC Interrupt Level Setup Register 3
	0x4088	ITCLV4	ITC Interrupt Level Setup Register 4
	0x408a	ITCLV5	ITC Interrupt Level Setup Register 5
	0x408c	ITCLV6	ITC Interrupt Level Setup Register 6
	0x408e	ITCLV7	ITC Interrupt Level Setup Register 7
	0x4090	ITCLV8	ITC Interrupt Level Setup Register 8
	0x4092	ITCLV9	ITC Interrupt Level Setup Register 9
	0x4094	ITCLV10	ITC Interrupt Level Setup Register 10
	0x4096	ITCLV11	ITC Interrupt Level Setup Register 11
Watchdog timer (WDT2)	0x40a0	WDTCLK	WDT2 Clock Control Register
	0x40a2	WDTCTL	WDT2 Control Register
	0x40a4	WDTCMP	WDT2 Counter Compare Match Register
Supply voltage detector (SVD4)	0x4100	SVDCLK	SVD4 Clock Control Register
	0x4102	SVDCTL	SVD4 Control Register
	0x4104	SVDINTF	SVD4 Status and Interrupt Flag Register
	0x4106	SVDINTE	SVD4 Interrupt Enable Register
16-bit timer (T16) Ch.0	0x4160	T16_0CLK	T16 Ch.0 Clock Control Register
	0x4162	T16_0MOD	T16 Ch.0 Mode Register
	0x4164	T16_0CTL	T16 Ch.0 Control Register
	0x4166	T16_0TR	T16 Ch.0 Reload Data Register
	0x4168	T16_0TC	T16 Ch.0 Counter Data Register
	0x416a	T16_0INTF	T16 Ch.0 Interrupt Flag Register
	0x416c	T16_0INTE	T16 Ch.0 Interrupt Enable Register

Peripheral circuit	Address	Register name		
Flash controller (FLASHC)	0x41b0	FLASHCWAIT	FLASHC Flash Read Cycle Register	
EEPROM controller (EEPROMC)	0x41c0	EPRCCTL0	EEPROMC Control Register 0	
	0x41c2	EPRCCTL1	EEPROMC Control Register 1	
	0x41c4	EPRCADR	EEPROMC Address Register	
	0x41c6	EPRCWDAT	EEPROMC Write Data Register	
	0x41c8	EPRCINTF	EEPROMC Interrupt Flag Register	
	0x41ca	EPRCINTE	EEPROMC Interrupt Enable Register	
I/O ports (PPORT)	0x4200	P0DAT	P0 Port Data Register	
	0x4202	P0IOEN	P0 Port Enable Register	
	0x4204	P0RCTL	P0 Port Pull-up/down Control Register	
	0x4206	P0INTF	P0 Port Interrupt Flag Register	
	0x4208	P0INTCTL	P0 Port Interrupt Control Register	
	0x420a	P0CHATEN	P0 Port Chattering Filter Enable Register	
	0x420c	P0MODSEL	P0 Port Mode Select Register	
	0x420e	P0FNCSEL	P0 Port Function Select Register	
	0x4210	P1DAT	P1 Port Data Register	
	0x4212	P1IOEN	P1 Port Enable Register	
	0x4214	P1RCTL	P1 Port Pull-up/down Control Register	
	0x4216	P1INTF	P1 Port Interrupt Flag Register	
	0x4218	P1INTCTL	P1 Port Interrupt Control Register	
	0x421a	P1CHATEN	P1 Port Chattering Filter Enable Register	
	0x421c	P1MODSEL	P1 Port Mode Select Register	
	0x421e	P1FNCSEL	P1 Port Function Select Register	
	0x4220	P2DAT	P2 Port Data Register	*
	0x4222	P2IOEN	P2 Port Enable Register	*
	0x4224	P2RCTL	P2 Port Pull-up/down Control Register	*
	0x4226	P2INTF	P2 Port Interrupt Flag Register	*
	0x4228	P2INTCTL	P2 Port Interrupt Control Register	*
	0x422a	P2CHATEN	P2 Port Chattering Filter Enable Register	*
	0x422c	P2MODSEL	P2 Port Mode Select Register	*
	0x422e	P2FNCSEL	P2 Port Function Select Register	*
	0x4230	P3DAT	P3 Port Data Register	*
	0x4232	P3IOEN	P3 Port Enable Register	*
	0x4234	P3RCTL	P3 Port Pull-up/down Control Register	*
	0x4236	P3INTF	P3 Port Interrupt Flag Register	*
	0x4238	P3INTCTL	P3 Port Interrupt Control Register	*
	0x423a	P3CHATEN	P3 Port Chattering Filter Enable Register	*
	0x423c	P3MODSEL	P3 Port Mode Select Register	*
	0x423e	P3FNCSEL	P3 Port Function Select Register	*
	0x4240	P4DAT	P4 Port Data Register	*
	0x4242	P4IOEN	P4 Port Enable Register	*
	0x4244	P4RCTL	P4 Port Pull-up/down Control Register	*
	0x4246	P4INTF	P4 Port Interrupt Flag Register	*
	0x4248	P4INTCTL	P4 Port Interrupt Control Register	*
	0x424a	P4CHATEN	P4 Port Chattering Filter Enable Register	*
	0x424c	P4MODSEL	P4 Port Mode Select Register	*
	0x424e	P4FNCSEL	P4 Port Function Select Register	*
	0x42d2	PDIOEN	Pd Port Enable Register	
	0x42d4	PDRCTL	Pd Port Pull-up/down Control Register	
	0x42dc	PDMODSEL	Pd Port Mode Select Register	
	0x42de	PDFNCSEL	Pd Port Function Select Register	
	0x42e0	PCLK	P Port Clock Control Register	
	0x42e2	PINTFGRP	P Port Interrupt Flag Group Register	
Universal port multiplexer (UPMUX)	0x4300	P0UPMUX0	P00–01 Universal Port Multiplexer Setting Register	
	0x4302	P0UPMUX1	P02–03 Universal Port Multiplexer Setting Register	
	0x4304	P0UPMUX2	P04–05 Universal Port Multiplexer Setting Register	
	0x4306	P0UPMUX3	P06–07 Universal Port Multiplexer Setting Register	
	0x4308	P1UPMUX0	P10–11 Universal Port Multiplexer Setting Register	
	0x430a	P1UPMUX1	P12–13 Universal Port Multiplexer Setting Register	
	0x430c	P1UPMUX2	P14–15 Universal Port Multiplexer Setting Register	
	0x430e	P1UPMUX3	P16–17 Universal Port Multiplexer Setting Register	
	0x4310	P2UPMUX0	P20–21 Universal Port Multiplexer Setting Register	*

Peripheral circuit	Address	Register name	
Universal port multiplexer (UPMUX)	0x4312	P2UPMUX1	P22–23 Universal Port Multiplexer Setting Register *
	0x4314	P2UPMUX2	P24–25 Universal Port Multiplexer Setting Register *
	0x4316	P2UPMUX3	P26–27 Universal Port Multiplexer Setting Register *
	0x4318	P3UPMUX0	P30–31 Universal Port Multiplexer Setting Register *
	0x431a	P3UPMUX1	P32–33 Universal Port Multiplexer Setting Register *
	0x431c	P3UPMUX2	P34–35 Universal Port Multiplexer Setting Register *
	0x431e	P3UPMUX3	P36–37 Universal Port Multiplexer Setting Register *
UART (UART3) Ch.0	0x4380	UA0CLK	UART3 Ch.0 Clock Control Register
	0x4382	UA0MOD	UART3 Ch.0 Mode Register
	0x4384	UA0BR	UART3 Ch.0 Baud-Rate Register
	0x4386	UA0CTL	UART3 Ch.0 Control Register
	0x4388	UA0TXD	UART3 Ch.0 Transmit Data Register
	0x438a	UA0RXD	UART3 Ch.0 Receive Data Register
	0x438c	UA0INTF	UART3 Ch.0 Status and Interrupt Flag Register
	0x438e	UA0INTE	UART3 Ch.0 Interrupt Enable Register
	0x4390	UA0CAWF	UART3 Ch.0 Carrier Waveform Register
16-bit timer (T16) Ch.1	0x43a0	T16_1CLK	T16 Ch.1 Clock Control Register
	0x43a2	T16_1MOD	T16 Ch.1 Mode Register
	0x43a4	T16_1CTL	T16 Ch.1 Control Register
	0x43a6	T16_1TR	T16 Ch.1 Reload Data Register
	0x43a8	T16_1TC	T16 Ch.1 Counter Data Register
	0x43aa	T16_1INTF	T16 Ch.1 Interrupt Flag Register
	0x43ac	T16_1INTE	T16 Ch.1 Interrupt Enable Register
Synchronous serial interface (SPIA) Ch.0	0x43b0	SPI0MOD	SPIA Ch.0 Mode Register
	0x43b2	SPI0CTL	SPIA Ch.0 Control Register
	0x43b4	SPI0TXD	SPIA Ch.0 Transmit Data Register
	0x43b6	SPI0RXD	SPIA Ch.0 Receive Data Register
	0x43b8	SPI0INTF	SPIA Ch.0 Interrupt Flag Register
	0x43ba	SPI0INTE	SPIA Ch.0 Interrupt Enable Register
I²C (I2C) Ch.0	0x43c0	I2C0CLK	I2C Ch.0 Clock Control Register
	0x43c2	I2C0MOD	I2C Ch.0 Mode Register
	0x43c4	I2C0BR	I2C Ch.0 Baud-Rate Register
	0x43c8	I2C0OADR	I2C Ch.0 Own Address Register
	0x43ca	I2C0CTL	I2C Ch.0 Control Register
	0x43cc	I2C0TXD	I2C Ch.0 Transmit Data Register
	0x43ce	I2C0RXD	I2C Ch.0 Receive Data Register
	0x43d0	I2C0INTF	I2C Ch.0 Status and Interrupt Flag Register
	0x43d2	I2C0INTE	I2C Ch.0 Interrupt Enable Register
DMM 16-bit PWM timer (T16B_DMM) Ch.0	0x5000	T16B0CLK	T16B_DMM Ch.0 Clock Control Register
	0x5002	T16B0CTL	T16B_DMM Ch.0 Counter Control Register
	0x5004	T16B0MC	T16B_DMM Ch.0 Max Counter Data Register
	0x5006	T16B0TC	T16B_DMM Ch.0 Timer Counter Data Register
	0x5008	T16B0CS	T16B_DMM Ch.0 Counter Status Register
	0x500a	T16B0INTF	T16B_DMM Ch.0 Interrupt Flag Register
	0x500c	T16B0INTE	T16B_DMM Ch.0 Interrupt Enable Register
	0x5010	T16B0CCCTL0	T16B_DMM Ch.0 Compare/Capture 0 Control Register
	0x5012	T16B0CCR0	T16B_DMM Ch.0 Compare/Capture 0 Data Register
	0x5018	T16B0CCCTL1	T16B_DMM Ch.0 Compare/Capture 1 Control Register
	0x501a	T16B0CCR1	T16B_DMM Ch.0 Compare/Capture 1 Data Register
DMM 16-bit PWM timer (T16B_DMM) Ch.1	0x5040	T16B1CLK	T16B_DMM Ch.1 Clock Control Register
	0x5042	T16B1CTL	T16B_DMM Ch.1 Counter Control Register
	0x5044	T16B1MC	T16B_DMM Ch.1 Max Counter Data Register
	0x5046	T16B1TC	T16B_DMM Ch.1 Timer Counter Data Register
	0x5048	T16B1CS	T16B_DMM Ch.1 Counter Status Register
	0x504a	T16B1INTF	T16B_DMM Ch.1 Interrupt Flag Register
	0x504c	T16B1INTE	T16B_DMM Ch.1 Interrupt Enable Register
	0x5050	T16B1CCCTL0	T16B_DMM Ch.1 Compare/Capture 0 Control Register
	0x5052	T16B1CCR0	T16B_DMM Ch.1 Compare/Capture 0 Data Register
	0x5058	T16B1CCCTL1	T16B_DMM Ch.1 Compare/Capture 1 Control Register
	0x505a	T16B1CCR1	T16B_DMM Ch.1 Compare/Capture 1 Data Register

Peripheral circuit	Address	Register name	
DMM 16-bit PWM timer (T16B_DMM) Ch.2	0x5080	T16B2CLK	T16B_DMM Ch.2 Clock Control Register
	0x5082	T16B2CTL	T16B_DMM Ch.2 Counter Control Register
	0x5084	T16B2MC	T16B_DMM Ch.2 Max Counter Data Register
	0x5086	T16B2TC	T16B_DMM Ch.2 Timer Counter Data Register
	0x5088	T16B2CS	T16B_DMM Ch.2 Counter Status Register
	0x508a	T16B2INTF	T16B_DMM Ch.2 Interrupt Flag Register
	0x508c	T16B2INTE	T16B_DMM Ch.2 Interrupt Enable Register
	0x5090	T16B2CCCTL0	T16B_DMM Ch.2 Compare/Capture 0 Control Register
	0x5092	T16B2CCR0	T16B_DMM Ch.2 Compare/Capture 0 Data Register
	0x5098	T16B2CCCTL1	T16B_DMM Ch.2 Compare/Capture 1 Control Register
	0x509a	T16B2CCR1	T16B_DMM Ch.2 Compare/Capture 1 Data Register
16-bit timer (T16) Ch.2	0x5260	T16_2CLK	T16 Ch.2 Clock Control Register
	0x5262	T16_2MOD	T16 Ch.2 Mode Register
	0x5264	T16_2CTL	T16 Ch.2 Control Register
	0x5266	T16_2TR	T16 Ch.2 Reload Data Register
	0x5268	T16_2TC	T16 Ch.2 Counter Data Register
	0x526a	T16_2INTF	T16 Ch.2 Interrupt Flag Register
	0x526c	T16_2INTE	T16 Ch.2 Interrupt Enable Register
Sound generator (SNDA_DMM)	0x5300	SNDCLK	SNDA_DMM Clock Control Register
	0x5302	SNDSEL	SNDA_DMM Select Register
	0x5304	SNDCTL	SNDA_DMM Control Register
	0x5306	SNDDAT	SNDA_DMM Data Register
	0x5308	SNDINTF	SNDA_DMM Interrupt Flag Register
	0x530a	SNDINTE	SNDA_DMM Interrupt Enable Register
LCD driver (LCD4B)	0x5400	LCD4CLK	LCD4B Clock Control Register
	0x5402	LCD4CTL	LCD4B Control Register
	0x5404	LCD4TIM1	LCD4B Timing Control Register 1
	0x5406	LCD4TIM2	LCD4B Timing Control Register 2
	0x5408	LCD4PWR	LCD4B Power Control Register
	0x540a	LCD4DSP	LCD4B Display Control Register
	0x540c	LCD4COMC0	LCD4B COM Pin Control Register 0
	0x5410	LCD4INTF	LCD4B Interrupt Flag Register
	0x5412	LCD4INTE	LCD4B Interrupt Enable Register
DMM controller (DSADC16)	0x5500	DSADC16CLK	DSADC16 Clock Control Register
	0x5502	DSADC16CONF	DSADC16 Configuration Register
	0x5504	DSADC16CTL	DSADC16 Control Register
	0x5506	DSADC16INIT	DSADC16 Initialize Control Register
	0x5508	DSADC16IE	DSADC16 Interrupt Enable Register
	0x550a	DSADC16IF	DSADC16 Interrupt Flag Register
	0x550c	DSADC16COMB	DSADC16 Comb Filter Result Register
	0x550e	DSADC16LPFHPF	DSADC16 Low Pass/High Pass Filter Result Register
	0x5510	DSADC16RMS1	DSADC16 RMS Result Register 1
	0x5512	DSADC16RMS2	DSADC16 RMS Result Register 2
	0x5514	DSADC16DC-PEAKMAX	DSADC16 DC Peak Hold MAX Result Register
	0x5516	DSADC16DC-PEAKMIN	DSADC16 DC Peak Hold MIN Result Register
	0x5518	DSADC16AC-PEAKMAX1	DSADC16 AC Peak Hold MAX Result Register 1
	0x551a	DSADC16AC-PEAKMAX2	DSADC16 AC Peak Hold MAX Result Register 2
	0x551c	DSADC16AC-PEAKMIN1	DSADC16 AC Peak Hold MIN Result Register 1
	0x551e	DSADC16AC-PEAKMIN2	DSADC16 AC Peak Hold MIN Result Register 2
	0x5520	VIRCTL	VIR Control Register
	0x5522	DMMSET1	DMM Setting Register 1
	0x5524	DMMSET2	DMM Setting Register 2
	0x5526	SMODESET	DMM SMODE Setting Register
	0x5528	AFENET1	AFE Network Setting Register 1
	0x552a	AFENET2	AFE Network Setting Register 2
	0x552c	AFENET3	AFE Network Setting Register 3

Peripheral circuit	Address	Register name	
DMM controller (DSADC16)	0x552e	CHPCTL	Chopper Amp Control Register
	0x5530	DSMVCMCTL	Sigma Delta Modulator & ADCVCM Control Register
	0x5532	TSRVRTEMP	TSRVR Temperature Correction Data Register
	0x5534	TSRVR	TSRVR Control Register
	0x5536	CMPOUT	Comparator Output Status Register
	0x553c	DCDCCTL	DCDC Control Register
	0x553e	AFESUB	AFE Sub-control Register
16-bit timer (T16) Ch.3	0x5580	T16_3CLK	T16 Ch.3 Clock Control Register
	0x5582	T16_3MOD	T16 Ch.3 Mode Register
	0x5584	T16_3CTL	T16 Ch.3 Control Register
	0x5586	T16_3TR	T16 Ch.3 Reload Data Register
	0x5588	T16_3TC	T16 Ch.3 Counter Data Register
	0x558a	T16_3INTF	T16 Ch.3 Interrupt Flag Register
	0x558c	T16_3INTE	T16 Ch.3 Interrupt Enable Register

* Available only in the S1C17M03.

4.7.1 System-Protect Function

The system-protect function protects control registers and bits from writings. They cannot be rewritten unless write protection is removed by writing 0x0096 to the MSCPROT.PROT[15:0] bits. This function is provided to prevent deadlock that may occur when a system-related register is altered by a runaway CPU. See "Control Registers" in each peripheral circuit to identify the registers and bits with write protection.

Note: Once write protection is removed using the MSCPROT.PROT[15:0] bits, write enabled status is maintained until write protection is applied again. After the registers/bits required have been altered, apply write protection.

4.8 Control Registers

MISC System Protect Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
MSCPROT	15–0	PROT[15:0]	0x0000	H0	R/W	–

Bits 15–0 PROT[15:0]

These bits protect the control registers related to the system against writings.
0x0096 (R/W): Disable system protection
Other than 0x0096 (R/W): Enable system protection

While the system protection is enabled, any data will not be written to the affected control bits (bits with "WP" or "R/WP" appearing in the R/W column).

MISC IRAM Size Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
MSCIRAMSZ	15–9	–	0x00	–	R	–
	8	(reserved)	0	H0	R/WP	Always set to 0.
	7–3	–	0x04	–	R	–
	2–0	IRAMSZ[2:0]	0x2	H0	R/WP	

Bits 15–3 Reserved

Bits 2–0 IRAMSZ[2:0]

These bits set the internal RAM size that can be used.

Table 4.8.1 Internal RAM Size Selections

MSCIRAMSZ.IRAMSZ[2:0] bits	Internal RAM size
0x7–0x3	Reserved
0x2	2KB
0x1	1KB
0x0	512B

FLASHC Flash Read Cycle Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
FLASHCWAIT	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	–
	1–0	RDWAIT[1:0]	0x1	H0	R/WP	

Bits 15–2 Reserved

Bits 1–0 RDWAIT[1:0]

These bits set the number of bus access cycles for reading from the Flash memory.

Table 4.8.2 Setting Number of Bus Access Cycles for Flash Read

FLASHCWAIT.RDWAIT[1:0] bits	Number of bus Access cycles	System clock frequency
0x3	4	6.7 MHz (max.)
0x2	3	
0x1	2	
0x0	1	3.4 MHz (max.)

Note: Be sure to set the FLASHCWAIT.RDWAIT[1:0] bits before the system clock is configured.

EEPROMC Control Register 0

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
EPRCCTL0	15–9	–	0x00	–	R	–
	8	EP_XPOR	1	H0	R/WP	
	7–2	–	0x00	–	R	
	1	EP_PWRSET	0	H0	R/WP	
	0	EP_WMODE	0	H0	R/WP	

Bits 15–9 Reserved

Bit 8 EP_XPOR

This bit controls the reset signal of the EEPROM.
1 (R/W): Negate the reset signal.
0 (R/W): Assert the reset signal.

When a reprogram/read completion interrupt has not occurred (EPRCINTF.RXBIF bit has not been set to 1) after starting an EEPROM reprogramming operation, write 0 to this bit to reset the EEPROM. While this bit is 0, the EEPROM control functions including reading of the EEPROM are all disabled. To resume the EEPROM operations again, write 1 to this bit to cancel the reset state after waiting for a longer time than the effective EEPROM reset pulse width t_{XPOR} (refer to "EEPROM Characteristics" in the "Electrical Characteristics" chapter).

Bits 7–2 Reserved

Bit 1 EP_PWRSET

This bit controls the programming power supply.
1 (R/W): Programming power supply ON
0 (R/W): Programming power supply OFF

When this bit is set to 1, the EEPROM programming power supply circuit goes on and it generates the EEPROM programming voltage by boosting the V_{PP} voltage. This bit is effective when the EPRCCTL0.EP_WMODE bit = 1.

Bit 0 EP_WMODE

This bit starts/stops reprogramming mode.
1 (R/W): Start reprogramming mode
0 (R/W): Stop reprogramming mode

Setting this bit to 1 puts the EEPROM into reprogramming mode to enable data reprogramming. Note that read data are indefinite when the EEPROM is read while this bit is 1.

EEPROMC Control Register 1

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
EPRCCTL1	15–8	–	0x00	–	R	–
	7–1	–	0x00	–	R	
	0	EP_CK	0	H0	WP	

Bits 15–1 Reserved

Bit 0 EP_CK

This bit controls the clock pulse output to reprogram the EEPROM.

1 (W): Output one clock pulse

0 (W): Ineffective

Writing 1 to this bit outputs a clock to the EEPROM to reprogram the EEPROM address specified by the EPRCADR.EP_ADDR[7:0] bits with the data specified by the EPRCWDAT.EP_WDAT[7:0] bits. This bit automatically reverts to 0 after writing 1.

This bit is effective when the EPRCCTL0.EP_WMODE bit = 1.

EEPROMC Address Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
EPRCADR	15–8	–	0x00	–	R	–
	7–0	EP_ADDR[7:0]	0x00	H0	R/WP	

Bits 15–8 Reserved

Bits 7–0 EP_ADDR[7:0]

These bits specify the EEPROM physical address (0 to 255) to be reprogrammed.

EEPROM (logical) address = 0x6000 + EPRCADR.EP_ADDR[7:0] bits

EEPROMC Write Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
EPRCWDAT	15–8	–	0x00	–	R	–
	7–0	EP_WDAT[7:0]	0x00	H0	R/WP	

Bits 15–8 Reserved

Bits 7–0 EP_WDAT[7:0]

These bits specify the 8-bit data to program the EEPROM.

EEPROMC Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
EPRCINTF	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1	ECCERIF	0	H0	R/WP	Cleared by writing 1.
	0	RXBIF	0	H0	R/WP	

Bits 15–2 Reserved

Bit 1 ECCERIF

Bit 0 RXBIF

These bits indicate the EEPROMC interrupt cause occurrence status.

1 (R): Cause of interrupt occurred

0 (R): No cause of interrupt occurred

1 (W): Clear flag

0 (W): Ineffective

The following shows the correspondence between the bit and interrupt:

EPRCINTF.ECCERIF bit: ECC interrupt

EPRCINTF.RXBIF bit: Reprogram/read completion interrupt

EEPROMC Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
EPRCINTE	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1	ECCERIE	0	H0	R/WP	
	0	RXBIE	0	H0	R/WP	

Bits 15–2 Reserved

Bit 1 ECCERIE
Bit 0 RXBIE

These bits enable EEPROMC interrupts.
1 (R/W): Enable interrupts
0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:
EPRCINTE.ECCERIE bit: ECC interrupt
EPRCINTE.RXBIE bit: Reprogram/read completion interrupt

Note: To prevent generating unnecessary interrupts, the corresponding interrupt flag should be cleared before enabling interrupts.

5 Interrupt Controller (ITC)

5.1 Overview

The features of the ITC are listed below.

- Honors interrupt requests from the peripheral circuits and outputs the interrupt request, interrupt level and vector number signals to the CPU.
- The interrupt level of each interrupt source is selectable from among eight levels.
- Priorities of the simultaneously generated interrupts are established from the interrupt level.
- Handles the simultaneously generated interrupts with the same interrupt level as smaller vector number has higher priority.

Figure 5.1.1 shows the configuration of the ITC.

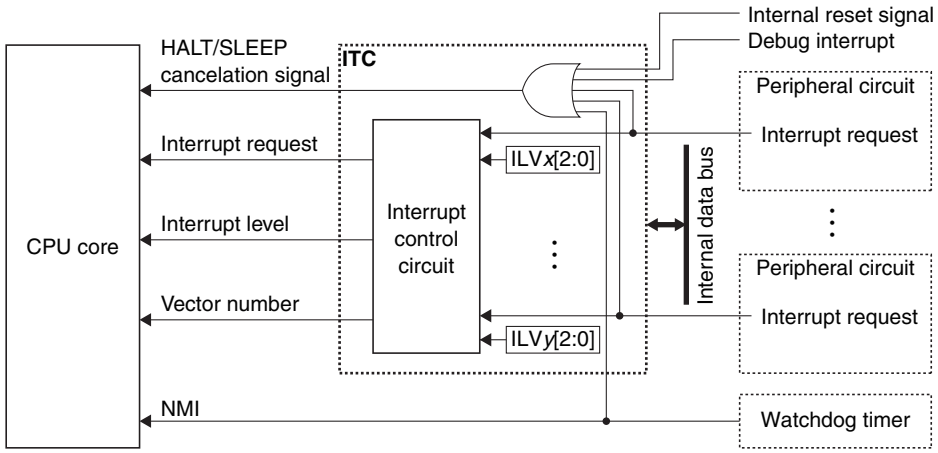


Figure 5.1.1 ITC Configuration

5.2 Vector Table

The vector table contains the vectors to the interrupt handler routines (handler routine start address) that will be read by the CPU to execute the handler when an interrupt occurs.
Table 5.2.1 shows the vector table.

Table 5.2.1 Vector Table

TTBR initial value = 0x8000

Vector number/ Software interrupt number	Vector address	Hardware interrupt name	Cause of hardware interrupt	Priority
0 (0x00)	TTBR + 0x00	Reset	• Low input to the #RESET pin • Power-on reset • Key reset • Watchdog timer overflow *2 • Supply voltage detector reset	1
1 (0x01)	TTBR + 0x04	Address misaligned interrupt	Memory access instruction	2
–	(0xfffc00)	Debugging interrupt	brk instruction, etc.	3
2 (0x02)	TTBR + 0x08	NMI	Watchdog timer overflow *2	4
3 (0x03)	TTBR + 0x0c	Reserved for C compiler	–	–

Vector number/ Software interrupt number	Vector address	Hardware interrupt name	Hardware interrupt flag	Priority
4 (0x04)	TTBR + 0x10	Supply voltage detector interrupt	Low power supply voltage detection	High *1 ↑
5 (0x05)	TTBR + 0x14	Port interrupt	Port input	
6 (0x06)	TTBR + 0x18	reserved	–	
7 (0x07)	TTBR + 0x1c	Clock generator interrupt	• IOSC oscillation stabilization waiting completion • OSC1 oscillation stabilization waiting completion • OSC3 oscillation stabilization waiting completion • OSC1 oscillation stop	
8 (0x08)	TTBR + 0x20	reserved	–	
9 (0x09)	TTBR + 0x24	16-bit timer Ch.0 interrupt	Underflow	
10 (0x0a)	TTBR + 0x28	UART Ch.0 interrupt	• End of transmission • Framing error • Parity error • Overrun error • Receive buffer two bytes full • Receive buffer one byte full • Transmit buffer empty	
11 (0x0b)	TTBR + 0x2c	16-bit timer Ch.1 interrupt	Underflow	
12 (0x0c)	TTBR + 0x30	Synchronous serial interface Ch.0 interrupt	• End of transmission • Receive buffer full • Transmit buffer empty • Overrun error	
13 (0x0d)	TTBR + 0x34	I^2C interrupt	• End of data transfer • General call address reception • NACK reception • STOP condition • START condition • Error detection • Receive buffer full • Transmit buffer empty	
14 (0x0e)	TTBR + 0x38	DMM 16-bit PWM timer Ch.0 interrupt	• Capture overwrite • Compare/capture • Counter MAX • Counter zero	
15 (0x0f)	TTBR + 0x3c	DMM 16-bit PWM timer Ch.1 interrupt	• Capture overwrite • Compare/capture • Counter MAX • Counter zero	
16 (0x10)	TTBR + 0x40	reserved	–	
17 (0x11)	TTBR + 0x44	Sound generator interrupt	• Sound buffer empty • Sound output completion	
18 (0x12)	TTBR + 0x48	reserved	–	
19 (0x13)	TTBR + 0x4c	LCD driver interrupt	Frame	
20 (0x14)	TTBR + 0x50	reserved	–	
21 (0x15)	TTBR + 0x54	EEPROM controller interrupt	• Reprogram/read completion • ECC	
22 (0x16)	TTBR + 0x58	16-bit timer Ch.2 interrupt	Underflow	
23 (0x17)	TTBR + 0x5c	reserved	–	
24 (0x18)	TTBR + 0x60	16-bit timer Ch.3 interrupt	Underflow	
25 (0x19)	TTBR + 0x64	DMM controller interrupt	• Conversion completion • Conversion result overwrite error • Continuity status change detection	
26 (0x1a)	TTBR + 0x68	DMM 16-bit PWM timer Ch.2 interrupt	• Capture overwrite • Compare/capture • Counter MAX • Counter zero	
27 (0x1b) : 31 (0x1f)	TTBR + 0x6c : TTBR + 0x7c	reserved : reserved	– : –	↓ Low *1

*1 When the same interrupt level is set

*2 Either reset or NMI can be selected as the watchdog timer interrupt with software.

5.2.1 Vector Table Base Address (TTBR)

The MSCTTBRL and MSCTTBRH registers are provided to set the base (start) address of the vector table in which interrupt vectors are programmed. "TTBR" described in Table 5.2.1 means the value set to these registers. After an initial reset, the MSCTTBRL and MSCTTBRH registers are set to address 0x8000. Therefore, even when the vector table location is changed, it is necessary that at least the reset vector be written to the above address. Bits 7 to 0 in the MSCTTBRL register are fixed at 0, so the vector table always begins from a 256-byte boundary address.

5.3 Initialization

The following shows an example of the initial setting procedure related to interrupts:

1. Execute the di instruction to set the CPU into interrupt disabled state.
2. If the vector table start address is different from the default address, set it to the MSCTTBRL and MSCTTBRH registers after removing system protection by writing 0x0096 to the MSCPROT.PROT[15:0] bits. Then, write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits to set system protection.
3. Set the interrupt enable bit of the peripheral circuit to 0 (interrupt disabled).
4. Set the interrupt level for the peripheral circuit using the ITCLV*x*.ILV*x*[2:0] bits in the ITC.
5. Configure the peripheral circuit and start its operation.
6. Clear the interrupt factor flag of the peripheral circuit.
7. Set the interrupt enable bit of the peripheral circuit to 1 (interrupt enabled).
8. Execute the ei instruction to set the CPU into interrupt enabled state.

5.4 Maskable Interrupt Control and Operations

5.4.1 Peripheral Circuit Interrupt Control

The peripheral circuit that generates interrupts includes an interrupt enable bit and an interrupt flag for each interrupt cause.

Interrupt flag: The flag is set to 1 when the interrupt cause occurs. The clear condition depends on the peripheral circuit.

Interrupt enable bit: By setting this bit to 1 (interrupt enabled), an interrupt request will be sent to the ITC when the interrupt flag is set to 1. When this bit is set to 0 (interrupt disabled), no interrupt request will be sent to the ITC even if the interrupt flag is set to 1. An interrupt request is also sent to the ITC if the status is changed to interrupt enabled when the interrupt flag is 1.

For specific information on causes of interrupts, interrupt flags, and interrupt enable bits, refer to the respective peripheral circuit descriptions.

Note: To prevent occurrence of unnecessary interrupts, the corresponding interrupt flag should be cleared before setting the interrupt enable bit to 1 (interrupt enabled) and before terminating the interrupt handler routine.

5.4.2 ITC Interrupt Request Processing

On receiving an interrupt signal from a peripheral circuit, the ITC sends an interrupt request, the interrupt level, and the vector number to the CPU. Vector numbers are determined by the ITC internal hardware for each interrupt cause, as shown in Table 5.2.1. The interrupt level is a value to configure the priority, and it can be set to between 0 (low) and 7 (high) using the ITCLV*x*.ILV*x*[2:0] bits provided for each interrupt source. The default ITC settings are level 0 for all maskable interrupts. Interrupt requests are not accepted by the CPU if the level is 0.

The ITC outputs the interrupt request with the highest priority to the CPU in accordance with the following conditions if interrupt requests are input to the ITC simultaneously from two or more peripheral circuits.

- The interrupt with the highest interrupt level takes precedence.
- If multiple interrupt requests are input with the same interrupt level, the interrupt with the lowest vector number takes precedence.

The other interrupts occurring at the same time are held until all interrupts with higher priority levels have been accepted by the CPU.
If an interrupt cause with higher priority occurs while the ITC is outputting an interrupt request signal to the CPU (before being accepted by the CPU), the ITC alters the vector number and interrupt level signals to the setting information on the more recent interrupt. The previously occurring interrupt is held. The held interrupt is canceled and no interrupt is generated if the interrupt flag in the peripheral circuit is cleared via software.

Note: Before changing the interrupt level, make sure that no interrupt of which the level is changed can be generated (the interrupt enable bit of the peripheral circuit is set to 0 or the peripheral circuit is deactivated).

5.4.3 Conditions to Accept Interrupt Requests by the CPU

The CPU accepts an interrupt request sent from the ITC when all of the following conditions are met:

- The IE (Interrupt Enable) bit of the PSR has been set to 1.
- The interrupt request that has occurred has a higher interrupt level than the value set in the IL[2:0] (Interrupt Level) bits of the PSR.
- No other interrupt request having higher priority, such as NMI, has occurred.

5.5 NMI

The watchdog timer embedded in this IC can generate a non-maskable interrupt (NMI). This interrupt takes precedence over other interrupts and is unconditionally accepted by the CPU.
For detailed information on generating NMI, refer to the "Watchdog Timer" chapter.

5.6 Software Interrupts

The CPU provides the "int *imm5*" and "intl *imm5*, *imm3*" instructions allowing the software to generate any interrupts. The operand *imm5* specifies a vector number (0–31) in the vector table. In addition to this, the intl instruction has the operand *imm3* to specify the interrupt level (0–7) to be set to the IL[2:0] bits in the PSR. The software interrupt cannot be disabled (non-maskable interrupt). The processor performs the same interrupt processing operation as that of the hardware interrupt.

5.7 Interrupt Processing by the CPU

The CPU samples interrupt requests for each cycle. On accepting an interrupt request, the CPU switches to interrupt processing immediately after execution of the current instruction has been completed.
Interrupt processing involves the following steps:

1. The PSR and current program counter (PC) values are saved to the stack.
2. The PSR IE bit is cleared to 0 (disabling subsequent maskable interrupts).
3. The PSR IL[2:0] bits are set to the received interrupt level. (The NMI does not affect the IL bits.)
4. The vector for the interrupt occurred is loaded to the PC to execute the interrupt handler routine.

When an interrupt is accepted, Step 2 prevents subsequent maskable interrupts. Setting the IE bit to 1 in the interrupt handler routine allows handling of multiple interrupts. In this case, since the IL[2:0] bits are changed by Step 3, only an interrupt with a higher level than that of the currently processed interrupt will be accepted.
Ending interrupt handler routines using the reti instruction returns the PSR to the state before the interrupt occurred. The program resumes processing following the instruction being executed at the time the interrupt occurred.

Note: When HALT or SLEEP mode is canceled, the CPU jumps to the interrupt handler routine after executing one instruction. To execute the interrupt handler routine immediately after HALT or SLEEP mode is canceled, place the nop instruction at just behind the halt/slp instruction.

5.8 Control Registers

MISC Vector Table Address Low Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
MSCTTBRL	15–8	TTBR[15:8]	0x80	H0	R/WP	–
	7–0	TTBR[7:0]	0x00	H0	R	

Bits 15–0 TTBR[15:0]

These bits set the vector table base address (16 low-order bits).

MISC Vector Table Address High Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
MSCTTBRH	15–8	–	0x00	–	R	–
	7–0	TTBR[23:16]	0x00	H0	R/WP	

Bits 15–8 Reserved

Bits 7–0 TTBR[23:16]

These bits set the vector table base address (eight high-order bits).

ITC Interrupt Level Setup Register *x*

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
ITCLV*x*	15–11	–	0x00	–	R	–
	10–8	ILVy_1[2:0]	0x0	H0	R/W	
	7–3	–	0x00	–	R	
	2–0	ILVy_0[2:0]	0x0	H0	R/W	

Bits 15–11 Reserved

Bits 7–3 Reserved

Bits 10–8 ILVy_1[2:0] ($y_1 = 2x + 1$)

Bits 2–0 ILVy_0[2:0] ($y_0 = 2x$)

These bits set the interrupt level of each interrupt.

Table 5.8.1 Interrupt Level and Priority Settings

ITCLV*x*.ILV*y*[2:0] bits	Interrupt level	Priority
0x7	7	High
0x6	6	↑
. . .	. . .	
0x1	1	↓
0x0	0	Low

The following shows the ITCLV*x* register configuration in this IC.

Table 5.8.2 List of ITCLV*x* Registers

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
ITCLV0 (ITC Interrupt Level Setup Register 0)	15–11	–	0x00	–	R	–
	10–8	ILV1[2:0]	0x0	H0	R/W	Port interrupt (ILVPPORT)
	7–3	–	0x00	–	R	–
	2–0	ILV0[2:0]	0x0	H0	R/W	Supply voltage detector interrupt (ILVSVD4)
ITCLV1 (ITC Interrupt Level Setup Register 1)	15–11	–	0x00	–	R	–
	10–8	ILV3[2:0]	0x0	H0	R/W	Clock generator interrupt (ILVCLG)
	7–0	–	0x00	–	R	–
ITCLV2 (ITC Interrupt Level Setup Register 2)	15–11	–	0x00	–	R	–
	10–8	ILV5[2:0]	0x0	H0	R/W	16-bit timer Ch.0 interrupt (ILVT16_0)
	7–0	–	0x00	–	R	–
ITCLV3 (ITC Interrupt Level Setup Register 3)	15–11	–	0x00	–	R	–
	10–8	ILV7[2:0]	0x0	H0	R/W	16-bit timer Ch.1 interrupt (ILVT16_1)
	7–3	–	0x00	–	R	–
	2–0	ILV6[2:0]	0x0	H0	R/W	UART Ch.0 interrupt (ILVUART3_0)
ITCLV4 (ITC Interrupt Level Setup Register 4)	15–11	–	0x00	–	R	–
	10–8	ILV9[2:0]	0x0	H0	R/W	I²C interrupt (ILVI2C_0)
	7–3	–	0x00	–	R	–
	2–0	ILV8[2:0]	0x0	H0	R/W	Synchronous serial interface Ch.0 interrupt (ILVSPIA_0)
ITCLV5 (ITC Interrupt Level Setup Register 5)	15–11	–	0x00	–	R	–
	10–8	ILV11[2:0]	0x0	H0	R/W	DMM 16-bit PWM timer Ch.1 interrupt (ILVT16B_DMM_1)
	7–3	–	0x00	–	R	–
	2–0	ILV10[2:0]	0x0	H0	R/W	DMM 16-bit PWM timer Ch.0 interrupt (ILVT16B_DMM_0)
ITCLV6 (ITC Interrupt Level Setup Register 6)	15–11	–	0x00	–	R	–
	10–8	ILV13[2:0]	0x0	H0	R/W	Sound generator interrupt (ILVSNDA_DMM_0)
	7–0	–	0x00	–	R	–
ITCLV7 (ITC Interrupt Level Setup Register 7)	15–11	–	0x00	–	R	–
	10–8	ILV15[2:0]	0x0	H0	R/W	LCD driver interrupt (ILVLCD4B)
	7–0	–	0x00	–	R	–
ITCLV8 (ITC Interrupt Level Setup Register 8)	15–11	–	0x00	–	R	–
	10–8	ILV17[2:0]	0x0	H0	R/W	EEPROM controller interrupt (ILVEPRC)
	7–0	–	0x00	–	R	–
ITCLV9 (ITC Interrupt Level Setup Register 9)	15–8	–	0x00	–	R	–
	7–3	–	0x00	–	R	–
	2–0	ILV18[2:0]	0x0	H0	R/W	16-bit timer Ch.2 interrupt (ILVT16_2)
ITCLV10 (ITC Interrupt Level Setup Register 10)	15–11	–	0x00	–	R	–
	10–8	ILV21[2:0]	0x0	–	R/W	DMM controller interrupt (ILVDSADC16_0)
	7–3	–	0x00	–	R	–
	2–0	ILV20[2:0]	0x0	–	R/W	16-bit timer Ch.3 interrupt (ILVT16_3)
ITCLV11 (ITC Interrupt Level Setup Register 11)	15–8	–	0x00	–	R	–
	7–3	–	0x00	–	R	
	2–0	ILV22[2:0]	0x0	–	R/W	DMM 16-bit PWM timer Ch.2 interrupt (ILVT16B_DMM_2)

6 I/O Ports (PPORT)

6.1 Overview

PPORT controls the I/O ports. The main features are outlined below.

- Allows port-by-port function configurations.
  - Each port can be configured with or without a pull-up or pull-down resistor.
  - Each port can be configured with or without a chattering filter.
  - Allows selection of the function (general-purpose I/O port (GPIO) function, up to four peripheral I/O functions) to be assigned to each port.
- Ports, except for those shared with debug pins, are initially placed into Hi-Z state.
  (No current passes through the pin during this Hi-Z state.)

Note: '*x*', which is used in the port names P*xy*, register names, and bit names, refers to a port group (*x* = 0, 1, 2, ··· , d) and '*y*' refers to a port number (*y* = 0, 1, 2, ··· , 7).

Figure 6.1.1 shows the configuration of PPORT.

Table 6.1.1 Port Configuration of S1C17M02/M03

Item		S1C17M02	S1C17M03
Port groups included	P0	P0[7:0] (8) *1, *2	P0[7:0] (8) *1, *2
	P1	P1[7:0] (8) *1, *2	P1[7:0] (8) *1, *2
	P2	–	P2[7:0] (8) *1, *2
	P3	–	P3[7:0] (8) *1, *2
	P4	–	P4[7:0] (8) *1, *2
	Pd	Pd[2:0] (Pd2: output only) (3) *1	Pd[2:0] (Pd2: output only) (3) *1
Total number of ports		Input/output port: 18 Output port: 1	Input/output port: 42 Output port: 1
Ports for debug function		Pd[2:0]	
Key-entry reset function		Supported (P0[3:0])	

*1 Ports with general-purpose I/O function (GPIO)
*2 Ports with interrupt function

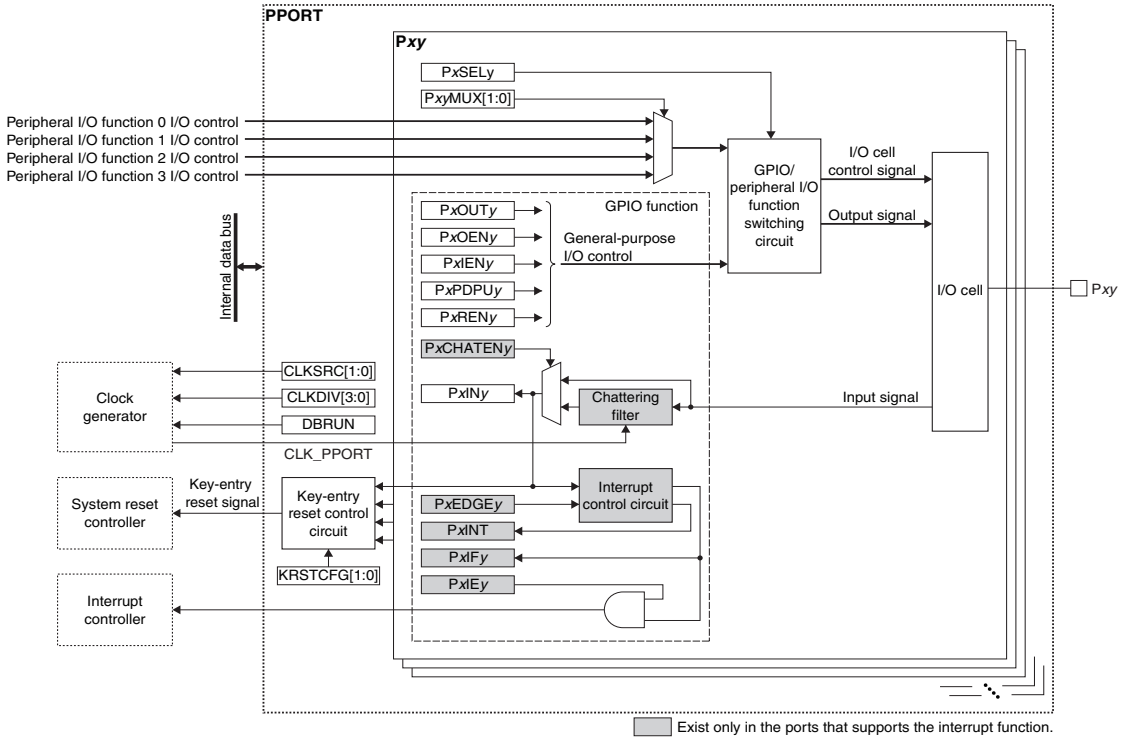


Figure 6.1.1 PPORT Configuration

6.2 I/O Cell Structure and Functions

Figure 6.2.1 shows the I/O cell Configuration.

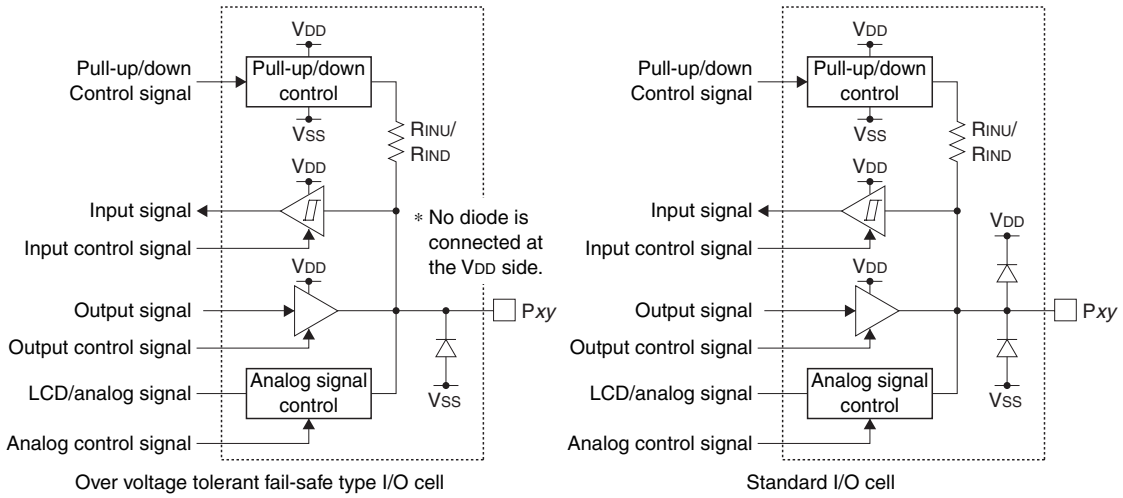


Figure 6.2.1 I/O Cell Configuration

Refer to “Pin Descriptions” in the “Overview” chapter for the cell type, either the over voltage tolerant fail-safe type I/O cell or the standard I/O cell, included in each port.

6.2.1 Schmitt Input

The input functions are all configured with the Schmitt interface level. When a port is set to input disable status (P*x*IOEN.P*x*IEN*y* bit = 0), unnecessary current is not consumed if the P*xy* pin is placed into floating status.

6.2.2 Over Voltage Tolerant Fail-Safe Type I/O Cell

The over voltage tolerant fail-safe type I/O cell allows interfacing without passing unnecessary current even if a voltage exceeding V_{DD} is applied to the port. Also unnecessary current is not consumed when the port is externally biased without supplying V_{DD}. However, be sure to avoid applying a voltage exceeding the recommended maximum operating power supply voltage to the port.

6.2.3 Pull-Up/Pull-Down

The GPIO port has a pull-up/pull-down function. Either pull-up or pull-down may be selected for each port individually. This function may also be disabled for the port that does not require pulling up/down.
When the port level is switched from low to high through the pull-up resistor included in the I/O cell or from high to low through the pull-down resistor, a delay will occur in the waveform rising/falling edge depending on the time constant by the pull-up/pull-down resistance and the pin load capacitance. The rising/falling time is commonly determined by the following equation:

$$t_{PR} = -R_{INU} \times (C_{IN} + C_{BOARD}) \times \ln(1 - V_{T+}/V_{DD}) \quad \text{(Eq. 6.1)}$$
$$t_{PF} = -R_{IND} \times (C_{IN} + C_{BOARD}) \times \ln(1 - V_{T-}/V_{DD})$$

Where

- t_{PR}: Rising time (port level = low → high) [second]
- t_{PF}: Falling time (port level = high → low) [second]
- V_{T+}: High level Schmitt input threshold voltage [V]
- V_{T-}: Low level Schmitt input threshold voltage [V]
- R_{INU}/R_{IND}: Pull-up/pull-down resistance [Ω]
- C_{IN}: Pin capacitance [F]
- C_{BOARD}: Parasitic capacitance on the board [F]

6.2.4 CMOS Output and High Impedance State

The I/O cells except for analog output can output signals in the V_{DD} and V_{SS} levels. Also the GPIO ports may be put into high-impedance (Hi-Z) state.

6.3 Clock Settings

6.3.1 PPORT Operating Clock

When using the chattering filter for entering external signals to PPORT, the PPORT operating clock CLK_PPORT must be supplied to PPORT from the clock generator.
The CLK_PPORT supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to “Clock Generator” in the “Power Supply, Reset, and Clocks” chapter).
2. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
3. Set the following PCLK register bits:
   - PCLK.CLKSRC[1:0] bits (Clock source selection)
   - PCLK.CLKDIV[3:0] bits (Clock division ratio selection = Clock frequency setting)
4. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

Settings in Step 3 determine the input sampling time of the chattering filter.

6.3.2 Clock Supply in SLEEP Mode

When using the chattering filter function during SLEEP mode, the PPORT operating clock CLK_PPORT must be configured so that it will keep suppling by writing 0 to the CLGOSC.*xxxx*SLPC bit for the CLK_PPORT clock source.
If the CLGOSC.*xxxx*SLPC bit for the CLK_PPORT clock source is 1, the CLK_PPORT clock source is deactivated during SLEEP mode and it disables the chattering filter function regardless of the P*x*CHATEN.P*x*CHATEN*y* bit setting (chattering filter enabled/disabled).

6.3.3 Clock Supply in DEBUG Mode

The CLK_PPORT supply during DEBUG mode should be controlled using the PCLK.DBRUN bit.
The CLK_PPORT supply to PPORT is suspended when the CPU enters DEBUG mode if the PCLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_PPORT supply resumes. The PPORT chattering filter stops operating when the CLK_PPORT supply is suspended. If the chattering filter is enabled in PPORT, the input port function is also deactivated. However, the control registers can be altered. If the PCLK.DBRUN bit = 1, the CLK_PPORT supply is not suspended and the chattering filter will keep operating in DEBUG mode.

6.4 Operations

6.4.1 Initialization

After a reset, the ports except for the debugging function are configured as shown below.

- Port input: Disabled
- Port output: Disabled
- Pull-up: Off
- Pull-down: Off
- Port pins: High impedance state
- Port function: Configured to GPIO

This status continues until the ports are configured via software. The debugging function ports are configured for debug signal input/output.

Initial settings when using a port for a peripheral I/O function

When using the P*xy* port for a peripheral I/O function, perform the following software initial settings:

1. Set the following P*x*IOEN register bits:
   - Set the P*x*IOEN.P*x*IEN*y* bit to 0. (Disable input)
   - Set the P*x*IOEN.P*x*OEN*y* bit to 0. (Disable output)
2. Set the P*x*MODSEL.P*x*SEL*y* bit to 0. (Disable peripheral I/O function)
3. Initialize the peripheral circuit that uses the pin.
4. Set the P*x*FNCSEL.P*xy*MUX[1:0] bits. (Select peripheral I/O function)
5. Set the P*x*MODSEL.P*x*SEL*y* bit to 1. (Enable peripheral I/O function)

For the list of the peripheral I/O functions that can be assigned to each port of this IC, refer to "Control Register and Port Function Configuration of this IC." For the specific information on the peripheral I/O functions, refer to the respective peripheral circuit chapter.

Initial settings when using a port as a general-purpose output port (only for the ports with GPIO function)

When using the P*xy* port pin as a general-purpose output pin, perform the following software initial settings:

1. Set the P*x*IOEN.P*x*OEN*y* bit to 1. (Enable output)
2. Set the P*x*MODSEL.P*x*SEL*y* bit to 0. (Enable GPIO function)

Initial settings when using a port as a general-purpose input port (only for the ports with GPIO function)

When using the P*xy* port pin as a general-purpose input pin, perform the following software initial settings:

1. Write 0 to the P*x*INTCTL.P*x*IE*y* bit. * (Disable interrupt)
2. When using the chattering filter, configure the PPORT operating clock (see "PPORT Operating Clock") and set the P*x*CHATEN.P*x*CHATEN*y* bit to 1. *

   When the chattering filter is not used, set the P*x*CHATEN.P*x*CHATEN*y* bit to 0 (supply of the PPORT operating clock is not required).
3. Configure the following P*x*RCTL register bits when pulling up/down the port using the internal pull-up or down resistor:
   - P*x*RCTL.P*x*PDPU*y* bit (Select pull-up or pull-down resistor)
   - Set the P*x*RCTL.P*x*REN*y* bit to 1. (Enable pull-up/down)

   Set the P*x*RCTL.P*x*REN*y* bit to 0 if the internal pull-up/down resistors are not used.
4. Set the P*x*MODSEL.P*x*SEL*y* bit to 0. (Enable GPIO function)
5. Configure the following bits when using the port input interrupt: *
   - Write 1 to the P*x*INTF.P*x*IF*y* bit. (Clear interrupt flag)
   - P*x*INTCTL.P*x*EDGE*y* bit (Select interrupt edge (input rising edge/falling edge))
   - Set the P*x*INTCTL.P*x*IE*y* bit to 1. (Enable interrupt)
6. Set the following P*x*IOEN register bits:
   - Set the P*x*IOEN.P*x*OEN*y* bit to 0. (Disable output)
   - Set the P*x*IOEN.P*x*IEN*y* bit to 1. (Enable input)

∗ Steps 1 and 5 are required for the ports with an interrupt function. Step 2 is required for the ports with a chattering filter function.

Table 6.4.1.1 lists the port status according to the combination of data input/output control and pull-up/down control.

Table 6.4.1.1 GPIO Port Control List

PxIOEN. PxIENy bit	PxIOEN. PxOENy bit	PxRCTL. PxRENy bit	PxRCTL. PxPDPUy bit	Input	Output	Pull-up/pull-down condition
0	0	0	×	Disabled		Off (Hi-Z) ∗1
0	0	1	0	Disabled		Pulled down
0	0	1	1	Disabled		Pulled up
1	0	0	×	Enabled	Disabled	Off (Hi-Z) ∗2
1	0	1	0	Enabled	Disabled	Pulled down
1	0	1	1	Enabled	Disabled	Pulled up
0	1	0	×	Disabled	Enabled	Off
0	1	1	0	Disabled	Enabled	Off
0	1	1	1	Disabled	Enabled	Off
1	1	1	0	Enabled	Enabled	Off
1	1	1	1	Enabled	Enabled	Off

∗1: Initial status. Current does not flow if the pin is placed into floating status.
∗2: Use of the pull-up or pull-down function is recommended, as undesired current will flow if the port input is set to floating status.

Note: If the P*x*MODSEL.P*x*SEL*y* bit for the port without a GPIO function is set to 0, the port goes into initial status (refer to "Initial Settings"). The GPIO control bits are configured to a read-only bit always read out as 0.

6.4.2 Port Input/Output Control

Peripheral I/O function control

The port for which a peripheral I/O function is selected is controlled by the peripheral circuit. For more information, refer to the respective peripheral circuit chapter.

Setting output data to a GPIO port

Write data (1 = high output, 0 = low output) to be output from the P*xy* pin to the P*x*DAT.P*x*OUT*y* bit.

Reading input data from a GPIO port

The data (1 = high input, 0 = low input) input from the P*xy* pin can be read out from the P*x*DAT.P*x*IN*y* bit.

Chattering filter function

Some ports have a chattering filter function and it can be controlled in each port. This function is enabled by setting the P*x*CHATEN.P*x*CHATEN*y* bit to 1. The input sampling time to remove chattering is determined by the CLK_PPORT frequency configured using the PCLK register in common to all ports. The chattering filter removes pulses with a shorter width than the input sampling time.

$$\text{Input sampling time} = \frac{2 \text{ to } 3}{\text{CLK\_PPORT frequency [Hz]}} \text{ [second]} \qquad \text{(Eq.6.2)}$$

Make sure the P*xy* port interrupt is disabled before altering the PCLK register and P*x*CHATEN.P*x*CHATEN*y* bit settings. A P*xy* port interrupt may erroneously occur if these settings are altered in an interrupt enabled status. Furthermore, enable the interrupt after a lapse of four or more CLK_PPORT cycles from enabling the chattering filter function.
If the clock generator is configured so that it will supply CLK_PPORT to PPORT in SLEEP mode, the chattering filter of the port will function even in SLEEP mode. If CLK_PPORT is configured to stop in SLEEP mode, PPORT inactivates the chattering filter during SLEEP mode to input pin status transitions directly to itself.

Key-entry reset function

This function issues a reset request when low-level pulses are input to all the specified ports simultaneously. Make the following settings when using this function:

1. Configure the ports to be used for key-entry reset as general-purpose input ports (refer to "Initial settings when using a port as a general-purpose input port (only for the ports with GPIO function)").
2. Configure the input pin combination for key-entry reset using the PCLK.KRSTCFG[1:0] bits.

Note: When enabling the key-entry reset function, be sure to configure the port pins to be used for it as general-purpose input pins before setting the PCLK.KRSTCFG[1:0] bits.

PPORT issues a reset request immediately after all the input pins specified by the PCLK.KRSTCFG[1:0] are set to a low level if the chattering filter function is disabled (initial status). To issue a reset request only when low-level signals longer than the time configured are input, enable the chattering filter function for all the ports used for key-entry reset.
The pins configured for key-entry reset can also be used as general-purpose input pins.

6.5 Interrupts

When the GPIO function is selected for the port with an interrupt function, the port input interrupt function can be used.

Table 6.5.1 Port Input Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Port input interrupt	P*x*INTF.P*x*IF*y*	Rising or falling edge of the input signal	Writing 1
	PINTFGRP.P*x*INT	Setting an interrupt flag in the port group	Clearing P*x*INTF.P*x*IF*y*

Interrupt edge selection

Port input interrupts will occur at the falling edge of the input signal when setting the P*x*INTCTL.P*x*EDGE*y* bit to 1, or the rising edge when setting to 0.

Interrupt enable

PPORT provides interrupt enable bits (P*x*INTCTL.P*x*IE*y* bit) corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the "Interrupt Controller" chapter.

Interrupt check in port group unit

When interrupts are enabled in two or more port groups, check the PINTFGRP.P*x*INT bit in the interrupt handler first. It helps minimize the handler codes for finding the port that has generated an interrupt. If this bit is set to 1, an interrupt has occurred in the port group. Next, check the P*x*INTF.P*x*IF*y* bit set to 1 in the port group to determine the port that has generated an interrupt. Clearing the P*x*INTF.P*x*IF*y* bit also clears the PINTFGRP.P*x*INT bit. If the port is set to interrupt disabled status by the P*x*INTCTL.P*x*IE*y* bit, the PINTFGRP.P*x*INT bit will not be set even if the P*x*INTF.P*x*IF*y* bit is set to 1.

6.6 Control Registers

This section describes the same control registers of all port groups as a single register. For the register and bit configurations in each port group and their initial values, refer to "Control Register and Port Function Configuration of this IC."

P*x* Port Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
P*x*DAT	15–8	P*x*OUT[7:0]	0x00	H0	R/W	–
	7–0	P*x*IN[7:0]	0x00	H0	R	

∗1: This register is effective when the GPIO function is selected.
∗2: The bit configuration differs depending on the port group.
∗3: The initial value may be changed by the port.

Bits 15–8 P*x*OUT[7:0]

These bits are used to set data to be output from the GPIO port pins.
1 (R/W): Output high level from the port pin
0 (R/W): Output low level from the port pin

When output is enabled (P*x*IOEN.P*x*OEN*y* bit = 1), the port pin outputs the data set here. Although data can be written when output is disabled (P*x*IOEN.P*x*OEN*y* bit = 0), it does not affect the pin status. These bits do not affect the outputs when the port is used as a peripheral I/O function.

Bits 7–0 P*x*IN[7:0]

The GPIO port pin status can be read out from these bits.
1 (R): Port pin = High level
0 (R): Port pin = Low level

The port pin status can be read out when input is enabled (P*x*IOEN.P*x*IEN*y* bit = 1). When input is disabled (P*x*IOEN.P*x*IEN*y* bit = 0), these bits are always read as 0.
When the port is used for a peripheral I/O function, the input value cannot be read out from these bits.

P*x* Port Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
P*x*IOEN	15–8	P*x*IEN[7:0]	0x00	H0	R/W	–
	7–0	P*x*OEN[7:0]	0x00	H0	R/W	

∗1: This register is effective when the GPIO function is selected.
∗2: The bit configuration differs depending on the port group.

Bits 15–8 P*x*IEN[7:0]

These bits enable/disable the GPIO port input.
1 (R/W): Enable (The port pin status is input.)
0 (R/W): Disable (Input data is fixed at 0.)

When both data output and data input are enabled, the pin output status controlled by this IC can be read.
These bits do not affect the input control when the port is used as a peripheral I/O function.

Bits 7–0 P*x*OEN[7:0]

These bits enable/disable the GPIO port output.
1 (R/W): Enable (Data is output from the port pin.)
0 (R/W): Disable (The port is placed into Hi-Z.)

These bits do not affect the output control when the port is used as a peripheral I/O function.

P*x* Port Pull-up/down Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
P*x*RCTL	15–8	P*x*PDPU[7:0]	0x00	H0	R/W	–
	7–0	P*x*REN[7:0]	0x00	H0	R/W	

∗1: This register is effective when the GPIO function is selected.
∗2: The bit configuration differs depending on the port group.

Bits 15–8 P*x*PDPU[7:0]

These bits select either the pull-up resistor or the pull-down resistor when using a resistor built into the port.
1 (R/W): Pull-up resistor
0 (R/W): Pull-down resistor

The selected pull-up/down resistor is enabled when the P*x*RCTL.P*x*REN*y* bit = 1.

Bits 7–0 P*x*REN[7:0]

These bits enable/disable the port pull-up/down control.
1 (R/W): Enable (The built-in pull-up/down resistor is used.)
0 (R/W): Disable (No pull-up/down control is performed.)

Enabling this function pulls up or down the port when output is disabled (P*x*IOEN.P*x*OEN*y* bit = 0). When output is enabled (P*x*IOEN.P*x*OEN*y* bit = 1), the P*x*RCTL.P*x*REN*y* bit setting is ineffective regardless of how the P*x*IOEN.P*x*IEN*y* bit is set and the port is not pulled up/down.
These bits do not affect the pull-up/down control when the port is used as a peripheral I/O function.

P*x* Port Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
P*x*INTF	15–8	–	0x00	–	R	–
	7–0	P*x*IF[7:0]	0x00	H0	R/W	Cleared by writing 1.

∗1: This register is effective when the GPIO function is selected.
∗2: The bit configuration differs depending on the port group.

Bits 15–8 Reserved

Bits 7–0 P*x*IF[7:0]

These bits indicate the port input interrupt cause occurrence status.
1 (R): Cause of interrupt occurred
0 (R): No cause of interrupt occurred
1 (W): Clear flag
0 (W): Ineffective

P*x* Port Interrupt Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
P*x*INTCTL	15–8	P*x*EDGE[7:0]	0x00	H0	R/W	–
	7–0	P*x*IE[7:0]	0x00	H0	R/W	

∗1: This register is effective when the GPIO function is selected.
∗2: The bit configuration differs depending on the port group.

Bits 15–8 P*x*EDGE[7:0]

These bits select the input signal edge to generate a port input interrupt.
1 (R/W): An interrupt will occur at a falling edge.
0 (R/W): An interrupt will occur at a rising edge.

Bits 7–0 P*x*IE[7:0]

These bits enable port input interrupts.
1 (R/W): Enable interrupts
0 (R/W): Disable interrupts

Note: To prevent generating unnecessary interrupts, the corresponding interrupt flag should be cleared before enabling interrupts.

P*x* Port Chattering Filter Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
P*x*CHATEN	15–8	–	0x00	–	R	–
	7–0	P*x*CHATEN[7:0]	0x00	H0	R/W	

∗1: The bit configuration differs depending on the port group.

Bits 15–8 Reserved

Bits 7–0 P*x*CHATEN[7:0]

These bits enable/disable the chattering filter function.
1 (R/W): Enable (The chattering filter is used.)
0 (R/W): Disable (The chattering filter is bypassed.)

P*x* Port Mode Select Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
P*x*MODSEL	15–8	–	0x00	–	R	–
	7–0	P*x*SEL[7:0]	0x00	H0	R/W	

∗1: The bit configuration differs depending on the port group.
∗2: The initial value may be changed by the port.

Bits 15–8 Reserved

Bits 7–0 P*x*SEL[7:0]

These bits select whether each port is used for the GPIO function or a peripheral I/O function.
1 (R/W): Use peripheral I/O function
0 (R/W): Use GPIO function

P*x* Port Function Select Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
P*x*FNCSEL	15–14	P*x*7MUX[1:0]	0x0	H0	R/W	–
	13–12	P*x*6MUX[1:0]	0x0	H0	R/W	
	11–10	P*x*5MUX[1:0]	0x0	H0	R/W	
	9–8	P*x*4MUX[1:0]	0x0	H0	R/W	
	7–6	P*x*3MUX[1:0]	0x0	H0	R/W	
	5–4	P*x*2MUX[1:0]	0x0	H0	R/W	
	3–2	P*x*1MUX[1:0]	0x0	H0	R/W	
	1–0	P*x*0MUX[1:0]	0x0	H0	R/W	

∗1: The bit configuration differs depending on the port group.
∗2: The initial value may be changed by the port.

Bits 15–14 P*x*7MUX[1:0]

: :

Bits 1–0 P*x*0MUX[1:0]

These bits select the peripheral I/O function to be assigned to each port pin.

Table 6.6.1 Selecting Peripheral I/O Function

PxFNCSEL.Pxy MUX[1:0] bits	Peripheral I/O function
0x3	Function 3
0x2	Function 2
0x1	Function 1
0x0	Function 0

This selection takes effect when the P*x*MODSEL.P*x*SEL*y* bit = 1.

P Port Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PCLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/WP	
	7–4	CLKDIV[3:0]	0x0	H0	R/WP	
	3–2	KRSTCFG[1:0]	0x0	H0	R/WP	
	1–0	CLKSRC[1:0]	0x0	H0	R/WP	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the PPORT operating clock is supplied in DEBUG mode or not.
1 (R/WP): Clock supplied in DEBUG mode
0 (R/WP): No clock supplied in DEBUG mode

Bits 7–4 CLKDIV[3:0]

These bits select the division ratio of the PPORT operating clock (chattering filter clock).

Bits 3–2 KRSTCFG[1:0]

These bits configure the key-entry reset function.

Table 6.6.2 Key-Entry Reset Function Settings

PCLK.KRSTCFG[1:0] bits	key-entry reset
0x3	Reset when P0[3:0] inputs = all low
0x2	Reset when P0[2:0] inputs = all low
0x1	Reset when P0[1:0] inputs = all low
0x0	Disable

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of PPORT (chattering filter).
The PPORT operating clock should be configured by selecting the clock source using the PCLK.CLKSRC[1:0] bits and the clock division ratio using the PCLK.CLKDIV[3:0] bits as shown in Table 6.6.3. These settings determine the input sampling time of the chattering filter.

Table 6.6.3 Clock Source and Division Ratio Settings

PCLK.CLKDIV[3:0] bits	PCLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0xf	1/32,768			1/1
0xe	1/16,384			
0xd	1/8,192			
0xc	1/4,096			
0xb	1/2,048			
0xa	1/1,024			
0x9	1/512			
0x8	1/256			
0x7	1/128			
0x6	1/64			
0x5	1/32			
0x4	1/16			
0x3	1/8			
0x2	1/4			
0x1	1/2			
0x0	1/1			

(Note) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

P Port Interrupt Flag Group Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PINTFGRP	15–13	–	0x0	–	R	–
	12	PcINT	0	H0	R	
	11	PbINT	0	H0	R	
	10	PaINT	0	H0	R	
	9	P9INT	0	H0	R	
	8	P8INT	0	H0	R	
	7	P7INT	0	H0	R	
	6	P6INT	0	H0	R	
	5	P5INT	0	H0	R	
	4	P4INT	0	H0	R	
	3	P3INT	0	H0	R	
	2	P2INT	0	H0	R	
	1	P1INT	0	H0	R	
	0	P0INT	0	H0	R	

∗1: Only the bits corresponding to the port groups that support interrupts are provided.

Bits 15–13 Reserved

Bits 12–0 P*x*INT

These bits indicate that P*x* port group includes a port that has generated an interrupt.

1 (R): A port generated an interrupt

0 (R): No port generated an interrupt

The PINTFGRP.P*x*INT bit is cleared when the interrupt flag for the port that has generated an interrupt is cleared.

6.7 Control Register and Port Function Configuration of this IC

This section shows the PPORT control register/bit configuration in this IC and the list of peripheral I/O functions selectable for each port.

Note: The control bits for the ports that are not available in the model are reserved bits. Do not alter them from the initial value.

6.7.1 P0 Port Group

The P0 port group supports the GPIO and interrupt functions.

Table 6.7.1.1 Control Registers for P0 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
P0DAT (P0 Port Data Register)	15	P0OUT7	0	H0	R/W	–	✓	✓
	14	P0OUT6	0	H0	R/W		✓	✓
	13	P0OUT5	0	H0	R/W		✓	✓
	12	P0OUT4	0	H0	R/W		✓	✓
	11	P0OUT3	0	H0	R/W		✓	✓
	10	P0OUT2	0	H0	R/W		✓	✓
	9	P0OUT1	0	H0	R/W		✓	✓
	8	P0OUT0	0	H0	R/W		✓	✓
	7	P0IN7	0	H0	R	–	✓	✓
	6	P0IN6	0	H0	R		✓	✓
	5	P0IN5	0	H0	R		✓	✓
	4	P0IN4	0	H0	R		✓	✓
	3	P0IN3	0	H0	R		✓	✓
	2	P0IN2	0	H0	R		✓	✓
	1	P0IN1	0	H0	R		✓	✓
	0	P0IN0	0	H0	R		✓	✓
P0IOEN (P0 Port Enable Register)	15	P0IEN7	0	H0	R/W	–	✓	✓
	14	P0IEN6	0	H0	R/W		✓	✓
	13	P0IEN5	0	H0	R/W		✓	✓
	12	P0IEN4	0	H0	R/W		✓	✓
	11	P0IEN3	0	H0	R/W		✓	✓
	10	P0IEN2	0	H0	R/W		✓	✓
	9	P0IEN1	0	H0	R/W		✓	✓
	8	P0IEN0	0	H0	R/W		✓	✓
	7	P0OEN7	0	H0	R/W	–	✓	✓
	6	P0OEN6	0	H0	R/W		✓	✓
	5	P0OEN5	0	H0	R/W		✓	✓
	4	P0OEN4	0	H0	R/W		✓	✓
	3	P0OEN3	0	H0	R/W		✓	✓
	2	P0OEN2	0	H0	R/W		✓	✓
	1	P0OEN1	0	H0	R/W		✓	✓
	0	P0OEN0	0	H0	R/W		✓	✓

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
P0RCTL	15	P0PDPU7	0	H0	R/W	–	✓	✓
(P0 Port Pull-up/down	14	P0PDPU6	0	H0	R/W		✓	✓
Control Register)	13	P0PDPU5	0	H0	R/W		✓	✓
	12	P0PDPU4	0	H0	R/W		✓	✓
	11	P0PDPU3	0	H0	R/W		✓	✓
	10	P0PDPU2	0	H0	R/W		✓	✓
	9	P0PDPU1	0	H0	R/W		✓	✓
	8	P0PDPU0	0	H0	R/W		✓	✓
	7	P0REN7	0	H0	R/W	–	✓	✓
	6	P0REN6	0	H0	R/W		✓	✓
	5	P0REN5	0	H0	R/W		✓	✓
	4	P0REN4	0	H0	R/W		✓	✓
	3	P0REN3	0	H0	R/W		✓	✓
	2	P0REN2	0	H0	R/W		✓	✓
	1	P0REN1	0	H0	R/W		✓	✓
	0	P0REN0	0	H0	R/W		✓	✓
P0INTF	15–8	–	0x00	–	R	–	–	–
(P0 Port Interrupt	7	P0IF7	0	H0	R/W	Cleared by writing 1.	✓	✓
Flag Register)	6	P0IF6	0	H0	R/W		✓	✓
	5	P0IF5	0	H0	R/W		✓	✓
	4	P0IF4	0	H0	R/W		✓	✓
	3	P0IF3	0	H0	R/W		✓	✓
	2	P0IF2	0	H0	R/W		✓	✓
	1	P0IF1	0	H0	R/W		✓	✓
	0	P0IF0	0	H0	R/W		✓	✓
P0INTCTL	15	P0EDGE7	0	H0	R/W	–	✓	✓
(P0 Port Interrupt	14	P0EDGE6	0	H0	R/W		✓	✓
Control Register)	13	P0EDGE5	0	H0	R/W		✓	✓
	12	P0EDGE4	0	H0	R/W		✓	✓
	11	P0EDGE3	0	H0	R/W		✓	✓
	10	P0EDGE2	0	H0	R/W		✓	✓
	9	P0EDGE1	0	H0	R/W		✓	✓
	8	P0EDGE0	0	H0	R/W		✓	✓
	7	P0IE7	0	H0	R/W	–	✓	✓
	6	P0IE6	0	H0	R/W		✓	✓
	5	P0IE5	0	H0	R/W		✓	✓
	4	P0IE4	0	H0	R/W		✓	✓
	3	P0IE3	0	H0	R/W		✓	✓
	2	P0IE2	0	H0	R/W		✓	✓
	1	P0IE1	0	H0	R/W		✓	✓
	0	P0IE0	0	H0	R/W		✓	✓
P0CHATEN	15–8	–	0x00	–	R	–	–	–
(P0 Port Chattering	7	P0CHATEN7	0	H0	R/W	–	✓	✓
Filter Enable Register)	6	P0CHATEN6	0	H0	R/W		✓	✓
	5	P0CHATEN5	0	H0	R/W		✓	✓
	4	P0CHATEN4	0	H0	R/W		✓	✓
	3	P0CHATEN3	0	H0	R/W		✓	✓
	2	P0CHATEN2	0	H0	R/W		✓	✓
	1	P0CHATEN1	0	H0	R/W		✓	✓
	0	P0CHATEN0	0	H0	R/W		✓	✓

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
P0MODSEL (P0 Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–
	7	P0SEL7	0	H0	R/W	–	✓	✓
	6	P0SEL6	0	H0	R/W		✓	✓
	5	P0SEL5	0	H0	R/W		✓	✓
	4	P0SEL4	0	H0	R/W		✓	✓
	3	P0SEL3	0	H0	R/W		✓	✓
	2	P0SEL2	0	H0	R/W		✓	✓
	1	P0SEL1	0	H0	R/W		✓	✓
	0	P0SEL0	0	H0	R/W		✓	✓
P0FNCSEL (P0 Port Function Select Register)	15–14	P07MUX[1:0]	0x0	H0	R/W	–	✓	✓
	13–12	P06MUX[1:0]	0x0	H0	R/W		✓	✓
	11–10	P05MUX[1:0]	0x0	H0	R/W		✓	✓
	9–8	P04MUX[1:0]	0x0	H0	R/W		✓	✓
	7–6	P03MUX[1:0]	0x0	H0	R/W		✓	✓
	5–4	P02MUX[1:0]	0x0	H0	R/W		✓	✓
	3–2	P01MUX[1:0]	0x0	H0	R/W		✓	✓
	1–0	P00MUX[1:0]	0x0	H0	R/W		✓	✓

Table 6.7.1.2 P0 Port Group Function Assignment

Port name	P0SEL*y* = 0	P0SEL*y* = 1								M02	M03
	GPIO	P0*y*MUX = 0x0 (Function 0)		P0*y*MUX = 0x1 (Function 1)		P0*y*MUX = 0x2 (Function 2)		P0*y*MUX = 0x3 (Function 3)			
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin		
P00	P00	CLG	EXOSC	UPMUX	*1	–	–	–	–	✓	✓
P01	P01	LCD4B	LFRO	UPMUX	*1	–	–	–	–	✓	✓
P02	P02	–	–	UPMUX	*1	SVD4	EXSVD0	–	–	✓	✓
P03	P03	–	–	UPMUX	*1	–	–	–	–	✓	✓
P04	P04	–	–	UPMUX	*1	–	–	–	–	✓	✓
P05	P05	–	–	UPMUX	*1	–	–	–	–	✓	✓
P06	P06	–	–	UPMUX	*1	–	–	–	–	✓	✓
P07	P07	–	–	UPMUX	*1	–	–	–	–	✓	✓

*1: Refer to the “Universal Port Multiplexer” chapter.

6.7.2 P1 Port Group

The P1 port group supports the GPIO and interrupt functions.

Table 6.7.2.1 Control Registers for P1 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
P1DAT (P1 Port Data Register)	15	P1OUT7	0	H0	R/W	–	✓	✓
	14	P1OUT6	0	H0	R/W		✓	✓
	13	P1OUT5	0	H0	R/W		✓	✓
	12	P1OUT4	0	H0	R/W		✓	✓
	11	P1OUT3	0	H0	R/W		✓	✓
	10	P1OUT2	0	H0	R/W		✓	✓
	9	P1OUT1	0	H0	R/W		✓	✓
	8	P1OUT0	0	H0	R/W		✓	✓
	7	P1IN7	0	H0	R	–	✓	✓
	6	P1IN6	0	H0	R		✓	✓
	5	P1IN5	0	H0	R		✓	✓
	4	P1IN4	0	H0	R		✓	✓
	3	P1IN3	0	H0	R		✓	✓
	2	P1IN2	0	H0	R		✓	✓
	1	P1IN1	0	H0	R		✓	✓
	0	P1IN0	0	H0	R		✓	✓

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
P1IOEN (P1 Port Enable Register)	15	P1IEN7	0	H0	R/W	–	✓	✓
	14	P1IEN6	0	H0	R/W		✓	✓
	13	P1IEN5	0	H0	R/W		✓	✓
	12	P1IEN4	0	H0	R/W		✓	✓
	11	P1IEN3	0	H0	R/W		✓	✓
	10	P1IEN2	0	H0	R/W		✓	✓
	9	P1IEN1	0	H0	R/W		✓	✓
	8	P1IEN0	0	H0	R/W		✓	✓
	7	P1OEN7	0	H0	R/W	–	✓	✓
	6	P1OEN6	0	H0	R/W		✓	✓
	5	P1OEN5	0	H0	R/W		✓	✓
	4	P1OEN4	0	H0	R/W		✓	✓
	3	P1OEN3	0	H0	R/W		✓	✓
	2	P1OEN2	0	H0	R/W		✓	✓
	1	P1OEN1	0	H0	R/W		✓	✓
	0	P1OEN0	0	H0	R/W		✓	✓
P1RCTL (P1 Port Pull-up/down Control Register)	15	P1PDPU7	0	H0	R/W	–	✓	✓
	14	P1PDPU6	0	H0	R/W		✓	✓
	13	P1PDPU5	0	H0	R/W		✓	✓
	12	P1PDPU4	0	H0	R/W		✓	✓
	11	P1PDPU3	0	H0	R/W		✓	✓
	10	P1PDPU2	0	H0	R/W		✓	✓
	9	P1PDPU1	0	H0	R/W		✓	✓
	8	P1PDPU0	0	H0	R/W		✓	✓
	7	P1REN7	0	H0	R/W	–	✓	✓
	6	P1REN6	0	H0	R/W		✓	✓
	5	P1REN5	0	H0	R/W		✓	✓
	4	P1REN4	0	H0	R/W		✓	✓
	3	P1REN3	0	H0	R/W		✓	✓
	2	P1REN2	0	H0	R/W		✓	✓
	1	P1REN1	0	H0	R/W		✓	✓
	0	P1REN0	0	H0	R/W		✓	✓
P1INTF (P1 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–
	7	P1IF7	0	H0	R/W	Cleared by writing 1.	✓	✓
	6	P1IF6	0	H0	R/W		✓	✓
	5	P1IF5	0	H0	R/W		✓	✓
	4	P1IF4	0	H0	R/W		✓	✓
	3	P1IF3	0	H0	R/W		✓	✓
	2	P1IF2	0	H0	R/W		✓	✓
	1	P1IF1	0	H0	R/W		✓	✓
	0	P1IF0	0	H0	R/W		✓	✓
P1INTCTL (P1 Port Interrupt Control Register)	15	P1EDGE7	0	H0	R/W	–	✓	✓
	14	P1EDGE6	0	H0	R/W		✓	✓
	13	P1EDGE5	0	H0	R/W		✓	✓
	12	P1EDGE4	0	H0	R/W		✓	✓
	11	P1EDGE3	0	H0	R/W		✓	✓
	10	P1EDGE2	0	H0	R/W		✓	✓
	9	P1EDGE1	0	H0	R/W		✓	✓
	8	P1EDGE0	0	H0	R/W		✓	✓
	7	P1IE7	0	H0	R/W	–	✓	✓
	6	P1IE6	0	H0	R/W		✓	✓
	5	P1IE5	0	H0	R/W		✓	✓
	4	P1IE4	0	H0	R/W		✓	✓
	3	P1IE3	0	H0	R/W		✓	✓
	2	P1IE2	0	H0	R/W		✓	✓
	1	P1IE1	0	H0	R/W		✓	✓
	0	P1IE0	0	H0	R/W		✓	✓

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
P1CHATEN	15–8	–	0x00	–	R	–	–	–
(P1 Port Chattering	7	P1CHATEN7	0	H0	R/W	–	✓	✓
Filter Enable Register)	6	P1CHATEN6	0	H0	R/W		✓	✓
	5	P1CHATEN5	0	H0	R/W		✓	✓
	4	P1CHATEN4	0	H0	R/W		✓	✓
	3	P1CHATEN3	0	H0	R/W		✓	✓
	2	P1CHATEN2	0	H0	R/W		✓	✓
	1	P1CHATEN1	0	H0	R/W		✓	✓
	0	P1CHATEN0	0	H0	R/W		✓	✓
P1MODSEL	15–8	–	0x00	–	R	–	–	–
(P1 Port Mode Select	7	P1SEL7	0	H0	R/W	–	✓	✓
Register)	6	P1SEL6	0	H0	R/W		✓	✓
	5	P1SEL5	0	H0	R/W		✓	✓
	4	P1SEL4	0	H0	R/W		✓	✓
	3	P1SEL3	0	H0	R/W		✓	✓
	2	P1SEL2	0	H0	R/W		✓	✓
	1	P1SEL1	0	H0	R/W		✓	✓
	0	P1SEL0	0	H0	R/W		✓	✓
P1FNCSEL	15–14	P17MUX[1:0]	0x0	H0	R/W	–	✓	✓
(P1 Port Function	13–12	P16MUX[1:0]	0x0	H0	R/W		✓	✓
Select Register)	11–10	P15MUX[1:0]	0x0	H0	R/W		✓	✓
	9–8	P14MUX[1:0]	0x0	H0	R/W		✓	✓
	7–6	P13MUX[1:0]	0x0	H0	R/W		✓	✓
	5–4	P12MUX[1:0]	0x0	H0	R/W		✓	✓
	3–2	P11MUX[1:0]	0x0	H0	R/W		✓	✓
	1–0	P10MUX[1:0]	0x0	H0	R/W		✓	✓

Table 6.7.2.2 P1 Port Group Function Assignment

Port name	P1SEL*y* = 0	P1SEL*y* = 1								M02	M03
	GPIO	P1*y*MUX = 0x0 (Function 0)		P1*y*MUX = 0x1 (Function 1)		P1*y*MUX = 0x2 (Function 2)		P1*y*MUX = 0x3 (Function 3)			
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin		
P10	P10	–	–	UPMUX	*1	–	–	LCD4B	SEG15	✓	✓
P11	P11	–	–	UPMUX	*1	–	–	LCD4B	SEG14	✓	✓
P12	P12	–	–	UPMUX	*1	–	–	LCD4B	SEG13	✓	✓
P13	P13	–	–	UPMUX	*1	–	–	LCD4B	SEG12	✓	✓
P14	P14	–	–	UPMUX	*1	–	–	LCD4B	SEG11	✓	✓
P15	P15	–	–	UPMUX	*1	–	–	LCD4B	SEG10	✓	✓
P16	P16	–	–	UPMUX	*1	–	–	LCD4B	SEG9	✓	✓
P17	P17	–	–	UPMUX	*1	–	–	LCD4B	SEG8	✓	✓

*1: Refer to the “Universal Port Multiplexer” chapter.

6.7.3 P2 Port Group

The P2 port group support the GPIO and interrupt functions.

Table 6.7.3.1 Control Registers for P2 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
P2DAT (P2 Port Data Register)	15	P2OUT7	0	H0	R/W	–	–	✓
	14	P2OUT6	0	H0	R/W		–	✓
	13	P2OUT5	0	H0	R/W		–	✓
	12	P2OUT4	0	H0	R/W		–	✓
	11	P2OUT3	0	H0	R/W		–	✓
	10	P2OUT2	0	H0	R/W		–	✓
	9	P2OUT1	0	H0	R/W		–	✓
	8	P2OUT0	0	H0	R/W		–	✓
	7	P2IN7	0	H0	R	–	–	✓
	6	P2IN6	0	H0	R		–	✓
	5	P2IN5	0	H0	R		–	✓
	4	P2IN4	0	H0	R		–	✓
	3	P2IN3	0	H0	R		–	✓
	2	P2IN2	0	H0	R		–	✓
	1	P2IN1	0	H0	R		–	✓
	0	P2IN0	0	H0	R		–	✓
P2IOEN (P2 Port Enable Register)	15	P2IEN7	0	H0	R/W	–	–	✓
	14	P2IEN6	0	H0	R/W		–	✓
	13	P2IEN5	0	H0	R/W		–	✓
	12	P2IEN4	0	H0	R/W		–	✓
	11	P2IEN3	0	H0	R/W		–	✓
	10	P2IEN2	0	H0	R/W		–	✓
	9	P2IEN1	0	H0	R/W		–	✓
	8	P2IEN0	0	H0	R/W		–	✓
	7	P2OEN7	0	H0	R/W	–	–	✓
	6	P2OEN6	0	H0	R/W		–	✓
	5	P2OEN5	0	H0	R/W		–	✓
	4	P2OEN4	0	H0	R/W		–	✓
	3	P2OEN3	0	H0	R/W		–	✓
	2	P2OEN2	0	H0	R/W		–	✓
	1	P2OEN1	0	H0	R/W		–	✓
	0	P2OEN0	0	H0	R/W		–	✓
P2RCTL (P2 Port Pull-up/down Control Register)	15	P2PDPU7	0	H0	R/W	–	–	✓
	14	P2PDPU6	0	H0	R/W		–	✓
	13	P2PDPU5	0	H0	R/W		–	✓
	12	P2PDPU4	0	H0	R/W		–	✓
	11	P2PDPU3	0	H0	R/W		–	✓
	10	P2PDPU2	0	H0	R/W		–	✓
	9	P2PDPU1	0	H0	R/W		–	✓
	8	P2PDPU0	0	H0	R/W		–	✓
	7	P2REN7	0	H0	R/W	–	–	✓
	6	P2REN6	0	H0	R/W		–	✓
	5	P2REN5	0	H0	R/W		–	✓
	4	P2REN4	0	H0	R/W		–	✓
	3	P2REN3	0	H0	R/W		–	✓
	2	P2REN2	0	H0	R/W		–	✓
	1	P2REN1	0	H0	R/W		–	✓
	0	P2REN0	0	H0	R/W		–	✓

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
P2INTF (P2 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–
	7	P2IF7	0	H0	R/W	Cleared by writing 1.	–	✓
	6	P2IF6	0	H0	R/W		–	✓
	5	P2IF5	0	H0	R/W		–	✓
	4	P2IF4	0	H0	R/W		–	✓
	3	P2IF3	0	H0	R/W		–	✓
	2	P2IF2	0	H0	R/W		–	✓
	1	P2IF1	0	H0	R/W		–	✓
	0	P2IF0	0	H0	R/W		–	✓
P2INTCTL (P2 Port Interrupt Control Register)	15	P2EDGE7	0	H0	R/W	–	–	✓
	14	P2EDGE6	0	H0	R/W		–	✓
	13	P2EDGE5	0	H0	R/W		–	✓
	12	P2EDGE4	0	H0	R/W		–	✓
	11	P2EDGE3	0	H0	R/W		–	✓
	10	P2EDGE2	0	H0	R/W		–	✓
	9	P2EDGE1	0	H0	R/W		–	✓
	8	P2EDGE0	0	H0	R/W		–	✓
	7	P2IE7	0	H0	R/W	–	–	✓
	6	P2IE6	0	H0	R/W		–	✓
	5	P2IE5	0	H0	R/W		–	✓
	4	P2IE4	0	H0	R/W		–	✓
	3	P2IE3	0	H0	R/W		–	✓
	2	P2IE2	0	H0	R/W		–	✓
	1	P2IE1	0	H0	R/W		–	✓
	0	P2IE0	0	H0	R/W		–	✓
P2CHATEN (P2 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–	–	–
	7	P2CHATEN7	0	H0	R/W	–	–	✓
	6	P2CHATEN6	0	H0	R/W		–	✓
	5	P2CHATEN5	0	H0	R/W		–	✓
	4	P2CHATEN4	0	H0	R/W		–	✓
	3	P2CHATEN3	0	H0	R/W		–	✓
	2	P2CHATEN2	0	H0	R/W		–	✓
	1	P2CHATEN1	0	H0	R/W		–	✓
	0	P2CHATEN0	0	H0	R/W		–	✓
P2MODSEL (P2 Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–
	7	P2SEL7	0	H0	R/W	–	–	✓
	6	P2SEL6	0	H0	R/W		–	✓
	5	P2SEL5	0	H0	R/W		–	✓
	4	P2SEL4	0	H0	R/W		–	✓
	3	P2SEL3	0	H0	R/W		–	✓
	2	P2SEL2	0	H0	R/W		–	✓
	1	P2SEL1	0	H0	R/W		–	✓
	0	P2SEL0	0	H0	R/W		–	✓
P2FNCSEL (P2 Port Function Select Register)	15–14	P27MUX[1:0]	0x0	H0	R/W	–	–	✓
	13–12	P26MUX[1:0]	0x0	H0	R/W		–	✓
	11–10	P25MUX[1:0]	0x0	H0	R/W		–	✓
	9–8	P24MUX[1:0]	0x0	H0	R/W		–	✓
	7–6	P23MUX[1:0]	0x0	H0	R/W		–	✓
	5–4	P22MUX[1:0]	0x0	H0	R/W		–	✓
	3–2	P21MUX[1:0]	0x0	H0	R/W		–	✓
	1–0	P20MUX[1:0]	0x0	H0	R/W		–	✓

Table 6.7.3.2 P2 Port Group Function Assignment

Port name	P2SELy = 0	P2SELy = 1								M02	M03
	GPIO	P2yMUX = 0x0 (Function 0)		P2yMUX = 0x1 (Function 1)		P2yMUX = 0x2 (Function 2)		P2yMUX = 0x3 (Function 3)			
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin		
P20	P20	–	–	UPMUX	*1	–	–	LCD4B	SEG31	–	✓
P21	P21	–	–	UPMUX	*1	–	–	LCD4B	SEG30	–	✓
P22	P22	–	–	UPMUX	*1	–	–	LCD4B	SEG29	–	✓
P23	P23	–	–	UPMUX	*1	–	–	LCD4B	SEG28	–	✓
P24	P24	–	–	UPMUX	*1	–	–	LCD4B	SEG27	–	✓
P25	P25	–	–	UPMUX	*1	–	–	LCD4B	SEG26	–	✓
P26	P26	–	–	UPMUX	*1	–	–	LCD4B	SEG25	–	✓
P27	P27	–	–	UPMUX	*1	–	–	LCD4B	SEG24	–	✓

*1: Refer to the “Universal Port Multiplexer” chapter.

6.7.4 P3 Port Group

The P3 port group supports the GPIO and interrupt functions.

Table 6.7.4.1 Control Registers for P3 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
P3DAT (P3 Port Data Register)	15	P3OUT7	0	H0	R/W	–	–	✓
	14	P3OUT6	0	H0	R/W		–	✓
	13	P3OUT5	0	H0	R/W		–	✓
	12	P3OUT4	0	H0	R/W		–	✓
	11	P3OUT3	0	H0	R/W		–	✓
	10	P3OUT2	0	H0	R/W		–	✓
	9	P3OUT1	0	H0	R/W		–	✓
	8	P3OUT0	0	H0	R/W		–	✓
	7	P3IN7	0	H0	R	–	–	✓
	6	P3IN6	0	H0	R		–	✓
	5	P3IN5	0	H0	R		–	✓
	4	P3IN4	0	H0	R		–	✓
	3	P3IN3	0	H0	R		–	✓
	2	P3IN2	0	H0	R		–	✓
	1	P3IN1	0	H0	R		–	✓
	0	P3IN0	0	H0	R		–	✓
P3IOEN (P3 Port Enable Register)	15	P3IEN7	0	H0	R/W	–	–	✓
	14	P3IEN6	0	H0	R/W		–	✓
	13	P3IEN5	0	H0	R/W		–	✓
	12	P3IEN4	0	H0	R/W		–	✓
	11	P3IEN3	0	H0	R/W		–	✓
	10	P3IEN2	0	H0	R/W		–	✓
	9	P3IEN1	0	H0	R/W		–	✓
	8	P3IEN0	0	H0	R/W		–	✓
	7	P3OEN7	0	H0	R/W	–	–	✓
	6	P3OEN6	0	H0	R/W		–	✓
	5	P3OEN5	0	H0	R/W		–	✓
	4	P3OEN4	0	H0	R/W		–	✓
	3	P3OEN3	0	H0	R/W		–	✓
	2	P3OEN2	0	H0	R/W		–	✓
	1	P3OEN1	0	H0	R/W		–	✓
	0	P3OEN0	0	H0	R/W		–	✓

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
P3RCTL (P3 Port Pull-up/down Control Register)	15	P3PDPU7	0	H0	R/W	–	–	✓
	14	P3PDPU6	0	H0	R/W		–	✓
	13	P3PDPU5	0	H0	R/W		–	✓
	12	P3PDPU4	0	H0	R/W		–	✓
	11	P3PDPU3	0	H0	R/W		–	✓
	10	P3PDPU2	0	H0	R/W		–	✓
	9	P3PDPU1	0	H0	R/W		–	✓
	8	P3PDPU0	0	H0	R/W		–	✓
	7	P3REN7	0	H0	R/W	–	–	✓
	6	P3REN6	0	H0	R/W		–	✓
	5	P3REN5	0	H0	R/W		–	✓
	4	P3REN4	0	H0	R/W		–	✓
	3	P3REN3	0	H0	R/W		–	✓
	2	P3REN2	0	H0	R/W		–	✓
	1	P3REN1	0	H0	R/W		–	✓
	0	P3REN0	0	H0	R/W		–	✓
P3INTF (P3 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–
	7	P3IF7	0	H0	R/W	Cleared by writing 1.	–	✓
	6	P3IF6	0	H0	R/W		–	✓
	5	P3IF5	0	H0	R/W		–	✓
	4	P3IF4	0	H0	R/W		–	✓
	3	P3IF3	0	H0	R/W		–	✓
	2	P3IF2	0	H0	R/W		–	✓
	1	P3IF1	0	H0	R/W		–	✓
	0	P3IF0	0	H0	R/W		–	✓
P3INTCTL (P3 Port Interrupt Control Register)	15	P3EDGE7	0	H0	R/W	–	–	✓
	14	P3EDGE6	0	H0	R/W		–	✓
	13	P3EDGE5	0	H0	R/W		–	✓
	12	P3EDGE4	0	H0	R/W		–	✓
	11	P3EDGE3	0	H0	R/W		–	✓
	10	P3EDGE2	0	H0	R/W		–	✓
	9	P3EDGE1	0	H0	R/W		–	✓
	8	P3EDGE0	0	H0	R/W		–	✓
	7	P3IE7	0	H0	R/W	–	–	✓
	6	P3IE6	0	H0	R/W		–	✓
	5	P3IE5	0	H0	R/W		–	✓
	4	P3IE4	0	H0	R/W		–	✓
	3	P3IE3	0	H0	R/W		–	✓
	2	P3IE2	0	H0	R/W		–	✓
	1	P3IE1	0	H0	R/W		–	✓
	0	P3IE0	0	H0	R/W		–	✓
P3CHATEN (P3 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–	–	–
	7	P3CHATEN7	0	H0	R/W	–	–	✓
	6	P3CHATEN6	0	H0	R/W		–	✓
	5	P3CHATEN5	0	H0	R/W		–	✓
	4	P3CHATEN4	0	H0	R/W		–	✓
	3	P3CHATEN3	0	H0	R/W		–	✓
	2	P3CHATEN2	0	H0	R/W		–	✓
	1	P3CHATEN1	0	H0	R/W		–	✓
	0	P3CHATEN0	0	H0	R/W		–	✓

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
P3MODSEL (P3 Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–
	7	P3SEL7	0	H0	R/W	–	–	✓
	6	P3SEL6	0	H0	R/W		–	✓
	5	P3SEL5	0	H0	R/W		–	✓
	4	P3SEL4	0	H0	R/W		–	✓
	3	P3SEL3	0	H0	R/W		–	✓
	2	P3SEL2	0	H0	R/W		–	✓
	1	P3SEL1	0	H0	R/W		–	✓
	0	P3SEL0	0	H0	R/W		–	✓
P3FNCSEL (P3 Port Function Select Register)	15–14	P37MUX[1:0]	0x0	H0	R/W	–	–	✓
	13–12	P36MUX[1:0]	0x0	H0	R/W		–	✓
	11–10	P35MUX[1:0]	0x0	H0	R/W		–	✓
	9–8	P34MUX[1:0]	0x0	H0	R/W		–	✓
	7–6	P33MUX[1:0]	0x0	H0	R/W		–	✓
	5–4	P32MUX[1:0]	0x0	H0	R/W		–	✓
	3–2	P31MUX[1:0]	0x0	H0	R/W		–	✓
	1–0	P30MUX[1:0]	0x0	H0	R/W		–	✓

Table 6.7.4.2 P3 Port Group Function Assignment

Port name	P3SEL*y* = 0	P3SEL*y* = 1								M02	M03
	GPIO	P3*y*MUX = 0x0 (Function 0)		P3*y*MUX = 0x1 (Function 1)		P3*y*MUX = 0x2 (Function 2)		P3*y*MUX = 0x3 (Function 3)			
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin		
P30	P30	–	–	UPMUX	*1	–	–	LCD4B	SEG23	–	✓
P31	P31	–	–	UPMUX	*1	–	–	LCD4B	SEG22	–	✓
P32	P32	–	–	UPMUX	*1	–	–	LCD4B	SEG21	–	✓
P33	P33	–	–	UPMUX	*1	–	–	LCD4B	SEG20	–	✓
P34	P34	–	–	UPMUX	*1	–	–	LCD4B	SEG19	–	✓
P35	P35	–	–	UPMUX	*1	–	–	LCD4B	SEG18	–	✓
P36	P36	–	–	UPMUX	*1	–	–	LCD4B	SEG17	–	✓
P37	P37	–	–	UPMUX	*1	–	–	LCD4B	SEG16	–	✓

*1: Refer to the “Universal Port Multiplexer” chapter.

6.7.5 P4 Port Group

The P4 port group supports the GPIO and interrupt functions.

Table 6.7.5.1 Control Registers for P4 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
P4DAT (P4 Port Data Register)	15	P4OUT7	0	H0	R/W	–	–	✓
	14	P4OUT6	0	H0	R/W		–	✓
	13	P4OUT5	0	H0	R/W		–	✓
	12	P4OUT4	0	H0	R/W		–	✓
	11	P4OUT3	0	H0	R/W		–	✓
	10	P4OUT2	0	H0	R/W		–	✓
	9	P4OUT1	0	H0	R/W		–	✓
	8	P4OUT0	0	H0	R/W		–	✓
	7	P4IN7	0	H0	R	–	–	✓
	6	P4IN6	0	H0	R		–	✓
	5	P4IN5	0	H0	R		–	✓
	4	P4IN4	0	H0	R		–	✓
	3	P4IN3	0	H0	R		–	✓
	2	P4IN2	0	H0	R		–	✓
	1	P4IN1	0	H0	R		–	✓
	0	P4IN0	0	H0	R		–	✓

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
P4IOEN (P4 Port Enable Register)	15	P4IEN7	0	H0	R/W	–	–	✓
	14	P4IEN6	0	H0	R/W		–	✓
	13	P4IEN5	0	H0	R/W		–	✓
	12	P4IEN4	0	H0	R/W		–	✓
	11	P4IEN3	0	H0	R/W		–	✓
	10	P4IEN2	0	H0	R/W		–	✓
	9	P4IEN1	0	H0	R/W		–	✓
	8	P4IEN0	0	H0	R/W		–	✓
	7	P4OEN7	0	H0	R/W	–	–	✓
	6	P4OEN6	0	H0	R/W		–	✓
	5	P4OEN5	0	H0	R/W		–	✓
	4	P4OEN4	0	H0	R/W		–	✓
	3	P4OEN3	0	H0	R/W		–	✓
	2	P4OEN2	0	H0	R/W		–	✓
	1	P4OEN1	0	H0	R/W		–	✓
	0	P4OEN0	0	H0	R/W		–	✓
P4RCTL (P4 Port Pull-up/down Control Register)	15	P4PDPU7	0	H0	R/W	–	–	✓
	14	P4PDPU6	0	H0	R/W		–	✓
	13	P4PDPU5	0	H0	R/W		–	✓
	12	P4PDPU4	0	H0	R/W		–	✓
	11	P4PDPU3	0	H0	R/W		–	✓
	10	P4PDPU2	0	H0	R/W		–	✓
	9	P4PDPU1	0	H0	R/W		–	✓
	8	P4PDPU0	0	H0	R/W		–	✓
	7	P4REN7	0	H0	R/W	–	–	✓
	6	P4REN6	0	H0	R/W		–	✓
	5	P4REN5	0	H0	R/W		–	✓
	4	P4REN4	0	H0	R/W		–	✓
	3	P4REN3	0	H0	R/W		–	✓
	2	P4REN2	0	H0	R/W		–	✓
	1	P4REN1	0	H0	R/W		–	✓
	0	P4REN0	0	H0	R/W		–	✓
P4INTF (P4 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–
	7	P4IF7	0	H0	R/W	Cleared by writing 1.	–	✓
	6	P4IF6	0	H0	R/W		–	✓
	5	P4IF5	0	H0	R/W		–	✓
	4	P4IF4	0	H0	R/W		–	✓
	3	P4IF3	0	H0	R/W		–	✓
	2	P4IF2	0	H0	R/W		–	✓
	1	P4IF1	0	H0	R/W		–	✓
	0	P4IF0	0	H0	R/W		–	✓
P4INTCTL (P4 Port Interrupt Control Register)	15	P4EDGE7	0	H0	R/W	–	–	✓
	14	P4EDGE6	0	H0	R/W		–	✓
	13	P4EDGE5	0	H0	R/W		–	✓
	12	P4EDGE4	0	H0	R/W		–	✓
	11	P4EDGE3	0	H0	R/W		–	✓
	10	P4EDGE2	0	H0	R/W		–	✓
	9	P4EDGE1	0	H0	R/W		–	✓
	8	P4EDGE0	0	H0	R/W		–	✓
	7	P4IE7	0	H0	R/W	–	–	✓
	6	P4IE6	0	H0	R/W		–	✓
	5	P4IE5	0	H0	R/W		–	✓
	4	P4IE4	0	H0	R/W		–	✓
	3	P4IE3	0	H0	R/W		–	✓
	2	P4IE2	0	H0	R/W		–	✓
	1	P4IE1	0	H0	R/W		–	✓
	0	P4IE0	0	H0	R/W		–	✓

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
P4CHATEN (P4 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–	–	–
	7	P4CHATEN7	0	H0	R/W	–	–	✓
	6	P4CHATEN6	0	H0	R/W		–	✓
	5	P4CHATEN5	0	H0	R/W		–	✓
	4	P4CHATEN4	0	H0	R/W		–	✓
	3	P4CHATEN3	0	H0	R/W		–	✓
	2	P4CHATEN2	0	H0	R/W		–	✓
	1	P4CHATEN1	0	H0	R/W		–	✓
	0	P4CHATEN0	0	H0	R/W		–	✓
P4MODSEL (P4 Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–
	7	P4SEL7	0	H0	R/W	–	–	✓
	6	P4SEL6	0	H0	R/W		–	✓
	5	P4SEL5	0	H0	R/W		–	✓
	4	P4SEL4	0	H0	R/W		–	✓
	3	P4SEL3	0	H0	R/W		–	✓
	2	P4SEL2	0	H0	R/W		–	✓
	1	P4SEL1	0	H0	R/W		–	✓
	0	P4SEL0	0	H0	R/W		–	✓
P4FNCSEL (P4 Port Function Select Register)	15–14	P47MUX[1:0]	0x0	H0	R/W	–	–	✓
	13–12	P46MUX[1:0]	0x0	H0	R/W		–	✓
	11–10	P45MUX[1:0]	0x0	H0	R/W		–	✓
	9–8	P44MUX[1:0]	0x0	H0	R/W		–	✓
	7–6	P43MUX[1:0]	0x0	H0	R/W		–	✓
	5–4	P42MUX[1:0]	0x0	H0	R/W		–	✓
	3–2	P41MUX[1:0]	0x0	H0	R/W		–	✓
	1–0	P40MUX[1:0]	0x0	H0	R/W		–	✓

Table 6.7.5.2 P4 Port Group Function Assignment

Port name	P4SEL*y* = 0	P4SEL*y* = 1								M02	M03
	GPIO	P4*y*MUX = 0x0 (Function 0)		P4*y*MUX = 0x1 (Function 1)		P4*y*MUX = 0x2 (Function 2)		P4*y*MUX = 0x3 (Function 3)			
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin		
P40	P40	–	–	–	–	–	–	–	–	–	✓
P41	P41	–	–	–	–	–	–	–	–	–	✓
P42	P42	–	–	–	–	–	–	–	–	–	✓
P43	P43	–	–	–	–	–	–	–	–	–	✓
P44	P44	–	–	–	–	–	–	–	–	–	✓
P45	P45	–	–	–	–	–	–	–	–	–	✓
P46	P46	–	–	–	–	–	–	–	–	–	✓
P47	P47	–	–	–	–	–	–	–	–	–	✓

6.7.6 Pd Port Group

The Pd0–Pd2 ports are configured as a debugging function port at initialization. The Pd port group supports the GPIO functions. The GPIO function of the Pd2 port supports output only, therefore, the pull-up/down function cannot be used.

Table 6.7.6.1 Control Registers for Pd Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
PDDAT (Pd Port Data Register)	15–11	–	0x00	–	R	–	–	–
	10	PDOUT2	0	H0	R/W	–	✓	✓
	9	PDOUT1	0	H0	R/W		✓	✓
	8	PDOUT0	0	H0	R/W		✓	✓
	7–3	–	0x00	–	R	–	–	–
	2	–	0	–	R	–	–	–
	1	PDIN1	X	H0	R		✓	✓
	0	PDIN0	X	H0	R		✓	✓

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
PDIOEN (Pd Port Enable Register)	15–11	–	0x00	–	R	–	–	–
	10	(reserved)	0	H0	R/W	–	✓	✓
	9	PDIEN1	0	H0	R/W		✓	✓
	8	PDIEN0	0	H0	R/W		✓	✓
	7–3	–	0x00	–	R	–	–	–
	2	PDOEN2	0	H0	R/W	–	✓	✓
	1	PDOEN1	0	H0	R/W		✓	✓
	0	PDOEN0	0	H0	R/W		✓	✓
PDRCTL (Pd Port Pull-up/down Control Register)	15–11	–	0x00	–	R	–	–	–
	10	(reserved)	0	H0	R/W	–	✓	✓
	9	PDPDPU1	0	H0	R/W		✓	✓
	8	PDPDPU0	0	H0	R/W		✓	✓
	7–5	–	0x00	–	R	–	–	–
	2	(reserved)	0	H0	R/W	–	✓	✓
	1	PDREN1	0	H0	R/W		✓	✓
	0	PDREN0	0	H0	R/W		✓	✓
PDINTF PDINTCTL PDCHATEN	15–0	–	0x0000	–	R	–	–	–
PDMODSEL (Pd Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–
	7–3	–	0	–	R	–	–	–
	2	PDSEL2	1	H0	R/W		✓	✓
	1	PDSEL1	1	H0	R/W		✓	✓
	0	PDSEL0	1	H0	R/W		✓	✓
PDFNCSEL (Pd Port Function Select Register)	15–8	–	0x00	–	R	–	–	–
	7–6	–	0x0	–	R			
	5–4	PD2MUX[1:0]	0x0	H0	R/W	–	✓	✓
	3–2	PD1MUX[1:0]	0x0	H0	R/W		✓	✓
	1–0	PD0MUX[1:0]	0x0	H0	R/W		✓	✓

Table 6.7.6.2 Pd Port Group Function Assignment

Port name	PDSEL*y* = 0	PDSEL*y* = 1								M02	M03
	GPIO	PD*y*MUX = 0x0 (Function 0)		PD*y*MUX = 0x1 (Function 1)		PD*y*MUX = 0x2 (Function 2)		PD*y*MUX = 0x3 (Function 3)			
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin		
Pd0	PD0	DBG	DST2	–	–	–	–	–	–	✓	✓
Pd1	PD1	DBG	DSIO	–	–	–	–	–	–	✓	✓
Pd2	PD2	DBG	DCLK	–	–	–	–	–	–	✓	✓

6.7.7 Common Registers between Port Groups

Table 6.7.7.1 Control Registers for Common Use with Port Groups

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
PCLK (P Port Clock Control Register)	15–9	–	0x00	–	R	–	–	–
	8	DBRUN	0	H0	R/WP	–	✓	✓
	7–4	CLKDIV[3:0]	0x0	H0	R/WP		✓	✓
	3–2	KRSTCFG[1:0]	0x0	H0	R/WP		✓	✓
	1–0	CLKSRC[1:0]	0x0	H0	R/WP		✓	✓
PINTFGRP (P Port Interrupt Flag Group Register)	15–8	–	0x00	–	R	–	–	–
	7–5	–	0x0	–	R			
	4	P4INT	0	H0	R	–	–	✓
	3	P3INT	0	H0	R		–	✓
	2	P2INT	0	H0	R		–	✓
	1	P1INT	0	H0	R		✓	✓
	0	P0INT	0	H0	R		✓	✓

7 Universal Port Multiplexer (UPMUX)

7.1 Overview

UPMUX is a multiplexer that allows software to assign the desired peripheral I/O function to an I/O port. The main features are outlined below.

- Allows programmable assignment of the synchronous serial interface, I²C, UART, clock generator, and sound generator peripheral I/O functions to the P0, P1, P2, and P3 port groups.
- The peripheral I/O function assigned via UPMUX is enabled by setting the P*x*FNCSEL.P*xy*MUX[1:0] bits to 0x1.

Note: '*x*', which is used in the port names P*xy*, register names, and bit names, refers to a port group (*x* = 0, 1, 2, 3) and '*y*' refers to a port number (*y* = 0, 1, 2, ··· , 7).

Figure 7.1.1 shows the configuration of UPMUX.

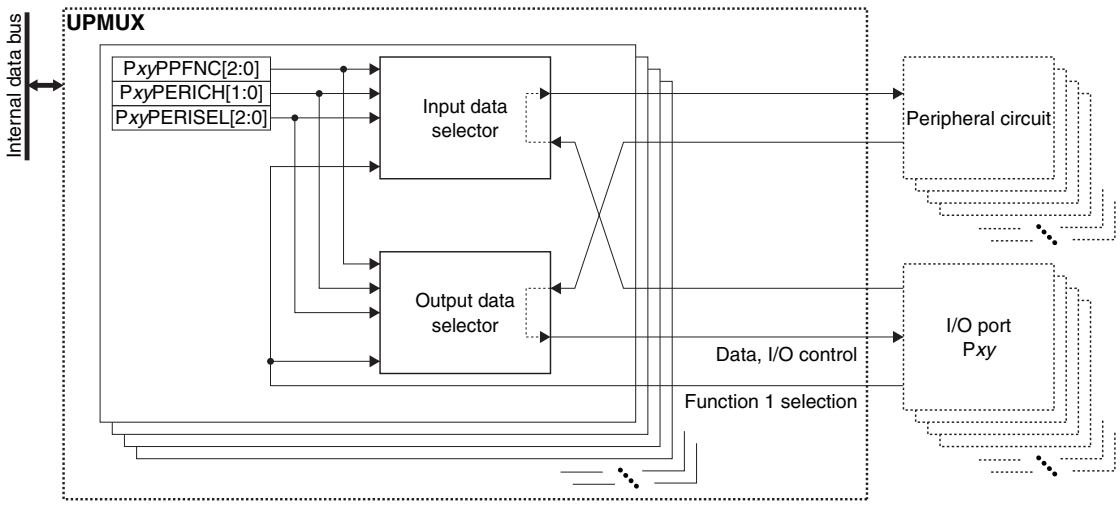


Figure 7.1.1 UPMUX Configuration

7.2 Peripheral Circuit I/O Function Assignment

An I/O function of a peripheral circuit supported may be assigned to peripheral I/O function 1 of an I/O port listed above. The following shows the procedure to assign a peripheral I/O function and enable it in the I/O port:

1. Configure the P*x*IOEN register of the I/O port.
   - Set the P*x*IOEN.P*x*IEN*y* bit to 0. (Disable input)
   - Set the P*x*IOEN.P*x*OEN*y* bit to 0. (Disable output)
2. Set the P*x*MODSEL.P*x*SEL*y* bit of the I/O port to 0. (Disable peripheral I/O function)
3. Set the following P*x*UPMUX*n* register bits (*n* = 0 to 3).
   - P*x*UPMUX*n*.P*xy*PERISEL[2:0] bits (Select peripheral circuit)
   - P*x*UPMUX*n*.P*xy*PERICH[1:0] bits (Select peripheral circuit channel)
   - P*x*UPMUX*n*.P*xy*PPFNC[2:0] bits (Select function to assign)
4. Initialize the peripheral circuit.
5. Set the P*x*FNCSEL.P*xy*MUX[1:0] bits of the I/O port to 0x1. (Select peripheral I/O function 1)
6. Set the P*x*MODSEL.P*x*SEL*y* bit of the I/O port to 1. (Enable peripheral I/O function)

7.3 Control Registers

P*xy–xz* Universal Port Multiplexer Setting Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
P*x*UPMUX*n*	15–13	P*xz*PPFNC[2:0]	0x0	H0	R/W	–
	12–11	P*xz*PERICH[1:0]	0x0	H0	R/W	
	10–8	P*xz*PERISEL[2:0]	0x0	H0	R/W	
	7–5	P*xy*PPFNC[2:0]	0x0	H0	R/W	
	4–3	P*xy*PERICH[1:0]	0x0	H0	R/W	
	2–0	P*xy*PERISEL[2:0]	0x0	H0	R/W	

∗1: '*x*' in the register name refers to a port group number and '*n*' refers to a register number (0–3).
∗2: '*x*' in the bit name refers to a port group number, '*y*' refers to an even port number (0, 2, 4, 6), and '*z*' refers to an odd port number (*z* = *y* + 1).

Bits 15–13 P*xz*PPFNC[2:0]
Bits 7–5 P*xy*PPFNC[2:0]

These bits specify the peripheral I/O function to be assigned to the port. (See Table 7.3.1.)

Bits 12–11 P*xz*PERICH[1:0]
Bits 4–3 P*xy*PERICH[1:0]

These bits specify a peripheral circuit channel number. (See Table 7.3.1.)

Bits 10–8 P*xz*PERISEL[2:0]
Bits 2–0 P*xy*PERISEL[2:0]

These bits specify a peripheral circuit. (See Table 7.3.1.)

Table 7.3.1 Peripheral I/O Function Selections

P*x*UPMUX*n*.P*xy*PPFNC[2:0] bits (Peripheral I/O function)	P*x*UPMUX*n*.P*xy*PERISEL[2:0] bits (Peripheral circuit)							
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
	None ∗	I2C	SPIA	UART3	Reserved	SNDA_DMM	CLG	Reserved
	P*x*UPMUX*n*.P*xy*PERICH[1:0] bits (Peripheral circuit channel)							
	–	0x0	0x0	0x0	–	0x0	0x0	–
	–	Ch.0	Ch.0	Ch.0	–	–	–	–
0x0	None ∗	None ∗	None ∗	None ∗	None ∗	None ∗	None ∗	None ∗
0x1	Reserved	SCL*n*	SDI*n*	USIN*n*	Reserved	BZOUT	FOUT	Reserved
0x2		SDA*n*	SDO*n*	USOUT*n*		#BZOUT	Reserved	
0x3		Reserved	SPICLK*n*	Reserved		Reserved		
0x4			#SPISS*n*					
0x5			Reserved					
0x6								
0x7								

∗ "None" means no assignment. Selecting this will put the P*xy* pin into Hi-Z status when peripheral I/O function 1 is selected and enabled in the I/O port.

Note: Do not assign a peripheral input function to two or more I/O ports. Although the I/O ports output the same waveforms when an output function is assigned to two or more I/O port, a skew occurs due to the internal delay.

8 Watchdog Timer (WDT2)

8.1 Overview

WDT2 restarts the system if a problem occurs, such as when the program cannot be executed normally.
The features of WDT2 are listed below.

- Includes a 10-bit up counter to count NMI/reset generation cycle.
- A counter clock source and clock division ratio are selectable.
- Can generate a reset or NMI in a cycle given via software.
- Can generate a reset at the next NMI generation cycle after an NMI is generated.

Figure 8.1.1 shows the configuration of WDT2.

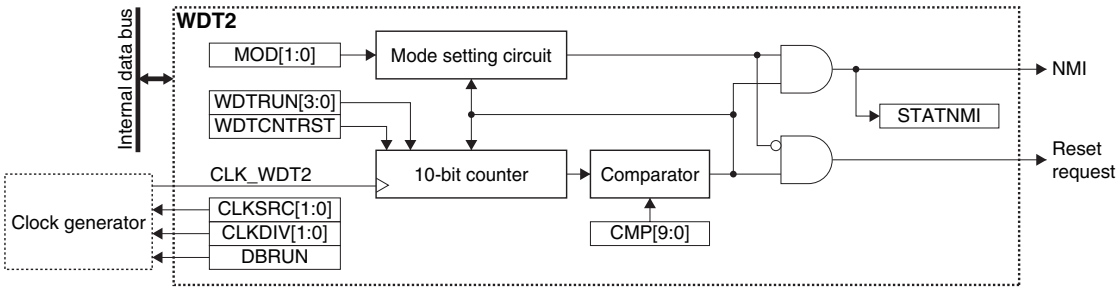


Figure 8.1.1 WDT2 Configuration

8.2 Clock Settings

8.2.1 WDT2 Operating Clock

When using WDT2, the WDT2 operating clock CLK_WDT2 must be supplied to WDT2 from the clock generator. The CLK_WDT2 supply should be controlled as in the procedure shown below.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Enable the clock source in the clock generator if it is stopped (refer to "Clock Generator" in the "Power Supply, Reset, and Clocks" chapter).
3. Set the following WDTCLK register bits:
   WDTCLK.CLKSRC[1:0] bits (Clock source selection)
   WDTCLK.CLKDIV[1:0] bits (Clock division ratio selection = Clock frequency setting)
4. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

8.2.2 Clock Supply in DEBUG Mode

The CLK_WDT2 supply during DEBUG mode should be controlled using the WDTCLK.DBRUN bit.
The CLK_WDT2 supply to WDT2 is suspended when the CPU enters DEBUG mode if the WDTCLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_WDT2 supply resumes. Although WDT2 stops operating when the CLK_WDT2 supply is suspended, the register retains the status before DEBUG mode was entered.
If the WDTCLK.DBRUN bit = 1, the CLK_WDT2 supply is not suspended and WDT2 will keep operating in DEBUG mode.

8.3 Operations

8.3.1 WDT2 Control

Activating WDT2

WDT2 should be initialized and started up with the procedure listed below.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Configure the WDT2 operating clock.
3. Set the WDTCTL.MOD[1:0] bits. (Select WDT2 operating mode)
4. Set the WDTCMP.CMP[9:0] bits. (Set NMI/reset generation cycle)
5. Write 1 to the WDTCTL.WDTCNTRST bit. (Reset WDT2 counter)
6. Write a value other than 0xa to the WDTCTL.WDTRUN[3:0] bits. (Start up WDT2)
7. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

NMI/reset generation cycle

Use the following equation to calculate the WDT2 NMI/reset generation cycle.

$$t_{WDT} = \frac{CMP + 1}{CLK\_WDT2} \qquad \text{(Eq. 8.1)}$$

Where

t_{WDT}: NMI/reset generation cycle [second]
CLK_WDT2: WDT2 operating clock frequency [Hz]
CMP: Setting value of the WDTCMP.CMP[9:0] bits

Example) t_{WDT} = 2.5 seconds when CLK_WDT2 = 256 Hz and the WDTCMP.CMP[9:0] bits = 639

Resetting WDT2 counter

To prevent an unexpected NMI/reset to be generated by WDT2, its embedded counter must be reset periodically via software while WDT2 is running.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Write 1 to the WDTCTL.WDTCNTRST bit. (Reset WDT2 counter)
3. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

A location should be provided for periodically processing this routine. Process this routine within the t_{WDT} cycle. After resetting, WDT2 starts counting with a new NMI/reset generation cycle.

Occurrence of counter compare match

If WDT2 is not reset within the t_{WDT} cycle for any reason and the counter reaches the setting value of the WDTCMP.CMP[9:0] bits, a compare match occurs to cause WDT2 to issue an NMI or reset according to the setting of the WDTCTL.MOD[1:0] bits.
If an NMI is issued, the WDTCTL.STATNMI bit is set to 1. This bit can be cleared to 0 by writing 1 to the WDTCTL.WDTCNTRST bit. Be sure to clear the WDTCTL.STATNMI bit in the NMI handler routine,
If a compare match occurs, the counter is automatically reset to 0 and it continues counting.

Deactivating WDT2

WDT2 should be stopped with the procedure listed below.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Write 0xa to the WDTCTL.WDTRUN[3:0] bits. (Stop WDT2)
3. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

8.3.2 Operations in HALT and SLEEP Modes

During HALT mode

WDT2 operates in HALT mode. HALT mode is therefore cleared by an NMI or reset if it continues for more than the NMI/reset generation cycle and the CPU executes the interrupt handler. To disable WDT2 in HALT mode, stop WDT2 by writing 0xa to the WDTCTL.WDTRUN[3:0] bits before executing the halt instruction. Reset WDT2 before resuming operations after HALT mode is cleared.

During SLEEP mode

WDT2 operates in SLEEP mode if the selected clock source is running. SLEEP mode is cleared by an NMI or reset if it continues for more than the NMI/reset generation cycle and the CPU executes the interrupt handler. Therefore, stop WDT2 by setting the WDTCTL.WDTRUN[3:0] bits before executing the slp instruction.
If the clock source stops in SLEEP mode, WDT2 stops. To prevent generation of an unnecessary NMI or reset after clearing SLEEP mode, reset WDT2 before executing the slp instruction. WDT2 should also be stopped as required using the WDTCTL.WDTRUN[3:0] bits.

8.4 Control Registers

WDT2 Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
WDTCLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/WP	
	7–6	–	0x0	–	R	
	5–4	CLKDIV[1:0]	0x0	H0	R/WP	
	3–2	–	0x0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/WP	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the WDT2 operating clock is supplied in DEBUG mode or not.
1 (R/WP): Clock supplied in DEBUG mode
0 (R/WP): No clock supplied in DEBUG mode

Bits 7–6 Reserved

Bits 5–4 CLKDIV[1:0]

These bits select the division ratio of the WDT2 operating clock (counter clock). The clock frequency should be set to around 256 Hz.

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of WDT2.

Table 8.4.1 Clock Source and Division Ratio Settings

WDTCLK. CLKDIV[1:0] bits	WDTCLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0x3	1/16,384	1/128	1/65,536	1/1
0x2	1/8,192		1/32,768	
0x1	1/4,096		1/16,384	
0x0	1/2,048		1/8,192	

(Note) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

WDT2 Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
WDTCTL	15–11	–	0x00	–	R	–
	10–9	MOD[1:0]	0x0	H0	R/WP	
	8	STATNMI	0	H0	R	
	7–5	–	0x0	–	R	
	4	WDTCNTRST	0	H0	WP	Always read as 0.
	3–0	WDTRUN[3:0]	0xa	H0	R/WP	–

Bits 15–11 Reserved

Bits 10–9 MOD[1:0]

These bits set the WDT2 operating mode.

Table 8.4.2 Operating Mode Setting

WDTCTL. MOD[1:0] bits	Operating mode	Description
0x3	Reserved	–
0x2	RESET after NMI mode	If the WDTCTL.STATNMI bit is not cleared to 0 after an NMI has occurred due to a counter compare match, WDT2 issues a reset when the next compare match occurs.
0x1	NMI mode	WDT2 issues an NMI when a counter compare match occurs.
0x0	RESET mode	WDT2 issues a reset when a counter compare match occurs.

Bit 8 STATNMI

This bit indicates that a counter compare match and NMI have occurred.

1 (R): NMI (counter compare match) occurred

0 (R): NMI not occurred

When the NMI generation function of WDT2 is used, read this bit in the NMI handler routine to confirm that WDT2 was the source of the NMI.

The WDTCTL.STATNMI bit set to 1 is cleared to 0 by writing 1 to the WDTCTL.WDTCNTRST bit.

Bits 7–5 Reserved

Bit 4 WDTCNTRST

This bit resets the 10-bit counter and the WDTCTL.STATNMI bit.

1 (WP): Reset

0 (WP): Ignored

0 (R): Always 0 when being read

Bits 3–0 WDTRUN[3:0]

These bits control WDT2 to run and stop.

0xa (WP): Stop

Values other than 0xa (WP): Run

0xa (R): Idle

0x0 (R): Running

Always 0x0 is read if a value other than 0xa is written.

Since an NMI or reset may be generated immediately after running depending on the counter value, WDT2 should also be reset concurrently when running WDT2.

WDT2 Counter Compare Match Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
WDTCMP	15–10	–	0x00	–	R	–
	9–0	CMP[9:0]	0x3ff	H0	R/WP	

Bits 15–10 Reserved

Bits 9–0 CMP[9:0]

These bits set the NMI/reset generation cycle.

The value set in this register is compared with the 10-bit counter value while WDT2 is running, and an NMI or reset is generated when they are matched.

9 Supply Voltage Detector (SVD4)

9.1 Overview

SVD4 is a supply voltage detector to monitor the power supply voltage on the VDD pin or the voltage applied to an external pin. The main features are listed below.

- Power supply voltage to be detected: Selectable from VDD and external power sources (EXSVD0 and EXSVD1) (Note: See the table below.)
- Detectable voltage level: Selectable from among 32 levels (max.) (Note: See the table below.)
- Detection results:
  - Can be read whether the power supply voltage is lower than the detection voltage level or not.
  - Can generate an interrupt or a reset when low power supply voltage is detected.
- Interrupt: 1 system (Low power supply voltage detection interrupt)
- Supports intermittent operations:
  - Three detection cycles are selectable.
  - Low power supply voltage detection count function to generate an interrupt/reset when low power supply voltage is successively detected the number of times specified.
  - Continuous operation is also possible.

Figure 9.1.1 shows the configuration of SVD4.

Table 9.1.1 SVD4 Configuration of S1C17M02/M03

Item	S1C17M02	S1C17M03
Power supply voltage to be detected	VDD and one externally input voltage (EXSVD0)	
Detectable voltage level	VDD: 19 levels (1.7 to 3.6 V)/external voltage: 19 levels (1.7 to 3.6 V)	

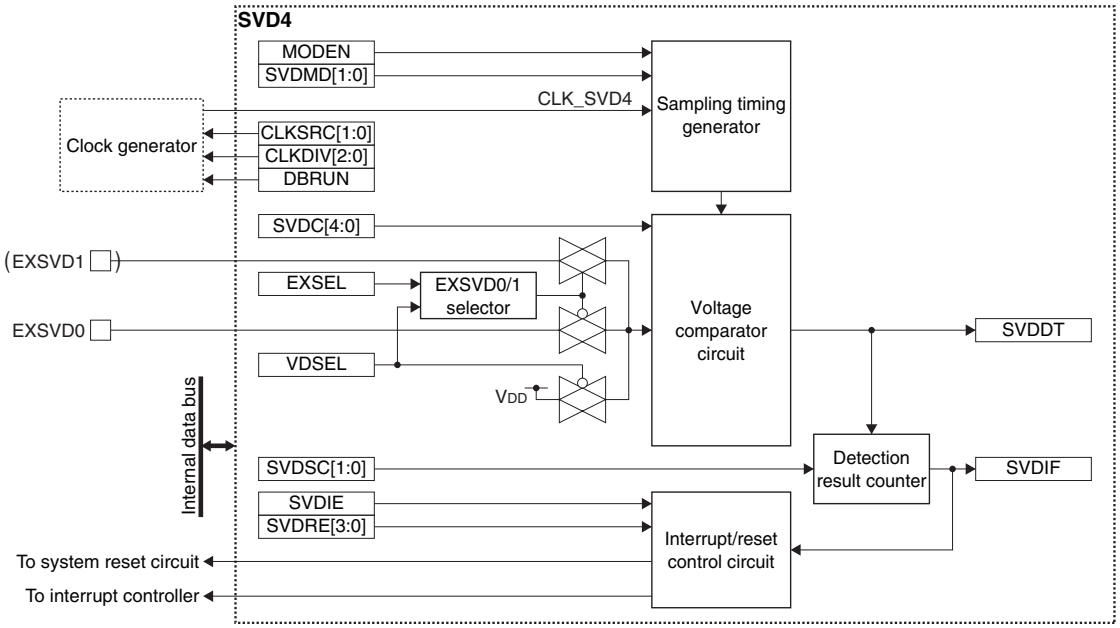


Figure 9.1.1 SVD4 Configuration

9.2 Input Pins and External Connection

9.2.1 Input Pins

Table 9.2.1.1 shows the SVD4 input pins.

Table 9.2.1.1 SVD4 Input Pins

Pin name	I/O*	Initial status*	Function
EXSVD0	A	A (Hi-Z)	External power supply voltage detection pin 0
EXSVD1	A	A (Hi-Z)	External power supply voltage detection pin 1

* Indicates the status when the pin is configured for SVD4.

If the port is shared with the EXSVD0/1 pin and other functions, the EXSVD0/1 function must be assigned to the port before SVD4 can be activated. For more information, refer to the "I/O Ports" chapter.

9.2.2 External Connection

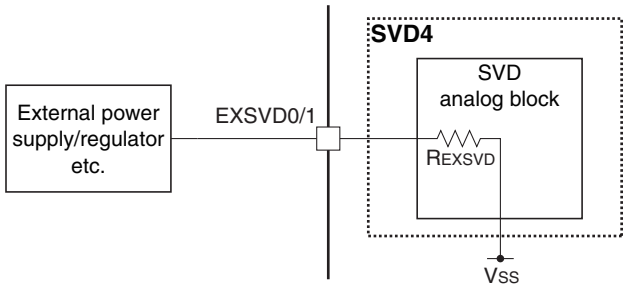


Figure 9.2.2.1 Connection between EXSVD0/1 Pin and External Power Supply

For the EXSVD0/1 pin input voltage range and the EXSVD input impedance, refer to "Supply Voltage Detector Characteristics" in the "Electrical Characteristics" chapter.

9.3 Clock Settings

9.3.1 SVD4 Operating Clock

When using SVD4, the SVD4 operating clock CLK_SVD4 must be supplied to SVD4 from the clock generator. The CLK_SVD4 supply should be controlled as in the procedure shown below.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Enable the clock source in the clock generator if it is stopped (refer to "Clock Generator" in the "Power Supply, Reset, and Clocks" chapter).
3. Set the following SVDCLK register bits:
   - SVDCLK.CLKSRC[1:0] bits (Clock source selection)
   - SVDCLK.CLKDIV[2:0] bits (Clock division ratio selection = Clock frequency setting)
4. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

The CLK_SVD4 frequency should be set to around 32 kHz.

9.3.2 Clock Supply in SLEEP Mode

When using SVD4 during SLEEP mode, the SVD4 operating clock CLK_SVD4 must be configured so that it will keep supplying by writing 0 to the CLGOSC.*xxxx*SLPC bit for the CLK_SVD4 clock source.
If the CLGOSC.*xxxx*SLPC bit for the CLK_SVD4 clock source is 1, the CLK_SVD4 clock source is deactivated during SLEEP mode and SVD4 stops with the register settings maintained at those before entering SLEEP mode. After the CPU returns to normal mode, CLK_SVD4 is supplied and the SVD4 operation resumes.

9.3.3 Clock Supply in DEBUG Mode

The CLK_SVD4 supply during DEBUG mode should be controlled using the SVDCLK.DBRUN bit.
The CLK_SVD4 supply to SVD4 is suspended when the CPU enters DEBUG mode if the SVDCLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_SVD4 supply resumes. Although SVD4 stops operating when the CLK_SVD4 supply is suspended, the registers retain the status before DEBUG mode was entered.
If the SVDCLK.DBRUN bit = 1, the CLK_SVD4 supply is not suspended and SVD4 will keep operating in DEBUG mode.

9.4 Operations

9.4.1 SVD4 Control

Starting detection

SVD4 should be initialized and activated with the procedure listed below.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Configure the operating clock using the SVDCLK.CLKSRC[1:0] and SVDCLK.CLKDIV[2:0] bits.
3. Set the following SVDCTL register bits:
   - SVDCTL.VDSEL and SVDCTL.EXSEL bits (Select detection voltage (V_{DD}, EXSVD0, or EXSVD1))
   - SVDCTL.SVDSC[1:0] bits (Set low power supply voltage detection counter)
   - SVDCTL.SVDC[4:0] bits (Set SVD detection voltage V_{SVD}/EXSVD detection voltage V_{SVD_EXT})
   - SVDCTL.SVDRE[3:0] bits (Select reset/interrupt mode)
   - SVDCTL.SVDMD[1:0] bits (Set intermittent operation mode)
4. Set the following bits when using the interrupt:
   - Write 1 to the SVDINTF.SVDIF bit. (Clear interrupt flag)
   - Set the SVDINTE.SVDIE bit to 1. (Enable SVD4 interrupt)
5. Set the SVDCTL.MODEN bit to 1. (Enable SVD4 detection)
6. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

Terminating detection

Follow the procedure shown below to stop SVD4 operation.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Write 0 to the SVDCTL.MODEN bit. (Disable SVD4 detection)
3. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

Reading detection results

The following two detection results can be obtained by reading the SVDINTF.SVDDT bit:

- When SVDINTF.SVDDT bit = 0
  Power supply voltage (V_{DD} or EXSVD0/1) ≥ SVD detection voltage V_{SVD} or EXSVD detection voltage V_{SVD_EXT}
- When SVDINTF.SVDDT bit = 1
  Power supply voltage (V_{DD} or EXSVD0/1) < SVD detection voltage V_{SVD} or EXSVD detection voltage V_{SVD_EXT}

Before reading the SVDINTF.SVDDT bit, wait for at least SVD circuit enable response time after 1 is written to the SVDCTL.MODEN bit (refer to "Supply Voltage Detector Characteristics, SVD circuit enable response time t_{SVDEN}" in the "Electrical Characteristics" chapter).
After the SVDCTL.SVDC[4:0] bits setting value is altered to change the SVD detection voltage V_{SVD}/EXSVD detection voltage V_{SVD_EXT} when the SVDCTL.MODEN bit = 1, wait for at least SVD circuit response time before reading the SVDINTF.SVDDT bit (refer to "Supply Voltage Detector Characteristics, SVD circuit response time t_{SVD}" in the "Electrical Characteristics" chapter).

9.4.2 SVD4 Operations

Continuous operation mode

SVD4 operates in continuous operation mode by default (SVDCTL.SVDMD[1:0] bits = 0x0). In this mode, SVD4 operates continuously while the SVDCTL.MODEN bit is set to 1 and it keeps loading the detection results to the SVDINTF.SVDDT bit. During this period, the current detection results can be obtained by reading the SVDINTF.SVDDT bit as necessary. Furthermore, an interrupt (if the SVDCTL.SVDRE[3:0] bits ≠ 0xa) or a reset (if the SVDCTL.SVDRE[3:0] bits = 0xa) can be generated when the SVDINTF.SVDDT bit is set to 1 (low power supply voltage is detected). This mode can keep detecting power supply voltage drop after the voltage detection masking time has elapsed even if the IC is placed into SLEEP status or accidental clock stoppage has occurred.

Intermittent operation mode

SVD4 operates in intermittent operation mode when the SVDCTL.SVDMD[1:0] bits are set to 0x1 to 0x3. In this mode, SVD4 turns on at an interval set using the SVDCTL.SVDMD[1:0] bits to perform detection operation and then it turns off while the SVDCTL.MODEN bit is set to 1. During this period, the latest detection results can be obtained by reading the SVDINTF.SVDDT bit as necessary. Furthermore, an interrupt or a reset can be generated when SVD4 has successively detected low power supply voltage the number of times specified by the SVDCTL.SVDSC[1:0] bits.

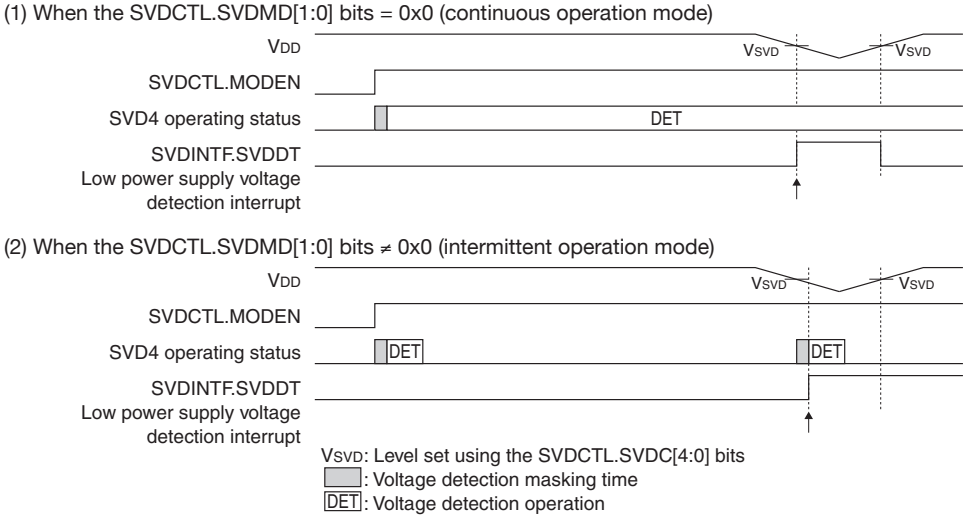


Figure 9.4.2.1 SVD4 Operations

9.5 SVD4 Interrupt and Reset

9.5.1 SVD4 Interrupt

Setting the SVDCTL.SVDRE[3:0] bits to a value other than 0xa allows use of the low power supply voltage detection interrupt function.

Table 9.5.1.1 Low Power Supply Voltage Detection Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Low power supply voltage detection	SVDINTF.SVDIF	In continuous operation mode When the SVDINTF.SVDDT bit is 1 In intermittent operation mode When low power supply voltage is successively detected the specified number of times	Writing 1

SVD4 provides the interrupt enable bit (SVDINTE.SVDIE bit) corresponding to the interrupt flag (SVDINTF.SVDIF bit). An interrupt request is sent to the interrupt controller only when the SVDINTF.SVDIF bit is set while the interrupt is enabled by the SVDINTE.SVDIE bit. For more information on interrupt control, refer to the "Interrupt Controller" chapter.

Once the SVDINTF.SVDIF bit is set, it will not be cleared even if the power supply voltage subsequently returns to a value exceeding the SVD detection voltage V_{SVD}/EXSVD detection voltage V_{SVD_EXT}. An interrupt may occur due to a temporary power supply voltage drop, check the power supply voltage status by reading the SVDINTF.SVDDT bit in the interrupt handler routine.

9.5.2 SVD4 Reset

Setting the SVDCTL.SVDRE[3:0] bits to 0xa allows use of the SVD4 reset issuance function.
The reset issuing timing is the same as that of the SVDINTF.SVDIF bit being set when a low voltage is detected.
After a reset has been issued, SVD4 enters continuous operation mode even if it was operating in intermittent operation mode, and continues operating. Issuing an SVD4 reset initializes the port assignment. However, when EXSVD0/1 is being detected, the input of the port for the EXSVD0/1 pin is sent to SVD4 so that SVD4 will continue the EXSVD0/1 detection operation.
If the power supply voltage reverts to the normal level, the SVDINTF.SVDDT bit goes 0 and the reset state is canceled. After that, SVD4 resumes operating in the operation mode set previously via the initialization routine.
During reset state, the SVD4 control bits are set as shown in Table 9.5.2.1.

Table 9.5.2.1 SVD4 Control Bits During Reset State

Control register	Control bit	Setting
SVDCLK	DBRUN	Reset to the initial values.
	CLKDIV[2:0]	
	CLKSRC[1:0]	
SVDCTL	VDSEL	The set value is retained.
	SVDSC[1:0]	Cleared to 0. (The set value becomes invalid as SVD4 enters continuous operation mode.)
	SVDC[4:0]	The set value is retained.
	SVDRE[3:0]	The set value (0xa) is retained.
	EXSEL	The set value is retained.
	SVDMD[1:0]	Cleared to 0 to set continuous operation mode.
	MODEN	The set value (1) is retained.
SVDINTF	SVDIF	The status (1) before being reset is retained.
SVDINTE	SVDIE	Cleared to 0.

9.6 Control Registers

SVD4 Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SVDCLK	15–9	–	0x00	–	R	–
	8	DBRUN	1	H0	R/WP	
	7	–	0	–	R	
	6–4	CLKDIV[2:0]	0x0	H0	R/WP	
	3–2	–	0x0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/WP	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the SVD4 operating clock is supplied in DEBUG mode or not.
1 (R/WP): Clock supplied in DEBUG mode
0 (R/WP): No clock supplied in DEBUG mode

Bit 7 Reserved

Bits 6–4 CLKDIV[2:0]

These bits select the division ratio of the SVD4 operating clock.

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of SVD4.

Table 9.6.1 Clock Source and Division Ratio Settings

SVDCLK.CLKDIV[2:0] bits	SVDCLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0x7, 0x6	Reserved	1/1	Reserved	1/1
0x5	1/128		1/512	
0x4	1/64		1/256	
0x3	1/32		1/128	
0x2	1/16		1/64	
0x1	1/8		1/32	
0x0	1/4		1/16	

(Note) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

Note: The clock frequency should be set to around 32 kHz.

SVD4 Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SVDCTL	15	VDSEL	0	H1	R/WP	–
	14–13	SVDSC[1:0]	0x0	H0	R/WP	Writing takes effect when the SVDCTL.SVDMD[1:0] bits are not 0x0.
	12–8	SVDC[4:0]	0x1e	H1	R/WP	–
	7–4	SVDRE[3:0]	0x0	H1	R/WP	
	3	EXSEL	0	H1	R/WP	
	2–1	SVDMD[1:0]	0x0	H0	R/WP	
	0	MODEN	0	H1	R/WP	

Bit 15 VDSEL

This bit selects the power supply voltage to be detected by SVD4.
1 (R/WP): Voltage applied to the EXSVD0/1 pin
0 (R/WP): V_{DD}

Bits 14–13 SVDSC[1:0]

These bits set the condition to generate an interrupt/reset (number of successive low voltage detections) in intermittent operation mode (SVDCTL.SVDMD[1:0] bits = 0x1 to 0x3).

Table 9.6.2 Interrupt/Reset Generating Condition in Intermittent Operation Mode

SVDCTL.SVDSC[1:0] bits	Interrupt/reset generating condition
0x3	Low power supply voltage is successively detected eight times.
0x2	Low power supply voltage is successively detected four times.
0x1	Low power supply voltage is successively detected twice.
0x0	Low power supply voltage is successively detected once.

This setting is ineffective in continuous operation mode (SVDCTL.SVDMD[1:0] bits = 0x0).

Bits 12–8 SVDC[4:0]

These bits select an SVD detection voltage V_{SVD}/EXSVD detection voltage V_{SVD_EXT} for detecting low voltage.

Table 9.6.3 Setting of SVD Detection Voltage V_{SVD}/EXSVD Detection Voltage V_{SVD_EXT}

SVDCTL.SVDC[4:0] bits	SVD detection voltage V_{SVD}/ EXSVD detection voltage V_{SVD_EXT} [V]
0x1f	High
0x1e	↑
0x1d	
:	
0x02	
0x01	↓
0x00	Low

For the configurable range and voltage values, refer to "Supply Voltage Detector Characteristics, SVD detection voltage V_{SVD}/EXSVD detection voltage V_{SVD_EXT}" in the "Electrical Characteristics" chapter.

Bits 7–4 SVDRE[3:0]

These bits enable/disable the reset issuance function when a low power supply voltage is detected.
0xa (R/WP): Enable (Issue reset)
Other than 0xa (R/WP): Disable (Generate interrupt)

For more information on the SVD4 reset issuance function, refer to "SVD4 Reset."

Bit 3 EXSEL

This bit selects the external voltage to be detected when the SVDCTL.VDSEL bit = 1.
1 (R/WP): EXSVD1
0 (R/WP): EXSVD0

Note: The EXSVD1 pin does not exist depending on the model (see "Power supply voltage to be detected" in Table 9.1.1). In this case, the external voltage detection function does not work if the SVDCTL.EXSEL bit is set to 1. When using the external voltage detection function (SVDCTL.VDSEL bit = 1), the SVDCTL.EXSEL bit should be set to 0.

Bits 2–1 SVDMD[1:0]

These bits select intermittent operation mode and its detection cycle.

Table 9.6.4 Intermittent Operation Mode Detection Cycle Selection

SVDCTL.SVDMD[1:0] bits	Operation mode (detection cycle)
0x3	Intermittent operation mode (CLK_SVD4/512)
0x2	Intermittent operation mode (CLK_SVD4/256)
0x1	Intermittent operation mode (CLK_SVD4/128)
0x0	Continuous operation mode

For more information on intermittent and continuous operation modes, refer to "SVD4 Operations."

Bit 0 MODEN

This bit enables/disables for the SVD4 circuit to operate.
1 (R/WP): Enable (Start detection operations)
0 (R/WP): Disable (Stop detection operations)

After this bit has been altered, wait until the value written is read out from this bit without subsequent operations being performed.

Notes:
- Writing 0 to the SVDCTL.MODEN bit resets the SVD4 hardware. However, the register values set and the interrupt flag are not cleared. The SVDCTL.MODEN bit is actually set to 0 after this processing has finished. If 1 is written to the SVDCTL.MODEN bit continuously without waiting for the bit being read as 0 at this time, writing 0 may be ignored and a malfunction may occur as the hardware restarts without resetting.
- The SVD4 internal circuit is initialized if the SVDCTL.SVDSC[1:0] bits, SVDCTL.SVDRE[3:0] bits, or SVDCTL.SVDMD[1:0] bits are altered while SVD4 is in operation after 1 is written to the SVDCTL.MODEN bit.

SVD4 Status and Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SVDINTF	15–9	–	0x00	–	R	–
	8	SVDDT	x	–	R	
	7–1	–	0x00	–	R	
	0	SVDIF	0	H1	R/W	Cleared by writing 1.

Bits 15–9 Reserved

Bit 8 SVDDT

The power supply voltage detection results can be read out from this bit.
1 (R): Power supply voltage (V_{DD} or EXSVD0/1) < SVD detection voltage V_{SVD} or EXSVD detection voltage V_{SVD_EXT}
0 (R): Power supply voltage (V_{DD} or EXSVD0/1) ≥ SVD detection voltage V_{SVD} or EXSVD detection voltage V_{SVD_EXT}

Bits 7–1 Reserved

Bit 0 SVDIF

This bit indicates the low power supply voltage detection interrupt cause occurrence status.

1 (R): Cause of interrupt occurred
0 (R): No cause of interrupt occurred
1 (W): Clear flag
0 (W): Ineffective

Note: The SVD4 internal circuit is initialized if the interrupt flag is cleared while SVD4 is in operation after 1 is written to the SVDCTL.MODEN bit.

SVD4 Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SVDINTE	15–8	–	0x00	–	R	–
	7–1	–	0x00	–	R	
	0	SVDIE	0	H0	R/W	

Bits 15–1 Reserved

Bit 0 SVDIE

This bit enables low power supply voltage detection interrupts.

1 (R/W): Enable interrupts
0 (R/W): Disable interrupts

Notes: • If the SVDCTL.SVDRE[3:0] bits are set to 0xa, no low power supply voltage detection interrupt will occur, as a reset is issued at the same timing as an interrupt.

• To prevent generating unnecessary interrupts, the corresponding interrupt flag should be cleared before enabling interrupts.

10 16-bit Timers (T16)

10.1 Overview

T16 is a 16-bit timer. The features of T16 are listed below.

- 16-bit presettable down counter
- Provides a reload data register for setting the preset value.
- A clock source and clock division ratio for generating the count clock are selectable.
- Repeat mode or one-shot mode is selectable.
- Can generate counter underflow interrupts.

Figure 10.1.1 shows the configuration of a T16 channel.

Table 10.1.1 T16 Channel Configuration of S1C17M02/M03

Item	S1C17M02	S1C17M03
Number of channels	4 channels (Ch.0–Ch.3)	
Event counter function	Not supported (No EXCL*m* pins are provided.)	
Peripheral clock output (Outputs the counter underflow signal.)	Ch.1 → Synchronous serial interface Ch.0 master clock	

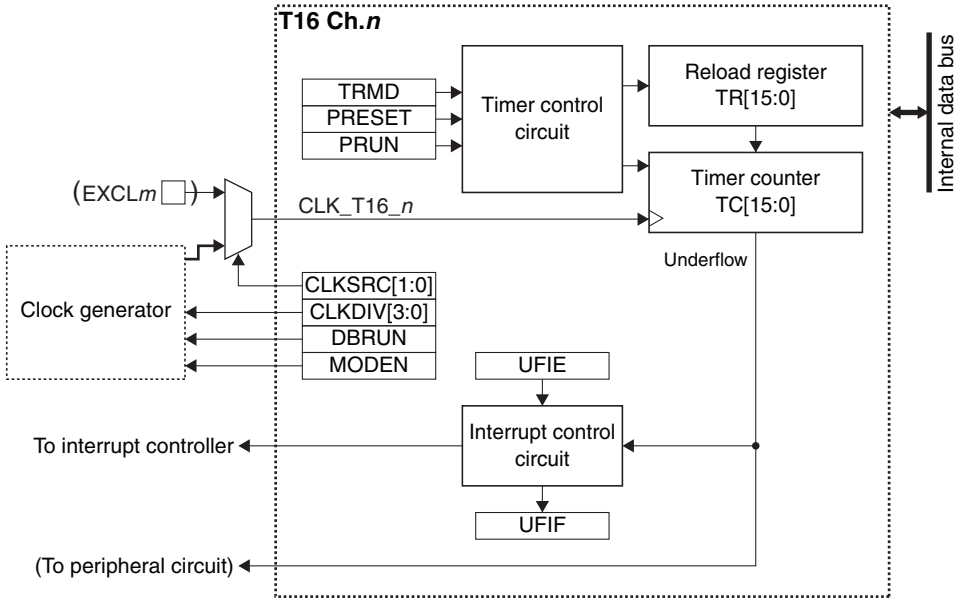


Figure 10.1.1 Configuration of a T16 Channel

10.2 Input Pin

Table 10.2.1 shows the T16 input pin.

Table 10.2.1 T16 Input Pin

Pin name	I/O*	Initial status*	Function
EXCL*m*	I	I (Hi-Z)	External event signal input pin

* Indicates the status when the pin is configured for T16.

If the port is shared with the EXCL*m* pin and other functions, the EXCL*m* input function must be assigned to the port before using the event counter function. For more information, refer to the "I/O Ports" chapter.

10.3 Clock Settings

10.3.1 T16 Operating Clock

When using T16 Ch.*n*, the T16 Ch.*n* operating clock CLK_T16_*n* must be supplied to T16 Ch.*n* from the clock generator. The CLK_T16_*n* supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to "Clock Generator" in the "Power Supply, Reset, and Clocks" chapter).
2. Set the following T16_*n*CLK register bits:
   - T16_*n*CLK.CLKSRC[1:0] bits (Clock source selection)
   - T16_*n*CLK.CLKDIV[3:0] bits (Clock division ratio selection = Clock frequency setting)

10.3.2 Clock Supply in SLEEP Mode

When using T16 during SLEEP mode, the T16 operating clock CLK_T16_*n* must be configured so that it will keep supplying by writing 0 to the CLGOSC.*xxxx*SLPC bit for the CLK_T16_*n* clock source.
If the CLGOSC.*xxxx*SLPC bit for the CLK_T16_*n* clock source is 1, the CLK_T16_*n* clock source is deactivated during SLEEP mode and T16 stops with the register settings and counter value maintained at those before entering SLEEP mode. After the CPU returns to normal mode, CLK_T16_*n* is supplied and the T16 operation resumes.

10.3.3 Clock Supply in DEBUG Mode

The CLK_T16_*n* supply during DEBUG mode should be controlled using the T16_*n*CLK.DBRUN bit.
The CLK_T16_*n* supply to T16 Ch.*n* is suspended when the CPU enters DEBUG mode if the T16_*n*CLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_T16_*n* supply resumes. Although T16 Ch.*n* stops operating when the CLK_T16_*n* supply is suspended, the counter and registers retain the status before DEBUG mode was entered. If the T16_*n*CLK.DBRUN bit = 1, the CLK_T16_*n* supply is not suspended and T16 Ch.*n* will keep operating in DEBUG mode.

10.3.4 Event Counter Clock

The channel that supports the event counter function counts down at the rising edge of the EXCL*m* pin input signal when the T16_*n*CLK.CLKSRC[1:0] bits are set to 0x3.

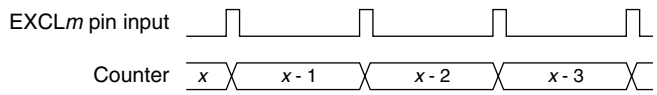


Figure 10.3.4.1 Count Down Timing

Note that the EXOSC clock is selected for the channel that does not support the event counter function.

10.4 Operations

10.4.1 Initialization

T16 Ch.*n* should be initialized and started counting with the procedure shown below.

1. Configure the T16 Ch.*n* operating clock (see "T16 Operating Clock").
2. Set the T16_*n*CTL.MODEN bit to 1. (Enable count operation clock)
3. Set the T16_*n*MOD.TRMD bit. (Select operation mode (Repeat mode or One-shot mode))
4. Set the T16_*n*TR register. (Set reload data (counter preset data))
5. Set the following bits when using the interrupt:
   - Write 1 to the T16_*n*INTF.UFIF bit. (Clear interrupt flag)
   - Set the T16_*n*INTE.UFIE bit to 1. (Enable underflow interrupt)

6. Set the following T16_*n*CTL register bits:
   - Set the T16_*n*CTL.PRESET bit to 1. (Preset reload data to counter)
   - Set the T16_*n*CTL.PRUN bit to 1. (Start counting)

10.4.2 Counter Underflow

Normally, the T16 counter starts counting down from the reload data value preset and generates an underflow signal when an underflow occurs. This signal is used to generate an interrupt and may be output to a specific peripheral circuit as a clock (T16 Ch.*n* must be set to repeat mode to generate a clock). The underflow cycle is determined by the T16 Ch.*n* operating clock setting and reload data (counter initial value) set in the T16_*n*TR register.
The following shows the equations to calculate the underflow cycle and frequency:

$$T = \frac{TR + 1}{f_{CLK\_T16\_n}} \qquad fr = \frac{f_{CLK\_T16\_n}}{TR + 1} \qquad \text{(Eq. 10.1)}$$

Where

T: Underflow cycle [s]
fr: Underflow frequency [Hz]
TR: T16_*n*TR register setting
$f_{CLK\_T16\_n}$: T16 Ch.*n* operating clock frequency [Hz]

10.4.3 Operations in Repeat Mode

T16 Ch.*n* enters repeat mode by setting the T16_*n*MOD.TRMD bit to 0.
In repeat mode, the count operation starts by writing 1 to the T16_*n*CTL.PRUN bit and continues until 0 is written. A counter underflow presets the T16_*n*TR register value to the counter, so underflow occurs periodically. Select this mode to generate periodic underflow interrupts or when using the timer to output a trigger/clock to the peripheral circuit.

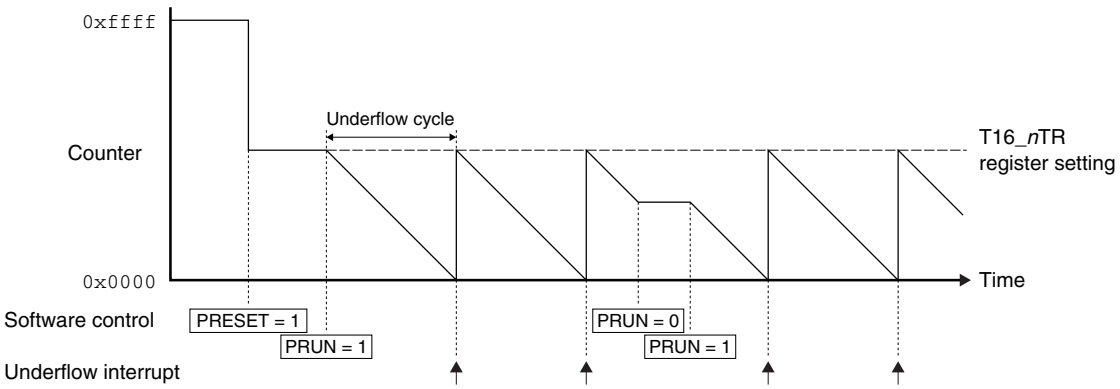


Figure 10.4.3.1 Count Operations in Repeat Mode

10.4.4 Operations in One-shot Mode

T16 Ch.*n* enters one-shot mode by setting the T16_*n*MOD.TRMD bit to 1.
In one-shot mode, the count operation starts by writing 1 to the T16_*n*CTL.PRUN bit and stops after the T16_*n*TR register value is preset to the counter when an underflow has occurred. At the same time the counter stops, the T16_*n*CTL.PRUN bit is cleared automatically. Select this mode to stop the counter after an interrupt has occurred once, such as for checking a specific lapse of time.

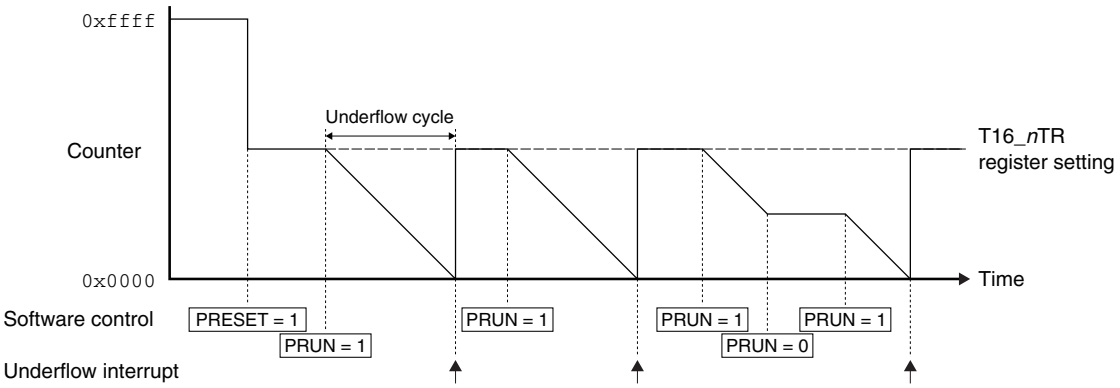


Figure 10.4.4.1 Count Operations in One-shot Mode

10.4.5 Counter Value Read

The counter value can be read out from the T16_*n*TC.TC[15:0] bits. However, since T16 operates on CLK_T16_*n*, one of the operations shown below is required to read correctly by the CPU.

- Read the counter value twice or more and check to see if the same value is read.
- Stop the timer and then read the counter value.

10.5 Interrupt

Each T16 channel has a function to generate the interrupt shown in Table 10.5.1.

Table 10.5.1 T16 Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Underflow	T16_*n*INTF.UFIF	When the counter underflows	Writing 1

The T16 provides an interrupt enable bit corresponding to the interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the "Interrupt Controller" chapter.

10.6 Control Registers

T16 Ch.*n* Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16_*n*CLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/W	
	7–4	CLKDIV[3:0]	0x0	H0	R/W	
	3–2	–	0x0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/W	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the T16 Ch.*n* operating clock is supplied in DEBUG mode or not.

1 (R/W): Clock supplied in DEBUG mode

0 (R/W): No clock supplied in DEBUG mode

Bits 7–4 CLKDIV[3:0]

These bits select the division ratio of the T16 Ch.*n* operating clock (counter clock).

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of T16 Ch.*n*.

Table 10.6.1 Clock Source and Division Ratio Settings

T16_*n*CLK.CLKDIV[3:0] bits	T16_*n*CLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC/EXCL*m*
0xf	1/32,768	1/1	1/32,768	1/1
0xe	1/16,384		1/16,384	
0xd	1/8,192		1/8,192	
0xc	1/4,096		1/4,096	
0xb	1/2,048		1/2,048	
0xa	1/1,024		1/1,024	
0x9	1/512		1/512	
0x8	1/256	1/256	1/256	
0x7	1/128	1/128	1/128	
0x6	1/64	1/64	1/64	
0x5	1/32	1/32	1/32	
0x4	1/16	1/16	1/16	
0x3	1/8	1/8	1/8	
0x2	1/4	1/4	1/4	
0x1	1/2	1/2	1/2	
0x0	1/1	1/1	1/1	

(Note 1) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.
(Note 2) When the T16_*n*CLK.CLKSRC[1:0] bits are set to 0x3, EXCL*m* is selected for the channel with an event counter function or EXOSC is selected for other channels.

T16 Ch.*n* Mode Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16_*n*MOD	15–8	–	0x00	–	R	–
	7–1	–	0x00	–	R	
	0	TRMD	0	H0	R/W	

Bits 15–1 Reserved

Bit 0 TRMD

This bit selects the T16 operation mode.
1 (R/W): One-shot mode
0 (R/W): Repeat mode

For detailed information on the operation mode, refer to "Operations in One-shot Mode" and "Operations in Repeat Mode."

T16 Ch.*n* Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16_*n*CTL	15–9	–	0x00	–	R	–
	8	PRUN	0	H0	R/W	
	7–2	–	0x00	–	R	
	1	PRESET	0	H0	R/W	
	0	MODEN	0	H0	R/W	

Bits 15–9 Reserved

Bit 8 PRUN

This bit starts/stops the timer.
1 (W): Start timer
0 (W): Stop timer
1 (R): Timer is running
0 (R): Timer is idle

By writing 1 to this bit, the timer starts count operations. However, the T16_*n*CTL.MODEN bit must be set to 1 in conjunction with this bit or it must be set in advance. While the timer is running, writing 0 to this bit stops count operations. When the counter stops due to a counter underflow in one-shot mode, this bit is automatically cleared to 0.

Bits 7–2 Reserved

Bit 1 PRESET

This bit presets the reload data stored in the T16_*n*TR register to the counter.

1 (W): Preset
0 (W): Ineffective
1 (R): Presetting in progress
0 (R): Presetting finished or normal operation

By writing 1 to this bit, the timer presets the T16_*n*TR register value to the counter. However, the T16_*n*CTL.MODEN bit must be set to 1 in conjunction with this bit or it must be set in advance. This bit retains 1 during presetting and is automatically cleared to 0 after presetting has finished.

Bit 0 MODEN

This bit enables the T16 Ch.*n* operations.

1 (R/W): Enable (Start supplying operating clock)
0 (R/W): Disable (Stop supplying operating clock)

T16 Ch.*n* Reload Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16_*n*TR	15–0	TR[15:0]	0xffff	H0	R/W	–

Bits 15–0 TR[15:0]

These bits are used to set the initial value to be preset to the counter.
The value set to this register will be preset to the counter when 1 is written to the T16_*n*CTL.PRESET bit or when the counter underflows.

Notes:
- The T16_*n*TR register cannot be altered while the timer is running (T16_*n*CTL.PRUN bit = 1), as an incorrect initial value may be preset to the counter.
- When one-shot mode is set, the T16_*n*TR.TR[15:0] bits should be set to a value equal to or greater than 0x0001.

T16 Ch.*n* Counter Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16_*n*TC	15–0	TC[15:0]	0xffff	H0	R	–

Bits 15–0 TC[15:0]

The current counter value can be read out from these bits.

T16 Ch.*n* Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16_*n*INTF	15–8	–	0x00	–	R	–
	7–1	–	0x00	–	R	
	0	UFIF	0	H0	R/W	Cleared by writing 1.

Bits 15–1 Reserved

Bit 0 UFIF

This bit indicates the T16 Ch.*n* underflow interrupt cause occurrence status.

1 (R): Cause of interrupt occurred
0 (R): No cause of interrupt occurred
1 (W): Clear flag
0 (W): Ineffective

T16 Ch.*n* Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16_*n*INTE	15–8	–	0x00	–	R	–
	7–1	–	0x00	–	R	
	0	UFIE	0	H0	R/W	

Bits 15–1 Reserved

Bit 0 UFIE

This bit enables T16 Ch.*n* underflow interrupts.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts

Note: To prevent generating unnecessary interrupts, the corresponding interrupt flag should be cleared before enabling interrupts.

11 UART (UART3)

11.1 Overview

The UART3 is an asynchronous serial interface. The features of the UART3 are listed below.

- Includes a baud rate generator for generating the transfer clock.
- Supports 7- and 8-bit data length (LSB first).
- Odd parity, even parity, or non-parity mode is selectable.
- The start bit length is fixed at 1 bit.
- The stop bit length is selectable from 1 bit and 2 bits.
- Supports full-duplex communications.
- Includes a 2-byte receive data buffer and a 1-byte transmit data buffer.
- Includes an RZI modulator/demodulator circuit to support IrDA 1.0-compatible infrared communications.
- Can detect parity error, framing error, and overrun error.
- Can generate receive buffer full (1 byte/2 bytes), transmit buffer empty, end of transmission, parity error, framing error, and overrun error interrupts.
- Input pin can be pulled up with an internal resistor.
- The output pin is configurable as an open-drain output.
- Provides the carrier modulation output function.

Figure 11.1.1 shows the UART3 configuration.

Table 11.1.1 UART3 Channel Configuration of S1C17M02/M03

Item	S1C17M02	S1C17M03
Number of channels	1 channel (Ch.0)	

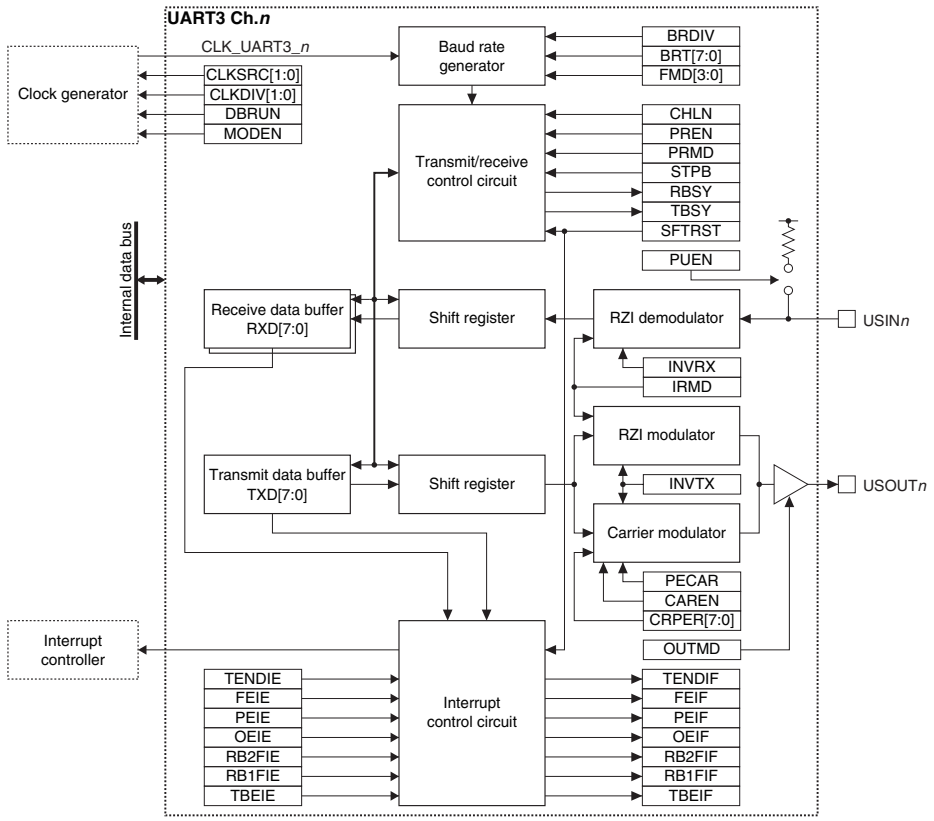


Figure 11.1.1 UART3 Configuration

11.2 Input/Output Pins and External Connections

11.2.1 List of Input/Output Pins

Table 11.2.1.1 lists the UART3 pins.

Table 11.2.1.1 List of UART3 Pins

Pin name	I/O*	Initial status*	Function
USIN*n*	I	I (Hi-Z)	UART3 Ch.*n* data input pin
USOUT*n*	O	O (High)	UART3 Ch.*n* data output pin

* Indicates the status when the pin is configured for the UART3.

If the port is shared with the UART3 pin and other functions, the UART3 input/output function must be assigned to the port before activating the UART3. For more information, refer to the "I/O Ports" chapter.

11.2.2 External Connections

Figure 11.2.2.1 shows a connection diagram between the UART3 in this IC and an external UART device.

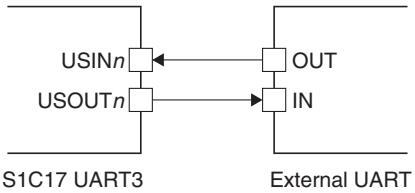


Figure 11.2.2.1 Connections between UART3 and an External UART Device

11.2.3 Input Pin Pull-Up Function

The UART3 includes a pull-up resistor for the USIN*n* pin. Setting the UA*n*MOD.PUEN bit to 1 enables the resistor to pull up the USIN*n* pin.

11.2.4 Output Pin Open-Drain Output Function

The USOUT*n* pin supports the open-drain output function. Default configuration is a push-pull output and it is switched to an open-drain output by setting the UA*n*MOD.OUTMD bit to 1.

11.2.5 Input/Output Signal Inverting Function

The UART3 can invert the signal polarities of the USIN*n* pin input and the USOUT*n* pin output by setting the UA*n*MOD.INVRX bit and the UA*n*MOD.INVTX bit, respectively, to 1.

Note: Unless otherwise specified, this chapter shows input/output signals with non-inverted waveforms (UA*n*MOD.INVRX bit = 0, UA*n*MOD.INVTX bit =0).

11.3 Clock Settings

11.3.1 UART3 Operating Clock

When using the UART3 Ch.*n*, the UART3 Ch.*n* operating clock CLK_UART3_*n* must be supplied to the UART3 Ch.*n* from the clock generator. The CLK_UART3_*n* supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to "Clock Generator" in the "Power Supply, Reset, and Clocks" chapter).
2. Set the following UA*n*CLK register bits:
   - UA*n*CLK.CLKSRC[1:0] bits (Clock source selection)
   - UA*n*CLK.CLKDIV[1:0] bits (Clock division ratio selection = Clock frequency setting)

   The UART3 operating clock should be selected so that the baud rate generator will be configured easily.

11.3.2 Clock Supply in SLEEP Mode

When using the UART3 during SLEEP mode, the UART3 operating clock CLK_UART3_*n* must be configured so that it will keep supplying by writing 0 to the CLGOSC.*xxxx*SLPC bit for the CLK_UART3_*n* clock source.

11.3.3 Clock Supply in DEBUG Mode

The CLK_UART3_*n* supply during DEBUG mode should be controlled using the UA*n*CLK.DBRUN bit.
The CLK_UART3_*n* supply to the UART3 Ch.*n* is suspended when the CPU enters DEBUG mode if the UA*n*CLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_UART3_*n* supply resumes. Although the UART3 Ch.*n* stops operating when the CLK_UART3_*n* supply is suspended, the output pin and registers retain the status before DEBUG mode was entered. If the UA*n*CLK.DBRUN bit = 1, the CLK_UART3_*n* supply is not suspended and the UART3 Ch.*n* will keep operating in DEBUG mode.

11.3.4 Baud Rate Generator

The UART3 includes a baud rate generator to generate the transfer (sampling) clock. The transfer rate is determined by the UA*n*MOD.BRDIV, UA*n*BR.BRT[7:0], and UA*n*BR.FMD[3:0] bit settings. Use the following equations to calculate the setting values for obtaining the desired transfer rate.

$$bps = \frac{CLK\_UART3}{\frac{BRT + 1}{BRDIV} + FMD} \qquad BRT = BRDIV \times \left(\frac{CLK\_UART3}{bps} - FMD\right) - 1 \qquad \text{(Eq. 11.1)}$$

Where

bps:	Transfer rate [bit/s]
CLK_UART3:	UART3 operating clock frequency [Hz]
BRDIV:	Baud rate division ratio (1/16 or 1/4) * Selected by the UA*n*MOD.BRDIV bit
BRT:	UA*n*BR.BRT[7:0] setting value (0 to 255)
FMD:	UA*n*BR.FMD[3:0] setting value (0 to 15)

For the transfer rate range configurable in the UART3, refer to "UART Characteristics, Transfer baud rates U_{BRT1} and U_{BRT2}" in the "Electrical Characteristics" chapter.

11.4 Data Format

The UART3 allows setting of the data length, stop bit length, and parity function. The start bit length is fixed at one bit.

Data length

With the UA*n*MOD.CHLN bit, the data length can be set to seven bits (UA*n*MOD.CHLN bit = 0) or eight bits (UA*n*MOD.CHLN bit = 1).

Stop bit length

With the UA*n*MOD.STPB bit, the stop bit length can be set to one bit (UA*n*MOD.STPB bit = 0) or two bits (UA*n*MOD.STPB bit = 1).

Parity function

The parity function is configured using the UA*n*MOD.PREN and UA*n*MOD.PRMD bits.

Table 11.4.1 Parity Function Setting

UA*n*MOD.PREN bit	UA*n*MOD.PRMD bit	Parity function
1	1	Odd parity
1	0	Even parity
0	*	Non parity

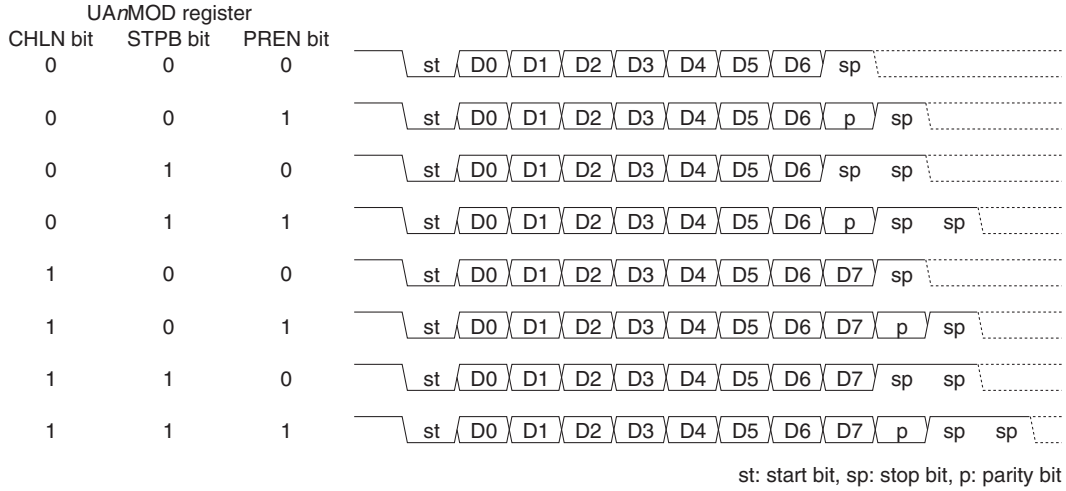


Figure 11.4.1 Data Format

11.5 Operations

11.5.1 Initialization

The UART3 Ch.*n* should be initialized with the procedure shown below.

1. Assign the UART3 Ch.*n* input/output function to the ports. (Refer to the "I/O Ports" chapter.)
2. Set the UA*n*CLK.CLKSRC[1:0] and UA*n*CLK.CLKDIV[1:0] bits. (Configure operating clock)
3. Configure the following UA*n*MOD register bits:
   - UA*n*MOD.BRDIV bit (Select baud rate division ratio (1/16 or 1/4))
   - UAnMOD.INVRX bit (Enable/disable USIN*n* input signal inversion)
   - UAnMOD.INVTX bit (Enable/disable USOUT*n* output signal inversion)
   - UA*n*MOD.PUEN bit (Enable/disable USIN*n* pin pull-up)
   - UA*n*MOD.OUTMD bit (Enable/disable USOUT*n* pin open-drain output)
   - UA*n*MOD.IRMD bit (Enable/disable IrDA interface)
   - UA*n*MOD.CHLN bit (Set data length (7 or 8 bits))
   - UA*n*MOD.PREN bit (Enable/disable parity function)
   - UA*n*MOD.PRMD bit (Select parity mode (even or odd))
   - UA*n*MOD.STPB bit (Set stop bit length (1 or 2 bits))
   - UA*n*MOD.CAREN bit (Enable/disable carrier modulation function)
   - UAnMOD.PECAR bit (Select carrier modulation period (H data period/L data period))
4. Set the UA*n*BR.BRT[7:0] and UA*n*BR.FMD[3:0] bits. (Set transfer rate)
5. Set the UA*n*CAWF.CRPER[7:0] bits. (Set carrier cycle)
6. Set the following UA*n*CTL register bits:
   - Set the UA*n*CTL.SFTRST bit to 1. (Execute software reset)
   - Set the UA*n*CTL.MODEN bit to 1. (Enable UART3 Ch.*n* operations)
7. Set the following bits when using the interrupt:
   - Write 1 to the interrupt flags in the UA*n*INTF register. (Clear interrupt flags)
   - Set the interrupt enable bits in the UA*n*INTE register to 1. * (Enable interrupts)

   * The initial value of the UA*n*INTF.TBEIF bit is 1, therefore, an interrupt will occur immediately after the UA*n*INTE.TBEIE bit is set to 1.

11.5.2 Data Transmission

A data sending procedure and the UART3 Ch.*n* operations are shown below. Figures 11.5.2.1 and 11.5.2.2 show a timing chart and a flowchart, respectively.

Data sending procedure

1. Check to see if the UA*n*INTF.TBEIF bit is set to 1 (transmit buffer empty).
2. Write transmit data to the UA*n*TXD register.
3. Wait for a UART3 interrupt when using the interrupt.
4. Repeat Steps 1 to 3 (or 1 and 2) until the end of transmit data.

UART3 data sending operations

The UART3 Ch.*n* starts data sending operations when transmit data is written to the UA*n*TXD register.
The transmit data in the UA*n*TXD register is automatically transferred to the shift register and the UA*n*INTF.TBEIF bit is set to 1 (transmit buffer empty).
The USOUT*n* pin outputs a start bit and the UA*n*INTF.TBSY bit is set to 1 (transmit busy). The shift register data bits are then output successively from the LSB. Following output of MSB, the parity bit (if parity is enabled) and the stop bit are output.

Even if transmit data is being output from the USOUT*n* pin, the next transmit data can be written to the UA*n*TXD register after making sure the UA*n*INTF.TBEIF bit is set to 1.

If no transmit data remains in the UA*n*TXD register after the stop bit has been output from the USOUT*n* pin, the UA*n*INTF.TBSY bit is cleared to 0 and the UA*n*INTF.TENDIF bit is set to 1 (transmission completed).

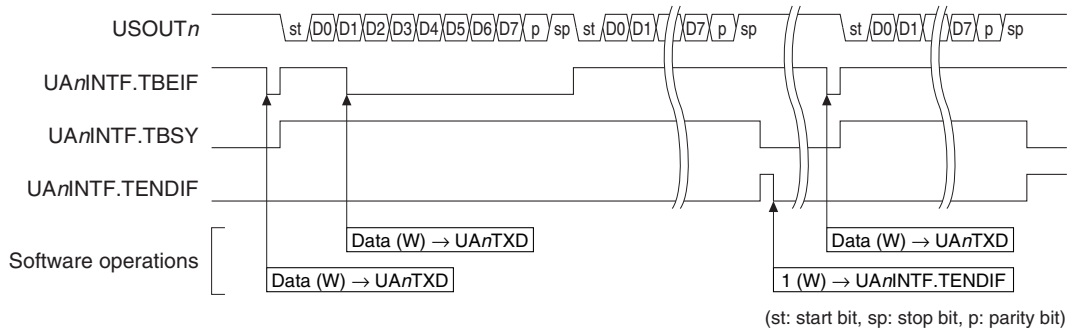


Figure 11.5.2.1 Example of Data Sending Operations

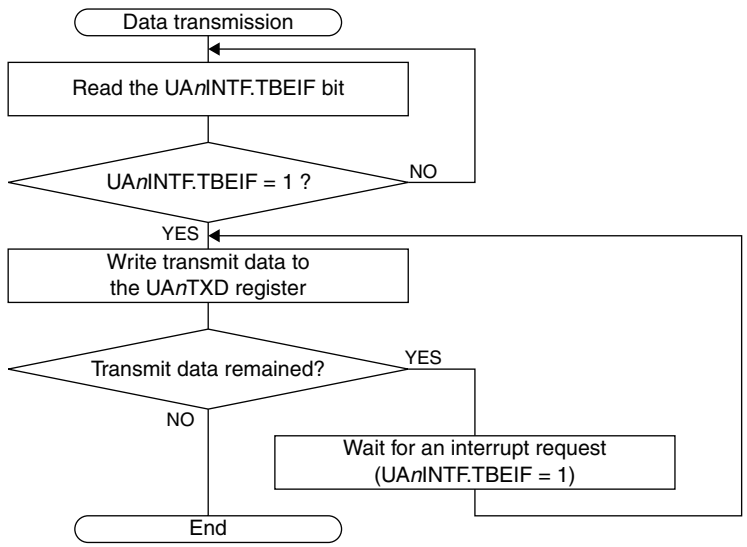


Figure 11.5.2.2 Data Transmission Flowchart

11.5.3 Data Reception

A data receiving procedure and the UART3 Ch.*n* operations are shown below. Figures 11.5.3.1 and 11.5.3.2 show a timing chart and flowcharts, respectively.

Data receiving procedure (read by one byte)

1. Wait for a UART3 interrupt when using the interrupt.
2. Check to see if the UA*n*INTF.RB1FIF bit is set to 1 (receive buffer one byte full).
3. Read the received data from the UA*n*RXD register.
4. Repeat Steps 1 to 3 (or 2 and 3) until the end of data reception.

Data receiving procedure (read by two bytes)

1. Wait for a UART3 interrupt when using the interrupt.
2. Check to see if the UA*n*INTF.RB2FIF bit is set to 1 (receive buffer two bytes full).
3. Read the received data from the UA*n*RXD register twice.
4. Repeat Steps 1 to 3 (or 2 and 3) until the end of data reception.

UART3 data receiving operations

The UART3 Ch.*n* starts data receiving operations when a start bit is input to the USIN*n* pin.
After the receive circuit has detected a low level as a start bit, it starts sampling the following data bits and loads the received data into the receive shift register. The UA*n*INTF.RBSY bit is set to 1 when the start bit is detected.
The UA*n*INTF.RBSY bit is cleared to 0 and the receive shift register data is transferred to the receive data buffer at the stop bit receive timing.
The receive data buffer consists of a 2-byte FIFO and receives data until it becomes full. When the receive data buffer receives the first data, it sets the UA*n*INTF.RB1FIF bit to 1 (receive buffer one byte full). If the second data is received without reading the first data, the UA*n*INTF.RB2FIF bit is set to 1 (receive buffer two bytes full).

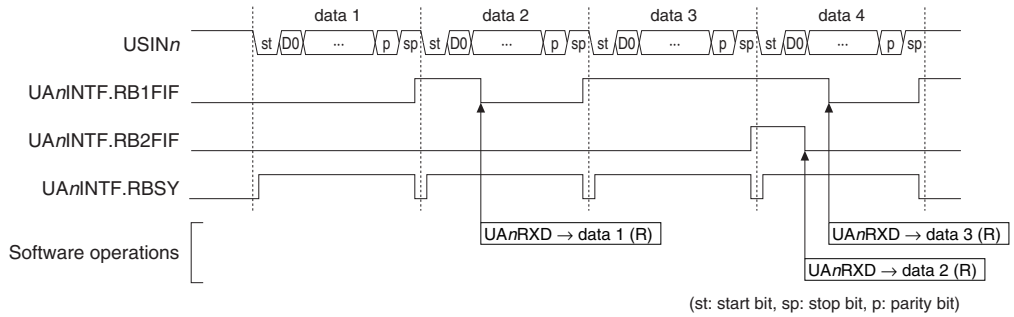


Figure 11.5.3.1 Example of Data Receiving Operations

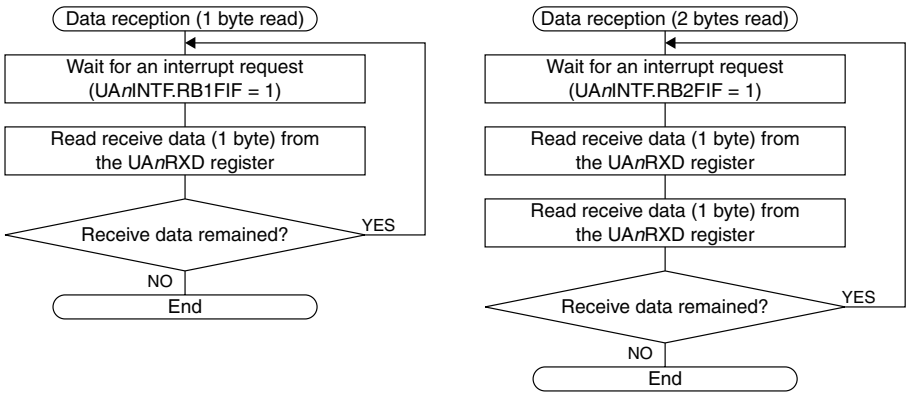


Figure 11.5.3.2 Data Reception Flowcharts

11.5.4 IrDA Interface

This UART3 includes an RZI modulator/demodulator circuit enabling implementation of IrDA 1.0-compatible infrared communication function simply by adding simple external circuits.
Set the UA*n*MOD.IRMD bit to 1 to use the IrDA interface.
Data transfer control is identical to that for normal interface even if the IrDA interface function is enabled.

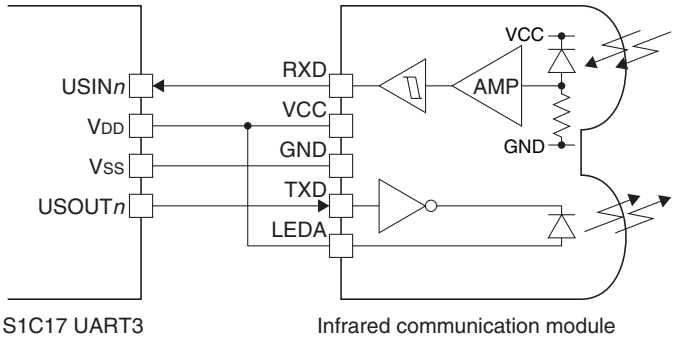


Figure 11.5.4.1 Example of Connections with an Infrared Communication Module

The transmit data output from the UART3 Ch.*n* transmit shift register is output from the USOUT*n* pin after the low pulse width is converted into 3/16 by the RZI modulator in SIR method.

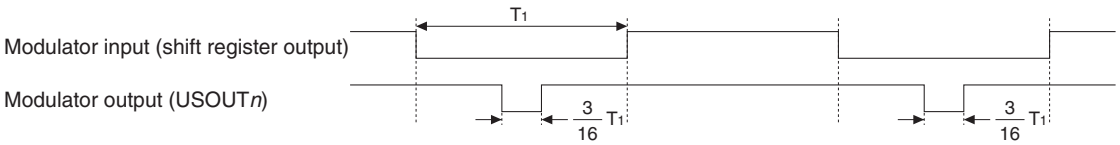


Figure 11.5.4.2 IrDA Transmission Signal Waveform

The received IrDA signal is input to the RZI demodulator and the low pulse width is converted into the normal width before input to the receive shift register.

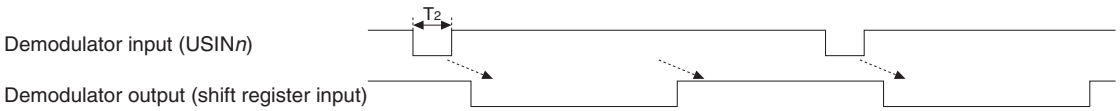


Figure 11.5.4.3 IrDA Receive Signal Waveform

Notes: • Set the baud rate division ratio to 1/16 when using the IrDA interface function.

• The low pulse width (T_2) of the IrDA signal input must be CLK_UART3 × 3 cycles or longer.

11.5.5 Carrier Modulation

The UART3 has a carrier modulation function.
Writing 1 to the UA*n*MOD.CAREN bit enables the carrier modulation function allowing carrier modulation waveforms to be output according to the UA*n*MOD.PECAR bit setting. Data transmit control is identical to that for normal interface even in this case.

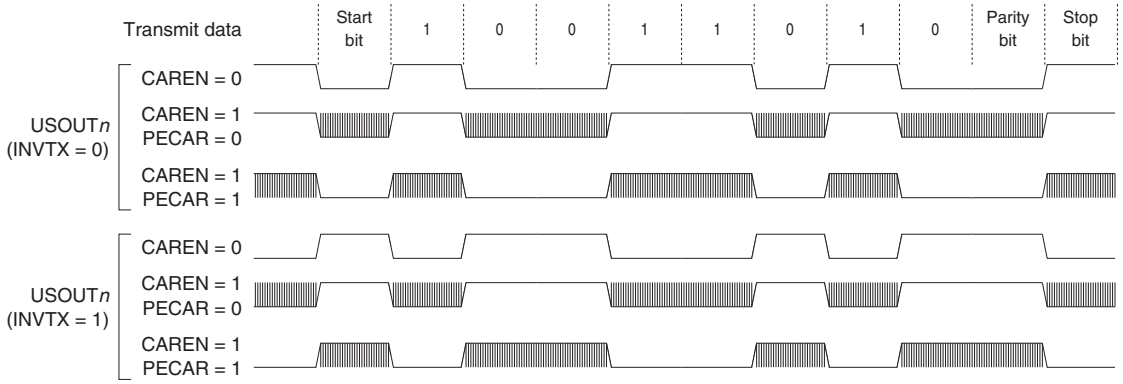


Figure 11.5.5.1 Carrier Modulation Waveform (UA*n*MOD.CHLN = 1, UA*n*MOD.STPB = 0, UA*n*MOD.PREN = 1)

The carrier modulation output frequency is determined by the UA*n*CAWF.CRPER[7:0] bit settings. Use the following equations to calculate the setting values for obtaining the desired frequency.

$$\text{Carrier modulation output frequency} = \frac{\text{CLK\_UART3}}{(\text{CRPER} + 1) \times 2} \text{ [Hz]} \qquad \text{(Eq. 11.2)}$$

Where

CLK_UART3: UART3 operating clock frequency [Hz]
CRPER: UA*n*CAWF.CRPER[7:0] setting value (0 to 255)

11.6 Receive Errors

Three different receive errors, framing error, parity error, and overrun error, may be detected while receiving data. Since receive errors are interrupt causes, they can be processed by generating interrupts.

11.6.1 Framing Error

The UART3 determines loss of sync if a stop bit is not detected (when the stop bit is received as 0) and assumes that a framing error has occurred. The received data that encountered an error is still transferred to the receive data buffer and the UA*n*INTF.FEIF bit (framing error interrupt flag) is set to 1 when the data becomes ready to read from the UA*n*RXD register.

Note: Framing error/parity error interrupt flag set timings
These interrupt flags will be set after the data that encountered an error is transferred to the receive data buffer. Note, however, that the set timing depends on the buffer status at that point.

- When the receive data buffer is empty
  The interrupt flag will be set when the data that encountered an error is transferred to the receive data buffer.
- When the receive data buffer has a one-byte free space
  The interrupt flag will be set when the first data byte already loaded is read out after the data that encountered an error is transferred to the second byte entry of the receive data buffer.

11.6.2 Parity Error

If the parity function is enabled, a parity check is performed when data is received. The UART3 checks matching between the data received in the shift register and its parity bit, and issues a parity error if the result is a non-match. The received data that encountered an error is still transferred to the receive data buffer and the UA*n*INTF.PEIF bit (parity error interrupt flag) is set to 1 when the data becomes ready to read from the UA*n*RXD register (see the Note on framing error).

11.6.3 Overrun Error

If the receive data buffer is still full (two bytes of received data have not been read) when a data reception to the shift register has completed, an overrun error occurs as the data cannot be transferred to the receive data buffer. When an overrun error occurs, the UA*n*INTF.OEIF bit (overrun error interrupt flag) is set to 1.

11.7 Interrupts

The UART3 has a function to generate the interrupts shown in Table 11.7.1.

Table 11.7.1 UART3 Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
End of transmission	UA*n*INTF.TENDIF	When the UA*n*INTF.TBEIF bit = 1 after the stop bit has been sent	Writing 1 or software reset
Framing error	UA*n*INTF.FEIF	Refer to the "Receive Errors."	Writing 1, reading received data that encountered an error, or software reset
Parity error	UA*n*INTF.PEIF	Refer to the "Receive Errors."	Writing 1, reading received data that encountered an error, or software reset
Overrun error	UA*n*INTF.OEIF	Refer to the "Receive Errors."	Writing 1 or software reset
Receive buffer two bytes full	UA*n*INTF.RB2FIF	When the second received data byte is loaded to the receive data buffer in which the first byte is already received	Reading received data or software reset
Receive buffer one byte full	UA*n*INTF.RB1FIF	When the first received data byte is loaded to the emptied receive data buffer	Reading data to empty the receive data buffer or software reset
Transmit buffer empty	UA*n*INTF.TBEIF	When transmit data written to the transmit data buffer is transferred to the shift register	Writing transmit data

The UART3 provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the "Interrupt Controller" chapter.

11.8 Control Registers

UART3 Ch.*n* Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UA*n*CLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/W	
	7–6	–	0x0	–	R	
	5–4	CLKDIV[1:0]	0x0	H0	R/W	
	3–2	–	0x0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/W	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the UART3 operating clock is supplied in DEBUG mode or not.

1 (R/W): Clock supplied in DEBUG mode

0 (R/W): No clock supplied in DEBUG mode

Bits 7–6 Reserved

Bits 5–4 CLKDIV[1:0]

These bits select the division ratio of the UART3 operating clock.

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of the UART3.

Table 11.8.1 Clock Source and Division Ratio Settings

UA*n*CLK.CLKDIV[1:0] bits	UA*n*CLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0x3	1/8	1/1	1/8	1/1
0x2	1/4		1/4	
0x1	1/2		1/2	
0x0	1/1		1/1	

(Note) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

Note: The UA*n*CLK register settings can be altered only when the UA*n*CTL.MODEN bit = 0.

UART3 Ch.*n* Mode Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UA*n*MOD	15–13	–	0x0	–	R	–
	12	PECAR	0	H0	R/W	
	11	CAREN	0	H0	R/W	
	10	BRDIV	0	H0	R/W	
	9	INVRX	0	H0	R/W	
	8	INVTX	0	H0	R/W	
	7	–	0	–	R	
	6	PUEN	0	H0	R/W	
	5	OUTMD	0	H0	R/W	
	4	IRMD	0	H0	R/W	
	3	CHLN	0	H0	R/W	
	2	PREN	0	H0	R/W	
	1	PRMD	0	H0	R/W	
	0	STPB	0	H0	R/W	

Bits 15–13 Reserved

Bit 12 PECAR

This bit selects the carrier modulation period.

1 (R/W): Carrier modulation during H data period

0 (R/W): Carrier modulation during L data period

Bit 11 CAREN

This bit enables the carrier modulation function.

1 (R/W): Enable carrier modulation function

0 (R/W): Disable carrier modulation function

Bit 10 BRDIV

This bit sets the UART3 operating clock division ratio for generating the transfer (sampling) clock using the baud rate generator.

1 (R/W): 1/4

0 (R/W): 1/16

Bit 9 INVRX

This bit enables the USIN*n* input inverting function.

1 (R/W): Enable input inverting function

0 (R/W): Disable input inverting function

Bit 8 INVTX

This bit enables the USOUT*n* output inverting function.

1 (R/W): Enable output inverting function

0 (R/W): Disable output inverting function

Bit 7 Reserved

Bit 6 PUEN

This bit enables pull-up of the USIN*n* pin.
1 (R/W): Enable pull-up
0 (R/W): Disable pull-up

Bit 5 OUTMD

This bit sets the USOUT*n* pin output mode.
1 (R/W): Open-drain output
0 (R/W): Push-pull output

Bit 4 IRMD

This bit enables the IrDA interface function.
1 (R/W): Enable IrDA interface function
0 (R/W): Disable IrDA interface function

Bit 3 CHLN

This bit sets the data length.
1 (R/W): 8 bits
0 (R/W): 7 bits

Bit 2 PREN

This bit enables the parity function.
1 (R/W): Enable parity function
0 (R/W): Disable parity function

Bit 1 PRMD

This bit selects either odd parity or even parity when using the parity function.
1 (R/W): Odd parity
0 (R/W): Even parity

Bit 0 STPB

This bit sets the stop bit length.
1 (R/W): 2 bits
0 (R/W): 1 bit

Notes: • The UA*n*MOD register settings can be altered only when the UA*n*CTL.MODEN bit = 0.

• Do not set both the UAnMOD.IRMD and UAnMOD.CAREN bits simultaneously.

UART3 Ch.*n* Baud–Rate Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UA*n*BR	15–12	–	0x0	–	R	–
	11–8	FMD[3:0]	0x0	H0	R/W	
	7–0	BRT[7:0]	0x00	H0	R/W	

Bits 15–12 Reserved

Bits 11–8 FMD[3:0]
Bits 7–0 BRT[7:0]

These bits set the UART3 transfer rate. For more information, refer to "Baud Rate Generator."

Notes: • The UA*n*BR register settings can be altered only when the UA*n*CTL.MODEN bit = 0.

• Do not set the UA*n*BR.FMD[3:0] bits to a value other than 0 to 3 when the UA*n*MOD.BRDIV bit = 1.

UART3 Ch.*n* Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UA*n*CTL	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1	SFTRST	0	H0	R/W	
	0	MODEN	0	H0	R/W	

Bits 15–2 Reserved

Bit 1 SFTRST

This bit issues software reset to the UART3.

1 (W): Issue software reset
0 (W): Ineffective
1 (R): Software reset is executing.
0 (R): Software reset has finished. (During normal operation)

Setting this bit resets the UART3 transmit/receive control circuit and interrupt flags. This bit is automatically cleared after the reset processing has finished.

Bit 0 MODEN

This bit enables the UART3 operations.

1 (R/W): Enable UART3 operations (The operating clock is supplied.)
0 (R/W): Disable UART3 operations (The operating clock is stopped.)

Note: If the UA*n*CTL.MODEN bit is altered from 1 to 0 while sending/receiving data, the data being sent/received cannot be guaranteed. When setting the UA*n*CTL.MODEN bit to 1 again after that, be sure to write 1 to the UA*n*CTL.SFTRST bit as well.

UART3 Ch.*n* Transmit Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UA*n*TXD	15–8	–	0x00	–	R	–
	7–0	TXD[7:0]	0x00	H0	R/W	

Bits 15–8 Reserved

Bits 7–0 TXD[7:0]

Data can be written to the transmit data buffer through these bits. Make sure the UA*n*INTF.TBEIF bit is set to 1 before writing data.

UART3 Ch.*n* Receive Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UA*n*RXD	15–8	–	0x00	–	R	–
	7–0	RXD[7:0]	0x00	H0	R	

Bits 15–8 Reserved

Bits 7–0 RXD[7:0]

The receive data buffer can be read through these bits. The receive data buffer consists of a 2-byte FIFO, and older received data is read first.

UART3 Ch.*n* Status and Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UA*n*INTF	15–10	–	0x00	–	R	–
	9	RBSY	0	H0/S0	R	
	8	TBSY	0	H0/S0	R	
	7	–	0	–	R	
	6	TENDIF	0	H0/S0	R/W	Cleared by writing 1.
	5	FEIF	0	H0/S0	R/W	Cleared by writing 1 or reading the UA*n*RXD register.
	4	PEIF	0	H0/S0	R/W	
	3	OEIF	0	H0/S0	R/W	Cleared by writing 1.
	2	RB2FIF	0	H0/S0	R	Cleared by reading the UA*n*RXD register.
	1	RB1FIF	0	H0/S0	R	
	0	TBEIF	1	H0/S0	R	Cleared by writing to the UA*n*TXD register.

Bits 15–10 Reserved

Bit 9 RBSY

This bit indicates the receiving status. (See Figure 11.5.3.1.)

1 (R): During receiving

0 (R): Idle

Bit 8 TBSY

This bit indicates the sending status. (See Figure 11.5.2.1.)

1 (R): During sending

0 (R): Idle

Bit 7 Reserved

Bit 6 TENDIF

Bit 5 FEIF

Bit 4 PEIF

Bit 3 OEIF

Bit 2 RB2FIF

Bit 1 RB1FIF

Bit 0 TBEIF

These bits indicate the UART3 interrupt cause occurrence status.

1 (R): Cause of interrupt occurred

0 (R): No cause of interrupt occurred

1 (W): Clear flag

0 (W): Ineffective

The following shows the correspondence between the bit and interrupt:

UA*n*INTF.TENDIF bit: End-of-transmission interrupt

UA*n*INTF.FEIF bit: Framing error interrupt

UA*n*INTF.PEIF bit: Parity error interrupt

UA*n*INTF.OEIF bit: Overrun error interrupt

UA*n*INTF.RB2FIF bit: Receive buffer two bytes full interrupt

UA*n*INTF.RB1FIF bit: Receive buffer one byte full interrupt

UA*n*INTF.TBEIF bit: Transmit buffer empty interrupt

UART3 Ch.*n* Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UA*n*INTE	15–8	–	0x00	–	R	–
	7	–	0	–	R	
	6	TENDIE	0	H0	R/W	
	5	FEIE	0	H0	R/W	
	4	PEIE	0	H0	R/W	
	3	OEIE	0	H0	R/W	
	2	RB2FIE	0	H0	R/W	
	1	RB1FIE	0	H0	R/W	
	0	TBEIE	0	H0	R/W	

Bits 15–7 Reserved

Bit 6 TENDIE
Bit 5 FEIE
Bit 4 PEIE
Bit 3 OEIE
Bit 2 RB2FIE
Bit 1 RB1FIE
Bit 0 TBEIE

These bits enable UART3 interrupts.
1 (R/W): Enable interrupts
0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:
UA*n*INTE.TENDIE bit: End-of-transmission interrupt
UA*n*INTE.FEIE bit: Framing error interrupt
UA*n*INTE.PEIE bit: Parity error interrupt
UA*n*INTE.OEIE bit: Overrun error interrupt
UA*n*INTE.RB2FIE bit: Receive buffer two bytes full interrupt
UA*n*INTE.RB1FIE bit: Receive buffer one byte full interrupt
UA*n*INTE.TBEIE bit: Transmit buffer empty interrupt

UART3 Ch.*n* Carrier Waveform Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UA*n*CAWF	15–8	–	0x00	–	R	–
	7–0	CRPER[7:0]	0x00	H0	R/W	

Bits 15–8 Reserved

Bits 7–0 CRPER[7:0]

These bits set the carrier modulation output frequency. For more information, refer to "Carrier Modulation."

12 Synchronous Serial Interface (SPIA)

12.1 Overview

SPIA is a synchronous serial interface. The features of SPIA are listed below.

- Supports both master and slave modes.
- Data length: 2 to 16 bits programmable
- Either MSB first or LSB first can be selected for the data format.
- Clock phase and polarity are configurable.
- Supports full-duplex communications.
- Includes separated transmit data buffer and receive data buffer registers.
- Can generate receive buffer full, transmit buffer empty, end of transmission, and overrun interrupts.
- Master mode allows use of a 16-bit timer to set baud rate.
- Slave mode is capable of being operated with the external input clock SPICLK*n* only.
- Slave mode is capable of being operated in SLEEP mode allowing wake-up by an SPIA interrupt.
- Input pins can be pulled up/down with an internal resistor.

Figure 12.1.1 shows the SPIA configuration.

Table 12.1.1 SPIA Channel Configuration of S1C17M02/M03

Item	S1C17M02	S1C17M03
Number of channels	1 channel (Ch.0)	
Internal clock input	Ch.0 ← 16-bit timer Ch.1	

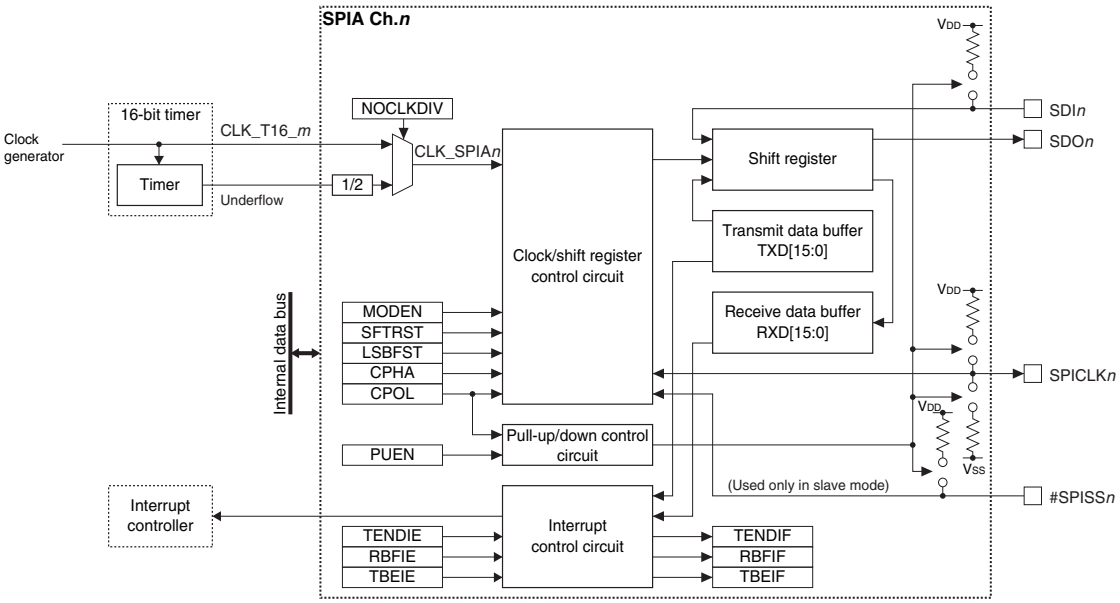


Figure 12.1.1 SPIA Configuration

12.2 Input/Output Pins and External Connections

12.2.1 List of Input/Output Pins

Table 12.2.1.1 lists the SPIA pins.

Table 12.2.1.1 List of SPIA Pins

Pin name	I/O*	Initial status*	Function
SDI*n*	I	I (Hi-Z)	SPIA Ch.*n* data input pin
SDO*n*	O or Hi-Z	Hi-Z	SPIA Ch.*n* data output pin
SPICLK*n*	I or O	I (Hi-Z)	SPIA Ch.*n* external clock input/output pin
#SPISS*n*	I	I (Hi-Z)	SPIA Ch.*n* slave select signal input pin

* Indicates the status when the pin is configured for SPIA.

If the port is shared with the SPIA pin and other functions, the SPIA input/output function must be assigned to the port before activating SPIA. For more information, refer to the "I/O Ports" chapter.

12.2.2 External Connections

SPIA operates in master mode or slave mode. Figures 12.2.2.1 and 12.2.2.2 show connection diagrams between SPIA in each mode and external SPI devices.

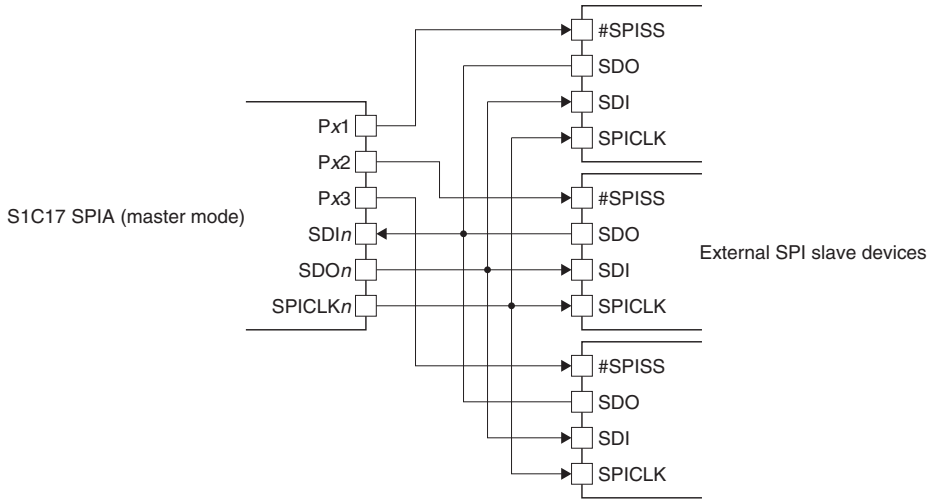


Figure 12.2.2.1 Connections between SPIA in Master Mode and External SPI Slave Devices

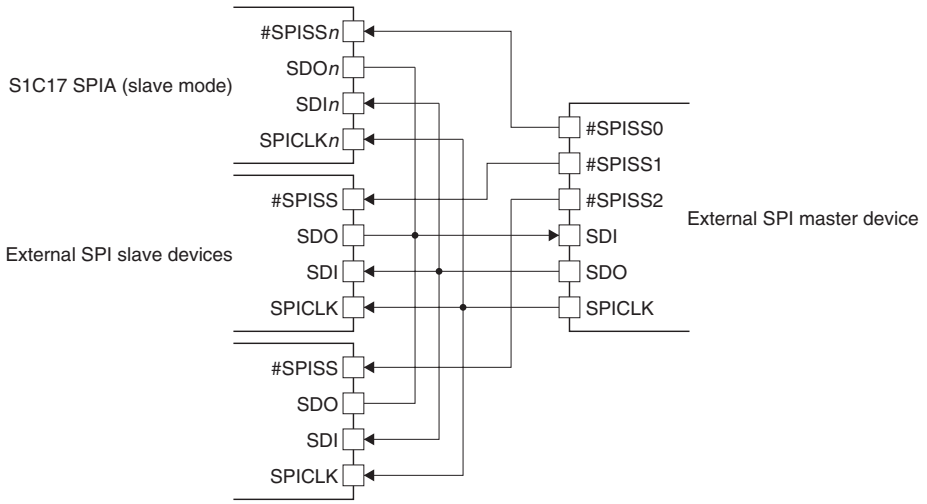


Figure 12.2.2.2 Connections between SPIA in Slave Mode and External SPI Master Device

12.2.3 Pin Functions in Master Mode and Slave Mode

The pin functions are changed according to the master or slave mode selection. The differences in pin functions between the modes are shown in Table 12.2.3.1.

Table 12.2.3.1 Pin Function Differences between Modes

Pin	Function in master mode	Function in slave mode
SDI*n*	Always placed into input state.	
SDO*n*	Always placed into output state.	This pin is placed into output state while a low level is applied to the #SPISS*n* pin or placed into Hi-Z state while a high level is applied to the #SPISS*n* pin.
SPICLK*n*	Outputs the SPI clock to external devices. Output clock polarity and phase can be configured if necessary.	Inputs an external SPI clock. Clock polarity and phase can be designated according to the input clock.
#SPISS*n*	Not used. This input function is not required to be assigned to the port. To output the slave select signal in master mode, use a general-purpose I/O port function.	Applying a low level to the #SPISS*n* pin enables SPIA to transmit/receive data. While a high level is applied to this pin, SPIA is not selected as a slave device. Data input to the SDI*n* pin and the clock input to the SPICLK*n* pin are ignored. When a high level is applied, the transmit/receive bit count is cleared to 0 and the already received bits are discarded.

12.2.4 Input Pin Pull-Up/Pull-Down Function

The SPIA input pins (SDI*n* in master mode or SDI*n*, SPICLK*n*, and #SPISS*n* pins in slave mode) have a pull-up or pull-down function as shown in Table 12.2.4.1. This function is enabled by setting the SPI*n*MOD.PUEN bit to 1.

Table 12.2.4.1 Pull-Up or Pull-Down of Input Pins

Pin	Master mode	Slave mode
SDI*n*	Pull-up	Pull-up
SPICLK*n*	–	SPI*n*MOD.CPOL bit = 1: Pull-up SPI*n*MOD.CPOL bit = 0: Pull-down
#SPISS*n*	–	Pull-up

12.3 Clock Settings

12.3.1 SPIA Operating Clock

Operating clock in master mode

In master mode, the SPIA operating clock is supplied from the 16-bit timer. The following two options are provided for the clock configuration.

Use the 16-bit timer operating clock without dividing

By setting the SPI*n*MOD.NOCLKDIV bit to 1, the operating clock CLK_T16_*m*, which is configured by selecting a clock source and a division ratio, for the 16-bit timer channel corresponding to the SPIA channel is input to SPIA as CLK_SPIA*n*. Since this clock is also used as the SPI clock SPICLK*n* without changing, the CLK_SPIA*n* frequency becomes the baud rate.

To supply CLK_SPIA*n* to SPIA, the 16-bit timer clock source must be enabled in the clock generator. It does not matter how the T16_*m*CTL.MODEN and T16_*m*CTL.PRUN bits of the corresponding 16-bit timer channel are set (1 or 0).

When setting this mode, the timer function of the corresponding 16-bit timer channel may be used for another purpose.

Use the 16-bit timer as a baud rate generator

By setting the SPI*n*MOD.NOCLKDIV bit to 0, SPIA inputs the underflow signal generated by the corresponding 16-bit timer channel and converts it to the SPICLK*n*. The 16-bit timer must be run with an appropriate reload data set. The SPICLK*n* frequency (baud rate) and the 16-bit timer reload data are calculated by the equations shown below.

$$f_{SPICLK} = \frac{f_{CLK\_SPIA}}{2 \times (RLD + 1)} \qquad RLD = \frac{f_{CLK\_SPIA}}{f_{SPICLK} \times 2} - 1 \qquad \text{(Eq. 12.1)}$$

Where

f_{SPICLK}: SPICLK*n* frequency [Hz] (= baud rate [bps])
f_{CLK_SPIA}: SPIA operating clock frequency [Hz]
RLD: 16-bit timer reload data value

For controlling the 16-bit timer, refer to the "16-bit Timers" chapter.

Operating clock in slave mode

SPIA set in slave mode operates with the clock supplied from the external SPI master to the SPICLK*n* pin. The 16-bit timer channel (including the clock source selector and the divider) corresponding to the SPIA channel is not used. Furthermore, the SPI*n*MOD.NOCLKDIV bit setting becomes ineffective.

SPIA keeps operating using the clock supplied from the external SPI master even if all the internal clocks halt during SLEEP mode, so SPIA can receive data and can generate receive buffer full interrupts.

12.3.2 Clock Supply in DEBUG Mode

In master mode, the operating clock supply during DEBUG mode should be controlled using the T16_*m*CLK.DBRUN bit.

The CLK_T16_*m* supply to SPIA Ch.*n* is suspended when the CPU enters DEBUG mode if the T16_*m*CLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_T16_*m* supply resumes. Although SPIA Ch.*n* stops operating when the CLK_T16_*m* supply is suspended, the output pins and registers retain the status before DEBUG mode was entered. If the T16_*m*CLK.DBRUN bit = 1, the CLK_T16_*m* supply is not suspended and SPIA Ch.*n* will keep operating in DEBUG mode.

SPIA in slave mode operates with the external SPI master clock input from the SPICLK*n* pin regardless of whether the CPU is placed into DEBUG mode or normal mode.

12.3.3 SPI Clock (SPICLK*n*) Phase and Polarity

The SPICLK*n* phase and polarity can be configured separately using the SPI*n*MOD.CPHA bit and the SPI*n*MOD.CPOL bit, respectively. Figure 12.3.3.1 shows the clock waveform and data input/output timing in each setting.

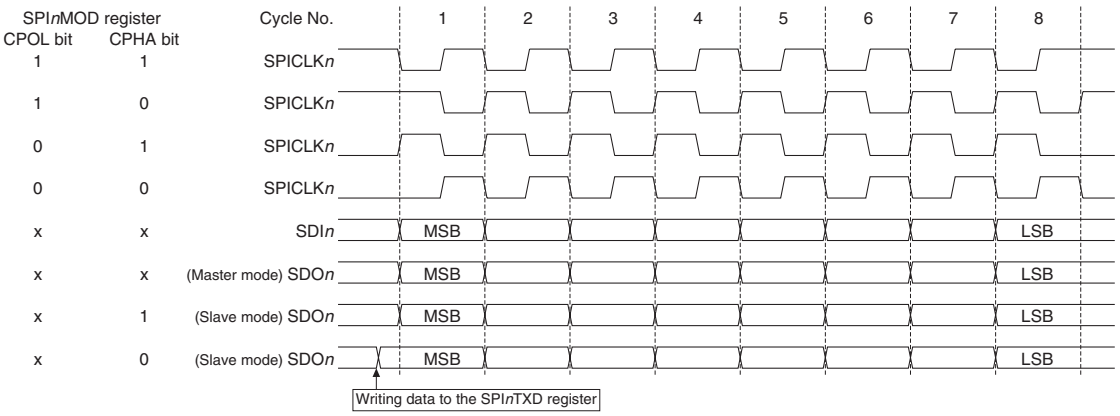


Figure 12.3.3.1 SPI Clock Phase and Polarity (SPI*n*MOD.LSBFST bit = 0, SPI*n*MOD.CHLN[3:0] bits = 0x7)

12.4 Data Format

The SPIA data length can be selected from 2 bits to 16 bits by setting the SPI*n*MOD.CHLN[3:0] bits. The input/output permutation is configurable to MSB first or LSB first using the SPI*n*MOD.LSBFST bit. Figure 12.4.1 shows a data format example when the SPI*n*MOD.CHLN[3:0] bits = 0x7, the SPI*n*MOD.CPOL bit = 0 and the SPI*n*MOD.CPHA bit = 0.

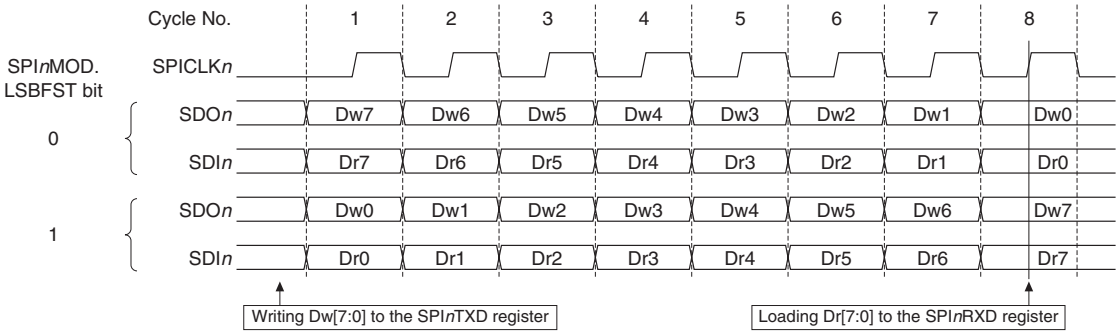


Figure 12.4.1 Data Format Selection Using the SPI*n*MOD.LSBFST Bit
(SPI*n*MOD.CHLN[3:0] bits = 0x7, SPI*n*MOD.CPOL bit = 0, SPI*n*MOD.CPHA bit = 0)

12.5 Operations

12.5.1 Initialization

SPIA Ch.*n* should be initialized with the procedure shown below.

1. <Master mode only> Generate a clock by controlling the 16-bit timer and supply it to SPIA Ch.*n*.
2. Configure the following SPI*n*MOD register bits:
   - SPI*n*MOD.PUEN bit (Enable input pin pull-up/down)
   - SPI*n*MOD.NOCLKDIV bit (Select master mode operating clock)
   - SPI*n*MOD.LSBFST bit (Select MSB first/LSB first)
   - SPI*n*MOD.CPHA bit (Select clock phase)
   - SPI*n*MOD.CPOL bit (Select clock polarity)
   - SPI*n*MOD.MST bit (Select master/slave mode)
3. Assign the SPIA Ch.*n* input/output function to the ports. (Refer to the "I/O Ports" chapter.)
4. Set the following SPI*n*CTL register bits:
   - Set the SPI*n*CTL.SFTRST bit to 1. (Execute software reset)
   - Set the SPI*n*CTL.MODEN bit to 1. (Enable SPIA Ch.*n* operations)
5. Set the following bits when using the interrupt:
   - Write 1 to the interrupt flags in the SPI*n*INTF register. (Clear interrupt flags)
   - Set the interrupt enable bits in the SPI*n*INTE register to 1. * (Enable interrupts)

   * The initial value of the SPI*n*INTF.TBEIF bit is 1, therefore, an interrupt will occur immediately after the SPI*n*INTE.TBEIE bit is set to 1.

12.5.2 Data Transmission in Master Mode

A data sending procedure and operations in master mode are shown below. Figures 12.5.2.1 and 12.5.2.2 show a timing chart and a flowchart, respectively.

Data sending procedure

1. Assert the slave select signal by controlling the general-purpose output port (if necessary).
2. Check to see if the SPI*n*INTF.TBEIF bit is set to 1 (transmit buffer empty).
3. Write transmit data to the SPI*n*TXD register.

4. Wait for an SPIA interrupt when using the interrupt.
5. Repeat Steps 2 to 4 (or 2 and 3) until the end of transmit data.
6. Negate the slave select signal by controlling the general-purpose output port (if necessary).

Data sending operations

SPIA Ch.*n* starts data sending operations when transmit data is written to the SPI*n*TXD register.
The transmit data in the SPI*n*TXD register is automatically transferred to the shift register and the SPI*n*INTF.TBEIF bit is set to 1. If the SPI*n*INTE.TBEIE bit = 1 (transmit buffer empty interrupt enabled), a transmit buffer empty interrupt occurs at the same time.
The SPICLK*n* pin outputs clocks of the number of the bits specified by the SPI*n*MOD.CHLN[3:0] bits and the transmit data bits are output in sequence from the SDO*n* pin in sync with these clocks.

Even if the clock is being output from the SPICLK*n* pin, the next transmit data can be written to the SPI*n*TXD register after making sure the SPI*n*INTF.TBEIF bit is set to 1.

If transmit data has not been written to the SPI*n*TXD register after the last clock is output from the SPICLK*n* pin, the clock output halts and the SPI*n*INTF.TENDIF bit is set to 1. At the same time SPIA issues an end-of-transmission interrupt request if the SPI*n*INTE.TENDIE bit = 1.

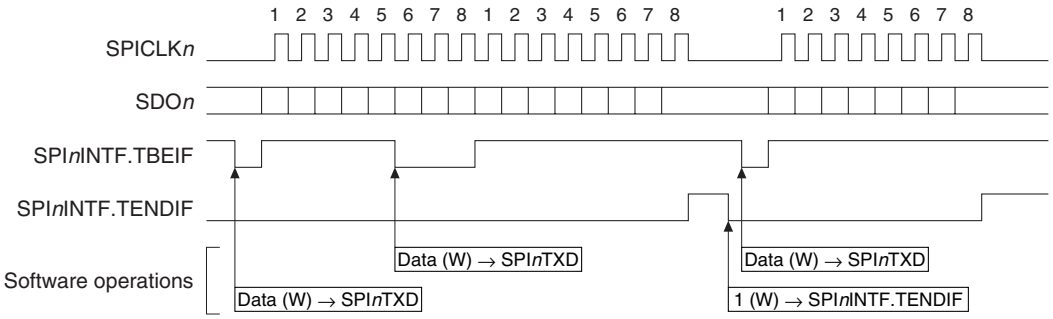


Figure 12.5.2.1 Example of Data Sending Operations in Master Mode (SPI*n*MOD.CHLN[3:0] bits = 0x7)

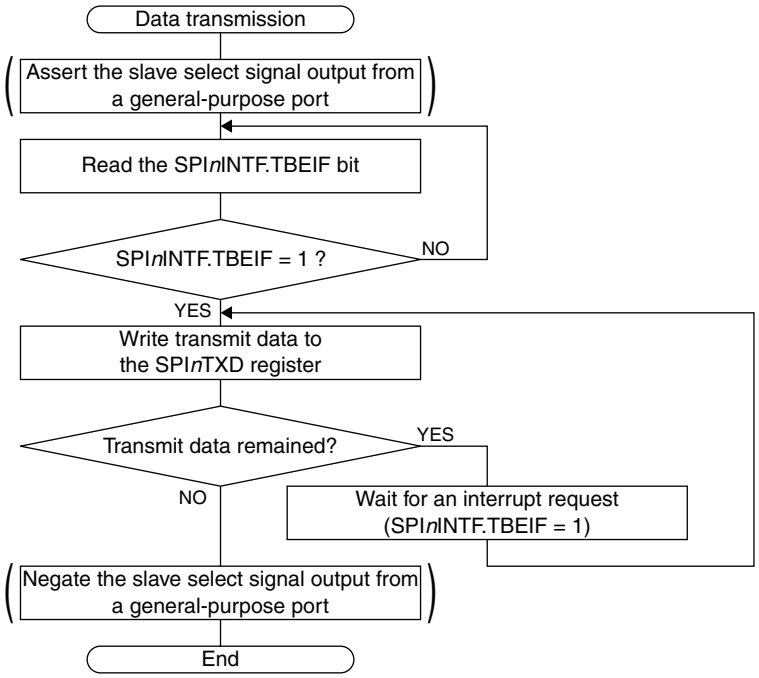


Figure 12.5.2.2 Data Transmission Flowchart in Master Mode

12.5.3 Data Reception in Master Mode

A data receiving procedure and operations in master mode are shown below. Figures 12.5.3.1 and 12.5.3.2 show a timing chart and flowcharts, respectively.

Data receiving procedure

1. Assert the slave select signal by controlling the general-purpose output port (if necessary).
2. Check to see if the SPI*n*INTF.TBEIF bit is set to 1 (transmit buffer empty).
3. Write dummy data (or transmit data) to the SPI*n*TXD register.
4. Wait for a transmit buffer empty interrupt (SPI*n*INTF.TBEIF bit = 1).
5. Write dummy data (or transmit data) to the SPI*n*TXD register.
6. Wait for a receive buffer full interrupt (SPI*n*INTF.RBFIF bit = 1).
7. Read the received data from the SPI*n*RXD register.
8. Repeat Steps 5 to 7 until the end of data reception.
9. Negate the slave select signal by controlling the general-purpose output port (if necessary).

Note: To perform continuous data reception without stopping SPICLK*n*, Steps 7 and 5 operations must be completed within the SPICLK*n* cycles equivalent to "Data bit length - 1" after Step 6.

Data receiving operations

SPIA Ch.*n* starts data receiving operations simultaneously with data sending operations when transmit data (may be dummy data if data transmission is not required) is written to the SPI*n*TXD register.
The SPICLK*n* pin outputs clocks of the number of the bits specified by the SPI*n*MOD.CHLN[3:0] bits. The transmit data bits are output in sequence from the SDO*n* pin in sync with these clocks and the receive data bits input from the SDI*n* pin are shifted into the shift register.
When the last clock is output from the SPICLK*n* pin and receive data bits are all shifted into the shift register, the received data is transferred to the receive data buffer and the SPI*n*INTF.RBFIF bit is set to 1. At the same time SPIA issues a receive buffer full interrupt request if the SPI*n*INTE.RBFIE bit = 1. After that, the received data in the receive data buffer can be read through the SPI*n*RXD register.

Note: If data of the number of the bits specified by the SPI*n*MOD.CHLN[3:0] bits is received when the SPI*n*INTF.RBFIF bit is set to 1, the SPI*n*RXD register is overwritten with the newly received data and the previously received data is lost. In this case, the SPI*n*INTF.OEIF bit is set.

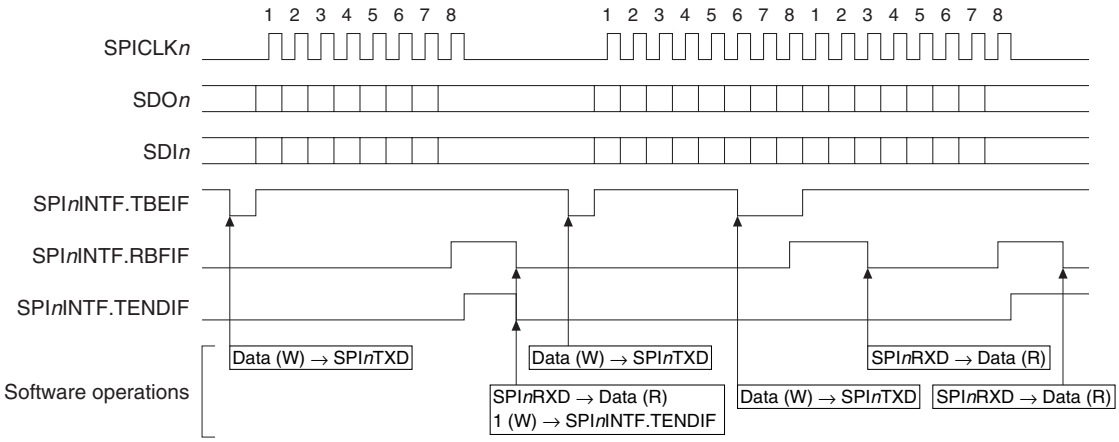


Figure 12.5.3.1 Example of Data Receiving Operations in Master Mode (SPI*n*MOD.CHLN[3:0] bits = 0x7)

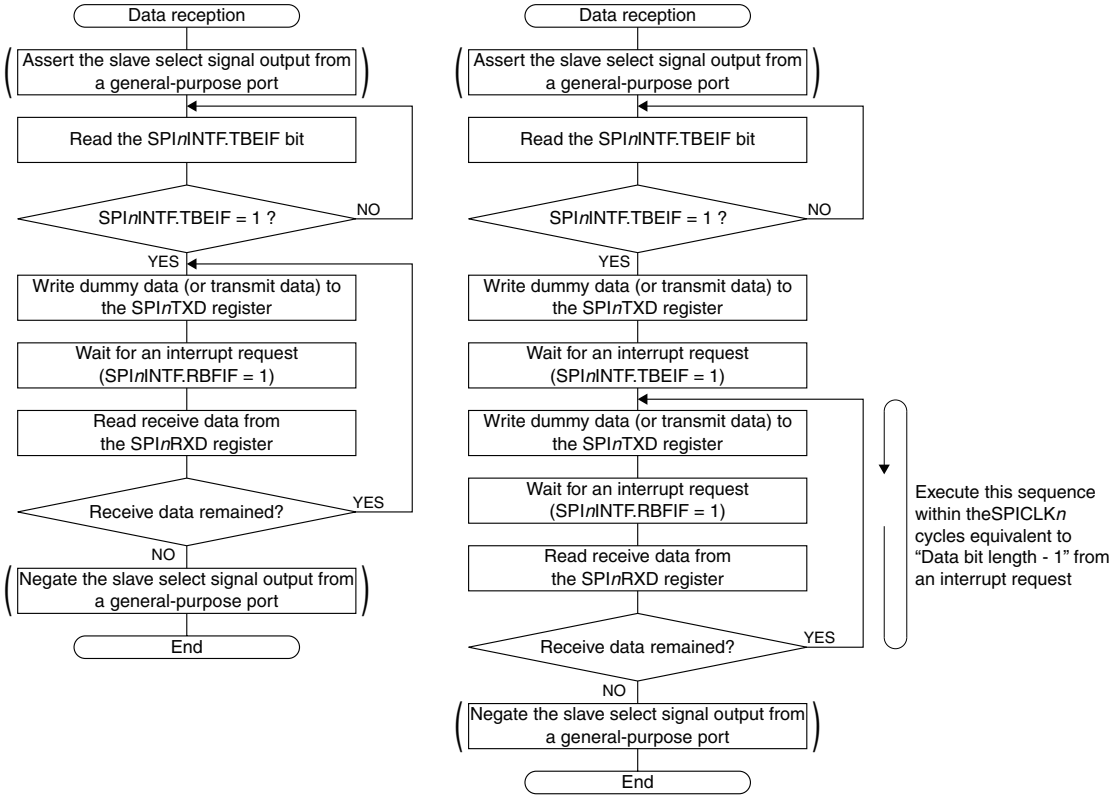


Figure 12.5.3.2 Data Reception Flowcharts in Master Mode

12.5.4 Terminating Data Transfer in Master Mode

A procedure to terminate data transfer in master mode is shown below.

1. Wait for an end-of-transmission interrupt (SPI*n*INTF.TENDIF bit = 1).
2. Set the SPI*n*CTL.MODEN bit to 0 to disable the SPIA Ch.*n* operations.
3. Stop the 16-bit timer to disable the clock supply to SPIA Ch.*n*.

12.5.5 Data Transfer in Slave Mode

A data sending/receiving procedure and operations in slave mode are shown below. Figures 12.5.5.1 and 12.5.5.2 show a timing chart and flowcharts, respectively.

Data sending procedure

1. Check to see if the SPI*n*INTF.TBEIF bit is set to 1 (transmit buffer empty).
2. Write transmit data to the SPI*n*TXD register.
3. Wait for a transmit buffer empty interrupt (SPI*n*INTF.TBEIF bit = 1).
4. Repeat Steps 2 and 3 until the end of transmit data.

Note: Transmit data must be written to the SPI*n*TXD register after the SPI*n*INTF.TBEIF bit is set to 1 by the time the sending SPI*n*TXD register data written is completed. If no transmit data is written during this period, the data bits input from the SDI*n* pin are shifted and output from the SDO*n* pin without being modified.

Data receiving procedure

1. Wait for a receive buffer full interrupt (SPI*n*INTF.RBFIF bit = 1).
2. Read the received data from the SPI*n*RXD register.
3. Repeat Steps 1 and 2 until the end of data reception.

Data transfer operations

The following shows the slave mode operations different from master mode:

- Slave mode operates with the SPI clock supplied from the external SPI master to the SPICLK*n* pin.
  The data transfer rate is determined by the SPICLK*n* frequency. It is not necessary to control the 16-bit timer.
- SPIA can operate as a slave device only when the slave select signal input from the external SPI master to the #SPISS*n* pin is set to the active (low) level.
  If #SPISS*n* = high, the software transfer control, the SPICLK*n* pin input, and the SDI*n* pin input are all ineffective. If the #SPISS*n* signal goes high during data transfer, the transfer bit counter is cleared and data in the shift register is discarded.
- Slave mode starts data transfer when SPICLK*n* is input from the external SPI master after the #SPISS*n* signal is asserted. Writing transmit data is not a trigger to start data transfer. Therefore, it is not necessary to write dummy data to the transmit data buffer when performing data reception only.
- Data transmission/reception can be performed even in SLEEP mode, it makes it possible to wake the CPU up using an SPIA interrupt.

Other operations are the same as master mode.

Notes: • If data of the number of bits specified by the SPI*n*MOD.CHLN[3:0] bits is received when the SPI*n*INTF.RBFIF bit is set to 1, the SPI*n*RXD register is overwritten with the newly received data and the previously received data is lost. In this case, the SPI*n*INTF.OEIF bit is set.

• When the clock for the first bit is input from the SPICLK*n* pin, SPIA starts sending the data currently stored in the shift register even if the SPI*n*INTF.TBEIF bit is set to 1.

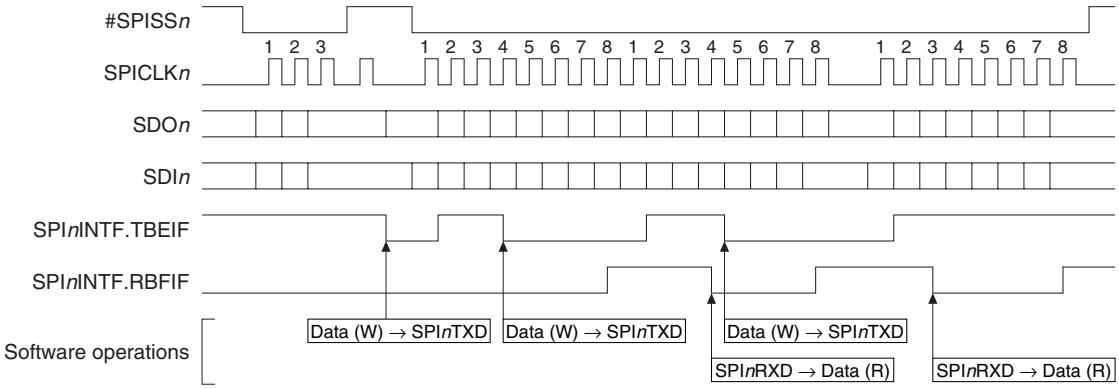


Figure 12.5.5.1 Example of Data Transfer Operations in Slave Mode (SPI*n*MOD.CHLN[3:0] bits = 0x7)

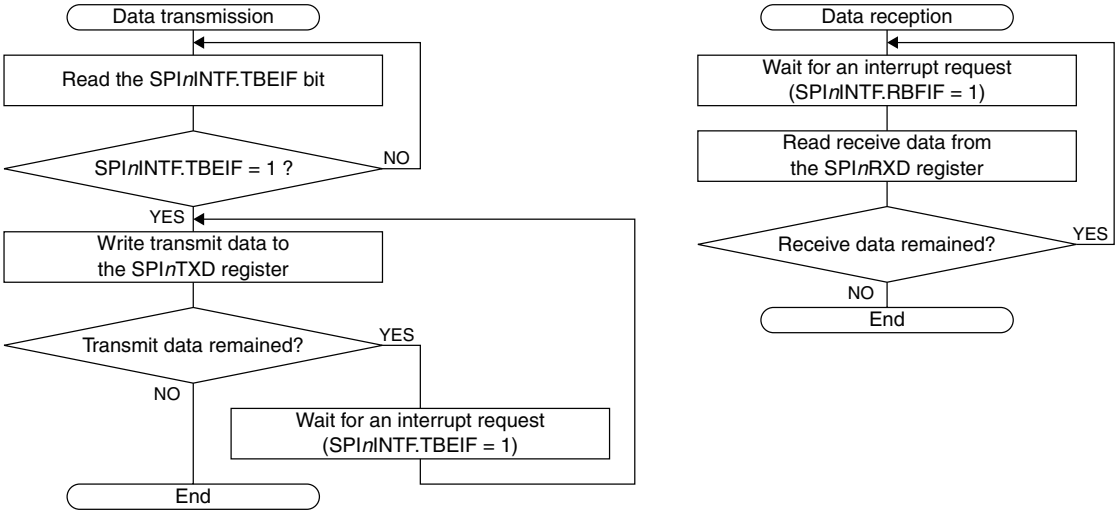


Figure 12.5.5.2 Data Transfer Flowcharts in Slave Mode

12.5.6 Terminating Data Transfer in Slave Mode

A procedure to terminate data transfer in slave mode is shown below.

1. Wait for an end-of-transmission interrupt (SPI*n*INTF.TENDIF bit = 1). Or determine end of transfer via the received data.
2. Set the SPI*n*CTL.MODEN bit to 0 to disable the SPIA Ch.*n* operations.

12.6 Interrupts

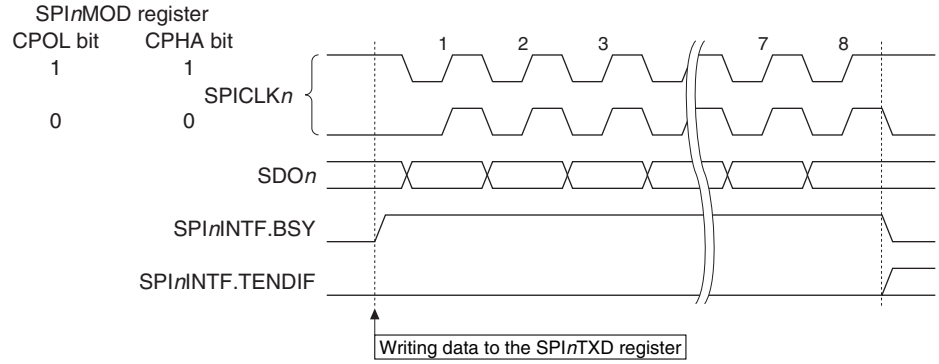
SPIA has a function to generate the interrupts shown in Table 12.6.1.

Table 12.6.1 SPIA Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
End of transmission	SPI*n*INTF.TENDIF	When the SPI*n*INTF.TBEIF bit = 1 after data of the specified bit length (defined by the SPI*n*MOD.CHLN[3:0] bits) has been sent	Writing 1
Receive buffer full	SPI*n*INTF.RBFIF	When data of the specified bit length is received and the received data is transferred from the shift register to the received data buffer	Reading the SPI*n*RXD register
Transmit buffer empty	SPI*n*INTF.TBEIF	When transmit data written to the transmit data buffer is transferred to the shift register	Writing to the SPI*n*TXD register
Overrun error	SPI*n*INTF.OEIF	When the receive data buffer is full (when the received data has not been read) at the point that receiving data to the shift register has completed	Writing 1

SPIA provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the "Interrupt Controller" chapter.
The SPI*n*INTF register also contains the BSY bit that indicates the SPIA operating status.
Figure 12.6.1 shows the SPI*n*INTF.BSY and SPI*n*INTF.TENDIF bit set timings.

Master mode



Slave mode

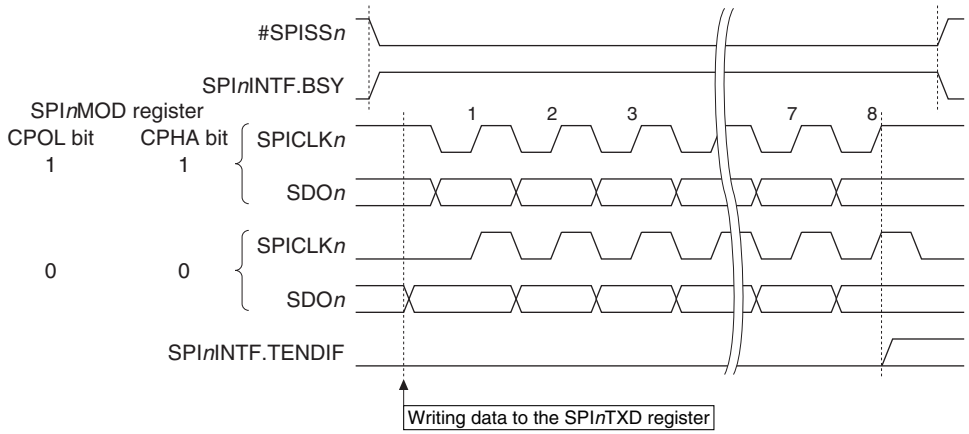


Figure 12.6.1 SPI*n*INTF.BSY and SPI*n*INTF.TENDIF Bit Set Timings (when SPI*n*MOD.CHLN[3:0] bits = 0x7)

12.7 Control Registers

SPIA Ch.*n* Mode Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPI*n*MOD	15–12	–	0x0	–	R	–
	11–8	CHLN[3:0]	0x7	H0	R/W	
	7–6	–	0x0	–	R	
	5	PUEN	0	H0	R/W	
	4	NOCLKDIV	0	H0	R/W	
	3	LSBFST	0	H0	R/W	
	2	CPHA	0	H0	R/W	
	1	CPOL	0	H0	R/W	
	0	MST	0	H0	R/W	

Bits 15–12 Reserved

Bits 11–8 CHLN[3:0]

These bits set the bit length of transfer data.

Table 12.7.1 Data Bit Length Settings

SPI*n*MOD.CHLN[3:0] bits	Data bit length
0xf	16 bits
0xe	15 bits
0xd	14 bits
0xc	13 bits
0xb	12 bits
0xa	11 bits
0x9	10 bits
0x8	9 bits
0x7	8 bits
0x6	7 bits
0x5	6 bits
0x4	5 bits
0x3	4 bits
0x2	3 bits
0x1	2 bits
0x0	Setting prohibited

Bits 7–6 Reserved

Bit 5 PUEN

This bit enables pull-up/down of the input pins.
1 (R/W): Enable pull-up/down
0 (R/W): Disable pull-up/down

For more information, refer to "Input Pin Pull-Up/Pull-Down Function."

Bit 4 NOCLKDIV

This bit selects SPICLK*n* in master mode. This setting is ineffective in slave mode.
1 (R/W): SPICLK*n* frequency = CLK_SPIA*n* frequency (= 16-bit timer operating clock frequency)
0 (R/W): SPICLK*n* frequency = 16-bit timer output frequency / 2

For more information, refer to "SPIA Operating Clock."

Bit 3 LSBFST

This bit configures the data format (input/output permutation).
1 (R/W): LSB first
0 (R/W): MSB first

Bit 2 CPHA
Bit 1 CPOL

These bits set the SPI clock phase and polarity. For more information, refer to "SPI Clock (SPICLK*n*) Phase and Polarity."

Bit 0 MST

This bit sets the SPIA operating mode (master mode or slave mode).
1 (R/W): Master mode
0 (R/W): Slave mode

Note: The SPI*n*MOD register settings can be altered only when the SPI*n*CTL.MODEN bit = 0.

SPIA Ch.*n* Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPI*n*CTL	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1	SFTRST	0	H0	R/W	
	0	MODEN	0	H0	R/W	

Bits 15–2 Reserved

Bit 1 SFTRST

This bit issues software reset to SPIA.

1 (W): Issue software reset
0 (W): Ineffective
1 (R): Software reset is executing.
0 (R): Software reset has finished. (During normal operation)

Setting this bit resets the SPIA shift register and transfer bit counter. This bit is automatically cleared after the reset processing has finished.

Bit 0 MODEN

This bit enables the SPIA operations.

1 (R/W): Enable SPIA operations (In master mode, the operating clock is supplied.)
0 (R/W): Disable SPIA operations (In master mode, the operating clock is stopped.)

Note: If the SPI*n*CTL.MODEN bit is altered from 1 to 0 while sending/receiving data, the data being sent/received cannot be guaranteed. When setting the SPI*n*CTL.MODEN bit to 1 again after that, be sure to write 1 to the SPI*n*CTL.SFTRST bit as well.

SPIA Ch.*n* Transmit Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPI*n*TXD	15–0	TXD[15:0]	0x0000	H0	R/W	–

Bits 15–0 TXD[15:0]

Data can be written to the transmit data buffer through these bits.
In master mode, writing to these bits starts data transfer.
Transmit data can be written when the SPI*n*INTF.TBEIF bit = 1 regardless of whether data is being output from the SDO*n* pin or not.
Note that the upper data bits that exceed the data bit length configured by the SPI*n*MOD.CHLN[3:0] bits will not be output from the SDO*n* pin.

Note: Be sure to avoid writing to the SPI*n*TXD register when the SPI*n*INTF.TBEIF bit = 0. Otherwise, transfer data cannot be guaranteed.

SPIA Ch.*n* Receive Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPI*n*RXD	15–0	RXD[15:0]	0x0000	H0	R	–

Bits 15–0 RXD[15:0]

The receive data buffer can be read through these bits. Received data can be read when the SPI*n*INTF.RBFIF bit = 1 regardless of whether data is being input from the SDI*n* pin or not. Note that the upper bits that exceed the data bit length configured by the SPI*n*MOD.CHLN[3:0] bits become 0.

Note: The SPI*n*RXD.RXD[15:0] bits are cleared to 0x0000 when 1 is written to the SPI*n*CTL.MODEN bit or the SPI*n*CTL.SFTRST bit.

SPIA Ch.*n* Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPI*n*INTF	15–8	–	0x00	–	R	–
	7	BSY	0	H0	R	
	6–4	–	0x0	–	R	
	3	OEIF	0	H0/S0	R/W	Cleared by writing 1.
	2	TENDIF	0	H0/S0	R/W	
	1	RBFIF	0	H0/S0	R	Cleared by reading the SPI*n*RXD register.
	0	TBEIF	1	H0/S0	R	Cleared by writing to the SPI*n*TXD register.

Bits 15–8 Reserved

Bit 7 BSY

This bit indicates the SPIA operating status.

1 (R): Transmit/receive busy (master mode), #SPISS*n* = Low level (slave mode)

0 (R): Idle

Bits 6–4 Reserved

Bit 3 OEIF

Bit 2 TENDIF

Bit 1 RBFIF

Bit 0 TBEIF

These bits indicate the SPIA interrupt cause occurrence status.

1 (R): Cause of interrupt occurred

0 (R): No cause of interrupt occurred

1 (W): Clear flag (OEIF, TENDIF)

0 (W): Ineffective

The following shows the correspondence between the bit and interrupt:

SPI*n*INTF.OEIF bit: Overrun error interrupt

SPI*n*INTF.TENDIF bit: End-of-transmission interrupt

SPI*n*INTF.RBFIF bit: Receive buffer full interrupt

SPI*n*INTF.TBEIF bit: Transmit buffer empty interrupt

SPIA Ch.*n* Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPI*n*INTE	15–8	–	0x00	–	R	–
	7–4	–	0x0	–	R	
	3	OEIE	0	H0	R/W	
	2	TENDIE	0	H0	R/W	
	1	RBFIE	0	H0	R/W	
	0	TBEIE	0	H0	R/W	

Bits 15–4 Reserved

Bit 3 OEIE

Bit 2 TENDIE

Bit 1 RBFIE

Bit 0 TBEIE

These bits enable SPIA interrupts.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:

SPI*n*INTE.OEIE bit: Overrun error interrupt

SPI*n*INTE.TENDIE bit: End-of-transmission interrupt

SPI*n*INTE.RBFIE bit: Receive buffer full interrupt

SPI*n*INTE.TBEIE bit: Transmit buffer empty interrupt

13 I²C (I2C)

13.1 Overview

The I2C is a subset of the I²C bus interface. The features of the I2C are listed below.

- Functions as an I²C bus master (single master) or a slave device.
- Supports standard mode (up to 100 kbit/s) and fast mode (up to 400 kbit/s).
- Supports 7-bit and 10-bit address modes.
- Supports clock stretching.
- Includes a baud rate generator for generating the clock in master mode.
- No clock source is required to run the I2C in slave mode, as it can run with the I²C bus signals only.
- Slave mode is capable of being operated in SLEEP mode allowing wake-up by an interrupt when an address match is detected.
- Master mode supports automatic bus clear sending function.
- Can generate receive buffer full, transmit buffer empty, and other interrupts.
- The input filter for the SDA and SCL inputs does not comply with the standard for removing noise spikes less than 50 ns.

Figure 13.1.1 shows the I2C configuration.

Table 13.1.1 I2C Channel Configuration of S1C17M02/M03

Item	S1C17M02	S1C17M03
Number of channels	1 channel (Ch.0)	

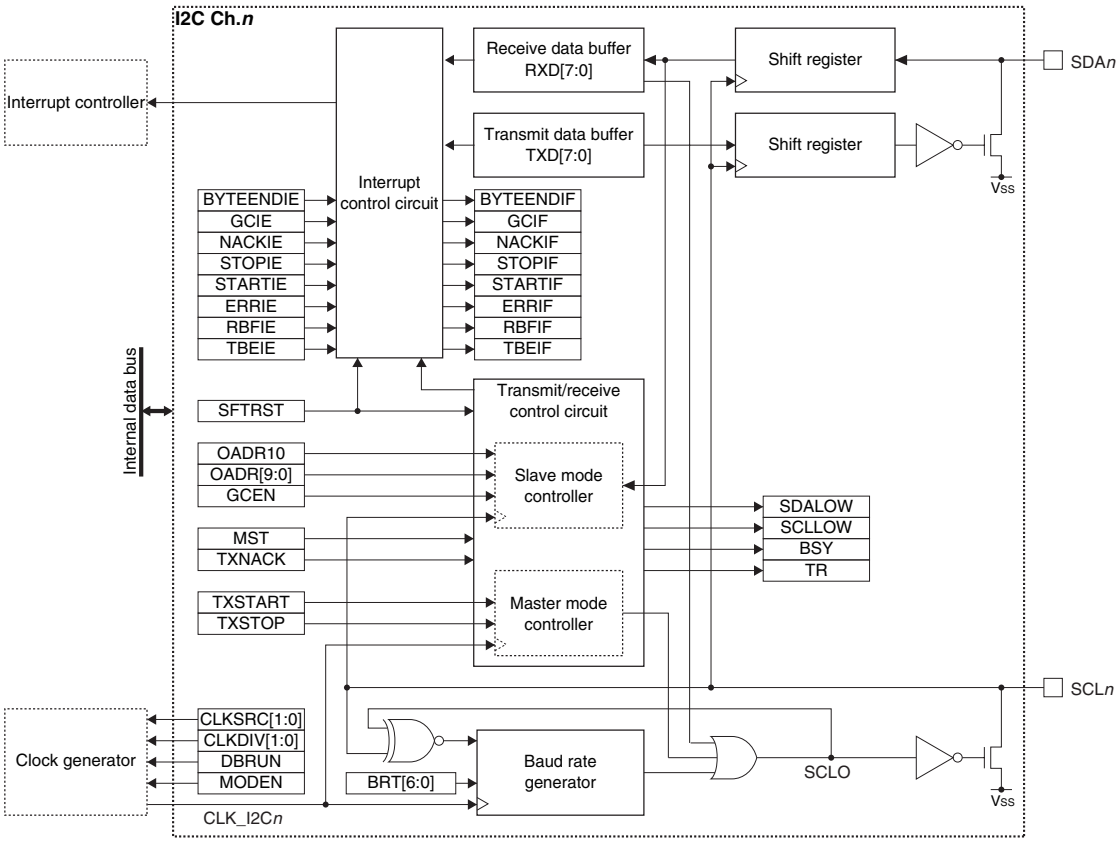


Figure 13.1.1 I2C Configuration

13.2 Input/Output Pins and External Connections

13.2.1 List of Input/Output Pins

Table 13.2.1.1 lists the I2C pins.

Table 13.2.1.1 List of I2C Pins

Pin name	I/O*	Initial status*	Function
SDA*n*	I/O	I	I2C bus serial data input/output pin
SCL*n*	I/O	I	I2C bus clock input/output pin

* Indicates the status when the pin is configured for the I2C.

If the port is shared with the I2C pin and other functions, the I2C input/output function must be assigned to the port before activating the I2C. For more information, refer to the "I/O Ports" chapter.

13.2.2 External Connections

Figure 13.2.2.1 shows a connection diagram between the I2C in this IC and external I2C devices.
The serial data (SDA) and serial clock (SCL) lines must be pulled up with an external resistor.
When the I2C is set into master mode, one or more slave devices that have a unique address may be connected to the I2C bus. When the I2C is set into slave mode, one or more master and slave devices that have a unique address may be connected to the I2C bus.

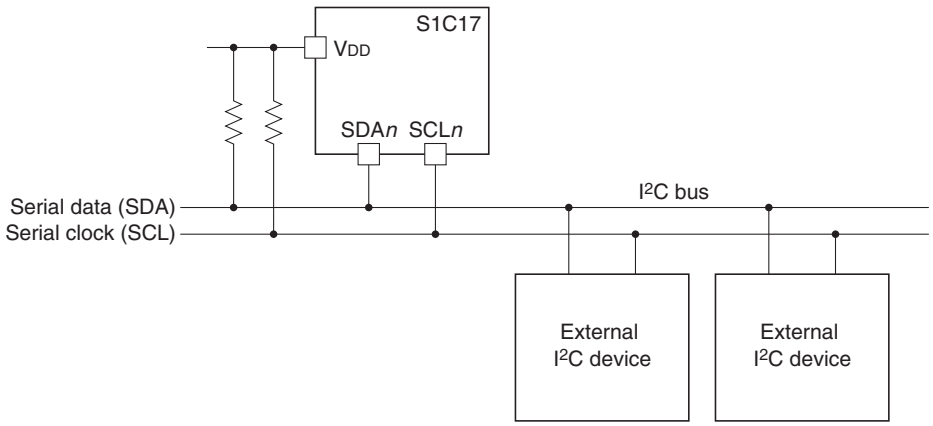


Figure 13.2.2.1 Connections between I2C and External I2C Devices

Notes: • The SDA and SCL lines must be pulled up to a VDD of this IC or lower voltage. However, if the I2C input/output ports are configured with the over voltage tolerant fail-safe type I/O, these lines can be pulled up to a voltage exceeding the VDD of this IC but within the recommended operating voltage range of this IC.

• The internal pull-up resistors for the I/O ports cannot be used for pulling up SDA and SCL.

• When the I2C is set into master mode, no other master device can be connected to the I2C bus.

13.3 Clock Settings

13.3.1 I2C Operating Clock

Master mode operating clock

When using the I2C Ch.*n* in master mode, the I2C Ch.*n* operating clock CLK_I2C*n* must be supplied to the I2C Ch.*n* from the clock generator. The CLK_I2C*n* supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to "Clock Generator" in the "Power Supply, Reset, and Clocks" chapter).
2. Set the following I2C*n*CLK register bits:
   - I2C*n*CLK.CLKSRC[1:0] bits (Clock source selection)
   - I2C*n*CLK.CLKDIV[1:0] bits (Clock division ratio selection = Clock frequency setting)

   When using the I2C in master mode during SLEEP mode, the I2C Ch.*n* operating clock CLK_I2C*n* must be configured so that it will keep supplying by writing 0 to the CLGOSC.*xxxx*SLPC bit for the CLK_I2C*n* clock source.
   The I2C operating clock should be selected so that the baud rate generator will be configured easily.

Slave mode operating clock

The I2C set to slave mode uses the SCL supplied from the I²C master as its operating clock. The clock setting by the I2C*n*CLK register is ineffective.
The I2C keeps operating using the clock supplied from the external I²C master even if all the internal clocks halt during SLEEP mode, so the I2C can receive data and can generate receive buffer full interrupts.

13.3.2 Clock Supply in DEBUG Mode

In master mode, the CLK_I2C*n* supply during DEBUG mode should be controlled using the I2C*n*CLK.DBRUN bit. The CLK_I2C*n* supply to the I2C Ch.*n* is suspended when the CPU enters DEBUG mode if the I2C*n*CLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_I2C*n* supply resumes. Although the I2C Ch.*n* stops operating when the CLK_I2C*n* supply is suspended, the output pin and registers retain the status before DEBUG mode was entered. If the I2C*n*CLK.DBRUN bit = 1, the CLK_I2C*n* supply is not suspended and the I2C Ch.*n* will keep operating in DEBUG mode.
In slave mode, the I2C Ch.*n* operates with the external I²C master clock input from the SCL*n* pin regardless of whether the CPU is placed into DEBUG mode or normal mode.

13.3.3 Baud Rate Generator

The I2C includes a baud rate generator to generate the serial clock SCL used in master mode. The I2C set to slave mode does not use the baud rate generator, as it operates with the serial clock input from the SCL*n* pin.

Setting data transfer rate (for master mode)

The transfer rate is determined by the I2C*n*BR.BRT[6:0] bit settings. Use the following equations to calculate the setting values for obtaining the desired transfer rate.

$$bps = \frac{f_{CLK\_I2Cn}}{(BRT + 3) \times 2} \qquad BRT = \frac{f_{CLK\_I2Cn}}{bps \times 2} - 3 \qquad \text{(Eq. 13.1)}$$

Where
- bps: Data transfer rate [bit/s]
- f_{CLK_I2Cn}: I2C operating clock frequency [Hz]
- BRT: I2C*n*BR.BRT[6:0] bits setting value (1 to 127)

∗ The equations above do not include SCL rising/falling time and delay time by clock stretching (see Figure 13.3.3.1).

Note: The I²C bus transfer rate is limited to 100 kbit/s in standard mode or 400 kbit/s in fast mode. Do not set a transfer rate exceeding the limit.

Baud rate generator clock output and operations for supporting clock stretching

Figure 13.3.3.1 shows the clock generated by the baud rate generator and the clock waveform on the I²C bus.

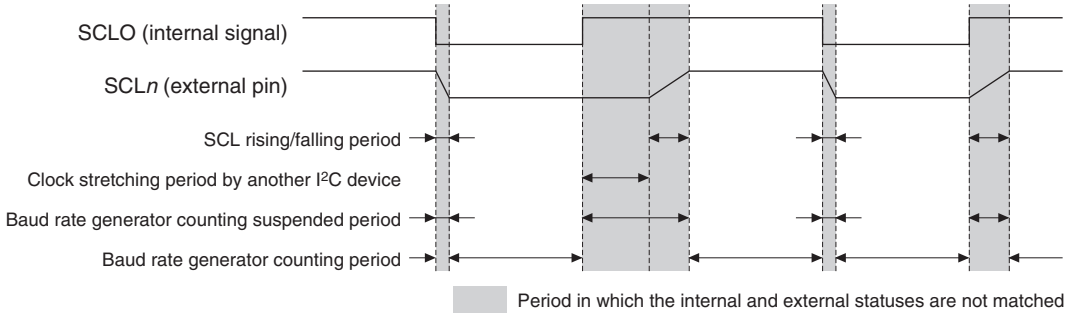


Figure 13.3.3.1 Baud Rate Generator Output Clock and SCL*n* Output Waveform

The baud rate generator output clock SCLO is compared with the SCL*n* pin status and the results are returned to the baud rate generator. If a mismatch has occurred between SCLO and SCL*n* pin levels, the baud rate generator suspends counting. This extends the clock to control data transfer during the SCL signal rising/falling period and clock stretching period in which SCL is fixed at low by a slave device.

13.4 Operations

13.4.1 Initialization

The I2C Ch.*n* should be initialized with the procedure shown below.

When using the I2C in master mode

1. Configure the operating clock and the baud rate generator using the I2C*n*CLK and I2C*n*BR registers.
2. Assign the I2C Ch.*n* input/output function to the ports. (Refer to the "I/O Ports" chapter.)
3. Set the following bits when using the interrupt:
   - Write 1 to the interrupt flags in the I2C*n*INTF register. (Clear interrupt flags)
   - Set the interrupt enable bits in the I2C*n*INTE register to 1. (Enable interrupts)
4. Set the following I2C*n*CTL register bits:
   - Set the I2C*n*CTL.MST bit to 1. (Set master mode)
   - Set the I2C*n*CTL.SFTRST bit to 1. (Execute software reset)
   - Set the I2C*n*CTL.MODEN bit to 1. (Enable I2C Ch.*n* operations)

When using the I2C in slave mode

1. Set the following I2C*n*MOD register bits:
   - I2C*n*MOD.OADR10 bit (Set 10/7-bit address mode)
   - I2C*n*MOD.GCEN bit (Enable response to general call address)
2. Set its own address to the I2C*n*OADR.OADR[9:0] (or OADR[6:0]) bits.
3. Assign the I2C Ch.*n* input/output function to the ports. (Refer to the "I/O Ports" chapter.)
4. Set the following bits when using the interrupt:
   - Write 1 to the interrupt flags in the I2C*n*INTF register. (Clear interrupt flags)
   - Set the interrupt enable bits in the I2C*n*INTE register to 1. (Enable interrupts)
5. Set the following I2C*n*CTL register bits:
   - Set the I2C*n*CTL.MST bit to 0. (Set slave mode)
   - Set the I2C*n*CTL.SFTRST bit to 1. (Execute software reset)
   - Set the I2C*n*CTL.MODEN bit to 1. (Enable I2C Ch.*n* operations)

13.4.2 Data Transmission in Master Mode

A data sending procedure in master mode and the I2C Ch.*n* operations are shown below. Figures 13.4.2.1 and 13.4.2.2 show an operation example and a flowchart, respectively.

Data sending procedure

1. Issue a START condition by setting the I2C*n*CTL.TXSTART bit to 1.
2. Wait for a transmit buffer empty interrupt (I2C*n*INTF.TBEIF bit = 1) or a START condition interrupt (I2C*n*INTF.STARTIF bit = 1).
   Clear the I2C*n*INTF.STARTIF bit by writing 1 after the interrupt has occurred.
3. Write the 7-bit slave address to the I2C*n*TXD.TXD[7:1] bits and 0 that represents WRITE as the data transfer direction to the I2C*n*TXD.TXD0 bit.
4. Wait for a transmit buffer empty interrupt (I2C*n*INTF.TBEIF bit = 1) generated when an ACK is received or a NACK reception interrupt (I2C*n*INTF.NACKIF bit = 1) generated when a NACK is received.
   i. Go to Step 5 if transmit data remains when a transmit buffer empty interrupt has occurred.
   ii. Go to Step 7 or 1 after clearing the I2C*n*INTF.NACKIF bit when a NACK reception interrupt has occurred.
5. Write transmit data to the I2C*n*TXD register.
6. Repeat Steps 4 and 5 until the end of transmit data.
7. Issue a STOP condition by setting the I2C*n*CTL.TXSTOP bit to 1.
8. Wait for a STOP condition interrupt (I2C*n*INTF.STOPIF bit = 1).
   Clear the I2C*n*INTF.STOPIF bit by writing 1 after the interrupt has occurred.

Data sending operations

Generating a START condition

The I2C Ch.*n* starts generating a START condition when the I2C*n*CTL.TXSTART bit is set to 1. When the generating operation has completed, the I2C Ch.*n* clears the I2C*n*CTL.TXSTART bit to 0 and sets both the I2C*n*INTF.STARTIF and I2C*n*INTF.TBEIF bits to 1.

Sending slave address and data

If the I2C*n*INTF.TBEIF bit = 1, a slave address or data can be written to the I2C*n*TXD register. The I2C Ch.*n* pulls down SCL to low and enters standby state until data is written to the I2C*n*TXD register. The writing operation triggers the I2C Ch.*n* to send the data to the shift register automatically and to output eight clock pulses and data bits to the I²C bus.

When the slave device returns an ACK as the response, the I2C*n*INTF.TBEIF bit is set to 1. After this interrupt occurs, the subsequent data may be sent or a STOP/repeated START condition may be issued to terminate transmission. If the slave device returns NACK, the I2C*n*INTF.NACKIF bit is set to 1 without setting the I2C*n*INTF.TBEIF bit.

Generating a STOP/repeated START condition

After the I2C*n*INTF.TBEIF bit is set to 1 (transmit buffer empty) or the I2C*n*INTF.NACKIF bit is set to 1 (NACK received), setting the I2C*n*CTL.TXSTOP bit to 1 generates a STOP condition. When the bus free time (t_{BUF} defined in the I²C Specifications) has elapsed after the STOP condition has been generated, the I2C*n*CTL.TXSTOP bit is cleared to 0 and the I2C*n*INTF.STOPIF bit is set to 1.

When setting the I2C*n*CTL.TXSTART bit to 1 while the I2C*n*INTF.TBEIF bit = 1 (transmit buffer empty) or the I2C*n*INTF.NACKIF bit = 1 (NACK received), the I2C Ch.*n* generates a repeated START condition. When the repeated START condition has been generated, the I2C*n*INTF.STARTIF and I2C*n*INTF.TBEIF bits are both set to 1 same as when a START condition has been generated.

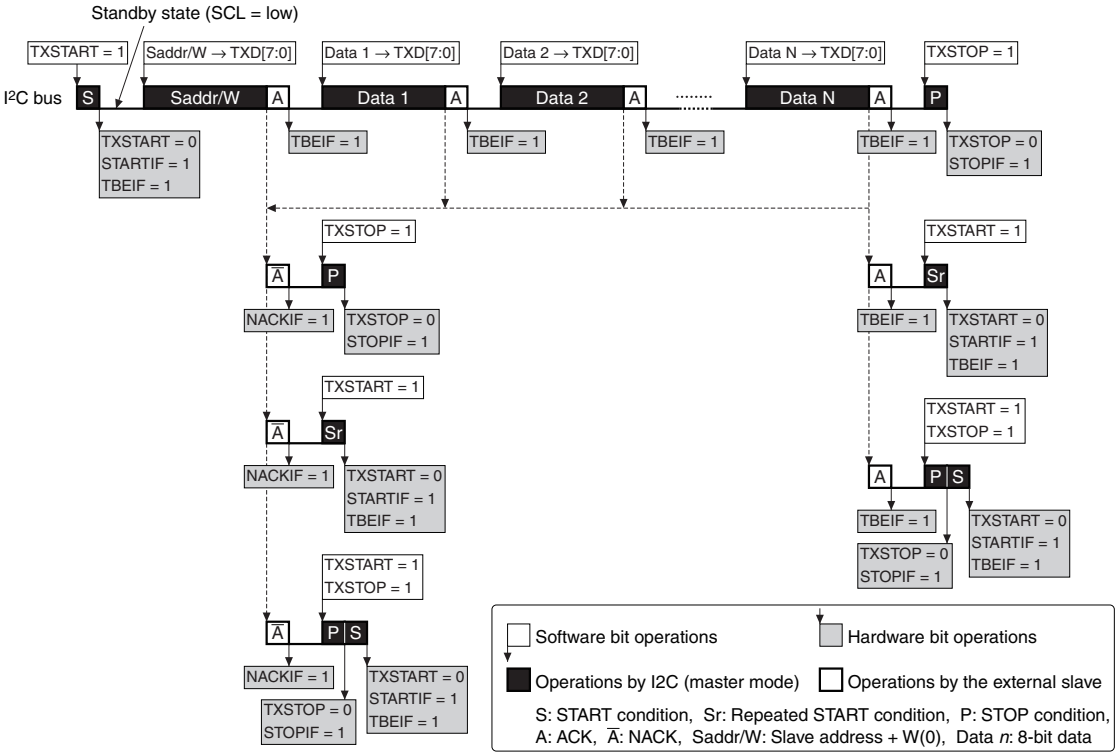


Figure 13.4.2.1 Example of Data Sending Operations in Master Mode

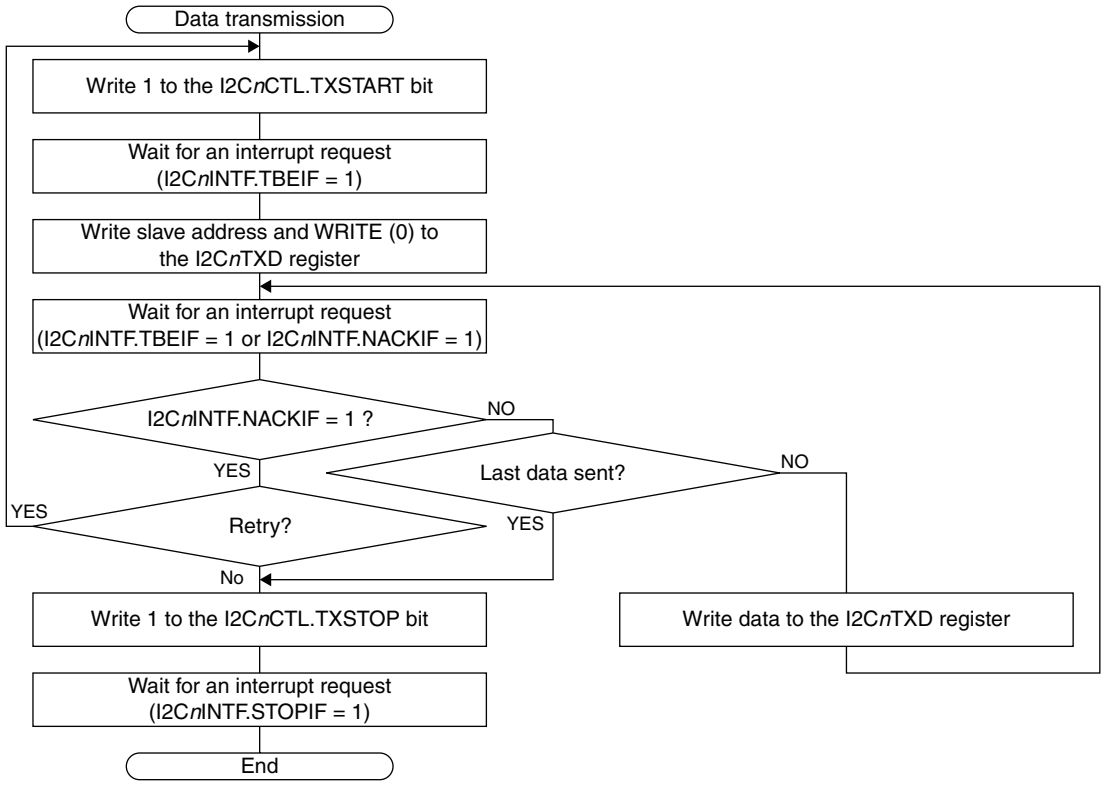


Figure 13.4.2.2 Master Mode Data Transmission Flowchart

13.4.3 Data Reception in Master Mode

A data receiving procedure in master mode and the I2C Ch.*n* operations are shown below. Figures 13.4.3.1 and 13.4.3.2 show an operation example and a flowchart, respectively.

Data receiving procedure

1. When receiving one-byte data, write 1 to the I2C*n*CTL.TXNACK bit.
2. Issue a START condition by setting the I2C*n*CTL.TXSTART bit to 1.
3. Wait for a transmit buffer empty interrupt (I2C*n*INTF.TBEIF bit = 1) or a START condition interrupt (I2C*n*INTF.STARTIF bit = 1).
   Clear the I2C*n*INTF.STARTIF bit by writing 1 after the interrupt has occurred.
4. Write the 7-bit slave address to the I2C*n*TXD.TXD[7:1] bits and 1 that represents READ as the data transfer direction to the I2C*n*TXD.TXD0 bit.
5. Wait for a receive buffer full interrupt (I2C*n*INTF.RBFIF bit = 1) generated when a one-byte reception has completed or a NACK reception interrupt (I2C*n*INTF.NACKIF bit = 1) generated when a NACK is received.
   i. Go to Step 6 when a receive buffer full interrupt has occurred.
   ii. Clear the I2C*n*INTF.NACKIF bit and issue a STOP condition by setting the I2C*n*CTL.TXSTOP bit to 1 when a NACK reception interrupt has occurred. Then go to Step 9 or Step 2 if making a retry.
6. Perform one of the operations below when the last or next-to-last data is received.
   i. When the next-to-last data is received, write 1 to the I2C*n*CTL.TXNACK bit to send a NACK after the last data is received, and then go to Step 7.
   ii. When the last data is received, read the received data from the I2C*n*RXD register and set the I2C*n*CTL.TXSTOP to 1 to generate a STOP condition. Then go to Step 9.
7. Read the received data from the I2C*n*RXD register.
8. Repeat Steps 5 to 7 until the end of data reception.
9. Wait for a STOP condition interrupt (I2C*n*INTF.STOPIF bit = 1).
   Clear the I2C*n*INTF.STOPIF bit by writing 1 after the interrupt has occurred.

Data receiving operations

Generating a START condition

It is the same as the data transmission in master mode.

Sending slave address

It is the same as the data transmission in master mode. Note, however, that the I2C*n*TXD.TXD0 bit must be set to 1 that represents READ as the data transfer direction to issue a request to the slave to send data.

Receiving data

After the slave address has been sent, the slave device sends an ACK and the first data. The I2C Ch.*n* sets the I2C*n*INTF.RBFIF bit to 1 after the data reception has completed. Furthermore, the I2C Ch.*n* returns an ACK. To return a NACK, such as for a response after the last data has been received, write 1 to the I2C*n*CTL.TXNACK bit before the I2C*n*INTF.RBFIF bit is set to 1.

The received data can be read out from the I2C*n*RXD register after a receive buffer full interrupt has occurred. The I2C Ch.*n* pulls down SCL to low and enters standby state until data is read out from the I2C*n*RXD register.

This reading triggers the I2C Ch.*n* to start subsequent data reception.

Generating a STOP or repeated START condition

It is the same as the data transmission in master mode.

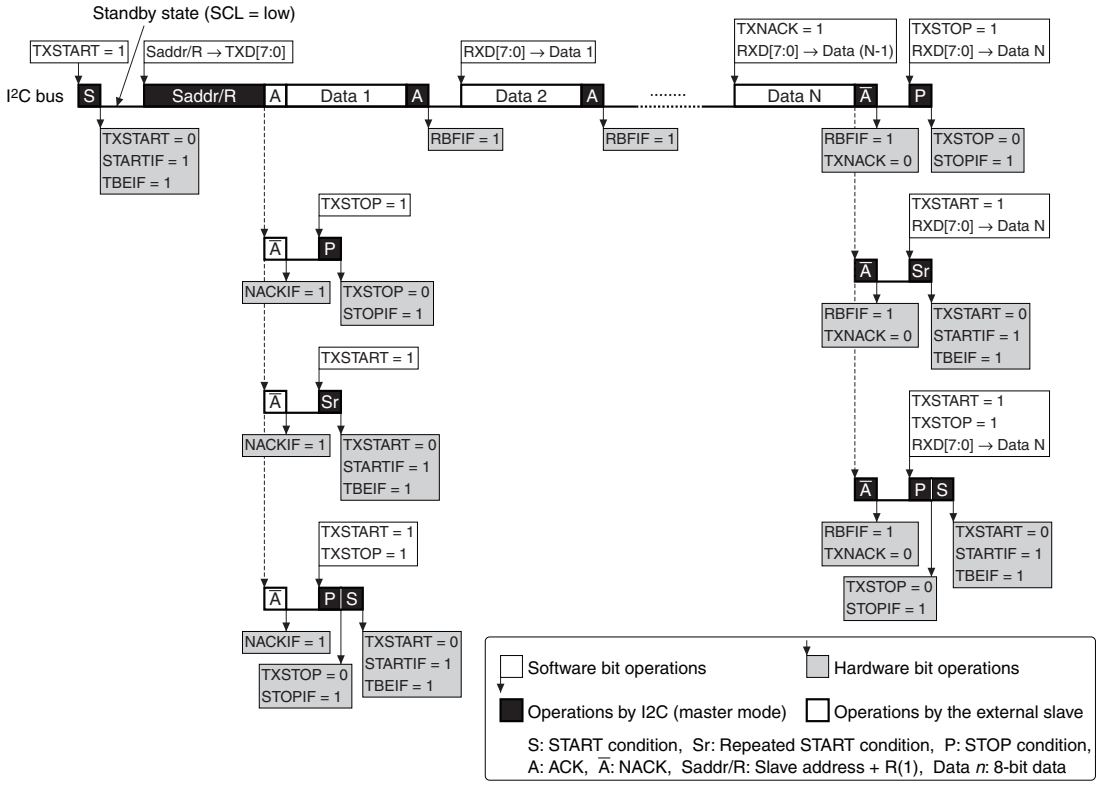


Figure 13.4.3.1 Example of Data Receiving Operations in Master Mode

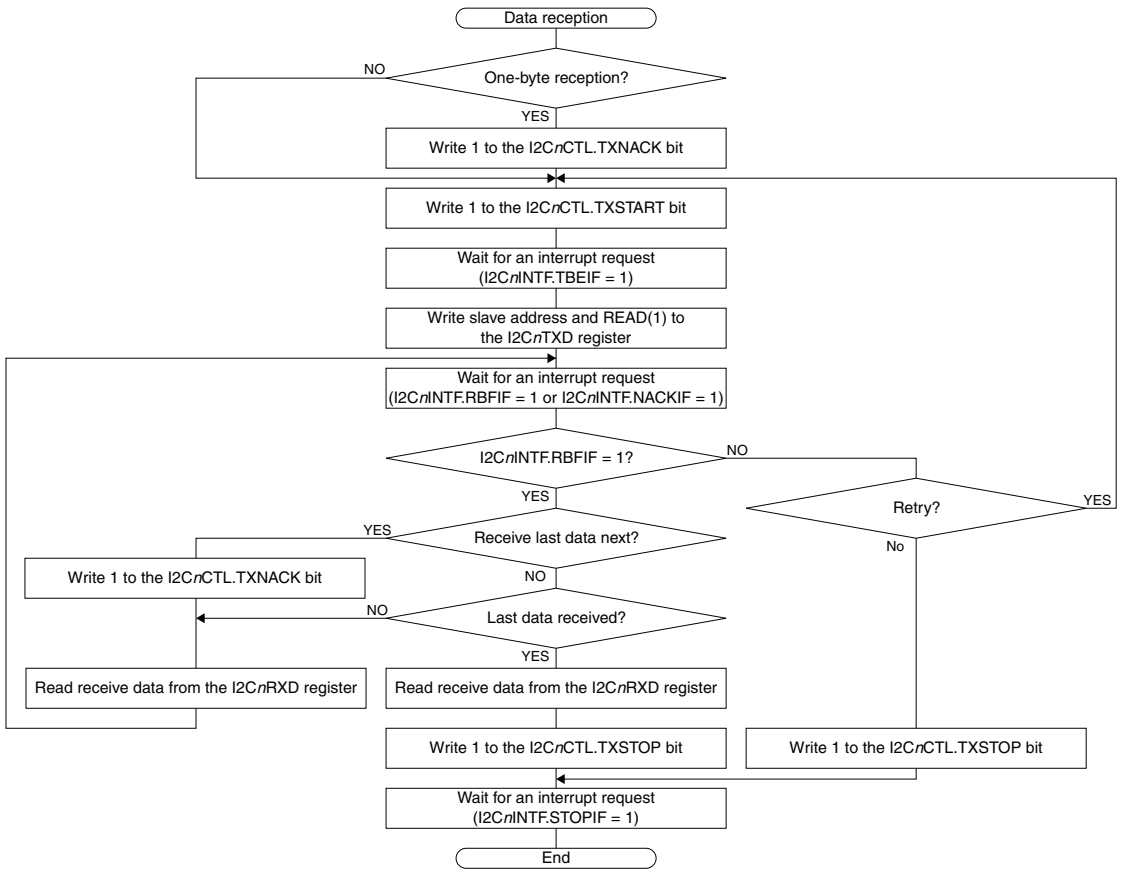


Figure 13.4.3.2 Master Mode Data Reception Flowchart

13.4.4 10-bit Addressing in Master Mode

A 10-bit address consists of the first address that contains two high-order bits and the second address that contains eight low-order bits.

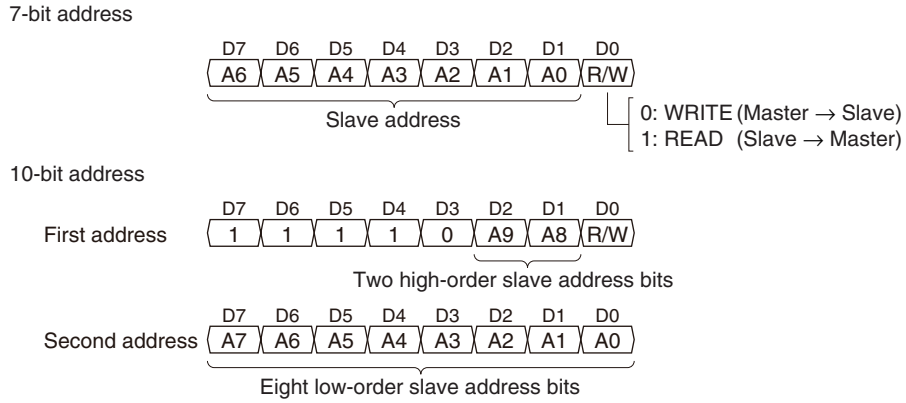


Figure 13.4.4.1 10-bit Address Configuration

The following shows a procedure to start data transfer in 10-bit address mode when the I2C Ch.*n* is placed into master mode (see the 7-bit mode descriptions above for control procedures when a NACK is received or sending/receiving data). Figure 13.4.4.2 shows an operation example.

Starting data transmission in 10-bit address mode

1. Issue a START condition by setting the I2C*n*CTL.TXSTART bit to 1.
2. Wait for a transmit buffer empty interrupt (I2C*n*INTF.TBEIF bit = 1) or a START condition interrupt (I2C*n*INTF.STARTIF bit = 1).
   Clear the I2C*n*INTF.STARTIF bit by writing 1 after the interrupt has occurred.
3. Write the first address to the I2C*n*TXD.TXD[7:1] bits and 0 that represents WRITE as the data transfer direction to the I2C*n*TXD.TXD0 bit.
4. Wait for a transmit buffer empty interrupt (I2C*n*INTF.TBEIF bit = 1).
5. Write the second address to the I2C*n*TXD.TXD[7:0] bits.
6. Wait for a transmit buffer empty interrupt (I2C*n*INTF.TBEIF bit = 1).
7. Perform data transmission.

Starting data reception in 10-bit address mode

1 to 6. These steps are the same as the data transmission starting procedure described above.

7. Issue a repeated START condition by setting the I2C*n*CTL.TXSTART bit to 1.
8. Wait for a transmit buffer empty interrupt (I2C*n*INTF.TBEIF bit = 1) or a START condition interrupt (I2C*n*INTF.STARTIF bit = 1).
   Clear the I2C*n*INTF.STARTIF bit by writing 1 after the interrupt has occurred.
9. Write the first address to the I2C*n*TXD.TXD[7:1] bits and 1 that represents READ as the data transfer direction to the I2C*n*TXD.TXD0 bit.
10. Perform data reception.

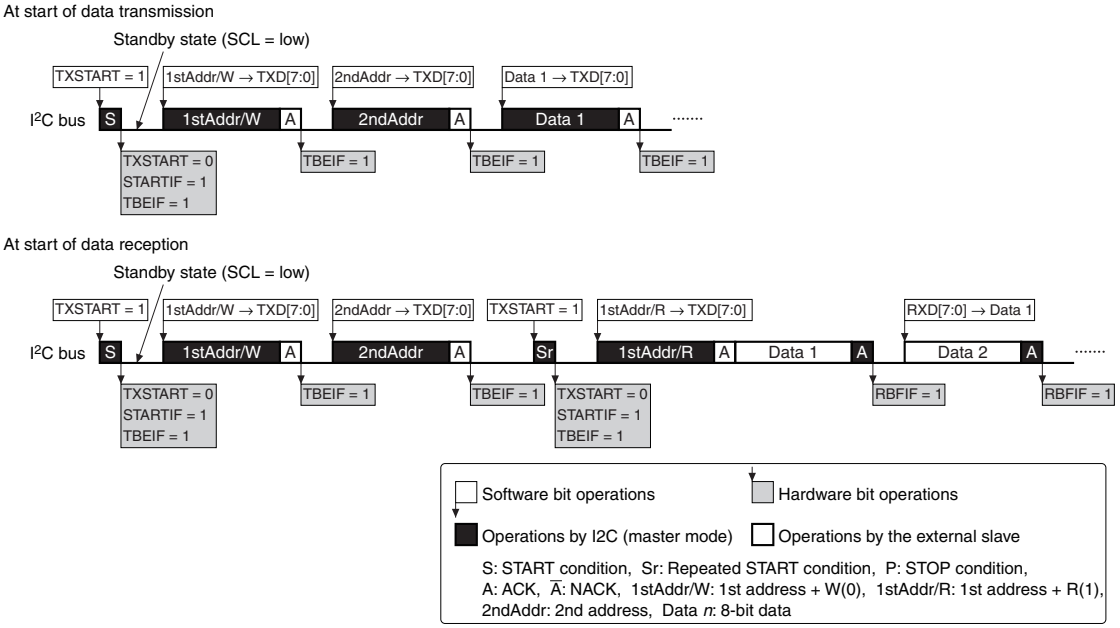


Figure 13.4.4.2 Example of Data Transfer Starting Operations in 10-bit Address Mode (Master Mode)

13.4.5 Data Transmission in Slave Mode

A data sending procedure in slave mode and the I2C Ch.*n* operations are shown below. Figures 13.4.5.1 and 13.4.5.2 show an operation example and a flowchart, respectively.

Data sending procedure

1. Wait for a START condition interrupt (I2C*n*INTF.STARTIF bit = 1).
   Clear the I2C*n*INTF.STARTIF bit by writing 1 after the interrupt has occurred.
2. Check to see if the I2C*n*INTF.TR bit = 1 (transmission mode).
   (Start a data receiving procedure if the I2C*n*INTF.TR bit = 0.)
3. Write transmit data to the I2C*n*TXD register.
4. Wait for a transmit buffer empty interrupt (I2C*n*INTF.TBEIF bit = 1), a NACK reception interrupt (I2C*n*INTF.NACKIF bit = 1), or a STOP condition interrupt (I2C*n*INTF.STOPIF bit = 1).
   i. Go to Step 3 when a transmit buffer empty interrupt has occurred.
   ii. Go to Step 5 after clearing the I2C*n*INTF.NACKIF bit when a NACK reception interrupt has occurred.
   iii. Go to Step 6 when a STOP condition interrupt has occurred.
5. Wait for a STOP condition interrupt (I2C*n*INTF.STOPIF bit = 1) or a START condition interrupt (I2C*n*INTF.STARTIF bit = 1).
   i. Go to Step 6 when a STOP condition interrupt has occurred.
   ii. Go to Step 2 when a START condition interrupt has occurred.
6. Clear the I2C*n*INTF.STOPIF bit and then terminate data sending operations.

Data sending operations

START condition detection and slave address check

While the I2C*n*CTL.MODEN bit = 1 and the I2C*n*CTL.MST bit = 0 (slave mode), the I2C Ch.*n* monitors the I²C bus. When the I2C Ch.*n* detects a START condition, it starts receiving of the slave address sent from the master. If the received address is matched with the own address set to the I2C*n*OADR.OADR[6:0] bits (when the I2C*n*MOD.OADR10 bit = 0 (7-bit address mode)) or the I2C*n*OADR.OADR[9:0] bits (when the I2C*n*MOD.OADR10 bit = 1 (10-bit address mode)), the I2C*n*INTF.STARTIF bit and the I2C*n*INTF.BSY bit are both set to 1. The I2C Ch.*n* sets the I2C*n*INTF.TR bit to the R/W bit value in the received address. If this value is 1, the I2C Ch.*n* sets the I2C*n*INTF.TBEIF bit to 1 and starts data sending operations.

Sending the first data byte

After the valid slave address has been received, the I2C Ch.*n* pulls down SCL to low and enters standby state until data is written to the I2C*n*TXD register. This puts the I²C bus into clock stretching state and the external master into standby state. When transmit data is written to the I2C*n*TXD register, the I2C Ch.*n* clears the I2C*n*INTF.TBEIF bit and sends an ACK to the master. The transmit data written in the I2C*n*TXD register is automatically transferred to the shift register and the I2C*n*INTF.TBEIF bit is set to 1. The data bits in the shift register are output in sequence to the I²C bus.

Sending subsequent data

If the I2C*n*INTF.TBEIF bit = 1, subsequent transmit data can be written during data transmission. If the I2C*n*INTF.TBEIF bit is still set to 1 when the data transmission from the shift register has completed, the I2C Ch.*n* pulls down SCL to low (sets the I²C bus into clock stretching state) until transmit data is written to the I2C*n*TXD register.

If the next transmit data already exists in the I2C*n*TXD register or data has been written after the above, the I2C Ch.*n* sends the subsequent eight-bit data when an ACK from the external master is received. At the same time, the I2C*n*INTF.BYTEENDIF bit is set to 1. If a NACK is received, the I2C*n*INTF.NACKIF bit is set to 1 without sending data.

STOP/repeated START condition detection

While the I2C*n*CTL.MST bit = 0 (slave mode) and the I2C*n*INTF.BSY = 1, the I2C Ch.*n* monitors the I²C bus. When the I2C Ch.*n* detects a STOP condition, it terminates data sending operations. At this time, the I2C*n*INTF.BSY bit is cleared to 0 and the I2C*n*INTF.STOPIF bit is set to 1. Also when the I2C Ch.*n* detects a repeated START condition, it terminates data sending operations. In this case, the I2C*n*INTF.STARTIF bit is set to 1.

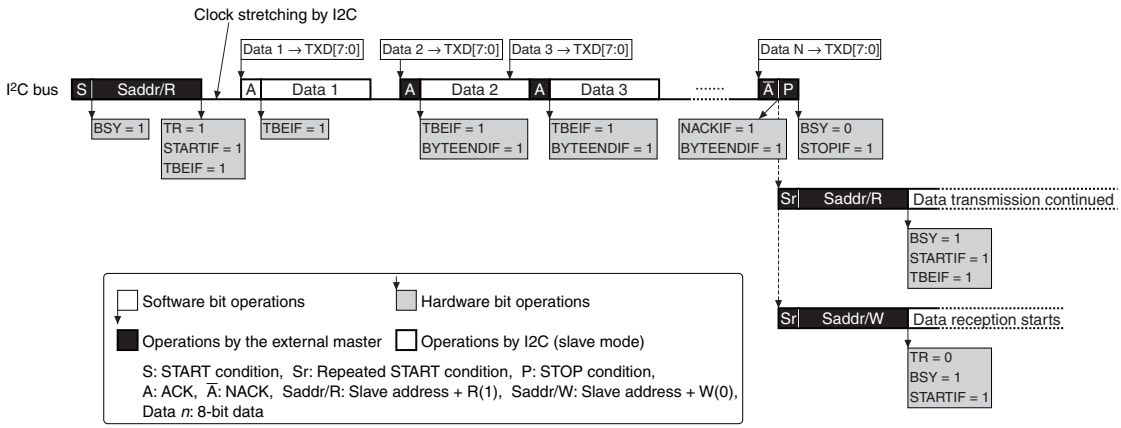


Figure 13.4.5.1 Example of Data Sending Operations in Slave Mode

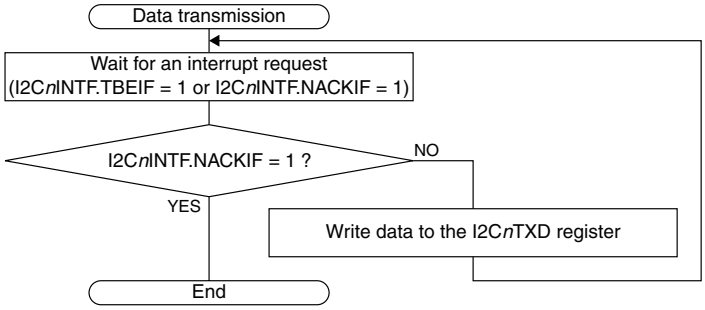


Figure 13.4.5.2 Slave Mode Data Transmission Flowchart

13.4.6 Data Reception in Slave Mode

A data receiving procedure in slave mode and the I2C Ch.*n* operations are shown below. Figures 13.4.6.1 and 13.4.6.2 show an operation example and a flowchart, respectively.

Data receiving procedure

1. When receiving one-byte data, write 1 to the I2C*n*CTL.TXNACK bit.
2. Wait for a START condition interrupt (I2C*n*INTF.STARTIF bit = 1).
3. Check to see if the I2C*n*INTF.TR bit = 0 (reception mode).
   (Start a data sending procedure if I2C*n*INTF.TR bit = 1.)
4. Clear the I2C*n*INTF.STARTIF bit by writing 1.
5. Wait for a receive buffer full interrupt (I2C*n*INTF.RBFIF bit = 1) generated when a one-byte reception has completed or an end of transfer interrupt (I2C*n*INTF.BYTEENDIF bit = 1).
   Clear the I2C*n*INTF.BYTEENDIF bit by writing 1 after the interrupt has occurred.
6. If the next receive data is the last one, write 1 to the I2C*n*CTL.TXNACK bit to send a NACK after it is received.
7. Read the received data from the I2C*n*RXD register.
8. Repeat Steps 5 to 7 until the end of data reception.
9. Wait for a STOP condition interrupt (I2C*n*INTF.STOPIF bit = 1) or a START condition interrupt (I2C*n*INTF.STARTIF bit = 1).
   i. Go to Step 10 when a STOP condition interrupt has occurred.
   ii. Go to Step 3 when a START condition interrupt has occurred.
10. Clear the I2C*n*INTF.STOPIF bit and then terminate data receiving operations.

Data receiving operations

START condition detection and slave address check

It is the same as the data transmission in slave mode.
However, the I2C*n*INTF.TR bit is cleared to 0 and the I2C*n*INTF.TBEIF bit is not set.
If the I2C*n*MOD.GCEN bit is set to 1 (general call address response enabled), the I2C Ch.*n* starts data receiving operations when the general call address is received.
Slave mode can be operated even in SLEEP mode, it makes it possible to wake the CPU up using an interrupt when an address match is detected.

Receiving the first data byte

After the valid slave address has been received, the I2C Ch.*n* sends an ACK and pulls down SCL to low until 1 is written to the I2C*n*INTF.STARTIF bit. This puts the I²C bus into clock stretching state and the external master into standby state. When 1 is written to the I2C*n*INTF.STARTIF bit, the I2C Ch.*n* releases SCL and receives data sent from the external master into the shift register. After eight-bit data has been received, the I2C Ch.*n* sends an ACK and pulls down SCL to low. The received data in the shift register is transferred to the receive data buffer and the I2C*n*INTF.RBFIF and I2C*n*INTF.BYTEENDIF bits are both set to 1. After that, the received data can be read out from the I2C*n*RXD register.

Receiving subsequent data

When the received data is read out from the I2C*n*RXD register after the I2C*n*INTF.RBFIF bit has been set to 1, the I2C Ch.*n* clears the I2C*n*INTF.RBFIF bit to 0, releases SCL, and receives subsequent data sent from the external master. After eight-bit data has been received, the I2C Ch.*n* sends an ACK and pulls down SCL to low. The received data in the shift register is transferred to the receive data buffer and the I2C*n*INTF.RBFIF and I2C*n*INTF.BYTEENDIF bits are both set to 1.

To return a NACK after eight-bit data is received, such as when terminating data reception, write 1 to the I2C*n*CTL.TXNACK bit before the data reception is completed. The I2C*n*CTL.TXNACK bit is automatically cleared to 0 after a NACK has been sent.

STOP/repeated START condition detection

It is the same as the data transmission in slave mode.

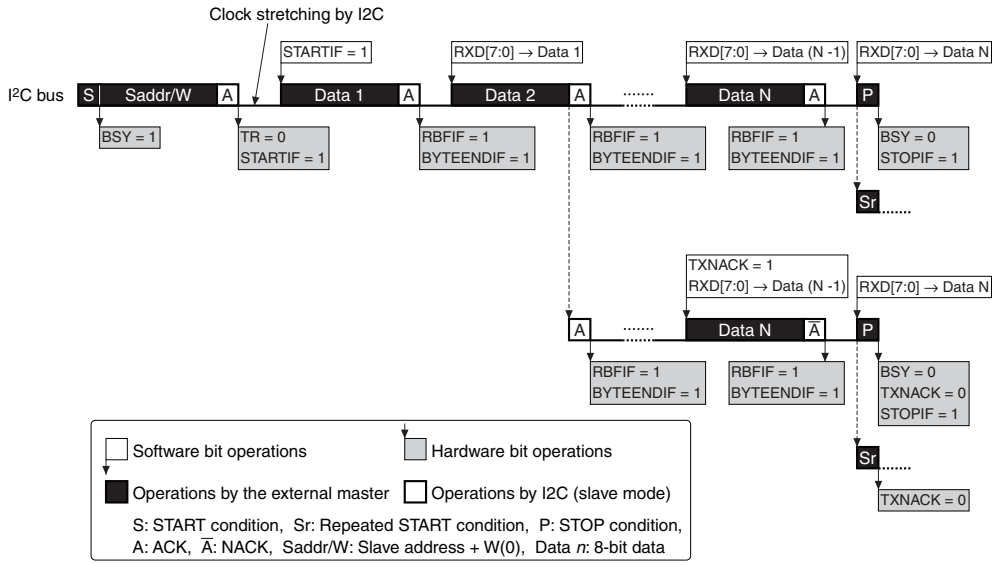


Figure 13.4.6.1 Example of Data Receiving Operations in Slave Mode

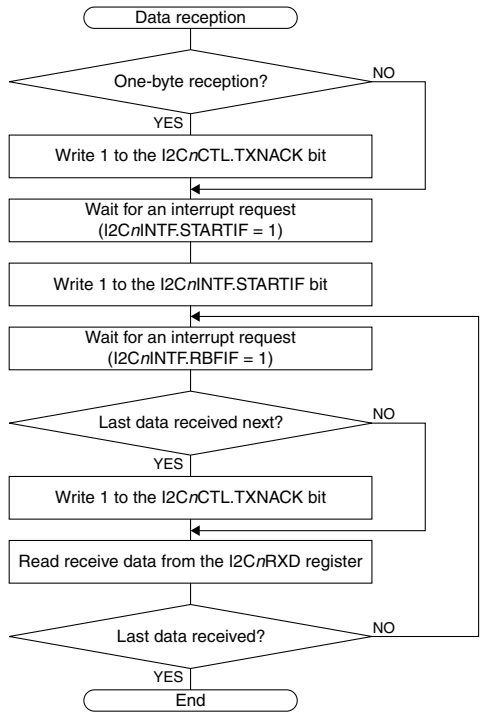


Figure 13.4.6.2 Slave Mode Data Reception Flowchart

13.4.7 Slave Operations in 10-bit Address Mode

The I2C Ch.*n* functions as a slave device in 10-bit address mode when the I2C*n*CTL.MST bit = 0 and the I2C*n*MOD.OADR10 bit = 1.

The following shows the address receiving operations in 10-bit address mode. Figure 13.4.7.1 shows an operation example. See Figure 13.4.4.1 for the 10-bit address configuration.

10-bit address receiving operations

After a START condition is issued, the master sends the first address that includes the two high-order slave address bits and the R/W bit (= 0). If the received two high-order slave address bits are matched with the I2C*n*OADR.OADR[9:8] bits, the I2C Ch.*n* returns an ACK. At this time, other slaves may returns an ACK as the two high-order bits may be matched.

Then the master sends the eight low-order slave address bits as the second address. If this address is matched with the I2C*n*OADR.OADR[7:0] bits, the I2C Ch.*n* returns an ACK and starts data receiving operations.

If the master issues a request to the slave to send data (data reception in the master), the master generates a repeated START condition and sends the first address with the R/W bit set to 1. This reception switches the I2C Ch.*n* to data sending mode.

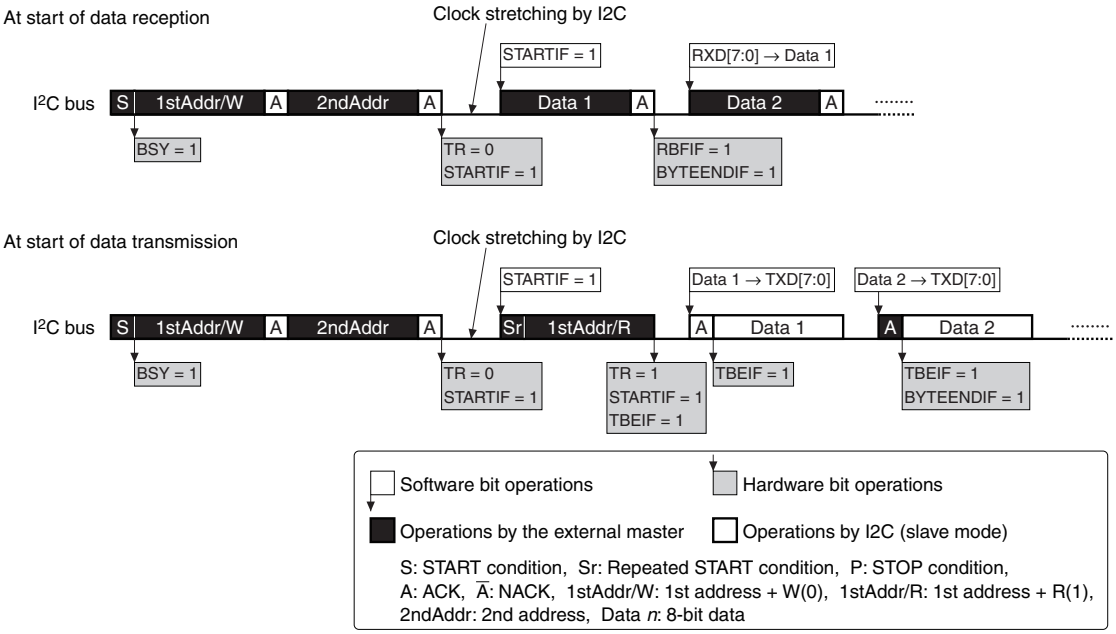


Figure 13.4.7.1 Example of Data Transfer Starting Operations in 10-bit Address Mode (Slave Mode)

13.4.8 Automatic Bus Clearing Operation

The I2C Ch.*n* set into master mode checks the SDA state immediately before generating a START condition. If SDA is set to a low level at this time, the I2C Ch.*n* automatically executes bus clearing operations that output up to ten clocks from the SCL*n* pin with SDA left free state.

When SDA goes high from low within nine clocks, the I2C Ch.*n* issues a START condition and starts normal operations. If SDA does not change from low when the I2C Ch.*n* outputs the ninth clock, it is regarded as an automatic bus clearing failure. In this case, the I2C Ch.*n* clears the I2C*n*CTL.TXSTART bit to 0 and sets both the I2C*n*INTF.ERRIF and I2C*n*INTF.STARTIF bits to 1.

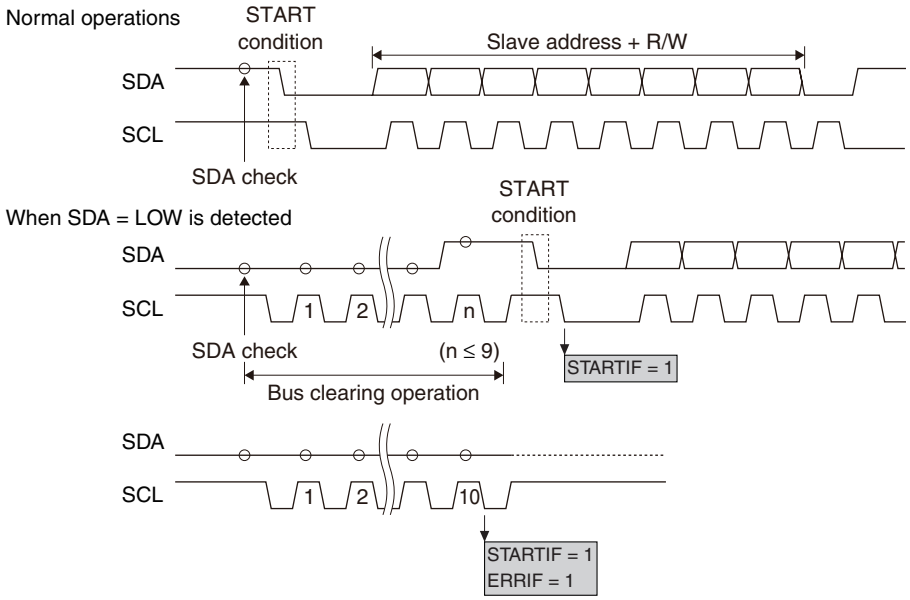


Figure 13.4.8.1 Automatic Bus Clearing Operation

13.4.9 Error Detection

The I2C includes a hardware error detection function.
Furthermore, the I2C*n*INTF.SDALOW and I2C*n*INTF.SCLLOW bits are provided to allow software to check whether the SDA and SCL lines are fixed at low. If unintended low level is detected on SDA or SCL, a software recovery processing, such as I2C Ch.*n* software reset, can be performed.

The table below lists the hardware error detection conditions and the notification method.

Table 13.4.9.1 Hardware Error Detection Function

No.	Error detecting period/timing	I²C bus line monitored and error condition	Notification method
1	While the I2C Ch.*n* controls SDA to high for sending address, data, or a NACK	SDA = low	I2C*n*INTF.ERRIF = 1
2	<Master mode only> When 1 is written to the I2C*n*CTL.TXSTART bit while the I2C*n*INTF.BSY bit = 0	SCL = low	I2C*n*INTF.ERRIF = 1 I2C*n*CTL.TXSTART = 0 I2C*n*INTF.STARTIF = 1
3	<Master mode only> When 1 is written to the I2C*n*CTL.TXSTOP bit while the I2C*n*INTF.BSY bit = 0	SCL = low	I2C*n*INTF.ERRIF = 1 I2C*n*CTL.TXSTOP = 0 I2C*n*INTF.STOPIF = 1
4	<Master mode only> When 1 is written to the I2C*n*CTL.TXSTART bit while the I2C*n*INTF.BSY bit = 0 (Refer to "Automatic Bus Clearing Operation.")	SDA Automatic bus clearing failure	I2C*n*INTF.ERRIF = 1 I2C*n*CTL.TXSTART = 0 I2C*n*INTF.STARTIF = 1

13.5 Interrupts

The I2C has a function to generate the interrupts shown in Table 13.5.1.

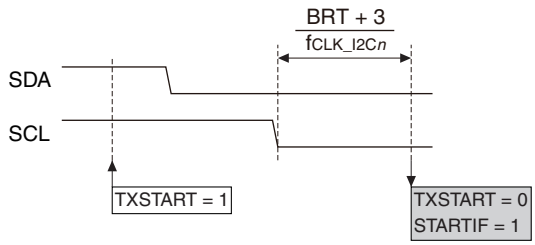
Table 13.5.1 I2C Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
End of data transfer	I2C*n*INTF.BYTEENDIF	When eight-bit data transfer and the following ACK/NACK transfer are completed	Writing 1, software reset
General call address reception	I2C*n*INTF.GCIF	Slave mode only: When the general call address is received	Writing 1, software reset
NACK reception	I2C*n*INTF.NACKIF	When a NACK is received	Writing 1, software reset
STOP condition	I2C*n*INTF. STOPIF	Master mode: When a STOP condition is generated and the bus free time (t_{BUF}) between STOP and START conditions has elapsed Slave mode: When a STOP condition is detected while the I2C Ch.*n* is selected as the slave currently accessed	Writing 1, software reset
START condition	I2C*n*INTF. STARTIF	Master mode: When a START condition is issued Slave mode: When an address match is detected (including general call)	Writing 1, software reset
Error detection	I2C*n*INTF. ERRIF	Refer to "Error Detection."	Writing 1, software reset
Receive buffer full	I2C*n*INTF. RBFIF	When received data is loaded to the receive data buffer	Reading received data (to empty the receive data buffer), software reset
Transmit buffer empty	I2C*n*INTF. TBEIF	Master mode: When a START condition is issued or when an ACK is received from the slave Slave mode: When transmit data written to the transmit data buffer is transferred to the shift register or when an address match is detected with R/W bit set to 1	Writing transmit data

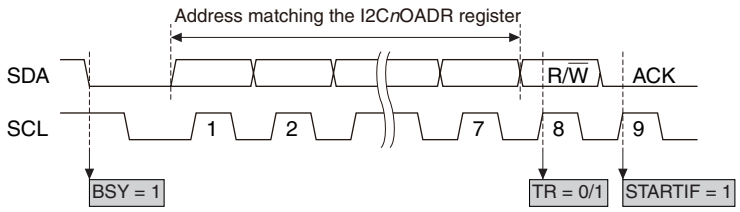
The I2C provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the "Interrupt Controller" chapter.

(1) START condition interrupt

Master mode

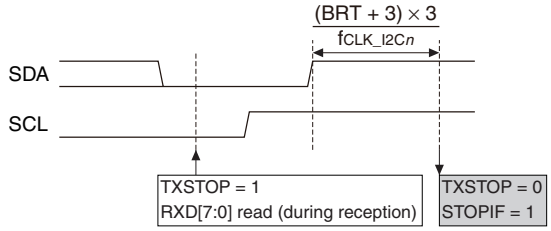


Slave mode

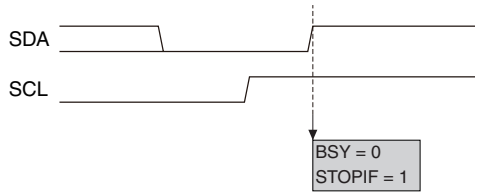


(2) STOP condition interrupt

Master mode



Slave mode



(f_{CLK_I2Cn}: I2C operating clock frequency [Hz], BRT: I2C*n*BR.BRT[6:0] bits setting value (1 to 127))

Figure 13.5.1 START/STOP Condition Interrupt Timings

13.6 Control Registers

I2C Ch.*n* Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C*n*CLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/W	
	7–6	–	0x0	–	R	
	5–4	CLKDIV[1:0]	0x0	H0	R/W	
	3–2	–	0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/W	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the I2C operating clock is supplied in DEBUG mode or not.

1 (R/W): Clock supplied in DEBUG mode

0 (R/W): No clock supplied in DEBUG mode

Bits 7–6 Reserved

Bits 5–4 CLKDIV[1:0]

These bits select the division ratio of the I2C operating clock.

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of the I2C.

Table 13.6.1 Clock Source and Division Ratio Settings

I2C*n*CLK.CLKDIV[1:0] bits	I2C*n*CLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0x3	1/8	1/1	1/8	1/1
0x2	1/4		1/4	
0x1	1/2		1/2	
0x0	1/1		1/1	

(Note) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

Note: The I2C*n*CLK register settings can be altered only when the I2C*n*CTL.MODEN bit = 0.

I2C Ch.*n* Mode Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C*n*MOD	15–8	–	0x00	–	R	–
	7–3	–	0x00	–	R	
	2	OADR10	0	H0	R/W	
	1	GCEN	0	H0	R/W	
	0	–	0	–	R	

Bits 15–3 Reserved

Bit 2 OADR10

This bit sets the number of own address bits for slave mode.
1 (R/W): 10-bit address
0 (R/W): 7-bit address

Bit 1 GCEN

This bit sets whether to respond to master general calls in slave mode or not.
1 (R/W): Respond to general calls.
0 (R/W): Do not respond to general calls.

Bit 0 Reserved

Note: The I2C*n*MOD register settings can be altered only when the I2C*n*CTL.MODEN bit = 0.

I2C Ch.*n* Baud-Rate Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C*n*BR	15–8	–	0x00	–	R	–
	7	–	0	–	R	
	6–0	BRT[6:0]	0x7f	H0	R/W	

Bits 15–7 Reserved

Bits 6–0 BRT[6:0]

These bits set the I2C Ch.*n* transfer rate for master mode. For more information, refer to "Baud Rate Generator."

Notes: • The I2C*n*BR register settings can be altered only when the I2C*n*CTL.MODEN bit = 0.

• Be sure to avoid setting the I2C*n*BR register to 0.

I2C Ch.*n* Own Address Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C*n*OADR	15–10	–	0x00	–	R	–
	9–0	OADR[9:0]	0x000	H0	R/W	

Bits 15–10 Reserved

Bits 9–0 OADR[9:0]

These bits set the own address for slave mode.
The I2C*n*OADR.OADR[9:0] bits are effective in 10-bit address mode (I2C*n*MOD.OADR10 bit = 1), or the I2C*n*OADR.OADR[6:0] bits are effective in 7-bit address mode (I2C*n*MOD.OADR10 bit = 0).

Note: The I2C*n*OADR register settings can be altered only when the I2C*n*CTL.MODEN bit = 0.

I2C Ch.*n* Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C*n*CTL	15–8	–	0x00	–	R	–
	7–6	–	0x0	–	R	
	5	MST	0	H0	R/W	
	4	TXNACK	0	H0/S0	R/W	
	3	TXSTOP	0	H0/S0	R/W	
	2	TXSTART	0	H0/S0	R/W	
	1	SFTRST	0	H0	R/W	
	0	MODEN	0	H0	R/W	

Bits 15–6 Reserved

Bit 5 MST

This bit selects the I2C Ch.*n* operating mode.

1 (R/W): Master mode

0 (R/W): Slave mode

Bit 4 TXNACK

This bit issues a request for sending a NACK at the next responding.

1 (W): Issue a NACK.

0 (W): Ineffective

1 (R): On standby or during sending a NACK

0 (R): NACK has been sent.

This bit is automatically cleared after a NACK has been sent.

Bit 3 TXSTOP

This bit issues a STOP condition in master mode. This bit is ineffective in slave mode.

1 (W): Issue a STOP condition.

0 (W): Ineffective

1 (R): On standby or during generating a STOP condition

0 (R): STOP condition has been generated.

This bit is automatically cleared when the bus free time (t_{BUF} defined in the I²C Specifications) has elapsed after the STOP condition has been generated.

Bit 2 TXSTART

This bit issues a START condition in master mode. This bit is ineffective in slave mode.

1 (W): Issue a START condition.

0 (W): Ineffective

1 (R): On standby or during generating a START condition

0 (R): START condition has been generated.

This bit is automatically cleared when a START condition has been generated.

Bit 1 SFTRST

This bit issues software reset to the I2C.

1 (W): Issue software reset

0 (W): Ineffective

1 (R): Software reset is executing.

0 (R): Software reset has finished. (During normal operation)

Setting this bit resets the I2C transmit/receive control circuit and interrupt flags. This bit is automatically cleared after the reset processing has finished.

Bit 0 MODEN

This bit enables the I2C operations.

1 (R/W): Enable I2C operations (The operating clock is supplied.)

0 (R/W): Disable I2C operations (The operating clock is stopped.)

Note: If the I2C*n*CTL.MODEN bit is altered from 1 to 0 while sending/receiving data, the data being sent/received cannot be guaranteed. When setting the I2C*n*CTL.MODEN bit to 1 again after that, be sure to write 1 to the I2C*n*CTL.SFTRST bit as well.

I2C Ch.*n* Transmit Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C*n*TXD	15–8	–	0x00	–	R	–
	7–0	TXD[7:0]	0x00	H0	R/W	

Bits 15–8 Reserved

Bits 7–0 TXD[7:0]

Data can be written to the transmit data buffer through these bits. Make sure the I2C*n*INTF.TBEIF bit is set to 1 before writing data.

Note: Be sure to avoid writing to the I2C*n*TXD register when the I2C*n*INTF.TBEIF bit = 0, otherwise transmit data cannot be guaranteed.

I2C Ch.*n* Receive Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C*n*RXD	15–8	–	0x00	–	R	–
	7–0	RXD[7:0]	0x00	H0	R	

Bits 15–8 Reserved

Bits 7–0 RXD[7:0]

The receive data buffer can be read through these bits.

I2C Ch.*n* Status and Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C*n*INTF	15–13	–	0x0	–	R	–
	12	SDALOW	0	H0	R	
	11	SCLLOW	0	H0	R	
	10	BSY	0	H0/S0	R	
	9	TR	0	H0	R	
	8	–	0	–	R	
	7	BYTEENDIF	0	H0/S0	R/W	Cleared by writing 1.
	6	GCIF	0	H0/S0	R/W	
	5	NACKIF	0	H0/S0	R/W	
	4	STOPIF	0	H0/S0	R/W	
	3	STARTIF	0	H0/S0	R/W	
	2	ERRIF	0	H0/S0	R/W	
	1	RBFIF	0	H0/S0	R	Cleared by reading the I2C*n*RXD register.
	0	TBEIF	0	H0/S0	R	Cleared by writing to the I2C*n*TXD register.

Bits 15–13 Reserved

Bit 12 SDALOW

This bit indicates that SDA is set to low level.

1 (R): SDA = Low level

0 (R): SDA = High level

Bit 11 SCLLOW

This bit indicates that SCL is set to low level.

1 (R): SCL = Low level

0 (R): SCL = High level

Bit 10 BSY

This bit indicates that the I²C bus is placed into busy status.

1 (R): I²C bus busy

0 (R): I²C bus free

Bit 9 TR

This bit indicates whether the I2C is set in transmission mode or not.

1 (R): Transmission mode

0 (R): Reception mode

Bit 8 Reserved

Bit 7 BYTEENDIF
Bit 6 GCIF
Bit 5 NACKIF
Bit 4 STOPIF
Bit 3 STARTIF
Bit 2 ERRIF
Bit 1 RBFIF
Bit 0 TBEIF

These bits indicate the I2C interrupt cause occurrence status.

1 (R): Cause of interrupt occurred

0 (R): No cause of interrupt occurred

1 (W): Clear flag (BYTEENDIF, GCIF, NACKIF, STOPIF, STARTIF, ERRIF)
Ineffective (RBFIF, TBEIF)

0 (W): Ineffective

The following shows the correspondence between the bit and interrupt:

I2C*n*INTF.BYTEENDIF bit: End of transfer interrupt
I2C*n*INTF.GCIF bit: General call address reception interrupt
I2C*n*INTF.NACKIF bit: NACK reception interrupt
I2C*n*INTF.STOPIF bit: STOP condition interrupt
I2C*n*INTF.STARTIF bit: START condition interrupt
I2C*n*INTF.ERRIF bit: Error detection interrupt
I2C*n*INTF.RBFIF bit: Receive buffer full interrupt
I2C*n*INTF.TBEIF bit: Transmit buffer empty interrupt

I2C Ch.*n* Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C*n*INTE	15–8	–	0x00	–	R	–
	7	BYTEENDIE	0	H0	R/W	
	6	GCIE	0	H0	R/W	
	5	NACKIE	0	H0	R/W	
	4	STOPIE	0	H0	R/W	
	3	STARTIE	0	H0	R/W	
	2	ERRIE	0	H0	R/W	
	1	RBFIE	0	H0	R/W	
	0	TBEIE	0	H0	R/W	

Bits 15–8 Reserved

Bit 7 BYTEENDIE
Bit 6 GCIE
Bit 5 NACKIE
Bit 4 STOPIE
Bit 3 STARTIE
Bit 2 ERRIE
Bit 1 RBFIE
Bit 0 TBEIE

These bits enable I2C interrupts.
1 (R/W): Enable interrupts
0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:
I2C*n*INTE.BYTEENDIE bit: End of transfer interrupt
I2C*n*INTE.GCIE bit: General call address reception interrupt
I2C*n*INTE.NACKIE bit: NACK reception interrupt
I2C*n*INTE.STOPIE bit: STOP condition interrupt
I2C*n*INTE.STARTIE bit: START condition interrupt
I2C*n*INTE.ERRIE bit: Error detection interrupt
I2C*n*INTE.RBFIE bit: Receive buffer full interrupt
I2C*n*INTE.TBEIE bit: Transmit buffer empty interrupt

14 DMM Controller (DSADC16)

14.1 Overview

DSADC16 is a measurement circuit for constructing a Digital Multi-Meter (DMM). It samples measurement voltages sent from the analog network unit and converts them into digital values using a ΣΔ A/D converter and digital filters. The features of the DSADC16 are listed below.

- Supported DMM measurement modes
  - DC voltage measurement (600 mV/6 V/60 V/600 V/1,000 V)
  - AC voltage (and frequency) measurement (600 mV/6 V/60 V/600 V/1,000 V)
  - Resistance measurement (600 Ω/6 kΩ/60 kΩ/600 kΩ/6 MΩ/60 MΩ)
  - Continuity check
  - Diode V_F measurement
  - Internal temperature measurement
  - DC current measurement (600 μA/6 mA/60 mA/600 mA/6 A/10 A)
  - AC current (and frequency) measurement (600 μA/6 mA/60 mA/600 mA/6 A/10 A)
  - Capacitance measurement (10 nF/100 nF/1 μF/10 μF/100 μF/1,000 μF)
- Includes a 16-bit ΣΔ A/D converter.
- Includes digital filters (Comb filter, low-pass filter, and high-pass filter).
- Sampling frequency: 25 kHz to 24.41 Hz (ΣΔ modulator frequency: 400 kHz)
- Supports continuous conversion mode only.
- Allows OSR (oversampling ratio) settings.
- Includes peak hold functions (DC and AC).
- Includes a square root circuit to determine true RMS (Root-Mean-Square) values.
- Includes an averaging circuit.
  - Mean square or mean absolute is selectable.
  - Programmable average sample count for square and absolute values.
- Can generate interrupts.
  - Conversion completion interrupt
  - Conversion result overwrite error interrupt
  - Continuity status change detection interrupt

Figure 14.1.1 shows the whole block diagram of the DMM function. Figures 14.1.2 to 14.1.4 show the configurations of the A/D conversion unit, DMM power unit, and analog network unit, respectively.

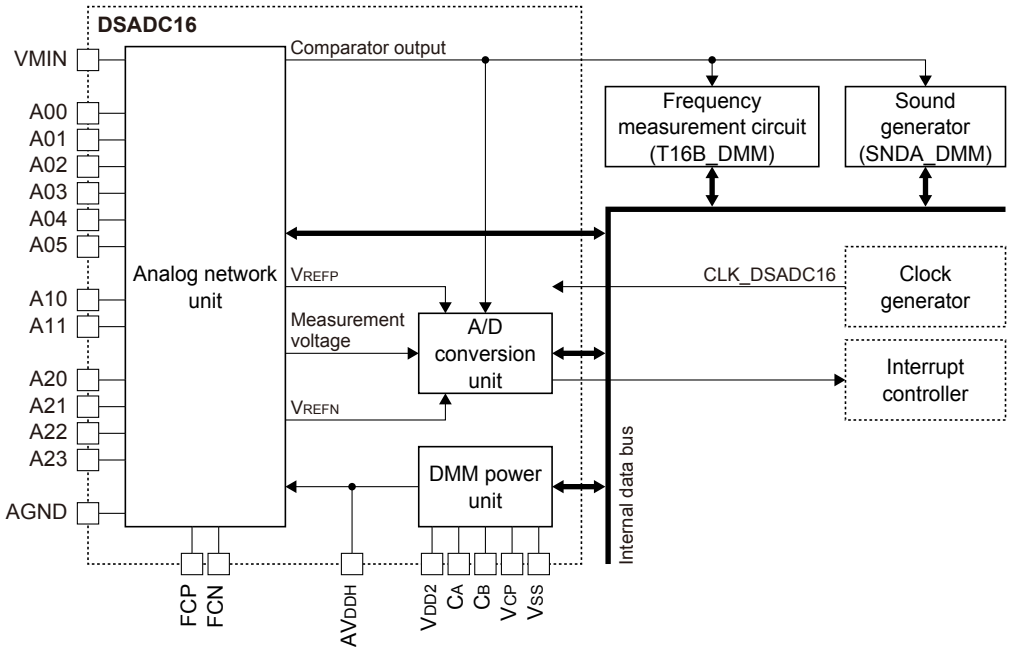


Figure 14.1.1 Whole Block Diagram of DMM Function

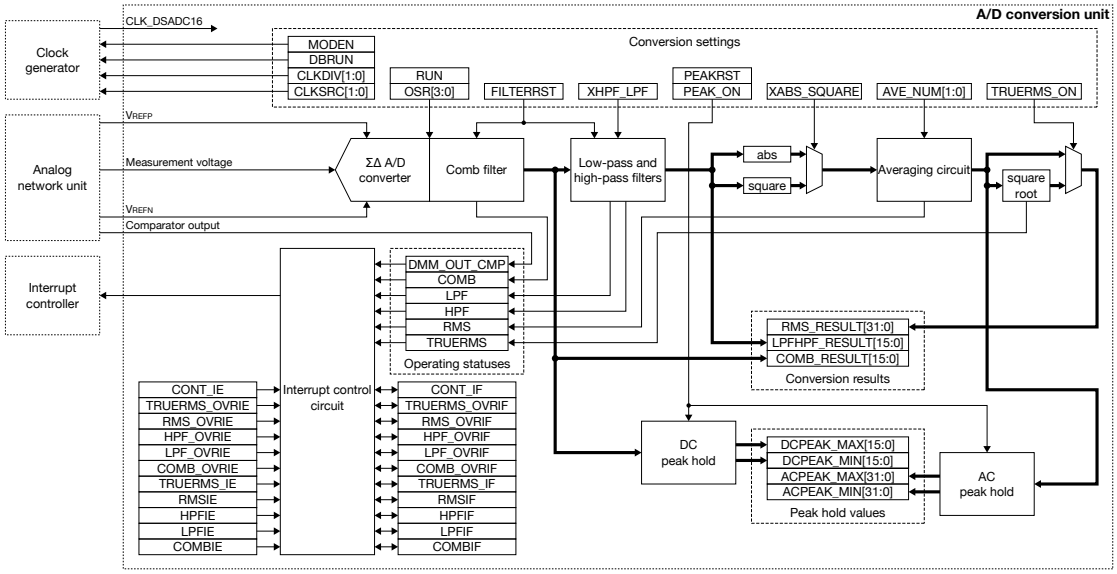


Figure 14.1.2 A/D Conversion Unit Configuration

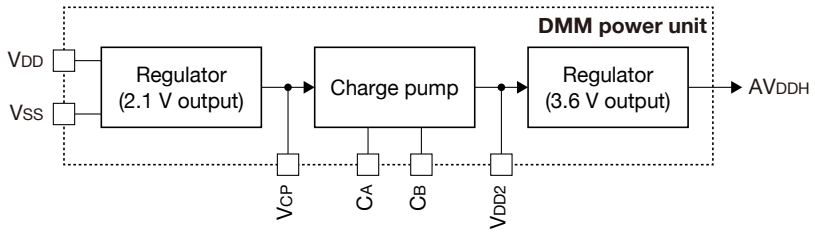


Figure 14.1.3 DMM Power Unit Configuration

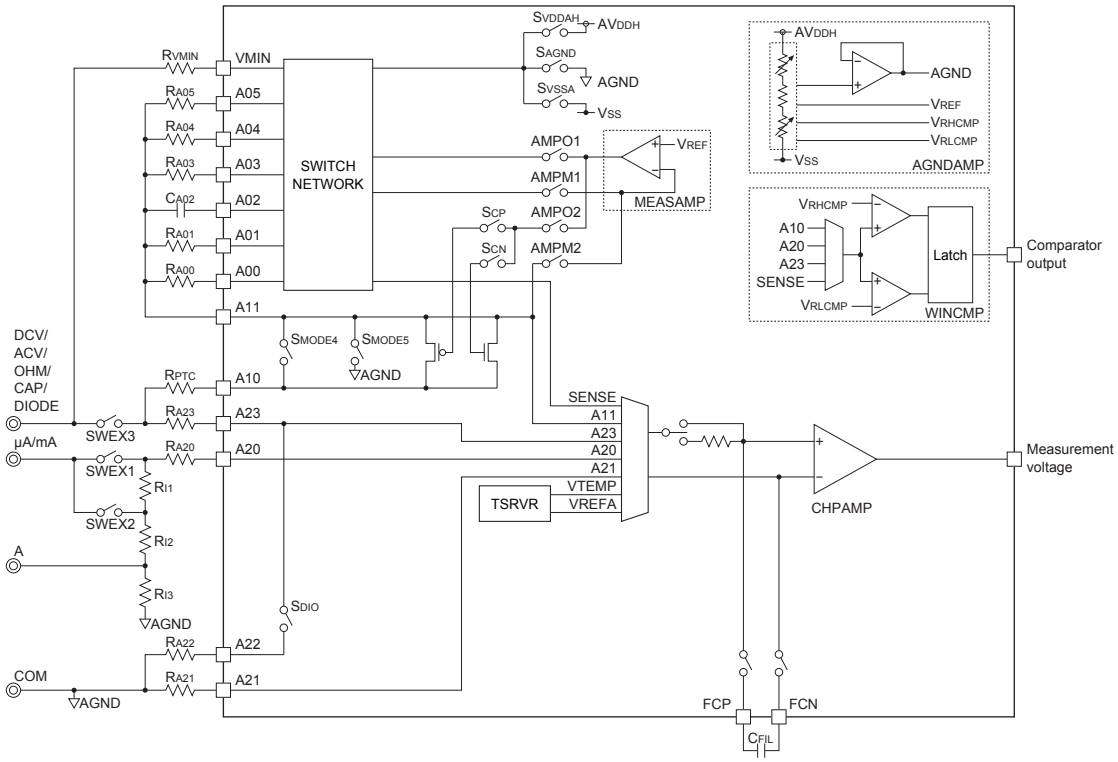


Figure 14.1.4 Analog Network Unit Configuration

Table 14.1.1 Functions of Analog Network Unit

Analog circuit	Function
AGNDAMP	Generates the analog GND voltage (AGND), WINCMP threshold voltages (V_{RHCMP} and V_{RLCMP}), and MEASAMP reference voltage (V_{REF}).
MEASAMP	Generates the reference voltage and current to be applied to the measuring object.
WINCMP	Determines the input voltage level and converts the input AC waveform into square waveform.
CHPAMP	Amplifies the input signal.
TSRVR	Internal temperature sensor
SWITCH NETWORK	Switches the signal paths according to the measurement mode.

14.2 Input/Output Pins and External Connections

14.2.1 List of Input/Output Pins

Table 14.2.1.1 lists the DSADC16 pins.

Table 14.2.1.1 List of DSADC16 Pins

Pin name	I/O	Initial status	Function
A00	A	–	DMM measurement pin
A01	A	–	DMM measurement pin
A02	A	–	DMM measurement pin
A03	A	–	DMM measurement pin
A04	A	–	DMM measurement pin
A05	A	–	DMM measurement pin
A10	A	–	DMM measurement pin
A11	A	–	DMM measurement pin
A20	A	–	DMM measurement pin
A21	A	–	DMM measurement pin
A22	A	–	DMM measurement pin
A23	A	–	DMM measurement pin
VMIN	A	–	DMM measurement pin
FCP	A	–	Filter capacitor connect pin
FCN	A	–	Filter capacitor connect pin
AV_{DDH}	A	–	Regulator output pin (3.6 V output, power supply for analog network unit)
AGND	A	–	Analog GND
V_{DD2}	A	–	3.6 V regulator power supply pin
C_A	A	–	Charge pump flying capacitor connect pin
C_B	A	–	Charge pump flying capacitor connect pin
V_{CP}	A	–	Regulator output pin (2.1 V output, power supply for charge pump)

14.2.2 External Connections

Figure 14.2.2.1 shows a connection diagram between DSADC16 and external components for configuring a DMM. Table 14.2.2.1 lists the external switch settings for each measurement mode.

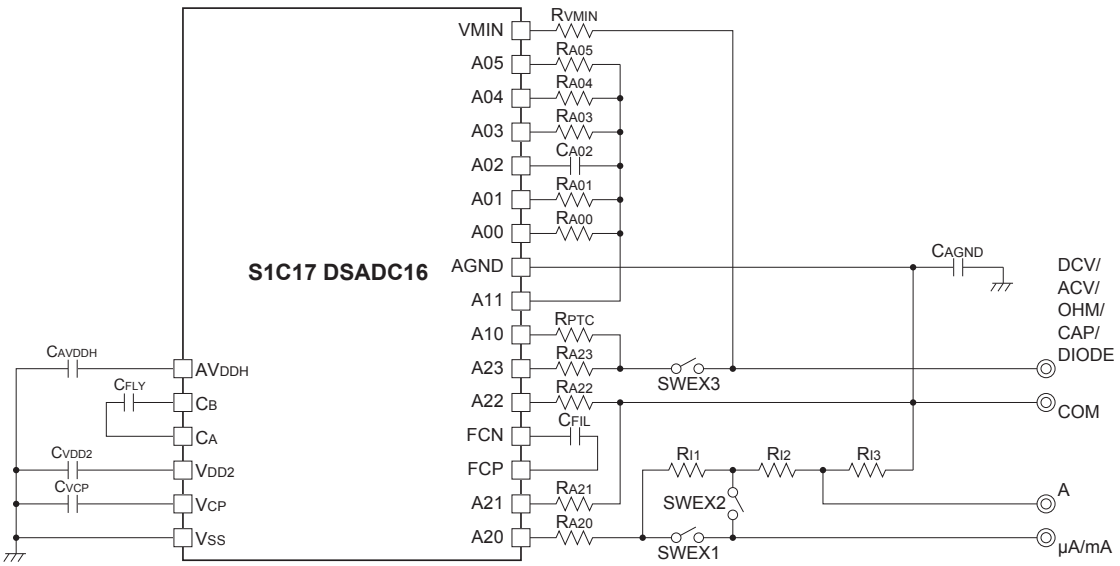


Figure 14.2.2.1 External Connection Diagram of Measurement Pins

Table 14.4.2.1 Measurement Pins and External Switch Settings

Measurement mode	Measurement range	Measurement pin	SWEX1	SWEX2	SWEX3
DC voltage measurement (DCV)	600 mV	DCV COM	OFF	OFF	ON
	6 V		OFF	OFF	OFF
	60 V				
	600 V				
	1,000 V				
AC voltage measurement (ACV)	600 mV	ACV COM	OFF	OFF	ON
	6 V		OFF	OFF	OFF
	60 V				
	600 V				
	1,000 V				
AC voltage and frequency measurement (Freq_ACV)	600 mV	ACV COM	OFF	OFF	ON
	6 V		OFF	OFF	OFF
	60 V				
	600 V				
	1,000 V				
Resistance measurement (OHM_CV, OHM_CC)	600 Ω	OHM COM	OFF	OFF	ON
	6 kΩ				
	60 kΩ				
	600 kΩ				
	6 MΩ				
	60 MΩ				
Continuity check (CONT)	–	OHM COM	OFF	OFF	ON
Diode V_F measurement (Diode)	–	DIODE COM	OFF	OFF	ON
Internal temperature measurement (Temp)	–	–	OFF	OFF	OFF
DC current measurement (DCI)	600 µA	µA/mA COM	ON	OFF	OFF
	6 mA				
	60 mA		OFF	ON	OFF
	600 mA				
	6 A	A COM	OFF	OFF	OFF
	10 A				
AC current measurement (ACI)	600 µA	µA/mA COM	ON	OFF	OFF
	6 mA				
	60 mA		OFF	ON	OFF
	600 mA				
	6 A	A COM	OFF	OFF	OFF
	10 A				
AC current and frequency measurement (Freq_ACI)	600 µA	µA/mA COM	ON	OFF	OFF
	6 mA				
	60 mA		OFF	ON	OFF
	600 mA				
	6 A	A COM	OFF	OFF	OFF
	10 A				
Capacitance measurement (CAP_CV, CAP_CC)	10 nF	CAP COM	OFF	OFF	ON
	100 nF				
	1 µF				
	10 µF				
	100 µF				
	1,000 µF				

14.3 Clock Settings

14.3.1 DSADC16 Operating Clock

When using DSADC16, the DSADC16 operating clock CLK_DSADC16 must be supplied to the DSADC16 from the clock generator. The CLK_DSADC16 supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to "Clock Generator" in the "Power Supply, Reset, and Clocks" chapter).
2. Set the following DSADC16CLK register bits:
   - DSADC16CLK.CLKSRC[1:0] bits (Clock source selection)
   - DSADC16CLK.CLKDIV[1:0] bits (Clock division ratio selection = Clock frequency setting)

The CLK_DSADC16 frequency should be set to 800 kHz.

14.3.2 Clock Supply in SLEEP Mode

When using DSADC16 during SLEEP mode, the DSADC16 operating clock CLK_DSADC16 must be configured so that it will keep supplying by writing 0 to the CLGOSC.*xxxx*SLPC bit for the CLK_SDSADC16 clock source. If the CLGOSC.*xxxx*SLPC bit for the CLK_DSADC16 clock source is 1, the CLK_DSADC16 clock source is de-activated during SLEEP mode and DSADC16 stops with the register settings maintained at those before entering SLEEP mode. After the CPU returns to normal mode, CLK_DSADC16 is supplied and the DSADC16 operation resumes.

14.3.3 Clock Supply in DEBUG Mode

The CLK_DSADC16 supply during DEBUG mode should be controlled using the DSADC16CLK.DBRUN bit. The CLK_DSADC16 supply to the DSADC16 is suspended when the CPU enters DEBUG mode if the DSADC-16CLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_DSADC16 supply resumes. Although the DSADC16 stops operating when the CLK_DSADC16 supply is suspended, the registers retain the status before DEBUG mode was entered. If the DSADC16CLK.DBRUN bit = 1, the CLK_DSADC16 supply is not suspended and DSADC16 will keep operating in DEBUG mode.

14.4 Operations

This section describes the DSADC16 operations. Note that a peripheral circuit is required for the operation in addition to the DSADC16 in the following measurements.

• For continuity check

The sound generator (SNDA_DMM) should be operated in addition to the DSADC16 for outputting the buzzer signal when a continuity status is detected. For more information, refer to the "Continuity Check Function" section.

• For frequency and capacitance measurements

The DMM 16-bit PWM Timers (T16B_DMM) must be configured as the frequency measurement circuit in addition to the DSADC16. For more information, refer to the "Frequency/Capacitance Measurement Function" section.

14.4.1 Initial Settings

The DSADC16 should be initialized with the procedure shown below.

1. Configure the DSADC16 operating clock (see the "Clock Settings" section).
2. Configure the following DSADC16CTL register bits:
   - DSADC16CTL.RANGESEL[2:0] bits (Set measurement range)
   - DSADC16CTL.FUNCSEL[3:0] bits (Set measurement mode)
3. Set the DSADC16CONF.MODEN bit to 1. (Enable DSADC16 operations)
4. Wait for at least 4 μs.
5. Write 0x7ff to the DSADC16IF register. (Clear interrupt flags)
6. Set the interrupt enable bits in the DSADC16IE register for the interrupt to be used to 1. (Enable interrupts)
7. Configure the following DSADC16CONF register bits: (Configure digital filter)
   - DSADC16CONF.AVE_NUM[1:0] bits (Set number of average samples)
   - DSADC16CONF.OSR[3:0] bits (Set oversampling ratio)
   - DSADC16CONF.TRUERMS_ON bit (Enable/disable square root circuit)
   - DSADC16CONF.XHPF_LPF bit (Select low-pass filter or high-pass filter)
   - DSADC16CONF.PEAK_ON bit (Enable/disable peak hold function)
   - DSADC16CONF.XABS_SQUARE bit (Select mean square or mean absolute)
8. Configure the following CHPCTL register bits: (Configure chopper amplifier)
   - CHPCTL.CHP_SET_CHP[2:0] bits (Set chopper cycle)
   - CHPCTL.CHP_SET_GAIN[2:0] bits (Set amplifier gain)
   - CHPCTL.CHP_SET_BIAS[2:0] bits (Set fully-differential amplifier bias current)
   - CHPCTL.CHP_SET_EN bit (Enable CHPAMP circuit)
9. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
10. Configure the following AFESUB register bits:
    - AFESUB.OHMCTL[3:0] bits (Set high resistance measurement)
    - AFESUB.CHP_SET_BIAS2[2:0] bits (Set buffer amplifier bias current)
11. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

Table 14.4.4.1 Recommended CHPAMP Settings

Measurement mode		Measure-ment range	CHPCTL register				AFESUB register	
			CHP_SET_ CHP[2:0]	CHP_SET_ GAIN[2:0]	CHP_SET_ BIAS[2:0]	CHP_SET_ EN	CHP_SET_ BIAS2[2:0]	OHMCTL[3:0]
DC voltage measurement (DCV)		600 mV	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		6 V	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		60 V	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		600 V	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		1,000 V	0x0 (2 cycles)	0x4 (6.5×)	0x4 (1×)	0x1	0x4 (1×)	–
AC voltage measurement (ACV)		600 mV	0x4 (none)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		6 V	0x4 (none)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		60 V	0x4 (none)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		600 V	0x4 (none)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		1,000 V	0x4 (none)	0x4 (6.5×)	0x4 (1×)	0x1	0x4 (1×)	–
DC current measurement (DCI)		600 µA	0x0 (2 cycles)	0x4 (6.5×)	0x4 (1×)	0x1	0x4 (1×)	–
		6 mA	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		60 mA	0x0 (2 cycles)	0x4 (6.5×)	0x4 (1×)	0x1	0x4 (1×)	–
		600 mA	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		6 A	0x0 (2 cycles)	0x4 (6.5×)	0x4 (1×)	0x1	0x4 (1×)	–
		10 A	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
AC current measurement (ACI)		600 µA	0x4 (none)	0x4 (6.5×)	0x4 (1×)	0x1	0x4 (1×)	–
		6 mA	0x4 (none)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		60 mA	0x4 (none)	0x4 (6.5×)	0x4 (1×)	0x1	0x4 (1×)	–
		600 mA	0x4 (none)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		6 A	0x4 (none)	0x4 (6.5×)	0x4 (1×)	0x1	0x4 (1×)	–
		10 A	0x4 (none)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
Resistance measurement (OHM_CV, OHM_CC)	CV	600 Ω	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		6k Ω	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		60 kΩ	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
	CC	600 Ω	0x0 (2 cycles)	0x2 (1×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		6k Ω	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		60 kΩ	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		600 kΩ	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		6 MΩ	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	0x8
		60 MΩ	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	0x8
Continuity check (CONT)	CV	–	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
	CC	–	0x0 (2 cycles)	0x2 (1×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
Capacitance measurement (CAP_CV, CAP_CC)		10 nF	–	–	–	–	–	–
		100 nF	–	–	–	–	–	–
		1 µF	–	–	–	–	–	–
		10 µF	–	–	–	–	–	–
		100 µF	–	–	–	–	–	–
		1,000 µF	–	–	–	–	–	–
Diode VF measurement (Diode)		–	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
AC voltage and frequency measurement (Freq_ACV)		600 mV	0x4 (none)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		6 V	0x4 (none)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		60 V	0x4 (none)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		600 V	0x4 (none)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		1,000 V	0x4 (none)	0x4 (6.5×)	0x4 (1×)	0x1	0x4 (1×)	–
AC current and frequency measurement (Freq_ACI)		600 µA	0x4 (none)	0x4 (6.5×)	0x4 (1×)	0x1	0x4 (1×)	–
		6 mA	0x4 (none)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		60 mA	0x4 (none)	0x4 (6.5×)	0x4 (1×)	0x1	0x4 (1×)	–
		600 mA	0x4 (none)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
		6 A	0x4 (none)	0x4 (6.5×)	0x4 (1×)	0x1	0x4 (1×)	–
		10 A	0x4 (none)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–
Internal temperature measurement (Temp)		–	0x0 (2 cycles)	0x0 (0.6×)	0x3 (0.25×)	0x1	0x3 (0.25×)	–

14.4.2 Starting Measurement

The following shows the measurement procedure using the DSADC16:

1. Set the external switches (see Figure 14.2.2.1 and Table 14.2.2.1).
2. Connect the object to be measured to the measurement pins (see Figure 14.2.2.1 and Table 14.2.2.1).
3. Write 1 to the DSADC16INIT.RUN bit. (Start measurement)
4. Wait for a conversion completion interrupt to occur.
5. Read the conversion result.
6. Write 0x7ff to the DSADC16IF register. (Clear interrupt flags)
7. Repeat Steps 4 to 6.

Writing 1 to the DSADC16INIT.RUN bit starts converting. The conversion is executed successively and the interrupt flag is set when a conversion has finished. Read the conversion result and clear the interrupt flag before the next conversion finishes. The conversion result is overwritten if the next conversion has finished before being read. Writing 0 to the DSADC16INIT.RUN bit stops converting. After the conversion has completed or terminated, reset the digital filter by writing 1 to the DSADC16INIT.FILTERRST bit. This reset process takes 4 μs. The measurement should be performed several times until stabilized results are obtained, as the digital filter process performs averaging.

14.4.3 Switching Measurement Mode and Range

The measurement mode and the measurement range can be set using the DSADC16CTL.FUNCSEL[3:0] bits and the DSADC16CTL.RANGESEL[2:0] bits, respectively. By setting these control bits, the analog network unit starts operating. For details of the measurement mode and measurement range, refer to the "Operations of Analog Network Unit" section.
The register to which the results are stored and the interrupt flags are different depending on the measurement mode. Table 14.4.3.2 list these differences between the measurement modes and the recommended settings of the OSR and number of average samples.

Switching measurement mode

The following shows the procedure to switch the measurement mode during measurement:

1. Write 0 to the DSADC16INIT.RUN bit. (Stop measurement)
2. Set the DSADC16CONF.MODEN bit to 0. (Disable DSADC16 operations)
3. Configure the following DSADC16CTL register bits:
   - DSADC16CTL.RANGESEL[2:0] bits (Set measurement range)
   - DSADC16CTL.FUNCSEL[3:0] bits (Set measurement mode)
4. Set the DSADC16CONF.MODEN bit to 1. (Enable DSADC16 operations)
5. Write 1 to the DSADC16INIT.FILTERRST bit. (Clear digital filter registers)
   The digital filter registers are cleared after 4 μs have elapsed from this writing.
6. Wait for at least 4 μs (wait time required between Steps 4 to 7).
7. Write 0x7ff to the DSADC16IF register. (Clear interrupt flags)
8. Set the interrupt enable bits in the DSADC16IE register for the interrupt to be used to 1. (Enable interrupts)
9. Configure the following DSADC16CONF register bits: (Configure digital filter)
   - DSADC16CONF.AVE_NUM[1:0] bits (Set number of average samples)
   - DSADC16CONF.OSR[3:0] bits (Set oversampling ratio)
   - DSADC16CONF.TRUERMS_ON bit (Enable/disable square root circuit)
   - DSADC16CONF.XHPF_LPF bit (Select low-pass filter or high-pass filter)
   - DSADC16CONF.PEAK_ON bit (Enable/disable peak hold function)
   - DSADC16CONF.XABS_SQUARE bit (Select mean square or mean absolute)

10. Configure the following CHPCTL register bits: (Configure chopper amplifier)
    - CHPCTL.CHP_SET_CHP[2:0] bits (Set chopper cycle)
    - CHPCTL.CHP_SET_GAIN[2:0] bits (Set amplifier gain)
    - CHPCTL.CHP_SET_BIAS[2:0] bits (Set fully-differential amplifier bias current)
    - CHPCTL.CHP_SET_EN bit (Enable CHPAMP circuit)
11. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
12. Configure the following AFESUB register bits:
    - AFESUB.OHMCTL[3:0] bits (Set high resistance measurement)
    - AFESUB.CHP_SET_BIAS2[2:0] bits (Set buffer amplifier bias current)
13. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

Switching measurement range

The following shows the procedure to switch the measurement range during measurement:

1. Write 1 to the DSADC16INIT.FILTERRST bit. (Clear digital filter registers)
2. Set the DSADC16CTL.RANGESEL[2:0] bits. (Set measurement range)
3. Configure the following CHPCTL register bits: (Configure chopper amplifier)
    - CHPCTL.CHP_SET_CHP[2:0] bits (Set chopper cycle)
    - CHPCTL.CHP_SET_GAIN[2:0] bits (Set amplifier gain)
    - CHPCTL.CHP_SET_BIAS[2:0] bits (Set fully-differential amplifier bias current)
    - CHPCTL.CHP_SET_EN bit (Enable CHPAMP circuit)
4. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
5. Configure the following AFESUB register bits:
    - AFESUB.OHMCTL[3:0] bits (Set high resistance measurement)
    - AFESUB.CHP_SET_BIAS2[2:0] bits (Set buffer amplifier bias current)
6. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

Table 14.4.3.1 Measurement Mode and Range Settings

Mode/ Range settings		DSADC16CTL.FUNCSEL[3:0] bits													
		0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd
		DCV	ACV	DCI	ACI	OHM _CC	OHM _CV	CONT	CAP _CC	CAP _CV	Diode	Freq _ACV	Freq _ACI	Temp	OFF
DSADC16CTL. RANGESEL[2:0] bits	0x0	600 mV	600 mV	600 µA	600 µA	600 Ω	600 Ω	CV	×	10 nF	Diode	600 mV	600 µA	Temp	OFF
	0x1	6 V	6 V	6 mA	6 mA	6 kΩ	6 kΩ	CC	×	100 nF	×	6 V	6 mA	×	×
	0x2	60 V	60 V	60 mA	60 mA	60 kΩ	60 kΩ	×	1 µF	×	×	60 V	60 mA	×	×
	0x3	600 V	600 V	600 mA	600 mA	600 kΩ	×	×	10 µF	×	×	600 V	600 mA	×	×
	0x4	1,000 V	1,000 V	6 A	6 A	6 MΩ	×	×	100 µF	×	×	1,000 V	6 A	×	×
	0x5	×	×	10 A	10 A	60 MΩ	×	×	1,000 µF	×	×	×	10 A	×	×
	0x6	×	×	×	×	×	×	×	×	×	×	×	×	×	×
	0x7	×	×	×	×	×	×	×	×	×	×	×	×	×	×

×: Setting prohibited (Normal IC operations cannot be guaranteed if these values are set.)

Table 14.4.3.2 Differences and Recommended Settings by Each Mode

Measurement mode	Filter used	Result register	Interrupt flag*1/ enable bit*2	DSADC16CONF register bit settings
DC voltage measurement (DCV)	Comb filter	DSADC16COMB	COMBIF/ COMBIE	OSR[3:0] = 0xa (OSR: 16,384)
AC voltage measurement (ACV)	Comb filter, LPF or HPF, Averaging circuit, Square root circuit	DSADC16RMS1, DSADC16RMS2	TRUERMSIF/ TRUERMSIE	AVE_NUM[1:0] = 0x3 (2,048 samples) OSR[3:0] = 0x0 (OSR: 16) TRUERMS_ON = 1 (True RMS value) XHPF_LPF = 0 (HPF) XABS_SQUARE = 1 (Mean square)
DC current measurement (DCI)	Comb filter	DSADC16COMB	COMBIF/ COMBIE	OSR[3:0] = 0xa (OSR: 16,384)
AC current measurement (ACI)	Comb filter, LPF or HPF, Averaging circuit, Square root circuit	DSADC16RMS1, DSADC16RMS2	TRUERMSIF/ TRUERMSIE	AVE_NUM[1:0] = 0x3 (2,048 samples) OSR[3:0] = 0x0 (OSR: 16) TRUERMS_ON = 1 (True RMS value) XHPF_LPF = 0 (HPF) XABS_SQUARE = 1 (Mean square)
Resistance measurement (OHM_CV, OHM_CC)	Comb filter	DSADC16COMB	COMBIF/ COMBIE	OSR[3:0] = 0xa (OSR: 16,384)
Continuity check (CONT)	Comb filter	DSADC16COMB	CONTIF/ CONTIE	OSR[3:0] = 0xa (OSR: 16,384)
Capacitance measurement (CAP_CV, CAP_CC)*3	–	–	–	–
Diode VF measurement (Diode)	Comb filter	DSADC16COMB	COMBIF/ COMBIE	OSR[3:0] = 0xa (OSR: 16,384)
AC voltage and frequency measurement (Freq_ACV)*3	Comb filter, LPF or HPF, Averaging circuit, Square root circuit	DSADC16RMS1, DSADC16RMS2	TRUERMSIF/ TRUERMSIE	AVE_NUM[1:0] = 0x3 (2,048 samples) OSR[3:0] = 0x0 (OSR: 16) TRUERMS_ON = 1 (True RMS value) XHPF_LPF = 0 (HPF) XABS_SQUARE = 1 (Mean square)
AC current and frequency measurement (Freq_ACI)*3	Comb filter, LPF or HPF, Averaging circuit, Square root circuit	DSADC16RMS1, DSADC16RMS2	TRUERMSIF/ TRUERMSIE	AVE_NUM[1:0] = 0x3 (2,048 samples) OSR[3:0] = 0x0 (OSR: 16) TRUERMS_ON = 1 (True RMS value) XHPF_LPF = 0 (HPF) XABS_SQUARE = 1 (Mean square)
Internal temperature measurement (Temp)	Comb filter	DSADC16COMB	COMBIF/ COMBIE	OSR[3:0] = 0xa (OSR: 16,384)

*1 DSADC16IF register
*2 DSADC16IE register
*3 For the capacitance and frequency measurements, refer to the “Details of Analog Network Unit” section.

14.4.4 DC Measurement Function

In DC measurement, the Comb filter only is used as the digital filter.

Comb filter

The Comb filter converts the over-sampled measurement voltages into 16-bit values. The OSR should be set using the DSADC16CONF.OSR[3:0] bits to use the Comb filter. Before changing the OSR, terminate measurement by setting the DSADC16INIT.RUN bit to 0. After changing the OSR, reset the filter registers by writing 1 to the DSADC16INIT.FILTERRST bit. Do not write a reserved value to the DSADC16CONF.OSR[3:0] bits, otherwise normal IC operation cannot be guaranteed.

The sampling frequency is changed according to the set OSR (e.g. when the OSR is set to 16, the sampling frequency becomes 25 kHz [= 400 kHz/16]).

Table 14.4.4.1 OSR Settings

DSADC16CONF. OSR[3:0] bits	OSR Settings	Sampling frequency
0xb–0xf	Reserved	
0xa	16,384	24.41 Hz
0x9	8,192	48.83 Hz
0x8	4,096	97.66 Hz
0x7	2,048	195.31 Hz
0x6	1,024	390.63 Hz
0x5	512	781.25 Hz
0x4	256	1.56 kHz
0x3	128	3.12 kHz
0x2	64	6.25 kHz
0x1	32	12.5 kHz
0x0	16	25 kHz

14.4.5 AC Measurement Function

In AC measurement, the measuring signal passes through the Comb filter and low-pass filter (LPF) or high-pass filter (HPF), and is sent to the averaging circuit to convert into a square or absolute value and calculate the average value for obtaining the RMS value. The average value is sent to the square root circuit to calculate the true RMS value. For the Comb filter, refer to the "DC Measurement Function" section.

LPF (low-pass filter)/HPF (high-pass filter)

LPF is a digital filter to remove high-frequency noise. The cut-off frequency of this LPF is 1,250 Hz.
HPF is a digital filter to remove DC component (low frequency). The cut-off frequency of this HPF is 1 Hz.

LPF/HPF selection

The DSADC16CONF.XHPF_LPF bit is used to select the filter of which the output results are used for averaging. The results that have been processed in the selected filter are loaded to the DSADC16LPFHPF.LPFHPFRESULT[15:0] bits.

DSADC16CONF.XHPF_LPF bit = 1: LPF only
DSADC16CONF.XHPF_LPF bit = 0: LPF and HPF

Averaging circuit

The averaging circuit converts the digital filter processing result into a square or absolute value and calculates the mean value.

Mean square/mean absolute selection

The DSADC16CONF.XABS_SUQUARE bit is used to select either mean square or mean absolute. To obtain true RMS values, select mean square.

DSADC16CONF.XABS_SUQUARE bit = 1: Mean square
DSADC16CONF.XABS_SUQUARE bit = 0: Mean absolute

Setting number of average samples

The DSADC16CONF.AVE_NUM[1:0] bits are used to set the number of samples to average the square or absolute value. Before altering the DSADC16CONF.AVE_NUM[1:0] bits, stop measurement by setting the DSADC16INIT.RUN bit to 0. After altering the DSADC16CONF.AVE_NUM[1:0] bits, write 1 to the DSADC16INIT.FILTERRST bit to reset the digital filter registers.

Table 14.4.5.1 Setting Number of Average Samples

DSADC16CONF.AVE_NUM[1:0] bits	Number of average samples
0x3	2,048
0x2	1,024
0x1	512
0x0	256

Square root circuit

The square root circuit calculates the true RMS value from the mean value output from the averaging circuit. The square root circuit is enabled by writing 1 to the DSADC16CONF.TRUERMS_ON bit. The following shows the specifications of the square root circuit:

- Input value (A): N-bit wide unsigned integer
- Calculation result (X): N ÷ 2-bit wide unsigned integer ($X = \sqrt{|A|}$)
- Handling after the decimal point: Round off
- Number of processing clocks: 2 clocks

AC measurement results

In AC measurement modes, the values loaded to the result registers (DSADC16RMS1 and DSADC16RMS2 registers) can be selected from three types by setting the DSADC16CONF.XABS_SUQUARE and DSADC16CONF.TRUERMS_ON bits.

Table 14.4.5.2 AC Measurement Results

DSADC16CONF. XABS_SUQUARE bit	DSADC16CONF. TRUERMS_ON bit	Measurement results
1	1	16-bit true RMS value
1	0	32-bit mean square value
0	1	Reserved
0	0	16-bit mean absolute value (default)

14.4.6 Frequency/Capacitance Measurement Function

Frequencies are measured using T16B_DMM, which has timer data comparison and capture functions, as the frequency measurement circuit. By using three T16B_DMM channels, the DSADC16 comparator output (T16B_DMM input pulses) is converted into the value equivalent to the frequency. Also in capacitance measurement, the capacitance value is calculated from the frequency value obtained by the T16B_DMM. For operations of the T16B_DMM as a 16-bit PWM timer, refer to the "DMM 16-bit PWM Timers" chapter.
Each T16B_DMM channel is used as follows:

• T16B_DMM Ch.0

T16B_DMM Ch.0 is set to comparator mode and generates the reference measurement period by counting the high-precision crystal oscillator output clock (32.768 kHz).

• T16B_DMM Ch.1

The DSADC16 comparator output and the T16B_DMM Ch.0 TOUT00 output are connected to the T16B_DMM Ch.1 EXCL10 input and the CAP10 input, respectively, in the IC. T16B_DMM Ch.1 is set to capture mode and counts the comparator output (input pulses). By using the CAP10 input as a trigger to capture the counter value, the number of input pulses for the measurement period generated by T16B_DMM Ch.0 can be obtained. The comparator output signal frequency can be calculated from this counter value (Case 1).

• T16B_DMM Ch.2

A small counter value (low frequency) decreases the accuracy of the Case 1 results obtained using T16B_DMM Ch.1. To obtain accurate results, T16B_DMM Ch.2 is run simultaneously.
The DSADC16 comparator output is connected to the T16B_DMM Ch.2 CAP20 input in the IC. T16B_DMM Ch.2 is set to capture mode and counts the high-precision crystal oscillator output clock (32.768 kHz). By using the CAP20 input as a trigger to capture the counter value, the number of the 32.768 kHz clock equivalent to the comparator output clock cycle can be obtained. The comparator output signal frequency can be calculated from this counter value (Case 2).

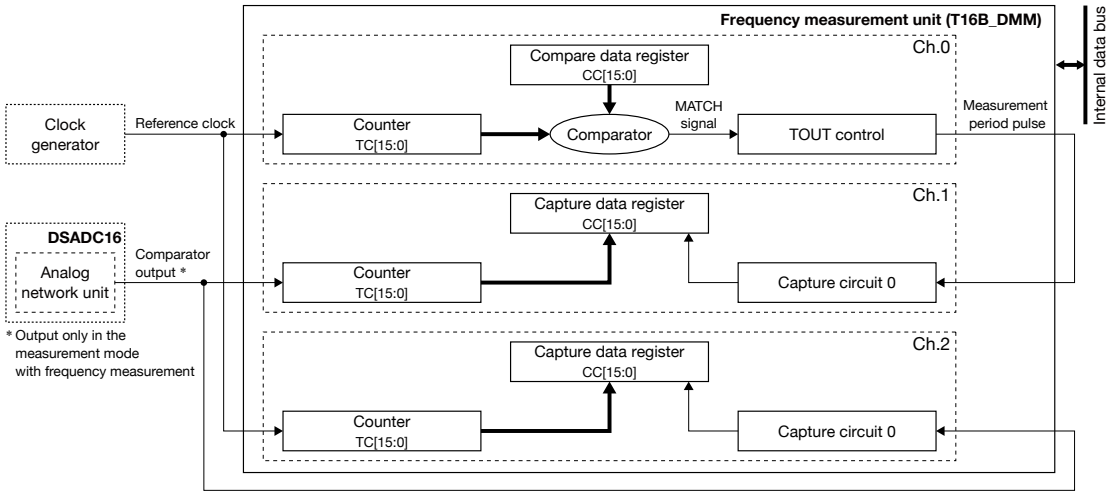


Figure 14.4.6.1 Frequency Measurement Circuit Block Diagram

Settings for frequency measurement

• T16B_DMM Ch.0 (comparator mode)

1. Configure the T16B_DMM Ch.0 operating clock.
   - Start the OSC1 oscillator.
   - Set the T16B0CLK.CLKSRC[2:0] bits to 0x1. (Select OSC1 as clock source)
   - Set the T16B0CLK.CLKDIV[3:0] bits to 0x0. (Set clock division ratio to 1/1)
2. Set the T16B0CTL.MODEN bit to 1. (Enable T16B_DMM Ch.0 operations)
3. Set the following bits to configure the comparator function:
   - Set the T16B0CCCTL0.CCMD bit to 0. (Set comparator mode)
   - Set the T16B0CCCTL0.CBUFMD[2:0] bits to 0x4. (Configure compare buffer)
4. Set the following bits to configure the TOUT00 generation condition.
   - Set the T16B0CCCTL0.TOUTMT bit to 0. (Select MATCH and MAX signals for TOUT generation)
   - Set the T16B0CCCTL0.TOUTMD[2:0] bits to 0x4. (Select toggle mode for TOUT generation)
   - Set the T16B0CCCTL0.TOUTINV bit to 0. (Select normal (active high) TOUT signal polarity)
5. Set the T16B0CCR0 register. (Set counter comparison value)

Table 14.4.6.1 Recommended Measurement Period Settings

Measurement mode	Measurement range	Measurement period (T16B0CCR0 setting value)
AC voltage and frequency measurement (Freq_ACV)	600 mV	0.5 seconds (0x1fff)
	6 V	
	60 V	
	600 V	
	1,000 V	
AC current and frequency measurement (Freq_ACI)	600 µA	0.5 seconds (0x1fff)
	6 mA	
	60 mA	
	600 mA	
	6 A	
	10 A	
Capacitance measurement (CAP_CV)	10 nF	0.5 seconds (0x1fff)
	100 nF	
Capacitance measurement (CAP_CC)	1 µF	0.5 seconds (0x1fff)
	10 µF	
	100 µF	
	1,000 µF	2 seconds (0x7fff)

6. Set the following bits for using the compare interrupt:
   - Write 1 to the T16B0INTF.CMPCAP0IF bit. (Clear interrupt flags)
   - Set the 16B0INTE.CMPCAP0IE bit to 1. (Enable interrupts)
7. Set the following T16B0CTL register bits:
   - Set the T16B0CTL.CNTMD[1:0] bits to 0x0. (Set up-count mode)
   - Set the T16B0CTL.ONEST bit to 0. (Set repeat mode)
   - Set the T16B0CTL.PRESET bit to 1. (Reset counter)

• T16B_DMM Ch.1 (capture mode)

1. Configure the T16B_DMM Ch.1 operating clock.
   - Set the T16B1CLK.CLKSRC[2:0] bits to 0x4. (Select EXCL10 as clock source)
   - Set the T16B0CLK.CLKDIV[3:0] bits to 0x0. (Set clock division ratio to 1/1)
2. Set the T16B1CTL.MODEN bit to 1. (Enable T16B_DMM Ch.1 operations)
3. Set the following T16B1CCCTL0 register bits:
   - Set the T16B1CCCTL0.CCMD bit to 1. (Set capture mode)
   - Set the T16B1CCCTL0.SCS bit to 1. (Set synchronous mode)
   - Set the T16B1CCCTL0.CAPIS[1:0] bits to 0x0. (Set external as trigger signal)
   - Set the T16B1CCCTL0.CAPTRG[1:0] bits to 0x1. (Set rising edge as trigger edge)
4. Set the T16B1MC register to 0xffff. (Set MAX counter data)
5. Set the following bits for using the capture interrupt:
   - Write 1 to the T16B1INTF.CMPCAP0IF bit. (Clear interrupt flags)
   - Set the T16B1INTE.CMPCAP0IE bit to 1. (Enable interrupts)
6. Set the following T16B1CTL register bits:
   - Set the T16B1CTL.CNTMD[1:0] bits to 0x0. (Set up-count mode)
   - Set the T16B1CTL.ONEST bit to 0. (Set repeat mode)
   - Set the T16B1CTL.PRESET bit to 1. (Reset counter)

• T16B_DMM Ch.2 (capture mode)

1. Configure the T16B_DMM Ch.2 operating clock.
   - Set the T16B2CLK.CLKSRC[2:0] bits to 0x1. (Select OSC1 as clock source)
   - Set the T16B0CLK.CLKDIV[3:0] bits to 0x0. (Set clock division ratio to 1/1)
2. Set the T16B2CTL.MODEN bit to 1. (Enable T16B_DMM Ch.2 operations)
3. Set the following T16B2CCCTL0 register bits:
   - Set the T16B2CCCTL0.CCMD bit to 1. (Set capture mode)
   - Set the T16B2CCCTL0.SCS bit to 1. (Set synchronous mode)
   - Set the T16B2CCCTL0.CAPIS[1:0] bits to 0x0. (Set external as trigger signal)
   - Set the T16B2CCCTL0.CAPTRG[1:0] bits to 0x1. (Set rising edge as trigger edge)
4. Set the T16B2MC register to 0xffff. (Set MAX counter data)
5. Set the following bits for using the capture interrupt:
   - Write 1 to the T16B2INTF.CMPCAP0IF bit. (Clear interrupt flags)
   - Set the 16B2INTE.CMPCAP0IE bit to 1. (Enable interrupts)
6. Set the following T16B2CTL register bits:
   - Set the T16B2CTL.CNTMD[1:0] bits to 0x0. (Set up-count mode)
   - Set the T16B2CTL.ONEST bit to 0. (Set repeat mode)
   - Set the T16B2CTL.PRESET bit to 1. (Reset counter)

Starting frequency measurement

Run the T16B_DMM channels in the order shown below to start frequency measurement.

1. Start T16B_DMM Ch.1 by setting the T16B1CTL.RUN bit to 1.
2. Start T16B_DMM Ch.2 by setting the T16B2CTL.RUN bit to 1.
3. Start T16B_DMM Ch.0 by setting the T16B0CTL.RUN bit to 1.
4. Start measurement using the DSADC16 (refer to the "Starting Measurement" section).

Frequency calculation

1. When a capture interrupt has occurred from T16B_DMM Ch.1, read the T16B1CCR0 register value (T16B_DMM Ch.1 captured value: n_{IN1}). When a capture interrupt has occurred from T16B_DMM Ch.2, read the T16B2CCR0 register value (T16B_DMM Ch.2 captured value: n_{E1}). Store them in a memory as the initial values.
2. When the succeeding capture interrupt has occurred from T16B_DMM Ch.1, read the T16B1CCR0 register value (T16B_DMM Ch.1 captured value: n_{IN2}). When the succeeding capture interrupt has occurred from T16B_DMM Ch.2, read the T16B2CCR0 register value (T16B_DMM Ch.2 captured value: n_{E2}). Store them as the updated values.
3. If the difference between the initial and updated values of T16B_DMM Ch.1 ($n_{IN2} - n_{IN1}$) is 4,000 or more, use Case 1 to calculate the frequency. Or use Case 2 if it is less than 4,000. This value (4,000) is the threshold value to select the calculation method and it is predefined to secure the calculation accuracy within 0.1%.

$$\text{Case 1: } F\,[\text{Hz}] = \frac{f_R \times (n_{IN2} - n_{IN1})}{n_R} \qquad \text{(Eq. 14.1)}$$

$$\text{Case 2: } F\,[\text{Hz}] = \frac{f_R \times (n_{IN2} - n_{IN1})}{n_{E2} - n_{E1}} \qquad \text{(Eq. 14.2)}$$

Where

f_R (Reference clock frequency) = 32,768 Hz

n_R (Measurement period) = 2 × T16B0CCR0 register setting value

Capacitance calculation

Capacitance values are calculated from the frequency with the equations shown below. There are two measurement methods in capacitance measurement mode, CV (Constant Voltage) method and CC (Constant Current) method, and the method used is determined with the measurement range selected. The equation to be used should be selected according to the measurement range as well.

• Measurement in 10 nF/100 nF range (CV method)

$$C_{in}\,[F] = \frac{1}{f \times R_{A0x} \times \left\{\ln\left(1 - \frac{V_{RLCMP}}{AV_{DDH}}\right) + \ln\left(\frac{V_{RHCMP}}{AV_{DDH}}\right) - \ln\left(1 - \frac{V_{RHCMP}}{AV_{DDH}}\right) - \ln\left(\frac{V_{RLCMP}}{AV_{DDH}}\right)\right\}} \qquad \text{(Eq. 14.3)}$$

Unit: R_{A0x} (Ω), f (Hz), V_{RHCMP} (V), V_{RLCMP} (V), AV_{DDH} (V)

• Measurement in 1 µF/10 µF/100 µF/1,000 µF range (CC method)

$$C_{in}\,[F] = \frac{1}{\frac{f \times R_{A0x} \times (V_{RHCMP} - V_{RLCMP})}{AV_{DDH} - V_{REF\,(charging)}} + \frac{f \times R_{A0x} \times (V_{RHCMP} - V_{RLCMP})}{V_{REF\,(discharging)}}} \qquad \text{(Eq. 14.4)}$$

Unit: R_{A0x} (Ω), f (Hz), AV_{DDH} (V), V_{RHCMP} (V), V_{RLCMP} (V), V_{REF} (V)

14.4.7 Peak Hold Function

The peak hold function holds the maximum and minimum values of the measurement results. The peak hold registers are separated for storing the maximum value and for the minimum value and they are provided individually for DC measurements and AC measurements. In a DC measurement mode, the measurement results of the Comb filter are stored. In an AC measurement mode, the mean square or mean absolute results are stored.
The peak hold function is enabled simultaneously for both DC and AC measurements by writing 1 to the DSADC16CONF.PEAK_ON bit. After this, the peak hold register is rewritten when the current maximum or minimum value is updated with a measurement result. The peak hold registers are not cleared and hold the stored value even if the DSADC16CONF.PEAK_ON bit is altered. The peak hold function can be reset by writing 1 to the DSADC16INIT.PEAKRST bit. The reset processing is completed after 4 μs from this writing and the peak hold registers revert to the initial value.

Table 14.4.7.1 AC Measurement Peak Hold Results

DSADC16CONF. PEAK_ON bit	DSADC16CONF. XABS_SUQUARE bit	DSADC16CONF. TRUERMS_ON bit	Peak hold results
1	1	1	Reserved
	1	0	32-bit mean square value
	0	1	Reserved
	0	0	16-bit mean absolute value (default)

Make several measurements before enabling the pack hold function.

14.4.8 Continuity Check Function

The continuity check function can enable the sound generator to output the buzzer signal while a continuity status is being detected. It is necessary to configure the sound generator when using this function. For more information on controlling the buzzer output and enabling the DMM controller linking function, refer to the “Sound Generator” chapter.
When a continuity status is detected, the CMPOUT.DMM_OUT_CMP bit is set to 1; when a non-continuity status is detected, the CMPOUT.DMM_OUT_CMP bit is set to 0.

14.4.9 Internal Temperature Measurement Function

The analog network unit includes a temperature sensor (TSRVR) allowing measurement of temperature inside the IC. The internal temperature can be determined from the A/D conversion value with the following equation:

$$\text{Measured temperature [°C]} = \frac{\text{COMB\_RESULT - TSRVR\_TEMP}}{65.54} + 25 \qquad \text{(Eq. 14.5)}$$

Where
COMB_RESULT: DSADC16COMB.COMB_RESULT[15:0] bit vbalue (signed 16-bit measured value)
TSRVR_TEMP: TSRVRTEMP.TSRVR_TEMP[15:0] bit vbalue (temperature correction data)

14.5 Details of Analog Network Unit

The analog network unit supports various measurements according to the internal analog switch settings and analog circuit parameter configurations. According to the measurement mode, the analog network unit outputs measurement voltages and the comparator output signal. The output signals are processed with the digital filters and T16B_DMM or SNDA_DMM according to the measurement mode.

14.5.1 DC Voltage Measurement

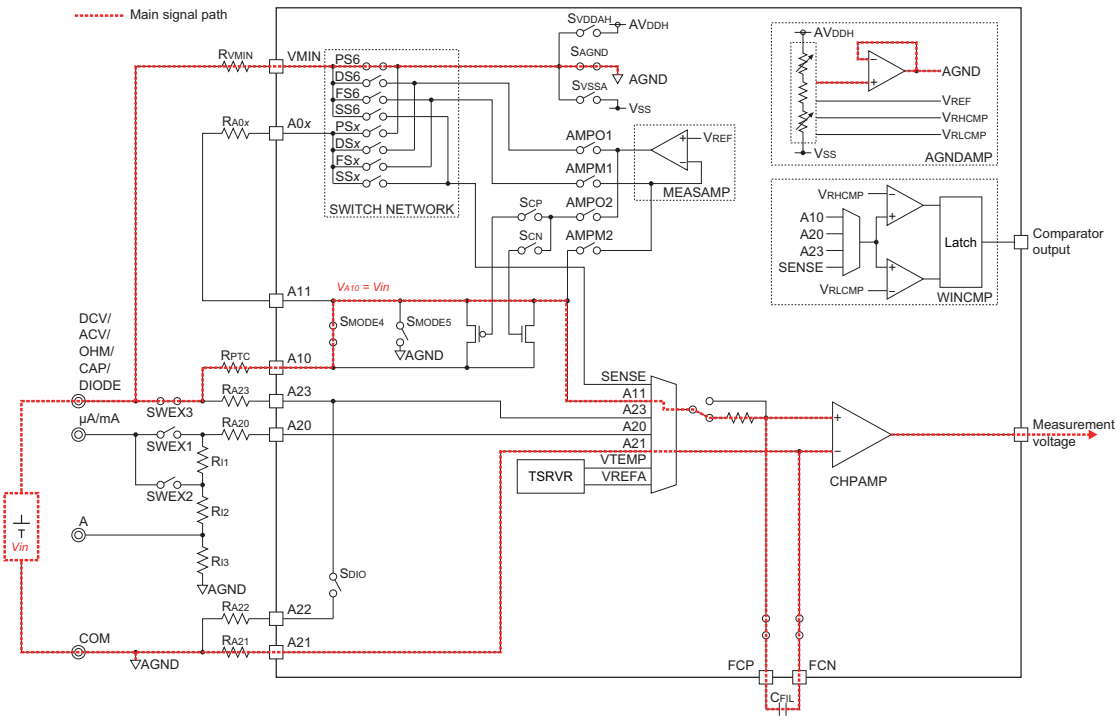
In DC voltage measurement mode, the DC voltage that is applied between the DCV (+) and COM (GND) pins can be measured. The measurement range can be selected from the following five ranges: 600 mV, 6 V, 60 V, 600 V, and 1,000 V. Table 14.5.1.1 shows the external switch settings. Figure 14.5.1.1 shows the main signal paths in the analog network unit. The A0*x* pin and resistor R_{A0x} in Figure 14.5.1.1 indicate one of the A00, A01, A03, and A05 pins and the resistor connected to that pin.
The operations in DC voltage measurement mode are shown below. Tables 14.5.1.2, 14.5.1.3, and 14.5.1.4 list the input conditions, external resistance values for reference, and internal settings, respectively.

Operation in 600 mV range

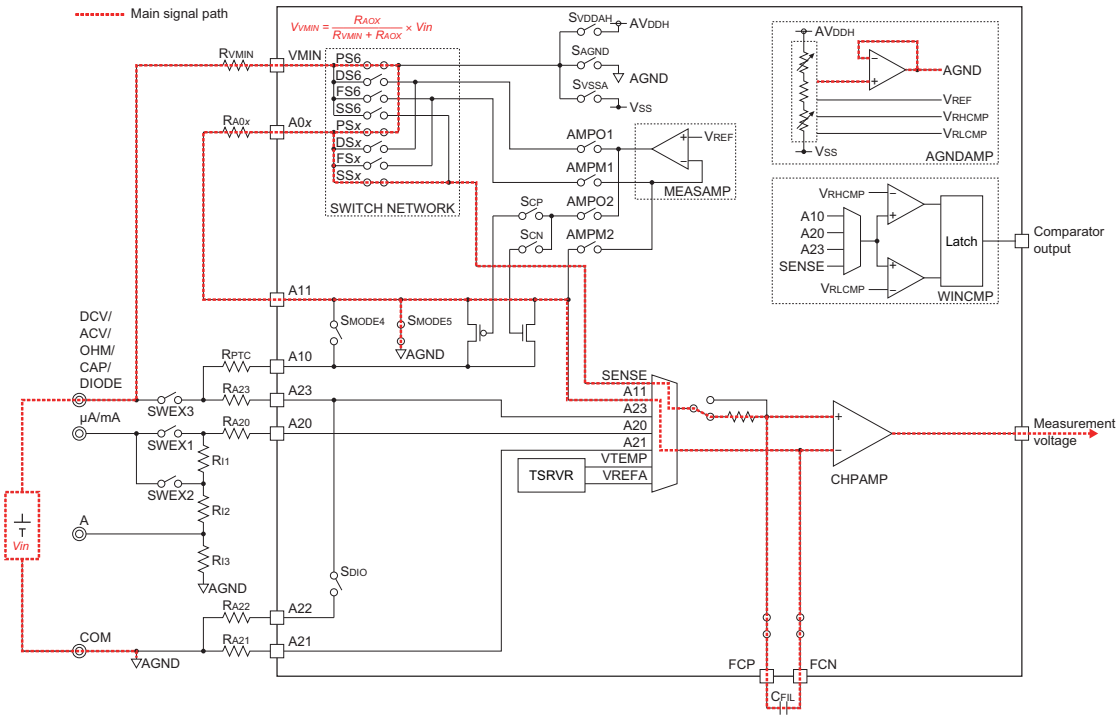
1. The voltage to be measured is input to the A10 pin.
2. The input voltage is sent to the CHPAMP through a filter and amplified by the predefined gain.
3. The voltage amplified in the CHPAMP is output as the measurement voltage.

Operation in 6 V/60 V/600 V/1,000 V range

1. The voltage to be measured is divided with the resistors R_{VMIN} and R_{A0x}, and input to the VMIN pin.
2. The input voltage is sent to the CHPAMP through a filter and amplified by the predefined gain.
3. The voltage amplified in the CHPAMP is output as the measurement voltage.



(1) When measuring in 600 mV range



(2) When measuring in a range other than 600 mV

∗ The expression in the figure assumes that the COM pin voltage is the GND level.

Figure 14.5.1.1 Main Signal Paths in DC Voltage Measurement Mode

Table 14.5.1.1 Measurement Pins and External Switch Settings

Range	Measurement pin	SWEX1	SWEX2	SWEX3
600 mV	DCV COM	OFF	OFF	ON
6 V		OFF	OFF	OFF
60 V				
600 V				
1,000 V				

Table 14.5.1.2 DC Voltage Measurement Input Conditions

Range	A0x pin	Resistor R_{A0x}	Input pin	Input voltage
600 mV	–	–	A10	Vin is directly input.
6 V	A05	R_{A05}	VMIN	Vin × R_{A05} / (R_{VMIN} + R_{A05})
60 V	A03	R_{A03}	VMIN	Vin × R_{A03} / (R_{VMIN} + R_{A03})
600 V	A01	R_{A01}	VMIN	Vin × R_{A01} / (R_{VMIN} + R_{A01})
1,000 V	A00	R_{A00}	VMIN	Vin × R_{A00} / (R_{VMIN} + R_{A00})

∗ The COM pin voltage is assumed as the GND level for the input voltage and Vin.

Table 14.5.1.3 External Resistance Values (for reference)

R_{VMIN}	R_{A00}	R_{A01}	R_{A03}	R_{A05}
10 MΩ	1 kΩ	10 kΩ	101 kΩ	1.11 MΩ

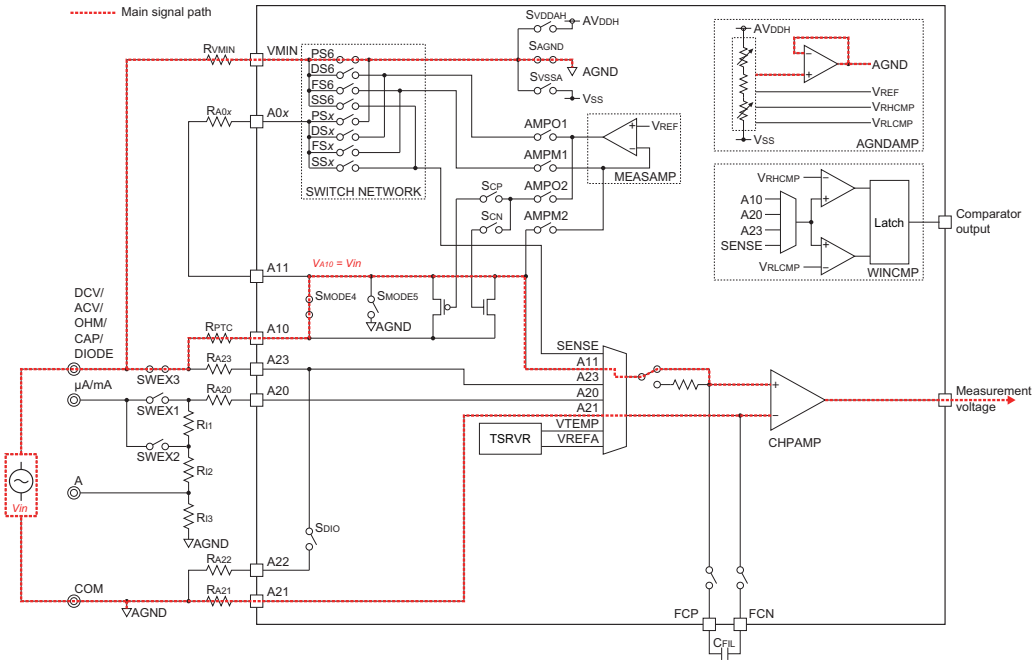
Table 14.5.1.4 Internal Settings for DC Voltage Measurement

Range	CHPAMP gain	AGND (V)	V_{REFP} (V)	V_{REFN} (V)
600 mV	0.6x	1.8	2.7	0.9
6 V				
60 V				
600 V				
1,000 V	6.5x			

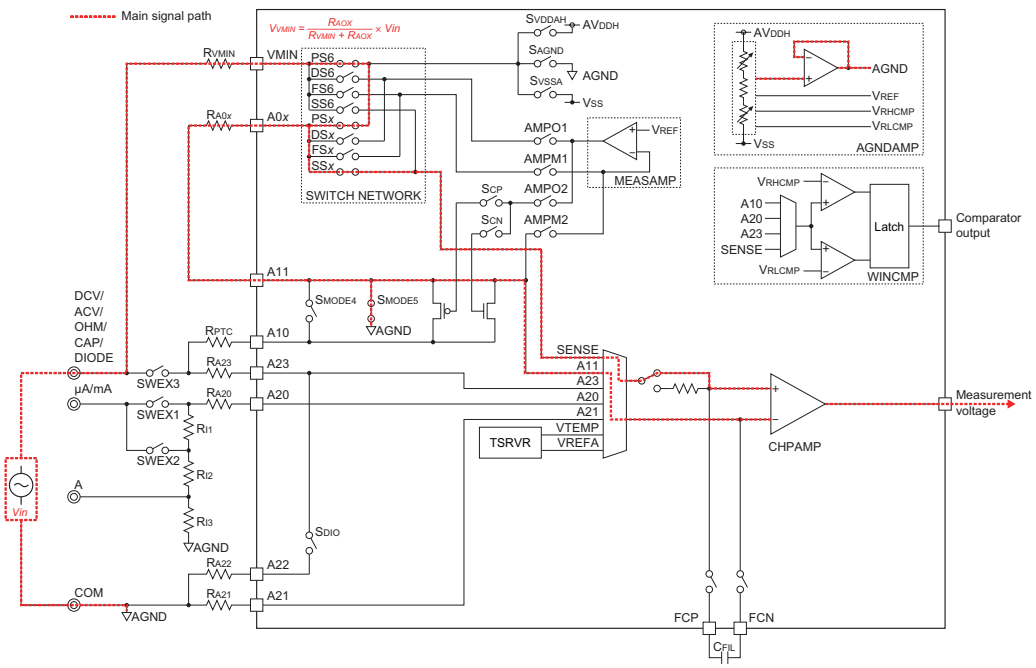
∗ The V_{SS} pin voltage level is assumed as the GND level.

14.5.2 AC Voltage Measurement

In AC voltage measurement mode, the AC voltage that is applied between the ACV and COM pins can be measured. The measurement range can be selected from the following five ranges: 600 mV, 6 V, 60 V, 600 V, and 1,000 V. Figure 14.5.2.1 shows the main signal paths in the analog network unit. The A0x pin and resistor RA0x in Figure 14.5.2.1 indicate one of the A00, A01, A03, and A05 pins and the resistor connected to that pin. The operation in AC voltage measurement mode is the same as DC voltage measurement mode except that the input voltage is sent to the CHAPAMP without passing through the filter (refer to the "DC Voltage Measurement" section).



(1) When measuring in 600 mV range



(2) When measuring in a range other than 600 mV

∗ The expression in the figure assumes that the COM pin voltage is the GND level.

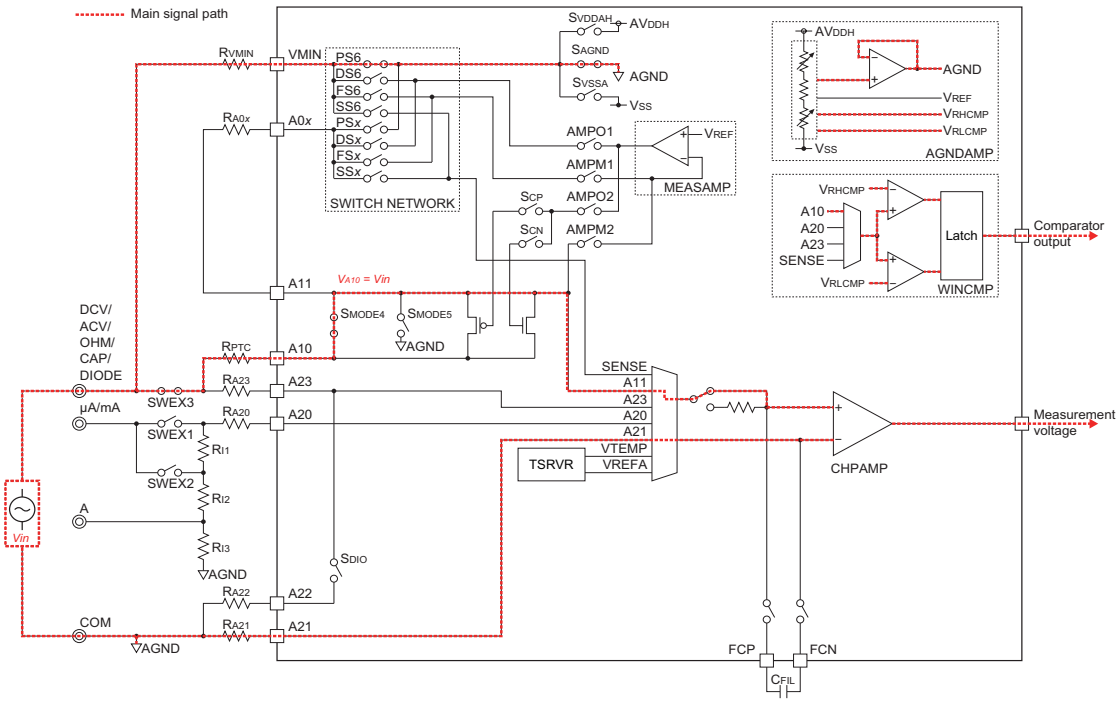
Figure 14.5.2.1 Main Signal Paths in AC Voltage Measurement Mode

14.5.3 AC Voltage and Frequency Measurement

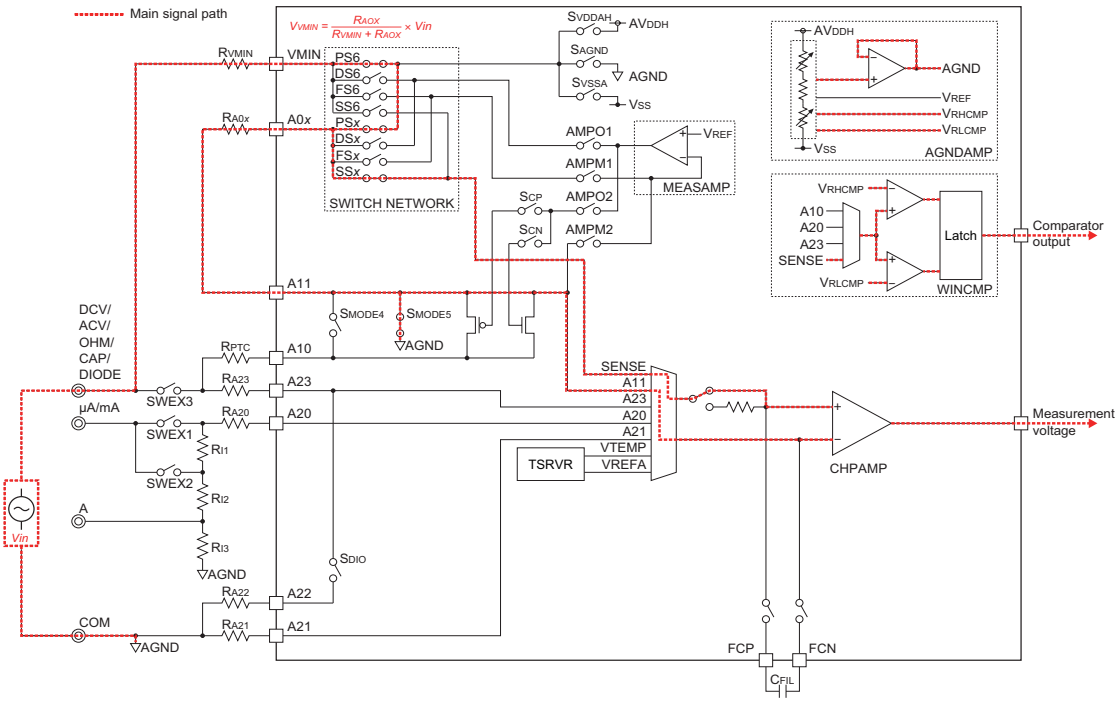
In AC voltage and frequency measurement mode, the AC voltage that is applied between the ACV and COM pins and its frequency can be measured. The measurement range can be selected from the following five ranges: 600 mV, 6 V, 60 V, 600 V, and 1,000 V. The AC voltage measurement function in this measurement mode is the same as that described in the "AC Voltage Measurement" section. Figure 14.5.3.1 shows the main signal paths in the analog network unit. The A0*x* pin and resistor R_{A0x} in Figure 14.5.3.1 indicate one of the A00, A01, A03, and A05 pins and the resistor connected to that pin.
The operation of the frequency measurement is shown below. Figure 14.5.3.2 shows the input and output waveforms of the WINCMP. Table 14.5.3.1 lists the internal settings.

Frequency measurement operation

1. The voltage level of the signal to be measured is divided with the resistors R_{VMIN} and R_{A0x}, and input to the VMIN pin (see Table 14.5.1.2 for the R_{0x} resistance values).
2. The WINCMP converts the input signal into a square wave by inverting the output logic level between H and L at the threshold levels V_{RHCMP} and V_{RLCMP} of the input waveform (see Figure 14.5.3.2).
3. The T16B_DMM counts the pulses of the generated square wave to convert them into the frequency (refer to the "Frequency/Capacitance Measurement Function" section).



(1) When measuring in 600 mV range



(2) When measuring in a range other than 600 mV

∗ The expression in the figure assumes that the COM pin voltage is the GND level.

Figure 14.5.3.1 Main Signal Paths in AC Voltage and Frequency Measurement Mode

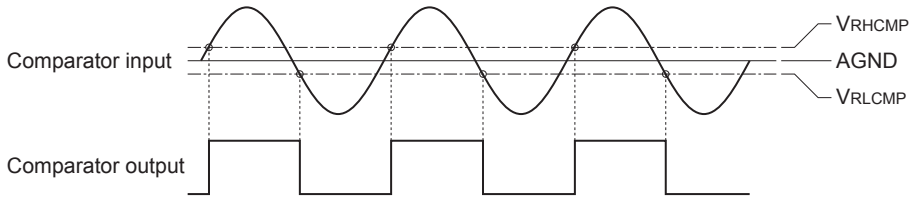


Figure 14.5.3.2 WINCMP Operation in AC Voltage and Frequency Measurement Mode

Table 14.5.3.1 Internal Settings for AC Voltage and Frequency Measurement

Range	CHPAMP gain	AGND (V)	VREFP (V)	VREFN (V)	VRHCMP (V)	VRLCMP (V)
600 mV	0.6x	1.8	2.7	0.9	1.80625	1.79375
6 V						
60 V						
600 V						
1,000 V	6.5x					

∗ The VSS pin voltage level is assumed as the GND level.

14.5.4 Resistance Measurement

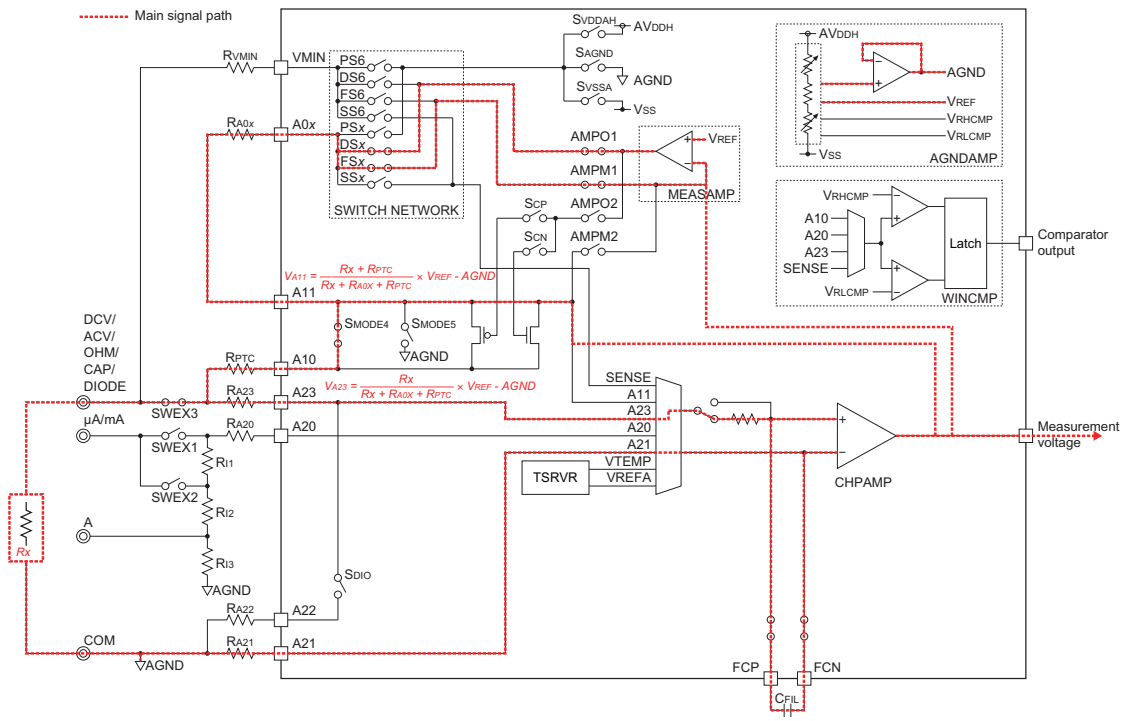
In resistance measurement mode, the resistance value that is connected between the OHM and COM pins can be measured. This mode supports two measurement methods: CV (Constant Voltage) method and CC (Constant Current) method. The measurement range can be selected from three ranges, 600 Ω, 6 kΩ, and 60 kΩ, in the CV method or six ranges, 600 Ω, 6 kΩ, 60 kΩ, 600 kΩ, 6 MΩ, and 60 MΩ, in the CC method. Table 14.5.4.1 shows the external switch settings. Figure 14.5.4.1 shows the main signal paths in the analog network unit. The A0x pin and resistor R_{A0x} in Figure 14.5.4.1 indicate one of the A00, A01, A03, A04, and A05 pins and the resistor connected to that pin.
The operations in resistance measurement mode are shown below. Tables 14.5.4.2, 14.5.4.3, and 14.5.4.4 list the input conditions, external resistance values for reference, and internal settings, respectively.

Operation in 600 Ω/6 kΩ/60 kΩ range (CV method)

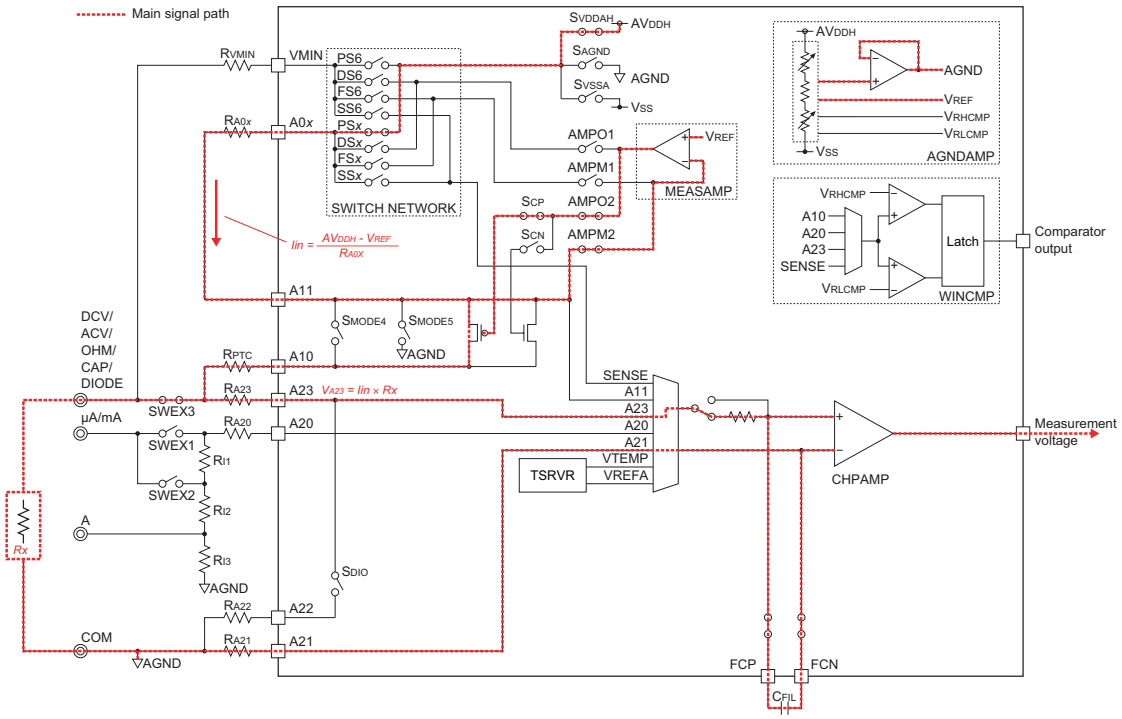
1. The reference voltage generated by the MEASAMP is applied to the resistor ladder composed of R_{A0x}, R_{PTC}, and Rx (resistance to be measured). The divided voltage that is applied to Rx is input to the A23 pin.
2. The input voltage is sent to the CHPAMP through a filter and amplified by the predefined gain.
3. The voltage amplified in the CHPAMP is output as the measurement voltage.

Operation in 600 Ω/6 kΩ/60 kΩ/600 kΩ/6 MΩ/60 MΩ range (CC method)

1. The potential difference between the AV_{DDH} voltage and the reference voltage generated by the MEASAMP is applied to the resistor R_{A0x} to generate constant current.
2. The generated current is fed to Rx (resistance to be measured) and the generated voltage is input to the A23 pin.
3. The input voltage is sent to the CHPAMP through a filter and amplified by the predefined gain.
4. The voltage amplified in the CHPAMP is output as the measurement voltage.



(1) CV method



(2) CC method

∗ The expression in the figure assumes that the COM pin voltage is the GND level.

Figure 14.5.4.1 Main Signal Paths in Resistance Measurement Mode

Table 14.5.4.1 Measurement Pins and External Switch Settings

Range	Measurement pin	SWEX1	SWEX2	SWEX3
600 Ω	OHM COM	OFF	OFF	ON
6 kΩ				
60 kΩ				
600 kΩ				
6 MΩ				
60 MΩ				

Table 14.5.4.2 Resistance Measurement Input Conditions

Measurement method	Range	A0x pin	Resistor R_{A0x}	Input pin	Input voltage
CV	600 Ω	A00	R_{A00}	A23	$Rx / (Rx + R_{A00} + R_{PTC}) \times V_{REF}$
	6 kΩ	A01	R_{A01}		$Rx / (Rx + R_{A01} + R_{PTC}) \times V_{REF}$
	60 kΩ	A03	R_{A03}		$Rx / (Rx + R_{A03} + R_{PTC}) \times V_{REF}$
CC	600 Ω	A00	R_{A00}		$(AV_{DDH} - V_{REF}) / R_{A00} \times Rx$
	6 kΩ	A00	R_{A00}		$(AV_{DDH} - V_{REF}) / R_{A00} \times Rx$
	60 kΩ	A01	R_{A01}		$(AV_{DDH} - V_{REF}) / R_{A01} \times Rx$
	600 kΩ	A03 A04	R_{A03} ∗2 R_{A04}		$(AV_{DDH} - V_{REF}) / \{(R_{A03} \times R_{A04}) / (R_{A03} + R_{A04})\} \times Rx$
	6 MΩ	A04 A05	R_{A04} ∗2 R_{A05}		$(AV_{DDH} - V_{REF}) / \{(R_{A04} \times R_{A05}) / (R_{A04} + R_{A05})\} \times Rx$
	60 MΩ	A04	R_{A04}		$(AV_{DDH} - V_{REF}) / R_{A04} \times Rx$

∗1 The COM pin voltage is assumed as the GND level for the input voltage. The V_{SS} pin voltage level is assumed as the GND level for V_{REF}, AGND, and AV_{DDH}.

∗2 In 600 kΩ range, R_{A0x} shows the resistors R_{A03} and R_{A04} connected in parallel. In 6 MΩ range, R_{A0x} shows the resistors R_{A04} and R_{A05} connected in parallel.

Table 14.5.4.3 External Resistance Values (for reference)

R_{PTC}	R_{A00}	R_{A01}	R_{A03}	R_{A04}	R_{A05}
200 Ω	1 kΩ	10 kΩ	101 kΩ	10 MΩ	1.11 MΩ

Table 14.5.4.4 Internal Settings for Resistance Measurement

Measurement method	Range	CHPAMP gain	AVDDH (V)	AGND (V)	VREF (V)	VREFP (V)	VREFN (V)
CV	600 Ω	0.6x	3.6	1.2	2.4	2.4	(Rx + RPTC) / (Rx + RA0x + RPTC) × VREF
	6 kΩ						
	60 kΩ						
CC	600 Ω	1x		0.6	2.6	2.7	0.9
	6 kΩ	0.6x			3.4		
	60 kΩ						
	600 kΩ						
	6 MΩ						
	60 MΩ						

∗ The VSS pin voltage level is assumed as the GND level.

14.5.5 Continuity Check

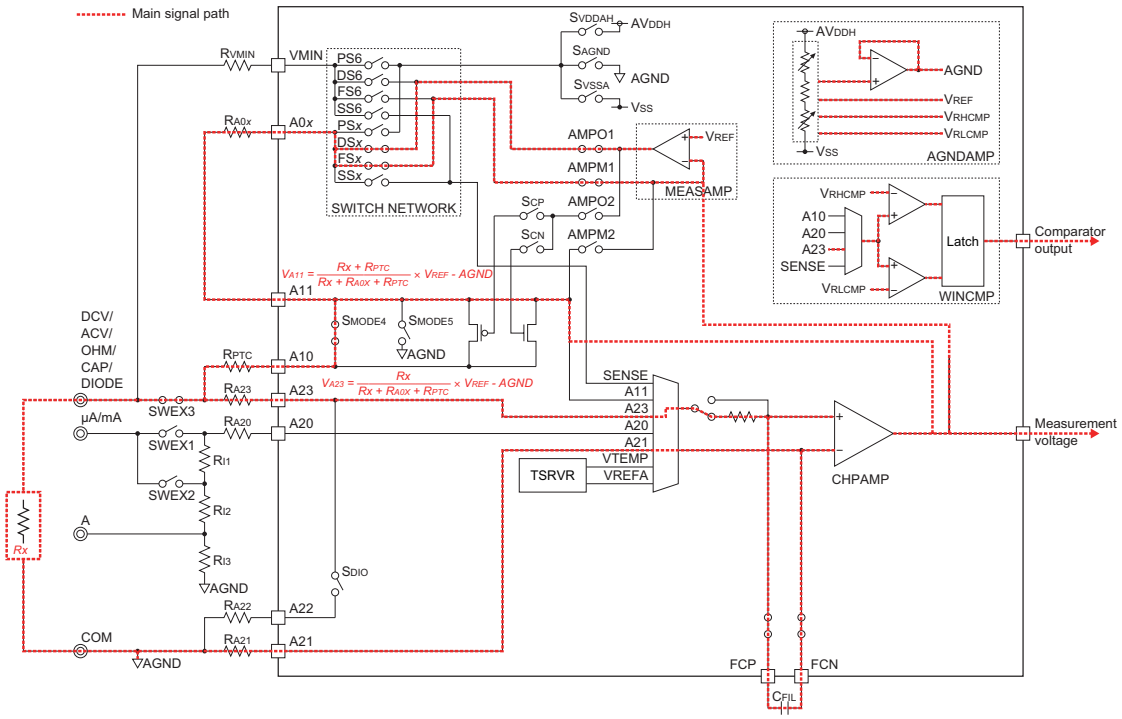
In continuity check mode, the resistance value that is connected between the OHM and COM pins is measured and the measured resistance value is confirmed if it is lower than the predefined value. The resistance measurement operation is the same as that described in the "Resistance Measurement" section. The 600 Ω range is used in this measurement mode. Figure 14.5.5.1 shows the main signal paths in the analog network unit.
The operation in continuity check mode is shown below. Figure 14.5.5.2 shows the input and output levels of the WINCMP. Table 14.5.5.1 shows the continuity detection and cancellation resistance values for reference.

Continuity check operation (to confirm if the measured resistance is lower than the predefined value) (CV method)

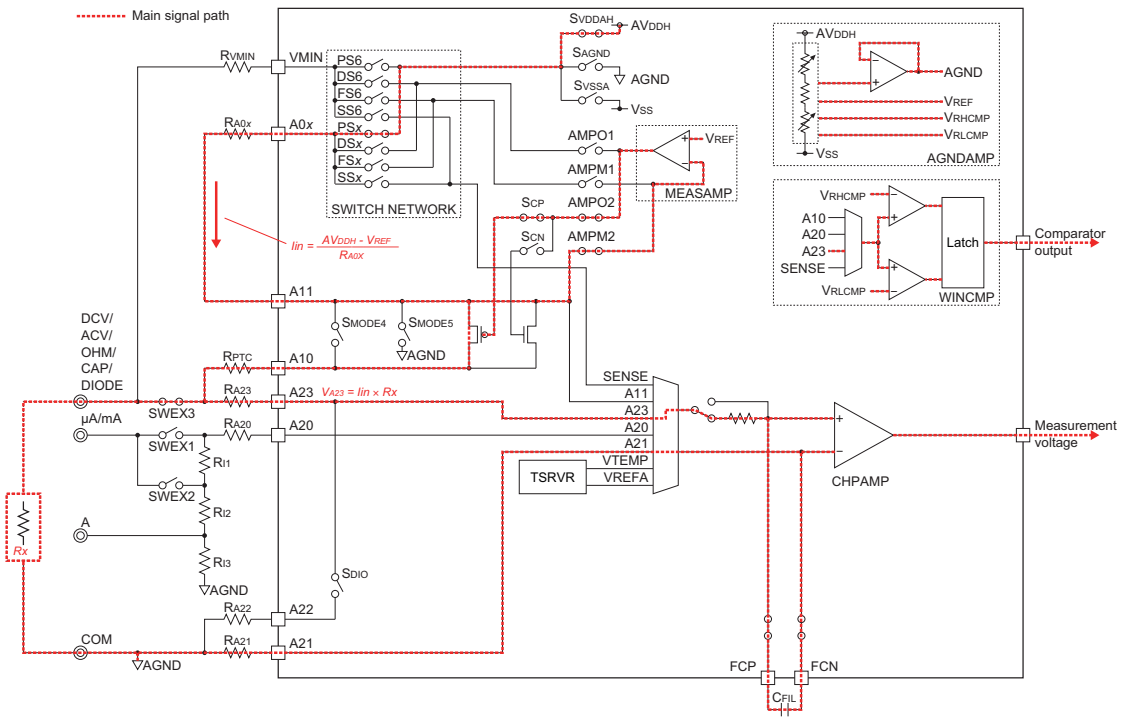
1. The reference voltage generated by the MEASAMP is applied to the resistor ladder composed of RA00, RPTC, and Rx (resistance to be measured). The divided voltage that is applied to Rx is input to the A23 pin.
2. The WINCMP compares the input voltage with the threshold levels (VRHCMP and VRLCMP) and the comparison result is output as an H or L level (see Figure 14.5.5.2).
3. For the control to sound the buzzer while a continuity status is detected, refer to the "Sound Generator" chapter.

Continuity check operation (to confirm if the measured resistance is lower than the predefined value) (CC method)

1. The potential difference between the AVDDH voltage and the reference voltage generated by the MEASAMP is applied to the resistor RA0x to generate constant current.
2. The generated current is fed to Rx (resistance to be measured) and the generated voltage is input to the A23 pin.
3. The WINCMP compares the input voltage with the threshold levels (VRHCMP and VRLCMP) and the comparison result is output as an H or L level (see Figure 14.5.5.2).
4. For the control to sound the buzzer while a continuity status is detected, refer to the "Sound Generator" chapter.



(1) CV method



(2) CC method

∗ The expression in the figure assumes that the COM pin voltage is the GND level.

Figure 14.5.5.1 Main Signal Paths in Continuity Check Mode

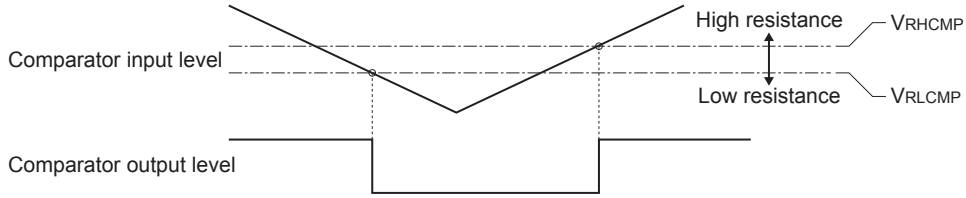


Figure 14.5.5.2 WINCMP Operation in Continuity Check Mode

Table 14.5.5.1 Continuity Detection and Cancellation Resistance Values (for reference)

Detection resistance	Cancellation resistance	Detection voltage	Cancellation voltage
12.6 Ω	250 Ω	1.2125 V	14 V

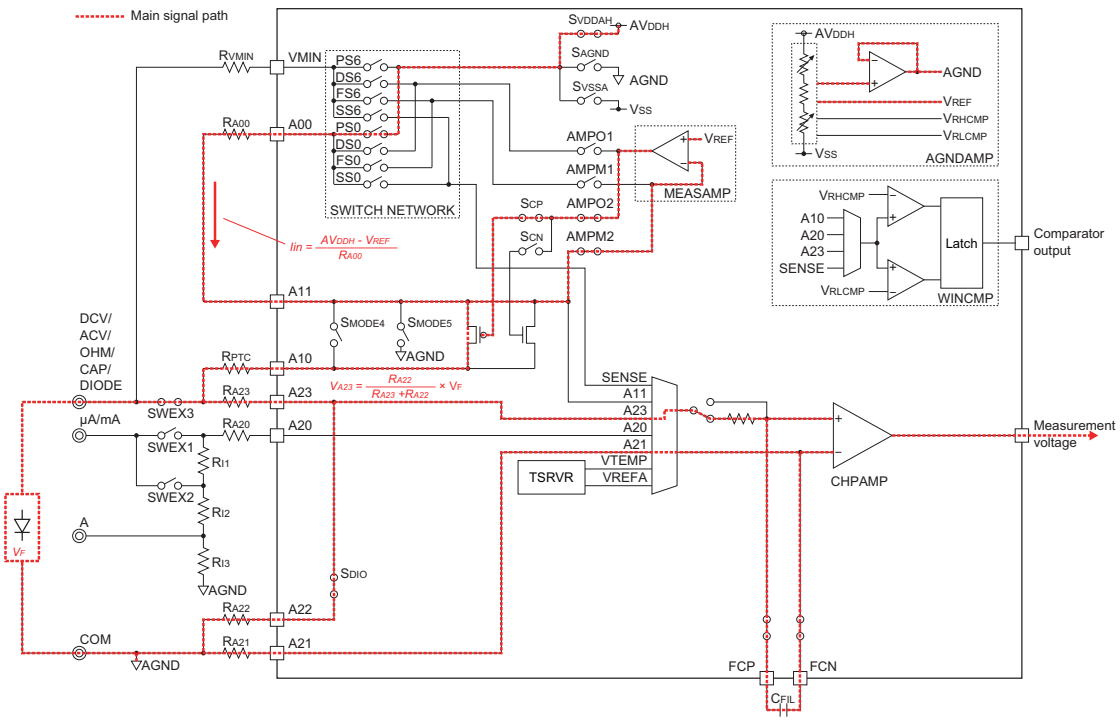
* The VSS pin voltage level is assumed as the GND level for the detection and cancellation voltages.

14.5.6 Diode VF Measurement

In diode VF measurement mode, the VF voltage of the diode that is connected between the DIODE (anode) and COM (cathode) pins can be measured. Table 14.5.6.1 shows the external switch settings. Figure 14.5.6.1 shows the main signal paths in the analog network unit.
The operation in diode VF measurement mode is shown below. Tables 14.5.6.2, 14.5.6.3, and 14.5.6.4 list the input conditions, external resistance values for reference, and internal settings, respectively.

Diode VF measurement operation

1. The potential difference between the AVDDH voltage and the reference voltage generated by the MEASAMP is applied to the resistor RA00 to generate constant current.
2. The generated current is fed to the diode to be measured and the VF voltage is generated.
3. The generated VF voltage is divided with the resistors RA23 and RA22 and input to the A23 pin.
4. The input voltage is sent to the CHPAMP through a filter and amplified by the predefined gain.
5. The voltage amplified in the CHPAMP is output as the measurement voltage.



* The expression in the figure assumes that the COM pin voltage is the GND level.

Figure 14.5.6.1 Main Signal Paths in Diode VF Measurement Mode

Table 14.5.6.1 Measurement Pins and External Switch Settings

Measurement pin	SWEX1	SWEX2	SWEX3
DIODE COM	OFF	OFF	ON

Table 14.5.6.2 Diode V_F Measurement Input Conditions

Input pin	Input voltage
A23	$R_{A22} / (R_{A22} + R_{A23}) \times V_F$

∗ The COM pin input voltage level is assumed as the GND level for the input voltage and V_F.

Table 14.5.6.3 External Resistance Values (for reference)

R_{A00}	R_{PTC}	R_{A22}	R_{A23}
1 kΩ	200 Ω	200 kΩ	300 kΩ

Table 14.5.6.4 Internal Settings for Diode V_F Measurement

CHPAMP gain	AV_{DDH} (V)	AGND (V)	V_{REF} (V)	V_{REFP} (V)	V_{REFN} (V)
0.6x	3.6	0.2	3.2	2.7	0.9

∗ The V_{SS} pin voltage level is assumed as the GND level.

14.5.7 Internal Temperature Measurement

In internal temperature measurement mode, the temperature in the analog network unit can be measured. Table 14.5.7.1 shows the external switch settings. Figure 14.5.7.1 shows the main signal paths in the analog network unit. The operation in internal temperature measurement mode is shown below. Table 14.5.7.2 lists the internal settings.

Internal temperature measurement operation

1. The temperature dependent voltage that is output from the temperature sensor embedded in the TSRVR of the analog network unit is passed through a filter only and output as the measurement voltage.
2. Calculate the temperature value from the A/D conversion value (refer to the "Internal Temperature Measurement Function" section).

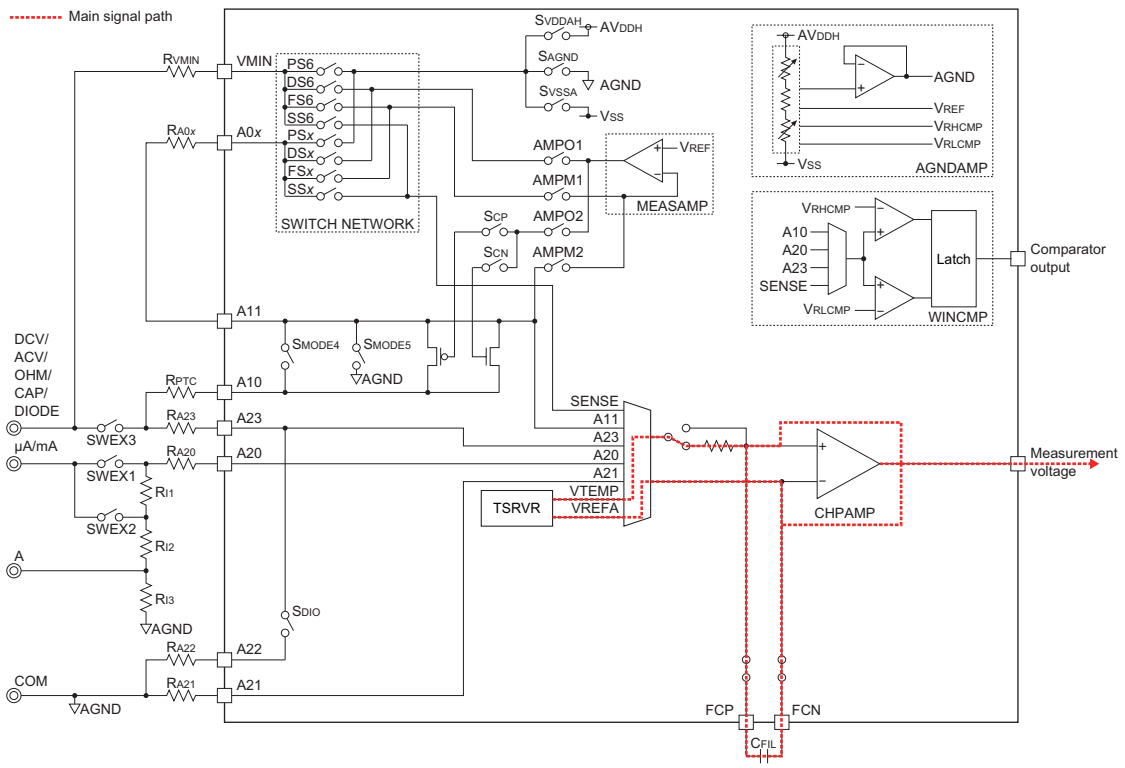


Figure 14.5.7.1 Main Signal Paths in Internal Temperature Measurement Mode

Table 14.5.7.1 External Switch Settings

Measurement pin	SWEX1	SWEX2	SWEX3
–	OFF	OFF	OFF

Table 14.5.7.2 Internal Settings for Internal Temperature Measurement

V_{REFP} (V)	V_{REFN} (V)
2.7	0.9

∗ The V_{SS} pin voltage level is assumed as the GND level.

14.5.8 DC Current Measurement

In DC current measurement mode, the DC current that flows from the COM pin to the μA/mA or A (ampere) pin can be measured. The measurement range can be selected from the following six ranges: 600 μA, 6 mA, 60 mA, 600 mA, 6 A, and 10 A. Table 14.5.8.1 shows the measurement pins and external switch settings. Figure 14.5.8.1 shows the main signal paths in the analog network unit.
The operations in DC current measurement mode are shown below. Tables 14.5.8.2, 14.5.8.3, and 14.5.8.4 list the input conditions, external resistance values for reference, and internal settings, respectively.

Operation in 600 μA/6 mA range

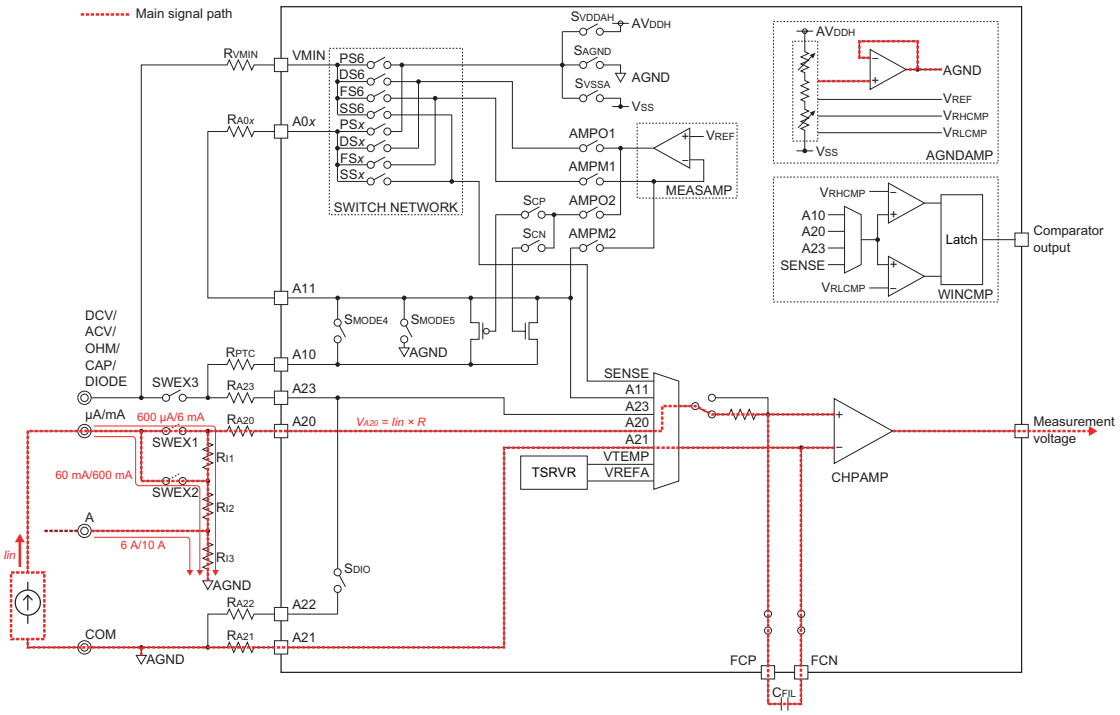
1. The current (Iin) to be measured is fed to the serially connected resistors R_{I1}, R_{I2}, and R_{I3}, and the generated voltage is input to the A20 pin.
2. The input voltage is sent to the CHPAMP through a filter and amplified by the predefined gain.
3. The voltage amplified in the CHPAMP is output as the measurement voltage.

Operation in 60 mA/600 mA range

1. The current (Iin) to be measured is fed to the serially connected resistors R_{I2}, and R_{I3}, and the generated voltage is input to the A20 pin.
2. The input voltage is sent to the CHPAMP through a filter and amplified by the predefined gain.
3. The voltage amplified in the CHPAMP is output as the measurement voltage.

Operation in 6 A/60 A range

1. The current (Iin) to be measured is fed to the resistor R_{I3}, and the generated voltage is input to the A20 pin.
2. The input voltage is sent to the CHPAMP through a filter and amplified by the predefined gain.
3. The voltage amplified in the CHPAMP is output as the measurement voltage.



∗ The expression in the figure assumes that the COM pin voltage is the GND level.

Figure 14.5.8.1 Main Signal Paths in DC Current Measurement Mode

Table 14.5.8.1 Measurement Pins and External Switch Settings

Range	Measurement pin	SWEX1	SWEX2	SWEX3
600 μA	μA/mA COM	ON	OFF	OFF
6 mA				
60 mA		OFF	ON	OFF
600 mA				
6 A	A COM	OFF	OFF	OFF
10 A				

Table 14.5.8.2 DC Current Measurement Input Conditions

Range	Input pin	Input voltage
600 μA	A20	$(R_{I1} + R_{I2} + R_{I3}) \times Iin$
6 mA		
60 mA		$(R_{I2} + R_{I3}) \times Iin$
600 mA		
6 A		$R_{I3} \times Iin$
10 A		

∗ The COM pin input voltage level is assumed as the GND level for the input voltage.

Table 14.5.8.3 External Resistance Values (for reference)

R_{I1}	R_{I2}	R_{I3}
100 Ω	1 Ω	0.01 Ω

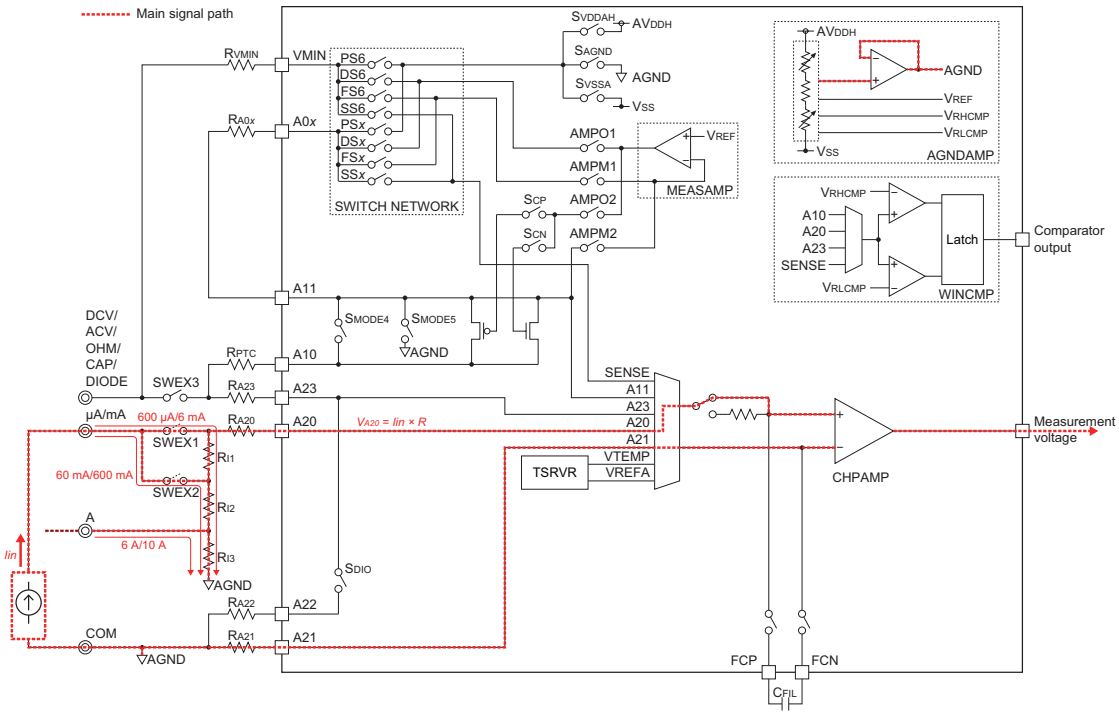
Table 14.5.8.4 Internal Settings for DC Current Measurement

Range	CHPAMP gain	AGND(V)	V_{REFP} (V)	V_{REFN} (V)
600 μA	6.5x	1.8	2.7	0.9
6 mA	0.6x			
60 mA	6.5x			
600 mA	0.6x			
6 A	6.5x			
10 A	0.6x			

∗ The V_{SS} pin voltage level is assumed as the GND level.

14.5.9 AC Current Measurement

In AC current measurement mode, the AC current that flows between the COM pin and the µA/mA or A (ampere) pin can be measured. Figure 14.5.9.1 shows the main signal paths in the analog network unit. The measurement ranges and operation in AC current measurement mode is the same as DC current measurement mode except that the input voltage is sent to the CHAPAMP without passing through the filter (refer to the "DC Current Measurement" section).



∗ The expression in the figure assumes that the COM pin voltage is the GND level.

Figure 14.5.9.1 Main Signal Paths in AC Current Measurement Mode

14.5.10 AC Current and Frequency Measurement

In AC current and frequency measurement mode, the AC current that flows between the COM pin and the µA/mA or A (ampere) pin and the frequency can be measured. The measurement range can be selected from the following six ranges: 600 µA, 6 mA, 60 mA, 600 mA, 6 A, and 10 A. The AC current measurement function in this measurement mode is the same as that described in the "AC Current Measurement" section. Figure 14.5.10.1 shows the main signal paths in the analog network unit.

The operation of the frequency measurement is shown below. Figure 14.5.10.2 shows the input and output waveforms of the WINCMP. Table 14.5.10.1 lists the internal settings.

Operation in 600 µA/6 mA range (frequency measurement operation)

1. The current (Iin) to be measured is fed to the serially connected resistors R_{I1}, R_{I2}, and R_{I3}, and the generated voltage is input to the A20 pin.
2. The WINCMP converts the input signal into a square wave by inverting the output logic level between H and L at the threshold levels V_{RHCMP} and V_{RLCMP} of the input waveform (see Figure 14.5.10.2).
3. The T16B_DMM counts the pulses of the generated square wave to convert them into frequency (refer to the "Frequency/Capacitance Measurement Function" section).

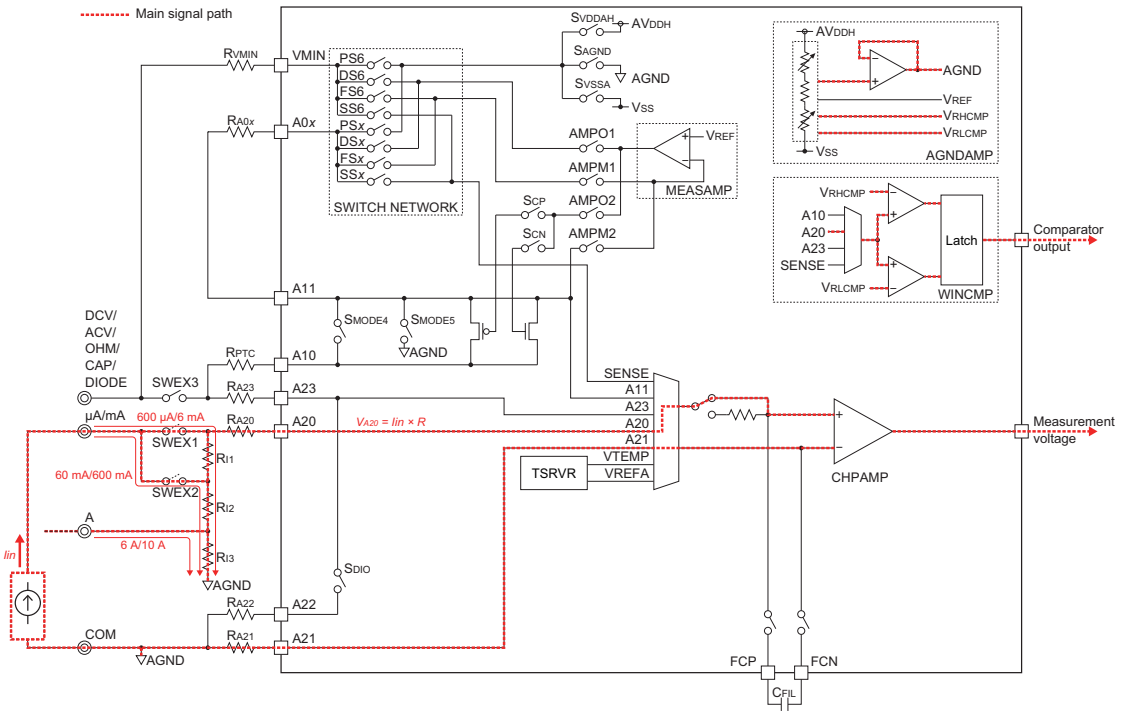
Operation in 60 mA/600 mA range (frequency measurement operation)

1. The current (Iin) to be measured is fed to the serially connected resistors R_{I2}, and R_{I3}, and the generated voltage is input to the A20 pin.

2. The WINCMP converts the input signal into a square wave by inverting the output logic level between H and L at the threshold levels VRHCMP and V3 of the input waveform (see Figure 14.5.10.2).
3. The T16B_DMM counts the pulses of the generated square wave to convert them into frequency (refer to the "Frequency/Capacitance Measurement Function" section).

Operation in 6 A/10 A range (frequency measurement operation)

1. The current (Iin) to be measured is fed to the resistor RI3, and the generated voltage is input to the A20 pin.
2. The WINCMP converts the input signal into a square wave by inverting the output logic level between H and L at the threshold levels VRHCMP and VRLCMP of the input waveform (see Figure 14.5.10.2).
3. The T16B_DMM counts the pulses of the generated square wave to convert them into frequency (refer to the "Frequency/Capacitance Measurement Function" section).



∗ The expression in the figure assumes that the COM pin voltage is the GND level.

Figure 14.5.10.1 Main Signal Paths in AC Current and Frequency Measurement Mode

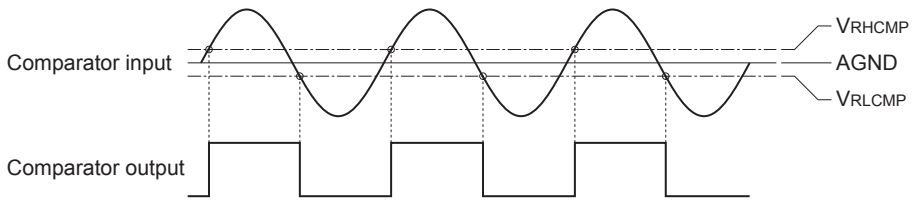


Figure 14.5.10.2 WINCMP Operation in AC Current and Frequency Measurement Mode

Table 14.5.10.1 Internal Settings for AC Current and Frequency Measurement

Range	CHPAMP gain	AGND (V)	VREFP (V)	VREFN (V)	VRHCMP (V)	VRLCMP (V)
600 μA	6.5x	1.8	2.7	0.9	1.80625	1.79375
6 mA	0.6x					
60 mA	6.5x					
600 mA	0.6x					
6 A	6.5x					
10 A	0.6x					

∗ The VSS pin voltage level is assumed as the GND level.

14.5.11 Capacitance Measurement

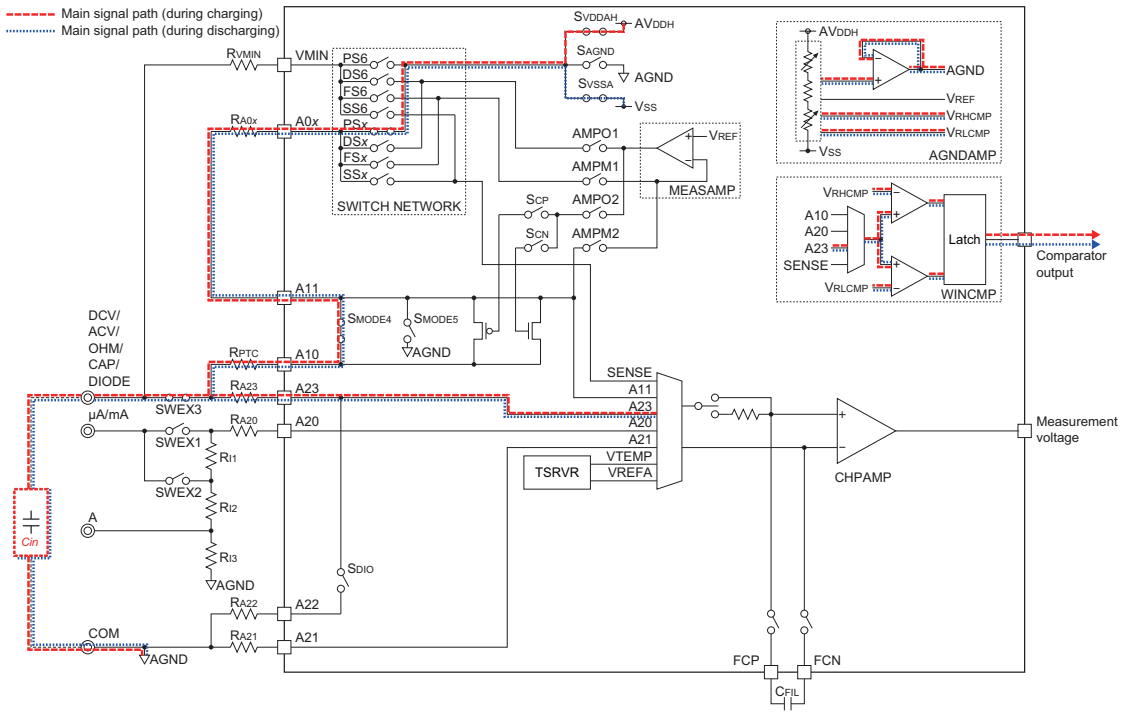
In capacitance measurement mode, the capacitance value that is connected between the CAP and COM pins can be measured. The measurement range can be selected from the following six ranges: 10 nF, 100 nF, 1 µF, 10 µF, 100 µF, and 1,000 µF. This mode supports two measurement methods: CV (Constant Voltage) method and CC (Constant Current) method. The CV method is used in the 10 nF and 100 nF measurement ranges; the CC method is used in the 1 µF, 10 µF, 100 µF, and 1,000 µF measurement ranges. Table 14.5.11.1 shows the external switch settings. Figure 14.5.11.1 shows the main signal paths in the analog network unit. The A0*x* pin and resistor R_{A0x} in Figure 14.5.11.1 indicate one of the A00, A01, A03, A04, and A05 pins and the resistor connected to that pin.
The operations in capacitance measurement mode are shown below. Tables 14.5.11.2, 14.5.11.3, and 14.5.11.4 list the input conditions, external resistance values for reference, and internal settings, respectively. Figure 14.5.11.2 shows the input and output waveforms of the WINCMP.

Operation in 10 nF/100 nF range (CV method)

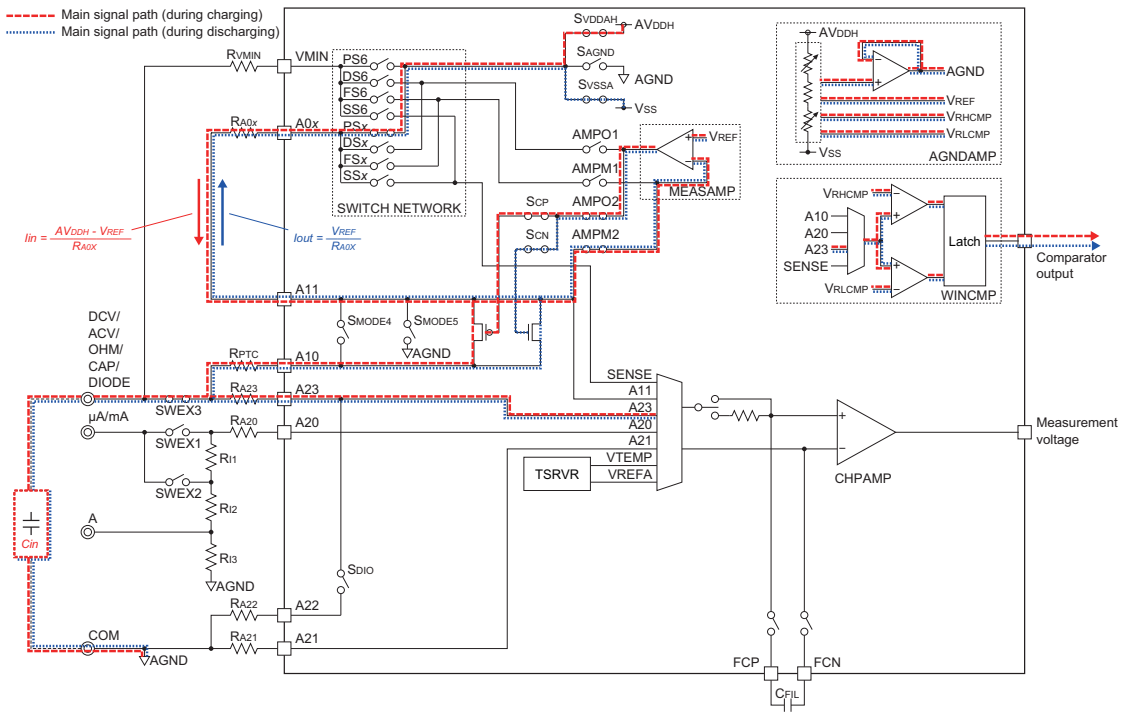
1. The AV_{DDH} voltage is applied to the serially connected R_{A0x}, R_{PTC}, and Cin (capacitor to be measured).
2. The capacitor is charged and the capacitor voltage is input to the A23 pin.
3. The WINCMP compares the input voltage with the V_{RHCMP} voltage and switches the output level from L to H when the input voltage exceeds V_{RHCMP}. At the same time, the discharge operation starts (Steps 4 to 6). (See Figure 14.5.11.2.)
4. The V_{SS} voltage is applied to the serially connected R_{A0x}, R_{PTC}, and Cin (capacitor to be measured).
5. The capacitor is discharged and the capacitor voltage is input to the A23 pin.
6. The WINCMP compares the input voltage with the V_{RLCMP} voltage and switches the output level from H to L when the input voltage drops below V_{RLCMP}. At the same time, the charge operation starts (Steps 1 to 3). (See Figure 14.5.11.2.)
7. The charge operation (Steps 1 to 3) and discharge operation (Steps 4 to 6) are repeated to output a square wave from the WINCMP.
8. The T16B_DMM counts the pulses of the generated square wave to convert them into frequency (refer to the "Frequency/Capacitance Measurement Function" section).
9. Calculate the capacitance value from the frequency obtained (refer to the "Frequency/Capacitance Measurement Function" section).

Operation in 1 µF/10 µF/100 µF/1,000 µF range (CC method)

1. The potential difference between the AV_{DDH} voltage and the reference voltage generated by the MEASAMP is applied to the resistor R_{A0x} to generate constant current.
2. The Cin (capacitor to be measured) is charged with this constant current and the capacitor voltage is input to the A23 pin.
3. The WINCMP compares the input voltage with the V_{RHCMP} voltage and switches the output level from L to H when the input voltage exceeds V_{RHCMP}. At the same time, the discharge operation starts (Steps 4 to 6). (See Figure 14.5.11.2.)
4. The potential difference between the V_{SS} voltage and the reference voltage generated by the MEASAMP is applied to the resistor R_{A0x} to generate constant current.
5. The Cin (capacitor to be measured) is discharged with this constant current and the capacitor voltage is input to the A23 pin.
6. The WINCMP compares the input voltage with the V_{RLCMP} voltage and switches the output level from H to L when the input voltage drops below V_{RLCMP}. At the same time, the charge operation starts (Steps 1 to 3). (See Figure 14.5.11.2.)
7. The charge operation (Steps 1 to 3) and discharge operation (Steps 4 to 6) are repeated to output a square wave from the WINCMP.
8. The T16B_DMM counts the pulses of the generated square wave to convert them into frequency (refer to the "Frequency/Capacitance Measurement Function" section).
9. Calculate the capacitance value from the frequency obtained (refer to the "Frequency/Capacitance Measurement Function" section).



(1) CV method



(2) CC method

Figure 14.5.11.1 Main Signal Paths in Capacitance Measurement Mode

Table 14.5.11.1 Measurement Pins and External Switch Settings

Range	Measurement pin	SWEX1	SWEX2	SWEX3
10 nF	CAP COM	OFF	OFF	ON
100 nF				
1 µF				
10 µF				
100 µF				
1,000 µF				

Table 14.5.11.2 Capacitance Measurement Input Conditions

Range	A0*x* pin	Resistor R_{A0x}	Input pin	Input voltage
10 nF	A05	R_{A05}	A23	–
100 nF	A03	R_{A03}		
1 µF	A03, A04	R_{A03}, R_{A04} *		
10 µF	A01	R_{A01}		
100 µF	A00	R_{A00}		
1,000 µF				

* In 1 µF range, R_{A0x} shows the resistors R_{A03} and R_{A04} connected in parallel.

Table 14.5.11.3 External Resistance Values (for reference)

R_{A00}	R_{A01}	R_{A03}	R_{A04}	R_{A05}	R_{PTC}
1 kΩ	10 kΩ	101 kΩ	10 MΩ	1.11 MΩ	200 Ω

Table 14.5.11.4 Internal Settings for Capacitance Measurement

Range	AV_{DDH} (V)	V_{SS} (V)	AGND (V)	V_{REF} (V)	V_{RHCMP} (V)	V_{RLCMP} (V)
10 nF	3.6	0	1.8	–	2.2	1
100 nF						
1 µF				2.8 (charge) 0.8 (discharge)	1.8	1.4
10 µF						
100 µF						
1,000 µF						1.6

* The V_{SS} pin voltage level is assumed as the GND level.

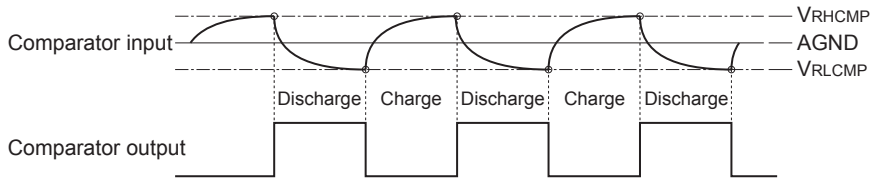


Figure 14.5.11.2 WINCMP Operation in Capacitance Measurement Mode

14.6 Interrupt

The DSADC16 has a function to generate the interrupt shown in Table 14.6.1.

Table 14..6.1 DSADC16 Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Comb filter conversion completion	DSADC16IF.COMBIF	When a conversion in the Comb filter has completed	Writing 1
LPF conversion completion	DSADC16IF.LPFIF	When a conversion in the LPF (low-pass filter) has completed	Writing 1
HPF conversion completion	DSADC16IF.HPFIF	When a conversion in the HPF (high-pass filter) has completed	Writing 1
Averaging circuit conversion completion	DSADC16IF.RMSIF	When a conversion in the averaging circuit has completed	Writing 1
Square root circuit conversion completion	DSADC16IF.TRUERMSIF	When a conversion in the square root circuit has completed	Writing 1
Comb filter conversion result overwrite error	DSADC16IF.COMB_OVRIF	When a new conversion result is loaded to the DSADC16COMB register while the DSADC16IF.COMBIF bit = 1	Writing 1
LPF conversion result overwrite error	DSADC16IF.LPF_OVRIF	When a new conversion result is loaded to the DSADC16LPFHPF register while the DSADC16IF.LPFIF bit = 1	Writing 1
HPF conversion result overwrite error	DSADC16IF.HPF_OVRIF	When a new conversion result is loaded to the DSADC16LPFHPF register while the DSADC16IF.HPFIF bit = 1	Writing 1
Averaging circuit conversion result overwrite error	DSADC16IF.RMS_OVRIF	When a new conversion result is loaded to the DSADC16RMS1 and 2 registers while the DSADC16IF.RMSIF bit = 1	Writing 1
Square root circuit conversion result overwrite error	DSADC16IF.RMS_OVRIF	When a new conversion result is loaded to the DSADC16RMS1 and 2 registers while the DSADC16IF.TRUERMSIF bit = 1	Writing 1
Continuity status change detection	DSADC16IF.CONTIF	When a continuity status change (continuity from/to non-continuity) is detected	Writing 1

Note that the conversion continues even if a conversion result overwrite error has occurred. Conversion result overwrite errors are decided regardless of whether the conversion result register has been read or not.

The DSADC16 provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the “Interrupt Controller” chapter.

14.7 Control Registers

DSADC16 Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSADC16CLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/W	
	7–6	–	0x0	–	R	
	5–4	CLKDIV[1:0]	0x0	H0	R/W	
	3–2	–	0x0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/W	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the DSADC16 operating clock is supplied in DEBUG mode or not.

1 (R/W): Clock supplied in DEBUG mode

0 (R/W): No clock supplied in DEBUG mode

Bits 7–6 Reserved

Bits 5–4 CLKDIV[1:0]

These bits select the division ratio of the DSADC16 operating clock.

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of the DSADC16.

Table 14.7.1 Clock Source and Division Ratio Settings

DSADC16CLK. CLKDIV[1:0] bits	DSADC16CLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0x3	1/8	1/16	1/16	1/16
0x2	1/4	1/8	1/8	1/8
0x1	1/2	1/4	1/4	1/4
0x0	1/1	1/1	1/1	1/1

(Note) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

Note: The DSADC16CLK register settings can be altered only when the DSADC16CONF.MODEN bit = 0.

DSADC16 Configuration Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSADC16CONF	15–14	–	0x0	–	R	–
	13–12	AVE_NUM[1:0]	0x0	H0	R/W	
	11–8	OSR[3:0]	0x0	H0	R/W	
	7	TRUERMS_ON	0	H0	R/W	
	6	XHPF_LPF	0	H0	R/W	
	5	PEAK_ON	0	H0	R/W	
	4	XABS_SQUARE	0	H0	R/W	
	3–1	–	0	–	R	
	0	MODEN	0	H0	R/W	

Bits 15–14 Reserved

Bits 13–12 AVE_NUM[1:0]

These bits set the number of samples for averaging of square/absolute values.

Table 14.7.2 Average Sample Count Settings

DSADC16CONF.AVE_NUM[1:0] bits	Number of Average Samples
0x3	2,048
0x2	1,024
0x1	512
0x0	256

Note: Make certain that the DSADC16INIT.RUN bit is set to 0 (measurement is stopped) before altering the DSADC16CONF.AVE_NUM[1:0] bits setting. After the the DSADC16CONF.AVE_NUM[1:0] bits are altered, write 1 to the DSADC16INIT.FILTERRST bit to reset inside the filter.

Bits 11–8 OSR[3:0]

These bits set the OSR (oversampling ratio).

Table 14.7.3 OSR Settings

DSADC16CONF. OSR[3:0] bits	OSR Settings	Sampling frequency
0xb–0xf	Reserved	
0xa	16,384	24.41 Hz
0x9	8,192	48.83 Hz
0x8	4,096	97.66 Hz
0x7	2,048	195.31 Hz
0x6	1,024	390.63 Hz
0x5	512	781.25 Hz
0x4	256	1.56 kHz
0x3	128	3.12 kHz
0x2	64	6.25 kHz
0x1	32	12.5 kHz
0x0	16	25 kHz

Note: Make certain that the DSADC16INIT.RUN bit is set to 0 (measurement is stopped) before altering the DSADC16CONF.OSR[3:0] bits setting. After the the DSADC16CONF.OSR[3:0] bits are altered, write 1 to the DSADC16INIT.FILTERRST bit to reset inside the filter.

Bit 7 TRUERMS_ON

This bit enables the square root circuit.

1 (R/W): Enable square root circuit (for obtaining true RMS values)

0 (R/W): Disable square root circuit (for obtaining mean square or mean absolute values)

When this bit is set to 1, true RMS values (square root circuit outputs) will be stored to the DSADC16RMS1 and 2 registers as AC measurement results.

When this bit is set to 0, either mean square values or mean absolute values (averaging circuit outputs) that have been selected with the DSADC16CONF.XABS_SQUARE bit will be stored to the DSADC16RMS1 and 2 registers as AC measurement results.

Bit 6 XHPF_LPF

This bit enables the high-pass filter (HPF) function.

1 (R/W): Enable LPF/Disable HPF

0 (R/W): Enable LPF/Enable HPF

When this bit is set to 1, AC measurement data, which has been processed from the analog network unit output by the Comb filter, is passed through the LPF and sent to the averaging circuit. LPF processing results will be stored to the DSADC16LPFHPF register.

When this bit is set to 0, AC measurement data is passed through the HPF and sent to the averaging circuit. HPF processing results will be stored to the DSADC16LPFHPF register.

Bit 5 PEAK_ON

This bit enables the peak hold function.
1 (R/W): Enable peak hold function
0 (R/W): Disable peak hold function

While this bit is set to 1, the peak hold circuit records the measurement result if it updates the currently set maximum/minimum value. In a DC measurement mode, the maximum value is stored to the DSADC16DCPEAKMAX register and the minimum value is stored to the DSADC16DCPEAKMIN register. In an AC measurement mode, the maximum value is stored to the DSADC16ACPEAKMAX1 and 2 registers and the minimum value is stored to the DSADC16ACPEAKMIN1 and 2 registers.
When this bit is set to 0, the peak hold function becomes ineffective and the peak hold registers will not be updated. However, the peak hold registers are not cleared and they retain the held values even if this bit is set to 0. It is necessary to write 1 to the DSADC16INIT.PEAKRST bit to clear the peak hold registers. The registers are cleared after 4 μs has elapsed from this writing.

Bit 4 XABS_SQUARE

This bit selects either mean square or mean absolute. Select mean square to obtain true RMS values.
1 (R/W): Mean square (The averaging will be applied to the squared values of the LPF/HPF results.)
0 (R/W): Mean absolute (The averaging will be applied to the absolute values of the LPF/HPF results.)

Bits 3–1 Reserved

Bit 0 MODEN

This bit enables the DSADC16 operations.
1 (R/W): Enable (Start supplying operating clock)
0 (R/W): Disable (Stop supplying operating clock)

DSADC16 Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSADC16CTL	15–8	–	0x00	–	R	–
	7	–	0	–	R	
	6–4	RANGESEL[2:0]	0x0	H0	R/W	
	3–0	FUNCSEL[3:0]	0x0	H0	R/W	

Bits 15–7 Reserved

Bits 6–4 RANGESEL[2:0]

These bits set the measurement range. See Table 14.7.4 for settings.
When altering the DSADC16CTL.RANGESEL[2:0] bits during A/D conversion, write 1 to the DSADC16INIT.FILTERRST bit to clear the digital filter registers in advance.

Bits 3–0 FUNCSEL[3:0]

These bits set the measurement mode. See Table 14.7.4 for settings.
When altering the DSADC16CTL.FUNCSEL[3:0] bits during A/D conversion, set the DSADC16INIT.RUN bit to 0 to terminate the measurement and write 1 to the DSADC16INIT.FILTERRST bit to clear the digital filter registers in advance.

Table 14.7.4 Measurement Mode and Range Settings

Mode/ Range settings		DSADC16CTL.FUNCSEL[3:0] bits													
		0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd
		DCV	ACV	DCI	ACI	OHM _CC	OHM _CV	CONT	CAP _CC	CAP _CV	Diode	Freq _ACV	Freq _ACI	Temp	OFF
DSADC16CTL. RANGESEL[2:0] bits	0x0	600 mV	600 mV	600 μA	600 μA	600 Ω	600 Ω	CV	×	10 nF	Diode	600 mV	600 μA	Temp	OFF
	0x1	6 V	6 V	6 mA	6 mA	6 kΩ	6 kΩ	CC	×	100 nF	×	6 V	6 mA	×	×
	0x2	60 V	60 V	60 mA	60 mA	60 kΩ	60 kΩ	×	1 μF	×	×	60 V	60 mA	×	×
	0x3	600 V	600 V	600 mA	600 mA	600 kΩ	×	×	10 μF	×	×	600 V	600 mA	×	×
	0x4	1,000 V	1,000 V	6 A	6 A	6 MΩ	×	×	100 μF	×	×	1,000 V	6 A	×	×
	0x5	×	×	10 A	10 A	60 MΩ	×	×	1,000 μF	×	×	×	10 A	×	×
	0x6	×	×	×	×	×	×	×	×	×	×	×	×	×	×
	0x7	×	×	×	×	×	×	×	×	×	×	×	×	×	×

×: Setting prohibited (Normal IC operations cannot be guaranteed if these values are set.)

DSADC16 Initialize Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSADC16INIT	15–10	–	0x00	–	R	–
	9	PEAKRST	0	H0	R/W	
	8	FILTERRST	0	H0	R/W	
	7–2	–	0x00	–	R	
	1	RUN	0	H0	R/W	
	0	–	0	–	R	

Bits 15–10 Reserved

Bit 9 PEAKRST

This bit clears the peak hold results ((DSADC16DCPEAKMAX, DSADC16DCPEAKMIN, DSADC16ACPEAKMAX1/2, and DSADC16ACPEAKMIN1/2 registers).

1 (W): Clear the peak hold results
0 (W): Ineffective
1 (R): Clearing
0 (R): Normal operation

Bit 8 FILTERRST

This bit clears the internal registers of the digital filter as well as the result registers. Write 1 to this bit before setting the measurement mode and range.

1 (W): Clear the digital filter registers
0 (W): Ineffective
1 (R): Clearing
0 (R): Normal operation

Bit 1 RUN

This bit starts/stops A/D conversion.

1 (R/W): Start A/D conversion
0 (R/W): Stop A/D conversion

Bit 0 Reserved

DSADC16 Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSADC16IE	15–11	–	0x00	–	R	–
	10	CONTIE	0	H0	R/W	
	9	TRUERMS_OVRIE	0	H0	R/W	
	8	TRUERMSIE	0	H0	R/W	
	7	RMS_OVRIE	0	H0	R/W	
	6	HPF_OVRIE	0	H0	R/W	
	5	LPF_OVRIE	0	H0	R/W	
	4	COMB_OVRIE	0	H0	R/W	
	3	RMSIE	0	H0	R/W	
	2	HPFIE	0	H0	R/W	
	1	LPFIE	0	H0	R/W	
	0	COMBIE	0	H0	R/W	

Bits 15–11 Reserved

Bit 10 CONTIE
Bit 9 TRUERMS_OVRIE
Bit 8 TRUERMSIE
Bit 7 RMS_OVRIE
Bit 6 HPF_OVRIE
Bit 5 LPF_OVRIE
Bit 4 COMB_OVRIE
Bit 3 RMSIE
Bit 2 HPFIE
Bit 1 LPFIE
Bit 0 COMBIE

These bits enable DSADC16 interrupts.
1 (R/W): Enable interrupts
0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:

DSADC16IE.CONTIE bit:	Continuity status change detection interrupt
DSADC16IE.TRUERMS_OVRIE bit:	Square root circuit conversion result overwrite error interrupt
DSADC16IE.TRUERMSIE bit:	Square root circuit conversion completion interrupt
DSADC16IE.RMS_OVRIE bit:	Averaging circuit conversion result overwrite error interrupt
DSADC16IE.HPF_OVRIE bit:	HPF (high-pass filter) conversion result overwrite error interrupt
DSADC16IE.LPF_OVRIE bit:	LPF (low-pass filter) conversion result overwrite error interrupt
DSADC16IE.COMB_OVRIE bit:	Comb filter conversion result overwrite error interrupt
DSADC16IE.RMSIE bit:	Averaging circuit conversion completion interrupt
DSADC16IE.HPFIE bit:	HPF (high-pass filter) conversion completion interrupt
DSADC16IE.LPFIE bit:	LPF (low-pass filter) conversion completion interrupt
DSADC16IE.COMBIE bit:	Comb filter conversion completion interrupt

DSADC16 Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSADC16IF	15	TRUERMS	0	H0	R	–
	14	RMS	0	H0	R	
	13	HPF	0	H0	R	
	12	LPF	0	H0	R	
	11	COMB	0	H0	R	
	10	CONTIF	0	H0	R/W	Cleared by writing 1.
	9	TRUERMS_OVRIF	0	H0	R/W	
	8	TRUERMSIF	0	H0	R/W	
	7	RMS_OVRIF	0	H0	R/W	
	6	HPF_OVRIF	0	H0	R/W	
	5	LPF_OVRIF	0	H0	R/W	
	4	COMB_OVRIF	0	H0	R/W	
	3	RMSIF	0	H0	R/W	
	2	HPFIF	0	H0	R/W	
	1	LPFIF	0	H0	R/W	
	0	COMBIF	0	H0	R/W	

Bit 15 TRUERMS
Bit 14 RMS
Bit 13 HPF
Bit 12 LPF
Bit 11 COMB

These bits indicate the conversion execution statuses.
1 (R): Conversion is in progress.
0 (R): Idle

The following shows the correspondence between the bit and the filter/circuit:
DSADC16IF.TRUERMS bit: Square root circuit
DSADC16IF.RMS bit: Averaging circuit
DSADC16IF.HPF bit: HPF (high-pass filter)
DSADC16IF.LPF bit: LPF (low-pass filter)
DSADC16IF.COMB bit: Comb filter

Bit 10 CONTIF
Bit 9 TRUERMS_OVRIF
Bit 8 TRUERMSIF
Bit 7 RMS_OVRIF
Bit 6 HPF_OVRIF
Bit 5 LPF_OVRIF
Bit 4 COMB_OVRIF
Bit 3 RMSIF
Bit 2 HPFIF
Bit 1 LPFIF
Bit 0 COMBIF

These bits indicate the DSADC16 interrupt cause occurrence status.
1 (R): Cause of interrupt occurred
0 (R): No cause of interrupt occurred
1 (W): Clear flag
0 (W): Ineffective

The following shows the correspondence between the bit and interrupt:
DSADC16IF.CONTIF bit: Continuity status change detection interrupt
DSADC16IF.TRUERMS_OVRIF bit: Square root circuit conversion result overwrite error interrupt
DSADC16IF.TRUERMSIF bit: Square root circuit conversion completion interrupt
DSADC16IF.RMS_OVRIF bit: Averaging circuit conversion result overwrite error interrupt
DSADC16IF.HPF_OVRIF bit: HPF (high-pass filter) conversion result overwrite error interrupt
DSADC16IF.LPF_OVRIF bit: LPF (low-pass filter) conversion result overwrite error interrupt
DSADC16IF.COMB_OVRIF bit: Comb filter conversion result overwrite error interrupt
DSADC16IF.RMSIF bit: Averaging circuit conversion completion interrupt
DSADC16IF.HPFIF bit: HPF (high-pass filter) conversion completion interrupt
DSADC16IF.LPFIF bit: LPF (low-pass filter) conversion completion interrupt
DSADC16IF.COMBIF bit: Comb filter conversion completion interrupt

DSADC16 Comb Filter Result Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSADC16COMB	15–0	COMB_RESULT[15:0]	0x0000	H0	R	–

Bits 15–0 COMB_RESULT[15:0]

These bits are the conversion results (signed 16-bit value) of the Comb filter.

DSADC16 Low Pass/High Pass Filter Result Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSADC16LPFHPF	15–0	LPFHPF_RESULT[15:0]	0x0000	H0	R	–

Bits 15–0 LPFHPF_RESULT[15:0]

These bits are the conversion results (signed 16-bit value) of the LPF (low-pass filter) or HPF (high-pass filter). When the DSADC16CONF.XHPF_LPF bit is set to 1, the LPF results are stored; when it is set to 0, the HPF results are stored.

DSADC16 RMS Result Register 1

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSADC16RMS1	15–0	RMS_RESULT[15:0]	0x0000	H0	R	–

Bits 15–0 RMS_RESULT[15:0]

These bits are the AC measurement results. This register allows reading of a 16-bit true RMS value or mean absolute value, or the low-order 16 bits of a 32-bit mean square value.

DSADC16 RMS Result Register 2

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSADC16RMS2	15–0	RMS_RESULT[31:16]	0x0000	H0	R	–

Bits 15–0 RMS_RESULT[31:16]

These bits are the AC measurement results. This register allows reading of the high-order 16 bits of a 32-bit mean square value.

DSADC16 DC Peak Hold MAX Result Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSADC16 DCPEAKMAX	15–0	DCPEAKMAX[15:0]	0x8000	H0	R	–

Bits 15–0 DCPEAKMAX[15:0]

These bits are the maximum value (signed 16-bit value) held by the peak hold circuit for DC measurements.

DSADC16 DC Peak Hold MIN Result Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSADC16 DCPEAKMIN	15–0	DCPEAKMIN[15:0]	0x7fff	H0	R	–

Bits 15–0 DCPEAKMIN[15:0]

These bits are the minimum value (signed 16-bit value) held by the peak hold circuit for DC measurements.

DSADC16 AC Peak Hold MAX Result Register 1

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSADC16 ACPEAKMAX1	15–0	ACPEAKMAX[15:0]	0x0000	H0	R	–

Bits 15–0 ACPEAKMAX[15:0]

These bits are the low-order 16 bits of the maximum value held by the peak hold circuit for AC measurements.

DSADC16 AC Peak Hold MAX Result Register 2

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSADC16 ACPEAKMAX2	15–0	ACPEAKMAX[31:16]	0x0000	H0	R	–

Bits 15–0 ACPEAKMAX[31:16]

These bits are the high-order 16 bits of the maximum value held by the peak hold circuit for AC measurements.

DSADC16 AC Peak Hold MIN Result Register 1

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSADC16 ACPEAKMIN1	15–0	ACPEAKMIN[15:0]	0x0000	H0	R	–

Bits 15–0 ACPEAKMIN[15:0]

These bits are the low-order 16 bits of the minimum value held by the peak hold circuit for AC measurements.

DSADC16 AC Peak Hold MIN Result Register 2

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSADC16 ACPEAKMIN2	15–0	ACPEAKMIN[31:16]	0x0000	H0	R	–

Bits 15–0 ACPEAKMIN[31:16]

These bits are the high-order 16 bits of the minimum value held by the peak hold circuit for AC measurements.

VIR Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
VIRCTL	15–12	(reserved)	0x0	H0	R/WP	Always set to 0x0.
	11	–	0	–	R	–
	10–9	(reserved)	0x0	H0	R/WP	Always set to 0x0.
	8	DMM_XRESET	0	H0	R/W	–
	7–6	(reserved)	0x3	H0	R/WP	Always set to 0x3.
	5–3	(reserved)	0x0	H0	R/WP	Always set to 0x0.
	2	(reserved)	1	H0	R/WP	Always set to 1.
	1	(reserved)	0	H0	R/WP	Always set to 0.
	0	VIR_EN	0	H0	R/W	–

This register is automatically configured when the measurement mode and range are set using the DSADC16CTL.FUNCSEL[3:0] and DSADC16CTL.RANGESEL[2:0] bits. Therefore, it is not necessary to alter in the application program, except when specially necessary. For the setting values in each measurement mode, refer to the "Reference Data" - "Bit Settings in Each Measurement Mode" section.

Bits 15–9 Reserved

Bit 8 DMM_XRESET

This bit enables clock synchronous input to the analog block.

1 (R/W): Enable clock input

0 (R/W): Disable clock input

Bits 7–1 Reserved

Bit 0 VIR_EN

This bit enables/disables the DMMVIREF circuit to operate.

1 (R/W): Enable DMMVIREF

0 (R/W): Disable DMMVIREF

DMM Setting Register 1

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMMSET1	15	–	0	–	R	–
	14–8	DMM_SET_IODS[6:0]	0x00	H0	R/W	
	7	–	0	–	R	
	6–0	DMM_SET_IOPS[6:0]	0x00	H0	R/W	

This register is automatically configured when the measurement mode and range are set using the DSADC16CTL.FUNCSEL[3:0] and DSADC16CTL.RANGESEL[2:0] bits. Therefore, it is not necessary to alter in the application program, except when specially necessary. For the setting values in each measurement mode, refer to the "Reference Data" - "Bit Settings in Each Measurement Mode" section.

Bit 15 Reserved

Bits 14–8 DMM_SET_IODS[6:0]

These bits control the MAMP output switches in the PORTSW circuit.

Table 14.7.5 PORTSW MAMP Output Switch Control

DMMSET1.DMM_SET_IODS[6:0] bits							Switch selected
Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	0	0	0	0	0	VMIN
0	1	0	0	0	0	0	A05
0	0	1	0	0	0	0	A04
0	0	0	1	0	0	0	A03
0	0	0	0	1	0	0	A02
0	0	0	0	0	1	0	A01
0	0	0	0	0	0	1	A00

Bit 7 Reserved

Bits 6–0 DMM_SET_IOPS[6:0]

These bits control the power switches in the PORTSW circuit.

Table 14.7.6 PORTSW Power Switch Control

DMMSET1.DMM_SET_IOPS[6:0] bits							Switch selected
Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	0	0	0	0	0	VMIN
0	1	0	0	0	0	0	A05
0	0	1	0	0	0	0	A04
0	0	0	1	0	0	0	A03
0	0	0	0	1	0	0	A02
0	0	0	0	0	1	0	A01
0	0	0	0	0	0	1	A00

DMM Setting Register 2

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMMSET2	15	–	0	–	R	–
	14–8	DMM_SET_IOSS[6:0]	0x00	H0	R/W	
	7	–	0	–	R	
	6–0	DMM_SET_IOFS[6:0]	0x00	H0	R/W	

This register is automatically configured when the measurement mode and range are set using the DSADC16CTL.FUNCSEL[3:0] and DSADC16CTL.RANGESEL[2:0] bits. Therefore, it is not necessary to alter in the application program, except when specially necessary. For the setting values in each measurement mode, refer to the "Reference Data" - "Bit Settings in Each Measurement Mode" section.

Bit 15 Reserved

Bits 14–8 DMM_SET_IOSS[6:0]

These bits control the sensor switches in the PORTSW circuit.

Table 14.7.7 PORTSW Sensor Switch Control

DMMSET2.DMM_SET_IOSS[6:0] bits							Switch selected
Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	0	0	0	0	0	VMIN
0	1	0	0	0	0	0	A05
0	0	1	0	0	0	0	A04
0	0	0	1	0	0	0	A03
0	0	0	0	1	0	0	A02
0	0	0	0	0	1	0	A01
0	0	0	0	0	0	1	A00

Bit 7 Reserved

Bits 6–0 DMM_SET_IOFS[6:0]

These bits control the feedback switches in the PORTSW circuit.

Table 14.7.8 PORTSW Feedback Switch Control

DMMSET2.DMM_SET_IOFS[6:0] bits							Switch selected
Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	0	0	0	0	0	VMIN
0	1	0	0	0	0	0	A05
0	0	1	0	0	0	0	A04
0	0	0	1	0	0	0	A03
0	0	0	0	1	0	0	A02
0	0	0	0	0	1	0	A01
0	0	0	0	0	0	1	A00

DMM SMODE Setting Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SMODESET	15	–	0	–	R	–
	14–12	DMM_SET_CMPIN[2:0]	0x0	H0	R/W	
	11–8	DMM_SET_MAMPVR[3:0]	0x0	H0	R/W	
	7	–	0	–	R	
	6–0	DMM_SET_SMODE[6:0]	0x00	H0	R/W	

This register is automatically configured when the measurement mode and range are set using the DSADC16CTL.FUNCSEL[3:0] and DSADC16CTL.RANGESEL[2:0] bits. Therefore, it is not necessary to alter in the application program, except when specially necessary. For the setting values in each measurement mode, refer to the "Reference Data" - "Bit Settings in Each Measurement Mode" section.

Bit 15 Reserved

Bits 14–12 DMM_SET_CMPIN[2:0]

These bits select the comparator input voltage for the WINCMP circuit.

Table 14.7.9 WINCMP Comparator Input Voltage Selections

SMODESET.DMM_SET_CMPIN[2:0] bits	Signal selected
0x7	PB4
0x6	A21
0x5	A10
0x4	A20
0x3	A23
0x2	A22
0x1	–
0x0	SENSE

Bits 11–8 DMM_SET_MAMPVR[3:0]

These bits select the MAMP reference voltage for the DMMAFE circuit. The voltage value of the selected signal depends on the SMODESET.DMM_SET_SMODE3 and AFENET2.DMM_SET_AGNDV[2:0] bit settings.

Table 14.7.10 DMMAFE MAMP Reference Voltage Selections

SMODESET. DMM_SET_SMODE3 bit	SMODESET. DMM_SET_MAMPVR[3:0] bits	Signal selected	AFENET2. DMM_SET_AGNDV[2:0] bits			
			0x3	0x2	0x1	0x0
1	0xf	PB5	–	–	–	–
1	0xe	VDSC12	2.4 V	2.4 V	2.4 V	2.4 V
1	0xd	AGND_N6	0 V	0.2 V	0.8 V	1.4 V
1	0xc	AGND_P6	0.6 V	1.0 V	1.6 V	2.2 V
1	0xb	AGND_N7	0 V	0 V	0.6 V	1.2 V
1	0xa	AGND_P7	0.8 V	1.2 V	1.8 V	2.4 V
1	0x9	AGND_N8	0 V	0 V	0.4 V	1.0 V
1	0x8	AGND_P8	1.0 V	1.4 V	2.0 V	2.6 V
1	0x7	AGND_N9	0 V	0 V	0.2 V	0.8 V
1	0x6	AGND_P9	1.2 V	1.6 V	2.2 V	2.8 V
1	0x5	VDSC03	0.4 V	0.4 V	0.4 V	0.4 V
1	0x4	VDSC15	3.2 V	3.2 V	3.2 V	3.2 V
1	0x3	VDSC02	0.2 V	0.2 V	0.2 V	0.2 V
1	0x2	VDSC16	3.4 V	3.4 V	3.4 V	3.4 V
1	0x1	VDSC13	2.6 V	2.6 V	2.6 V	2.6 V
1	0x0	VDSC17	3.5 V	3.5 V	3.5 V	3.5 V
0	0xf	PB5	–	–	–	–
0	0xe	AGND_N6	0 V	0.2 V	0.8 V	1.4 V
0	0xd	AGND_N7	0 V	0 V	0.6 V	1.2 V
0	0xc	AGND_N8	0 V	0 V	0.4 V	1.0 V
0	0xb	AGND_N9	0 V	0 V	0.2 V	0.8 V
0	0xa	VDSC03	0.4 V	0.4 V	0.4 V	0.4 V
0	0x9	VDSC02	0.2 V	0.2 V	0.2 V	0.2 V
0	0x8	VDSC13	2.6 V	2.6 V	2.6 V	2.6 V
0	0x7	VDSC12	2.4 V	2.4 V	2.4 V	2.4 V
0	0x6	AGND_P6	0.6 V	1.0 V	1.6 V	2.2 V
0	0x5	AGND_P7	0.8 V	1.2 V	1.8 V	2.4 V
0	0x4	AGND_P8	1.0 V	1.4 V	2.0 V	2.6 V
0	0x3	AGND_P9	1.2 V	1.6 V	2.2 V	2.8 V
0	0x2	VDSC15	3.2 V	3.2 V	3.2 V	3.2 V
0	0x1	VDSC16	3.4 V	3.4 V	3.4 V	3.4 V
0	0x0	VDSC17	3.5 V	3.5 V	3.5 V	3.5 V

Bit 7 Reserved

Bits 6–0 DMM_SET_SMODE[6:0]

These bits control the switches in the MEASAMP circuit. The SMODESET.DMM_SET_SMODE6, 5, and 4 bit settings (1 = On, 0 = Off) are applied to the SMODE6, 5, and 4 switches, respectively. The SMODESET.DMM_SET_SMODE[2:0] bits controls the switches as shown in Table 14.7.11.

Table 14.7.11 Switch Selection Decode Table

SMODESET. DMM_SET_SMODE[2:0] bits	SAGND	SVDDAH	SVSSA	AMPO1, AMPM1	AMPO2, AMPM2	SCP	SCN
0x7	0	0	1	0	1	0	1
0x6	0	1	0	0	1	1	0
0x5	1	0	0	1	0	0	0
0x4	0	0	0	1	0	0	0
0x3	0	0	1	0	0	0	0
0x2	0	1	0	0	0	0	0
0x1	1	0	0	0	0	0	0
0x0	0	0	0	0	0	0	0

AFE Network Setting Register 1

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
AFENET1	15	–	0	–	R	–
	14–12	DMM_SET_FLTINN[2:0]	0x0	H0	R/W	
	11–8	DMM_SET_FLTINP[3:0]	0x0	H0	R/W	
	7–4	DMM_SET_CMPRL[3:0]	0x0	H0	R/W	
	3–0	DMM_SET_CMPRH[3:0]	0x0	H0	R/W	

This register is automatically configured when the measurement mode and range are set using the DSADC16CTL.FUNCSEL[3:0] and DSADC16CTL.RANGESEL[2:0] bits. Therefore, it is not necessary to alter in the application program, except when specially necessary. For the setting values in each measurement mode, refer to the "Reference Data" - "Bit Settings in Each Measurement Mode" section.

Bit 15 Reserved

Bits 14–12 DMM_SET_FLTINN[2:0]

These bits select the pre-filter L side input voltage for the PFILTER circuit.

Table 14.7.12 PFILTER Pre-filter L Side Input Voltage Selections

AFENET1.DMM_SET_FLTINN[2:0] bits	Signal selected
0x7	PB5
0x6	PB4
0x5	A21
0x4	A22
0x3	AGND_FB
0x2	VREFA
0x1	A11
0x0	SENSE

Bits 11–8 DMM_SET_FLTINP[3:0]

These bits select the pre-filter H side input voltage for the PFILTER circuit.

Table 14.7.13 PFILTER Pre-filter H Side Input Voltage Selections

AFENET1.DMM_SET_FLTINP[3:0] bits	Signal selected
0xf	–
0xe	–
0xd	PB5
0xc	PB4
0xb	A21
0xa	A22
0x9	A20
0x8	A23
0x7	VREF
0x6	–
0x5	VTEMP
0x4	–
0x3	–
0x2	A11
0x1	FB
0x0	SENSE

Bits 7–4 DMM_SET_CMPRL[3:0]

These bits select the comparator L side reference voltage for the WINCMP circuit.

Table 14.7.14 AFENET1.DMM_SET_CMPRL[3:0] Bit Settings in Each Measurement Mode

AFENET1. DMM_SET_CMPRL[3:0] bits	Capacitance measurement (CAP_CV, CAP_CC)	AC voltage and frequency measurement (Freq_ACV)	AC current and frequency measurement (Freq_ACI)	Continuity check (CONT)
0xf	×	×	×	1.25 V
0xe	×	×	×	1.225 V
0xd	×	×	×	1.2125 V
0xc	×	×	×	1.20625 V
0xb	×	1.8 V	1.8 V	×
0xa	×	1.79375 V	1.79375 V	×
0x9	×	1.7875 V	1.7875 V	×
0x8	×	1.775 V	1.775 V	×
0x7	×	×	×	×
0x6	1.8 V	×	×	×
0x5	1.6 V	×	×	×
0x4	1.4 V	×	×	×
0x3	1.2 V	×	×	×
0x2	1 V	×	×	×
0x1	0.8 V	×	×	×
0x0	0.4 V	×	×	×

×: Ineffective

Bits 3–0 DMM_SET_CMPRH[3:0]

These bits select the comparator H side reference voltage for the WINCMP circuit.

Table 14.7.15 AFENET1.DMM_SET_CMPRH[3:0] Bit Settings in Each Measurement Mode

AFENET1. DMM_SET_CMPRH[3:0] bits	Capacitance measurement (CAP_CV, CAP_CC)	AC voltage and frequency measurement (Freq_ACV)	AC current and frequency measurement (Freq_ACI)	Continuity check (CONT)
0xf	×	×	×	1.8 V
0xe	×	×	×	1.6 V
0xd	×	×	×	1.4 V
0xc	×	×	×	1.25 V
0xb	×	1.825 V	1.825 V	×
0xa	×	1.8125 V	1.8125 V	×
0x9	×	1.80625 V	1.80625 V	×
0x8	×	1.8 V	1.8 V	×
0x7	×	×	×	×
0x6	2.6 V	×	×	×
0x5	2.4 V	×	×	×
0x4	2.2 V	×	×	×
0x3	2 V	×	×	×
0x2	1.8 V	×	×	×
0x1	1.6 V	×	×	×
0x0	1.4 V	×	×	×

×: Ineffective

AFE Network Setting Register 2

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
AFENET2	15	–	0	–	R	–
	14–12	DMM_SET_ADVRL[2:0]	0x0	H0	R/W	
	11	–	0	–	R	
	10–8	DMM_SET_ADVRH[2:0]	0x0	H0	R/W	
	7–3	–	0x00	–	R	
	2–0	DMM_SET_AGNDV[2:0]	0x0	H0	R/W	

This register is automatically configured when the measurement mode and range are set using the DSADC16CTL. FUNCSEL[3:0] and DSADC16CTL.RANGESEL[2:0] bits. Therefore, it is not necessary to alter in the application program, except when specially necessary. For the setting values in each measurement mode, refer to the "Reference Data" - "Bit Settings in Each Measurement Mode" section.

Bit 15 Reserved

Bits 14–12 DMM_SET_ADVRL[2:0]

These bits select the L side reference voltage for the A/D conversion in the DMMAFE circuit.

Table 14.7.16 DMMAFE L Side Reference Voltage Selections for A/D Conversion

AFENET2.DMM_SET_ADVRL[2:0] bits	Signal selected
0x7	–
0x6	VREF
0x5	–
0x4	FB
0x3	PB5
0x2	A21
0x1	AGND_FB
0x0	A11

Bit 11 Reserved

Bits 10–8 DMM_SET_ADVRH[2:0]

These bits select the H side reference voltage for the A/D conversion in the DMMAFE circuit.

Table 14.7.17 DMMAFE H Side Reference Voltage Selections for A/D Conversion

AFENET2.DMM_SET_ADVRH[2:0] bits	Signal selected
0x7	–
0x6	AGND_FB
0x5	–
0x4	A11
0x3	PB4
0x2	VREF
0x1	–
0x0	FB

Bits 7–3 Reserved

Bits 2–0 DMM_SET_AGNDV[2:0]

These bits select the reference voltage for the AGNDAMP circuit.

Table 14.7.18 AGNDAMP Reference Voltage Selections

AFENET2.DMM_SET_AGNDV[2:0] bits	Voltage (V)
0x7	0
0x6	0.2
0x5	0.4
0x4	0.6
0x3	0.8
0x2	1.0
0x1	1.2
0x0	1.8

AFE Network Setting Register 3

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
AFENET3	15–14	–	0x0	–	R	–
	13–12	DMM_SET_ FLTMODE[1:0]	0x0	H0	R/W	
	11	DMM_SET_DIOSW	0	H0	R/W	
	10	DMM_EN_CMP	0	H0	R/W	
	9	DMM_EN_AGND	0	H0	R/W	
	8	DMM_EN_MAMP	0	H0	R/W	
	7–6	–	0x0	–	R	
	5–4	(reserved)	0x0	H0	R/WP	Always set to 0x0.
	3–2	–	0x0	–	R	–
	1–0	(reserved)	0x0	H0	R/WP	Always set to 0x0.

This register is automatically configured when the measurement mode and range are set using the DSADC16CTL.FUNCSEL[3:0] and DSADC16CTL.RANGESEL[2:0] bits. Therefore, it is not necessary to alter in the application program, except when specially necessary. For the setting values in each measurement mode, refer to the "Reference Data" - "Bit Settings in Each Measurement Mode" section.

Bits 15–14 Reserved

Bits 13–12 DMM_SET_FLTMODE[1:0]

These bits select the filter resistance in the PFILTER circuit.

Table 14.7.19 PFILTER Filter Resistance Selections

AFENET3.DMM_SET_FLTMODE[1:0] bits	Filter resistance
0x3	None
0x2	0 Ω
0x1	10 kΩ
0x0	100 kΩ

Bit 11 DMM_SET_DIOSW

This bit sets the DIOSW in the DMMAFE circuit.

1 (R/W): Diode V_F measurement

0 (R/W): Other measurements

Bit 10 DMM_EN_CMP

This bit enables/disables the WINCMP circuit to operate.

1 (R/W): Enable WINCMP

0 (R/W): Disable WINCMP

Bit 9 DMM_EN_AGND

This bit enables/disables the AGNDAMP circuit to operate.

1 (R/W): Enable AGNDAMP

0 (R/W): Disable AGNDAMP

Bit 8 DMM_EN_MAMP

This bit enables/disables MAMP in the DMMAFE circuit to operate.

1 (R/W): Enable MAMP

0 (R/W): Disable MAMP

Bits 7–0 Reserved

Chopper Amp Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CHPCTL	15–13	CHP_SET_CHP[2:0]	0x0	H0	R/W	–
	12	(reserved)	0	H0	R/WP	Always set to 0.
	11	–	0	–	R	–
	10–8	CHP_SET_GAIN[2:0]	0x0	H0	R/W	
	7–5	CHP_SET_BIAS[2:0]	0x0	H0	R/W	
	4–1	(reserved)	0x0	H0	R/WP	Always set to 0x0.
	0	CHP_EN	0	H0	R/W	–

For the recommended setting values of this register, refer to Table 14.4.4.1 in the "Initial Settings" section.

Bits 15–13 CHP_SET_CHP[2:0]

These bits set the chopper cycle of the CHPAMP circuit.

Table 14.7.20 CHPAMP Chopper Cycle Settings

CHPCTL.CHP_SET_CHP[2:0] bits	Chopper cycle
0x7–0x4	None
0x3	16 cycles
0x2	8 cycles
0x1	4 cycles
0x0	2 cycles

Bits 12–11 Reserved

Bits 10–8 CHP_SET_GAIN[2:0]

These bits set the amplifier gain of the CHPAMP circuit.

Table 14.7.21 CHPAMP Amplifier Gain Settings

CHPCTL.CHP_SET_GAIN[2:0] bits	Amplifier gain
0x7	–
0x6	9.5x
0x5	–
0x4	6.5x
0x3	5x
0x2	1x
0x1	0.8x
0x0	0.6x

Bits 7–5 CHP_SET_BIAS[2:0]

These bits set the bias current of the fully-differential amplifier in the CHPAMP circuit.

Table 14.7.22 CHPAMP Fully-Differential Amplifier Bias Current Settings

CHPCTL.CHP_SET_BIAS[2:0] bits	Amplifier bias current
0x7–0x4	1x
0x3	0.25x
0x2	0.375x
0x1	0.5x
0x0	0.625x

Bits 4–1 Reserved

Bit 0 CHP_EN

This bit enables/disables the CHPAMP circuit to operate.

1 (R/W): Enable CHPAMP

0 (R/W): Disable CHPAMP

Sigma Delta Modulator & ADCVCM Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DSMVCMCTL	15	–	0	–	R	–
	14–12	(reserved)	0x0	H0	R/WP	Always set to 0x0.
	11	–	0	–	R	–
	10	VCM_SET_VRMD	0	H0	R/W	
	9	VCM_EN_REF	0	H0	R/W	
	8	VCM_EN_ACM	0	H0	R/W	
	7–6	–	0x0	–	R	
	5–1	(reserved)	0x00	H0	R/WP	Always set to 0x00.
	0	DSM_EN	0	H0	R/W	–

This register is automatically configured when the measurement mode and range are set using the DSADC16CTL.FUNCSEL[3:0] and DSADC16CTL.RANGESEL[2:0] bits. Therefore, it is not necessary to alter in the application program, except when specially necessary. For the setting values in each measurement mode, refer to the "Reference Data" - "Bit Settings in Each Measurement Mode" section.

Bits 15–11 Reserved

Bit 10 VCM_SET_VRMD

This bit selects the ADCVCM circuit output voltage.
1 (R/W): External input voltage
0 (R/W): Internally generated voltage

Bit 9 VCM_EN_REF

This bit enables/disables the ADCVCM circuit to output the V_{REFP} and V_{REFN} voltages.
1 (R/W): Enable output
0 (R/W): Disable output

Bit 8 VCM_EN_ACM

This bit enables/disables the ADCVCM circuit to output the V_{ACM} voltage.
1 (R/W): Enable output
0 (R/W): Disable output

Bits 7–1 Reserved

Bit 0 DSM_EN

This bit enables/disables the DSADCMOD circuit to operate.
1 (R/W): Enable DSADCMOD
0 (R/W): Disable DSADCMOD

TSRVR Temperature Correction Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
TSRVRTEMP	15–0	TSRVR_TEMP[15:0]	0x0000	H0	R	–

Bits 15–0 TSRVR_TEMP[15:0]

These bits show the temperature correction data (signed 16-bit value) for the TSRVR circuit. It is the reference A/D conversion value at 25°C.

TSRVR Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
TSRVR	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1	(reserved)	0	H0	R/WP	Always set to 0.
	0	TSR_EN	0	H0	R/W	–

This register is automatically configured when the measurement mode and range are set using the DSADC16CTL.FUNCSEL[3:0] and DSADC16CTL.RANGESEL[2:0] bits. Therefore, it is not necessary to alter in the application program, except when specially necessary. For the setting values in each measurement mode, refer to the "Reference Data" - "Bit Settings in Each Measurement Mode" section.

Bits 15–1 Reserved

Bit 0 TSR_EN

This bit enables/disables the TSRVR circuit to operate.

1 (R/W): Enable TSRVR

0 (R/W): Disable TSRVR

Comparator Output Status Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CMPOUT	15–9	–	0	H0	R	–
	8	DMM_OUT_CMP	0	H0	R	
	7–1	–	0	H0	R	
	0	(reserved)	1	H0	R/WP	Always set to 1.

Bits 15–9 Reserved

Bit 8 DMM_OUT_CMP

This bit indicates the WINCMP circuit output status. The inverted Comparator output value is read out.

1 (R): Comparator output = 0

0 (R): Comparator output = 1

Bits 7–0 Reserved

DCDC Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DCDCCTL	15–11	(reserved)	0x00	H0	R/WP	Always set to 0x00.
	10	–	0	–	R	–
	9	(reserved)	1	H0	R/WP	Always set to 1.
	8	LDO_CP_ON	1	H0	R/W	–
	7–6	–	0x0	–	R	
	5–1	(reserved)	0x00	H0	R/WP	Always set to 0x00.
	0	CHG_EN	0	H0	R/W	–

This register is automatically configured when the measurement mode and range are set using the DSADC16CTL.FUNCSEL[3:0] and DSADC16CTL.RANGESEL[2:0] bits. Therefore, it is not necessary to alter in the application program, except when specially necessary. For the setting values in each measurement mode, refer to the "Reference Data" - "Bit Settings in Each Measurement Mode" section.

Bits 15–9 Reserved

Bit 8 LDO_CP_ON

This bit enables/disables the LDO2P0V circuit to operate.

1 (R/W): Enable LDO2P0V

0 (R/W): Disable LDO2P0V

Bits 7–1 Reserved

Bit 0 CHG_EN

This bit enables/disables the CHREG circuit to operate.

1 (R/W): Enable CHREG

0 (R/W): Disable CHREG

AFE Sub-control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
AFESUB	15–9	–	0x00	H0	R	–
	8–5	OHMCTL[3:0]	0x0	H0	R/WP	
	4-2	CHP_SET_BIAS2[2:0]	0x0	H0	R/WP	
	1–0	(reserved)	0x0	H0	R/WP	Always set to 0x0.

For the recommended setting values of this register, refer to Table 14.4.4.1 in the "Initial Settings" section.

Bits 15–9 Reserved

Bits 8–5 OHMCTL[3:0]

These bits set the analog network unit into high resistance measurement mode.

Table 14.7.23 High Resistance Measurement Setting

AFESUB.OHMCTL[3:0] bits	Setting
0xf–0x9	Reserved
0x8	High resistance measurement
0x7–0x1	Reserved
0x0	Reserved (default)

Bits 4–2 CHP_SET_BIAS2[2:0]

These bits set the bias current of the buffer amplifier in the CHPAMP circuit.

Table 14.7.24 CHPAMP Buffer Amplifier Bias Current Settings

AFESUB.CHP_SET_BIAS2[2:0] bits	Amplifier bias current
0x7–0x4	1x
0x3	0.25x
0x2	0.375x
0x1	0.5x
0x0	0.625x

Bits 1–0 Reserved

14.8 Reference Data

14.8.1 I/O Pin Equivalent Circuit

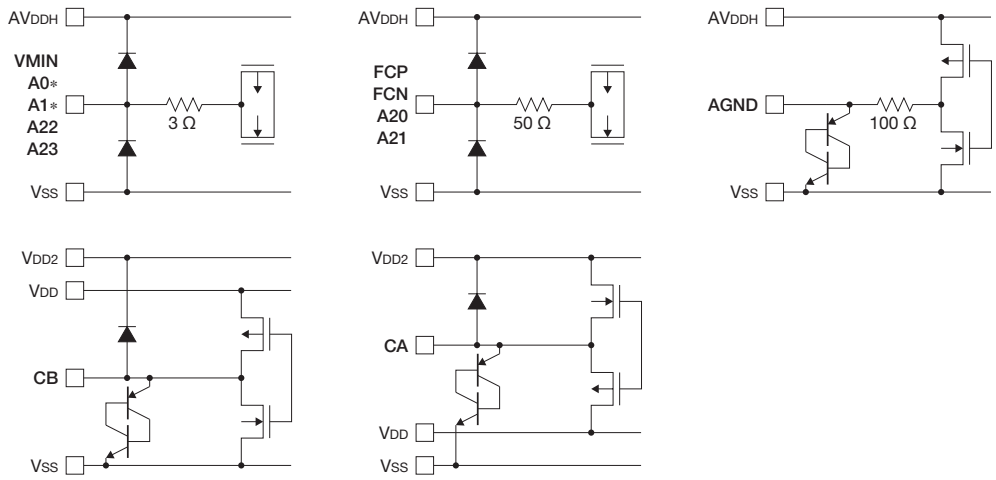


Figure 14.8.1.1 I/O Pin Equivalent Circuit

14.8.2 DMM Specification Examples

The following shows specification examples of the DMM that can be constructed with the S1C17M02/M03. (Note that the characteristics cannot be guaranteed.)

Electrical characteristics

Temperature: 0 to 50°C
Operating voltage: V_{DD} = 3 V

DC voltage measurement

Table 14.8.2.1 DC Voltage Measurement Specification Example

Range (Maximum range)	Resolution	Accuracy
600 mV	0.1 mV	±0.05% rdg. ±2 dgt.
6 V	0.001 V	
60 V	0.01 V	
600 V	0.1 V	
1,000 V	1 V	

AC voltage measurement

Table 14.8.2.2 AC Voltage Measurement Specification Example

Range (Maximum range)	Resolution	Accuracy	Frequency
600 mV	0.1 mV	±0.2% rdg. ±2 dgt.	40 to 500 Hz
6 V	0.001 V		
60 V	0.01 V		
600 V	0.1 V		
1,000 V	1 V		

DC current measurement

Table 14.8.2.3 DC Current Measurement Specification Example

Range (Maximum range)	Resolution	Accuracy
600 µA	0.1 µA	±0.2% rdg. ±2 dgt.
6 mA	0.001 mA	
60 mA	0.01 mA	
600 mA	0.1 mA	
6 A	0.001 A	
10 A	0.01 A	

Resistance accuracy: 1.0%

AC current measurement

Table 14.8.2.4 AC Current Measurement Specification Example

Range (Maximum range)	Resolution	Accuracy	Frequency
600 µA	0.1 µA	±0.6% rdg. ±5 dgt.	40 to 500 Hz
6 mA	0.001 mA		
60 mA	0.01 mA		
600 mA	0.1 mA		
6 A	0.001 A		
10 A	0.01 A		

Resistance accuracy: 1.0%

Resistance CV measurement

Table 14.8.2.5 Resistance CV Measurement Specification Example

Range (Maximum range)	Resolution	Accuracy
600 Ω	0.1 Ω	±0.5% rdg. ±1 dgt.
6 kΩ	0.001 kΩ	
60 kΩ	0.01 kΩ	

Resistance accuracy: 1.0%

Resistance CC measurement

Table 14.8.2.6 Resistance CC Measurement Specification Example

Range (Maximum range)	Resolution	Accuracy
600 Ω	0.1 Ω	±0.5% rdg. ±1 dgt.
6 kΩ	0.001 kΩ	
60 kΩ	0.01 kΩ	
600 kΩ	0.1 kΩ	
6 MΩ	0.001 MΩ	
60 MΩ	0.01 MΩ	

Resistance accuracy: 1.0%

Capacitance measurement

Table 14.8.2.7 Capacitance Measurement Specification Example

Range (Maximum range)	Resolution	Accuracy
10 nF	0.01 nF	±0.9% rdg. ±1 dgt.
100 nF	0.1 nF	
1 µF	0.001 µF	
10 µF	0.01 µF	
100 µF	0.1 µF	
1,000 µF	1 µF	

Frequency measurement

Table 14.8.2.8 Frequency Measurement Specification Example

Range (Maximum range)	Resolution	Accuracy
99.99 Hz	0.01 Hz	±0.1% rdg. ±1 dgt.
999.9 Hz	0.1 Hz	
9.999 kHz	0.001 kHz	
99.99 kHz	0.01 kHz	

Continuity check

Table 14.8.2.9 Continuity Check Specification Example

Detection resistance	Cancellation resistance
12.6 Ω	250 Ω

Internal temperature measurement

Table 14.8.2.10 Internal Temperature Measurement Specification Example

Condition	Accuracy
Ta = 0 to 50°C	±4°C
Ta = -40 to 85°C	±6°C

14.8.3 Bit Settings in Each Measurement Mode

The following tables list the control bit settings for the analog network unit. These bits are automatically configured according to the DSADC16CTL.FUNCSEL[3:0] and DSADC16CTL.RANGESEL[2:0] bit settings (measurement mode and measurement range specified). For details of the control bits, refer to the "Control Registers" section.

Table 14.8.3.1 Bit Settings (DC voltage measurement, AC voltage measurement)

Bit name	Bit settings by mode and range									
	DCV					ACV				
	600 mV	6 V	60 V	600 V	1,000 V	600 mV	6 V	60 V	600 V	1,000 V
VIRCTL.DMM_XRESET bit	1	1	1	1	1	1	1	1	1	1
VIRCTL.VIR_EN bit	1	1	1	1	1	1	1	1	1	1
DMMSET1.DMM_SET_IODS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
DMMSET1.DMM_SET_IOPS[6:0] bits	0x40	0x60	0x48	0x42	0x41	0x40	0x60	0x48	0x42	0x41
DMMSET2.DMM_SET_IOSS[6:0] bits	0x00	0x20	0x08	0x02	0x01	0x00	0x20	0x08	0x02	0x01
DMMSET2.DMM_SET_IOFS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
SMODSET.DMM_SET_CMPIN[2:0] bits	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
SMODSET.DMM_SET_MAMPVR[3:0] bits	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
SMODSET.DMM_SET_SMODE[6:0] bits	0x11	0x20	0x20	0x20	0x20	0x11	0x20	0x20	0x20	0x20
AFENET1.DMM_SET_FLTINN[2:0] bits	0x5	0x1	0x1	0x1	0x1	0x5	0x1	0x1	0x1	0x1
AFENET1.DMM_SET_FLTINP[3:0] bits	0x2	0x0	0x0	0x0	0x0	0x2	0x0	0x0	0x0	0x0
AFENET1.DMM_SET_CMPRL[3:0] bits	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
AFENET1.DMM_SET_CMPRH[3:0] bits	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
AFENET2.DMM_SET_ADVRL[2:0] bits	0x7	0x7	0x7	0x7	0x7	0x7	0x7	0x7	0x7	0x7
AFENET2.DMM_SET_ADVRH[2:0] bits	0x7	0x7	0x7	0x7	0x7	0x7	0x7	0x7	0x7	0x7
AFENET2.DMM_SET_AGNDV[2:0] bits	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
AFENET3.DMM_SET_FLTMODE[1:0] bits	0x0	0x0	0x0	0x0	0x0	0x3	0x3	0x3	0x3	0x3
AFENET3.DMM_SET_DIOSW bit	0	0	0	0	0	0	0	0	0	0
AFENET3.DMM_EN_CMP bit	0	0	0	0	0	0	0	0	0	0
AFENET3.DMM_EN_AGND bit	1	1	1	1	1	1	1	1	1	1
AFENET3.DMM_EN_MAMP bit	0	0	0	0	0	0	0	0	0	0
DSMVCMCTL.VCM_SET_VRMD bit	0	0	0	0	0	0	0	0	0	0
DSMVCMCTL.VCM_EN_REF bit	1	1	1	1	1	1	1	1	1	1
DSMVCMCTL.VCM_EN_ACM bit	1	1	1	1	1	1	1	1	1	1
DSMVCMCTL.DSM_EN bit	1	1	1	1	1	1	1	1	1	1
TSRVR.TSR_EN bit	0	0	0	0	0	0	0	0	0	0
DCDCCTL.LDP_CP_ON bit	1	1	1	1	1	1	1	1	1	1
DCDCCTL.CHG_EN bit	1	1	1	1	1	1	1	1	1	1

Table 14.8.3.2 Bit Settings (Resistance measurement, Continuity check)

Bit name	Bit settings by mode and range										
	OHM									CONT	
	CV			CC						CV	CC
	600 Ω	6 kΩ	60 kΩ	600 Ω	6 kΩ	60 kΩ	600 kΩ	6 MΩ	60 MΩ	–	–
VIRCTL.DMM_XRESET bit	1	1	1	1	1	1	1	1	1	1	1
VIRCTL.VIR_EN bit	1	1	1	1	1	1	1	1	1	1	1
DMMSET1.DMM_SET_IODS[6:0] bits	0x01	0x02	0x08	0x00	0x00	0x00	0x00	0x00	0x00	0x01	0x00
DMMSET1.DMM_SET_IOPS[6:0] bits	0x00	0x00	0x00	0x01	0x01	0x02	0x18	0x30	0x10	0x00	0x01
DMMSET2.DMM_SET_IOSS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
DMMSET2.DMM_SET_IOFS[6:0] bits	0x01	0x02	0x08	0x00	0x00	0x00	0x00	0x00	0x00	0x01	0x00
SMODSET.DMM_SET_CMPIN[2:0] bits	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x3	0x3
SMODSET.DMM_SET_MAMPVR[3:0] bits	0x7	0x7	0x7	0x8	0x1	0x1	0x1	0x1	0x1	0x7	0x8
SMODSET.DMM_SET_SMODE[6:0] bits	0x14	0x14	0x14	0x06	0x06	0x06	0x06	0x06	0x06	0x14	0x14
AFENET1.DMM_SET_FLTINN[2:0] bits	0x5	0x5	0x5	0x5	0x5	0x5	0x5	0x5	0x5	0x5	0x5
AFENET1.DMM_SET_FLTINP[3:0] bits	0x8	0x8	0x8	0x8	0x8	0x8	0x8	0x8	0x8	0x8	0x8
AFENET1.DMM_SET_CMPRL[3:0] bits	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0xD	0xD
AFENET1.DMM_SET_CMPRH[3:0] bits	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0xD	0xD
AFENET2.DMM_SET_ADVRL[2:0] bits	0x0	0x0	0x0	0x7	0x7	0x7	0x7	0x7	0x7	0x0	0x7
AFENET2.DMM_SET_ADVRH[2:0] bits	0x0	0x0	0x0	0x7	0x7	0x7	0x7	0x7	0x7	0x0	0x7
AFENET2.DMM_SET_AGNDV[2:0] bits	0x1	0x1	0x1	0x4	0x4	0x4	0x4	0x4	0x4	0x1	0x4
AFENET3.DMM_SET_FLTMODE[1:0] bits	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
AFENET3.DMM_SET_DIOSW bit	0	0	0	0	0	0	0	0	0	0	0
AFENET3.DMM_EN_CMP bit	0	0	0	0	0	0	0	0	0	1	1
AFENET3.DMM_EN_AGND bit	1	1	1	1	1	1	1	1	1	1	1
AFENET3.DMM_EN_MAMP bit	1	1	1	1	1	1	1	1	1	1	1
DSMVCMCTL.VCM_SET_VRMD bit	1	1	1	0	0	0	0	0	0	1	0
DSMVCMCTL.VCM_EN_REF bit	1	1	1	1	1	1	1	1	1	1	1
DSMVCMCTL.VCM_EN_ACM bit	1	1	1	1	1	1	1	1	1	1	1
DSMVCMCTL.DSM_EN bit	1	1	1	1	1	1	1	1	1	1	1
TSRVR.TSR_EN bit	0	0	0	0	0	0	0	0	0	0	0
DCDCCTL.LDP_CP_ON bit	1	1	1	1	1	1	1	1	1	1	1
DCDCCTL.CHG_EN bit	1	1	1	1	1	1	1	1	1	1	1

Table 14.8.3.3 Bit Settings (Capacitance measurement, Diode VF measurement)

Bit name	Bit settings by mode and range						
	CAP						Diode
	CV		CC				–
	10 nF	100 nF	1 µF	10 µF	100 µF	1,000 µF	–
VIRCTL.DMM_XRESET bit	1	1	1	1	1	1	1
VIRCTL.VIR_EN bit	1	1	1	1	1	1	1
DMMSET1.DMM_SET_IODS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00	0x00
DMMSET1.DMM_SET_IOPS[6:0] bits	0x08	0x08	0x18	0x02	0x01	0x01	0x01
DMMSET2.DMM_SET_IOSS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00	0x00
DMMSET2.DMM_SET_IOFS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00	0x00
SMODSET.DMM_SET_CMPIN[2:0] bits	0x3	0x3	0x3	0x3	0x3	0x3	0x0
SMODSET.DMM_SET_MAMPVR[3:0] bits	0x0	0x0	0x3	0x3	0x3	0x3	0x2
SMODSET.DMM_SET_SMODE[6:0] bits	0x1a	0x1a	0x0e	0x0e	0x0e	0x0e	0x06
AFENET1.DMM_SET_FLTINN[2:0] bits	0x7	0x7	0x7	0x7	0x7	0x7	0x5
AFENET1.DMM_SET_FLTINP[3:0] bits	0xF	0xF	0xF	0xF	0xF	0xF	0x8
AFENET1.DMM_SET_CMPRL[3:0] bits	0x2	0x2	0x4	0x4	0x4	0x5	0x0
AFENET1.DMM_SET_CMPRH[3:0] bits	0x4	0x4	0x2	0x2	0x2	0x2	0x0
AFENET2.DMM_SET_ADVRL[2:0] bits	0x7	0x7	0x7	0x7	0x7	0x7	0x7
AFENET2.DMM_SET_ADVRH[2:0] bits	0x7	0x7	0x7	0x7	0x7	0x7	0x7
AFENET2.DMM_SET_AGNDV[2:0] bits	0x0	0x0	0x0	0x0	0x0	0x0	0x6
AFENET3.DMM_SET_FLTMODE[1:0] bits	0	0	0	0	0	0	0
AFENET3.DMM_SET_DIOSW bit	0	0	0	0	0	0	1
AFENET3.DMM_EN_CMP bit	1	1	1	1	1	1	0
AFENET3.DMM_EN_AGND bit	1	1	1	1	1	1	1
AFENET3.DMM_EN_MAMP bit	0	0	1	1	1	1	1
DSMVCMCTL.VCM_SET_VRMD bit	0	0	0	0	0	0	0
DSMVCMCTL.VCM_EN_REF bit	0	0	0	0	0	0	1
DSMVCMCTL.VCM_EN_ACM bit	0	0	0	0	0	0	1
DSMVCMCTL.DSM_EN bit	0	0	0	0	0	0	1
TSRVR.TSR_EN bit	0	0	0	0	0	0	0
DCDCCTL.LDP_CP_ON bit	1	1	1	1	1	1	1
DCDCCTL.CHG_EN bit	1	1	1	1	1	1	1

Table 14.8.3.4 Bit Settings (DC current measurement)

Bit name	Bit settings by mode and range					
	DCI					
	600 µA	6 mA	60 mA	600 mA	6 A	10 A
VIRCTL.DMM_XRESET bit	1	1	1	1	1	1
VIRCTL.VIR_EN bit	1	1	1	1	1	1
DMMSET1.DMM_SET_IODS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00
DMMSET1.DMM_SET_IOPS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00
DMMSET2.DMM_SET_IOSS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00
DMMSET2.DMM_SET_IOFS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00
SMODSET.DMM_SET_CMPIN[2:0] bits	0x0	0x0	0x0	0x0	0x0	0x0
SMODSET.DMM_SET_MAMPVR[3:0] bits	0x0	0x0	0x0	0x0	0x0	0x0
SMODSET.DMM_SET_SMODE[6:0] bits	0x20	0x20	0x20	0x20	0x20	0x20
AFENET1.DMM_SET_FLTINN[2:0] bits	0x5	0x5	0x5	0x5	0x5	0x5
AFENET1.DMM_SET_FLTINP[3:0] bits	0x9	0x9	0x9	0x9	0x9	0x9
AFENET1.DMM_SET_CMPRL[3:0] bits	0x0	0x0	0x0	0x0	0x0	0x0
AFENET1.DMM_SET_CMPRH[3:0] bits	0x0	0x0	0x0	0x0	0x0	0x0
AFENET2.DMM_SET_ADVRL[2:0] bits	0x7	0x7	0x7	0x7	0x7	0x7
AFENET2.DMM_SET_ADVRH[2:0] bits	0x7	0x7	0x7	0x7	0x7	0x7
AFENET2.DMM_SET_AGNDV[2:0] bits	0x0	0x0	0x0	0x0	0x0	0x0
AFENET3.DMM_SET_FLTMODE[1:0] bits	0x0	0x0	0x0	0x0	0x0	0x0
AFENET3.DMM_SET_DIOSW bit	0	0	0	0	0	0
AFENET3.DMM_EN_CMP bit	0	0	0	0	0	0
AFENET3.DMM_EN_AGND bit	1	1	1	1	1	1
AFENET3.DMM_EN_MAMP bit	0	0	0	0	0	0
DSMVCMCTL.VCM_SET_VRMD bit	0	0	0	0	0	0
DSMVCMCTL.VCM_EN_REF bit	1	1	1	1	1	1
DSMVCMCTL.VCM_EN_ACM bit	1	1	1	1	1	1
DSMVCMCTL.DSM_EN bit	1	1	1	1	1	1
TSRVR.TSR_EN bit	0	0	0	0	0	0
DCDCCTL.LDP_CP_ON bit	1	1	1	1	1	1
DCDCCTL.CHG_EN bit	1	1	1	1	1	1

Table 14.8.3.5 Bit Settings (AC current measurement)

Bit name	Bit settings by mode and range					
	ACI					
	600 µA	6 mA	60 mA	600 mA	6 A	10 A
VIRCTL.DMM_XRESET bit	1	1	1	1	1	1
VIRCTL.VIR_EN bit	1	1	1	1	1	1
DMMSET1.DMM_SET_IODS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00
DMMSET1.DMM_SET_IOPS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00
DMMSET2.DMM_SET_IOSS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00
DMMSET2.DMM_SET_IOFS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00
SMODSET.DMM_SET_CMPIN[2:0] bits	0x0	0x0	0x0	0x0	0x0	0x0
SMODSET.DMM_SET_MAMPVR[3:0] bits	0x0	0x0	0x0	0x0	0x0	0x0
SMODSET.DMM_SET_SMODE[6:0] bits	0x20	0x20	0x20	0x20	0x20	0x20
AFENET1.DMM_SET_FLTINN[2:0] bits	0x5	0x5	0x5	0x5	0x5	0x5
AFENET1.DMM_SET_FLTINP[3:0] bits	0x9	0x9	0x9	0x9	0x9	0x9
AFENET1.DMM_SET_CMPRL[3:0] bits	0x0	0x0	0x0	0x0	0x0	0x0
AFENET1.DMM_SET_CMPRH[3:0] bits	0x0	0x0	0x0	0x0	0x0	0x0
AFENET2.DMM_SET_ADVRL[2:0] bits	0x7	0x7	0x7	0x7	0x7	0x7
AFENET2.DMM_SET_ADVRH[2:0] bits	0x7	0x7	0x7	0x7	0x7	0x7
AFENET2.DMM_SET_AGNDV[2:0] bits	0x0	0x0	0x0	0x0	0x0	0x0
AFENET3.DMM_SET_FLTMODE[1:0] bits	0x3	0x3	0x3	0x3	0x3	0x3
AFENET3.DMM_SET_DIOSW bit	0	0	0	0	0	0
AFENET3.DMM_EN_CMP bit	0	0	0	0	0	0
AFENET3.DMM_EN_AGND bit	1	1	1	1	1	1
AFENET3.DMM_EN_MAMP bit	0	0	0	0	0	0
DSMVCMCTL.VCM_SET_VRMD bit	0	0	0	0	0	0
DSMVCMCTL.VCM_EN_REF bit	1	1	1	1	1	1
DSMVCMCTL.VCM_EN_ACM bit	1	1	1	1	1	1
DSMVCMCTL.DSM_EN bit	1	1	1	1	1	1
TSRVR.TSR_EN bit	0	0	0	0	0	0
DCDCCTL.LDP_CP_ON bit	1	1	1	1	1	1
DCDCCTL.CHG_EN bit	1	1	1	1	1	1

Table 14.8.3.6 Bit Settings (AC current and frequency measurement)

Bit name	Bit settings by mode and range					
	Freq_ACI					
	600 µA	6 mA	60 mA	600 mA	6 A	10 A
VIRCTL.DMM_XRESET bit	1	1	1	1	1	1
VIRCTL.VIR_EN bit	1	1	1	1	1	1
DMMSET1.DMM_SET_IODS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00
DMMSET1.DMM_SET_IOPS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00
DMMSET2.DMM_SET_IOSS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00
DMMSET2.DMM_SET_IOFS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00
SMODSET.DMM_SET_CMPIN[2:0] bits	0x6	0x6	0x6	0x6	0x6	0x6
SMODSET.DMM_SET_MAMPVR[3:0] bits	0x0	0x0	0x0	0x0	0x0	0x0
SMODSET.DMM_SET_SMODE[6:0] bits	0x20	0x20	0x20	0x20	0x20	0x20
AFENET1.DMM_SET_FLTINN[2:0] bits	0x5	0x5	0x5	0x5	0x5	0x5
AFENET1.DMM_SET_FLTINP[3:0] bits	0x9	0x9	0x9	0x9	0x9	0x9
AFENET1.DMM_SET_CMPRL[3:0] bits	0x0	0xa	0xa	0xa	0xa	0xa
AFENET1.DMM_SET_CMPRH[3:0] bits	0x0	0x9	0x9	0x9	0x9	0x9
AFENET2.DMM_SET_ADVRL[2:0] bits	0x7	0x7	0x7	0x7	0x7	0x7
AFENET2.DMM_SET_ADVRH[2:0] bits	0x7	0x7	0x7	0x7	0x7	0x7
AFENET2.DMM_SET_AGNDV[2:0] bits	0x0	0x0	0x0	0x0	0x0	0x0
AFENET3.DMM_SET_FLTMODE[1:0] bits	0x3	0x3	0x3	0x3	0x3	0x3
AFENET3.DMM_SET_DIOSW bit	0	0	0	0	0	0
AFENET3.DMM_EN_CMP bit	1	1	1	1	1	1
AFENET3.DMM_EN_AGND bit	1	1	1	1	1	1
AFENET3.DMM_EN_MAMP bit	0	0	0	0	0	0
DSMVCMCTL.VCM_SET_VRMD bit	0	0	0	0	0	0
DSMVCMCTL.VCM_EN_REF bit	1	1	1	1	1	1
DSMVCMCTL.VCM_EN_ACM bit	1	1	1	1	1	1
DSMVCMCTL.DSM_EN bit	1	1	1	1	1	1
TSRVR.TSR_EN bit	0	0	0	0	0	0
DCDCCTL.LDP_CP_ON bit	1	1	1	1	1	1
DCDCCTL.CHG_EN bit	1	1	1	1	1	1

Table 14.8.3.7 Bit Settings (AC voltage and frequency measurement, internal temperature measurement)

Bit name	Bit settings by mode and range					
	Freq_ACV					Temp
	600 mV	6 V	60 V	600 V	1,000 V	–
VIRCTL.DMM_XRESET bit	1	1	1	1	1	1
VIRCTL.VIR_EN bit	1	1	1	1	1	1
DMMSET1.DMM_SET_IODS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00
DMMSET1.DMM_SET_IOPS[6:0] bits	0x40	0x60	0x48	0x42	0x41	0x00
DMMSET2.DMM_SET_IOSS[6:0] bits	0x00	0x20	0x08	0x02	0x01	0x00
DMMSET2.DMM_SET_IOFS[6:0] bits	0x00	0x00	0x00	0x00	0x00	0x00
SMODSET.DMM_SET_CMPIN[2:0] bits	0x0	0x0	0x0	0x0	0x0	0x0
SMODSET.DMM_SET_MAMPVR[3:0] bits	0x0	0x0	0x0	0x0	0x0	0x0
SMODSET.DMM_SET_SMODE[6:0] bits	0x11	0x20	0x20	0x20	0x20	0x20
AFENET1.DMM_SET_FLTINN[2:0] bits	0x5	0x1	0x1	0x1	0x1	0x2
AFENET1.DMM_SET_FLTINP[3:0] bits	0x2	0x0	0x0	0x0	0x0	0x5
AFENET1.DMM_SET_CMPRL[3:0] bits	0x0	0xa	0xa	0xa	0xa	0x0
AFENET1.DMM_SET_CMPRH[3:0] bits	0x0	0x9	0x9	0x9	0x9	0x0
AFENET2.DMM_SET_ADVRL[2:0] bits	0x7	0x7	0x7	0x7	0x7	0x7
AFENET2.DMM_SET_ADVRH[2:0] bits	0x7	0x7	0x7	0x7	0x7	0x7
AFENET2.DMM_SET_AGNDV[2:0] bits	0x0	0x0	0x0	0x0	0x0	0x0
AFENET3.DMM_SET_FLTMODE[1:0] bits	0x3	0x3	0x3	0x3	0x3	0x0
AFENET3.DMM_SET_DIOSW bit	0	0	0	0	0	0
AFENET3.DMM_EN_CMP bit	0	1	1	1	1	0
AFENET3.DMM_EN_AGND bit	1	1	1	1	1	0
AFENET3.DMM_EN_MAMP bit	0	0	0	0	0	0
DSMVCMCTL.VCM_SET_VRMD bit	0	0	0	0	0	0
DSMVCMCTL.VCM_EN_REF bit	1	1	1	1	1	1
DSMVCMCTL.VCM_EN_ACM bit	1	1	1	1	1	1
DSMVCMCTL.DSM_EN bit	1	1	1	1	1	1
TSRVR.TSR_EN bit	0	0	0	0	0	1
DCDCCTL.LDP_CP_ON bit	1	1	1	1	1	1
DCDCCTL.CHG_EN bit	1	1	1	1	1	1

15 DMM 16-bit PWM Timers (T16B_DMM)

Note: T16B_DMM is a frequency measurement circuit dedicated for the DMM controller. It cannot be used as a general-purpose timer. This chapter describes the operations as a 16-bit PWM timer. For the operations and settings of frequency measurement, refer to the "DMM Controller" chapter.

15.1 Overview

T16B_DMM is a 16-bit PWM timer with comparator/capture functions. The features of T16B_DMM are listed below.

- Counter block
  - 16-bit up/down counter
  - A clock source and a clock division ratio for generating the count clock are selectable in each channel.
  - The count mode is configurable from combinations of up, down, or up/down count operations, and one-shot operations (counting for one cycle configured) or repeat operations (counting continuously until stopped via software).
  - Supports an event counter function using an external clock.
- Comparator/capture block
  - Supports up to six comparator/capture circuits to be included per one channel.
  - The comparator compares the counter value with the values specified via software to generate interrupt signals and a PWM waveform. (Can be used as an interval timer, PWM waveform generator, and external event counter.)
  - The capture circuit captures counter values using external/software trigger signals and generates interrupts. (Can be used to measure external event periods/cycles.)

Figure 15.1.1 shows the T16B_DMM configuration.

Table 15.1.1 T16B_DMM Channel Configuration of S1C17M02/M03

Item	S1C17M02	S1C17M03
Number of channels	3 channels (Ch.0–Ch.2)	
Event counter function	Ch.0: Not used. Ch.1: EXCL10 input Ch.2: Not used.	
Number of comparator/ capture circuits per channel	2 systems (0 and 1)	
Timer generating signal output	Ch.0: TOUT00 output Ch.1: Not used. Ch.2: Not used.	
Capture signal input	Ch.0: Not used. Ch.1: CAP10 input Ch.2: CAP20 input	

Note: In this chapter, '*n*' refers to a channel number, and '*m*' refers to an input/output pin number or a comparator/capture circuit number in a channel.

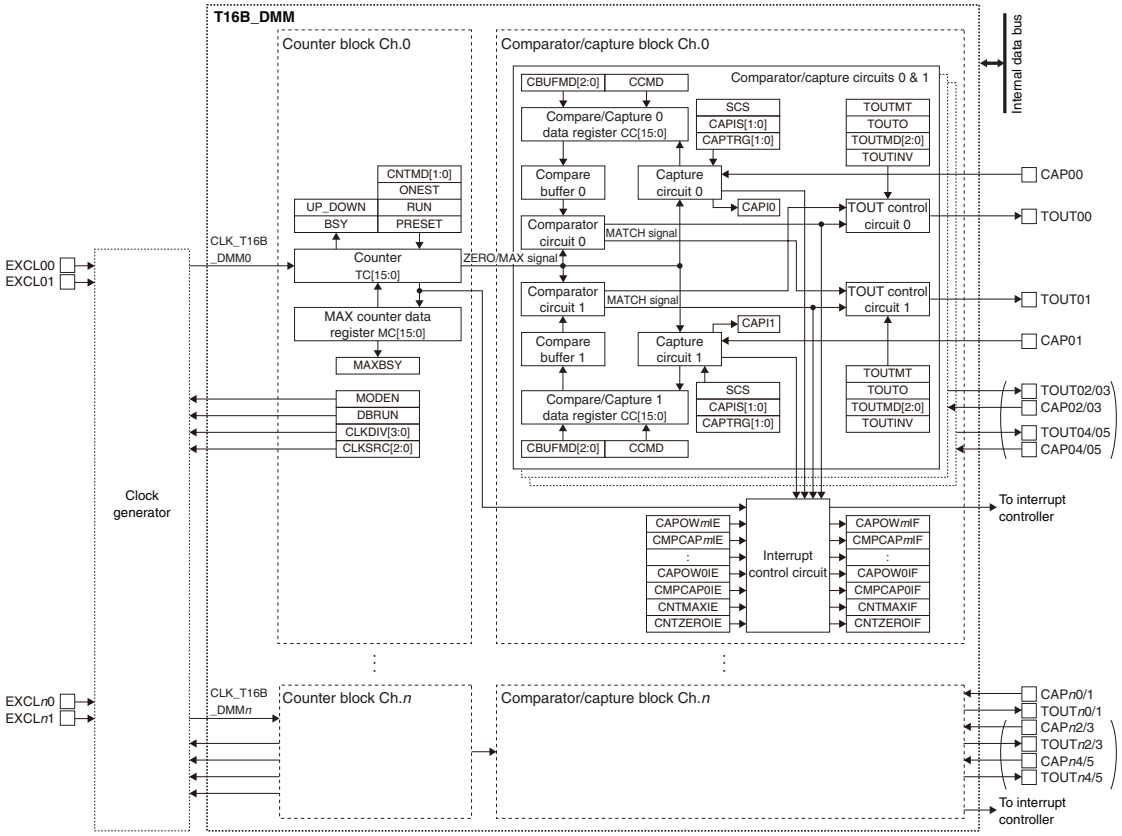


Figure 15.1.1 T16B_DMM Configuration

15.2 Input/Output Pins

Table 15.2.1 lists the T16B_DMM pins.

Table 15.2.1 List of T16B_DMM Pins

Pin name	I/O*	Initial status*	Function
EXCL*nm*	I	I (Hi-Z)	External clock input
TOUT*nm*/CAP*nm*	O or I	O (L)	TOUT signal output (in comparator mode) or capture trigger signal input (in capture mode)

* Indicates the status when the pin is configured for T16B_DMM.

If the port is shared with the T16B_DMM pin and other functions, the T16B_DMM input/output function must be assigned to the port before activating T16B_DMM. For more information, refer to the "I/O Ports" chapter.

Note: The T16B_DMM has no external input/output pins. The inputs/outputs of the channels are connected inside the MCU for DMM frequency measurements (refer to the "Frequency/Capacitance Measurement Function" section in the "DMM Controller" chapter).

15.3 Clock Settings

15.3.1 T16B_DMM Operating Clock

When using T16B_DMM Ch.*n*, the T16B_DMM Ch.*n* operating clock CLK_T16B_DMM*n* must be supplied to T16B_DMM Ch.*n* from the clock generator. The CLK_T16B_DMM*n* supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to "Clock Generator" in the "Power Supply, Reset, and Clocks" chapter).

   When an external clock is used, select the EXCL*nm* pin function (refer to the "I/O Ports" chapter).
2. Set the following T16B*n*CLK register bits:
   - T16B*n*CLK.CLKSRC[2:0] bits (Clock source selection)
   - T16B*n*CLK.CLKDIV[3:0] bits (Clock division ratio selection = Clock frequency setting)

15.3.2 Clock Supply in SLEEP Mode

When using T16B_DMM during SLEEP mode, the T16B_DMM operating clock CLK_T16B_DMM*n* must be configured so that it will keep supplying by writing 0 to the CLGOSC.*xxxx*SLPC bit for the CLK_T16B_DMM*n* clock source.
If the CLGOSC.*xxxx*SLPC bit for the CLK_T16B_DMM*n* clock source is 1, the CLK_T16B_DMM*n* clock source is deactivated during SLEEP mode and T16B_DMM stops with the register settings and counter value maintained at those before entering SLEEP mode. After the CPU returns to normal mode, CLK_T16B_DMM*n* is supplied and the T16B_DMM operation resumes.

15.3.3 Clock Supply in DEBUG Mode

The CLK_T16B_DMM*n* supply during DEBUG mode should be controlled using the T16B*n*CLK.DBRUN bit.
The CLK_T16B_DMM*n* supply to T16B_DMM Ch.*n* is suspended when the CPU enters DEBUG mode if the T16B*n*CLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_T16B_DMM*n* supply resumes. Although T16B_DMM Ch.*n* stops operating when the CLK_T16B_DMM*n* supply is suspended, the counter and registers retain the status before DEBUG mode was entered. If the T16B*n*CLK.DBRUN bit = 1, the CLK_T16B_DMM*n* supply is not suspended and T16B_DMM Ch.*n* will keep operating in DEBUG mode.

15.3.4 Event Counter Clock

When EXCL*nm* is selected as the clock source using the T16B*n*CLK.CLKSRC[2:0] bits, the channel functions as a timer or event counter that counts the EXCL*nm* pin input clocks.
The counter counts rising edges of the input signal. This can be changed so that the counter will count falling edges of the original signal by selecting EXCL*nm* inverted input as the clock source.

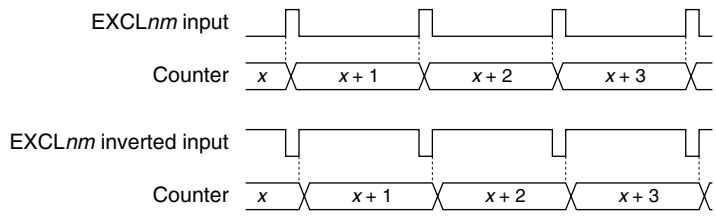


Figure 15.3.4.1 Count Timing (During Count Up Operation)

Note: When running the counter using the event counter clock, two dummy clocks must be input before the first counting up/down can be performed.

15.4 Operations

15.4.1 Counter Block Operations

The counter in each counter block channel is a 16-bit up/down counter that counts the selected operating clock (count clock).

Count mode

The T16B*n*CTL.CNTMD[1:0] bits allow selection of up, down, and up/down mode. The T16B*n*CTL.ONEST bit allows selection of repeat and one-shot mode. The counter operates in six counter modes specified with a combination of these modes.

Repeat mode enables the counter to continue counting until stopped via software. Select this mode to generate periodic interrupts at desired intervals or to generate timer output waveforms.

One-shot mode enables the counter to stop automatically. Select this mode to stop the counter after an interrupt has occurred once, such as for measuring pulse width or external event intervals and checking a specific lapse of time.

Up, down, and up/down mode configures the counter as an up counter, down counter and up/down counter, respectively.

MAX counter data register

The MAX counter data register (T16B*n*MC.MC[15:0] bits) is used to set the maximum value of the counter (hereafter referred to as MAX value). This setting limits the count range to 0x0000–MAX value and determines the count and interrupt cycles. When the counter is set to repeat mode, the MAX value can be rewritten in the procedure shown below even if the counter is running.

1. Check to see if the T16B*n*CTL.MAXBSY bit is set to 0.
2. Write the MAX value to the T16B*n*MC.MC[15:0] bits.

Note: When rewriting the MAX value, the new MAX value should be written after the counter has been reset to the previously set MAX value.

Counter reset

Setting the T16B*n*CTL.PRESET bit to 1 resets the counter. This clears the counter to 0x0000 in up or up/down mode, or presets the MAX value to the counter in down mode.

The counter is also cleared to 0x0000 when the counter value exceeds the MAX value during count up operation.

Counting start

To start counting, set the T16B*n*CTL.RUN bit to 1. The counting stop control depends on the count mode set.

Counter value read

The counter value can be read out from the T16B*n*TC.TC[15:0] bits. However, since T16B_DMM operates on CLK_T16B_DMM*n*, one of the operations shown below is required to read correctly by the CPU.

- Read the counter value twice or more and check to see if the same value is read.
- Stop the timer and then read the counter value.

Counter status check

The counter operating status can be checked using the T16B*n*CS.BSY bit. The T16B*n*CS.BSY bit is set to 1 while the counter is running or 0 while the counter is idle.

The current count direction can also be checked using the T16B*n*CS.UP_DOWN bit. The T16B*n*CS.UP_DOWN bit is set to 1 during count up operation or 0 during count down operation.

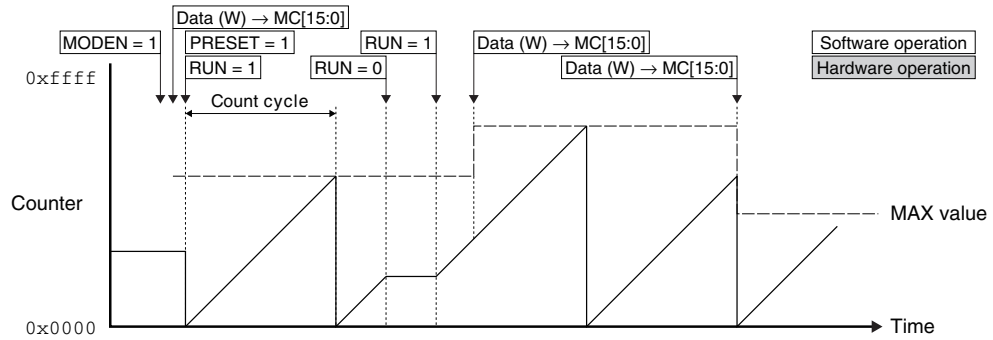
Operations in repeat up count and one-shot up count modes

In these modes, the counter operates as an up counter and counts from 0x0000 (or current value) to the MAX value.

In repeat up count mode, the counter returns to 0x0000 if it exceeds the MAX value and continues counting until the T16B*n*CTL.RUN bit is set to 0.

If the MAX value is altered to a value larger than the current counter value during counting, the counter keeps counting up to the new MAX value. If the MAX value is altered to a value smaller than the current counter value, the counter is cleared to 0x0000 and continues counting up to the new MAX value.
In one-shot up count mode, the counter returns to 0x0000 if it exceeds the MAX value and stops automatically at that point.

(1) Repeat up count mode



(2) One-shot up count mode

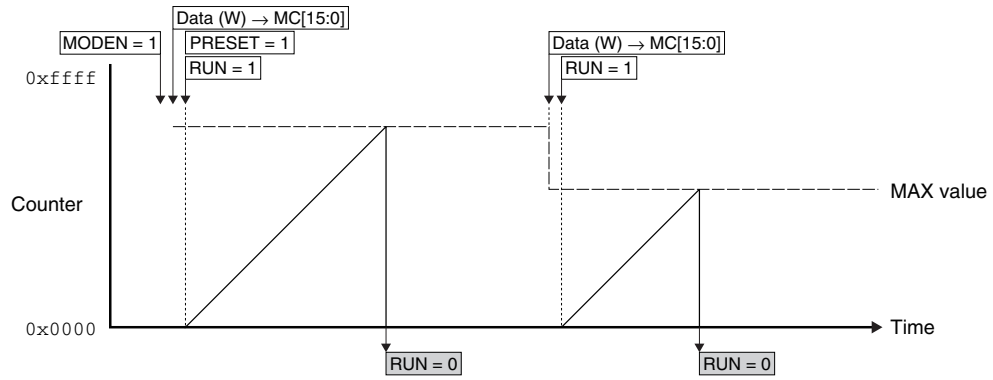
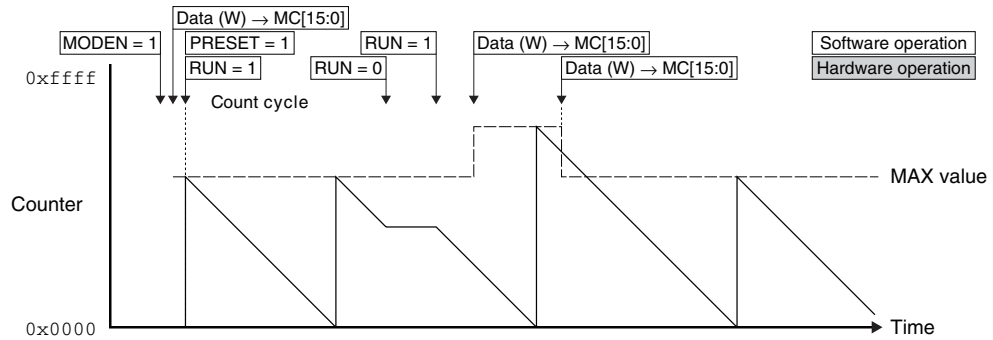


Figure 15.4.1.1 Operations in Repeat Up Count and One-shot Up Count Modes

Operations in repeat down count and one-shot down count modes

In these modes, the counter operates as a down counter and counts from the MAX value (or current value) to 0x0000.
In repeat down count mode, the counter returns to the MAX value if a counter underflow occurs and continues counting until the T16B*n*CTL.RUN bit is set to 0. If the MAX value is altered during counting, the counter keeps counting down to 0x0000 and continues counting down from the new MAX value after a counter underflow occurs.
In one-shot down count mode, the counter returns to the MAX value if a counter underflow occurs and stops automatically at that point.

(1) Repeat down count mode



(2) One-shot down count mode

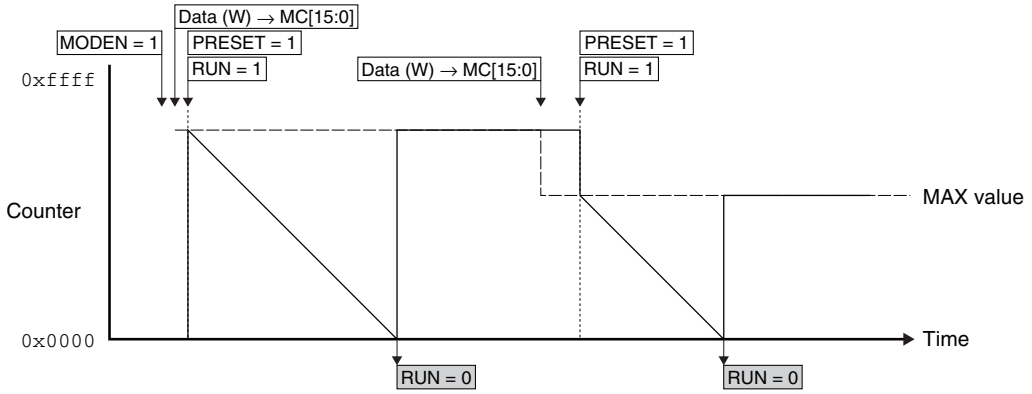


Figure 15.4.1.2 Operations in Repeat Down Count and One-shot Down Count Modes

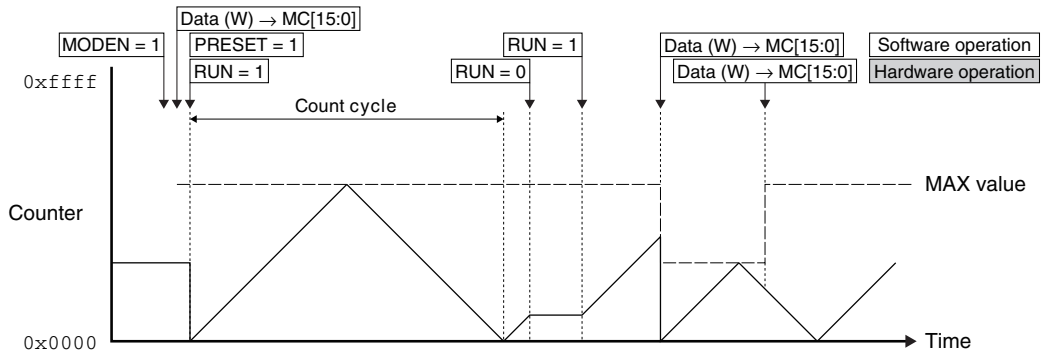
Operations in repeat up/down count and one-shot up/down count modes

In these modes, the counter operates as an up/down counter and counts as 0x0000 (or current value) → the MAX value → 0x0000.

In repeat up/down count mode, the counter repeats counting up from 0x0000 to the MAX value and counting down from the MAX value to 0x0000 until the T16B*n*CTL.RUN bit is set to 0. If the MAX value is altered to a value larger than the current counter value during count up operation, the counter keeps counting up to the new MAX value. If the MAX value is altered to a value smaller than the current counter value, the counter is cleared to 0x0000 and continues counting up to the new MAX value. If the MAX value is altered during count down operation, the counter keeps counting down to 0x0000 and then starts counting up to the new MAX value.

In one-shot up/down count mode, the counter stops automatically when it reaches 0x0000 during count down operation.

(1) Repeat up/down count mode



(2) One-shot up/down count mode

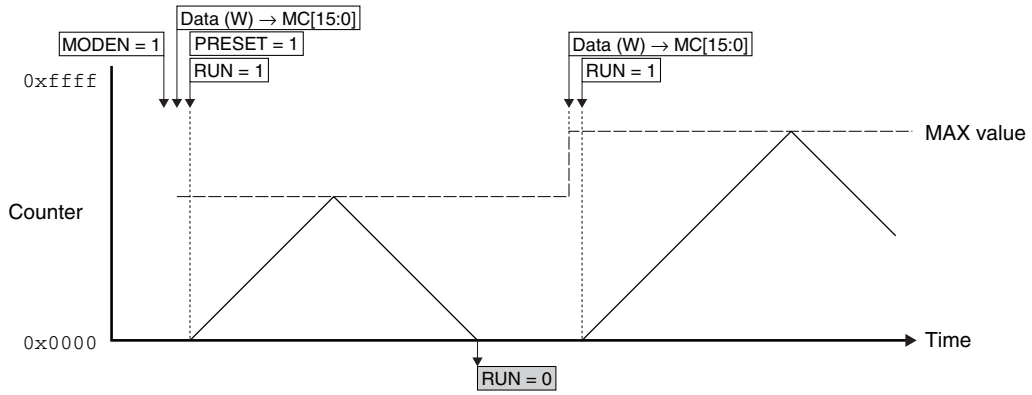


Figure 15.4.1.3 Operations in Repeat Up/Down Count and One-shot Up/Down Count Modes

15.4.2 Comparator/Capture Block Operations

The comparator/capture block functions as a comparator to compare the counter value with the register value set or a capture circuit to capture counter values using the external/software trigger signals.

Comparator/capture block operating mode

The comparator/capture block includes two systems (four or six systems) of comparator/capture circuits and each system can be set to comparator mode or capture mode, individually.

Set the T16B*n*CCCTL*m*.CCMD bit to 0 to set the comparator/capture circuit *m* to comparator mode or 1 to set it to capture mode.

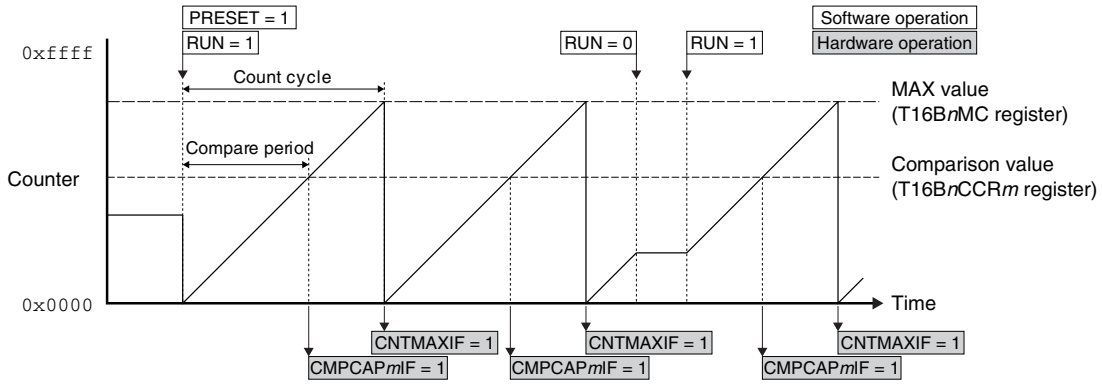
Operations in comparator mode

The comparator mode compares the counter value and the value set via software. It generates an interrupt and toggles the timer output signal level when the values are matched. The T16B*n*CCR*m* register functions as the compare data register used for setting a comparison value in this mode. The TOUT*nm*/CAP*nm* pin is configured to the TOUT*nm* pin.

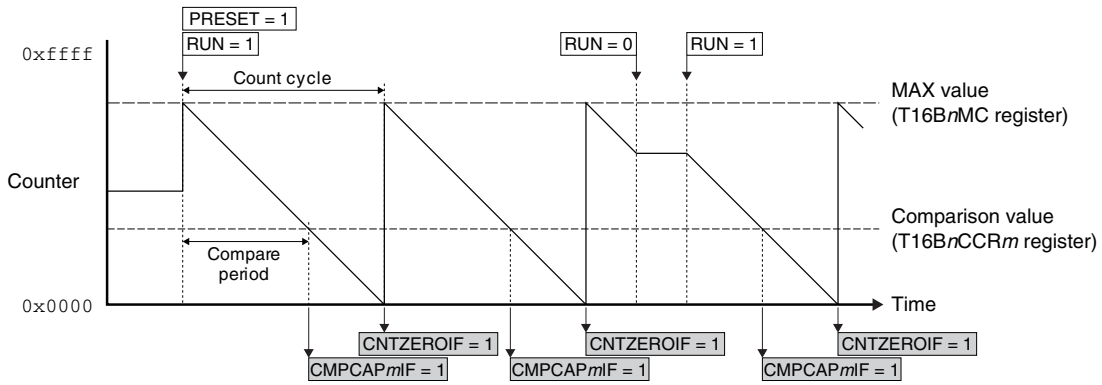
When the counter reaches the value set in the T16B*n*CCR*m* register during counting, the comparator asserts the MATCH signal and sets the T16B*n*INTF.COMPCAP*m*IF bit (compare interrupt flag) to 1.

When the counter reaches the MAX value in comparator mode, the T16B*n*INTF.CNTMAXIF bit (counter MAX interrupt flag) is set to 1. When the counter reaches 0x0000, the T16B*n*INTF.CNTZEROIF bit (counter zero interrupt flag) is set to 1.

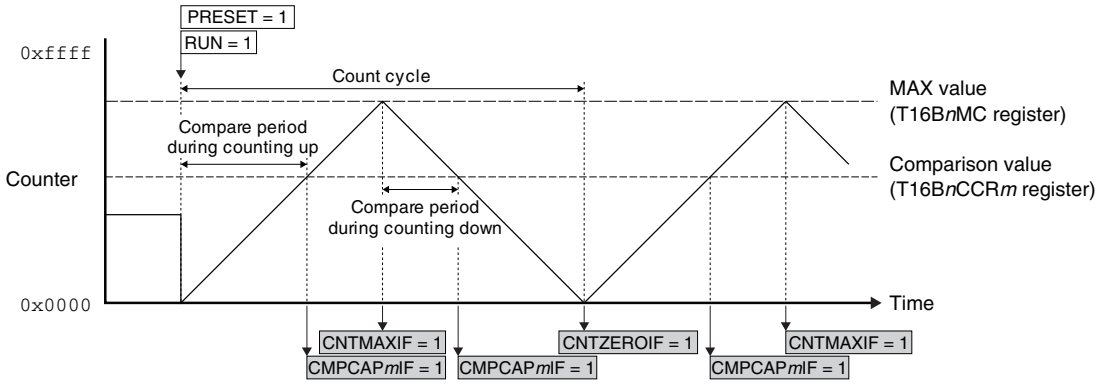
(1) Repeat up count mode



(2) Repeat down count mode



(3) Repeat up/down count mode



(Note that the T16B*n*INTF.CMPCAP*m*IF/CNTMAXIF/CNTZEROIF bit clearing operations via software are omitted from the figure.)

Figure 15.4.2.1 Operation Examples in Comparator Mode

The time from counter = 0x0000 or MAX value to occurrence of a compare interrupt (compare period) and the time to occurrence of a counter MAX or counter zero interrupt (count cycle) can be calculated as follows:

During counting up

$$\text{Compare period} = \frac{(CC + 1)}{f_{CLK\_T16B\_DMM}} \text{ [s]} \qquad \text{Count cycle} = \frac{(MAX + 1)}{f_{CLK\_T16B\_DMM}} \text{ [s]} \qquad \text{(Eq. 15.1)}$$

During counting down

$$\text{Compare period} = \frac{(MAX - CC + 1)}{f_{CLK\_T16B\_DMM}} \text{ [s]} \qquad \text{Count cycle} = \frac{(MAX + 1)}{f_{CLK\_T16B\_DMM}} \text{ [s]} \qquad \text{(Eq. 15.2)}$$

Where

CC: T16B*n*CCR*m* register setting value (0 to 65,535)

MAX: T16B*n*MC register setting value (0 to 65,535)

$f_{CLK\_T16B\_DMM}$: Count clock frequency [Hz]

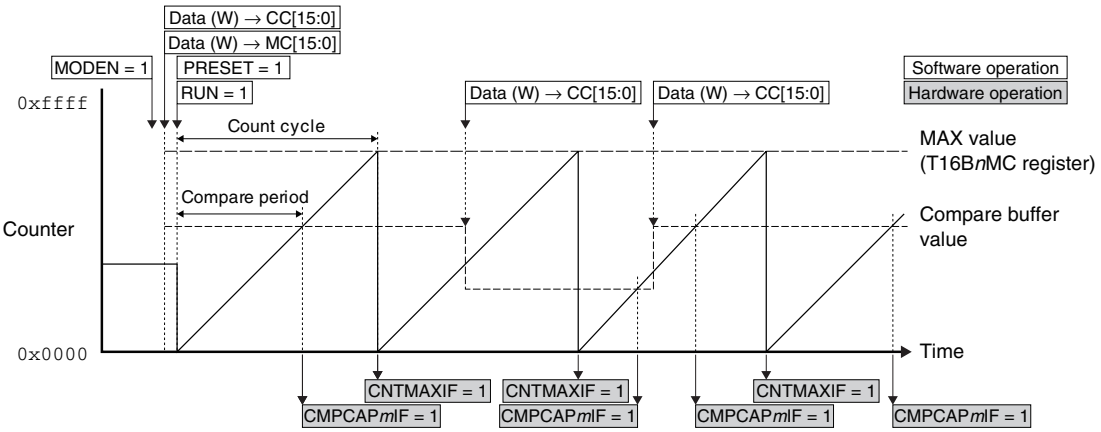
The comparator MATCH signal and counter MAX/ZERO signals are also used to generate a timer output waveform (TOUT). Refer to "TOUT Output Control" for more information.

Compare buffer

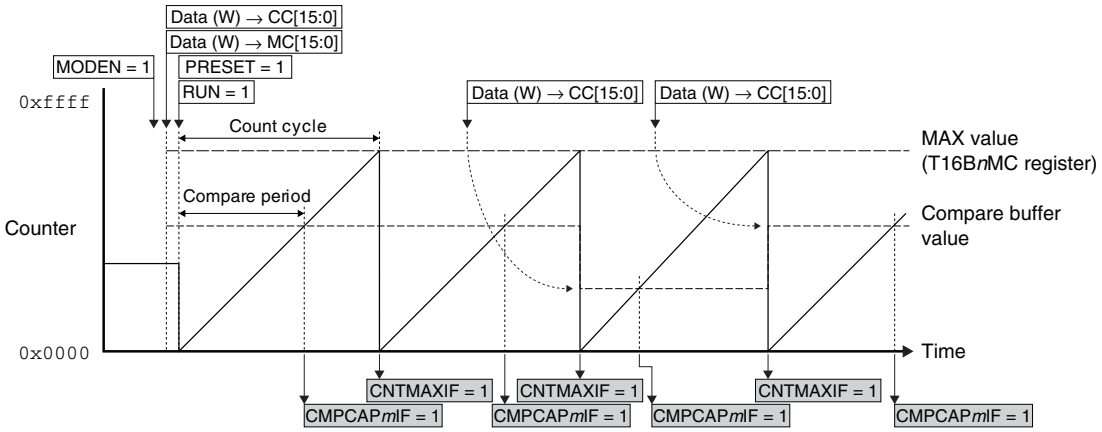
The comparator loads the comparison value, which has been written to the T16B*n*CCR*m* register, to the compare buffer before comparing it with the counter value. For example, when generating a PWM waveform, the waveform with the desired duty ratio may not be generated if the comparison value is altered asynchronous to the count operation. To avoid this problem, the timing to load the comparison value to the compare buffer can be configured using the T16B*n*CCCTL*m*.CBUFMD[2:0] bits for synchronization with the count operation.

(1) Repeat up count mode

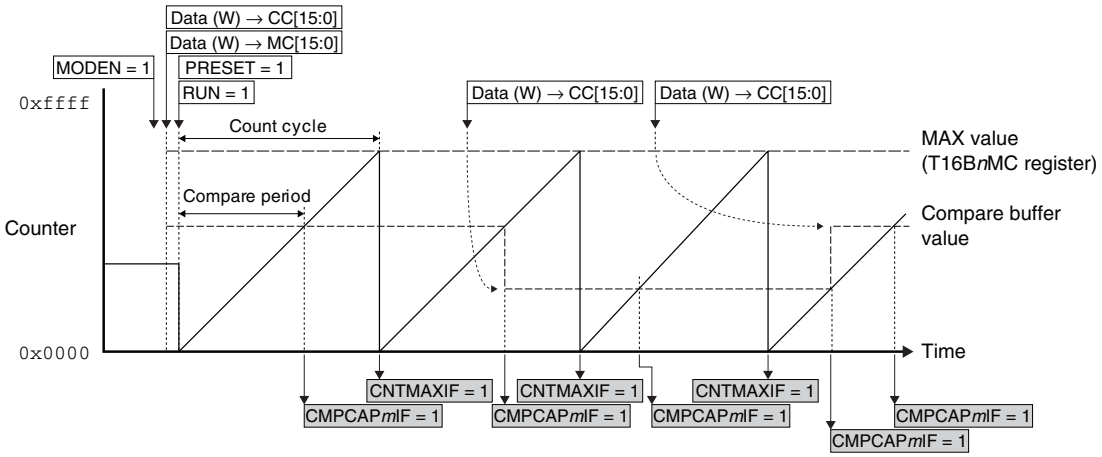
(1.1) T16B*n*CCCTL*m*.CBUFMD[2:0] bits = 0x0



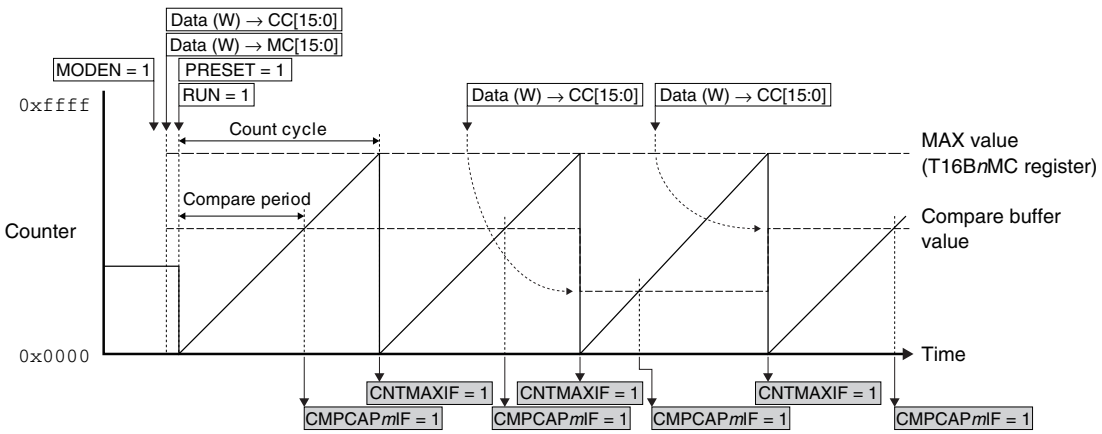
(1.2) T16B*n*CCCTL*m*.CBUFMD[2:0] bits = 0x1



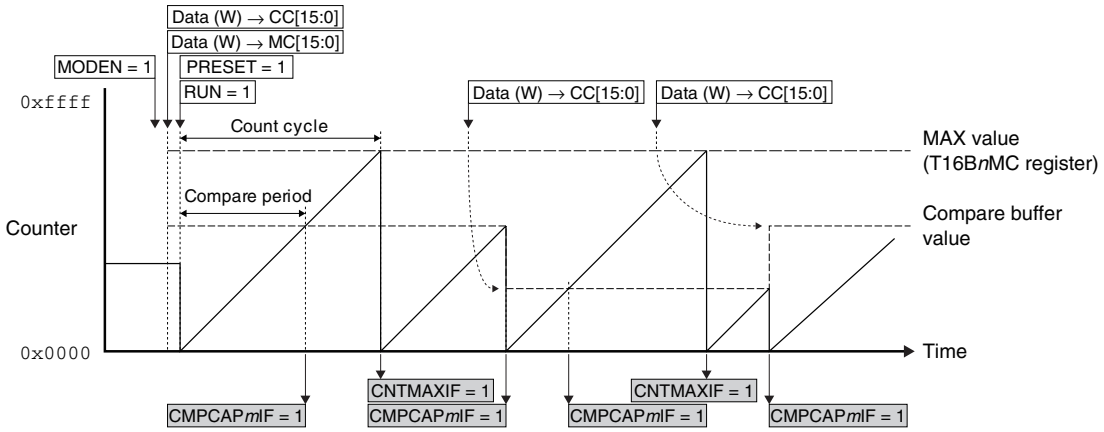
(1.3) T16B*n*CCCTL*m*.CBUFMD[2:0] bits = 0x2



(1.4) T16B*n*CCCTL*m*.CBUFMD[2:0] bits = 0x3

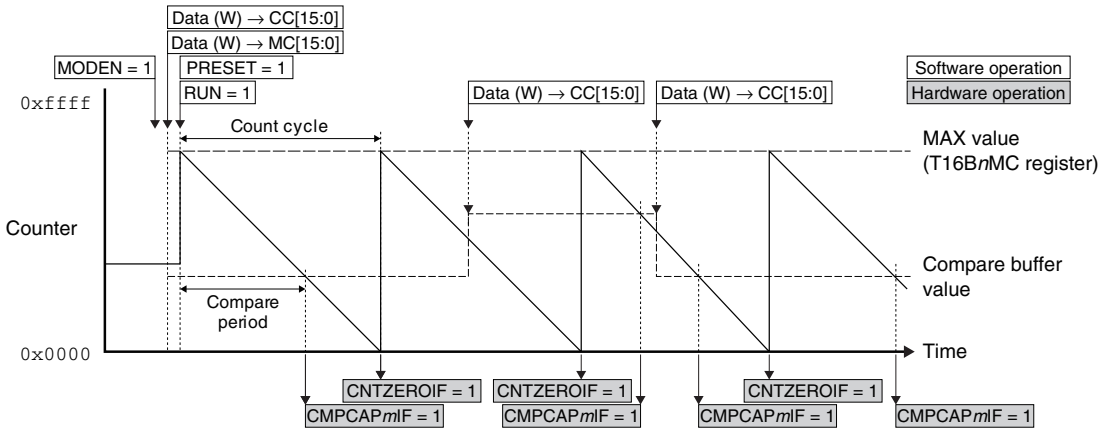


(1.5) T16B*n*CCCTL*m*.CBUFMD[2:0] bits = 0x4

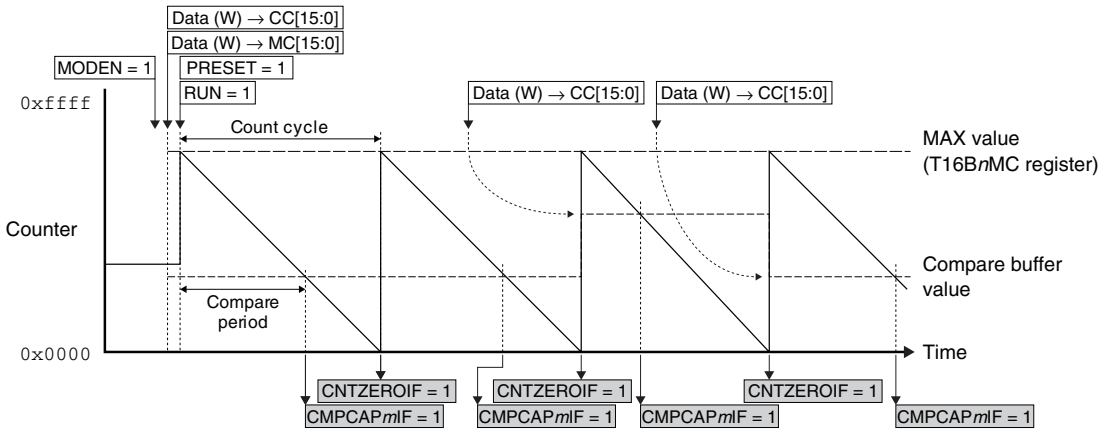


(2) Repeat down count mode

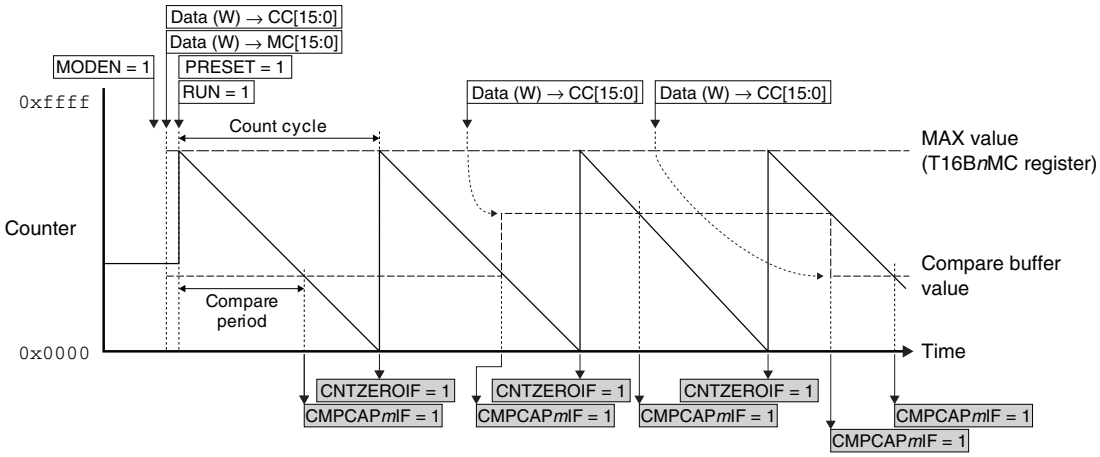
(2.1) T16B*n*CCCTL*m*.CBUFMD[2:0] bits = 0x0



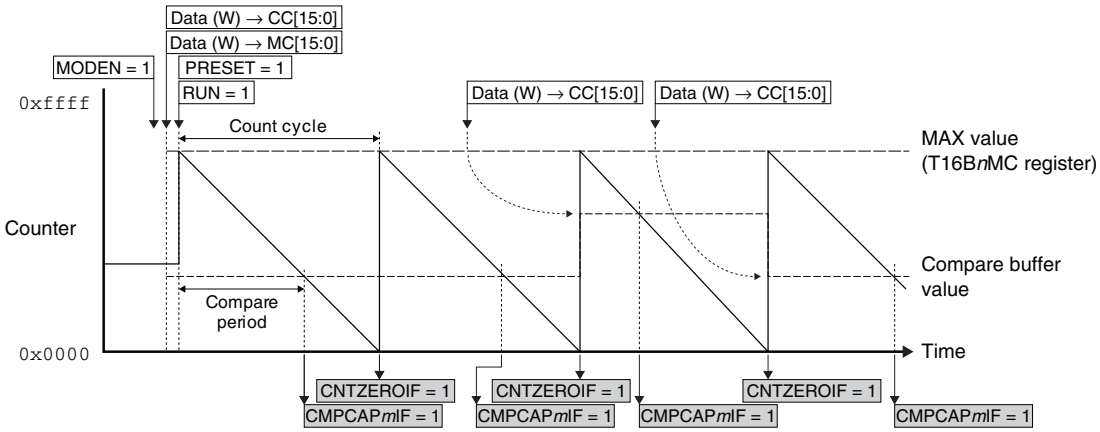
(2.2) T16B*n*CCCTL*m*.CBUFMD[2:0] bits = 0x1



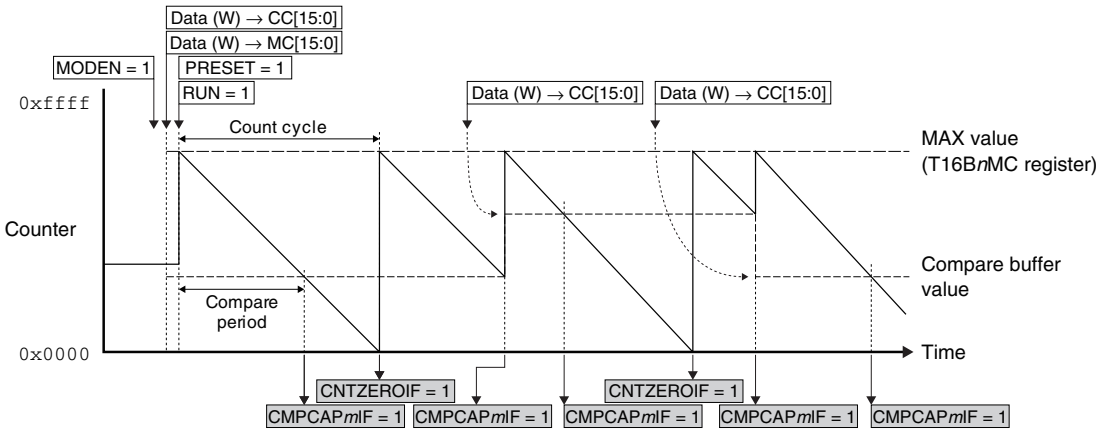
(2.3) T16B*n*CCCTL*m*.CBUFMD[2:0] bits = 0x2



(2.4) T16B*n*CCCTL*m*.CBUFMD[2:0] bits = 0x3

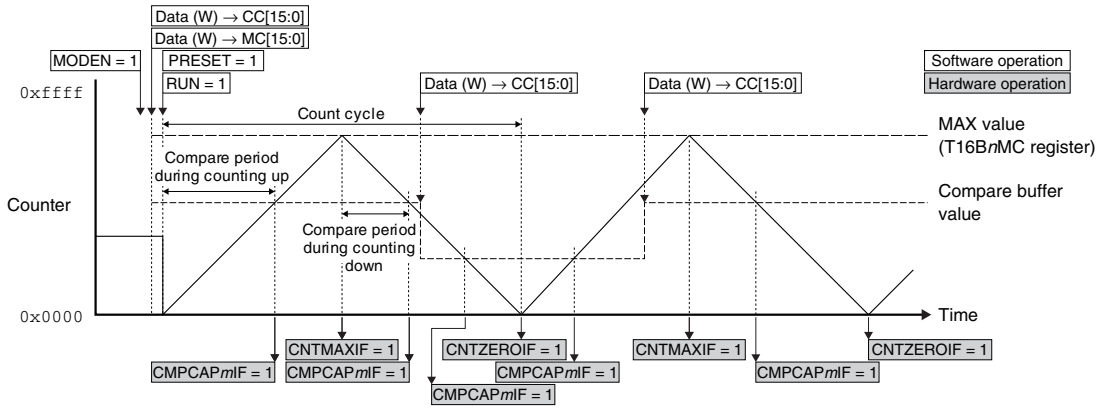


(2.5) T16B*n*CCCTL*m*.CBUFMD[2:0] bits = 0x4

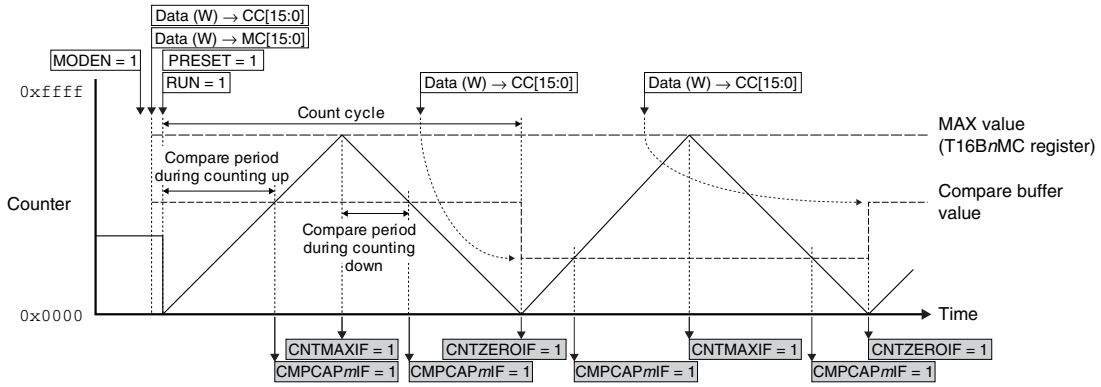


(3) Repeat up/down count mode

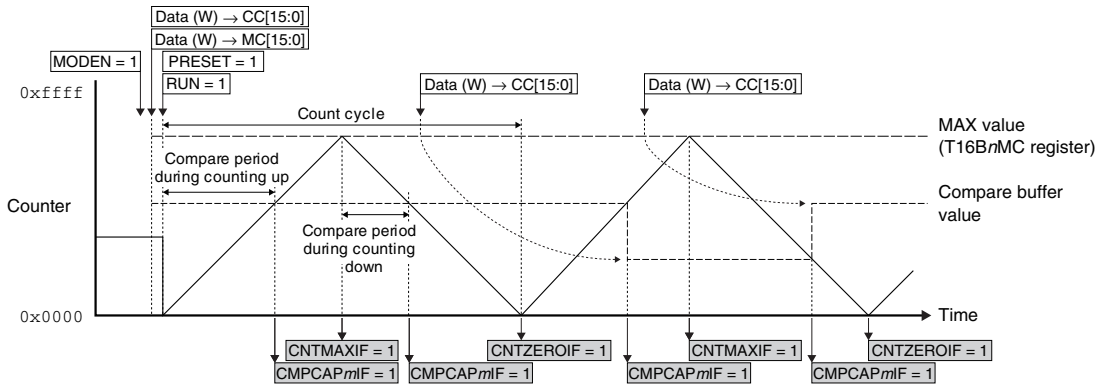
(3.1) T16B*n*CCCTL*m*.CBUFMD[2:0] bits = 0x0

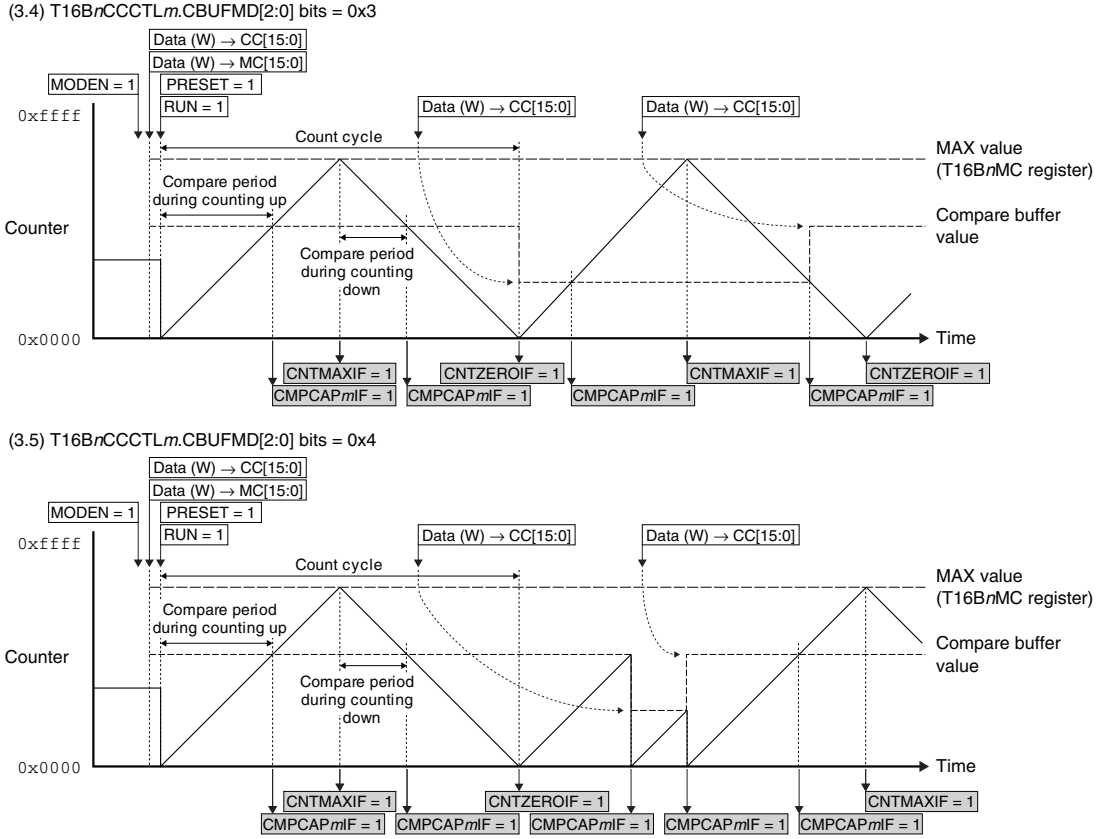


(3.2) T16B*n*CCCTL*m*.CBUFMD[2:0] bits = 0x1



(3.3) T16B*n*CCCTL*m*.CBUFMD[2:0] bits = 0x2





(Note that the T16B*n*INTF.CMPCAP*m*IF/CNTMAXIF/CNTZEROIF bit clearing operations via software are omitted from the figure.)

Figure 15.4.2.2 Compare Buffer Operations

Operations in capture mode

The capture mode captures the counter value when an external event, such as a key entry, occurs (at the specified edge of the external input/software trigger signal). In this mode, the T16B*n*CCR*m* register functions as the capture register from which the captured data is read. Furthermore, the TOUT*nm*/CAP*nm* pin is configured to the CAP*nm* pin.

The trigger signal and the trigger edge to capture the counter value are selected using the T16B*n*CCCTL*m*.CAPIS[1:0] bits and the T16B*n*CCCTL*m*.CAPTRG[1:0] bits, respectively.

When a specified trigger edge is input during counting, the current counter value is loaded to the T16B*n*CCR*m* register. At the same time the T16B*n*INTF.CMPCAP*m*IF bit is set. The interrupt occurred by this bit can be used to read the captured data from the T16B*n*CCR*m* register. For example, external event cycles and pulse widths can be measured from the difference between two captured counter values read.

If the captured data stored in the T16B*n*CCR*m* register is overwritten by the next trigger when the T16B*n*INTF.CMPCAP*m*IF bit is still set, an overwrite error occurs (the T16B*n*INTF.CAPOW*m*IF bit is set).

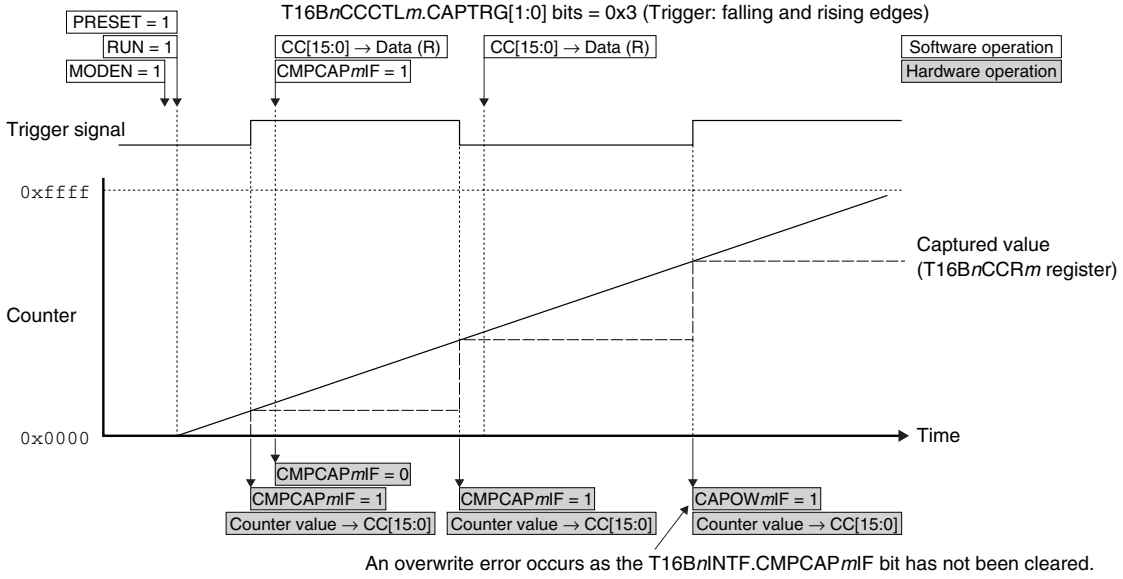


Figure 15.4.2.3 Operations in Capture Mode (Example in One-shot Up Count Mode)

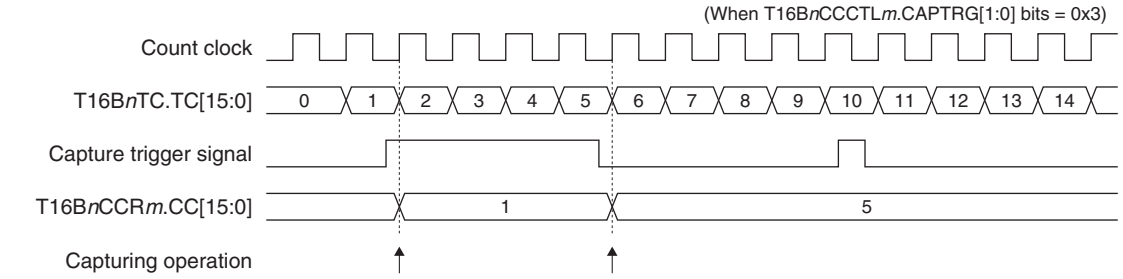
Synchronous capture mode/asynchronous capture mode

The capture circuit can operate in two operating modes: synchronous capture mode and asynchronous capture mode.

Synchronous capture mode is provided to avoid the possibility of invalid data reading by capturing counter data simultaneously with the counter being counted up/down. Set the T16B*n*CCCTL*m*.SCS bit to 1 to set the capture circuit to synchronous capture mode. This mode captures counter data by synchronizing the capture signal with the counter clock.

On the other hand, asynchronous capture mode can capture counter data by detecting a trigger pulse even if the pulse is shorter than the counter clock cycle that becomes invalid in synchronous capture mode. Set the T16B*n*CCCTL*m*.SCS bit to 0 to set the capture circuit to asynchronous capture mode.

(1) Synchronous capture mode



(2) Asynchronous capture mode

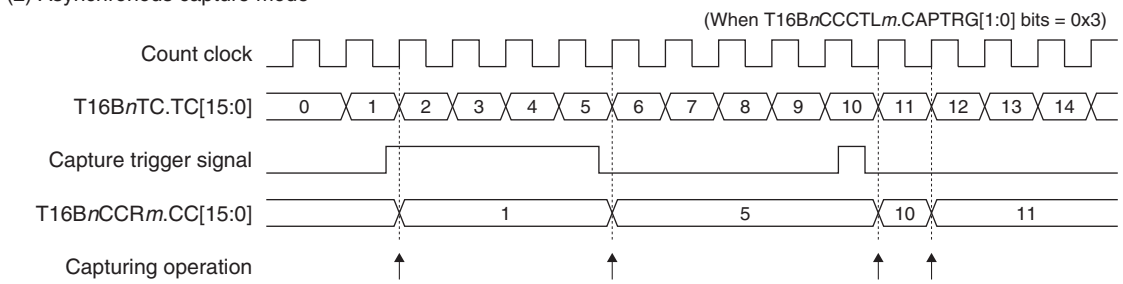


Figure 15.4.2.4 Synchronous Capture Mode/Asynchronous Capture Mode

15.4.3 TOUT Output Control

Comparator mode can generate TOUT signals using the comparator MATCH and counter MAX/ZERO signals. The generated signals can be output to outside the IC. Figure 15.4.3.1 shows the TOUT output circuits (circuits 0 and 1).

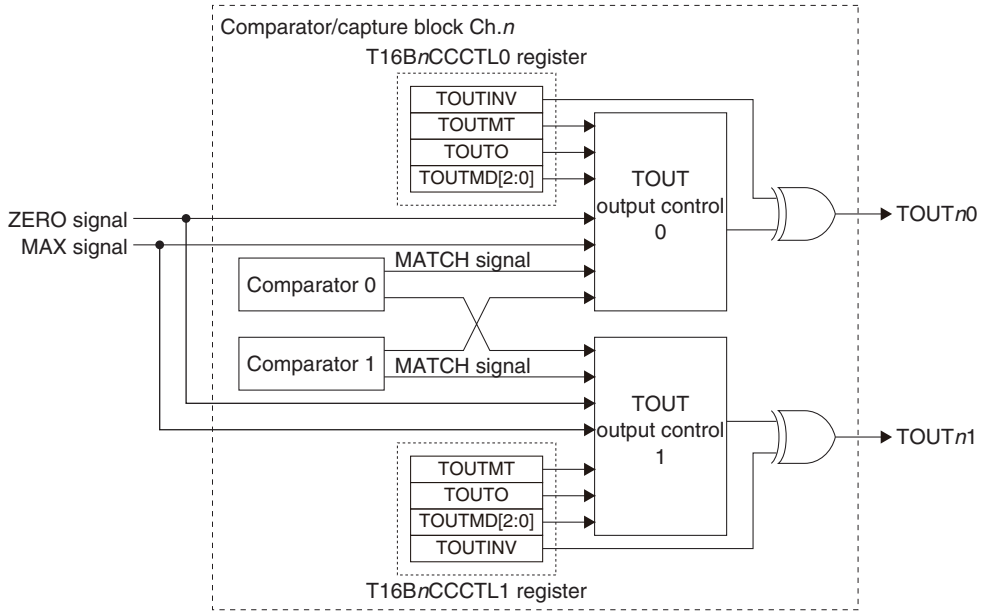


Figure 15.4.3.1 TOUT Output Circuits (Circuits 0 and 1)

Each timer channel includes two (four, or six) TOUT output circuits and their signal generation and output can be controlled individually.

TOUT generation mode

The T16B*n*CCCTL*m*.TOUTMD[2:0] bits are used to set how the TOUT signal waveform is changed by the MATCH and MAX/ZERO signals.

Furthermore, when the T16B*n*CCCTL*m*.TOUTMT bit is set to 1, the TOUT circuit uses the MATCH signal output from another system in the circuit pair (0 and 1, 2 and 3, 4 and 5). This makes it possible to change the signal twice within a counter cycle.

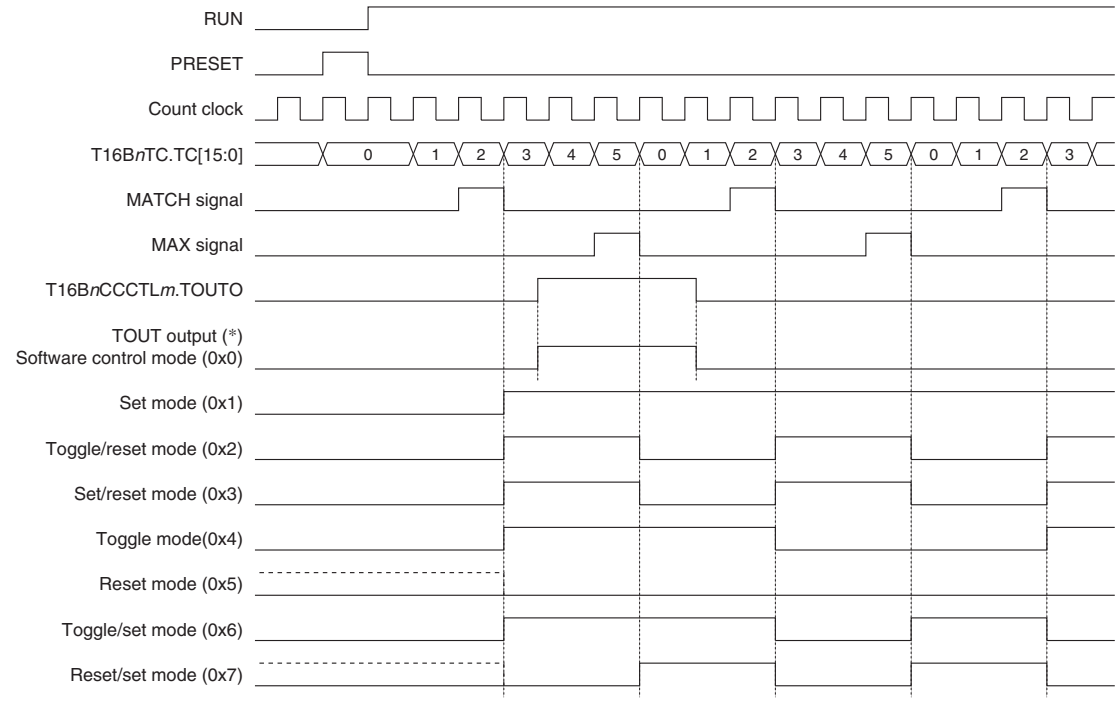
TOUT signal polarity

The TOUT signal polarity (active level) can be set using the T16B*n*CCCTL*m*.TOUTINV bit. It is set to active high by setting the T16B*n*CCCTL*m*.TOUTINV bit to 0 and active low by setting to 1.

Figures 15.4.3.2 and 15.4.3.3 show the TOUT output waveforms.

(1) Repeat up count mode

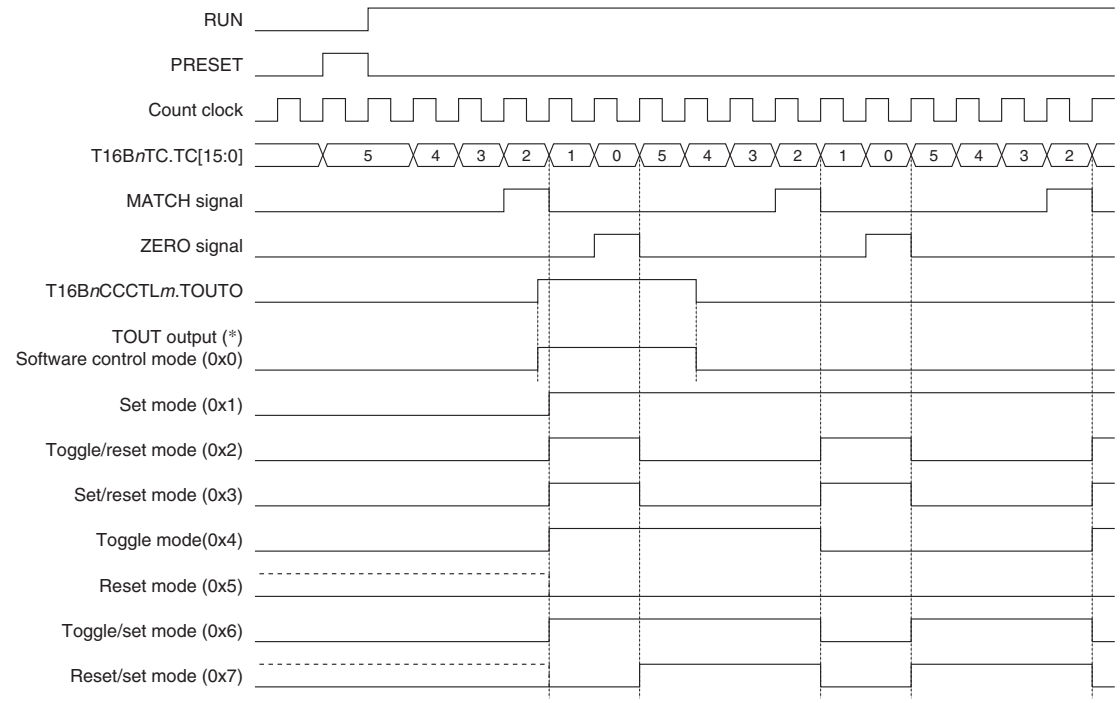
(MAX value = 5, Compare buffer value = 2, T16B*n*CCCTL*m*.TOUTINV bit = 0)



∗ ( ) indicates the T16B*n*CCCTL*m*.TOUTMD[2:0] bit-setting value.

(2) Repeat down count mode

(MAX value = 5, Compare buffer value = 2, T16B*n*CCCTL*m*.TOUTINV bit = 0)



∗ ( ) indicates the T16B*n*CCCTL*m*.TOUTMD[2:0] bit-setting value.

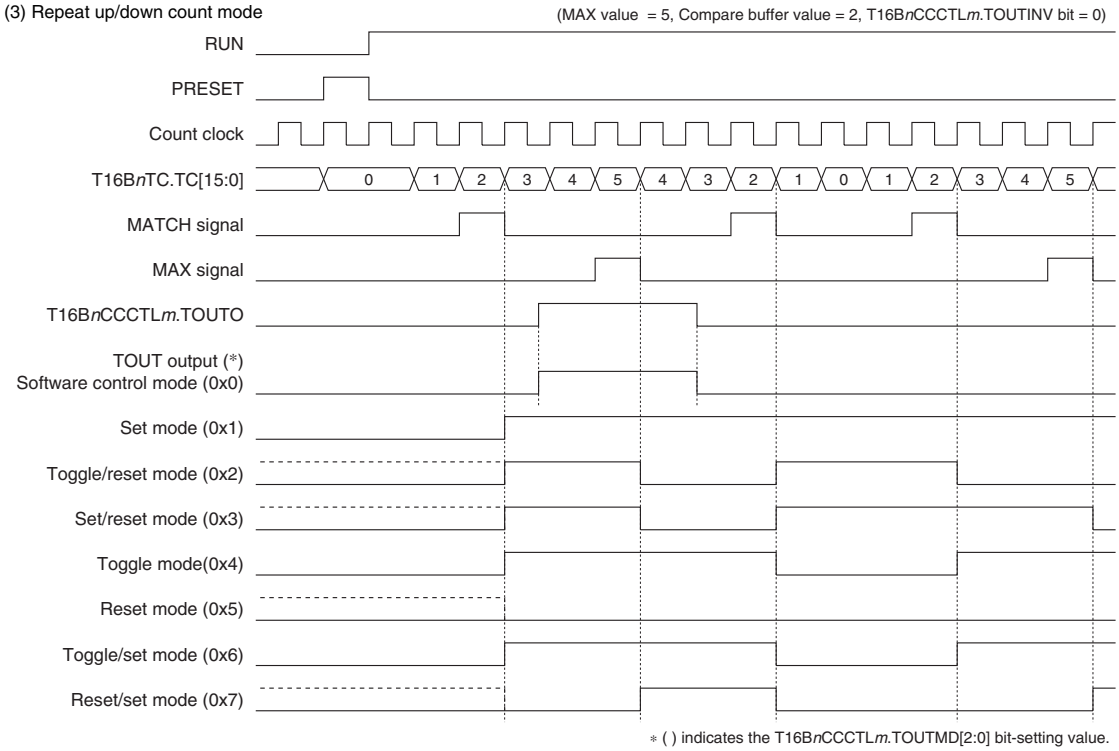
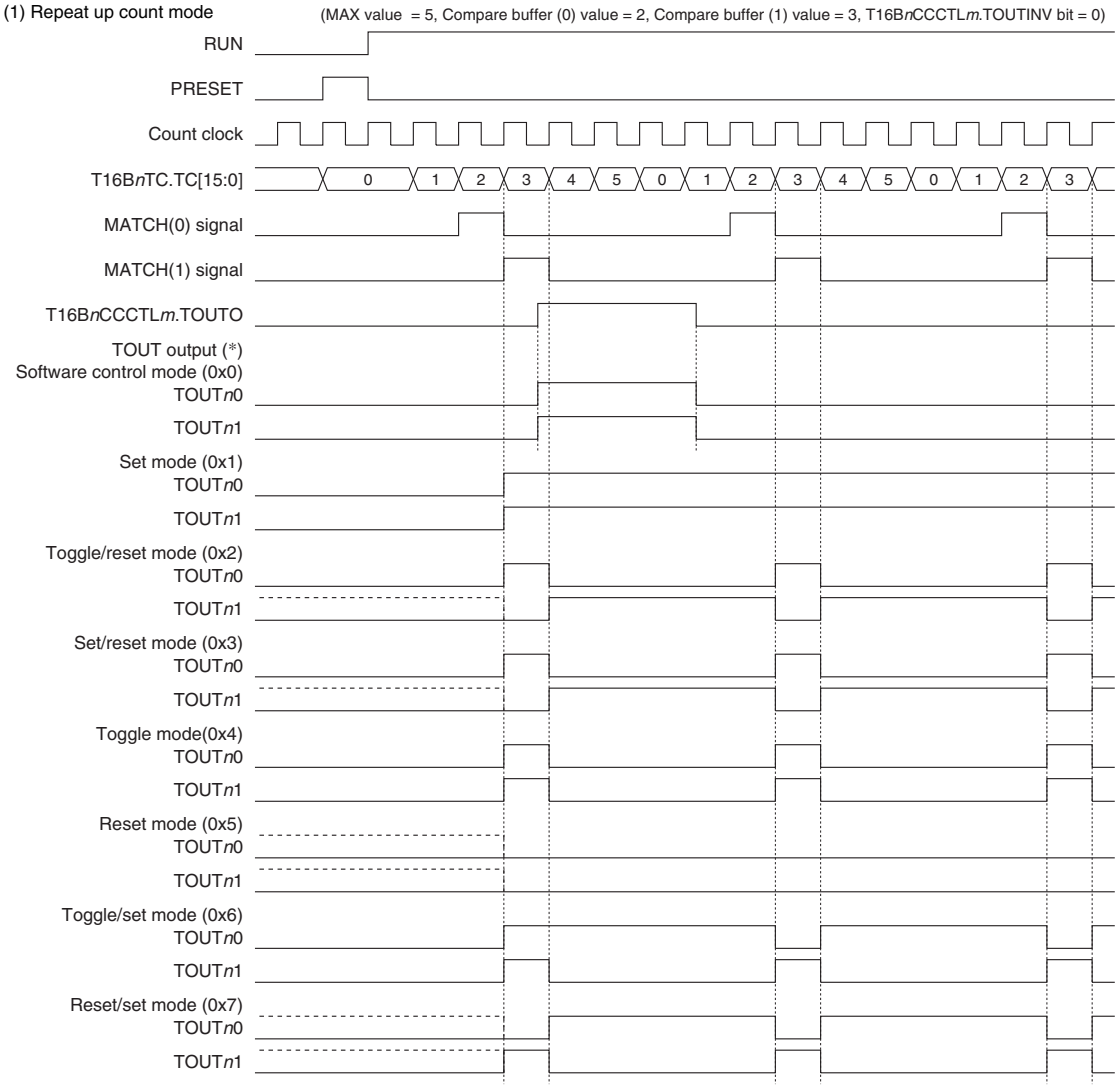


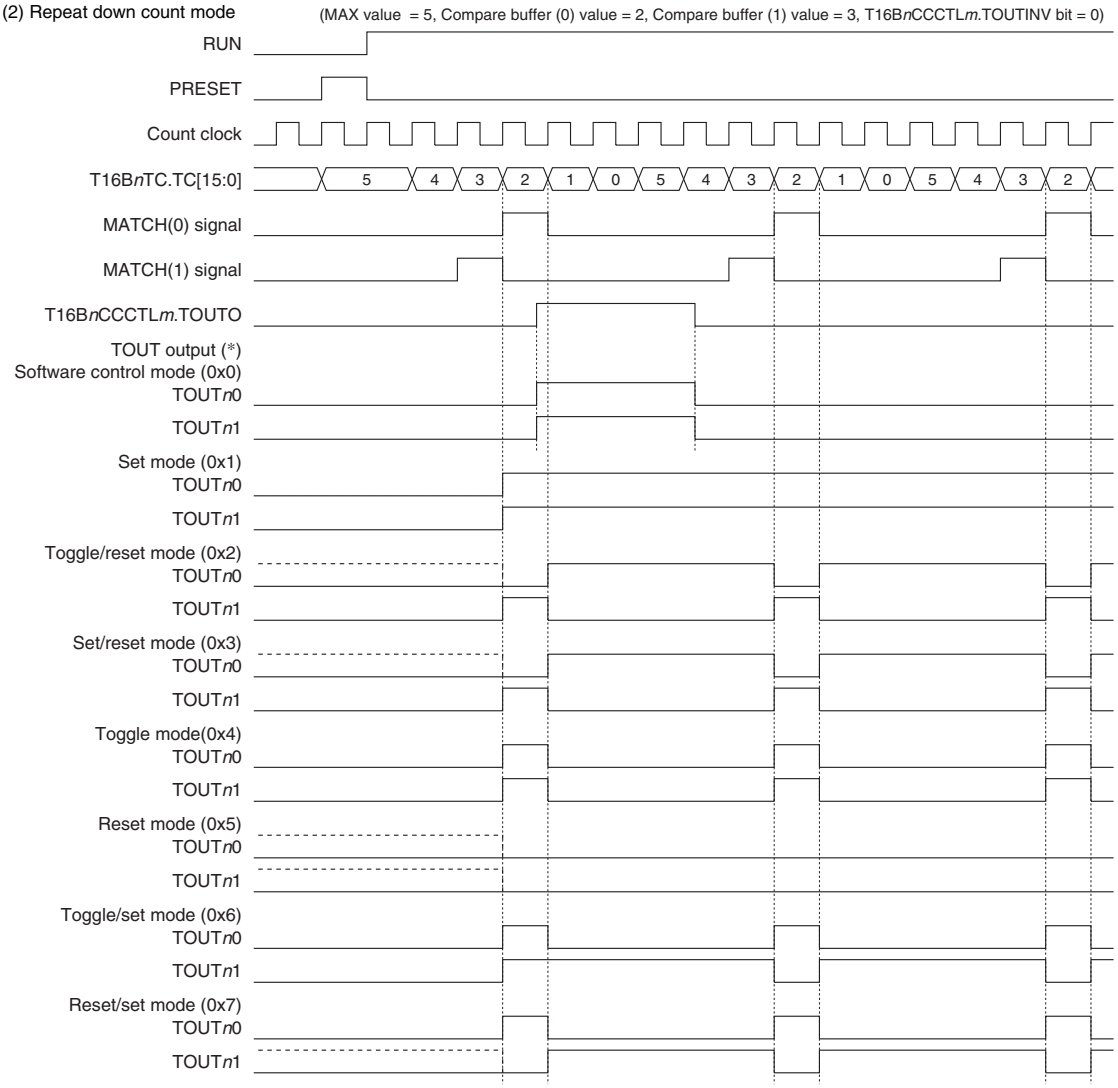
Figure 15.4.3.2 TOUT Output Waveform (T16B*n*CCCTL*m*.TOUTMT bit = 0)

(1) Repeat up count mode



∗ ( ) indicates the T16B*n*CCCTL*m*.TOUTMD[2:0] bit-setting value.

(2) Repeat down count mode



∗ ( ) indicates the T16B*n*CCCTL*m*.TOUTMD[2:0] bit-setting value.

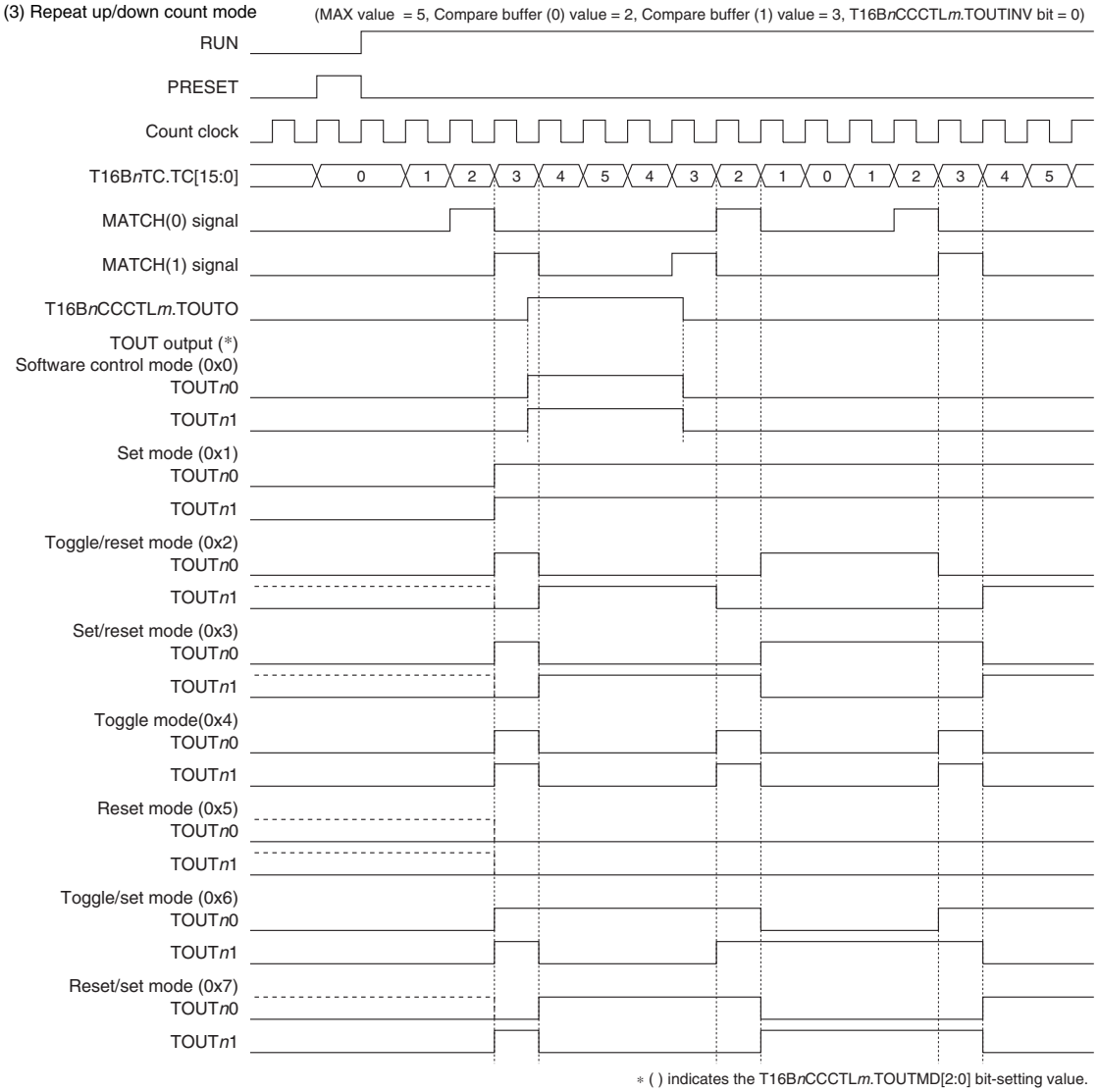


Figure 15.4.3.3 TOUT Output Waveform (T16B*n*CCCTL0.TOUTMT bit = 1, T16B*n*CCCTL1.TOUTMT bit = 0)

15.5 Interrupt

Each T16B_DMM channel has a function to generate the interrupt shown in Table 15.5.1.

Table 15.5.1 T16B_DMM Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Capture overwrite	T16B*n*INTF.CAPOW*m*IF	When the T16B*n*INTF.CMPCAP*m*IF bit =1 and the T16B*n*CCR*m* register is overwritten with new captured data in capture mode	Writing 1
Compare/ capture	T16B*n*INTF.CMPCAP*m*IF	When the counter value becomes equal to the compare buffer value in comparator mode When the counter value is loaded to the T16B*n*CCR*m* register by a capture trigger input in capture mode	Writing 1
Counter MAX	T16B*n*INTF.CNTMAXIF	When the counter reaches the MAX value	Writing 1
Counter zero	T16B*n*INTF.CNTZEROIF	When the counter reaches 0x0000	Writing 1

T16B_DMM provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the "Interrupt Controller" chapter.

15.6 Control Registers

T16B_DMM Ch.*n* Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B*n*CLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/W	
	7–4	CLKDIV[3:0]	0x0	H0	R/W	
	3	–	0	–	R	
	2–0	CLKSRC[2:0]	0x0	H0	R/W	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the T16B_DMM Ch.*n* operating clock is supplied in DEBUG mode or not.

1 (R/W): Clock supplied in DEBUG mode

0 (R/W): No clock supplied in DEBUG mode

Bits 7–4 CLKDIV[3:0]

These bits select the division ratio of the T16B_DMM Ch.*n* operating clock (counter clock).

Bit 3 Reserved

Bits 2–0 CLKSRC[2:0]

These bits select the clock source of T16B_DMM Ch.*n*.

Table 15.6.1 Clock Source and Division Ratio Settings

T16B*n*CLK. CLKDIV[3:0] bits	T16B*n*CLK.CLKSRC[2:0] bits							
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
	IOSC	OSC1	OSC3	EXOSC	EXCL*n*0	EXCL*n*1	EXCL*n*0 inverted input	EXCL*n*1 inverted input
0xf	1/32,768	1/1	1/32,768	1/1	1/1	1/1	1/1	1/1
0xe	1/16,384		1/16,384					
0xd	1/8,192		1/8,192					
0xc	1/4,096		1/4,096					
0xb	1/2,048		1/2,048					
0xa	1/1,024		1/1,024					
0x9	1/512		1/512					
0x8	1/256	1/256	1/256					
0x7	1/128	1/128	1/128					
0x6	1/64	1/64	1/64					
0x5	1/32	1/32	1/32					
0x4	1/16	1/16	1/16					
0x3	1/8	1/8	1/8					
0x2	1/4	1/4	1/4					
0x1	1/2	1/2	1/2					
0x0	1/1	1/1	1/1					

(Note) The oscillator circuits/external inputs that are not supported in this IC cannot be selected as the clock source.

T16B_DMM Ch.*n* Counter Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B*n*CTL	15–9	–	0x00	–	R	–
	8	MAXBSY	0	H0	R	
	7–6	–	0x0	–	R	
	5–4	CNTMD[1:0]	0x0	H0	R/W	
	3	ONEST	0	H0	R/W	
	2	RUN	0	H0	R/W	
	1	PRESET	0	H0	R/W	
	0	MODEN	0	H0	R/W	

Bits 15–9 Reserved

Bit 8 MAXBSY

This bit indicates whether data can be written to the T16B*n*MC register or not.

1 (R): Busy status (cannot be written)

0 (R): Idle (can be written)

While this bit is 1, the T16B*n*MC register is loading the MAX value. Data writing is prohibited during this period.

Bits 7–6 Reserved

Bits 5–4 CNTMD[1:0]

These bits select the counter up/down mode. The count mode is configured with this selection and the T16B*n*CTL.ONEST bit setting (see Table 15.6.2).

Bit 3 ONEST

This bit selects the counter repeat/one-shot mode. The count mode is configured with this selection and the T16B*n*CTL.CNTMD[1:0] bit settings (see Table 15.6.2).

Table 15.6.2 Count Mode

T16B*n*CTL.CNTMD[1:0] bits	Count mode	
	T16B*n*CTL.ONEST bit = 1	T16B*n*CTL.ONEST bit = 0
0x3	Reserved	
0x2	One-shot up/down count mode	Repeat up/down count mode
0x1	One-shot down count mode	Repeat down count mode
0x0	One-shot up count mode	Repeat up count mode

Bit 2 RUN

This bit starts/stops counting.

1 (W): Start counting

0 (W): Stop counting

1 (R): Counting

0 (R): Idle

By writing 1 to this bit, the counter block starts count operations. However, the T16B*n*CTL.MODEN bit must be set to 1 in conjunction with this bit or it must be set in advance. While the timer is running, writing 0 to the T16B*n*CTL.RUN bit stops count operations. When the counter stops by the counter MAX/ZERO signal in one-shot mode, this bit is automatically cleared to 0.

Bit 1 PRESET

This bit resets the counter.

1 (W): Reset

0 (W): Ineffective

1 (R): Resetting in progress

0 (R): Resetting finished or normal operation

In up mode or up/down mode, the counter is cleared to 0x0000 by writing 1 to this bit. In down mode, the MAX value, which has been set to the T16B*n*MC register, is preset to the counter. However, the T16B*n*CTL.MODEN bit must be set to 1 in conjunction with this bit or it must be set in advance.

Bit 0 MODEN

This bit enables the T16B_DMM Ch.*n* operations.

1 (R/W): Enable (Start supplying operating clock)

0 (R/W): Disable (Stop supplying operating clock)

Note: The counter reset operation using the T16B*n*CTL.PRESET bit and the counting start operation using the T16B*n*CTL.RUN bit take effect only when the T16B*n*CTL.MODEN bit = 1.

T16B_DMM Ch.*n* Max Counter Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B*n*MC	15–0	MC[15:0]	0xffff	H0	R/W	–

Bits 15–0 MC[15:0]

These bits are used to set the MAX value to preset to the counter. For more information, refer to "Counter Block Operations - MAX counter data register."

Notes: • When one-shot mode is selected, do not alter the T16B*n*MC.MC[15:0] bits (MAX value) during counting.

• Make sure the T16B*n*CTL.MODEN bit is set to 1 before writing data to the T16B*n*MC.MC[15:0] bits. If the T16B*n*CTL.MODEN bit = 0 when writing to the T16B*n*MC.MC[15:0] bits, set the T16B*n*CTL.MODEN bit to 1 until the T16B*n*CS.BSY bit is set to 0 from 1.

• Do not set the T16B*n*MC.MC[15:0] bits to 0x0000.

T16B_DMM Ch.*n* Timer Counter Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B*n*TC	15–0	TC[15:0]	0x0000	H0	R	–

Bits 15–0 TC[15:0]

The current counter value can be read out through these bits.

T16B_DMM Ch.*n* Counter Status Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B*n*CS	15–8	–	0x00	–	R	–
	7	CAPI5	0	H0	R	
	6	CAPI4	0	H0	R	
	5	CAPI3	0	H0	R	
	4	CAPI2	0	H0	R	
	3	CAPI1	0	H0	R	
	2	CAPI0	0	H0	R	
	1	UP_DOWN	1	H0	R	
	0	BSY	0	H0	R	

Bits 15–8 Reserved

Bit 7 CAPI5
Bit 6 CAPI4
Bit 5 CAPI3
Bit 4 CAPI2
Bit 3 CAPI1
Bit 2 CAPI0

These bits indicate the signal level currently input to the CAP*nm* pin.
1 (R): Input signal = High level
0 (R): Input signal = Low level

The following shows the correspondence between the bit and the CAP*nm* pin:
T16B*n*CS.CAPI5 bit: CAP*n*5 pin
T16B*n*CS.CAPI4 bit: CAP*n*4 pin
T16B*n*CS.CAPI3 bit: CAP*n*3 pin
T16B*n*CS.CAPI2 bit: CAP*n*2 pin
T16B*n*CS.CAPI1 bit: CAP*n*1 pin
T16B*n*CS.CAPI0 bit: CAP*n*0 pin

Note: The configuration of the T16B*n*CS.CAPI*m* bits depends on the model. The bits corresponding to the CAP*nm* pins that do not exist are read-only bits and are always fixed at 0.

Bit 1 UP_DOWN

This bit indicates the currently set count direction.

1 (R): Count up

0 (R): Count down

Bit 0 BSY

This bit indicates the counter operating status.

1 (R): Running

0 (R): Idle

T16B_DMM Ch.*n* Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B*n*INTF	15–14	–	0x0	–	R	–
	13	CAPOW5IF	0	H0	R/W	Cleared by writing 1.
	12	CMPCAP5IF	0	H0	R/W	
	11	CAPOW4IF	0	H0	R/W	
	10	CMPCAP4IF	0	H0	R/W	
	9	CAPOW3IF	0	H0	R/W	
	8	CMPCAP3IF	0	H0	R/W	
	7	CAPOW2IF	0	H0	R/W	
	6	CMPCAP2IF	0	H0	R/W	
	5	CAPOW1IF	0	H0	R/W	
	4	CMPCAP1IF	0	H0	R/W	
	3	CAPOW0IF	0	H0	R/W	
	2	CMPCAP0IF	0	H0	R/W	
	1	CNTMAXIF	0	H0	R/W	
	0	CNTZEROIF	0	H0	R/W	

Bits 15–14 Reserved

Bit 13 CAPOW5IF
Bit 12 CMPCAP5IF
Bit 11 CAPOW4IF
Bit 10 CMPCAP4IF
Bit 9 CAPOW3IF
Bit 8 CMPCAP3IF
Bit 7 CAPOW2IF
Bit 6 CMPCAP2IF
Bit 5 CAPOW1IF
Bit 4 CMPCAP1IF
Bit 3 CAPOW0IF
Bit 2 CMPCAP0IF
Bit 1 CNTMAXIF
Bit 0 CNTZEROIF

These bits indicate the T16B_DMM Ch.*n* interrupt cause occurrence status.

1 (R): Cause of interrupt occurred

0 (R): No cause of interrupt occurred

1 (W): Clear flag

0 (W): Ineffective

The following shows the correspondence between the bit and interrupt:
T16B*n*INTF.CAPOW5IF bit: Capture 5 overwrite interrupt
T16B*n*INTF.CMPCAP5IF bit: Compare/capture 5 interrupt
T16B*n*INTF.CAPOW4IF bit: Capture 4 overwrite interrupt
T16B*n*INTF.CMPCAP4IF bit: Compare/capture 4 interrupt
T16B*n*INTF.CAPOW3IF bit: Capture 3 overwrite interrupt
T16B*n*INTF.CMPCAP3IF bit: Compare/capture 3 interrupt
T16B*n*INTF.CAPOW2IF bit: Capture 2 overwrite interrupt
T16B*n*INTF.CMPCAP2IF bit: Compare/capture 2 interrupt
T16B*n*INTF.CAPOW1IF bit: Capture 1 overwrite interrupt
T16B*n*INTF.CMPCAP1IF bit: Compare/capture 1 interrupt
T16B*n*INTF.CAPOW0IF bit: Capture 0 overwrite interrupt
T16B*n*INTF.CMPCAP0IF bit: Compare/capture 0 interrupt
T16B*n*INTF.CNTMAXIF bit: Counter MAX interrupt
T16B*n*INTF.CNTZEROIF bit: Counter zero interrupt

Note: The configuration of the T16B*n*INTF.CAPOW*m*IF and T16B*n*INTF.CMPCAP*m*IF bits depends on the model. The bits corresponding to the comparator/capture circuits that do not exist are read-only bits and are always fixed at 0.

T16B_DMM Ch.*n* Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B*n*INTE	15–14	–	0x0	–	R	–
	13	CAPOW5IE	0	H0	R/W	
	12	CMPCAP5IE	0	H0	R/W	
	11	CAPOW4IE	0	H0	R/W	
	10	CMPCAP4IE	0	H0	R/W	
	9	CAPOW3IE	0	H0	R/W	
	8	CMPCAP3IE	0	H0	R/W	
	7	CAPOW2IE	0	H0	R/W	
	6	CMPCAP2IE	0	H0	R/W	
	5	CAPOW1IE	0	H0	R/W	
	4	CMPCAP1IE	0	H0	R/W	
	3	CAPOW0IE	0	H0	R/W	
	2	CMPCAP0IE	0	H0	R/W	
	1	CNTMAXIE	0	H0	R/W	
	0	CNTZEROIE	0	H0	R/W	

Bits 15–14 Reserved

Bit 13 CAPOW5IE
Bit 12 CMPCAP5IE
Bit 11 CAPOW4IE
Bit 10 CMPCAP4IE
Bit 9 CAPOW3IE
Bit 8 CMPCAP3IE
Bit 7 CAPOW2IE
Bit 6 CMPCAP2IE
Bit 5 CAPOW1IE
Bit 4 CMPCAP1IE
Bit 3 CAPOW0IE
Bit 2 CMPCAP0IE
Bit 1 CNTMAXIE
Bit 0 CNTZEROIE

These bits enable T16B_DMM Ch.*n* interrupts.
1 (R/W): Enable interrupts
0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:
T16B*n*INTE.CAPOW5IE bit: Capture 5 overwrite interrupt
T16B*n*INTE.CMPCAP5IE bit: Compare/capture 5 interrupt
T16B*n*INTE.CAPOW4IE bit: Capture 4 overwrite interrupt
T16B*n*INTE.CMPCAP4IE bit: Compare/capture 4 interrupt
T16B*n*INTE.CAPOW3IE bit: Capture 3 overwrite interrupt
T16B*n*INTE.CMPCAP3IE bit: Compare/capture 3 interrupt
T16B*n*INTE.CAPOW2IE bit: Capture 2 overwrite interrupt
T16B*n*INTE.CMPCAP2IE bit: Compare/capture 2 interrupt
T16B*n*INTE.CAPOW1IE bit: Capture 1 overwrite interrupt
T16B*n*INTE.CMPCAP1IE bit: Compare/capture 1 interrupt
T16B*n*INTE.CAPOW0IE bit: Capture 0 overwrite interrupt
T16B*n*INTE.CMPCAP0IE bit: Compare/capture 0 interrupt
T16B*n*INTE.CNTMAXIE bit: Counter MAX interrupt
T16B*n*INTE.CNTZEROIE bit: Counter zero interrupt

Notes: • The configuration of the T16B*n*INTE.CAPOW*m*IE and T16B*n*INTE.CMPCAP*m*IE bits depends on the model. The bits corresponding to the comparator/capture circuits that do not exist are read-only bits and are always fixed at 0.

• To prevent generating unnecessary interrupts, the corresponding interrupt flag should be cleared before enabling interrupts.

T16B_DMM Ch.*n* Comparator/Capture *m* Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B*n*CCCTL*m*	15	SCS	0	H0	R/W	–
	14–12	CBUFMD[2:0]	0x0	H0	R/W	
	11–10	CAPIS[1:0]	0x0	H0	R/W	
	9–8	CAPTRG[1:0]	0x0	H0	R/W	
	7	–	0	–	R	
	6	TOUTMT	0	H0	R/W	
	5	TOUTO	0	H0	R/W	
	4–2	TOUTMD[2:0]	0x0	H0	R/W	
	1	TOUTINV	0	H0	R/W	
	0	CCMD	0	H0	R/W	

Bit 15 SCS

This bit selects either synchronous capture mode or asynchronous capture mode.
1 (R/W): Synchronous capture mode
0 (R/W): Asynchronous capture mode

For more information, refer to "Comparator/Capture Block Operations - Synchronous capture mode/ asynchronous capture mode." The T16B*n*CCCTL*m*.SCS bit is control bit for capture mode and is ineffective in comparator mode.

Bits 14–12 CBUFMD[2:0]

These bits select the timing to load the comparison value written in the T16B*n*CCR*m* register to the compare buffer. The T16B*n*CCCTL*m*.CBUFMD[2:0] bits are control bits for comparator mode and are ineffective in capture mode.

Table 15.6.3 Timings to Load Comparison Value to Compare Buffer

T16B*n*CCCTL*m*. CBUFMD[2:0] bits	Count mode	Comparison Value load timing
0x7–0x5	Reserved	
0x4	Up mode	When the counter becomes equal to the comparison value set previously Also the counter is reset to 0x0000 simultaneously.
	Down mode	When the counter becomes equal to the comparison value set previously Also the counter is reset to the MAX value simultaneously.
	Up/down mode	When the counter becomes equal to the comparison value set previously Also the counter is reset to 0x0000 simultaneously.
0x3	Up mode	When the counter reverts to 0x0000
	Down mode	When the counter reverts to the MAX value
	Up/down mode	When the counter becomes equal to the comparison value set previously or when the counter reverts to 0x0000
0x2	Up mode	When the counter becomes equal to the comparison value set previously
	Down mode	
	Up/down mode	
0x1	Up mode	When the counter reaches the MAX value
	Down mode	When the counter reaches 0x0000
	Up/down mode	When the counter reaches 0x0000 or the MAX value
0x0	Up mode	At the CLK_T16B_DMM*n* rising edge after writing to the T16B*n*CCR*m* register
	Down mode	
	Up/down mode	

Bits 11–10 CAPIS[1:0]

These bits select the trigger signal for capturing (see Table 15.6.4). The T16B*n*CCCTL*m*.CAPIS[1:0] bits are control bits for capture mode and are ineffective in comparator mode.

Bits 9–8 CAPTRG[1:0]

These bits select the trigger edge(s) of the trigger signal at which the counter value is captured in the T16B*n*CCR*m* register in capture mode (see Table 15.6.4). The T16B*n*CCCTL*m*.CAPTRG[1:0] bits are control bits for capture mode and are ineffective in comparator mode.

Table 15.6.4 Trigger Signal/Edge for Capturing Counter Value

T16B*n*CCCTL*m*. CAPTRG[1:0] bits (Trigger edge)	Trigger condition		
	T16B*n*CCCTL*m*.CAPIS[1:0] bits (Trigger signal)		
	0x0 (External trigger signal)	0x2 (Software trigger signal = L)	0x3 (Software trigger signal = H)
0x3 (↑ & ↓)	Rising or falling edge of the CAP*nm* pin input signal	Altering the T16B*n*CCCTL*m*.CAPIS[1:0] bits from 0x2 to 0x3, or from 0x3 to 0x2	
0x2 (↓)	Falling edge of the CAP*nm* pin input signal	Altering the T16B*n*CCCTL*m*.CAPIS[1:0] bits from 0x3 to 0x2	
0x1 (↑)	Rising edge of the CAP*nm* pin input signal	Altering the T16B*n*CCCTL*m*.CAPIS[1:0] bits from 0x2 to 0x3	
0x0	Not triggered (disable capture function)		

Bit 7 Reserved

Bit 6 TOUTMT

This bit selects whether the comparator MATCH signal of another system is used for generating the TOUT*nm* signal or not.

1 (R/W): Generate TOUT using two comparator MATCH signals of the comparator circuit pair (0 and 1, 2 and 3, 4 and 5)

0 (R/W): Generate TOUT using one comparator MATCH signal of comparator *m* and the counter MAX or ZERO signals

The T16B*n*CCCTL*m*.TOUTMT bit is control bit for comparator mode and is ineffective in capture mode.

Bit 5 TOUTO

This bit sets the TOUT*nm* signal output level when software control mode (T16B*n*CCCTL*m*.TOUTMD[2:0] = 0x0) is selected for the TOUT*nm* output.

1 (R/W): High level output

0 (R/W): Low level output

The T16B*n*CCCTL*m*.TOUTO bit is control bit for comparator mode and is ineffective in capture mode.

Bits 4–2 TOUTMD[2:0]

These bits configure how the TOUT*nm* signal waveform is changed by the comparator MATCH and counter MAX/ZERO signals.

The T16B*n*CCCTL*m*.TOUTMD[2:0] bits are control bits for comparator mode and are ineffective in capture mode.

Table 15.6.5 TOUT Generation Mode

T16B*n*CCCTL*m*. TOUTMD[2:0] bits	TOUT generation mode and operations			
	T16B*n*CCCTL*m*. TOUTMT bit	Count mode	Output signal	Change in the signal
0x7	**Reset/set mode**			
	0	Up count mode Up/down count mode	TOUT*nm*	The signal becomes inactive by the MATCH signal and it becomes active by the MAX signal.
		Down count mode	TOUT*nm*	The signal becomes inactive by the MATCH signal and it becomes active by the ZERO signal.
	1	All count modes	TOUT*nm*	The signal becomes inactive by the MATCH*m* signal and it becomes active by the MATCH*m*+1 signal.
			TOUT*nm*+1	The signal becomes inactive by the MATCH*m*+1 signal and it becomes active by the MATCH*m* signal.
0x6	**Toggle/set mode**			
	0	Up count mode Up/down count mode	TOUT*nm*	The signal is inverted by the MATCH signal and it becomes active by the MAX signal.
		Down count mode	TOUT*nm*	The signal is inverted by the MATCH signal and it becomes active by the ZERO signal.
	1	All count modes	TOUT*nm*	The signal is inverted by the MATCH*m* signal and it becomes active by the MATCH*m*+1 signal.
			TOUT*nm*+1	The signal is inverted by the MATCH*m*+1 signal and it becomes active by the MATCH*m* signal.
0x5	**Reset mode**			
	0	All count modes	TOUT*nm*	The signal becomes inactive by the MATCH signal.
	1	All count modes	TOUT*nm*	The signal becomes inactive by the MATCH*m* or MATCH*m*+1 signal.
			TOUT*nm*+1	The signal becomes inactive by the MATCH*m*+1 or MATCH*m* signal.
0x4	**Toggle mode**			
	0	All count modes	TOUT*nm*	The signal is inverted by the MATCH signal.
	1	All count modes	TOUT*nm*	The signal is inverted by the MATCH*m* or MATCH*m*+1 signal.
			TOUT*nm*+1	The signal is inverted by the MATCH*m*+1 or MATCH*m* signal.
0x3	**Set/reset mode**			
	0	Up count mode Up/down count mode	TOUT*nm*	The signal becomes active by the MATCH signal and it becomes inactive by the MAX signal.
		Down count mode	TOUT*nm*	The signal becomes active by the MATCH signal and it becomes inactive by the ZERO signal.
	1	All count modes	TOUT*nm*	The signal becomes active by the MATCH*m* signal and it becomes inactive by the MATCH*m*+1 signal.
			TOUT*nm*+1	The signal becomes active by the MATCH*m*+1 signal and it becomes inactive by the MATCH*m* signal.
0x2	**Toggle/reset mode**			
	0	Up count mode Up/down count mode	TOUT*nm*	The signal is inverted by the MATCH signal and it becomes inactive by the MAX signal.
		Down count mode	TOUT*nm*	The signal is inverted by the MATCH signal and it becomes inactive by the ZERO signal.
	1	All count modes	TOUT*nm*	The signal is inverted by the MATCH*m* signal and it becomes inactive by the MATCH*m*+1 signal.
			TOUT*nm*+1	The signal is inverted by the MATCH*m*+1 signal and it becomes inactive by the MATCH*m* signal.
0x1	**Set mode**			
	0	All count modes	TOUT*nm*	The signal becomes active by the MATCH signal.
	1	All count modes	TOUT*nm*	The signal becomes active by the MATCH*m* or MATCH*m*+1 signal.
			TOUT*nm*+1	The signal becomes active by the MATCH*m*+1 or MATCH*m* signal.
0x0	**Software control mode**			
	*	All count modes	TOUT*nm*	The signal becomes active by setting the T16B*n*CCCTL*m*. TOUTO bit to 1 and it becomes inactive by setting to 0.

Bit 1 TOUTINV

This bit selects the TOUT*nm* signal polarity.
1 (R/W): Inverted (active low)
0 (R/W): Normal (active high)

The T16B*n*CCCTL*m*.TOUTINV bit is control bit for comparator mode and is ineffective in capture mode.

Bit 0 CCMD

This bit selects the operating mode of the comparator/capture circuit *m*.
1 (R/W): Capture mode (T16B*n*CCR*m* register = capture register)
0 (R/W): Comparator mode (T16B*n*CCR*m* register = compare data register)

T16B_DMM Ch.*n* Compare/Capture *m* Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B*n*CCR*m*	15–0	CC[15:0]	0x0000	H0	R/W	–

Bits 15–0 CC[15:0]

In comparator mode, this register is configured as the compare data register and used to set the comparison value to be compared with the counter value.
In capture mode, this register is configured as the capture register and the counter value captured by the capture trigger signal is loaded.

16 Sound Generator (SNDA_DMM)

16.1 Overview

SNDA_DMM is a sound generator that generates melodies and buzzer signals. The features of the SNDA_DMM are listed below.

- Sound output mode is selectable from three types.
  1. Normal buzzer mode (for normal buzzer output of which the output duration is controlled via software)
     - Output frequency: Can be set within the range of 512 Hz to 16,384 Hz.
     - Duty ratio: Can be set within the range of 0 % to 100 %.
  2. One-shot buzzer mode (for short buzzer output such as a clicking sound)
     - Output frequency: Can be set within the range of 512 Hz to 16,384 Hz.
     - Duty ratio: Can be set within the range of 0 % to 100 %.
     - One-shot output duration: Can be set within the range of 15.6 ms to 250 ms. (16 types)
  3. Melody mode (for playing single note melody)
     - Pitch: Can be set within the range of 128 Hz to 16,384 Hz. (Scale: 3 octave from C3 to C6 with reference to A4 = 443 Hz)
     - Duration: Can be set within the range of half note/rest to thirty-second note/rest. (7 types)
     - Tempo: Can be set within the range of 30 to 480. (16 types)
     - Other: Tie can be specified.
- A piezoelectric buzzer can be driven with the inverted and non-inverted output pins.
- Can control the non-inverted output pin status while sound stops.
- DMM linking mode
  Enables outputting the buzzer signal while the DMM controller detects continuity in continuity check mode.

Figure 16.1.1 shows the SNDA_DMM configuration.

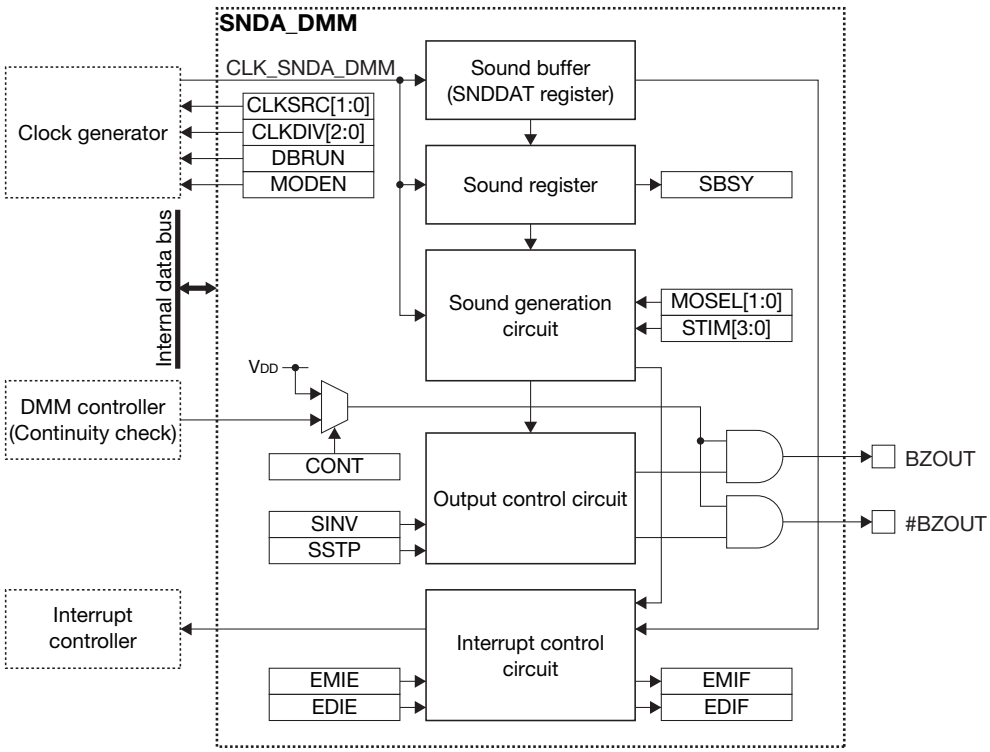


Figure 16.1.1 SNDA_DMM Configuration

16.2 Output Pins and External Connections

16.2.1 List of Output Pins

Table 16.2.1.1 lists the SNDA_DMM pins.

Table 16.2.1.1 List of SNDA_DMM Pins

Pin name	I/O*	Initial status*	Function
BZOUT	O	O (Low)	Non-inverted buzzer output pin
#BZOUT	O	O (Low)	Inverted buzzer output pin

* Indicates the status when the pin is configured for SNDA_DMM

If the port is shared with the SNDA_DMM pin and other functions, the SNDA_DMM output function must be assigned to the port before activating the SNDA_DMM. For more information, refer to the "I/O Ports" chapter.

16.2.2 Output Pin Drive Mode

The drive mode of the BZOUT and #BZOUT pins can be set to one of the two types shown below using the SNDSEL.SINV bit.

Direct drive mode (SNDSEL.SINV bit = 0)

This mode drives both the BZOUT and #BZOUT pins to low while the buzzer signal output is off to prevent the piezoelectric buzzer from applying unnecessary bias.

Normal drive mode (SNDSEL.SINV bit = 1)

In this mode, the #BZOUT pin always outputs the inverted signal of the BZOUT pin even when the buzzer output is off.

16.2.3 External Connections

Figures 16.2.2.1 and 16.2.2.2 show connection diagrams between SNDA_DMM and a piezoelectric buzzer.

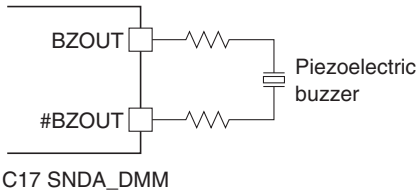


Figure 16.2.2.1 Connection between SNDA_DMM and Piezoelectric Buzzer (Direct Drive)

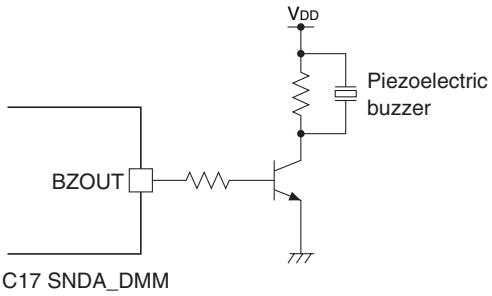


Figure 16.2.2.2 Connection between SNDA_DMM and Piezoelectric Buzzer (Single Pin Drive)

16.3 Clock Settings

16.3.1 SNDA_DMM Operating Clock

When using SNDA_DMM, the SNDA_DMM operating clock CLK_SNDA_DMM must be supplied to SNDA_DMM from the clock generator. The CLK_SNDA_DMM supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to "Clock Generator" in the "Power Supply, Reset, and Clocks" chapter).
2. Set the following SNDCLK register bits:
   - SNDCLK.CLKSRC[1:0] bits (Clock source selection)
   - SNDCLK.CLKDIV[2:0] bits (Clock division ratio selection = Clock frequency setting)

The CLK_SNDA_DMM frequency should be set to around 32,768 Hz.

16.3.2 Clock Supply in SLEEP Mode

When using SNDA_DMM during SLEEP mode, the SNDA_DMM operating clock CLK_SNDA_DMM must be configured so that it will keep supplying by writing 0 to the CLGOSC.*xxxx*SLPC bit for the CLK_SNDA_DMM clock source.
If the CLGOSC.*xxxx*SLPC bit for the CLK_SNDA_DMM clock source is 1, the CLK_SNDA_DMM clock source is deactivated during SLEEP mode and SNDA_DMM stops with the register settings maintained at those before entering SLEEP mode. After the CPU returns to normal mode, CLK_SNDA_DMM is supplied and the SNDA_DMM operation resumes.

16.3.3 Clock Supply in DEBUG Mode

The CLK_SNDA_DMM supply during DEBUG mode should be controlled using the SNDCLK.DBRUN bit.
The CLK_SNDA_DMM supply to SNDA_DMM is suspended when the CPU enters DEBUG mode if the SNDCLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_SNDA_DMM supply resumes. Although SNDA_DMM stops operating when the CLK_SNDA_DMM supply is suspended, the output pin and registers retain the status before DEBUG mode was entered. If the SNDCLK.DBRUN bit = 1, the CLK_SNDA_DMM supply is not suspended and SNDA_DMM will keep operating in DEBUG mode.

16.4 Operations

16.4.1 Initialization

SNDA_DMM should be initialized with the procedure shown below.

1. Assign the SNDA_DMM output function to the ports. (Refer to the "I/O Ports" and "Universal Port Multiplexer" chapters.)
2. Configure the SNDA_DMM operating clock.
3. Set the SNDCTL.MODEN bit to 1. (Enable SNDA_DMM operations)
4. Set the following SNDSEL register bits:
   - SNDSEL.SINV bit (Set output pin drive mode)
   - SNDSEL.CONT bit (Set DMM linking mode)
5. Set the following bits when using the interrupt:
   - Write 1 to the interrupt flags in the SNDINTF register. (Clear interrupt flags)
   - Set the interrupt enable bits in the SNDINTE register to 1. (Enable interrupts)

16.4.2 Buzzer Output in Normal Buzzer Mode

Normal buzzer mode generates a buzzer signal with the software specified frequency and duty ratio, and outputs the generated signal to outside the IC. The buzzer output duration can also be controlled via software.
An output start/stop procedure and the SNDA_DMM operations are shown below.

Normal buzzer output start/stop procedure

1. Set the SNDSEL.MOSEL[1:0] bits to 0x0. (Set normal buzzer mode)
2. Write data to the following sound buffer (SNDDAT register) bits. (Start buzzer output)
   - SNDDAT.SLEN[5:0] bits (Set buzzer output signal duty ratio)
   - SNDDAT.SFRQ[7:0] bits (Set buzzer output signal frequency)
3. Write 1 to the SNDCTL.SSTP bit after the output period has elapsed. (Stop buzzer output)

Normal buzzer output operations

When data is written to the sound buffer (SNDDAT register), SNDA_DMM clears the SNDINTF.EMIF bit (sound buffer empty interrupt flag) to 0 and starts buzzer output operations.
The data written to the sound buffer is loaded into the sound register in sync with the CLK_SNDA_DMM clock. At the same time, the SNDINTF.EMIF bit and SNDINTF.SBSY bit are both set to 1. The output pin outputs the buzzer signal with the frequency/duty ratio specified.
Writing 1 to the SNDCTL.SSTP bit stops buzzer output and sets the SNDINTF.EDIF bit (sound output completion interrupt flag) to 1. The SNDINTF.SBSY bit is cleared to 0.
Figure 16.4.2.1 shows a buzzer output timing chart in normal buzzer mode.

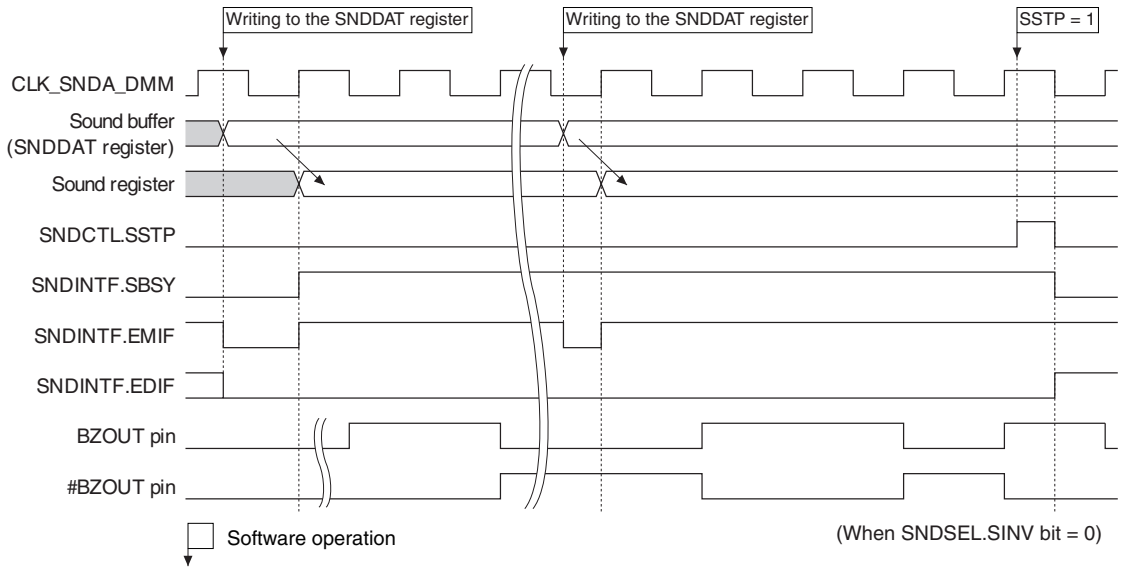


Figure 16.4.2.1 Buzzer Output Timing Chart in Normal Buzzer Mode

Buzzer output waveform configuration (normal buzzer mode/one-shot buzzer mode)

Set the buzzer signal frequency and duty ratio (high period/cycle) using the SNDDAT.SFRQ[7:0] and SNDDAT.SLEN[5:0] bits, respectively. Use the following equations to calculate these setting values.

$$\text{SNDDAT.SFRQ[7:0] bits} = \frac{f_{CLK\_SNDA\_DMM}}{f_{BZOUT}} - 1 \qquad \text{(Eq. 16.1)}$$

$$\text{SNDDAT.SLEN[5:0] bits} = \left(\frac{f_{CLK\_SNDA\_DMM}}{f_{BZOUT}} \times \frac{DUTY}{100}\right) - 1 \qquad \text{(Eq. 16.2)}$$

Where
$f_{CLK\_SNDA\_DMM}$: CLK_SNDA_DMM frequency [Hz]
f_{BZOUT}: Buzzer signal frequency [Hz]
DUTY: Buzzer signal duty ratio [%]

However, the following settings are prohibited:

- Settings as SNDDAT.SFRQ[7:0] bits ≤ SNDDAT.SLEN[5:0] bits
- Settings as SNDDAT.SFRQ[7:0] bits = 0x00

Table 16.4.2.1 Buzzer Frequency Settings (when $f_{CLK\_SNDA\_DMM}$ = 32,768 Hz)

SNDDAT. SFRQ[7:0] bits	Frequency [Hz]	SNDDAT. SFRQ[7:0] bits	Frequency [Hz]	SNDDAT. SFRQ[7:0] bits	Frequency [Hz]	SNDDAT. SFRQ[7:0] bits	Frequency [Hz]
0x3f	512.0	0x2f	682.7	0x1f	1,024.0	0x0f	2,048.0
0x3e	520.1	0x2e	697.2	0x1e	1,057.0	0x0e	2,184.5
0x3d	528.5	0x2d	712.3	0x1d	1,092.3	0x0d	2,340.6
0x3c	537.2	0x2c	728.2	0x1c	1,129.9	0x0c	2,520.6
0x3b	546.1	0x2b	744.7	0x1b	1,170.3	0x0b	2,730.7
0x3a	555.4	0x2a	762.0	0x1a	1,213.6	0x0a	2,978.9
0x39	565.0	0x29	780.2	0x19	1,260.3	0x09	3,276.8
0x38	574.9	0x28	799.2	0x18	1,310.7	0x08	3,640.9
0x37	585.1	0x27	819.2	0x17	1,365.3	0x07	4,096.0
0x36	595.8	0x26	840.2	0x16	1,424.7	0x06	4,681.1
0x35	606.8	0x25	862.3	0x15	1,489.5	0x05	5,461.3
0x34	618.3	0x24	885.6	0x14	1,560.4	0x04	6,553.6
0x33	630.2	0x23	910.2	0x13	1,638.4	0x03	8,192.0
0x32	642.5	0x22	936.2	0x12	1,724.6	0x02	10,922.7
0x31	655.4	0x21	963.8	0x11	1,820.4	0x01	16,384.0
0x30	668.7	0x20	993.0	0x10	1,927.5	0x00	Cannot be set

Table 16.4.2.2 Buzzer Duty Ratio Setting Examples (when $f_{CLK\_SNDA\_DMM}$ = 32,768 Hz)

SNDDAT. SLEN[5:0] bits	Duty ratio by buzzer frequency					
	16,384 Hz	8,192 Hz	4,096 Hz	2,048 Hz	1,024 Hz	512 Hz
0x3f	–	–	–	–	–	–
0x3e	–	–	–	–	–	98.4
0x3d	–	–	–	–	–	96.9
0x3c	–	–	–	–	–	95.3
0x3b	–	–	–	–	–	93.8
0x3a	–	–	–	–	–	92.2
0x39	–	–	–	–	–	90.6
0x38	–	–	–	–	–	89.1
0x37	–	–	–	–	–	87.5
0x36	–	–	–	–	–	85.9
0x35	–	–	–	–	–	84.4
0x34	–	–	–	–	–	82.8
0x33	–	–	–	–	–	81.3
0x32	–	–	–	–	–	79.7
0x31	–	–	–	–	–	78.1
0x30	–	–	–	–	–	76.6
0x2f	–	–	–	–	–	75.0
0x2e	–	–	–	–	–	73.4
0x2d	–	–	–	–	–	71.9
0x2c	–	–	–	–	–	70.3
0x2b	–	–	–	–	–	68.8
0x2a	–	–	–	–	–	67.2
0x29	–	–	–	–	–	65.6
0x28	–	–	–	–	–	64.1
0x27	–	–	–	–	–	62.5
0x26	–	–	–	–	–	60.9
0x25	–	–	–	–	–	59.4
0x24	–	–	–	–	–	57.8
0x23	–	–	–	–	–	56.3
0x22	–	–	–	–	–	54.7
0x21	–	–	–	–	–	53.1
0x20	–	–	–	–	–	51.6
0x1f	–	–	–	–	–	50.0
0x1e	–	–	–	–	96.9	48.4
0x1d	–	–	–	–	93.8	46.9
0x1c	–	–	–	–	90.6	45.3
0x1b	–	–	–	–	87.5	43.8
0x1a	–	–	–	–	84.4	42.2
0x19	–	–	–	–	81.3	40.6
0x18	–	–	–	–	78.1	39.1
0x17	–	–	–	–	75.0	37.5

SNDDAT. SLEN[5:0] bits	Duty ratio by buzzer frequency					
	16,384 Hz	8,192 Hz	4,096 Hz	2,048 Hz	1,024 Hz	512 Hz
0x16	–	–	–	–	71.9	35.9
0x15	–	–	–	–	68.8	34.4
0x14	–	–	–	–	65.6	32.8
0x13	–	–	–	–	62.5	31.3
0x12	–	–	–	–	59.4	29.7
0x11	–	–	–	–	56.3	28.1
0x10	–	–	–	–	53.1	26.6
0x0f	–	–	–	–	50.0	25.0
0x0e	–	–	–	93.8	46.9	23.4
0x0d	–	–	–	87.5	43.8	21.9
0x0c	–	–	–	81.3	40.6	20.3
0x0b	–	–	–	75.0	37.5	18.8
0x0a	–	–	–	68.8	34.4	17.2
0x09	–	–	–	62.5	31.3	15.6
0x08	–	–	–	56.3	28.1	14.1
0x07	–	–	–	50.0	25.0	12.5
0x06	–	–	87.5	43.8	21.9	10.9
0x05	–	–	75.0	37.5	18.8	9.4
0x04	–	–	62.5	31.3	15.6	7.8
0x03	–	–	50.0	25.0	12.5	6.3
0x02	–	75.0	37.5	18.8	9.4	4.7
0x01	–	50.0	25.0	12.5	6.3	3.1
0x00	50.0	25.0	12.5	6.3	3.1	1.6

16.4.3 Buzzer Output in One-shot Buzzer Mode

One-shot buzzer mode is provided for clicking sound and short-duration buzzer output. This mode generates a buzzer signal with the software specified frequency and duty ratio, and outputs the generated signal for the short duration specified.
An output start procedure and the SNDA_DMM operations are shown below. For the buzzer output waveform, refer to "Buzzer Output in Normal Buzzer Mode."

One-shot buzzer output start procedure

1. Set the following SNDSEL register bits:
   - Set the SNDSEL.MOSEL[1:0] bits to 0x1. (Set one-shot buzzer mode)
   - SNDSEL.STIM[3:0] bits (Set output duration)
2. Write data to the following sound buffer (SNDDAT register) bits. (Start buzzer output)
   - SNDDAT.SLEN[5:0] bits (Set buzzer output signal duty ratio)
   - SNDDAT.SFRQ[7:0] bits (Set buzzer output signal frequency)

One-shot buzzer output operations

When data is written to the sound buffer (SNDDAT register), SNDA_DMM clears the SNDINTF.EMIF bit (sound buffer empty interrupt flag) to 0 and starts buzzer output operations.
The data written to the sound buffer is loaded into the sound register in sync with the CLK_SNDA_DMM clock. At the same time, the SNDINTF.EMIF bit and SNDINTF.SBSY bit are both set to 1. The output pin outputs the buzzer signal with the frequency/duty ratio specified.
The buzzer output automatically stops when the duration specified by the SNDSEL.STIM[3:0] bits has elapsed. At the same time, the SNDINTF.EDIF bit (sound output completion interrupt flag) is set to 1 and the SNDINTF.SBSY bit is cleared to 0.
Figure 16.4.3.1 shows a buzzer output timing chart in one-shot buzzer mode.

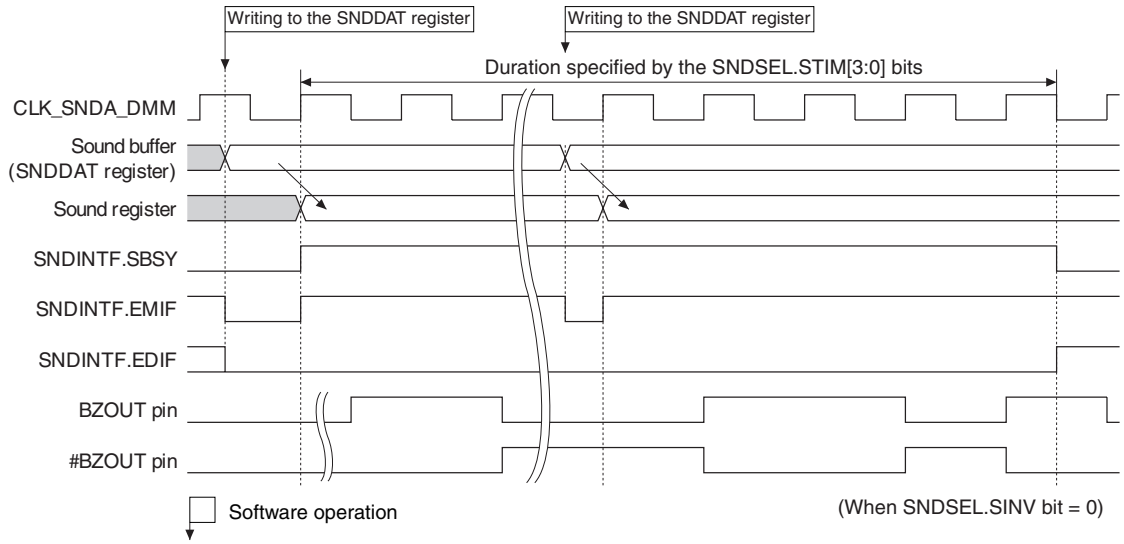


Figure 16.4.3.1 Buzzer Output Timing Chart in One-shot Buzzer Mode

16.4.4 Output in Melody Mode

Melody mode generates the buzzer signal with a melody according to the data written to the sound buffer (SNDDAT register) successively, and outputs the generated signal to outside the IC. An output start procedure and the SNDA_DMM operations are shown below.

Melody output start procedure

1. Set the following SNDSEL register bits:
   - Set the SNDSEL.MOSEL[1:0] bits to 0x2. (Set melody mode)
   - SNDSEL.STIM[3:0] bits (Set tempo)
2. Write data to the following sound buffer (SNDDAT register) bits. (Start sound output)
   - SNDDAT.MDTI bit (Set tie)
   - SNDDAT.MDRS bit (Set note/rest)
   - SNDDAT.SLEN[5:0] bits (Set duration)
   - SNDDAT.SFRQ[7:0] bits (Set scale)
3. Check to see if the SNDINTF.EMIF bit is set to 1 (an interrupt can be used).
4. Repeat Steps 2 and 3 until the end of the melody.

Melody output operations

When data is written to the sound buffer (SNDDAT register), SNDA_DMM clears the SNDINTF.EMIF bit (sound buffer empty interrupt flag) to 0 and starts sound output operations.

The data written to the sound buffer is loaded into the sound register by the internal trigger signal. At the same time, the SNDINTF.EMIF bit and SNDINTF.SBSY bit are both set to 1. The output pin outputs the sound specified.

The sound output stops if data is not written to the sound buffer (SNDDAT register) until the next trigger is issued. At the same time, the SNDINTF.EDIF bit (sound output completion interrupt flag) is set to 1 and the SNDINTF.SBSY bit is cleared to 0.

Figure 16.4.4.1 shows a melody mode operation timing chart.

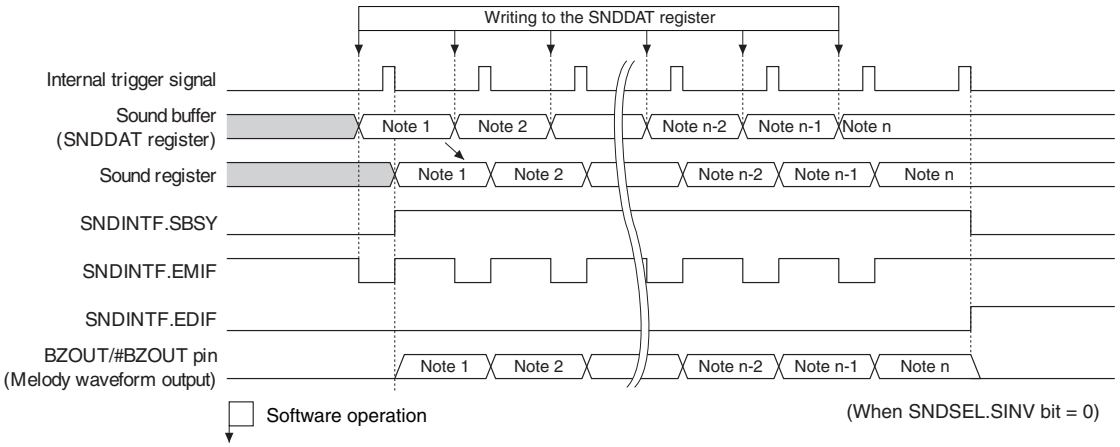


Figure 16.4.4.1 Melody Mode Operation Timing Chart

Melody output waveform configuration

Note/rest (duration) specification

Notes and rests can be specified using the SNDDAT.MDRS and SNDDAT.SLEN[5:0] bits.

Table 16.4.4.1 Note/Rest Specification (when $f_{CLK\_SNDA\_DMM}$ = 32,768 Hz)

SNDDAT.SLEN[5:0] bits	SNDDAT.MDRS bit	
	0: Note	1: Rest
0x0f	Half note	Half rest
0x0b	Dotted quarter note	Dotted quarter rest
0x07	Quarter note	Quarter rest
0x05	Dotted eighth note	Dotted eighth rest
0x03	Eighth note	Eighth rest
0x01	Sixteenth note	Sixteenth rest
0x00	Thirty-second note	Thirty-second rest
Other	Setting prohibited	

Tie specification

A tie takes effect by setting the SNDDAT.MDTI bit to 1 and the current note with this bit set to 1 and the following note are played continuously. A slur cannot be specified.

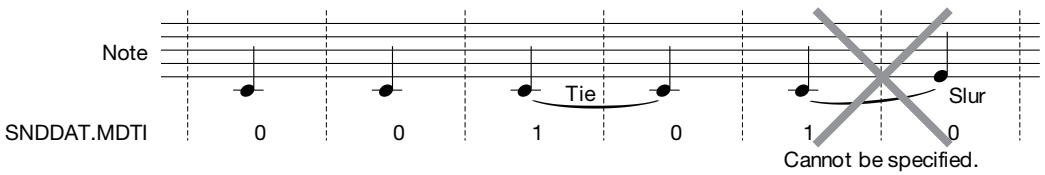


Figure 16.4.4.2 Tie and Slur

Scale specification

Scales can be specified using the SNDDAT.SFRQ[7:0] bits.

Table 16.4.4.2 Scale Specification (when $f_{CLK\_SNDA\_DMM}$ = 32,768 Hz)

SNDDAT.SFRQ[7:0] bits	Scale	Frequency [Hz]
0xf8	C3	131.60
0xea	C#3 / D♭3	139.44
0xdd	D3	147.60
0xd1	D#3 / E♭3	156.04
0xc5	E3	165.49
0xba	F3	175.23
0xaf	F#3 / G♭3	186.18
0xa5	G3	197.40
0x9c	G#3 / A♭3	208.71
0x93	A3	221.41
0x8b	A#3 / B♭3	234.06
0x83	B3	248.24

SNDDAT.SFRQ[7:0] bits	Scale	Frequency [Hz]
0x7c	C4	262.14
0x75	C#4 / D♭4	277.69
0x6e	D4	295.21
0x68	D#4 / E♭4	312.08
0x62	E4	330.99
0x5c	F4	352.34
0x57	F#4 / G♭4	372.36
0x52	G4	394.80
0x4e	G#4 / A♭4	414.78
0x49	A4	442.81
0x45	A#4 / B♭4	468.11
0x41	B4	496.48
0x3d	C5	528.52
0x3a	C#5 / D♭5	555.39
0x37	D5	585.14
0x33	D#5 / E♭5	630.15
0x30	E5	668.73
0x2e	F5	697.19
0x2b	F#5 / G♭5	744.73
0x29	G5	780.19
0x26	G#5 / A♭5	840.21
0x24	A5	885.62
0x22	A#5 / B♭5	936.23
0x20	B5	992.97
0x1e	C6	1057.03

16.4.5 DMM Linking Mode

The SNDSEL.CONT bit is provided to switch between normal mode and DMM linking mode.

Normal mode (SNDSEL.CONT bit = 0)

In this mode, SNDA_DMM starts outputting the buzzer signal by writing data to the sound buffer (SNDDAT register) as described in the previous sections. The DMM statue does not affect the buzzer output.

DMM linking mode (SNDSEL.CONT bit = 1)

In this mode, SNDA_DMM does not start outputting the buzzer signal even if data is written to the sound buffer (SNDDAT register).

When the DMM controller in continuity check mode detects a continuity status, the DMM controller outputs a buzzer output control signal to SNDA_DMM. This enables SNDA_DMM to output the buzzer signal according to the sound output mode and sound buffer data, that have been set in advance, to the BZOUT and #BZOUT pins. After that, when the continuity check result becomes a non-continuity status, the DMM controller negates the buzzer output control signal to stop the buzzer output. While in non-continuity status, the BZOUT pin goes low and the #BZOUT pin outputs a high or low level according to the SNDSEL.SINV bit setting (output pin drive mode).

16.5 Interrupts

SNDA_DMM has a function to generate the interrupts shown in Table 16.5.1.

Table 16.5.1 SNDA_DMM Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Sound buffer empty	SNDINTF.EMIF	When data in the sound buffer (SNDDAT register) is transferred to the sound register or 1 is written to the SNDCTL.SSTP bit	Writing to the SNDDAT register
Sound output completion	SNDINTF.EDIF	When a sound output has completed	Writing 1 or writing to the SNDDAT register

SNDA_DMM provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the “Interrupt Controller” chapter.

16.6 Control Registers

SNDA_DMM Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SNDCLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/W	
	7	–	0	–	R	
	6–4	CLKDIV[2:0]	0x0	H0	R/W	
	3–2	–	0x0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/W	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the SNDA_DMM operating clock is supplied in DEBUG mode or not.

1 (R/W): Clock supplied in DEBUG mode

0 (R/W): No clock supplied in DEBUG mode

Bit 7 Reserved

Bits 6–4 CLKDIV[2:0]

These bits select the division ratio of the SNDA_DMM operating clock.

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of SNDA_DMM.

Table 16.6.1 Clock Source and Division Ratio Settings

SNDCLK. CLKDIV[2:0] bits	SNDCLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0x7	Reserved	1/1	Reserved	1/1
0x6				
0x5	1/128		1/512	
0x4	1/64		1/256	
0x3	1/32		1/128	
0x2	1/16		1/64	
0x1	1/8		1/32	
0x0	1/4		1/16	

(Note) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

Note: The SNDCLK register settings can be altered only when the SNDCTL.MODEN bit = 0.

SNDA_DMM Select Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SNDSEL	15–12	–	0x0	–	R	–
	11–8	STIM[3:0]	0x0	H0	R/W	
	7–4	–	0x0	–	R	
	3	CONT	0	H0	R/W	
	2	SINV	0	H0	R/W	
	1–0	MOSEL[1:0]	0x0	H0	R/W	

Bits 15–12 Reserved

Bits 11–8 STIM[3:0]

These bits select a tempo (when melody mode is selected) or a one-shot buzzer output duration (when one-shot buzzer mode is selected).

Table 16.6.2 Tempo/One-shot Buzzer Output Duration Selections (when $f_{CLK\_SNDA\_DMM}$ = 32,768 Hz)

SNDSEL. STIM[3:0] bits	Tempo (= Quarter note/minute)	One-shot buzzer output duration [ms]
0xf	30	250.0
0xe	32	234.4
0xd	34.3	218.8
0xc	36.9	203.1
0xb	40	187.5
0xa	43.6	171.9
0x9	48	156.3
0x8	53.3	140.6
0x7	60	125.0
0x6	68.6	109.4
0x5	80	93.8
0x4	96	78.1
0x3	120	62.5
0x2	160	46.9
0x1	240	31.3
0x0	480	15.6

Note: Be sure to avoid altering these bits when SNDINTF.SBSY bit = 1.

Bits 7–4 Reserved

Bit 3 CONT

This bit selects DMM linking mode.

1 (R/W): DMM linking mode

0 (R/W): Normal mode

For more information, refer to "DMM Linking Mode."

Bit 2 SINV

This bit selects an output pin drive mode.

1 (R/W): Normal drive mode

0 (R/W): Direct drive mode

For more information, refer to "Output Pin Drive Mode."

Bits 1–0 MOSEL[1:0]

These bits select a sound output mode.

Table 16.6.3 Sound Output Mode Selection

SNDSEL.MOSEL[1:0] bits	Sound output mode
0x3	Reserved
0x2	Melody mode
0x1	One-shot buzzer mode
0x0	Normal buzzer mode

SNDA_DMM Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SNDCTL	15–9	–	0x00	–	R	–
	8	SSTP	0	H0	R/W	
	7–1	–	0x00	–	R	
	0	MODEN	0	H0	R/W	

Bits 15–9 Reserved

Bit 8 SSTP

This bit stops sound output.

1 (W): Stop sound output

0 (W): Ineffective

1 (R): In stop process

0 (R): Stop process completed/Idle

The SNDCTL.SSTP bit is used to stop buzzer output in normal buzzer mode. After 1 is written, this bit is cleared to 0 when the sound output has completed. Also in one-shot buzzer mode/melody mode, writing 1 to this bit can forcibly terminate the sound output.

Bits 7–1 Reserved

Bit 0 MODEN

This bit enables the SNDA_DMM operations.
1 (R/W): Enable SNDA_DMM operations (The operating clock is supplied.)
0 (R/W): Disable SNDA_DMM operations (The operating clock is stopped.)

SNDA_DMM Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SNDDAT	15	MDTI	0	H0	R/W	–
	14	MDRS	0	H0	R/W	
	13–8	SLEN[5:0]	0x00	H0	R/W	
	7–0	SFRQ[7:0]	0xff	H0	R/W	

This register functions as a sound buffer. Writing data to this register starts sound output. For detailed information on the setting data, refer to "Buzzer output waveform configuration (normal buzzer mode/one-shot buzzer mode)" and "Melody output waveform configuration."

Bit 15 MDTI

This bit specifies a tie (continuous play between the current note with this bit set to 1 and the subsequent note) in melody mode.
1 (R/W): Enable tie
0 (R/W): Disable tie

The successive notes must be the same scale. Therefore, a slur cannot be specified. If a continuous play is specified to the notes with a different scale, the setting of this bit is ignored.
This bit is also ignored in normal buzzer mode/one-shot buzzer mode.

Bit 14 MDRS

This bit selects the output type in melody mode from a note or a rest .
1 (R/W): Rest
0 (R/W): Note

When a rest is selected, the BZOUT pin goes low and the #BZOUT pin goes high during the output duration. This bit is ignored in normal buzzer mode/one-shot buzzer mode.

Bits 13–8 SLEN[5:0]

These bits select a duration (when melody mode is selected) or a buzzer signal duty ratio (when normal buzzer mode/one-shot buzzer mode is selected).

Bits 7–0 SFRQ[7:0]

These bits select a scale (when melody mode is selected) or a buzzer signal frequency (when normal buzzer mode/one-shot buzzer mode is selected).

Notes: • In normal buzzer mode/one-shot buzzer mode, only the low-order 6 bits (SNDDAT.SFRQ[5:0] bits) are effective within the SNDDAT.SFRQ[7:0] bits. Always set the SNDDAT.SFRQ[7:6] bits to 0x0.

• The SNDDAT register allows 16-bit data writing only. Data writings in 8-bit size will be ignored.

SNDA_DMM Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SNDINTF	15–9	–	0x00	–	R	–
	8	SBSY	0	H0	R	
	7–2	–	0x00	–	R	
	1	EMIF	1	H0	R	Cleared by writing to the SNDDAT register.
	0	EDIF	0	H0	R/W	Cleared by writing 1 or writing to the SNDDAT register.

Bits 15–9 Reserved

Bit 8 SBSY

This bit indicates the sound output status. (See Figures 16.4.2.1, 16.4.3.1, and 16.4.4.1.)

1 (R): Outputting

0 (R): Idle

Bits 7–2 Reserved

Bit 1 EMIF

Bit 0 EDIF

These bits indicate the SNDA_DMM interrupt cause occurrence status.

1 (R): Cause of interrupt occurred

0 (R): No cause of interrupt occurred

1 (W): Clear flag

0 (W): Ineffective

The following shows the correspondence between the bit and interrupt:

SNDINTF.EMIF bit: Sound buffer empty interrupt

SNDINTF.EDIF bit: Sound output completion interrupt

SNDA_DMM Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SNDINTE	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1	EMIE	0	H0	R/W	
	0	EDIE	0	H0	R/W	

Bits 15–2 Reserved

Bit 1 EMIE

Bit 0 EDIE

These bits enable SNDA_DMM interrupts.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:

SNDINTE.EMIE bit: Sound buffer empty interrupt

SNDINTE.EDIE bit: Sound output completion interrupt

17 LCD Driver (LCD4B)

17.1 Overview

LCD4B is an LCD driver to drive an LCD panel. The features of the LCD4B are listed below.

- The frame frequency is configurable into 16 steps.
- Provides all on, all off, and inverse display functions as well as normal display.
- The segment and common pin assignments can be inverted.
- Provides a partial common output drive function.
- Provides an n-segment-line inverse AC drive function.
- The LCD contrast is adjustable into 32 steps.
- Includes a power supply for 1/3 bias driving
  (allows external voltages to be applied, voltage-dividing resistors included).
- Can generate interrupts every frame.

Figure 17.1.1 shows the LCD4B configuration.

Table 17.1.1 LCD4B Configuration of S1C17M02/M03

Item	S1C17M02	S1C17M03
Number of segments supported	Max. 64 segments (16SEG × 4COM)	Max. 128 segments (32SEG × 4COM)
SEG/COM outputs	16SEG × 1–4COM	32SEG × 1–4COM
Drive bias	1/3 bias	
Embedded display data RAM	16 bytes	32 bytes
LCD power supply	Internal generation mode/ External voltage application mode 1/2/3	
LFRO output	Available	

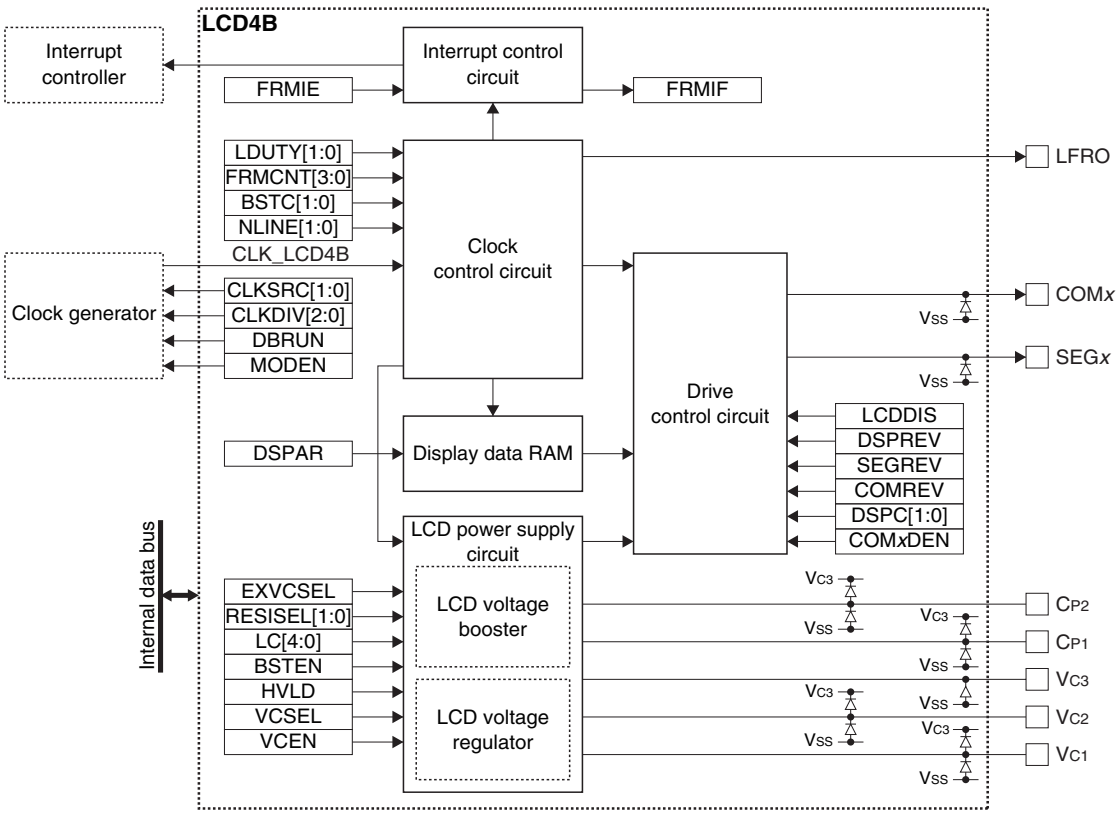


Figure 17.1.1 LCD4B Configuration

17.2 Output Pins and External Connections

17.2.1 List of Output Pins

Table 17.2.1.1 lists the LCD4B pins.

Table 17.2.1.1 List of LCD4B Pins

Pin name	I/O*1	Initial status*1	Function
COM0–3	A	Hi-Z / O (L)*2	Common data output-only pins
SEG0–7	A	Hi-Z / O (L)*2	Segment data output-only pins
SEG8–15	A	Hi-Z / O (L)*2	General-purpose IO/segment data output pins
SEG16–31	A	Hi-Z / O (L)*2	General-purpose IO/segment data output pins (Not available in the S1C17M02)
LFRO	O	O (L)	Frame signal monitoring output pin
V_{C1}	P	–	LCD panel drive power supply pin
V_{C2}	P	–	LCD panel drive power supply pin
V_{C3}	P	–	LCD panel drive power supply pin
C_{P1}	A	–	LCD voltage booster capacitor connecting pin
C_{P2}	A	–	LCD voltage booster capacitor connecting pin

*1: Indicates the status when the pin is configured for LCD4B. *2: When LCD4CTL.LCDDIS bit = 1

If the port is shared with the LCD4B pin and other functions, the LCD4B output function must be assigned to the port before activating the LCD4B. For more information, refer to the "I/O Ports" chapter.

Notes: • Be sure to avoid using the V_{C1} to V_{C3} pin outputs for driving external circuits.

• When an LCD panel is connected, the LCD4CTL.LCDDIS bit should be set to 1. If it is set to 0, the LCD panel characteristics may fluctuate.

17.2.2 External Connections

Figure 17.2.2.1 shows a connection diagram between LCD4B and an LCD panel.

Note: When the panel is connected, the LCD4CTL.LCDDIS bit must be set to 1 to bias the panel even if display is turned off.

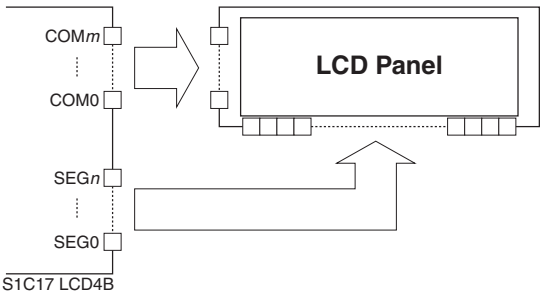


Figure 17.2.2.1 Connections between LCD4B and an LCD Panel

17.3 Clock Settings

17.3.1 LCD4B Operating Clock

When using LCD4B, the LCD4B operating clock CLK_LCD4B must be supplied to LCD4B from the clock generator. The CLK_LCD4B supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to "Clock Generator" in the "Power Supply, Reset, and Clocks" chapter).
2. Set the following LCD4CLK register bits:
   - LCD4CLK.CLKSRC[1:0] bits (Clock source selection)
   - LCD4CLK.CLKDIV[2:0] bits (Clock division ratio selection = Clock frequency setting)

The CLK_LCD4B frequency should be set to around 32 kHz.

17.3.2 Clock Supply in SLEEP Mode

When using LCD4B during SLEEP mode, the LCD4B operating clock CLK_LCD4B must be configured so that it will keep supplying by writing 0 to the CLGOSC.*xxxx*SLPC bit for the CLK_LCD4B clock source.

17.3.3 Clock Supply in DEBUG Mode

The CLK_LCD4B supply during DEBUG mode should be controlled using the LCD4CLK.DBRUN bit.
The CLK_LCD4B supply to LCD4B is suspended when the CPU enters DEBUG mode if the LCD4CLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_LCD4B supply resumes. Although LCD4B stops operating and the display is turned off when the CLK_LCD4B supply is suspended, the registers retain the status before DEBUG mode was entered. If the LCD4CLK.DBRUN bit = 1, the CLK_LCD4B supply is not suspended and LCD4B will keep operating in DEBUG mode.

17.3.4 Frame Frequency

The LCD4B frame signal is generated by dividing CLK_LCD4B. The frame frequency is determined by selecting a division ratio from 16 variations depending on the drive duty using the LCD4TIM1.FRMCNT[3:0] bits. Use the following equation to calculate the frame frequency.

$$f_{FR} = \frac{f_{CLK\_LCD4B}}{16 \times (FRMCNT + 1) \times (LDUTY + 1)} \qquad \text{(Eq. 17.1)}$$

Where

- f_{FR}: Frame frequency [Hz]
- f_{CLK_LCD4B}: LCD4B operating clock frequency [Hz]
- FRMCNT: LCD4TIM1.FRMCNT[3:0] setting value (0 to 15)
- LDUTY: LCD4TIM1.LDUTY[1:0] setting value (0 to 3)

Table 17.3.4.1 lists frame frequency settings when f_{CLK_LCD4B} = 32,768 Hz as an example.

Table 17.3.4.1 Frame Frequency Settings (when f_{CLK_LCD4B} = 32,768 Hz)

LCD4TIM1. FRMCNT[3:0] bits	Frame frequency [Hz]			
	1/4 duty	1/3 duty	1/2 duty	Static
0xf	32.0	42.7	64.0	128.0
0xe	34.1	45.5	68.3	136.5
0xd	36.6	48.8	73.1	146.3
0xc	39.4	52.5	78.8	157.5
0xb	42.7	56.9	85.3	170.7
0xa	46.5	62.1	93.1	186.2
0x9	51.2	68.3	102.4	204.8
0x8	56.9	75.9	113.8	227.6
0x7	64.0	85.3	128.0	256.0
0x6	73.1	97.5	146.3	292.6
0x5	85.3	113.8	170.7	341.3
0x4	102.4	136.5	204.8	409.6
0x3	128.0	170.7	256.0	512.0
0x2	170.7	227.6	341.3	682.7
0x1	256.0	341.3	512.0	1,024.0
0x0	512.0	682.7	1,024.0	2,048.0

17.4 LCD Power Supply

The LCD drive voltages V_{C1} to V_{C3} can be generated by the internal LCD power supply circuit (LCD voltage regulator and LCD voltage booster). One or all voltages can also be applied from outside the IC.

17.4.1 Internal Generation Mode

This mode generates all the LCD drive voltages V_{C1} to V_{C3} on the chip. To put LCD4B into internal generation mode, set both the LCD4PWR.VCEN and LCD4PWR.BSTEN bits to 1 to turn both the LCD voltage regulator and LCD voltage booster on after setting the LCD4PWR.EXVCSEL bit to 0. The LCD4PWR.RESISEL[1:0] bits should be set to 0x0 to disable the internal LCD voltage dividing resistors. Figure 17.4.1.1 shows an external connection example for internal generation mode.

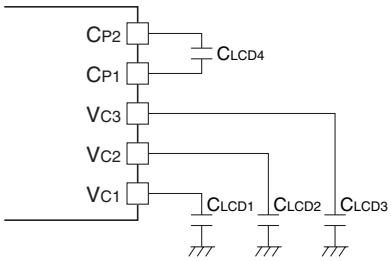


Figure 17.4.1.1 External Connection Example for Internal Generation Mode

17.4.2 External Voltage Application Mode 1

In this mode, all the LCD drive voltages V_{C1} to V_{C3} are applied from outside the IC. To put LCD4B into external voltage application mode 1, set the LCD4PWR.EXVCSEL bit to 1 and set both the LCD4PWR.VCEN and LCD4PWR.BSTEN bits to 0 to turn both the LCD voltage regulator and LCD voltage booster off. The LCD4PWR.RESISEL[1:0] bits should be set to 0x0 to disable the internal LCD voltage dividing resistors. Figure 17.4.2.1 shows an external connection example for external voltage application mode 1.

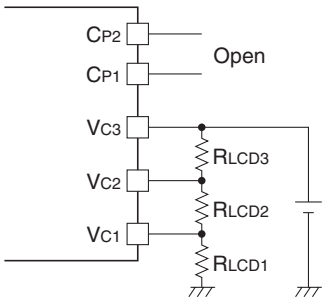


Figure 17.4.2.1 External Connection Example for External Voltage Application Mode 1 (resistor divider)

17.4.3 External Voltage Application Mode 2

In this mode, one of the LCD drive voltages V_{C1} to V_{C2} are applied from outside the IC and other voltages are internally generated. To put LCD4B into external voltage application mode 2, set the LCD4PWR.EXVCSEL bit to 1, set the LCD4PWR.VCEN bit to 0 to turn the LCD voltage regulator off, and set the LCD4PWR.BSTEN bit to 1 to turn the LCD voltage booster on. The LCD4PWR.RESISEL[1:0] bits should be set to 0x0 to disable the internal LCD voltage dividing resistors. Figure 17.4.3.1 shows an external connection example for external voltage application mode 2.

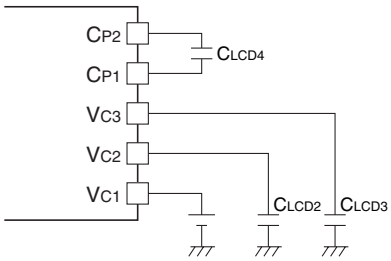


Figure 17.4.3.1 External Connection Example for External Voltage Application Mode 2 (when V_{C1} is applied)

17.4.4 External Voltage Application Mode 3

In this mode, the LCD drive voltage V_{C3} is applied from outside the IC and the V_{C1} and V_{C2} voltages are generated using the internal LCD voltage dividing resistors. To put LCD4B into external voltage application mode 3, set the LCD4PWR.EXVCSEL bit to 1 and set both the LCD4PWR.VCEN and LCD4PWR.BSTEN bits to 0 to turn both the LCD voltage regulator and LCD voltage booster off. Also set the LCD4PWR.RESISEL[1:0] bits to 0x1, 0x2, or 0x3 to use the internal LCD voltage dividing resistors according to the LCD panel load. A capacitor should be connected to the V_{C1} to V_{C3} pins while taking fluctuation of LCD load into consideration. Figure 17.4.4.1 shows an external connection example for external voltage application mode 3.

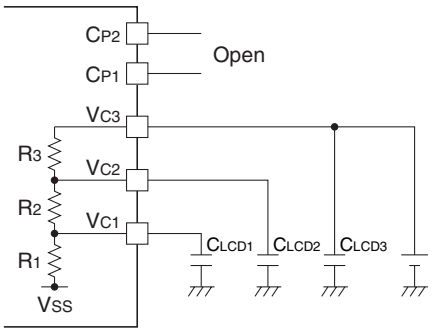


Figure 17.4.4.1 External Connection Example for External Voltage Application Mode 3

17.4.5 LCD Power Supply Circuit Settings

When using internal generation mode

Select the reference voltage for boosting voltage generated by the LCD voltage regulator according to the power supply voltage V_{DD}. Refer to "LCD Driver (LCD4B) Characteristics" in the "Electrical Characteristics" chapter and set the LCD4PWR.VCSEL bit. Current consumption can be reduced by selecting reference voltage V_{C2} as compared with reference voltage V_{C1}. By setting the LCD4PWR.HVLD bit to 1, the LCD voltage regulator enters heavy load protection mode and ensures stable V_{C1} to V_{C3} outputs. Heavy load protection mode should be set when the display has inconsistencies in density. Current consumption increases in heavy load protection mode, therefore do not set heavy load protection mode if unnecessary.

When using internal generation mode or external voltage application mode 2

Set the booster clock frequency used in the LCD voltage booster using the LCD4TIM2.BSTC[1:0] bits. Set it to the frequency that provides the best V_{C1}–V_{C3} output stability after being evaluated using the actual circuit board.

When using external voltage application mode 3

LCD4B includes voltage dividing resistors to generate the LCD drive voltages V_{C1} and V_{C2} from the V_{C3} that is applied externally. The resistance values can be adjusted according to the external LCD panel load by setting the LCD4PWR.REGISEL[1:0] bits.

LCD contrast adjustment

The LCD panel contrast can only be adjusted in internal generation mode using the LCD4PWR.LC[4:0] bits. For the adjustment range, refer to "LCD Driver (LCD4B) Characteristics" in the "Electrical Characteristics" chapter.

17.5 Operations

17.5.1 Initialization

The LCD4B should be initialized with the procedure shown below.

1. Assign the LCD4B output function to the ports. (Refer to the "I/O Ports" chapter.)
2. Configure the LCD4CLK.CLKSRC[1:0] and LCD4CLK.CLKDIV[2:0] bits. (Configure operating clock)
3. Configure the following LCD4CTL register bits:
   - Write 1 to the LCD4CTL.MODEN bit. (Enable LCD4B operating clock)
   - Write 1 to the LCD4CTL.LCDDIS bit. (Enable LCD driver pin discharge at display off)
4. Configure the following LCD4TIM1 register bits:
   - LCD4TIM1.LDUTY[1:0] bits (Set drive duty)
   - LCD4TIM1.FRMCNT[3:0] bits (Set frame frequency)
5. Configure the following LCD4TIM2 register bits:
   - LCD4TIM2.NLINE[1:0] bits (Set n-line inverse AC drive)
   - LCD4TIM2.BSTC[1:0] bits (Set booster clock frequency)
6. Set the LCD4PWR.EXVCSEL bit. (Select external voltage application mode/internal generation mode)
7. Configure the following LCD4PWR register bits:
   - LCD4PWR.RESISEL[1:0] bits (Select internal voltage dividing resistors)
   - LCD4PWR.VCEN bit (Enable LCD voltage regulator)
   - LCD4PWR.VCSEL bit (Set reference voltage for boosting)
   - LCD4PWR.BSTEN bit (Enable LCD voltage booster)
   - LCD4PWR.LC[4:0] bits (Set LCD contrast initial value)
8. Configure the following LCD4DSP register bits:
   - LCD4DSP.DSPAR bit (Select display area)
   - LCD4DSP.COMREV bit (Select COM pin assignment direction)
   - LCD4DSP.SEGREV bit (Select SEG pin assignment direction)
9. Write display data to the display data RAM.
10. Set the following bits when using the interrupt:
    - Write 1 to the LCD4INTF.FRMIF bit. (Clear interrupt flag)
    - Set the LCD4INTE.FRMIE bit to 1. (Enable LCD4B interrupt)

17.5.2 Display On/Off

The LCD display state is controlled using the LCD4DSP.DSPC[1:0] bits.

Table 17.5.2.1 LCD Display Control

LCD4DSP.DSPC[1:0] bits	LCD display
0x3	All off (static drive)
0x2	All on
0x1	Normal display
0x0	Display off

Selecting "Display off" stops the drive voltage supply and the LCD driver pin outputs are all set to Vss level when the LCD4CTL.LCDDIS bit = 1.
Since "All on" and "All off" directly control the driving waveform output by the LCD driver, data in the display data RAM is not altered. The common pins are set to dynamic drive for "All on" and to static drive for "All off." This function can be used to make the display flash on and off without altering the display memory.

Note: When "Display off" is selected while the external LCD drive voltages are being supplied in an external voltage application mode, the electric charges of V_{C3} must be discharged in the following procedure.
1. Turn the external power supply off.
2. Set the LCD4PWR.EXVCSEL bit to 0. (Select internal generation mode)
3. Set the LCD4PWR.EXVCSEL bit to 1. (Select external voltage application mode)

17.5.3 Inverted Display

The LCD panel display can be inverted (black/white inversion) using merely control bit manipulation, without rewriting the display data RAM. Setting the LCD4DSP.DSPREV bit to 0 inverts the display; setting it to 1 returns the display to normal status. Note that the display will not be inverted when the LCD4DSP.DSPC[1:0] bits = 0x3 (All off).

17.5.4 Drive Duty Switching

Drive duty can be set to 1/4 to 1/2 or static drive using the LCD4TIM1.LDUTY[1:0] bits. Table 17.5.4.1 shows the correspondence between the LCD4TIM1.LDUTY[1:0] bit settings, drive duty, and maximum number of display segments.

Table 17.5.4.1 Drive Duty Settings

LCD4TIM1. LDUTY[1:0] bits	Duty	Valid COM pins	Valid SEG pins		Max. number of display dots/segments	
			S1C17M02	S1C17M03	S1C17M02	S1C17M03
0x3	1/4	COM0–3	SEG0–15	SEG0–31	64	128
0x2	1/3	COM0–2			48	96
0x1	1/2	COM0–1			32	64
0x0	Static	COM0			16	32

Unused common pins output an OFF waveform that turns the segments off.

17.5.5 Drive Waveforms

Figures 17.5.5.1 to 17.5.5.4 show drive waveform examples by drive duty.

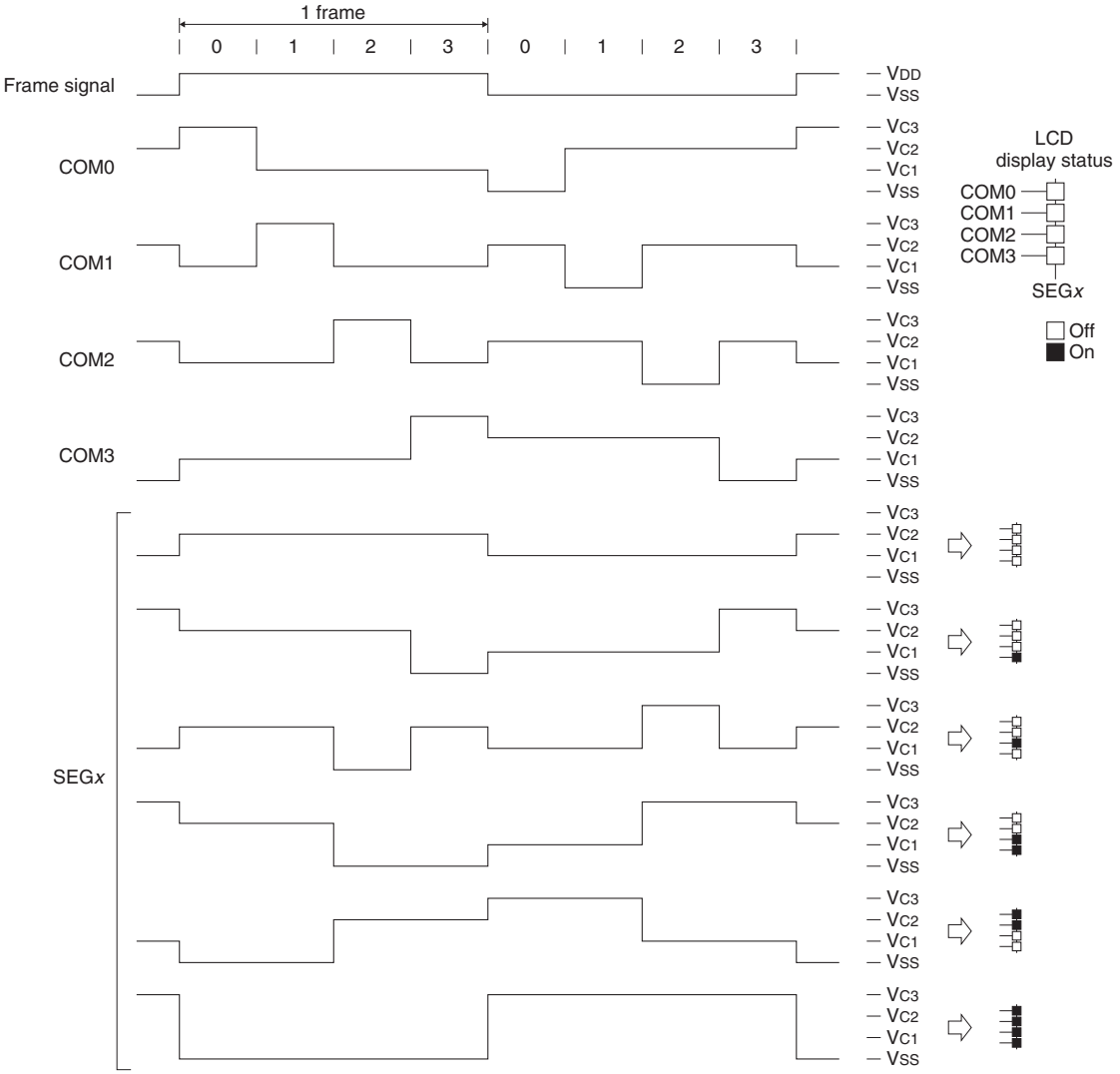


Figure 17.5.5.1 1/4 Duty Drive Waveform

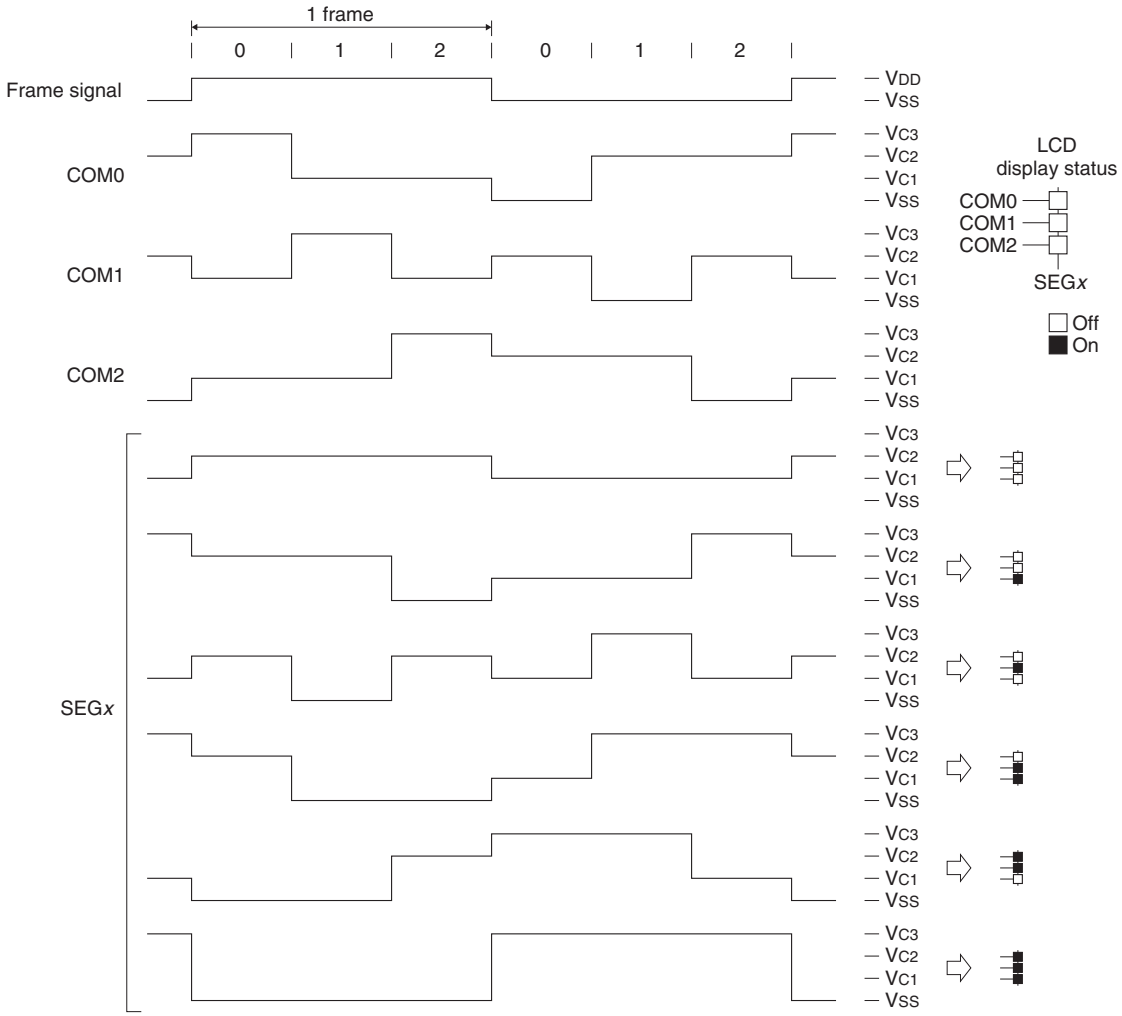


Figure 17.5.5.2 1/3 Duty Drive Waveform

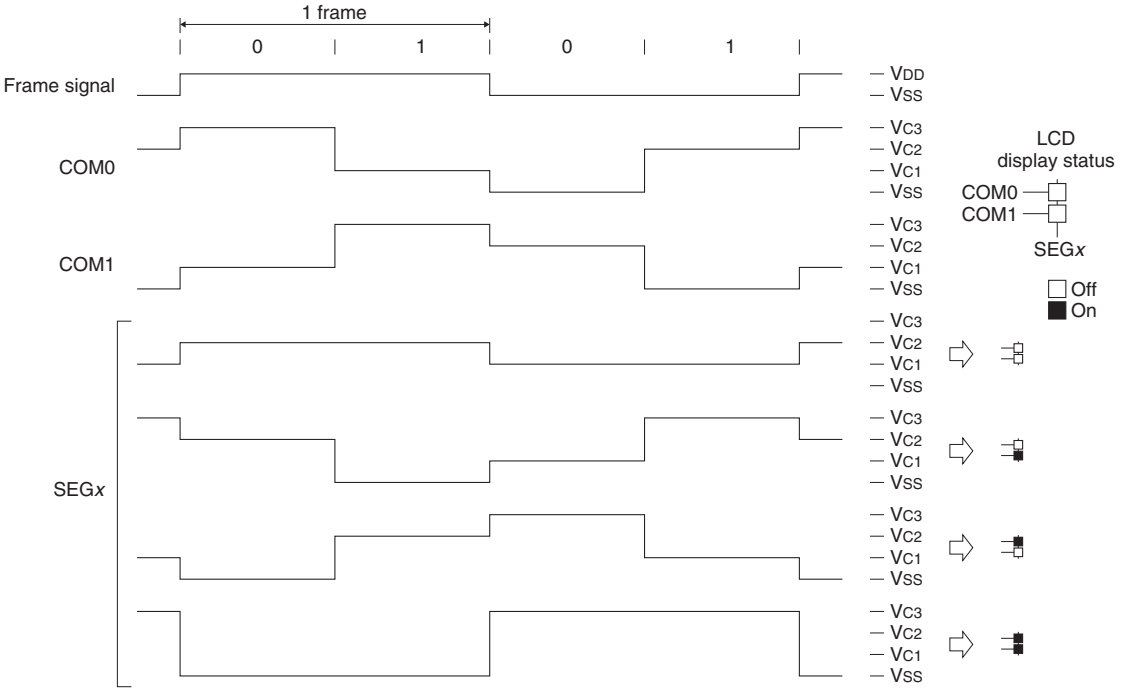


Figure 17.5.5.3 1/2 Duty Drive Waveform

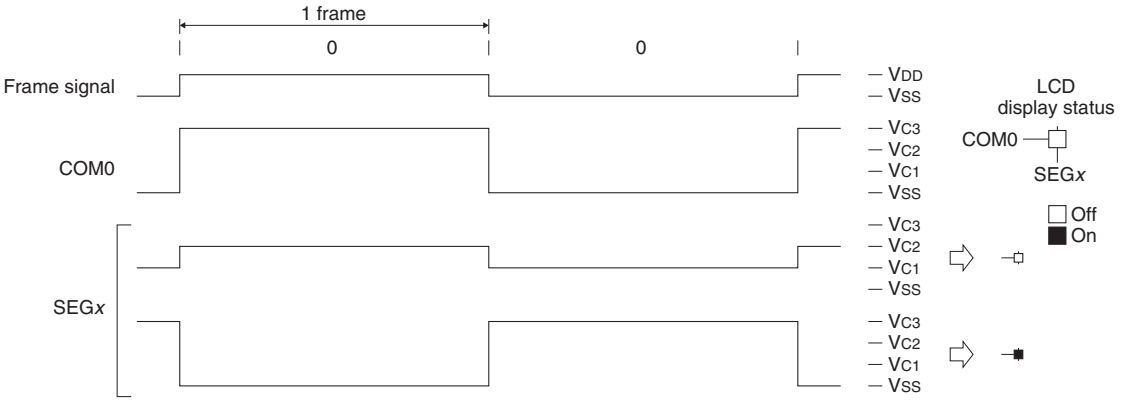


Figure 17.5.5.4 Static Drive Waveform

17.5.6 Partial Common Output Drive

By setting the LCD4COMC∗.COM*x*DEN bit (*x* = COM No.) to 0, any common outputs can be set to off waveform regardless of the display data RAM contents. The partial common output drive function limits the display to the required area only to reduce power consumption.

17.5.7 n-Segment-Line Inverse AC Drive

The n-line inverse AC drive function may improve the display quality when being reduced such as when crosstalk occurs. To activate the n-line inverse AC drive function, select the number of lines to be inverted using the LCD4TIM2.NLINE[1:0] bits. The setting value should be determined after being evaluated using the actual circuit board. Note that using the n-line inverse AC drive function increases current consumption.

Table 17.5.7.1 Selecting Number of Inverse Lines

LCD4TIM2.NLINE[1:0] bits	Number of inverse lines
0x3	3 lines
0x2	2 lines
0x1	1 line
0x0	Normal drive

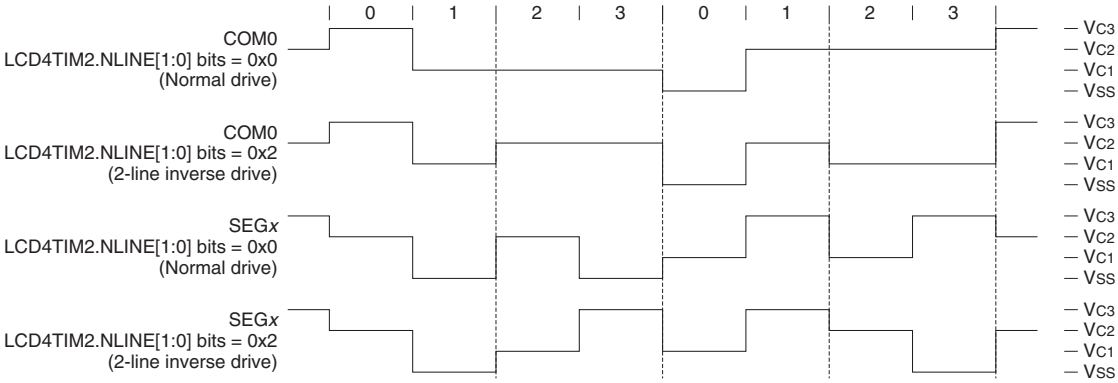


Figure 17.5.7.1 1/4 Duty Normal Drive Waveform and 2-line Inverse Drive Waveform

17.6 Display Data RAM

The display data RAM is located beginning with address 0x7000.
The correspondence between the memory bits of the display data RAM and the common/segment pins varies depending on the selected conditions below.

- Drive duty (1/4 to 1/2 or static drive)
- Segment pin assignment (normal or inverse)
- Common pin assignment (normal or inverse)

Figures 17.6.3.1 to 17.6.3.4 show the correspondence between display data RAM and the common/segment pins by drive duty.
Writing 1 to the display data RAM bit corresponding to a segment on the LCD panel turns the segment on, while writing 0 turns the segment off. Since the display memory is a RAM allowing reading and writing, bits can be controlled individually using logic operation instructions (read-modify-write instructions).
The area unused for display can be used as general-purpose RAM.

17.6.1 Display Area Selection

In the display data RAM, two screen areas can be allocated and the LCD4DSP.DSPAR bit can be used to switch between the screens. Setting the LCD4DSP.DSPAR bit to 0 selects display area 0; setting to 1 selects display area 1.

17.6.2 Segment Pin Assignment

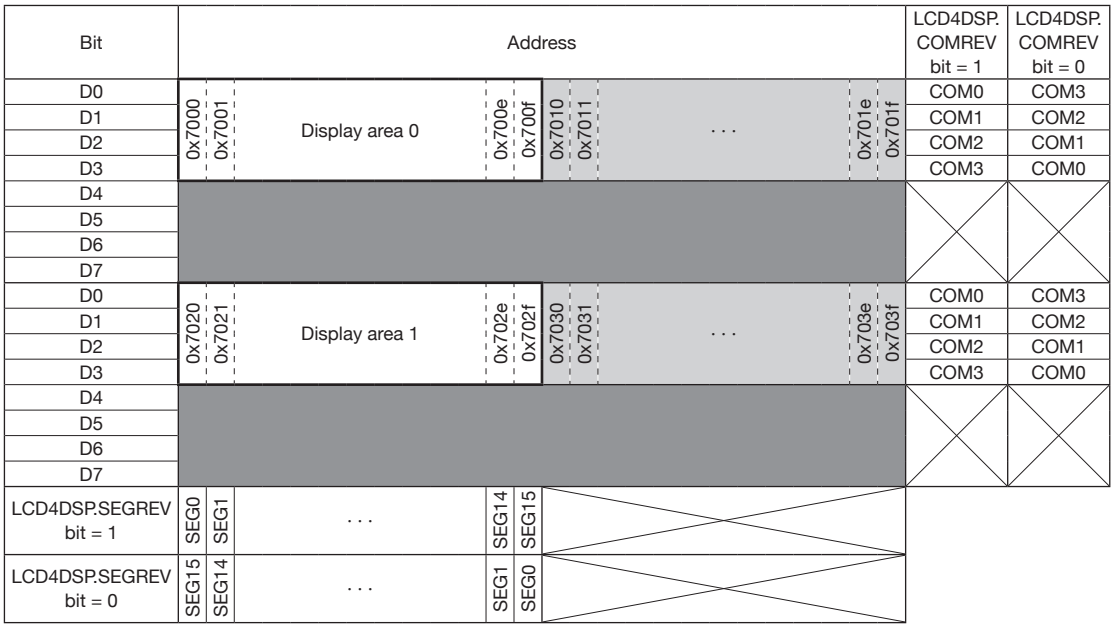
The display data RAM address assignment for the segment pins can be inverted using the LCD4DSP.SEGREV bit. When the LCD4DSP.SEGREV bit is set to 1, memory addresses are assigned to segment pins in ascending order. When the LCD4DSP.SEGREV bit is set to 0, memory addresses are assigned to segment pins in descending order.

17.6.3 Common Pin Assignment

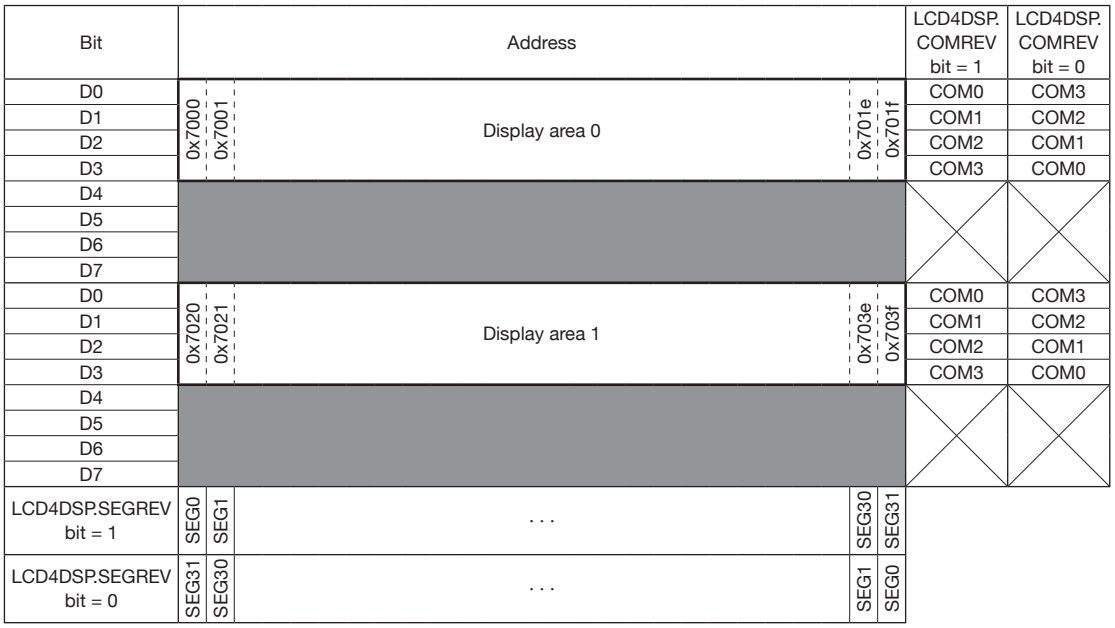
The display data RAM bit assignment for the common pins can be inverted using the LCD4DSP.COMREV bit. When the LCD4DSP.COMREV bit is set to 1, memory bits are assigned to common pins in ascending order. When the LCD4DSP.COMREV bit is set to 0, memory bits are assigned to common pins in descending order.

Unused area (general-purpose RAM)
Unimplemented area

1/4 duty



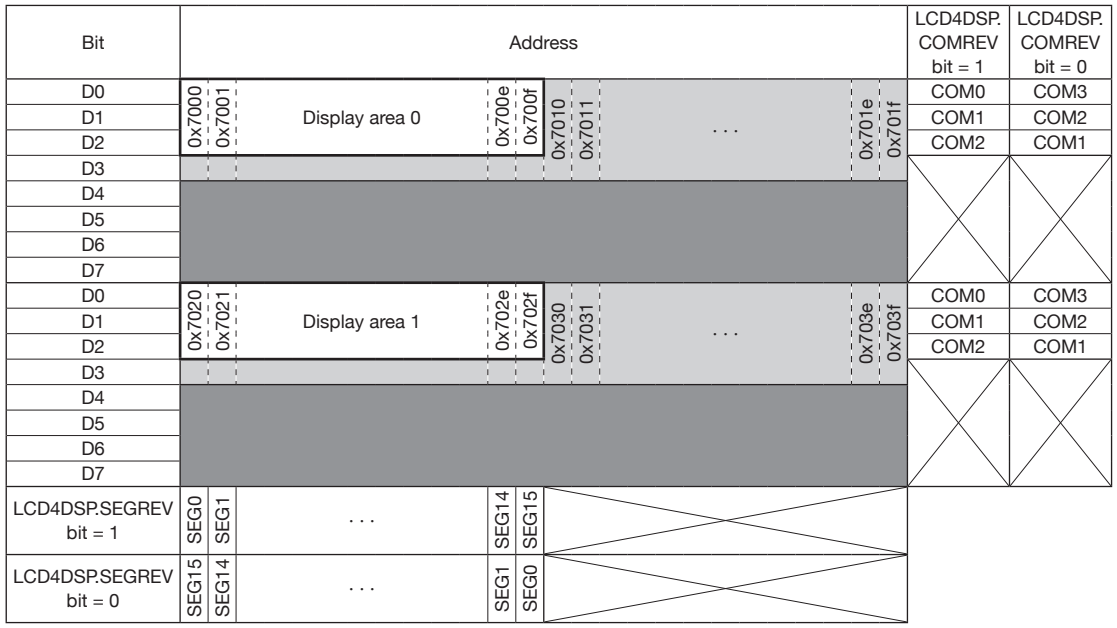
(1) S1C17M02



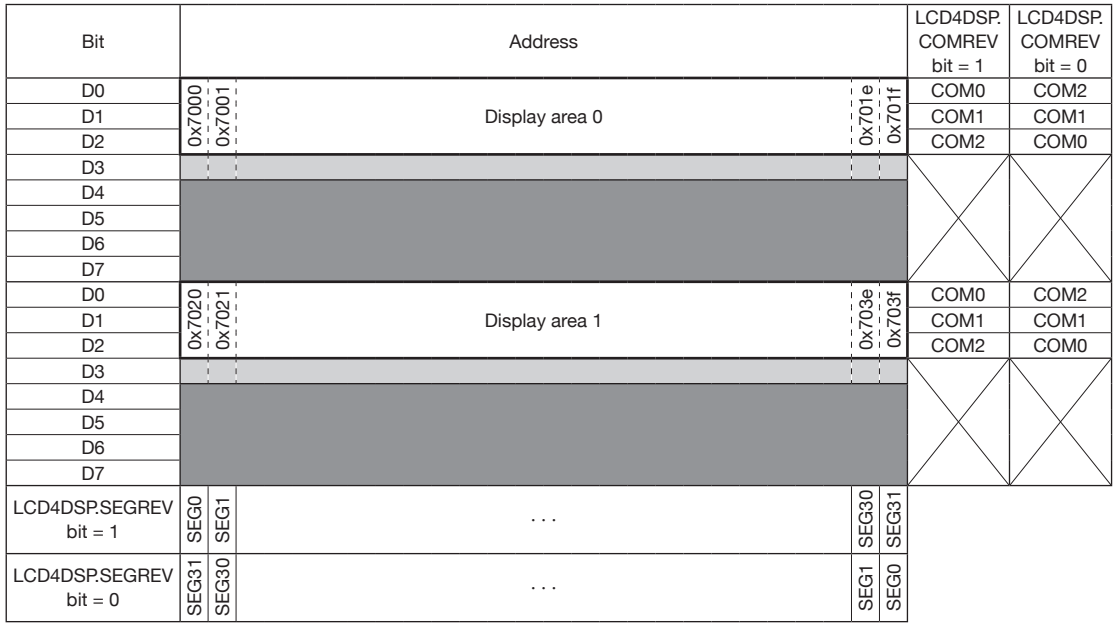
(2) S1C17M03

Figure 17.6.3.1 Display Data RAM Map (1/4 duty)

1/3 duty



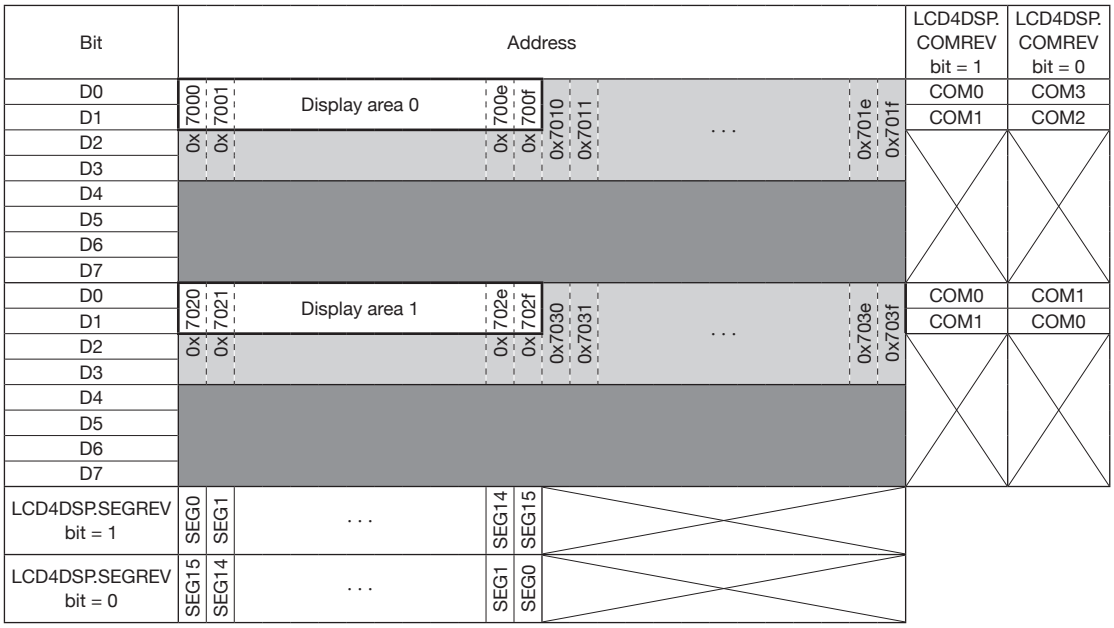
(1) S1C17M02



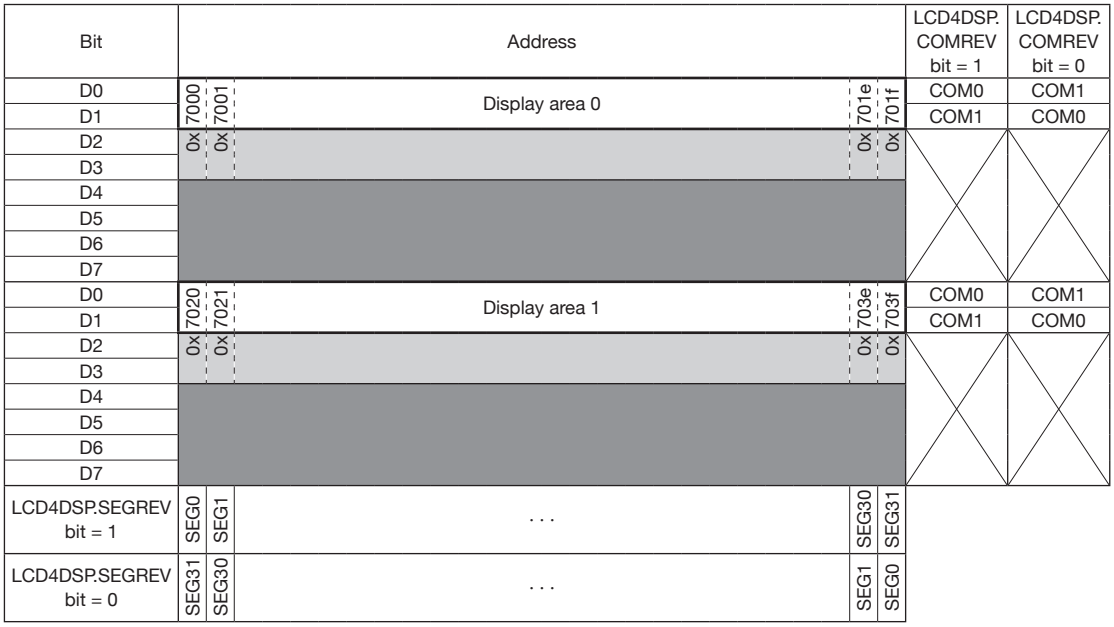
(2) S1C17M03

Figure 17.6.3.2 Display Data RAM Map (1/3 duty)

1/2 duty



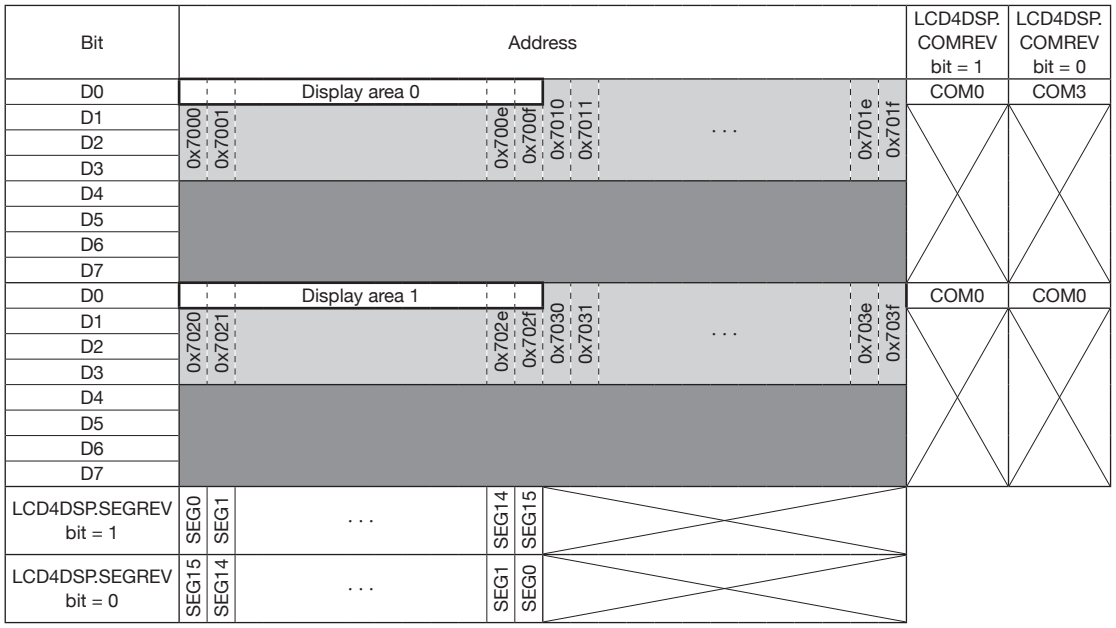
(1) S1C17M02



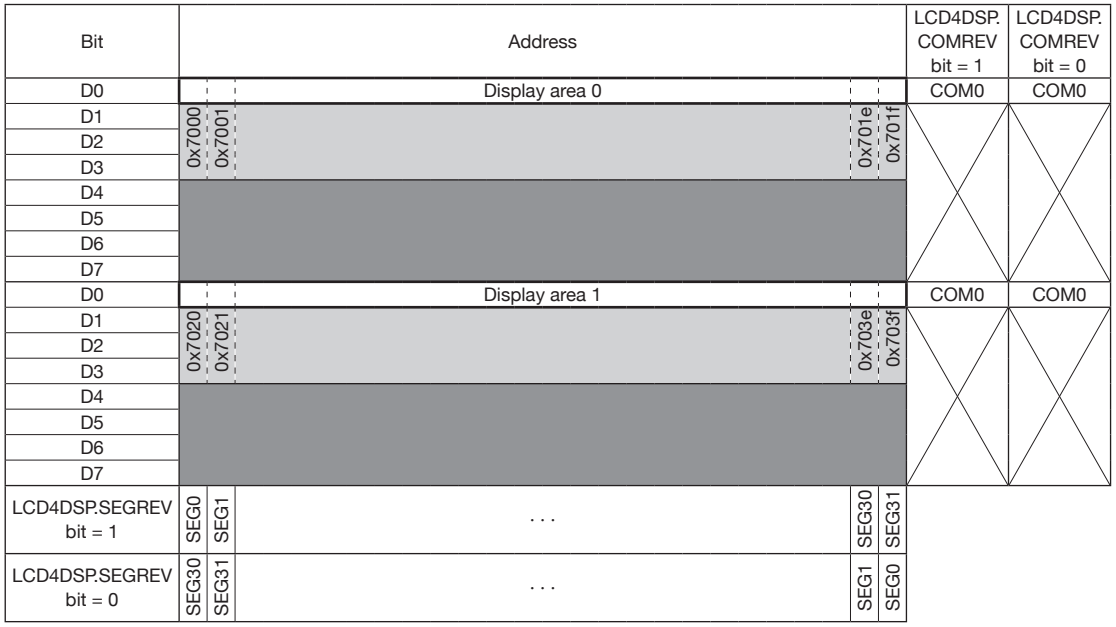
(2) S1C17M03

Figure 17.6.3.3 Display Data RAM Map (1/2 duty)

Static drive



(1) S1C17M02



(2) S1C17M03

Figure 17.6.3.4 Display Data RAM Map (static drive)

17.7 Interrupt

The LCD4B has a function to generate the interrupt shown in Table 17.7.1.

Table 17.7.1 LCD4B Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Frame	LCD4INTF.FRMIF	Frame switching	Writing 1

The LCD4B provides an interrupt enable bit corresponding to the interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the "Interrupt Controller" chapter.

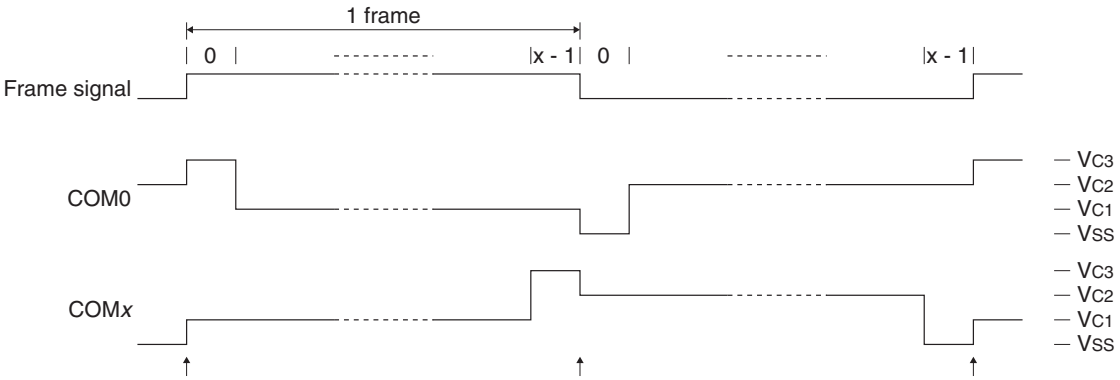


Figure 17.7.1 Frame Interrupt Timings (1/x duty)

17.8 Control Registers

LCD4B Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4CLK	15–9	–	0x00	–	R	–
	8	DBRUN	1	H0	R/W	
	7	–	0	–	R	
	6–4	CLKDIV[2:0]	0x0	H0	R/W	
	3–2	–	0x0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/W	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the LCD4B operating clock is supplied in DEBUG mode or not.

1 (R/W): Clock supplied in DEBUG mode

0 (R/W): No clock supplied in DEBUG mode

Bit 7 Reserved

Bits 6–4 CLKDIV[2:0]

These bits select the division ratio of the LCD4B operating clock.

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of the LCD4B.

Table 17.8.1 Clock Source and Division Ratio Settings

LCD4CLK.CLKDIV[2:0] bits	LCD4CLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0x7	Reserved	1/1	Reserved	1/1
0x6				
0x5	1/128		1/512	
0x4	1/64		1/256	
0x3	1/32		1/128	
0x2	1/16		1/64	
0x1	1/8		1/32	
0x0	1/4		1/16	

(Note) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

Note: The LCD4CLK register settings can be altered only when the LCD4CTL.MODEN bit = 0.

LCD4B Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4CTL	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1	LCDDIS	0	H0	R/W	
	0	MODEN	0	H0	R/W	

Bits 15–2 Reserved

Bit 1 LCDDIS

This bit enables the SEG/COM-pin discharge operations when "Display off" is selected.

1 (R/W): Enable SEG/COM-pin discharge operations

0 (R/W): Disable SEG/COM-pin discharge operations

Setting this bit to 1 configures the SEG/COM pins to output a low level when "Display off" is selected. Setting to 0 configures the SEG/COM pins to enter Hi-Z status when "Display off" is selected.

Bit 0 MODEN

This bit enables the LCD4B operations.

1 (R/W): Enable LCD4B operations

0 (R/W): Disable LCD4B operations

Setting this bit to 1 starts supplying the operating clock to LCD4B.

Note: If the LCD4CTL.MODEN bit is altered from 1 to 0 while the LCD panel is displaying, the LCD display is automatically turned off and the LCD4DSP.DSPC[1:0] bits are set to 0x0. Also the LCD voltage regulator is automatically turned off and the LCD4PWR.VCEN bit is set to 0.

LCD4B Timing Control Register 1

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4TIM1	15–12	–	0x0	–	R	–
	11–8	FRMCNT[3:0]	0x7	H0	R/W	
	7–2	–	0x00	–	R	
	1–0	LDUTY[1:0]	0x3	H0	R/W	

Bits 15–12 Reserved

Bits 11–8 FRMCNT[3:0]

These bits set the frame frequency. For more information, refer to "Frame Frequency."

Bits 7–2 Reserved

Bits 1–0 LDUTY[1:0]

These bits set the drive duty. For more information, refer to "Drive Duty Switching."

LCD4B Timing Control Register 2

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4TIM2	15–10	–	0x00	–	R	–
	9–8	BSTC[1:0]	0x1	H0	R/W	
	7–2	–	0x00	–	R	
	1–0	NLINE[1:0]	0x0	H0	R/W	

Bits 15–10 Reserved

Bits 9–8 BSTC[1:0]

These bits select the booster clock frequency for the LCD voltage booster.

Table 17.8.2 Booster Clock Frequency

LCD4TIM2.BSTC[1:0] bits	Booster clock frequency [Hz]
0x3	$f_{CLK\_LCD4B}/64$
0x2	$f_{CLK\_LCD4B}/32$
0x1	$f_{CLK\_LCD4B}/16$
0x0	$f_{CLK\_LCD4B}/4$

f_{CLK_LCD4B}: LCD4B operating clock frequency [Hz]

Bits 7–2 Reserved

Bits 1–0 NLINE[1:0]

These bits enable the n-line inverse AC drive function and set the number of inverse lines. For more information, refer to "n-Segment-Line Inverse AC Drive."

LCD4B Power Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4PWR	15	EXVCSEL	1	H0	R/W	–
	14–13	RESISEL[1:0]	0x0	H0	R/W	
	12–8	LC[4:0]	0x00	H0	R/W	
	7–5	–	0x0	–	R	
	4	BSTEN	0	H0	R/W	
	3	–	0	–	R	
	2	HVLD	0	H0	R/W	
	1	VCSEL	0	H0	R/W	
	0	VCEN	0	H0	R/W	

Bit 15 EXVCSEL

This bit selects the LCD drive power supply mode (external voltage application mode or internal generation mode).

1 (R/W): External voltage application mode

0 (R/W): Internal generation mode

Note: Be sure to avoid applying voltages to the V_{C1} to V_{C3} pins when the LCD4PWR.EXVCSEL bit is set to 0, as the LCD power supply pins are short-circuited to GND.

Bits 14–13 RESISEL[1:0]

These bits select the internal LCD voltage dividing resistor value.

Table 17.8.3 Internal LCD Voltage Divider Resistor Value Adjustment

LCD4PWR.RESISEL[1:0] bits	Internal resistor value
0x3	Large
0x2	↕
0x1	Small
0x0	Internal voltage dividing resistors are not used.

Bits 12–8 LC[4:0]

These bits set the LCD panel contrast.

Table 17.8.4 LCD Contrast Adjustment

LCD4PWR.LC[4:0] bits	Contrast
0x1f	High (dark)
0x1e	↑
:	:
0x01	↓
0x00	Low (light)

Bits 7–5 Reserved

Bit 4 BSTEN

This bit turns the LCD voltage booster on and off.
1 (R/W): LCD voltage booster on
0 (R/W): LCD voltage booster off

For more information, refer to "LCD Power Supply."

Bit 3 Reserved

Bit 2 HVLD

This bit sets the LCD voltage regulator into heavy load protection mode.
1 (R/W): Heavy load protection mode
0 (R/W): Normal mode

For more information, refer to "LCD Voltage Regulator Settings."

Bit 1 VCSEL

This bit sets the LCD voltage regulator output (reference voltage for boosting).
1 (R/W): V_{C2}
0 (R/W): V_{C1}

For more information, refer to "LCD Voltage Regulator Settings."

Note: The LCD4PWR.VCSEL bit must be set to 0 in an external voltage application mode.

Bit 0 VCEN

This bit turns the LCD voltage regulator on and off.
1 (R/W): LCD voltage regulator on
0 (R/W): LCD voltage regulator off

For more information, refer to "LCD Power Supply."

Note: Before setting the LCD4PWR.VCEN bit to 1, set the LCD4PWR.EXVCSEL bit to 0. Setting the LCD4PWR.EXVCSEL bit to 1 automatically clears the LCD4PWR.VCEN bit to 0.

LCD4B Display Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4DSP	15–8	–	0x00	–	R	–
	7	–	0	–	R	
	6	SEGREV	1	H0	R/W	
	5	COMREV	1	H0	R/W	
	4	DSPREV	1	H0	R/W	
	3	–	0	–	R	
	2	DSPAR	0	H0	R/W	
	1–0	DSPC[1:0]	0x0	H0	R/W	

Bits 15–7 Reserved

Bit 6 SEGREV

This bit selects the segment pin assignment direction.
1 (R/W): Normal assignment
0 (R/W): Inverse assignment

For more information, see Figures 17.6.3.1 to 17.6.3.4.

Bit 5 COMREV

This bit selects the common pin assignment direction.
1 (R/W): Normal assignment
0 (R/W): Inverse assignment

For more information, see Figures 17.6.3.1 to 17.6.3.4.

Bit 4 DSPREV

This bit controls black/white inversion on the LCD display.
1 (R/W): Normal display
0 (R/W): Inverted display

Bit 3 Reserved

Bit 2 DSPAR

This bit switches the display area in the display data RAM.
1 (R/W): Display area 1
0 (R/W): Display area 0

Bits 1–0 DSPC[1:0]

These bits control the LCD display on/off and select a display mode. For more information, refer to "Display On/Off."

LCD4B COM Pin Control Register 0

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4COMC0	15–8	–	0x00	–	R	–
	7–4	–	0x0	–	R	
	3	COM3DEN	1	H0	R/W	
	2	COM2DEN	1	H0	R/W	
	1	COM1DEN	1	H0	R/W	
	0	COM0DEN	1	H0	R/W	

Bits 15–4 Reserved

Bits 3–0 COM*x*DEN

These bits configure the partial drive of the COM*x* pins.
1 (R/W): Normal output
0 (R/W): Off waveform output

LCD4B Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4INTF	15–8	–	0x00	–	R	–
	7–1	–	0x00	–	R	
	0	FRMIF	0	H0	R/W	Cleared by writing 1.

Bits 15–1 Reserved

Bit 0 FRMIF

This bit indicates the frame interrupt cause occurrence status.
1 (R): Cause of interrupt occurred
0 (R): No cause of interrupt occurred
1 (W): Clear flag
0 (W): Ineffective

LCD4B Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4INTE	15–8	–	0x00	–	R	–
	7–1	–	0x00	–	R	
	0	FRMIE	0	H0	R/W	

Bits 15–1 Reserved

Bit 0 FRMIE

This bit enables the frame interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt

18 Multiplier/Divider (COPRO2)

18.1 Overview

COPRO2 is the coprocessor that provides multiplier/divider functions. The features of COPRO2 are listed below.

- Multiplication: Supports signed/unsigned multiplications.
  (16 bits × 16 bits = 32 bits)
  Can be executed in 1 cycle.
- Multiplication and accumulation (MAC): Supports signed/unsigned MAC operations with overflow detection function. (16 bits × 16 bits + 32 bits = 32 bits)
  Can be executed in 1 cycle.
- Division: Supports signed/unsigned divisions.
  (32 bits ÷ 32 bits = 32 bits with 32-bit reminder)
  Can be executed in 17 to 20 cycles.
  Overflow detection and division by zero processing are not supported.

Figure 18.1.1 shows the COPRO2 configuration.

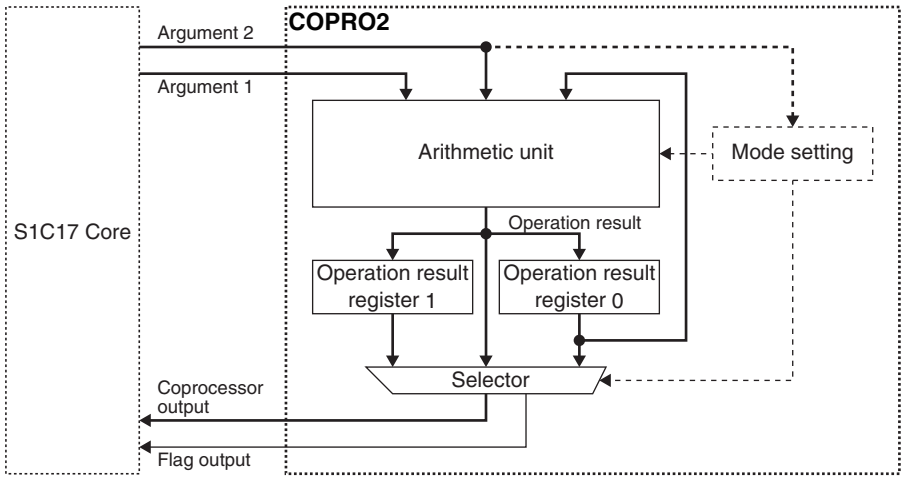


Figure 18.1.1 COPRO2 Configuration

18.2 Operation Mode and Output Mode

COPRO2 operates according to the operation mode specified by the application program. As listed in Table 18.2.1, COPRO2 supports 11 operations.
The multiplication, division and MAC results are 32-bit data, therefore, the S1C17 Core cannot read them in one access cycle. The output mode is provided to specify the high-order 16 bits or low-order 16 bits of the operation result register 0 or 1 to be read from COPRO2.
The operation and output modes can be specified with a 7-bit data by writing it to the mode setting register in COPRO2. Use a "`ld.cw`" instruction for this writing.

`ld.cw %rd,%rs` %rs[6:0] is written to the mode setting register. (%rd: not used)
`ld.cw %rd,`*`imm7`* *imm7*[6:0] is written to the mode setting register. (%rd: not used)

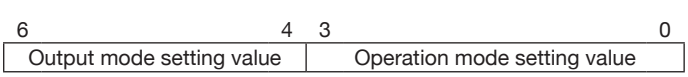


Figure 18.2.1 Mode Setting Register

Table 18.2.1 Mode Settings

Setting value (D[6:4])	Output mode	Setting value (D[3:0])	Operation mode
0x0	**16 low-order bits output mode 0** The low-order 16 bits of the operation result register 0 can be read as the coprocessor output.	0x0	**Initialize mode 0** Clears the operation result registers 0 and 1 to 0x0.
0x1	**16 high-order bits output mode 0** The high-order 16 bits of the operation result register 0 can be read as the coprocessor output.	0x1	**Initialize mode 1** Loads the 16-bit augend into the low-order 16 bits of the operation result register 0.
0x2	**16 low-order bits output mode 1** The low-order 16 bits of the operation result register 1 can be read as the coprocessor output.	0x2	**Initialize mode 2** Loads the 32-bit data into the operation result register 0.
0x3	**16 high-order bits output mode 1** The high-order 16 bits of the operation result register 1 can be read as the coprocessor output.	0x3	**Operation result read mode** Outputs the data in the operation result registers 0 and 1 without computation.
0x4–0x7	Reserved	0x4	**Unsigned multiplication mode** Performs unsigned multiplication.
		0x5	**Signed multiplication mode** Performs signed multiplication.
		0x6	**Unsigned MAC mode** Performs unsigned MAC operation.
		0x7	**Signed MAC mode** Performs signed MAC operation.
		0x8	**Unsigned division mode** Performs unsigned division.
		0x9	**Signed division mode** Performs signed division.
		0xa	**Initialize mode 3** Loads the 32-bit data into the operation result register 1.
		0xb–0xf	Reserved

18.3 Multiplication

The multiplication function performs "A (32 bits) = B (16 bits) × C (16 bits)."
The following shows a procedure to perform a multiplication:

1. Set the mode to 0x04 (unsigned multiplication, 16 low-order bits output mode 0) or 0x05 (signed multiplication, 16 low-order bits output mode 0).
2. Send the 16-bit multiplicand (B) and 16-bit multiplier (C) to COPRO2 using a "`ld.ca`" instruction.
3. Read the one-half result (16 low-order bits = A[15:0]) and the flag status.
4. Set the mode to 0x13 (operation result read, 16 high-order bits output mode 0).
5. Read another one-half result (16 high-order bits = A[31:16]).

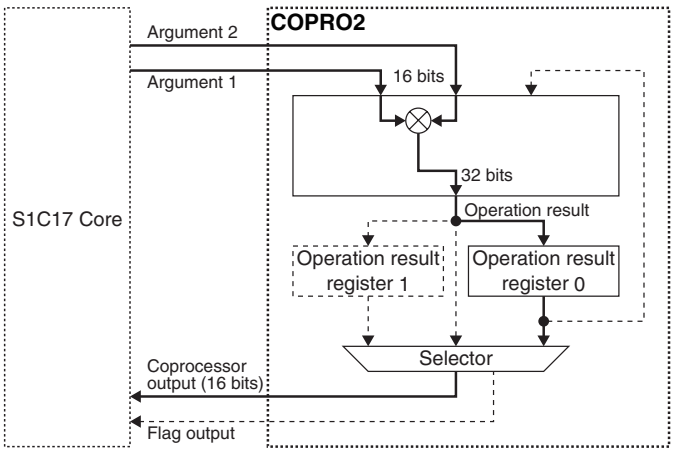


Figure 18.3.1 Data Path in Multiplication Mode

Table 18.3.1 Operation in Multiplication Mode

Mode setting value	Instruction	Operations	Flags	Remarks
0x04 or 0x05	`ld.ca %rd,%rs`	res0[31:0] ← %rd × %rs %rd ← res0[15:0]	psr (CVZN) ← 0b0000	The operation result register 0 keeps the operation result until it is rewritten by other operation.
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res0[31:0] ← %rd × *imm7/16* %rd ← res0[15:0]		
0x14 or 0x15	`ld.ca %rd,%rs`	res0[31:0] ← %rd × %rs %rd ← res0[31:16]		
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res0[31:0] ← %rd × *imm7/16* %rd ← res0[31:16]		

res0: operation result register 0

Example:

```
ld.cw %r0,0x04 ; Sets the mode (unsigned multiplication mode and 16 low-order bits output mode 0).
ld.ca %r0,%r1  ; Performs "res0[31:0] = %r0[15:0] × %r1[15:0]" and loads the 16 low-order bits of the
                 result to %r0.
ld.cw %r0,0x13 ; Sets the mode (operation result read mode and 16 high-order bits output mode 0).
ld.ca %r1,%r0  ; Loads the 16 high-order bits of the result to %r1.
```

18.4 Division

The division function performs "A (32 bits) = B (32 bits) ÷ C (32 bits), D (32 bits) = remainder."
The following shows a procedure to perform a division:

1. Set the mode to 0x02 (initialize mode 2).
2. Set the 32-bit dividend (B) to the operation result register 0 using a "`ld.cf`" instruction.
3. Set the mode to 0x08 (unsigned division, 16 low-order bits output mode 0) or 0x09 (signed division, 16 low-order bits output mode 0).
4. Send the 32-bit divisor (C) to COPRO2 using a "`ld.ca`" instruction.
5. Read the one-half result (16 low-order bits = A[15:0]) of the operation result register 0 (quotient) and the flag status.
6. Set the mode to 0x13 (operation result read, 16 high-order bits output mode 0).
7. Read another one-half result (16 high-order bits = A[31:16]) of the operation result register 0 (quotient).
8. Set the mode to 0x23 (operation result read, 16 low-order bits output mode 1).
9. Read the one-half result (16 low-order bits = D[15:0]) of the operation result register 1 (remainder).
10. Set the mode to 0x33 (operation result read, 16 high-order bits output mode 1).
11. Read another one-half result (16 high-order bits = D[31:16]) of the operation result register 1 (remainder).

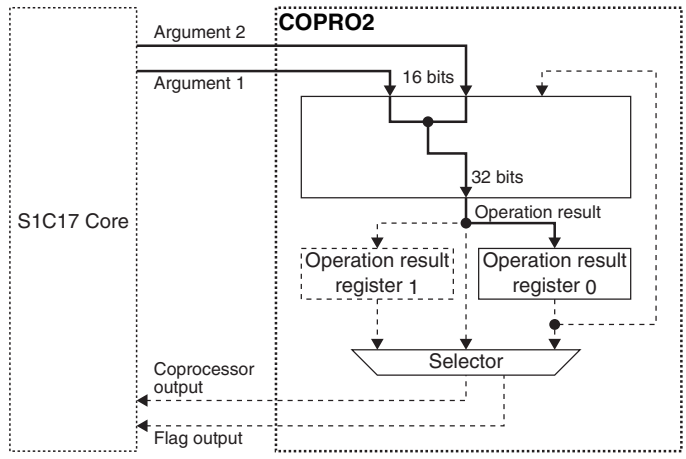


Figure 18.4.1 Data Path in Initialize Mode 2

Table 18.4.1 Initializing the Operation Result Register 0 (32 bits)

Mode setting value	Instruction	Operations	Remarks
0x02	`ld.cf %rd,%rs`	res0[31:16] ← %rd res0[15:0] ← %rs	
	(`ext` *imm9*) `ld.cf %rd,`*imm7*	res0[31:16] ← %rd res0[15:0] ← *imm7/16*	

res0: operation result register 0

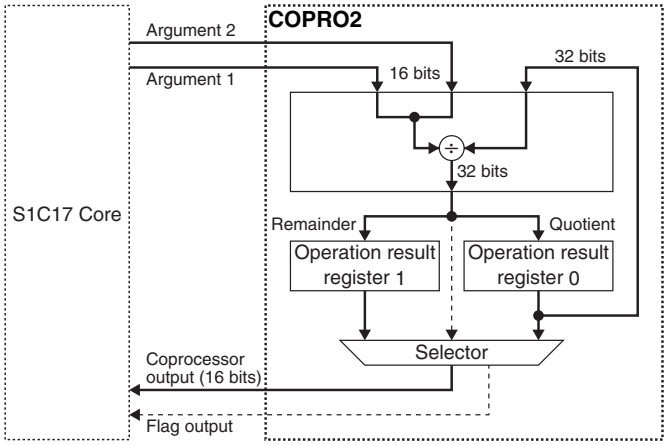


Figure 18.4.2 Data Path in Division Mode

Table 18.4.2 Operation in Division Mode

Mode setting value	Instruction	Operations	Flags	Remarks
0x08 or 0x09	`ld.ca %rd,%rs`	res0[31:0] ÷ {%rd, %rs} res0[31:0] ← Quotient res1[31:0] ← Remainder %rd ← res0[15:0] (Quotient)	psr (CVZN) ← 0b0000	The operation result registers 0 and 1 keep the operation results until they are rewritten by other operation. COPRO2 does not support 0 ÷ 0 division.
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res0[31:0] ÷ {%rd, *imm7/16*} res0[31:0] ← Quotient res1[31:0] ← Remainder %rd ← res0[15:0] (Quotient)		
0x18 or 0x19	`ld.ca %rd,%rs`	res0[31:0] ÷ {%rd, %rs} res0[31:0] ← Quotient res1[31:0] ← Remainder %rd ← res0[31:16] (Quotient)		
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res0[31:0] ÷ {%rd, *imm7/16*} res0[31:0] ← Quotient res1[31:0] ← Remainder %rd ← res0[31:16] (Quotient)		
0x28 or 0x29	`ld.ca %rd,%rs`	res0[31:0] ÷ {%rd, %rs} res0[31:0] ← Quotient res1[31:0] ← Remainder %rd ← res1[15:0] (Remainder)		
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res0[31:0] ÷ {%rd, *imm7/16*} res0[31:0] ← Quotient res1[31:0] ← Remainder %rd ← res1[15:0] (Remainder)		
0x38 or 0x39	`ld.ca %rd,%rs`	res0[31:0] ÷ {%rd, %rs} res0[31:0] ← Quotient res1[31:0] ← Remainder %rd ← res1[31:16] (Remainder)		
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res0[31:0] ÷ {%rd, *imm7/16*} res0[31:0] ← Quotient res1[31:0] ← Remainder %rd ← res1[31:16] (Remainder)		

res0: operation result register 0, res1: operation result register 1

Example:

```
ld.cw %r0,0x02 ; Sets the mode (initialize mode 2).
ld.cf %r0,%r1  ; Set the dividend {%r0, %r1} to the operation result register 0.
ld.cw %r0,0x08 ; Sets the mode (unsigned division mode and 16 low-order bits output mode 0).
ld.ca %r0,%r1  ; Performs "res0[31:0] (quotient), res1[31:0] (remainder) = res0[31:0] ÷ {%r0[15:0],
                 %r1[15:0]}" and loads the 16 low-order bits of the result (quotient) to %r0.
ld.ca %r1,%r0  ; Loads the 16 low-order bits of the result (quotient) to %r1.
ld.cw %r0,0x13 ; Sets the mode (operation result read mode and 16 high-order bits output mode 0).
ld.ca %r2,%r0  ; Loads the 16 high-order bits of the result (quotient) to %r2.
ld.cw %r0,0x23 ; Sets the mode (operation result read mode and 16 low-order bits output mode 1).
ld.ca %r3,%r0  ; Loads the 16 low-order bits of the result (remainder) to %r3.
ld.cw %r0,0x33 ; Sets the mode (operation result read mode and 16 high-order bits output mode 1).
ld.ca %r4,%r0  ; Loads the 16 high-order bits of the result (remainder) to %r4.
```

18.5 MAC

The MAC (multiplication and accumulation) function performs "A (32 bits) = B (16 bits) × C (16 bits) + A (32 bits)."
The following shows a procedure to perform a MAC operation:

1. Set the initial value (A) to the operation result register 0.
   - To clear the operation result registers (A = 0):
     Set the mode to 0x00 (initialize mode 0). (It is not necessary to send 0x00 to COPRO2 with another instruction.)
   - To load a 16-bit value to the operation result register 0:
     Set the operation mode to 0x01 (initialize mode 1) and then send the initial value (16 bits) to COPRO2 using a "`ld.cf`" instruction.
   - To load a 32-bit value to the operation result register 0:
     Set the operation mode to 0x02 (initialize mode 2) and then send the initial value (32 bits) to COPRO2 using a "`ld.cf`" instruction.
2. Set the mode to 0x06 (unsigned MAC, 16 low-order bits output mode 0) or 0x07 (signed MAC, 16 low-order bits output mode 0).
3. Repeat sending the 16-bit multiplicand (B) and 16-bit multiplier (C) to COPRO2 the number of times required using a "`ld.ca`" instruction.
4. Read the one-half result (16 low-order bits = A[15:0]) and the flag status.
5. Set the mode to 0x13 (operation result read, 16 high-order bits output mode).
6. Read another one-half result (16 high-order bits = A[31:16]).

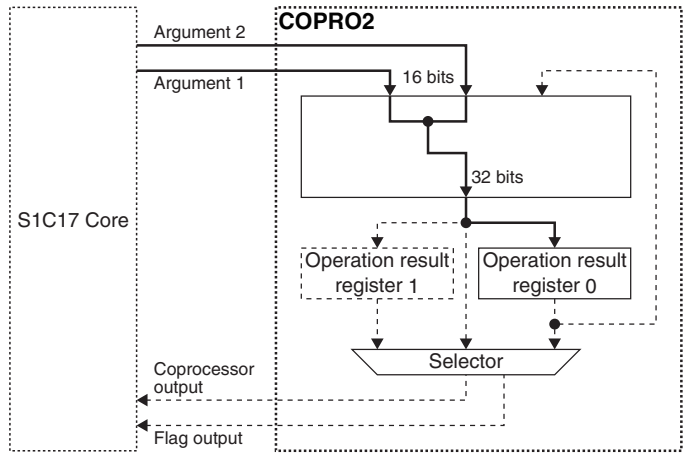


Figure 18.5.1 Data Path in Initialize Mode

Table 18.5.1 Initializing the Operation Result Register 0

Mode setting value	Instruction	Operations	Remarks
0x00	–	res0[31:0] ← 0x0 res1[31:0] ← 0x0	Setting the operating mode executes the initialization without sending data.
0x01	`ld.cf %rd,%rs`	res0[31:16] ← 0x0 res0[15:0] ← %rs	
	(`ext` *imm9*) `ld.cf %rd,`*imm7*	res0[31:16] ← 0x0 res0[15:0] ← *imm7/16*	
0x02	`ld.cf %rd,%rs`	res0[31:16] ← %rd res0[15:0] ← %rs	
	(`ext` *imm9*) `ld.cf %rd,`*imm7*	res0[31:16] ← %rd res0[15:0] ← *imm7/16*	

res0: operation result register 0, res1: operation result register 1

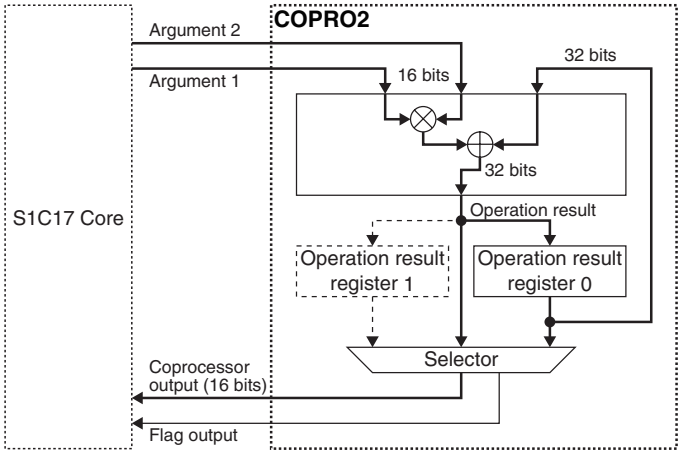


Figure 18.5.2 Data Path in MAC Mode

Table 18.5.2 Operation in MAC Mode

Mode setting value	Instruction	Operations	Flags	Remarks
0x06 or 0x07	`ld.ca %rd,%rs`	res0[31:0] ← %rd × %rs + res0[31:0] %rd ← res0[15:0]	psr (CVZN) ← 0b0100 if an overflow has occurred Otherwise psr (CVZN) ← 0b0000	The operation result register 0 keeps the operation result until it is rewritten by other operation. Overflow can be detected only in signed MAC mode (it does not occur in unsigned MAC mode).
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res0[31:0] ← %rd × *imm7/16* + res0[31:0] %rd ← res0[15:0]		
0x16 or 0x17	`ld.ca %rd,%rs`	res0[31:0] ← %rd × %rs + res0[31:0] %rd ← res0[31:16]		
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res0[31:0] ← %rd × *imm7/16* + res0[31:0] %rd ← res0[31:16]		

res0: operation result register 0

Example:

```
ld.cw %r0,0x00 ; Sets the mode (initialize mode 0) to clear the operation result register 0 to 0x0000.
ld.cw %r0,0x07 ; Sets the mode (signed MAC mode and 16 low-order bits output mode 0).
ld.ca %r0,%r1  ; Performs "res0[31:0] = %r0[15:0] × %r1[15:0] + res0[31:0]" and loads the 16 low-
                 order bits of the result to %r0.
ld.cw %r0,0x13 ; Sets the mode (operation result read mode and 16 high-order bits output mode 0).
ld.ca %r1,%r0  ; Loads the 16 high-order bits of the result to %r1.
```

Conditions to set the overflow (V) flag

An overflow occurs in a signed MAC operation and the overflow (V) flag is set to 1 when the signs of the multiplication result, operation result register value, and multiplication & accumulation result match the following conditions:

Table 18.5.3 Conditions to Set the Overflow (V) Flag

Mode setting value	Sign of multiplication result	Sign of operation result register value	Sign of multiplication & accumulation result
0x07	0 (positive)	0 (positive)	1 (negative)
0x07	1 (negative)	1 (negative)	0 (positive)

An overflow occurs when a MAC operation performs addition of positive values and a negative value results, or it performs addition of negative values and a positive value results. The coprocessor holds the operation result until the overflow (V) flag is cleared.

Conditions to clear the overflow (V) flag

The overflow (V) flag that has been set will be cleared when an overflow has not been occurred during execution of the "`ld.ca`" instruction for MAC operation or when the "`ld.ca`" or "`ld.cf`" instruction is executed in an operation mode other than operation result read mode.

18.6 Reading Operation Results

The "`ld.ca`" instruction cannot load a 32-bit operation result to a CPU register, so a multiplication, division or MAC operation returns the one-half (16 bits according to the output mode) result (A[15:0] or A[31:16]) and the flag status to the CPU registers. Another one-half should be read by setting COPRO2 into operation result read mode. The operation result register keeps the loaded operation result until it is rewritten by other operation.

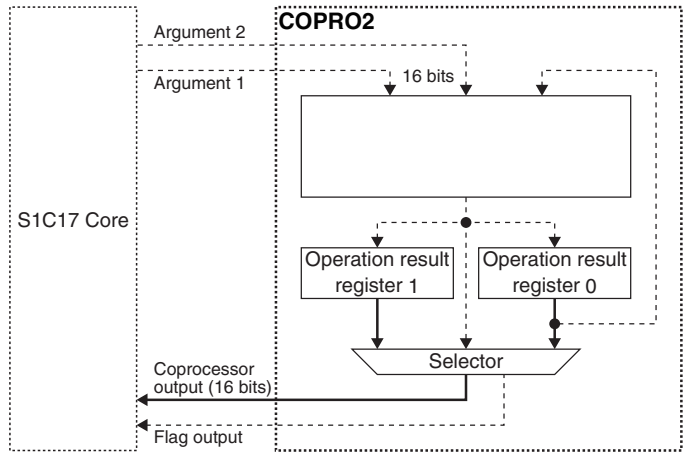


Figure 18.6.1 Data Path in Operation Result Read Mode

Table 18.6.1 Operation in Operation Result Read Mode

Mode setting value	Instruction	Operations	Flags	Remarks
0x03	`ld.ca %rd,%rs`	%rd ← res[15:0]	psr (CVZN) ← 0b0000	This operation mode does not affect the operation result registers 0 and 1.
	`ld.ca %rd,`*`imm7`*	%rd ← res[15:0]		
0x13	`ld.ca %rd,%rs`	%rd ← res[31:16]		
	`ld.ca %rd,`*`imm7`*	%rd ← res[31:16]		
0x23	`ld.ca %rd,%rs`	%rd ← res1[15:0]		
	`ld.ca %rd,`*`imm7`*	%rd ← res1[15:0]		
0x33	`ld.ca %rd,%rs`	%rd ← res1[31:16]		
	`ld.ca %rd,`*`imm7`*	%rd ← res1[31:16]		

res0: operation result register 0, res1: operation result register 1

19 Electrical Characteristics

19.1 Absolute Maximum Ratings

(V_{SS} = 0 V)

Item	Symbol	Condition		Rated value	Unit
Power supply voltage	V_{DD}			-0.3 to 7.0	V
Flash programming voltage	V_{PP}			-0.3 to 8.0	V
LCD power supply voltage	V_{C1}			-0.3 to 7.0	V
	V_{C2}			-0.3 to 7.0	V
	V_{C3}			-0.3 to 7.0	V
Input voltage	V_I	P00–02, P10–17, P20–27, P30–37, P40–47, VMIN, A00–05, A10–11, A20–23, FCP, FCM		-0.3 to 7.0	V
		P03–07, PD0–D1, #RESET		-0.3 to V_{DD} + 0.5	V
Output voltage	V_O	P00–07, P10–17, P20–27, P30–37, P40–47, PD0–D2		-0.3 to V_{DD} + 0.5	V
		A00–05, A10–11, A20–23, AGND, FCP, FCM		-0.3 to 7.0	V
High level output current	I_{OH}	1 pin	P00–07, P10–17, P20–27, P30–37, P40–47,	-10	mA
		Total of all pins	PD0–D2	-20	mA
Low level output current	I_{OL}	1 pin	P00–07, P10–17, P20–27, P30–37, P40–47,	10	mA
		Total of all pins	PD0–D2	20	mA
Operating temperature	Ta			-40 to 85	°C
Storage temperature	Tstg			-65 to 125	°C

19.2 Recommended Operating Conditions

(V_{SS} = 0 V) ∗1

Item	Symbol	Condition		Min.	Typ.	Max.	Unit
Power supply voltage	V_{DD}	For normal operation		2.1	–	3.6	V
		When analog circuits are operated		2.2	–	3.6	V
		For Flash programming	When V_{PP} is supplied externally	2.2	–	3.6	V
			When V_{PP} is generated internally	2.2	–	3.6	V
		For EEPROM programming	When V_{PP} is generated internally	2.2	–	3.6	V
Flash programming voltage	V_{PP}			7.3	7.5	7.7	V
LCD power supply voltage	V_{C1}	When an external voltage is applied		–	1.0	1.8	V
	V_{C2}	$V_{C1} \leq V_{C2} \leq V_{C3}$, $V_{C1} \leq V_{DD}$		–	2.0	3.6	V
	V_{C3}	∗2		–	3.0	5.4	V
OSC1 oscillator oscillation frequency	f_{OSC1}	Crystal oscillator		–	32.768	–	kHz
EXOSC external clock frequency	f_{EXOSC}	When supplied from an external oscillator		0.016	–	6.7	MHz
Bypass capacitor between V_{SS} and V_{DD}	C_{PW1}			–	3.3	–	µF
Capacitor between V_{SS} and V_{D1}	C_{PW2}			–	1	–	µF
Capacitors between V_{SS} and $V_{C1–3}$	$C_{LCD1–3}$	∗2		–	1	–	µF
Capacitor between C_{P1} and C_{P2}	C_{LCD4}	∗2		–	1	–	µF
Gate capacitor for OSC1 oscillator	C_{G1}	When the crystal oscillator is used ∗3		0	–	25	pF
Drain capacitor for OSC1 oscillator	C_{D1}	When the crystal oscillator is used ∗3		–	0	–	pF
DSIO pull-up resistor	R_{DBG}	∗4		–	10	–	kΩ
Resistor for DMM measurement	R_{VMIN}			–	10	–	MΩ
Resistor for DMM measurement	R_{A05}			–	1.11	–	MΩ
Resistor for DMM measurement	R_{A04}			–	10	–	MΩ
Resistor for DMM measurement	R_{A03}			–	101	–	kΩ
Capacitor for DMM measurement	C_{A02}			–	0.1	–	µF
Resistor for DMM measurement	R_{A01}			–	10	–	kΩ
Resistor for DMM measurement	R_{A00}			–	1	–	kΩ
Resistor for DMM measurement	R_{PTC}			–	200	–	Ω
Resistor for DMM measurement	R_{A23}			–	300	–	kΩ
Resistor for DMM measurement	R_{A22}			–	200	–	kΩ
Capacitor between FCN and FCP	C_{FIL}			–	27	–	nF
Resistor for DMM measurement	R_{A21}			–	10	–	kΩ
Resistor for DMM measurement	R_{A20}			–	10	–	kΩ
Resistor for DMM measurement	R_{I1}			–	100	–	Ω
Resistor for DMM measurement	R_{I2}			–	1	–	Ω
Resistor for DMM measurement	R_{I3}			–	0.01	–	Ω
Capacitor between V_{SS} and V_{PP}	C_{VPP}			–	0.1	–	µF
Capacitor between V_{SS} and A_{VDDH}	C_{AVDDH}			–	1	–	µF

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Capacitor between V_{SS} and V_{DD2}	C_{VDD2}		–	2.2	–	µF
Capacitor between C_A and C_B	C_{FLY}		–	0.1	–	µF
Capacitor between V_{SS} and V_{CP}	C_{VCP}		–	1	–	µF
Capacitor between V_{SS} and AGND	C_{AGND}		–	0.1	–	µF

∗1 The potential variation of the V_{SS} voltage should be suppressed to within ±0.3 V on the basis of the ground potential of the MCU mounting board while the Flash is being programmed, as it affects the Flash memory characteristics (programming count).
∗2 The V_{C1}–V_{C3} pins can be left open when the LCD driver is not used. Furthermore, the C_{P1}–C_{P2} pins are not used.
∗3 The component values should be determined after performing matching evaluation of the resonator mounted on the printed circuit board actually used.
∗4 R_{DBG} is not required when using the DSIO pin as a general-purpose I/O port.
∗5 The component values should be determined after evaluating operations using an actual mounting board.

19.3 Current Consumption

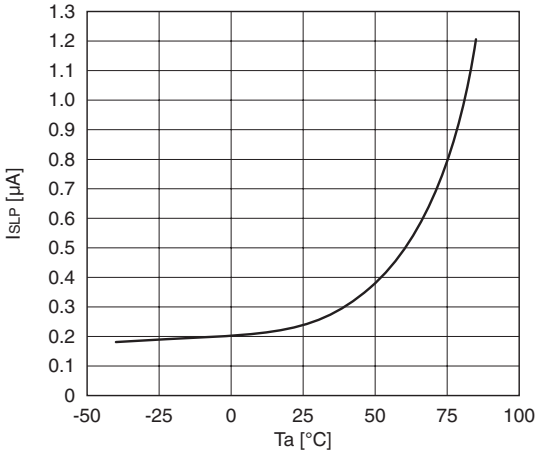
Unless otherwise specified: V_{DD} = 2.1 to 3.6 V, V_{SS} = 0 V, Ta = 25°C, EXOSC = OFF, PWGVD1CTL.REGMOD[1:0] bits = 0x0 (automatic mode), FLASHCWAIT.RDWAIT[1:0] bits = 0x1 (2 cycles)

Item	Symbol	Condition	Ta	Min.	Typ.	Max.	Unit
Current consumption in SLEEP mode	I_{SLP}	IOSC = OFF, OSC1 = OFF, OSC3 = OFF	25°C	–	0.24	1.20	µA
			85°C	–	1.70	12.0	µA
Current consumption in HALT mode	I_{HALT1}	IOSC = ON, OSC1 = 32.768 kHz∗1, OSC3 = OFF		–	40.0	75.0	µA
	I_{HALT2}	IOSC = OFF, OSC1 = 32.768 kHz∗1, OSC3 = OFF		–	0.9	2.0	µA
		IOSC = OFF, OSC1 = 32 kHz (internal oscillator)∗2, OSC3 = OFF		–	1.8	4.2	µA
	I_{HALT3}	IOSC = OFF, OSC1 = 32.768 kHz∗1, OSC3 = 3.2 MHz (internal oscillator)∗3		–	160	270	µA
		IOSC = OFF, OSC1 = 32 kHz (internal oscillator)∗2, OSC3 = 6.4 MHz (internal oscillator)∗4		–	200	340	µA
Current consumption in RUN mode	I_{RUN1}∗5	IOSC = ON, OSC1 = 32.768 kHz∗1, OSC3 = OFF, SYSCLK = IOSC		–	125	200	µA
		IOSC = ON, OSC1 = 32.768 kHz∗1, OSC3 = OFF, SYSCLK = IOSC, FLASHCWAIT.RDWAIT[1:0] bits = 0x0 (1 cycle)		–	145	240	µA
	I_{RUN2}∗5	IOSC = OFF, OSC1 = 32.768 kHz∗1, OSC3 = OFF, SYSCLK = OSC1		–	5.0	8.0	µA
		IOSC = OFF, OSC1 = 32 kHz (internal oscillator)∗2, OSC3 = OFF, SYSCLK = OSC1		–	6.0	10.5	µA
	I_{RUN3}∗5	IOSC = OFF, OSC1 = 32.768 kHz∗1, OSC3 = 3.2 MHz (internal oscillator)∗3, SYSCLK = OSC3, FLASHCWAIT.RDWAIT[1:0] bits = 0x1 (2 cycles)		–	540	850	µA
		IOSC = OFF, OSC1 = 32.768 kHz∗1, OSC3 = 6.4 MHz (internal oscillator)∗4, SYSCLK = OSC3, FLASHCWAIT.RDWAIT[1:0] bits = 0x2 (3 cycles)		–	825	1,300	µA

∗1 OSC1 oscillator: CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.INV1N[1:0] bits = 0x0, CLGOSC1.CGI1[2:0] bits = 0x0, CLGOSC1.OSDEN bit = 0, C_{G1} = C_{D1} = 0 pF, Crystal resonator = MC-146 (manufactured by Seiko Epson Corporation, R_1 = 65 kΩ (Max.), C_L = 7 pF)
∗2 OSC1 oscillator: CLGOSC1.OSC1SELCR bit = 1
∗3 OSC3 oscillator: CLGOSC3.OSC3FQ bit = 0
∗4 OSC3 oscillator: CLGOSC3.OSC3FQ bit = 1
∗5 The current consumption values were measured when a test program consisting of 60.5 % ALU instructions, 17 % branch instructions, 12 % RAM read instructions, and 10.5 % RAM write instructions was executed continuously in the Flash memory.

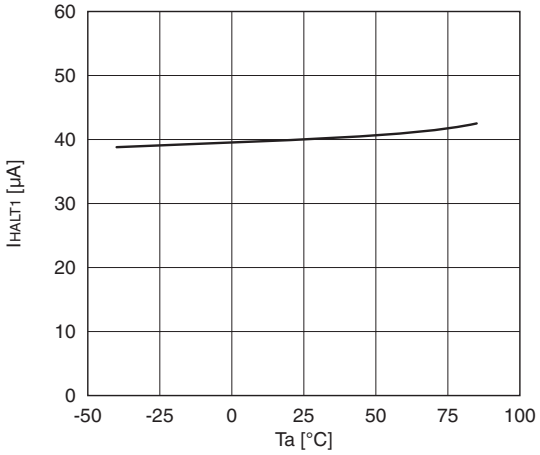
Current consumption-temperature characteristic in SLEEP mode

IOSC = OFF, OSC1 = OFF, OSC3 = OFF, V_{DD} = 3.6 V,
Typ. value



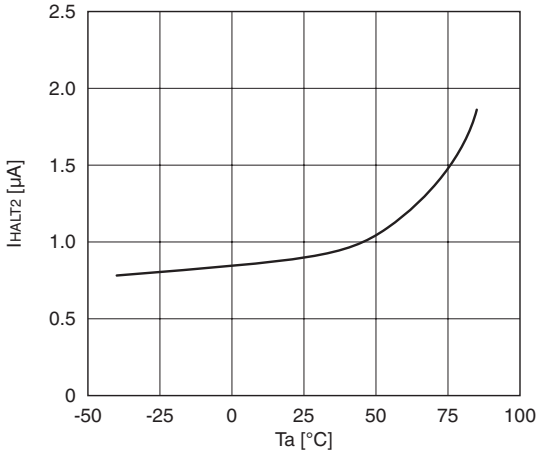
Current consumption-temperature characteristic in HALT mode (IOSC operation)

IOSC = ON, OSC1 = 32.768 kHz (crystal oscillator),
OSC3 = OFF, Typ. value



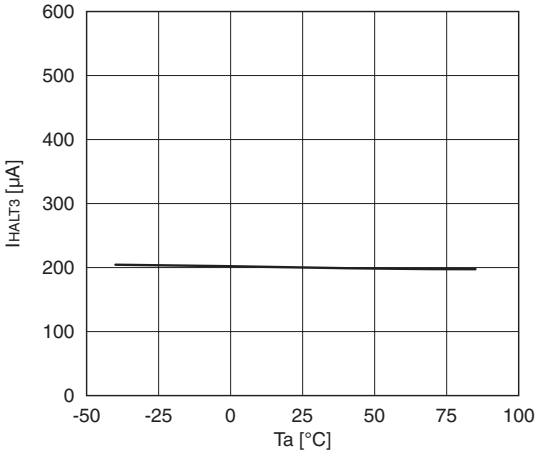
Current consumption-temperature characteristic in HALT mode (OSC1 operation)

IOSC = OFF, OSC1 = 32.768 kHz (crystal oscillator),
OSC3 = OFF, Typ. value



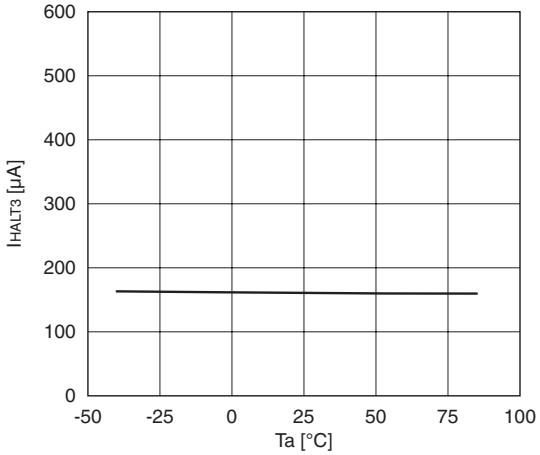
Current consumption-temperature characteristic in HALT mode (OSC3 operation)

IOSC = OFF, OSC1 = 32.768 kHz (crystal oscillator),
OSC3 = 6.4 MHz, Typ. value



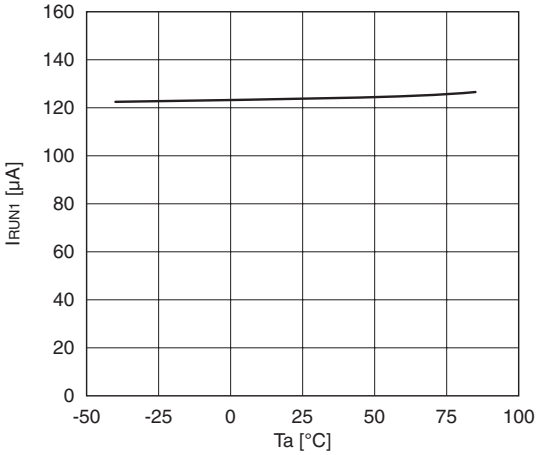
Current consumption-temperature characteristic in HALT mode (OSC3 operation)

IOSC = OFF, OSC1 = 32.768 kHz (crystal oscillator),
OSC3 = 3.2 MHz, Typ. value



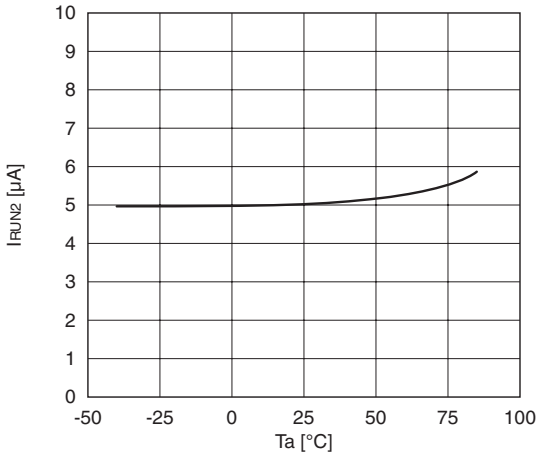
Current consumption-temperature characteristic in RUN mode (IOSC operation)

IOSC = ON, OSC1 = 32.768 kHz (crystal oscillator),
OSC3 = OFF, Typ. value



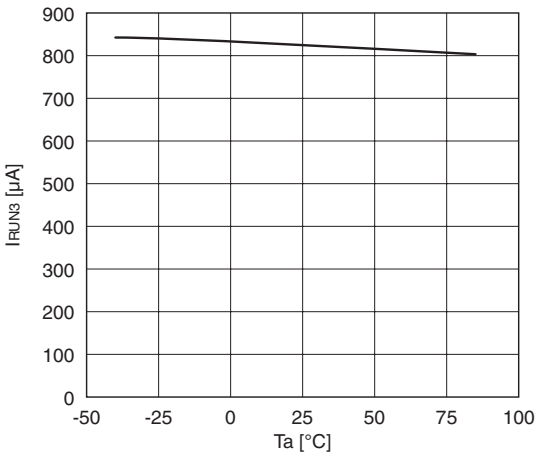
Current consumption-temperature characteristic in RUN mode (OSC1 operation)

IOSC = OFF, OSC1 = 32.768 kHz (crystal oscillator),
OSC3 = OFF, Typ. value



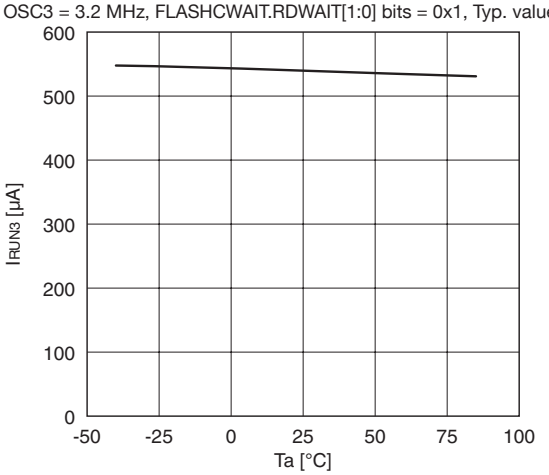
Current consumption-temperature characteristic in RUN mode (OSC3 operation)

IOSC = OFF, OSC1 = 32.768 kHz (crystal oscillator),
OSC3 = 6.4 MHz, FLASHCWAIT.RDWAIT[1:0] bits = 0x2, Typ. value



Current consumption-temperature characteristic in RUN mode (OSC3 operation)

IOSC = OFF, OSC1 = 32.768 kHz (crystal oscillator),
OSC3 = 3.2 MHz, FLASHCWAIT.RDWAIT[1:0] bits = 0x1, Typ. value

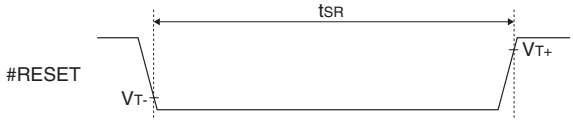


19.4 System Reset Controller (SRC) Characteristics

#RESET pin characteristics

Unless otherwise specified: V_{DD} = 2.1 to 3.6 V, V_{SS} = 0 V, Ta = -40 to 85°C

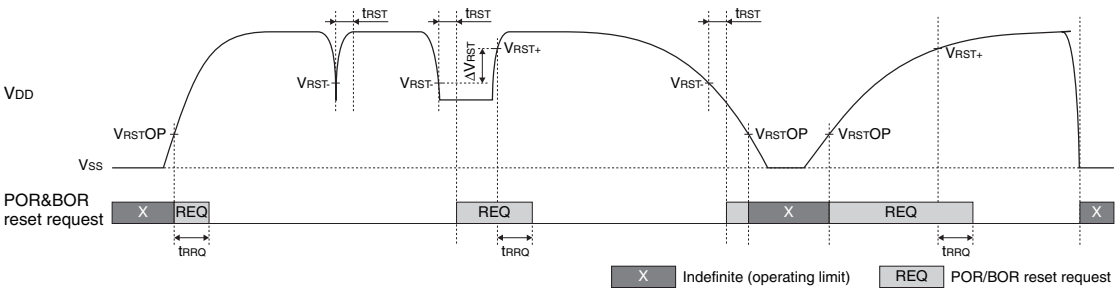
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level Schmitt input threshold voltage	V_{T+}		0.5 × V_{DD}	–	0.8 × V_{DD}	V
Low level Schmitt input threshold voltage	V_{T-}		0.2 × V_{DD}	–	0.5 × V_{DD}	V
Schmitt input hysteresis voltage	ΔV_T		180	–	–	mV
Input pull-up resistance	R_{IN}		100	200	500	kΩ
Pin capacitance	C_{IN}		–	–	15	pF
Reset Low pulse width	t_{SR}		25	–	–	µs



POR/BOR characteristics

Unless otherwise specified: V_{DD} = 2.1 to 3.6 V, V_{SS} = 0 V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
POR/BOR canceling voltage	V_{RST+}		1.15	–	1.75	V
POR/BOR detection voltage	V_{RST-}		1.05	–	1.60	V
POR/BOR hysteresis voltage	ΔV_{RST}		40	110	–	mV
POR/BOR detection response time	t_{RST}		–	–	500	µs
POR/BOR operating limit voltage	$V_{RST}OP$		–	0.5	0.95	V
POR/BOR reset request hold time	t_{RRQ}		0.01	–	4	ms



Note: When performing a power-on-reset again after the power is turned off, decrease the V_{DD} voltage to $V_{RST}OP$ or less.

Reset hold circuit characteristics

Unless otherwise specified: V_{DD} = 2.1 to 3.6 V, V_{SS} = 0 V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Reset hold time*1	t_{RSTR}		0.5	–	0.9	ms

*1 Time until the internal reset signal is negated after the reset request is canceled.

19.5 Clock Generator (CLG) Characteristics

Oscillator circuit characteristics including resonators change depending on conditions (board pattern, components used, etc.). Use these characteristic values as a reference and perform matching evaluation using the actual printed circuit board.

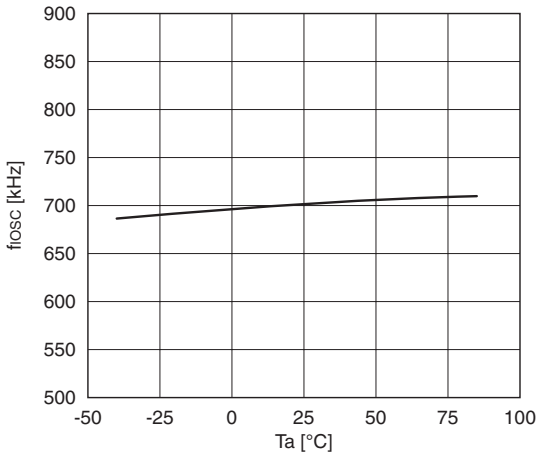
IOSC oscillator circuit characteristics

Unless otherwise specified: V_{DD} = 2.1 to 3.6 V, V_{SS} = 0 V, Ta = -40 to 85°C

Item	Symbol	Condition	Ta	Min.	Typ.	Max.	Unit
Oscillation start time	t_{staI}			–	–	3	µs
Oscillation frequency	f_{IOSC}		25°C	679	700	721	kHz
			-40 to 85°C	651	700	749	kHz

IOSC oscillation frequency-temperature characteristic

V_{DD} = 2.1 to 3.6 V, Typ. value



OSC1 oscillator circuit characteristics

Unless otherwise specified: V_{DD} = 2.1 to 3.6 V, V_{SS} = 0 V, Ta = 25°C

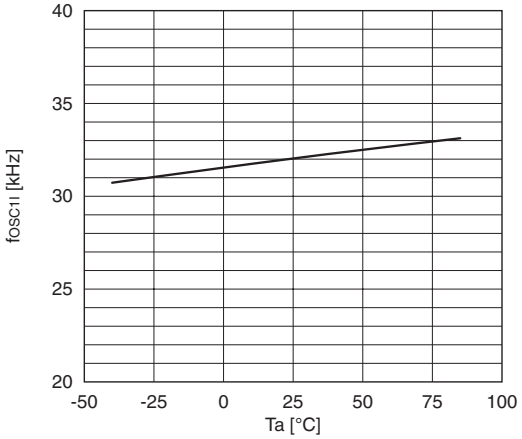
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Crystal oscillator oscillation start time*1	t_{sta1C}	CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.INV1N[1:0] bits = 0x1, CLGOSC1.INV1B[1:0] bits = 0x2, CLGOSC1.OSC1BUP bit = 1	–	–	3	s
Crystal oscillator internal gate capacitance	C_{GI1C}	CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.CGI1[2:0] bits = 0x0	–	12	–	pF
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.CGI1[2:0] bits = 0x1	–	14	–	pF
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.CGI1[2:0] bits = 0x2	–	16	–	pF
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.CGI1[2:0] bits = 0x3	–	18	–	pF
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.CGI1[2:0] bits = 0x4	–	19	–	pF
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.CGI1[2:0] bits = 0x5	–	21	–	pF
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.CGI1[2:0] bits = 0x6	–	23	–	pF
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.CGI1[2:0] bits = 0x7	–	24	–	pF
Crystal oscillator internal drain capacitance	C_{DI1C}	CLGOSC1.OSC1SELCR bit = 0,	–	6	–	pF

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Crystal oscillator oscillator circuit current - oscillation inverter drivability ratio *1	I_{OSC1C}	CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.INV1N/INV1B[1:0] bits = 0x0	–	70	–	%
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.INV1N/INV1B[1:0] bits = 0x1 (reference)	–	100	–	%
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.INV1N/INV1B[1:0] bits = 0x2	–	130	–	%
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.INV1N/INV1B[1:0] bits = 0x3	–	300	–	%
Crystal oscillator oscillation stop detector c	TBD	CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.OSDEN bit = 1	–	0.025	0.1	µA
Internal oscillator oscillation start time	t_{sta1I}	CLGOSC1.OSC1SELCR bit = 1	–	–	100	µs
Internal oscillator oscillation frequency	f_{OSC1I}	CLGOSC1.OSC1SELCR bit = 1	31.04	32	32.96	kHz

*1 CLGOSC1.CGI1[2:0] bits = 0x0, Crystal resonator = MC-146 (manufactured by Seiko Epson Corporation, R_1 = 65 kΩ (Max.), C_L = 7 pF)

OSC1 internal oscillation frequency-temperature characteristic

Typ. value



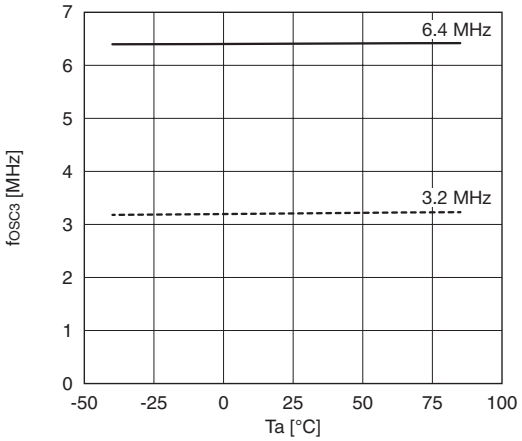
OSC3 oscillator circuit characteristics

Unless otherwise specified: V_{DD} = 2.1 to 3.6 V, V_{SS} = 0 V, Ta = 25°C

Item	Symbol	Condition	Ta	Min.	Typ.	Max.	Unit
Oscillation start time	t_{sta3}			–	–	3	µs
Oscillation frequency	f_{OSC3}	CLGOSC3.OSC3FQ bit = 0	-40 to 85°C	3.04	3.2	3.36	MHz
		CLGOSC3.OSC3FQ bit = 1	-40 to 85°C	6.08	6.4	6.72	MHz

OSC3 oscillation frequency-temperature characteristic

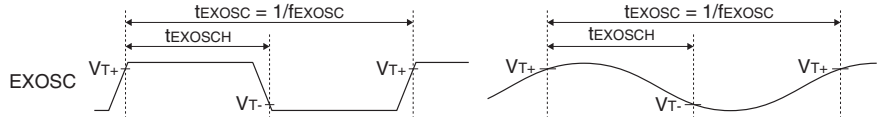
Typ. value



EXOSC external clock input characteristics

Unless otherwise specified: V_{DD} = 2.1 to 3.6 V, V_{SS} = 0 V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
EXOSC external clock duty ratio	t_{EXOSCD}	t_{EXOSCD} = t_{EXOSCH}/t_{EXOSC}	46	–	54	%
High level Schmitt input threshold voltage	V_{T+}		0.5 × V_{DD}	–	0.8 × V_{DD}	V
Low level Schmitt input threshold voltage	V_{T-}		0.2 × V_{DD}	–	0.5 × V_{DD}	V
Schmitt input hysteresis voltage	ΔV_T		180	–	–	mV



19.6 Flash Memory Characteristics

Unless otherwise specified: V_{DD} = 2.2 to 3.6 V, V_{SS} = 0 V *1, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Programming count *2	C_{FEP}	Programmed data is guaranteed to be retained for 10 years.	1,000	–	–	times

*1 The potential variation of the V_{SS} voltage should be suppressed to within ±0.3 V on the basis of the ground potential of the MCU mounting board while the Flash is being programmed, as it affects the Flash memory characteristics (programming count).

*2 Assumed that Erasing + Programming as count of 1. The count includes programming in the factory for shipment with ROM data programmed.

19.7 EEPROM Characteristics

Unless otherwise specified: V_{DD} = 2.2 to 3.6 V, V_{SS} = 0 V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Programming count	C_{EEP}	Programmed data is guaranteed to be retained for 10 years.	100,000	–	–	times
Programming time	t_{PRG}		–	0.2	15	ms
Programming current *1	I_{EEPROM}		–	3.2	10	mA
Programming power supply start-up time	t_{CPST}		*2	–	–	ms
Effective EEPROM reset pulse width	t_{XPOR}		500	–	–	ns

*1 The value is added to the current consumption in RUN mode.

*2 Determine the value referencing the equation below.

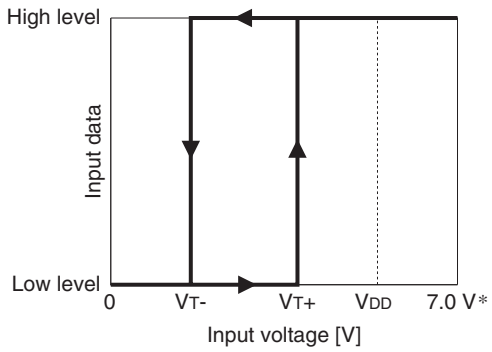
t_{CPST} = 37,500 × C_{VPP} + 15

t_{CPST}: Programming power supply start-up time [µs]、C_{VPP}: External smoothing capacitance [µF]

19.8 Input/Output Port (PPORT) Characteristics

Unless otherwise specified: V_{DD} = 2.1 to 3.6 V, V_{SS} = 0 V, Ta = -40 to 85°C

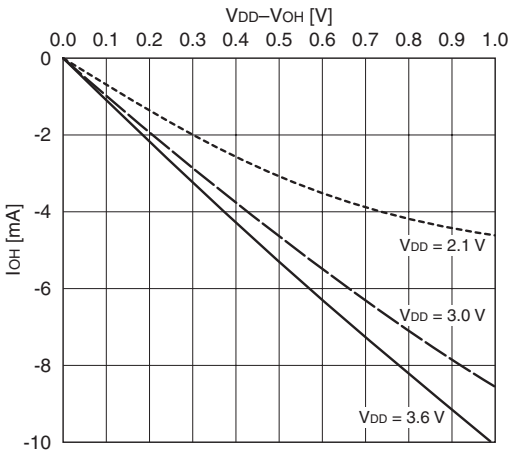
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level Schmitt input threshold voltage	V_{T+}	P00–07, P10–17, P20–27, P30–37, P40–47, PD0–D1	0.5 × V_{DD}	–	0.8 × V_{DD}	V
Low level Schmitt input threshold voltage	V_{T-}	P00–07, P10–17, P20–27, P30–37, P40–47, PD0–D1	0.2 × V_{DD}	–	0.5 × V_{DD}	V
Schmitt input hysteresis voltage	ΔV_T	P00–07, P10–17, P20–27, P30–37, P40–47, PD0–D1	180	–	–	mV
High level output current	I_{OH}	P00–07, P10–17, P20–27, P30–37, P40–47, PD0–D2, V_{OH} = 0.9 × V_{DD}	–	–	-0.5	mA
Low level output current	I_{OL}	P00–07, P10–17, P20–27, P30–37, P40–47, PD0–D2, V_{OL} = 0.1 × V_{DD}	0.5	–	–	mA
Leakage current	I_{LEAK}	P00–07, P10–17, P20–27, P30–37, P40–47, PD0–D2	-150	–	150	nA
Input pull-up resistance	R_{INU}	P00–07, P10–17, P20–27, P30–37, P40–47, PD0–D1	100	200	500	kΩ
Input pull-down resistance	R_{IND}	P00–07, P10–17, P20–27, P30–37, P40–47, PD0–D1	100	200	500	kΩ
Pin capacitance	C_{IN}	P00–07, P10–17, P20–27, P30–37, P40–47, PD0–D1	–	–	15	pF



(* For over voltage tolerant fail-safe type port)

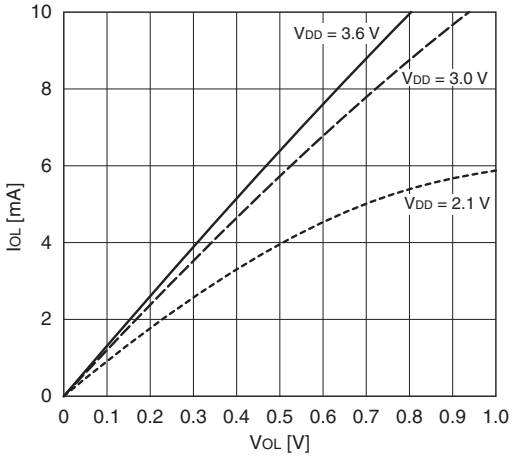
High-level output current characteristic

Ta = 85°C, Max. value



Low-level output current characteristic

Ta = 85°C, Min. value



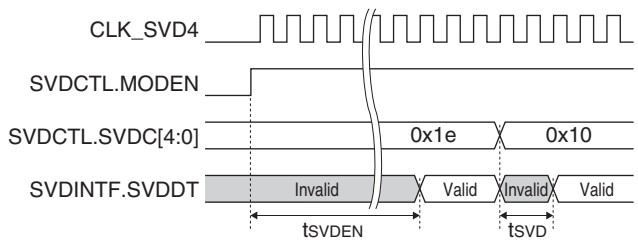
19.9 Supply Voltage Detector (SVD4) Characteristics

Unless otherwise specified: V_{DD} = 2.1 to 3.6 V, V_{SS} = 0 V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
EXSVD pin input voltage range	V_{EXSVD}		0	–	3.6	V
EXSVD input impedance	R_{EXSVD}	SVDCTL.SVDC[4:0] bits = 0x00	366	407	448	kΩ
		SVDCTL.SVDC[4:0] bits = 0x01	388	431	474	kΩ
		SVDCTL.SVDC[4:0] bits = 0x02	409	455	500	kΩ
		SVDCTL.SVDC[4:0] bits = 0x03	431	479	527	kΩ
		SVDCTL.SVDC[4:0] bits = 0x04	452	503	553	kΩ
		SVDCTL.SVDC[4:0] bits = 0x05	474	527	579	kΩ
		SVDCTL.SVDC[4:0] bits = 0x06	495	550	606	kΩ
		SVDCTL.SVDC[4:0] bits = 0x07	517	574	632	kΩ
		SVDCTL.SVDC[4:0] bits = 0x08	539	598	658	kΩ
		SVDCTL.SVDC[4:0] bits = 0x09	560	622	685	kΩ
		SVDCTL.SVDC[4:0] bits = 0x0a	582	646	711	kΩ
		SVDCTL.SVDC[4:0] bits = 0x0b	603	670	737	kΩ
		SVDCTL.SVDC[4:0] bits = 0x0c	625	694	763	kΩ
		SVDCTL.SVDC[4:0] bits = 0x0d	646	718	790	kΩ
		SVDCTL.SVDC[4:0] bits = 0x0e	668	742	816	kΩ
		SVDCTL.SVDC[4:0] bits = 0x0f	689	766	842	kΩ
		SVDCTL.SVDC[4:0] bits = 0x10	711	790	869	kΩ
		SVDCTL.SVDC[4:0] bits = 0x11	754	838	921	kΩ
		SVDCTL.SVDC[4:0] bits = 0x12	775	862	948	kΩ
		SVDCTL.SVDC[4:0] bits = 0x13–0x1f	–	–	–	kΩ

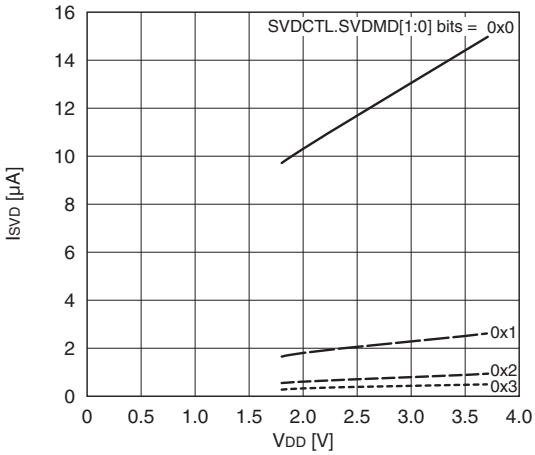
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
EXSVD detection voltage	V_{SVD_EXT}	SVDCTL.SVDC[4:0] bits = 0x00	1.65	1.7	1.75	V
		SVDCTL.SVDC[4:0] bits = 0x01	1.75	1.8	1.85	V
		SVDCTL.SVDC[4:0] bits = 0x02	1.85	1.9	1.95	V
		SVDCTL.SVDC[4:0] bits = 0x03	1.95	2.0	2.05	V
		SVDCTL.SVDC[4:0] bits = 0x04	2.04	2.1	2.16	V
		SVDCTL.SVDC[4:0] bits = 0x05	2.14	2.2	2.26	V
		SVDCTL.SVDC[4:0] bits = 0x06	2.24	2.3	2.36	V
		SVDCTL.SVDC[4:0] bits = 0x07	2.34	2.4	2.46	V
		SVDCTL.SVDC[4:0] bits = 0x08	2.43	2.5	2.57	V
		SVDCTL.SVDC[4:0] bits = 0x09	2.53	2.6	2.67	V
		SVDCTL.SVDC[4:0] bits = 0x0a	2.63	2.7	2.77	V
		SVDCTL.SVDC[4:0] bits = 0x0b	2.73	2.8	2.87	V
		SVDCTL.SVDC[4:0] bits = 0x0c	2.82	2.9	2.98	V
		SVDCTL.SVDC[4:0] bits = 0x0d	2.92	3.0	3.08	V
		SVDCTL.SVDC[4:0] bits = 0x0e	3.02	3.1	3.18	V
		SVDCTL.SVDC[4:0] bits = 0x0f	3.12	3.2	3.28	V
		SVDCTL.SVDC[4:0] bits = 0x10	3.21	3.3	3.39	V
		SVDCTL.SVDC[4:0] bits = 0x11	3.41	3.5	3.59	V
		SVDCTL.SVDC[4:0] bits = 0x12	3.51	3.6	3.69	V
		SVDCTL.SVDC[4:0] bits = 0x13–0x1f	–	–	–	V
SVD detection voltage	V_{SVD}	SVDCTL.SVDC[4:0] bits = 0x00	1.65	1.7	1.75	V
		SVDCTL.SVDC[4:0] bits = 0x01	1.75	1.8	1.85	V
		SVDCTL.SVDC[4:0] bits = 0x02	1.85	1.9	1.95	V
		SVDCTL.SVDC[4:0] bits = 0x03	1.95	2.0	2.05	V
		SVDCTL.SVDC[4:0] bits = 0x04	2.04	2.1	2.16	V
		SVDCTL.SVDC[4:0] bits = 0x05	2.14	2.2	2.26	V
		SVDCTL.SVDC[4:0] bits = 0x06	2.24	2.3	2.36	V
		SVDCTL.SVDC[4:0] bits = 0x07	2.34	2.4	2.46	V
		SVDCTL.SVDC[4:0] bits = 0x08	2.43	2.5	2.57	V
		SVDCTL.SVDC[4:0] bits = 0x09	2.53	2.6	2.67	V
		SVDCTL.SVDC[4:0] bits = 0x0a	2.63	2.7	2.77	V
		SVDCTL.SVDC[4:0] bits = 0x0b	2.73	2.8	2.87	V
		SVDCTL.SVDC[4:0] bits = 0x0c	2.82	2.9	2.98	V
		SVDCTL.SVDC[4:0] bits = 0x0d	2.92	3.0	3.08	V
		SVDCTL.SVDC[4:0] bits = 0x0e	3.02	3.1	3.18	V
		SVDCTL.SVDC[4:0] bits = 0x0f	3.12	3.2	3.28	V
		SVDCTL.SVDC[4:0] bits = 0x10	3.21	3.3	3.39	V
		SVDCTL.SVDC[4:0] bits = 0x11	3.41	3.5	3.59	V
		SVDCTL.SVDC[4:0] bits = 0x12	3.51	3.6	3.69	V
		SVDCTL.SVDC[4:0] bits = 0x13–0x1f	–	–	–	V
SVD circuit enable response time	t_{SVDEN}	*1	–	–	500	µs
SVD circuit response time	t_{SVD}		–	–	60	µs
SVD circuit current	I_{SVD}	SVDCTL.SVDMD[1:0] bits = 0x0, SVDCTL.SVDC[4:0] bits = 0x00, CLK_SVD4 = 32 kHz, Ta = 25°C	–	15	28	µA
		SVDCTL.SVDMD[1:0] bits = 0x1, SVDCTL.SVDC[4:0] bits = 0x00, CLK_SVD4 = 32 kHz, Ta = 25°C	–	2.6	6	µA
		SVDCTL.SVDMD[1:0] bits = 0x2, SVDCTL.SVDC[4:0] bits = 0x00, CLK_SVD4 = 32 kHz, Ta = 25°C	–	1.3	3	µA
		SVDCTL.SVDMD[1:0] bits = 0x3, SVDCTL.SVDC[4:0] bits = 0x00, CLK_SVD4 = 32 kHz, Ta = 25°C	–	0.7	1.7	µA

*1 If CLK_SVD4 is configured in the neighborhood of 32 kHz, the SVDINTF.SVDDT bit is masked during the t_{SVDEN} period and it retains the previous value.



SVD circuit current - power supply voltage characteristic

Ta = 25°C, SVDCTL.SVDC[4:0] bits = 0x00, CLK_SVD4 = 32 kHz, Typ. value



19.10 UART (UART3) Characteristics

Unless otherwise specified: V_{DD} = 2.1 to 3.6 V, V_{SS} = 0 V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Transfer baud rate	U_{BRT1}	Normal mode	150	–	921,600	bps
	U_{BRT2}	IrDA mode	150	–	115,200	bps

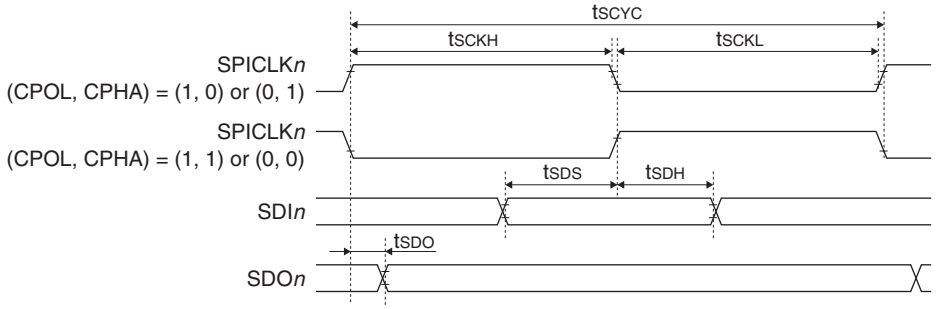
19.11 Synchronous Serial Interface (SPIA) Characteristics

Unless otherwise specified: V_{DD} = 2.1 to 3.6 V, V_{SS} = 0 V, Ta = -40 to 85°C

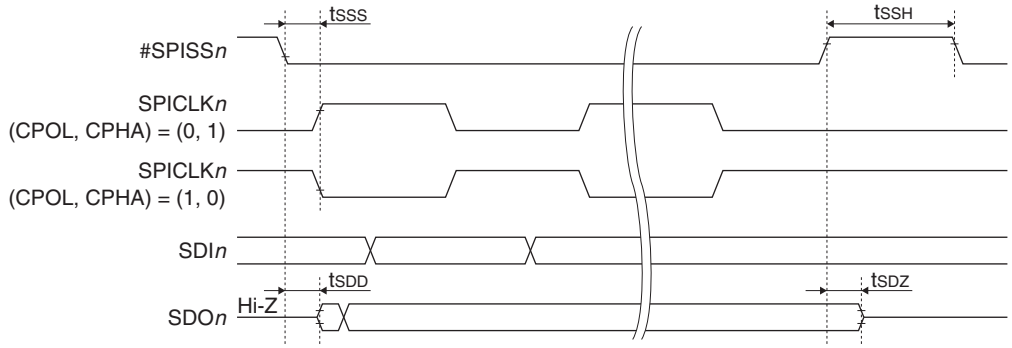
Item	Symbol	Condition	V_{DD}	Min.	Typ.	Max.	Unit
SPICLK*n* cycle time	t_{SCYC}		3.0 to 3.6 V	300	–	–	ns
			2.1 to 3.0 V	500	–	–	ns
SPICLK*n* High pulse width	t_{SCKH}		3.0 to 3.6 V	120	–	–	ns
			2.1 to 3.0 V	200	–	–	ns
SPICLK*n* Low pulse width	t_{SCKL}		3.0 to 3.6 V	120	–	–	ns
			2.1 to 3.0 V	200	–	–	ns
SDI*n* setup time	t_{SDS}		3.0 to 3.6 V	80	–	–	ns
			2.1 to 3.0 V	100	–	–	ns
SDI*n* hold time	t_{SDH}		3.0 to 3.6 V	40	–	–	ns
			2.1 to 3.0 V	50	–	–	ns
SDO*n* output delay time	t_{SDO}	C_L = 15 pF *1	3.0 to 3.6 V	–	–	100	ns
			2.1 to 3.0 V	–	–	120	ns
#SPISS*n* setup time	t_{SSS}			80	–	–	ns
#SPISS*n* High pulse width	t_{SSH}			100	–	–	ns
SDO*n* output start time	t_{SDD}	C_L = 15 pF *1		–	–	100	ns
SDO*n* output stop time	t_{SDZ}	C_L = 15 pF *1		–	–	100	ns

*1 C_L = Pin load

Master and slave modes



Slave mode

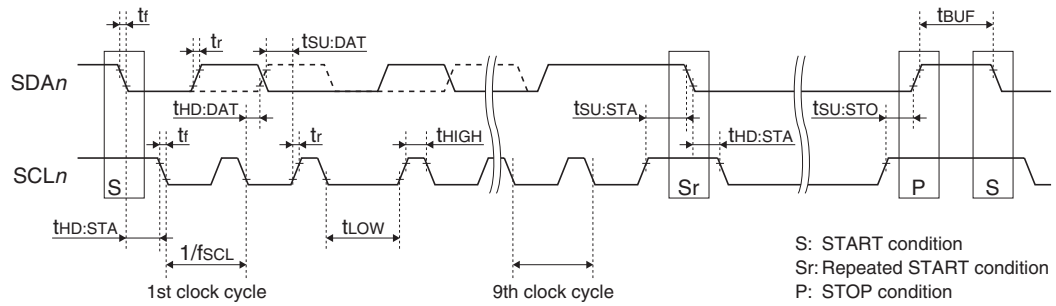


19.12 I²C (I2C) Characteristics

Unless otherwise specified: V_{DD} = 2.1 to 3.6 V, V_{SS} = 0 V, Ta = -40 to 85°C

Item	Symbol	Condition	Standard mode			Fast mode			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
SCL*n* frequency	f_{SCL}		0	–	100	0	–	400	kHz
Hold time (repeated) START condition *	$t_{HD:STA}$		4.0	–	–	0.6	–	–	µs
SCL*n* Low pulse width	t_{LOW}		4.7	–	–	1.3	–	–	µs
SCL*n* High pulse width	t_{HIGH}		4.0	–	–	0.6	–	–	µs
Repeated START condition setup time	$t_{SU:STA}$		4.7	–	–	0.6	–	–	µs
Data hold time	$t_{HD:DAT}$		0	–	–	0	–	–	µs
Data setup time	$t_{SU:DAT}$		250	–	–	100	–	–	ns
SDA*n*, SCL*n* rise time	t_r		–	–	1,000	–	–	300	ns
SDA*n*, SCL*n* fall time	t_f		–	–	300	–	–	300	ns
STOP condition setup time	$t_{SU:STO}$		4.0	–	–	0.6	–	–	µs
Bus free time	t_{BUF}		4.7	–	–	1.3	–	–	µs

* After this period, the first clock pulse is generated.



19.13 LCD Driver (LCD4B) Characteristics

The LCD driver characteristics varies depending on the panel load (panel size, drive duty, number of display pixels and display contents), so evaluate them by connecting to the actually used LCD panel.

Unless otherwise specified: V_{DD} = 2.1 to 3.6 V, V_{SS} = 0 V, Ta = 25°C, LCD4TIM2.BSTC[1:0] bits = 0x1 (Voltage booster clock = 2 kHz), No panel load

Item	Symbol	Condition		Min.	Typ.	Max.	Unit
LCD drive voltage (V_{C2} reference voltage) LCD4PWR.VCSEL bit = 1	V_{C1}	Connect 1 MΩ load resistor between V_{SS} and V_{C1}		0.323 × V_{C3} (Typ.)	–	0.344 × V_{C3} (Typ.)	V
	V_{C2}	Connect 1 MΩ load resistor between V_{SS} and V_{C2}		0.646 × V_{C3} (Typ.)	–	0.687 × V_{C3} (Typ.)	V
	V_{C3}	Connect 1 MΩ load resistor between V_{SS} and V_{C3}	LCD4PWR.LC[4:0] bits = 0x00	–	–	–	V
			LCD4PWR.LC[4:0] bits = 0x01	–	–	–	V
			LCD4PWR.LC[4:0] bits = 0x02	–	–	–	V
			LCD4PWR.LC[4:0] bits = 0x03	2.61	2.69	2.77	V
			LCD4PWR.LC[4:0] bits = 0x04	2.68	2.76	2.84	V
			LCD4PWR.LC[4:0] bits = 0x05	2.75	2.84	2.92	V
			LCD4PWR.LC[4:0] bits = 0x06	2.82	2.91	3.00	V
			LCD4PWR.LC[4:0] bits = 0x07	2.90	2.98	3.07	V
			LCD4PWR.LC[4:0] bits = 0x08	2.97	3.06	3.15	V
			LCD4PWR.LC[4:0] bits = 0x09	3.04	3.13	3.23	V
			LCD4PWR.LC[4:0] bits = 0x0a	3.11	3.21	3.30	V
			LCD4PWR.LC[4:0] bits = 0x0b	3.26	3.36	3.46	V
			LCD4PWR.LC[4:0] bits = 0x0c	3.40	3.51	3.61	V
			LCD4PWR.LC[4:0] bits = 0x0d	3.55	3.66	3.77	V
			LCD4PWR.LC[4:0] bits = 0x0e	3.69	3.81	3.92	V
			LCD4PWR.LC[4:0] bits = 0x0f	3.84	3.95	4.07	V
			LCD4PWR.LC[4:0] bits = 0x10	3.98	4.10	4.23	V
			LCD4PWR.LC[4:0] bits = 0x11	4.13	4.25	4.38	V
			LCD4PWR.LC[4:0] bits = 0x12	4.27	4.40	4.53	V
			LCD4PWR.LC[4:0] bits = 0x13	4.41	4.55	4.69	V
			LCD4PWR.LC[4:0] bits = 0x14	4.56	4.70	4.84	V
			LCD4PWR.LC[4:0] bits = 0x15	4.70	4.85	5.00	V
			LCD4PWR.LC[4:0] bits = 0x16	4.85	5.00	5.15	V
			LCD4PWR.LC[4:0] bits = 0x17	4.92	5.07	5.23	V
			LCD4PWR.LC[4:0] bits = 0x18	4.99	5.15	5.30	V
			LCD4PWR.LC[4:0] bits = 0x19	5.07	5.22	5.38	V
			LCD4PWR.LC[4:0] bits = 0x1a	5.14	5.30	5.46	V
			LCD4PWR.LC[4:0] bits = 0x1b	5.21	5.37	5.53	V
			LCD4PWR.LC[4:0] bits = 0x1c	5.28	5.45	5.61	V
			LCD4PWR.LC[4:0] bits = 0x1d	5.36	5.52	5.69	V
			LCD4PWR.LC[4:0] bits = 0x1e	5.43	5.60	5.76	V
			LCD4PWR.LC[4:0] bits = 0x1f	5.50	5.67	5.84	V
LCD drive voltage (V_{C1} reference voltage) LCD4PWR.VCSEL bit = 0	V_{C1}	Connect 1 MΩ load resistor between V_{SS} and V_{C1}		0.323 × V_{C3} (Typ.)	–	0.344 × V_{C3} (Typ.)	V
	V_{C2}	Connect 1 MΩ load resistor between V_{SS} and V_{C2}		0.646 × V_{C3} (Typ.)	–	0.687 × V_{C3} (Typ.)	V
	V_{C3}	Connect 1 MΩ load resistor between V_{SS} and V_{C3}	LCD4PWR.LC[4:0] bits = 0x00	–	–	–	V
			LCD4PWR.LC[4:0] bits = 0x01	–	–	–	V
			LCD4PWR.LC[4:0] bits = 0x02	–	–	–	V
			LCD4PWR.LC[4:0] bits = 0x03	2.61	2.69	2.77	V
			LCD4PWR.LC[4:0] bits = 0x04	2.68	2.76	2.84	V
			LCD4PWR.LC[4:0] bits = 0x05	2.75	2.84	2.92	V
			LCD4PWR.LC[4:0] bits = 0x06	2.82	2.91	3.00	V
			LCD4PWR.LC[4:0] bits = 0x07	2.90	2.98	3.07	V
			LCD4PWR.LC[4:0] bits = 0x08	2.97	3.06	3.15	V
			LCD4PWR.LC[4:0] bits = 0x09	3.04	3.13	3.23	V
			LCD4PWR.LC[4:0] bits = 0x0a	3.11	3.21	3.30	V
			LCD4PWR.LC[4:0] bits = 0x0b	3.26	3.36	3.46	V
			LCD4PWR.LC[4:0] bits = 0x0c	3.40	3.51	3.61	V
			LCD4PWR.LC[4:0] bits = 0x0d	3.55	3.66	3.77	V
			LCD4PWR.LC[4:0] bits = 0x0e	3.69	3.81	3.92	V
			LCD4PWR.LC[4:0] bits = 0x0f	3.84	3.95	4.07	V
			LCD4PWR.LC[4:0] bits = 0x10	3.98	4.10	4.23	V
			LCD4PWR.LC[4:0] bits = 0x11	4.13	4.25	4.38	V
			LCD4PWR.LC[4:0] bits = 0x12	4.27	4.40	4.53	V
			LCD4PWR.LC[4:0] bits = 0x13	4.41	4.55	4.69	V
			LCD4PWR.LC[4:0] bits = 0x14	4.56	4.70	4.84	V

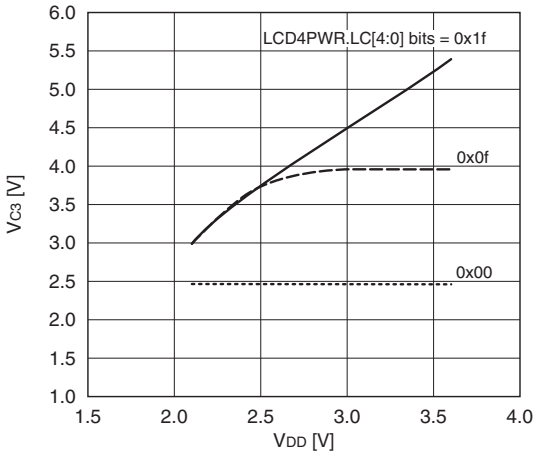
Item	Symbol	Condition		Min.	Typ.	Max.	Unit
LCD drive voltage (V_{C1} reference voltage) LCD4PWR.VCSEL bit = 0	V_{C3}		LCD4PWR.LC[4:0] bits = 0x15	4.70	4.85	5.00	V
			LCD4PWR.LC[4:0] bits = 0x16	4.85	5.00	5.15	V
		Connect 1 MΩ load resistor between V_{SS} and V_{C3}	LCD4PWR.LC[4:0] bits = 0x17	4.92	5.07	5.23	V
			LCD4PWR.LC[4:0] bits = 0x18	4.99	5.15	5.30	V
			LCD4PWR.LC[4:0] bits = 0x19	5.07	5.22	5.38	V
			LCD4PWR.LC[4:0] bits = 0x1a	5.14	5.30	5.46	V
			LCD4PWR.LC[4:0] bits = 0x1b	5.21	5.37	5.53	V
			LCD4PWR.LC[4:0] bits = 0x1c	5.28	5.45	5.61	V
			LCD4PWR.LC[4:0] bits = 0x1d	5.36	5.52	5.69	V
			LCD4PWR.LC[4:0] bits = 0x1e	5.43	5.60	5.76	V
			LCD4PWR.LC[4:0] bits = 0x1f	5.50	5.67	5.84	V
Segment/Common output current	I_{SEGH}	SEG*xx*, COM*y* V_{SEGH} = V_{C3}/V_{C2}/V_{C1} - 0.1 V, Ta = -40 to 85°C		–	–	-10	µA
	I_{SEGL}	SEG*xx*, COM*y* V_{SEGL} = V_{SS}/V_{C2}/V_{C1} + 0.1 V, Ta = -40 to 85°C		10	–	–	µA
LCD circuit current (V_{C2} reference voltage)	I_{LCD2}	LCD4DSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD4PWR.VCSEL bit = 1 *1 *2		–	2.2	5.3	µA
		LCD4DSP.DSPC[1:0] bits = 0x2 (all on), LCD4PWR.VCSEL bit = 1 *1 *2		–	1	2.3	µA
LCD circuit current (V_{C1} reference voltage)	I_{LCD1}	LCD4DSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD4PWR.VCSEL bit = 0 *1 *2		–	4.2	10	µA
		LCD4DSP.DSPC[1:0] bits = 0x2 (all on), LCD4PWR.VCSEL bit = 0 *1 *2		–	1.8	4.1	µA
LCD circuit current in heavy load protection mode (V_{C2} reference voltage)	I_{LCD2H}	LCD4DSP.DSPC[1:0] bits = 0x2 (all on), LCD4PWR.VCSEL bit = 1, LCD4PWR.HVLD bit = 1 *1 *2		–	16	33	µA
LCD circuit current in heavy load protection mode (V_{C1} reference voltage)	I_{LCD1H}	LCD4DSP.DSPC[1:0] bits = 0x2 (all on), LCD4PWR.VCSEL bit = 0, LCD4PWR.HVLD bit = 1 *1 *2		–	9	19	µA
LCD circuit current (when internal voltage dividing resistors are used)	I_{LCDR1}	LCD4DSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD4PWR.RESISEL[1:0] bits = 0x1 *2		–	23	48	µA
		LCD4DSP.DSPC[1:0] bits = 0x2 (all on), LCD4PWR.RESISEL[1:0] bits = 0x1 *2		–	22	46	µA
	I_{LCDR2}	LCD4DSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD4PWR.RESISEL[1:0] bits = 0x2 *2		–	9	20	µA
		LCD4DSP.DSPC[1:0] bits = 0x2 (all on), LCD4PWR.RESISEL[1:0] bits = 0x2 *2		–	8	18	µA
	I_{LCDR3}	LCD4DSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD4PWR.RESISEL[1:0] bits = 0x3 *2		–	5	12	µA
		LCD4DSP.DSPC[1:0] bits = 0x2 (all on), LCD4PWR.RESISEL[1:0] bits = 0x3 *2		–	4	10	µA

*1 Other LCD driver settings: LCD4PWR.LC[4:0] bits = 0x1f, CLK_LCD4B = 32 kHz, LCD4TIM1.FRMCNT[4:0] bits = 0x03 (frame frequency = 64 Hz)

*2 The value is added to the current consumption in HALT/RUN mode. Current consumption increases according to the display contents and panel load.

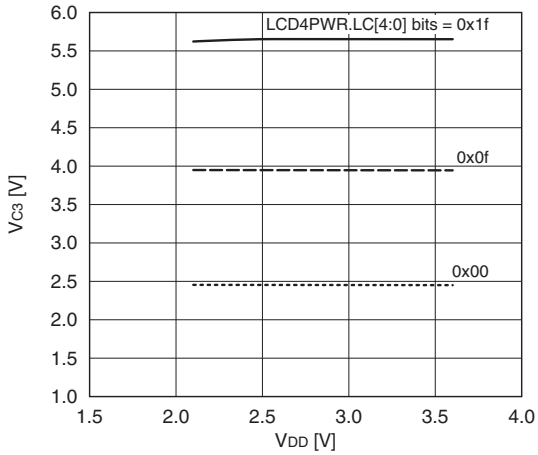
LCD drive voltage-supply voltage characteristic (V_{C2} reference voltage)

Ta = 25°C, Typ. value, when a 1 MΩ load resistor is connected between V_{SS} and V_{C3} (no panel load)



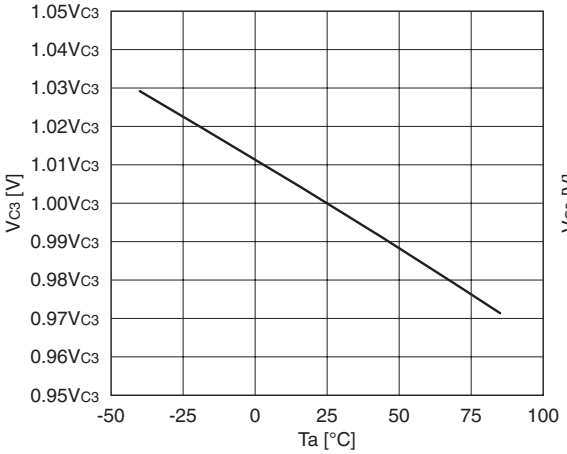
LCD drive voltage-supply voltage characteristic (V_{C1} reference voltage)

Ta = 25°C, Typ. value, when a 1 MΩ load resistor is connected between V_{SS} and V_{C3} (no panel load)



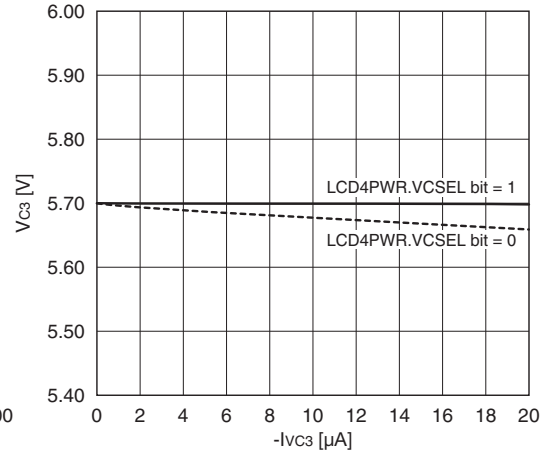
LCD drive voltage-temperature characteristic (V_{C1}/V_{C2} reference voltage)

Typ. value



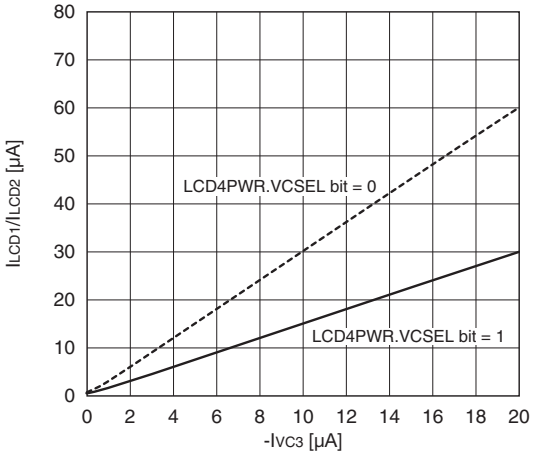
LCD drive voltage-load characteristic

V_{DD} = 3.6 V, Ta = 25°C, Typ. value, LCD4PWR.LC[4:0] bits = 0x1f, when a load is connected to the V_{C3} pin only



LCD circuit current-load characteristic

V_{DD} = 3.6 V, Ta = 25°C, Typ. value, LCD4PWR.LC[4:0] bits = 0x1f, when a load is connected to the V_{C3} pin only

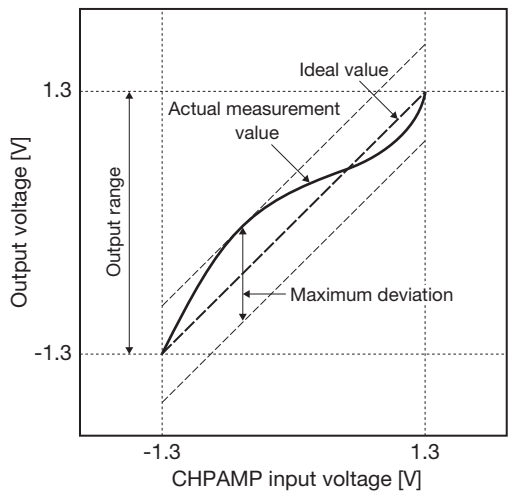


19.14 DMM Controller (DSADC16) Characteristics

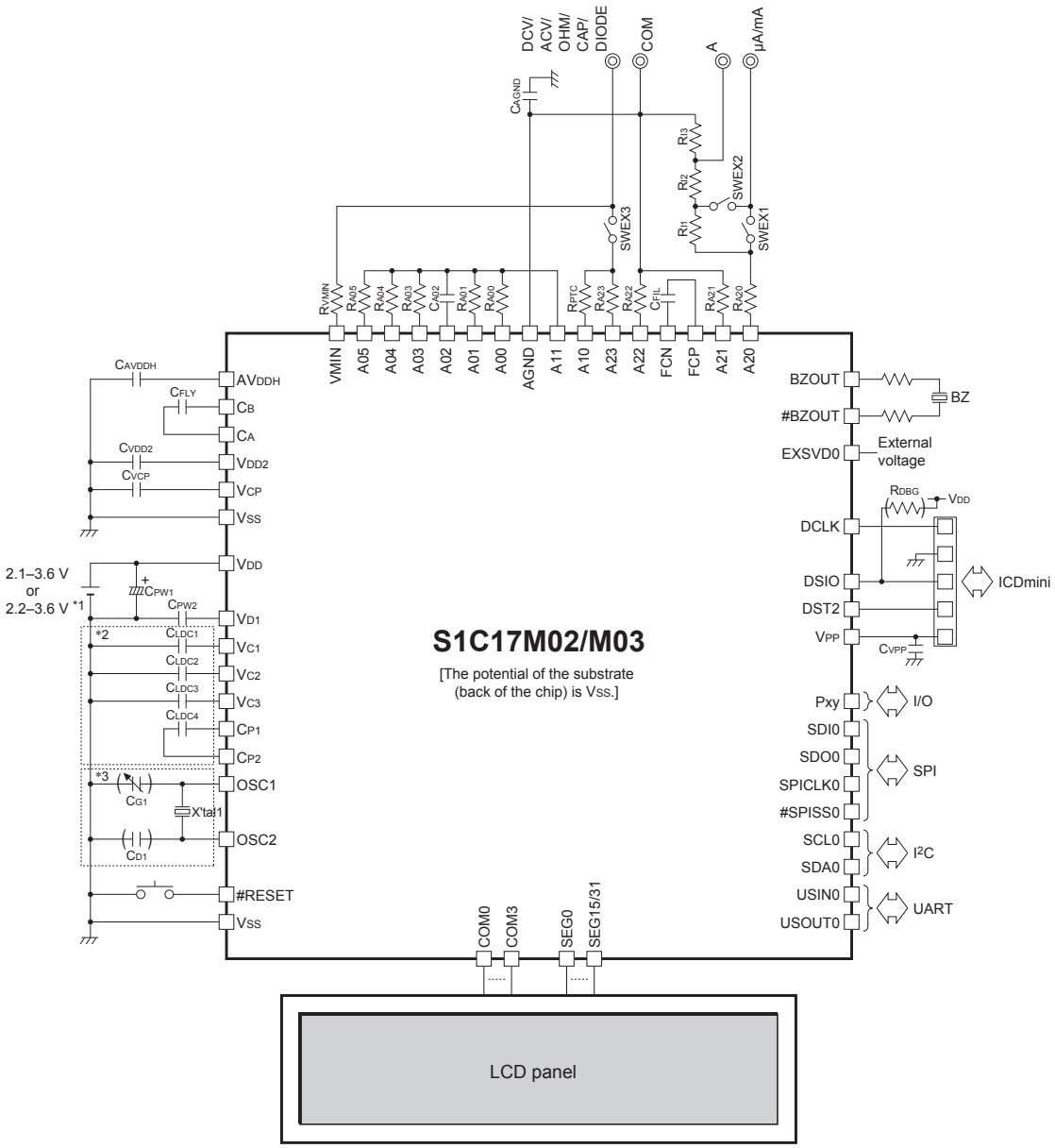
Unless otherwise specified: V_{DD} = 2.2 to 3.6 V, V_{SS} = 0 V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Operating frequency	f_{CLK}		–	800	–	kHz
Modulator frequency	f_{MOD}		–	400	–	kHz
Sampling frequency	f_{SMP}		24	–	25,000	Hz
Linearity error *1	LIN_{ERR}	CHPAMP input voltage range = ±1.3 V CHP_SET_CHP[2:0] bits = 0x0 (2 cycles) CHP_SET_GAIN[2:0] bits = 0x2 (1x) CHP_SET_BIAS[2:0] bits = 0x3 (0.25x) CHP_SET_BIAS2[2:0] bits = 0x3 (0.25x)	–	±0.05	±0.13	%
AV_{DDH} voltage	V_{AVDDH}		–	3.6	–	V
AGND voltage	V_{AGND18}		–	1.8	–	V
	V_{AGND12}		–	1.2	–	V
	V_{AGND06}		–	0.6	–	V
	V_{AGND02}		–	0.2	–	V
V_{ACM} voltage	V_{ACM18}		–	1.8	–	V
V_{REFP} voltage	V_{REFP27}		–	2.7	–	V
	V_{REFP24}		–	2.4	–	V
V_{REFN} voltage	V_{REFN09}		–	0.9	–	V
V_{REF} voltage	V_{REF34}		–	3.4	–	V
	V_{REF32}		–	3.2	–	V
	V_{REF28}		–	2.8	–	V
	V_{REF24}		–	2.4	–	V
	V_{REF08}		–	0.8	–	V
Measurement switch ON resistance	R_{ASW16}	PS0–1	–	8	–	Ω
	R_{PSW20}	S_{AVDDH}	–	4	–	Ω
	R_{ASW40}	DS0–1	–	16	–	Ω
	R_{ASW80}	PS2–5, DS2–5, S_{MODE4}, S_{MODE5}, AMPO1, S_{AGND}, S_{DIO}	–	24	–	Ω
	R_{NSW80}	S_{VSSA}	–	16	–	Ω
	R_{ASW400}	FS0–6, SS0–6	–	54	–	Ω
	$R_{ASW3600}$	AMPM1–2, AMPO1	–	1,511	–	Ω
Filter resistance	R_{F0}		–	104	–	kΩ
	R_{F1}		–	14	–	kΩ
	R_{F2}		–	4	–	kΩ
Comparator detection voltage accuracy	V_{DET}	DMM_SET_CMPRH[3:0] bits = 0x8 DMM_SET_CMPRL[3:0] bits = 0xb	-20	–	20	mV

*1 This value shows the ratio of the maximum deviation from the ideal value to the output range when a voltage is directly input to CHPAMP.
The output voltage was calculated from an expression regarding the relation between an input voltage and the output digital value (two-point correction).



20 Basic External Connection Diagram



∗1: For Flash/EEPROM programming or for analog circuit operations
∗2: When the internal LCD power supply is used
∗3: When OSC1 crystal oscillator is selected
( ): Do not mount components if unnecessary.

Sample external components

Symbol	Name	Recommended components
X'tal1	32 kHz crystal resonator	MC-146 (R_1 = 65 kΩ (Max.), C_L = 7 pF) manufactured by Seiko Epson Corporation
C_{G1}	OSC1 gate capacitor	Trimmer capacitor or ceramic capacitor
C_{D1}	OSC1 drain capacitor	Ceramic capacitor
C_{PW1}	Bypass capacitor between V_{SS} and V_{DD}	Ceramic capacitor or electrolytic capacitor
C_{PW2}	Capacitor between V_{SS} and V_{D1}	Ceramic capacitor
$C_{LCD1–3}$	Capacitors between V_{SS} and $V_{C1–3}$	Ceramic capacitor
C_{LCD4}	Capacitor between C_{P1} and C_{P2}	Ceramic capacitor
BZ	Piezoelectric buzzer	PKLCS1212E4001-R1 manufactured by Murata Manufacturing Co., Ltd.
R_{DBG}	DSIO pull-up resistor	Thick film chip resistor
C_{VPP}	Capacitor between V_{SS} and V_{PP}	Ceramic capacitor
R_{VMIN}	Resistor for DMM measurement	Thick film chip resistor
R_{A05}	Resistor for DMM measurement	Thick film chip resistor
R_{A04}	Resistor for DMM measurement	Thick film chip resistor
R_{A03}	Resistor for DMM measurement	Thick film chip resistor
C_{A02}	Capacitor for DMM measurement	Ceramic capacitor
R_{A01}	Resistor for DMM measurement	Thick film chip resistor
R_{A00}	Resistor for DMM measurement	Thick film chip resistor
R_{PTC}	Resistor for DMM measurement	PTC thermistor
R_{A23}	Resistor for DMM measurement	Thick film chip resistor
R_{A22}	Resistor for DMM measurement	Thick film chip resistor
C_{FIL}	Capacitor between FCN and FCP	Ceramic capacitor
R_{A21}	Resistor for DMM measurement	Thick film chip resistor
R_{A20}	Resistor for DMM measurement	Thick film chip resistor
R_{I1}	Resistor for DMM measurement	Thick film chip resistor
R_{I2}	Resistor for DMM measurement	Thick film chip resistor
R_{I3}	Resistor for DMM measurement	Thick film chip resistor
SWEX1	Measurement mode select switch	General-purpose switch
SWEX2	Measurement mode select switch	General-purpose switch
SWEX3	Measurement mode select switch	General-purpose switch
C_{AVDDH}	Capacitor between V_{SS} and AV_{DDH}	Ceramic capacitor
C_{FLY}	Capacitor between C_A and C_B	Ceramic capacitor
C_{VDD2}	Capacitor between V_{SS} and V_{DD2}	Ceramic capacitor
C_{VCP}	Capacitor between V_{SS} and V_{CP}	Ceramic capacitor
C_{AGND}	Capacitor between V_{SS} and A_{GND}	Ceramic capacitor

* For recommended component values, refer to “Recommended Operating Conditions” in the “Electrical Characteristics” chapter.

21 Package

QFP13-64PIN (P-LQFP064-1010-0.50)

(Unit: mm)

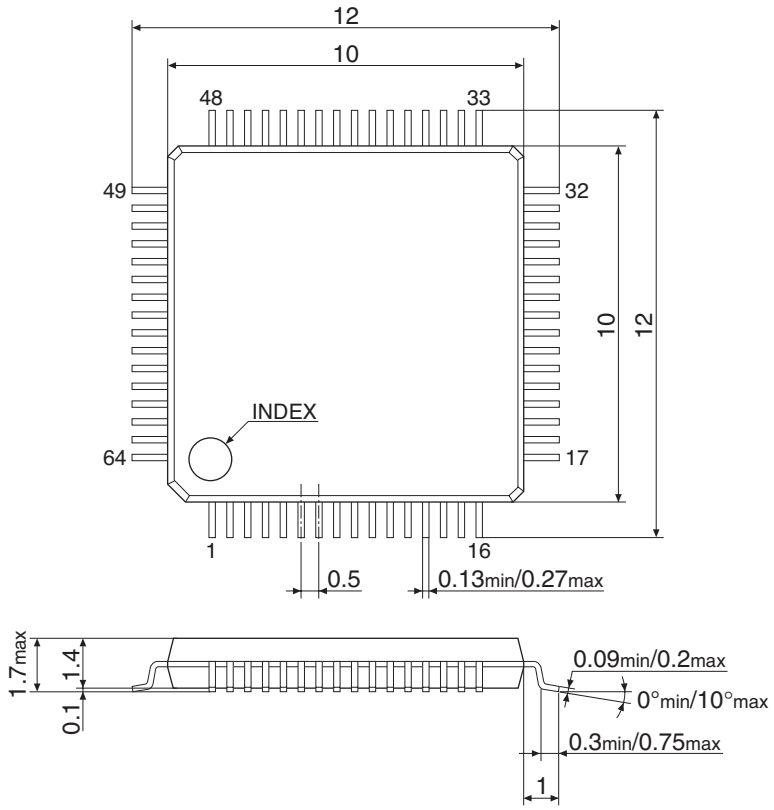


Figure 21.1 QFP13-64PIN Package Dimensions

QFP15-100PIN (P-LQFP100-1414-0.50)

(Unit: mm)

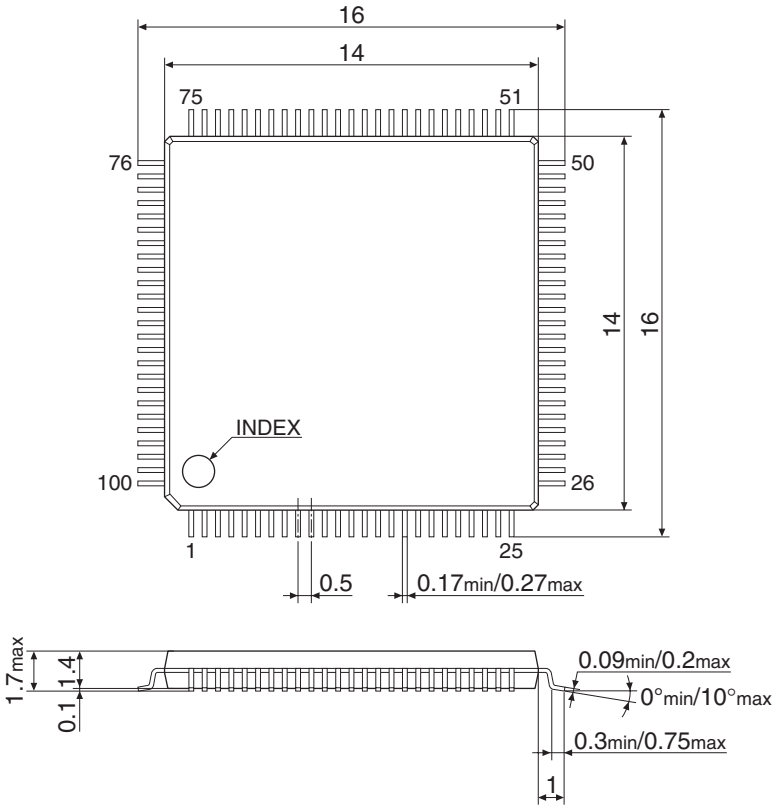


Figure 21.2 QFP15-100PIN Package Dimensions

Appendix A List of Peripheral Circuit Control Registers

0x4000–0x4008 Misc Registers (MISC)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000	MSCPROT (MISC System Protect Register)	15–0	PROT[15:0]	0x0000	H0	R/W	–
0x4002	MSCIRAMSZ (MISC IRAM Size Register)	15–9	–	0x00	–	R	–
		8	(reserved)	0	H0	R/WP	Always set to 0.
		7–3	–	0x04	–	R	–
		2–0	IRAMSZ[2:0]	0x2	H0	R/WP	
0x4004	MSCTTBRL (MISC Vector Table Address Low Register)	15–8	TTBR[15:8]	0x80	H0	R/WP	–
		7–0	TTBR[7:0]	0x00	H0	R	
0x4006	MSCTTBRH (MISC Vector Table Address High Register)	15–8	–	0x00	–	R	–
		7–0	TTBR[23:16]	0x00	H0	R/WP	
0x4008	MSCPSR (MISC PSR Register)	15–8	–	0x00	–	R	–
		7–5	PSRIL[2:0]	0x0	H0	R	
		4	PSRIE	0	H0	R	
		3	PSRC	0	H0	R	
		2	PSRV	0	H0	R	
		1	PSRZ	0	H0	R	
		0	PSRN	0	H0	R	

0x4020 Power Generator (PWG)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4020	PWGVD1CTL (V_{D1} Regulator Control Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1–0	REGMODE[1:0]	0x0	H0	R/WP	

0x4040–0x4054 Clock Generator (CLG)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4040	CLGSCLK (CLG System Clock Control Register)	15	WUPMD	0	H0	R/WP	–
		14	–	0	–	R	
		13–12	WUPDIV[1:0]	0x0	H0	R/WP	
		11–10	–	0x0	–	R	
		9–8	WUPSRC[1:0]	0x0	H0	R/WP	
		7–6	–	0x0	–	R	
		5–4	CLKDIV[1:0]	0x0	H0	R/WP	
		3–2	–	0x0	–	R	
		1–0	CLKSRC[1:0]	0x0	H0	R/WP	
0x4042	CLGOSC (CLG Oscillation Control Register)	15–12	–	0x0	–	R	–
		11	EXOSCSLPC	1	H0	R/W	
		10	OSC3SLPC	1	H0	R/W	
		9	OSC1SLPC	1	H0	R/W	
		8	IOSCSLPC	1	H0	R/W	
		7–4	–	0x0	–	R	
		3	EXOSCEN	0	H0	R/W	
		2	OSC3EN	0	H0	R/W	
		1	OSC1EN	0	H0	R/W	
		0	IOSCEN	1	H0	R/W	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4046	CLGOSC1 (CLG OSC1 Control Register)	15	–	0	–	R	–
		14	OSDRB	1	H0	R/WP	
		13	OSDEN	0	H0	R/WP	
		12	OSC1BUP	1	H0	R/WP	
		11	OSC1SELCR	0	H0	R/WP	
		10–8	CGI1[2:0]	0x0	H0	R/WP	
		7–6	INV1B[1:0]	0x2	H0	R/WP	
		5–4	INV1N[1:0]	0x1	H0	R/WP	
		3–2	–	0x0	–	R	
		1–0	OSC1WT[1:0]	0x2	H0	R/WP	
0x4048	CLGOSC3 (CLG OSC3 Control Register)	15–11	–	0x00	–	R	–
		10	OSC3FQ	0	–	R/WP	
		9-8	–	0x0	–	R	
		7–3	–	0x00	–	R	
		2–0	OSC3WT[2:0]	0x2	H0	R/WP	
0x404c	CLGINTF (CLG Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–6	–	0x0	H0	R	
		5	OSC1STPIF	0	H0	R/W	Cleared by writing 1.
		4–3	–	0x0	–	R	–
		2	OSC3STAIF	0	H0	R/W	Cleared by writing 1.
		1	OSC1STAIF	0	H0	R/W	
		0	IOSCSTAIF	0	H0	R/W	
0x404e	CLGINTE (CLG Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–6	–	0	–	R	
		5	OSC1STPIE	0	H0	R/W	
		4–3	–	0x0	–	R	
		2	OSC3STAIE	0	H0	R/W	
		1	OSC1STAIE	0	H0	R/W	
		0	IOSCSTAIE	0	H0	R/W	
0x4050	CLGFOUT (CLG FOUT Control Register)	15–8	–	0x00	–	R	–
		7	–	0	–	R	
		6–4	FOUTDIV[2:0]	0x0	H0	R/W	
		3–2	FOUTSRC[1:0]	0x0	H0	R/W	
		1	–	0	–	R	
		0	FOUTEN	0	H0	R/W	
0x4052	CLGTRIM1 (CLG Oscillation Frequency Trimming Register 1)	15–14	–	0x0	–	R	–
		13–8	OSC3AJ[5:0]	*	H0	R/WP	* Determined by factory adjustment.
		7–6	–	0x0	–	R	–
		5–0	IOSCAJ[5:0]	*	H0	R/WP	* Determined by factory adjustment.
0x4054	CLGTRIM2 (CLG Oscillation Frequency Trimming Register 2)	15–8	–	0x00	–	R	–
		7–6	–	0x0	–	R	
		5–0	OSC1AJ[5:0]	*	H0	R/WP	* Determined by factory adjustment.

0x4080–0x4096 Interrupt Controller (ITC)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4080	ITCLV0 (ITC Interrupt Level Setup Register 0)	15–11	–	0x00	–	R	–
		10–8	ILV1[2:0]	0x0	H0	R/W	Port interrupt (ILVPPORT)
		7–3	–	0x00	–	R	–
		2–0	ILV0[2:0]	0x0	H0	R/W	Supply voltage detector interrupt (ILVSVD4)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4082	ITCLV1 (ITC Interrupt Level Setup Register 1)	15–11	–	0x00	–	R	–
		10–8	ILV3[2:0]	0x0	H0	R/W	Clock generator interrupt (ILVCLG)
		7–0	–	0x00	–	R	–
0x4084	ITCLV2 (ITC Interrupt Level Setup Register 2)	15–11	–	0x00	–	R	–
		10–8	ILV5[2:0]	0x0	H0	R/W	16-bit timer Ch.0 interrupt (ILVT16_0)
		7–0	–	0x00	–	R	–
0x4086	ITCLV3 (ITC Interrupt Level Setup Register 3)	15–11	–	0x00	–	R	–
		10–8	ILV7[2:0]	0x0	H0	R/W	16-bit timer Ch.1 interrupt (ILVT16_1)
		7–3	–	0x00	–	R	–
		2–0	ILV6[2:0]	0x0	H0	R/W	UART Ch.0 interrupt (ILVUART3_0)
0x4088	ITCLV4 (ITC Interrupt Level Setup Register 4)	15–11	–	0x00	–	R	–
		10–8	ILV9[2:0]	0x0	H0	R/W	I^2C interrupt (ILVI2C_0)
		7–3	–	0x00	–	R	–
		2–0	ILV8[2:0]	0x0	H0	R/W	Synchronous serial interface Ch.0 interrupt (ILVSPIA_0)
0x408a	ITCLV5 (ITC Interrupt Level Setup Register 5)	15–11	–	0x00	–	R	–
		10–8	ILV11[2:0]	0x0	H0	R/W	DMM 16-bit PWM timer Ch.1 interrupt (ILVT16B_DMM_1)
		7–3	–	0x00	–	R	–
		2–0	ILV10[2:0]	0x0	H0	R/W	DMM 16-bit PWM timer Ch.0 interrupt (ILVT16B_DMM_0)
0x408c	ITCLV6 (ITC Interrupt Level Setup Register 6)	15–11	–	0x00	–	R	–
		10–8	ILV13[2:0]	0x0	H0	R/W	Sound generator interrupt (ILVSNDA_DMM_0)
		7–0	–	0x00	–	R	–
0x408e	ITCLV7 (ITC Interrupt Level Setup Register 7)	15–11	–	0x00	–	R	–
		10–8	ILV15[2:0]	0x0	H0	R/W	LCD driver interrupt (ILVLCD4B)
		7–0	–	0x00	–	R	–
0x4090	ITCLV8 (ITC Interrupt Level Setup Register 8)	15–11	–	0x00	–	R	–
		10–8	ILV17[2:0]	0x0	H0	R/W	EEPROM controller interrupt (ILVEPRC)
		7–0	–	0x00	–	R	–
0x4092	ITCLV9 (ITC Interrupt Level Setup Register 9)	15–8	–	0x00	–	R	–
		7–3	–	0x00	–	R	–
		2–0	ILV18[2:0]	0x0	H0	R/W	16-bit timer Ch.2 interrupt (ILVT16_2)
0x4094	ITCLV10 (ITC Interrupt Level Setup Register 10)	15–11	–	0x00	–	R	–
		10–8	ILV21[2:0]	0x0	–	R/W	DMM controller interrupt (ILVDSADC16_0)
		7–3	–	0x00	–	R	–
		2–0	ILV20[2:0]	0x0	–	R/W	16-bit timer Ch.3 interrupt (ILVT16_3)
0x4096	ITCLV11 (ITC Interrupt Level Setup Register 11)	15–8	–	0x00	–	R	–
		7–3	–	0x00	–	R	
		2–0	ILV22[2:0]	0x0	–	R/W	DMM 16-bit PWM timer Ch.2 interrupt (ILVT16B_DMM_2)

0x40a0–0x40a4 Watchdog Timer (WDT2)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x40a0	WDTCLK (WDT2 Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/WP	
		7–6	–	0x0	–	R	
		5–4	CLKDIV[1:0]	0x0	H0	R/WP	
		3–2	–	0x0	–	R	
		1–0	CLKSRC[1:0]	0x0	H0	R/WP	
0x40a2	WDTCTL (WDT2 Control Register)	15–11	–	0x00	–	R	–
		10–9	MOD[1:0]	0x0	H0	R/WP	
		8	STATNMI	0	H0	R	
		7–5	–	0x0	–	R	
		4	WDTCNTRST	0	H0	WP	Always read as 0.
		3–0	WDTRUN[3:0]	0xa	H0	R/WP	–
0x40a4	WDTCMP (WDT2 Counter Compare Match Register)	15–10	–	0x00	–	R	–
		9–0	CMP[9:0]	0x3ff	H0	R/WP	

0x4100–0x4106 Supply Voltage Detector (SVD4)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4100	SVDCLK (SVD4 Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	1	H0	R/WP	
		7	–	0	–	R	
		6–4	CLKDIV[2:0]	0x0	H0	R/WP	
		3–2	–	0x0	–	R	
		1–0	CLKSRC[1:0]	0x0	H0	R/WP	
0x4102	SVDCTL (SVD4 Control Register)	15	VDSEL	0	H1	R/WP	–
		14–13	SVDSC[1:0]	0x0	H0	R/WP	Writing takes effect when the SVDCTL.SVDMD[1:0] bits are not 0x0.
		12–8	SVDC[4:0]	0x1e	H1	R/WP	–
		7–4	SVDRE[3:0]	0x0	H1	R/WP	
		3	–	0	–	R	
		2–1	SVDMD[1:0]	0x0	H0	R/WP	
		0	MODEN	0	H1	R/WP	
0x4104	SVDINTF (SVD4 Status and Interrupt Flag Register)	15–9	–	0x00	–	R	–
		8	SVDDT	x	–	R	
		7–1	–	0x00	–	R	
		0	SVDIF	0	H1	R/W	Cleared by writing 1.
0x4106	SVDINTE (SVD4 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	SVDIE	0	H0	R/W	

0x4160–0x416c 16-bit Timer (T16) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4160	T16_0CLK (T16 Ch.0 Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	
		7–4	CLKDIV[3:0]	0x0	H0	R/W	
		3–2	–	0x0	–	R	
		1–0	CLKSRC[1:0]	0x0	H0	R/W	
0x4162	T16_0MOD (T16 Ch.0 Mode Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	TRMD	0	H0	R/W	
0x4164	T16_0CTL (T16 Ch.0 Control Register)	15–9	–	0x00	–	R	–
		8	PRUN	0	H0	R/W	
		7–2	–	0x00	–	R	
		1	PRESET	0	H0	R/W	
		0	MODEN	0	H0	R/W	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4166	T16_0TR (T16 Ch.0 Reload Data Register)	15–0	TR[15:0]	0xffff	H0	R/W	–
0x4168	T16_0TC (T16 Ch.0 Counter Data Register)	15–0	TC[15:0]	0xffff	H0	R	–
0x416a	T16_0INTF (T16 Ch.0 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIF	0	H0	R/W	Cleared by writing 1.
0x416c	T16_0INTE (T16 Ch.0 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIE	0	H0	R/W	

0x41b0 Flash Controller (FLASHC)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x41b0	FLASHCWAIT (FLASHC Flash Read Cycle Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	–
		1–0	RDWAIT[1:0]	0x1	H0	R/WP	

0x41c0–0x41ca EEPROM Controller (EEPROMC)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x41c0	EPRCCTL0 (EEPROMC Control Register 0)	15–9	–	0x00	–	R	–
		8	EP_XPOR	1	H0	R/WP	
		7–2	–	0x00	–	R	
		1	EP_PWRSET	0	H0	R/WP	
		0	EP_WMODE	0	H0	R/WP	
0x41c2	EPRCCTL1 (EEPROMC Control Register 1)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	EP_CK	0	H0	WP	
0x41c4	EPRCADR (EEPROMC Address Register)	15–8	–	0x00	–	R	–
		7–0	EP_ADDR[7:0]	0x00	H0	R/WP	
0x41c6	EPRCWDAT (EEPROMC Write Data Register)	15–8	–	0x00	–	R	–
		7–0	EP_WDAT[7:0]	0x00	H0	R/WP	
0x41c8	EPRCINTF (EEPROMC Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1	ECCERIF	0	H0	R/WP	Cleared by writing 1.
		0	RXBIF	0	H0	R/WP	
0x41ca	EPRCINTE (EEPROMC Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1	ECCERIE	0	H0	R/WP	
		0	RXBIE	0	H0	R/WP	

0x4200–0x42e2 I/O Ports (PPORT)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
0x4200	P0DAT (P0 Port Data Register)	15	P0OUT7	0	H0	R/W	–	✓	✓
		14	P0OUT6	0	H0	R/W		✓	✓
		13	P0OUT5	0	H0	R/W		✓	✓
		12	P0OUT4	0	H0	R/W		✓	✓
		11	P0OUT3	0	H0	R/W		✓	✓
		10	P0OUT2	0	H0	R/W		✓	✓
		9	P0OUT1	0	H0	R/W		✓	✓
		8	P0OUT0	0	H0	R/W		✓	✓
		7	P0IN7	0	H0	R	–	✓	✓
		6	P0IN6	0	H0	R		✓	✓
		5	P0IN5	0	H0	R		✓	✓
		4	P0IN4	0	H0	R		✓	✓
		3	P0IN3	0	H0	R		✓	✓
		2	P0IN2	0	H0	R		✓	✓
		1	P0IN1	0	H0	R		✓	✓
		0	P0IN0	0	H0	R		✓	✓
0x4202	P0IOEN (P0 Port Enable Register)	15	P0IEN7	0	H0	R/W	–	✓	✓
		14	P0IEN6	0	H0	R/W		✓	✓
		13	P0IEN5	0	H0	R/W		✓	✓
		12	P0IEN4	0	H0	R/W		✓	✓
		11	P0IEN3	0	H0	R/W		✓	✓
		10	P0IEN2	0	H0	R/W		✓	✓
		9	P0IEN1	0	H0	R/W		✓	✓
		8	P0IEN0	0	H0	R/W		✓	✓
		7	P0OEN7	0	H0	R/W	–	✓	✓
		6	P0OEN6	0	H0	R/W		✓	✓
		5	P0OEN5	0	H0	R/W		✓	✓
		4	P0OEN4	0	H0	R/W		✓	✓
		3	P0OEN3	0	H0	R/W		✓	✓
		2	P0OEN2	0	H0	R/W		✓	✓
		1	P0OEN1	0	H0	R/W		✓	✓
		0	P0OEN0	0	H0	R/W		✓	✓
0x4204	P0RCTL (P0 Port Pull-up/down Control Register)	15	P0PDPU7	0	H0	R/W	–	✓	✓
		14	P0PDPU6	0	H0	R/W		✓	✓
		13	P0PDPU5	0	H0	R/W		✓	✓
		12	P0PDPU4	0	H0	R/W		✓	✓
		11	P0PDPU3	0	H0	R/W		✓	✓
		10	P0PDPU2	0	H0	R/W		✓	✓
		9	P0PDPU1	0	H0	R/W		✓	✓
		8	P0PDPU0	0	H0	R/W		✓	✓
		7	P0REN7	0	H0	R/W	–	✓	✓
		6	P0REN6	0	H0	R/W		✓	✓
		5	P0REN5	0	H0	R/W		✓	✓
		4	P0REN4	0	H0	R/W		✓	✓
		3	P0REN3	0	H0	R/W		✓	✓
		2	P0REN2	0	H0	R/W		✓	✓
		1	P0REN1	0	H0	R/W		✓	✓
		0	P0REN0	0	H0	R/W		✓	✓
0x4206	P0INTF (P0 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–
		7	P0IF7	0	H0	R/W	Cleared by writing 1.	✓	✓
		6	P0IF6	0	H0	R/W		✓	✓
		5	P0IF5	0	H0	R/W		✓	✓
		4	P0IF4	0	H0	R/W		✓	✓
		3	P0IF3	0	H0	R/W		✓	✓
		2	P0IF2	0	H0	R/W		✓	✓
		1	P0IF1	0	H0	R/W		✓	✓
		0	P0IF0	0	H0	R/W		✓	✓

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
0x4208	P0INTCTL (P0 Port Interrupt Control Register)	15	P0EDGE7	0	H0	R/W	–	✓	✓
		14	P0EDGE6	0	H0	R/W		✓	✓
		13	P0EDGE5	0	H0	R/W		✓	✓
		12	P0EDGE4	0	H0	R/W		✓	✓
		11	P0EDGE3	0	H0	R/W		✓	✓
		10	P0EDGE2	0	H0	R/W		✓	✓
		9	P0EDGE1	0	H0	R/W		✓	✓
		8	P0EDGE0	0	H0	R/W		✓	✓
		7	P0IE7	0	H0	R/W	–	✓	✓
		6	P0IE6	0	H0	R/W		✓	✓
		5	P0IE5	0	H0	R/W		✓	✓
		4	P0IE4	0	H0	R/W		✓	✓
		3	P0IE3	0	H0	R/W		✓	✓
		2	P0IE2	0	H0	R/W		✓	✓
		1	P0IE1	0	H0	R/W		✓	✓
		0	P0IE0	0	H0	R/W		✓	✓
0x420a	P0CHATEN (P0 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–	–	–
		7	P0CHATEN7	0	H0	R/W	–	✓	✓
		6	P0CHATEN6	0	H0	R/W		✓	✓
		5	P0CHATEN5	0	H0	R/W		✓	✓
		4	P0CHATEN4	0	H0	R/W		✓	✓
		3	P0CHATEN3	0	H0	R/W		✓	✓
		2	P0CHATEN2	0	H0	R/W		✓	✓
		1	P0CHATEN1	0	H0	R/W		✓	✓
		0	P0CHATEN0	0	H0	R/W		✓	✓
0x420c	P0MODSEL (P0 Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–
		7	P0SEL7	0	H0	R/W	–	✓	✓
		6	P0SEL6	0	H0	R/W		✓	✓
		5	P0SEL5	0	H0	R/W		✓	✓
		4	P0SEL4	0	H0	R/W		✓	✓
		3	P0SEL3	0	H0	R/W		✓	✓
		2	P0SEL2	0	H0	R/W		✓	✓
		1	P0SEL1	0	H0	R/W		✓	✓
		0	P0SEL0	0	H0	R/W		✓	✓
0x420e	P0FNCSEL (P0 Port Function Select Register)	15–14	P07MUX[1:0]	0x0	H0	R/W	–	✓	✓
		13–12	P06MUX[1:0]	0x0	H0	R/W		✓	✓
		11–10	P05MUX[1:0]	0x0	H0	R/W		✓	✓
		9–8	P04MUX[1:0]	0x0	H0	R/W		✓	✓
		7–6	P03MUX[1:0]	0x0	H0	R/W		✓	✓
		5–4	P02MUX[1:0]	0x0	H0	R/W		✓	✓
		3–2	P01MUX[1:0]	0x0	H0	R/W		✓	✓
		1–0	P00MUX[1:0]	0x0	H0	R/W		✓	✓
0x4210	P1DAT (P1 Port Data Register)	15	P1OUT7	0	H0	R/W	–	✓	✓
		14	P1OUT6	0	H0	R/W		✓	✓
		13	P1OUT5	0	H0	R/W		✓	✓
		12	P1OUT4	0	H0	R/W		✓	✓
		11	P1OUT3	0	H0	R/W		✓	✓
		10	P1OUT2	0	H0	R/W		✓	✓
		9	P1OUT1	0	H0	R/W		✓	✓
		8	P1OUT0	0	H0	R/W		✓	✓
		7	P1IN7	0	H0	R	–	✓	✓
		6	P1IN6	0	H0	R		✓	✓
		5	P1IN5	0	H0	R		✓	✓
		4	P1IN4	0	H0	R		✓	✓
		3	P1IN3	0	H0	R		✓	✓
		2	P1IN2	0	H0	R		✓	✓
		1	P1IN1	0	H0	R		✓	✓
		0	P1IN0	0	H0	R		✓	✓

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
0x4212	P1IOEN (P1 Port Enable Register)	15	P1IEN7	0	H0	R/W	–	✓	✓
		14	P1IEN6	0	H0	R/W		✓	✓
		13	P1IEN5	0	H0	R/W		✓	✓
		12	P1IEN4	0	H0	R/W		✓	✓
		11	P1IEN3	0	H0	R/W		✓	✓
		10	P1IEN2	0	H0	R/W		✓	✓
		9	P1IEN1	0	H0	R/W		✓	✓
		8	P1IEN0	0	H0	R/W		✓	✓
		7	P1OEN7	0	H0	R/W	–	✓	✓
		6	P1OEN6	0	H0	R/W		✓	✓
		5	P1OEN5	0	H0	R/W		✓	✓
		4	P1OEN4	0	H0	R/W		✓	✓
		3	P1OEN3	0	H0	R/W		✓	✓
		2	P1OEN2	0	H0	R/W		✓	✓
		1	P1OEN1	0	H0	R/W		✓	✓
		0	P1OEN0	0	H0	R/W		✓	✓
0x4214	P1RCTL (P1 Port Pull-up/down Control Register)	15	P1PDPU7	0	H0	R/W	–	✓	✓
		14	P1PDPU6	0	H0	R/W		✓	✓
		13	P1PDPU5	0	H0	R/W		✓	✓
		12	P1PDPU4	0	H0	R/W		✓	✓
		11	P1PDPU3	0	H0	R/W		✓	✓
		10	P1PDPU2	0	H0	R/W		✓	✓
		9	P1PDPU1	0	H0	R/W		✓	✓
		8	P1PDPU0	0	H0	R/W		✓	✓
		7	P1REN7	0	H0	R/W	–	✓	✓
		6	P1REN6	0	H0	R/W		✓	✓
		5	P1REN5	0	H0	R/W		✓	✓
		4	P1REN4	0	H0	R/W		✓	✓
		3	P1REN3	0	H0	R/W		✓	✓
		2	P1REN2	0	H0	R/W		✓	✓
		1	P1REN1	0	H0	R/W		✓	✓
		0	P1REN0	0	H0	R/W		✓	✓
0x4216	P1INTF (P1 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–
		7	P1IF7	0	H0	R/W	Cleared by writing 1.	✓	✓
		6	P1IF6	0	H0	R/W		✓	✓
		5	P1IF5	0	H0	R/W		✓	✓
		4	P1IF4	0	H0	R/W		✓	✓
		3	P1IF3	0	H0	R/W		✓	✓
		2	P1IF2	0	H0	R/W		✓	✓
		1	P1IF1	0	H0	R/W		✓	✓
		0	P1IF0	0	H0	R/W		✓	✓
0x4218	P1INTCTL (P1 Port Interrupt Control Register)	15	P1EDGE7	0	H0	R/W	–	✓	✓
		14	P1EDGE6	0	H0	R/W		✓	✓
		13	P1EDGE5	0	H0	R/W		✓	✓
		12	P1EDGE4	0	H0	R/W		✓	✓
		11	P1EDGE3	0	H0	R/W		✓	✓
		10	P1EDGE2	0	H0	R/W		✓	✓
		9	P1EDGE1	0	H0	R/W		✓	✓
		8	P1EDGE0	0	H0	R/W		✓	✓
		7	P1IE7	0	H0	R/W	–	✓	✓
		6	P1IE6	0	H0	R/W		✓	✓
		5	P1IE5	0	H0	R/W		✓	✓
		4	P1IE4	0	H0	R/W		✓	✓
		3	P1IE3	0	H0	R/W		✓	✓
		2	P1IE2	0	H0	R/W		✓	✓
		1	P1IE1	0	H0	R/W		✓	✓
		0	P1IE0	0	H0	R/W		✓	✓

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
0x421a	P1CHATEN	15–8	–	0x00	–	R	–	–	–
	(P1 Port Chattering	7	P1CHATEN7	0	H0	R/W	–	✓	✓
	Filter Enable Register)	6	P1CHATEN6	0	H0	R/W		✓	✓
		5	P1CHATEN5	0	H0	R/W		✓	✓
		4	P1CHATEN4	0	H0	R/W		✓	✓
		3	P1CHATEN3	0	H0	R/W		✓	✓
		2	P1CHATEN2	0	H0	R/W		✓	✓
		1	P1CHATEN1	0	H0	R/W		✓	✓
		0	P1CHATEN0	0	H0	R/W		✓	✓
0x421c	P1MODSEL	15–8	–	0x00	–	R	–	–	–
	(P1 Port Mode Select	7	P1SEL7	0	H0	R/W	–	✓	✓
	Register)	6	P1SEL6	0	H0	R/W		✓	✓
		5	P1SEL5	0	H0	R/W		✓	✓
		4	P1SEL4	0	H0	R/W		✓	✓
		3	P1SEL3	0	H0	R/W		✓	✓
		2	P1SEL2	0	H0	R/W		✓	✓
		1	P1SEL1	0	H0	R/W		✓	✓
		0	P1SEL0	0	H0	R/W		✓	✓
0x421e	P1FNCSEL	15–14	P17MUX[1:0]	0x0	H0	R/W	–	✓	✓
	(P1 Port Function	13–12	P16MUX[1:0]	0x0	H0	R/W		✓	✓
	Select Register)	11–10	P15MUX[1:0]	0x0	H0	R/W		✓	✓
		9–8	P14MUX[1:0]	0x0	H0	R/W		✓	✓
		7–6	P13MUX[1:0]	0x0	H0	R/W		✓	✓
		5–4	P12MUX[1:0]	0x0	H0	R/W		✓	✓
		3–2	P11MUX[1:0]	0x0	H0	R/W		✓	✓
		1–0	P10MUX[1:0]	0x0	H0	R/W		✓	✓
0x4220	P2DAT	15	P2OUT7	0	H0	R/W	–	–	✓
	(P2 Port Data	14	P2OUT6	0	H0	R/W		–	✓
	Register)	13	P2OUT5	0	H0	R/W		–	✓
		12	P2OUT4	0	H0	R/W		–	✓
		11	P2OUT3	0	H0	R/W		–	✓
		10	P2OUT2	0	H0	R/W		–	✓
		9	P2OUT1	0	H0	R/W		–	✓
		8	P2OUT0	0	H0	R/W		–	✓
		7	P2IN7	0	H0	R	–	–	✓
		6	P2IN6	0	H0	R		–	✓
		5	P2IN5	0	H0	R		–	✓
		4	P2IN4	0	H0	R		–	✓
		3	P2IN3	0	H0	R		–	✓
		2	P2IN2	0	H0	R		–	✓
		1	P2IN1	0	H0	R		–	✓
		0	P2IN0	0	H0	R		–	✓
0x4222	P2IOEN	15	P2IEN7	0	H0	R/W	–	–	✓
	(P2 Port Enable	14	P2IEN6	0	H0	R/W		–	✓
	Register)	13	P2IEN5	0	H0	R/W		–	✓
		12	P2IEN4	0	H0	R/W		–	✓
		11	P2IEN3	0	H0	R/W		–	✓
		10	P2IEN2	0	H0	R/W		–	✓
		9	P2IEN1	0	H0	R/W		–	✓
		8	P2IEN0	0	H0	R/W		–	✓
		7	P2OEN7	0	H0	R/W	–	–	✓
		6	P2OEN6	0	H0	R/W		–	✓
		5	P2OEN5	0	H0	R/W		–	✓
		4	P2OEN4	0	H0	R/W		–	✓
		3	P2OEN3	0	H0	R/W		–	✓
		2	P2OEN2	0	H0	R/W		–	✓
		1	P2OEN1	0	H0	R/W		–	✓
		0	P2OEN0	0	H0	R/W		–	✓

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
0x4224	P2RCTL (P2 Port Pull-up/down Control Register)	15	P2PDPU7	0	H0	R/W	–	–	✓
		14	P2PDPU6	0	H0	R/W		–	✓
		13	P2PDPU5	0	H0	R/W		–	✓
		12	P2PDPU4	0	H0	R/W		–	✓
		11	P2PDPU3	0	H0	R/W		–	✓
		10	P2PDPU2	0	H0	R/W		–	✓
		9	P2PDPU1	0	H0	R/W		–	✓
		8	P2PDPU0	0	H0	R/W		–	✓
		7	P2REN7	0	H0	R/W	–	–	✓
		6	P2REN6	0	H0	R/W		–	✓
		5	P2REN5	0	H0	R/W		–	✓
		4	P2REN4	0	H0	R/W		–	✓
		3	P2REN3	0	H0	R/W		–	✓
		2	P2REN2	0	H0	R/W		–	✓
		1	P2REN1	0	H0	R/W		–	✓
		0	P2REN0	0	H0	R/W		–	✓
0x4226	P2INTF (P2 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–
		7	P2IF7	0	H0	R/W	Cleared by writing 1.	–	✓
		6	P2IF6	0	H0	R/W		–	✓
		5	P2IF5	0	H0	R/W		–	✓
		4	P2IF4	0	H0	R/W		–	✓
		3	P2IF3	0	H0	R/W		–	✓
		2	P2IF2	0	H0	R/W		–	✓
		1	P2IF1	0	H0	R/W		–	✓
		0	P2IF0	0	H0	R/W		–	✓
0x4228	P2INTCTL (P2 Port Interrupt Control Register)	15	P2EDGE7	0	H0	R/W	–	–	✓
		14	P2EDGE6	0	H0	R/W		–	✓
		13	P2EDGE5	0	H0	R/W		–	✓
		12	P2EDGE4	0	H0	R/W		–	✓
		11	P2EDGE3	0	H0	R/W		–	✓
		10	P2EDGE2	0	H0	R/W		–	✓
		9	P2EDGE1	0	H0	R/W		–	✓
		8	P2EDGE0	0	H0	R/W		–	✓
		7	P2IE7	0	H0	R/W	–	–	✓
		6	P2IE6	0	H0	R/W		–	✓
		5	P2IE5	0	H0	R/W		–	✓
		4	P2IE4	0	H0	R/W		–	✓
		3	P2IE3	0	H0	R/W		–	✓
		2	P2IE2	0	H0	R/W		–	✓
		1	P2IE1	0	H0	R/W		–	✓
		0	P2IE0	0	H0	R/W		–	✓
0x422a	P2CHATEN (P2 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–	–	–
		7	P2CHATEN7	0	H0	R/W	–	–	✓
		6	P2CHATEN6	0	H0	R/W		–	✓
		5	P2CHATEN5	0	H0	R/W		–	✓
		4	P2CHATEN4	0	H0	R/W		–	✓
		3	P2CHATEN3	0	H0	R/W		–	✓
		2	P2CHATEN2	0	H0	R/W		–	✓
		1	P2CHATEN1	0	H0	R/W		–	✓
		0	P2CHATEN0	0	H0	R/W		–	✓

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
0x422c	P2MODSEL (P2 Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–
		7	P2SEL7	0	H0	R/W	–	–	✓
		6	P2SEL6	0	H0	R/W		–	✓
		5	P2SEL5	0	H0	R/W		–	✓
		4	P2SEL4	0	H0	R/W		–	✓
		3	P2SEL3	0	H0	R/W		–	✓
		2	P2SEL2	0	H0	R/W		–	✓
		1	P2SEL1	0	H0	R/W		–	✓
		0	P2SEL0	0	H0	R/W		–	✓
0x422e	P2FNCSEL (P2 Port Function Select Register)	15–14	P27MUX[1:0]	0x0	H0	R/W	–	–	✓
		13–12	P26MUX[1:0]	0x0	H0	R/W		–	✓
		11–10	P25MUX[1:0]	0x0	H0	R/W		–	✓
		9–8	P24MUX[1:0]	0x0	H0	R/W		–	✓
		7–6	P23MUX[1:0]	0x0	H0	R/W		–	✓
		5–4	P22MUX[1:0]	0x0	H0	R/W		–	✓
		3–2	P21MUX[1:0]	0x0	H0	R/W		–	✓
		1–0	P20MUX[1:0]	0x0	H0	R/W		–	✓
0x4230	P3DAT (P3 Port Data Register)	15	P3OUT7	0	H0	R/W	–	–	✓
		14	P3OUT6	0	H0	R/W		–	✓
		13	P3OUT5	0	H0	R/W		–	✓
		12	P3OUT4	0	H0	R/W		–	✓
		11	P3OUT3	0	H0	R/W		–	✓
		10	P3OUT2	0	H0	R/W		–	✓
		9	P3OUT1	0	H0	R/W		–	✓
		8	P3OUT0	0	H0	R/W		–	✓
		7	P3IN7	0	H0	R	–	–	✓
		6	P3IN6	0	H0	R		–	✓
		5	P3IN5	0	H0	R		–	✓
		4	P3IN4	0	H0	R		–	✓
		3	P3IN3	0	H0	R		–	✓
		2	P3IN2	0	H0	R		–	✓
		1	P3IN1	0	H0	R		–	✓
		0	P3IN0	0	H0	R		–	✓
0x4232	P3IOEN (P3 Port Enable Register)	15	P3IEN7	0	H0	R/W	–	–	✓
		14	P3IEN6	0	H0	R/W		–	✓
		13	P3IEN5	0	H0	R/W		–	✓
		12	P3IEN4	0	H0	R/W		–	✓
		11	P3IEN3	0	H0	R/W		–	✓
		10	P3IEN2	0	H0	R/W		–	✓
		9	P3IEN1	0	H0	R/W		–	✓
		8	P3IEN0	0	H0	R/W		–	✓
		7	P3OEN7	0	H0	R/W	–	–	✓
		6	P3OEN6	0	H0	R/W		–	✓
		5	P3OEN5	0	H0	R/W		–	✓
		4	P3OEN4	0	H0	R/W		–	✓
		3	P3OEN3	0	H0	R/W		–	✓
		2	P3OEN2	0	H0	R/W		–	✓
		1	P3OEN1	0	H0	R/W		–	✓
		0	P3OEN0	0	H0	R/W		–	✓

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
0x4234	P3RCTL (P3 Port Pull-up/down Control Register)	15	P3PDPU7	0	H0	R/W	–	–	✓
		14	P3PDPU6	0	H0	R/W		–	✓
		13	P3PDPU5	0	H0	R/W		–	✓
		12	P3PDPU4	0	H0	R/W		–	✓
		11	P3PDPU3	0	H0	R/W		–	✓
		10	P3PDPU2	0	H0	R/W		–	✓
		9	P3PDPU1	0	H0	R/W		–	✓
		8	P3PDPU0	0	H0	R/W		–	✓
		7	P3REN7	0	H0	R/W	–	–	✓
		6	P3REN6	0	H0	R/W		–	✓
		5	P3REN5	0	H0	R/W		–	✓
		4	P3REN4	0	H0	R/W		–	✓
		3	P3REN3	0	H0	R/W		–	✓
		2	P3REN2	0	H0	R/W		–	✓
		1	P3REN1	0	H0	R/W		–	✓
		0	P3REN0	0	H0	R/W		–	✓
0x4236	P3INTF (P3 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–
		7	P3IF7	0	H0	R/W	Cleared by writing 1.	–	✓
		6	P3IF6	0	H0	R/W		–	✓
		5	P3IF5	0	H0	R/W		–	✓
		4	P3IF4	0	H0	R/W		–	✓
		3	P3IF3	0	H0	R/W		–	✓
		2	P3IF2	0	H0	R/W		–	✓
		1	P3IF1	0	H0	R/W		–	✓
		0	P3IF0	0	H0	R/W		–	✓
0x4238	P3INTCTL (P3 Port Interrupt Control Register)	15	P3EDGE7	0	H0	R/W	–	–	✓
		14	P3EDGE6	0	H0	R/W		–	✓
		13	P3EDGE5	0	H0	R/W		–	✓
		12	P3EDGE4	0	H0	R/W		–	✓
		11	P3EDGE3	0	H0	R/W		–	✓
		10	P3EDGE2	0	H0	R/W		–	✓
		9	P3EDGE1	0	H0	R/W		–	✓
		8	P3EDGE0	0	H0	R/W		–	✓
		7	P3IE7	0	H0	R/W	–	–	✓
		6	P3IE6	0	H0	R/W		–	✓
		5	P3IE5	0	H0	R/W		–	✓
		4	P3IE4	0	H0	R/W		–	✓
		3	P3IE3	0	H0	R/W		–	✓
		2	P3IE2	0	H0	R/W		–	✓
		1	P3IE1	0	H0	R/W		–	✓
		0	P3IE0	0	H0	R/W		–	✓
0x423a	P3CHATEN (P3 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–	–	–
		7	P3CHATEN7	0	H0	R/W	–	–	✓
		6	P3CHATEN6	0	H0	R/W		–	✓
		5	P3CHATEN5	0	H0	R/W		–	✓
		4	P3CHATEN4	0	H0	R/W		–	✓
		3	P3CHATEN3	0	H0	R/W		–	✓
		2	P3CHATEN2	0	H0	R/W		–	✓
		1	P3CHATEN1	0	H0	R/W		–	✓
		0	P3CHATEN0	0	H0	R/W		–	✓

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
0x423c	P3MODSEL (P3 Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–
		7	P3SEL7	0	H0	R/W	–	–	✓
		6	P3SEL6	0	H0	R/W		–	✓
		5	P3SEL5	0	H0	R/W		–	✓
		4	P3SEL4	0	H0	R/W		–	✓
		3	P3SEL3	0	H0	R/W		–	✓
		2	P3SEL2	0	H0	R/W		–	✓
		1	P3SEL1	0	H0	R/W		–	✓
		0	P3SEL0	0	H0	R/W		–	✓
0x423e	P3FNCSEL (P3 Port Function Select Register)	15–14	P37MUX[1:0]	0x0	H0	R/W	–	–	✓
		13–12	P36MUX[1:0]	0x0	H0	R/W		–	✓
		11–10	P35MUX[1:0]	0x0	H0	R/W		–	✓
		9–8	P34MUX[1:0]	0x0	H0	R/W		–	✓
		7–6	P33MUX[1:0]	0x0	H0	R/W		–	✓
		5–4	P32MUX[1:0]	0x0	H0	R/W		–	✓
		3–2	P31MUX[1:0]	0x0	H0	R/W		–	✓
		1–0	P30MUX[1:0]	0x0	H0	R/W		–	✓
0x4240	P4DAT (P4 Port Data Register)	15	P4OUT7	0	H0	R/W	–	–	✓
		14	P4OUT6	0	H0	R/W		–	✓
		13	P4OUT5	0	H0	R/W		–	✓
		12	P4OUT4	0	H0	R/W		–	✓
		11	P4OUT3	0	H0	R/W		–	✓
		10	P4OUT2	0	H0	R/W		–	✓
		9	P4OUT1	0	H0	R/W		–	✓
		8	P4OUT0	0	H0	R/W		–	✓
		7	P4IN7	0	H0	R	–	–	✓
		6	P4IN6	0	H0	R		–	✓
		5	P4IN5	0	H0	R		–	✓
		4	P4IN4	0	H0	R		–	✓
		3	P4IN3	0	H0	R		–	✓
		2	P4IN2	0	H0	R		–	✓
		1	P4IN1	0	H0	R		–	✓
		0	P4IN0	0	H0	R		–	✓
0x4242	P4IOEN (P4 Port Enable Register)	15	P4IEN7	0	H0	R/W	–	–	✓
		14	P4IEN6	0	H0	R/W		–	✓
		13	P4IEN5	0	H0	R/W		–	✓
		12	P4IEN4	0	H0	R/W		–	✓
		11	P4IEN3	0	H0	R/W		–	✓
		10	P4IEN2	0	H0	R/W		–	✓
		9	P4IEN1	0	H0	R/W		–	✓
		8	P4IEN0	0	H0	R/W		–	✓
		7	P4OEN7	0	H0	R/W	–	–	✓
		6	P4OEN6	0	H0	R/W		–	✓
		5	P4OEN5	0	H0	R/W		–	✓
		4	P4OEN4	0	H0	R/W		–	✓
		3	P4OEN3	0	H0	R/W		–	✓
		2	P4OEN2	0	H0	R/W		–	✓
		1	P4OEN1	0	H0	R/W		–	✓
		0	P4OEN0	0	H0	R/W		–	✓

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
0x4244	P4RCTL (P4 Port Pull-up/down Control Register)	15	P4PDPU7	0	H0	R/W	–	–	✓
		14	P4PDPU6	0	H0	R/W		–	✓
		13	P4PDPU5	0	H0	R/W		–	✓
		12	P4PDPU4	0	H0	R/W		–	✓
		11	P4PDPU3	0	H0	R/W		–	✓
		10	P4PDPU2	0	H0	R/W		–	✓
		9	P4PDPU1	0	H0	R/W		–	✓
		8	P4PDPU0	0	H0	R/W		–	✓
		7	P4REN7	0	H0	R/W	–	–	✓
		6	P4REN6	0	H0	R/W		–	✓
		5	P4REN5	0	H0	R/W		–	✓
		4	P4REN4	0	H0	R/W		–	✓
		3	P4REN3	0	H0	R/W		–	✓
		2	P4REN2	0	H0	R/W		–	✓
		1	P4REN1	0	H0	R/W		–	✓
		0	P4REN0	0	H0	R/W		–	✓
0x4246	P4INTF (P4 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–
		7	P4IF7	0	H0	R/W	Cleared by writing 1.	–	✓
		6	P4IF6	0	H0	R/W		–	✓
		5	P4IF5	0	H0	R/W		–	✓
		4	P4IF4	0	H0	R/W		–	✓
		3	P4IF3	0	H0	R/W		–	✓
		2	P4IF2	0	H0	R/W		–	✓
		1	P4IF1	0	H0	R/W		–	✓
		0	P4IF0	0	H0	R/W		–	✓
0x4248	P4INTCTL (P4 Port Interrupt Control Register)	15	P4EDGE7	0	H0	R/W	–	–	✓
		14	P4EDGE6	0	H0	R/W		–	✓
		13	P4EDGE5	0	H0	R/W		–	✓
		12	P4EDGE4	0	H0	R/W		–	✓
		11	P4EDGE3	0	H0	R/W		–	✓
		10	P4EDGE2	0	H0	R/W		–	✓
		9	P4EDGE1	0	H0	R/W		–	✓
		8	P4EDGE0	0	H0	R/W		–	✓
		7	P4IE7	0	H0	R/W	–	–	✓
		6	P4IE6	0	H0	R/W		–	✓
		5	P4IE5	0	H0	R/W		–	✓
		4	P4IE4	0	H0	R/W		–	✓
		3	P4IE3	0	H0	R/W		–	✓
		2	P4IE2	0	H0	R/W		–	✓
		1	P4IE1	0	H0	R/W		–	✓
		0	P4IE0	0	H0	R/W		–	✓
0x424a	P4CHATEN (P4 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–	–	–
		7	P4CHATEN7	0	H0	R/W	–	–	✓
		6	P4CHATEN6	0	H0	R/W		–	✓
		5	P4CHATEN5	0	H0	R/W		–	✓
		4	P4CHATEN4	0	H0	R/W		–	✓
		3	P4CHATEN3	0	H0	R/W		–	✓
		2	P4CHATEN2	0	H0	R/W		–	✓
		1	P4CHATEN1	0	H0	R/W		–	✓
		0	P4CHATEN0	0	H0	R/W		–	✓

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
0x424c	P4MODSEL (P4 Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–
		7	P4SEL7	0	H0	R/W	–	–	✓
		6	P4SEL6	0	H0	R/W		–	✓
		5	P4SEL5	0	H0	R/W		–	✓
		4	P4SEL4	0	H0	R/W		–	✓
		3	P4SEL3	0	H0	R/W		–	✓
		2	P4SEL2	0	H0	R/W		–	✓
		1	P4SEL1	0	H0	R/W		–	✓
		0	P4SEL0	0	H0	R/W		–	✓
0x424e	P4FNCSEL (P4 Port Function Select Register)	15–14	P47MUX[1:0]	0x0	H0	R/W	–	–	✓
		13–12	P46MUX[1:0]	0x0	H0	R/W		–	✓
		11–10	P45MUX[1:0]	0x0	H0	R/W		–	✓
		9–8	P44MUX[1:0]	0x0	H0	R/W		–	✓
		7–6	P43MUX[1:0]	0x0	H0	R/W		–	✓
		5–4	P42MUX[1:0]	0x0	H0	R/W		–	✓
		3–2	P41MUX[1:0]	0x0	H0	R/W		–	✓
		1–0	P40MUX[1:0]	0x0	H0	R/W		–	✓
0x42d0	PDDAT (Pd Port Data Register)	15–11	–	0x00	–	R	–	–	–
		10	PDOUT2	0	H0	R/W	–	✓	✓
		9	PDOUT1	0	H0	R/W		✓	✓
		8	PDOUT0	0	H0	R/W		✓	✓
		7–3	–	0x00	–	R	–	–	–
		2	–	0	–	R	–	–	–
		1	PDIN1	X	H0	R		✓	✓
		0	PDIN0	X	H0	R		✓	✓
0x42d2	PDIOEN (Pd Port Enable Register)	15–11	–	0x00	–	R	–	–	–
		10	(reserved)	0	H0	R/W	–	✓	✓
		9	PDIEN1	0	H0	R/W		✓	✓
		8	PDIEN0	0	H0	R/W		✓	✓
		7–3	–	0x00	–	R	–	–	–
		2	PDOEN2	0	H0	R/W	–	✓	✓
		1	PDOEN1	0	H0	R/W		✓	✓
		0	PDOEN0	0	H0	R/W		✓	✓
0x42d4	PDRCTL (Pd Port Pull-up/down Control Register)	15–11	–	0x00	–	R	–	–	–
		10	(reserved)	0	H0	R/W	–	✓	✓
		9	PDPDPU1	0	H0	R/W		✓	✓
		8	PDPDPU0	0	H0	R/W		✓	✓
		7–5	–	0x00	–	R	–	–	–
		2	(reserved)	0	H0	R/W	–	✓	✓
		1	PDREN1	0	H0	R/W		✓	✓
		0	PDREN0	0	H0	R/W		✓	✓
0x42dc	PDMODSEL (Pd Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–
		7–3	–	0	–	R	–	–	–
		2	PDSEL2	1	H0	R/W		✓	✓
		1	PDSEL1	1	H0	R/W		✓	✓
		0	PDSEL0	1	H0	R/W		✓	✓
0x42de	PDFNCSEL (Pd Port Function Select Register)	15–8	–	0x00	–	R	–	–	–
		7–6	–	0x0	–	R			
		5–4	PD2MUX[1:0]	0x0	H0	R/W	–	✓	✓
		3–2	PD1MUX[1:0]	0x0	H0	R/W		✓	✓
		1–0	PD0MUX[1:0]	0x0	H0	R/W		✓	✓
0x42e0	PCLK (P Port Clock Control Register)	15–9	–	0x00	–	R	–	–	–
		8	DBRUN	0	H0	R/WP	–	✓	✓
		7–4	CLKDIV[3:0]	0x0	H0	R/WP		✓	✓
		3–2	KRSTCFG[1:0]	0x0	H0	R/WP		✓	✓
		1–0	CLKSRC[1:0]	0x0	H0	R/WP		✓	✓

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
0x42e2	PINTFGRP (P Port Interrupt Flag Group Register)	15–8	–	0x00	–	R	–	–	–
		7–5	–	0x0	–	R			
		4	P4INT	0	H0	R	–	–	✓
		3	P3INT	0	H0	R		–	✓
		2	P2INT	0	H0	R		–	✓
		1	P1INT	0	H0	R		✓	✓
		0	P0INT	0	H0	R		✓	✓

0x4300–0x431e Universal Port Multiplexer (UPMUX)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
0x4300	P0UPMUX0 (P00–01 Universal Port Multiplexer Setting Register)	15–13	P01PPFNC[2:0]	0x0	H0	R/W	–	✓	✓
		12–11	P01PERICH[1:0]	0x0	H0	R/W		✓	✓
		10–8	P01PERISEL[2:0]	0x0	H0	R/W		✓	✓
		7–5	P00PPFNC[2:0]	0x0	H0	R/W		✓	✓
		4–3	P00PERICH[1:0]	0x0	H0	R/W		✓	✓
		2–0	P00PERISEL[2:0]	0x0	H0	R/W		✓	✓
0x4302	P0UPMUX1 (P02–03 Universal Port Multiplexer Setting Register)	15–13	P03PPFNC[2:0]	0x0	H0	R/W	–	✓	✓
		12–11	P03PERICH[1:0]	0x0	H0	R/W		✓	✓
		10–8	P03PERISEL[2:0]	0x0	H0	R/W		✓	✓
		7–5	P02PPFNC[2:0]	0x0	H0	R/W		✓	✓
		4–3	P02PERICH[1:0]	0x0	H0	R/W		✓	✓
		2–0	P02PERISEL[2:0]	0x0	H0	R/W		✓	✓
0x4304	P0UPMUX2 (P04–05 Universal Port Multiplexer Setting Register)	15–13	P05PPFNC[2:0]	0x0	H0	R/W	–	✓	✓
		12–11	P05PERICH[1:0]	0x0	H0	R/W		✓	✓
		10–8	P05PERISEL[2:0]	0x0	H0	R/W		✓	✓
		7–5	P04PPFNC[2:0]	0x0	H0	R/W		✓	✓
		4–3	P04PERICH[1:0]	0x0	H0	R/W		✓	✓
		2–0	P04PERISEL[2:0]	0x0	H0	R/W		✓	✓
0x4306	P0UPMUX3 (P06–07 Universal Port Multiplexer Setting Register)	15–13	P07PPFNC[2:0]	0x0	H0	R/W	–	✓	✓
		12–11	P07PERICH[1:0]	0x0	H0	R/W		✓	✓
		10–8	P07PERISEL[2:0]	0x0	H0	R/W		✓	✓
		7–5	P06PPFNC[2:0]	0x0	H0	R/W		✓	✓
		4–3	P06PERICH[1:0]	0x0	H0	R/W		✓	✓
		2–0	P06PERISEL[2:0]	0x0	H0	R/W		✓	✓
0x4308	P1UPMUX0 (P10–11 Universal Port Multiplexer Setting Register)	15–13	P11PPFNC[2:0]	0x0	H0	R/W	–	✓	✓
		12–11	P11PERICH[1:0]	0x0	H0	R/W		✓	✓
		10–8	P11PERISEL[2:0]	0x0	H0	R/W		✓	✓
		7–5	P10PPFNC[2:0]	0x0	H0	R/W		✓	✓
		4–3	P10PERICH[1:0]	0x0	H0	R/W		✓	✓
		2–0	P10PERISEL[2:0]	0x0	H0	R/W		✓	✓
0x430a	P1UPMUX1 (P12–13 Universal Port Multiplexer Setting Register)	15–13	P13PPFNC[2:0]	0x0	H0	R/W	–	✓	✓
		12–11	P13PERICH[1:0]	0x0	H0	R/W		✓	✓
		10–8	P13PERISEL[2:0]	0x0	H0	R/W		✓	✓
		7–5	P12PPFNC[2:0]	0x0	H0	R/W		✓	✓
		4–3	P12PERICH[1:0]	0x0	H0	R/W		✓	✓
		2–0	P12PERISEL[2:0]	0x0	H0	R/W		✓	✓
0x430c	P1UPMUX2 (P14–15 Universal Port Multiplexer Setting Register)	15–13	P15PPFNC[2:0]	0x0	H0	R/W	–	✓	✓
		12–11	P15PERICH[1:0]	0x0	H0	R/W		✓	✓
		10–8	P15PERISEL[2:0]	0x0	H0	R/W		✓	✓
		7–5	P14PPFNC[2:0]	0x0	H0	R/W		✓	✓
		4–3	P14PERICH[1:0]	0x0	H0	R/W		✓	✓
		2–0	P14PERISEL[2:0]	0x0	H0	R/W		✓	✓
0x430e	P1UPMUX3 (P16–17 Universal Port Multiplexer Setting Register)	15–13	P17PPFNC[2:0]	0x0	H0	R/W	–	✓	✓
		12–11	P17PERICH[1:0]	0x0	H0	R/W		✓	✓
		10–8	P17PERISEL[2:0]	0x0	H0	R/W		✓	✓
		7–5	P16PPFNC[2:0]	0x0	H0	R/W		✓	✓
		4–3	P16PERICH[1:0]	0x0	H0	R/W		✓	✓
		2–0	P16PERISEL[2:0]	0x0	H0	R/W		✓	✓

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M02	M03
0x4310	P2UPMUX0 (P20–21 Universal Port Multiplexer Setting Register)	15–13	P21PPFNC[2:0]	0x0	H0	R/W	–	–	✓
		12–11	P21PERICH[1:0]	0x0	H0	R/W		–	✓
		10–8	P21PERISEL[2:0]	0x0	H0	R/W		–	✓
		7–5	P20PPFNC[2:0]	0x0	H0	R/W		–	✓
		4–3	P20PERICH[1:0]	0x0	H0	R/W		–	✓
		2–0	P20PERISEL[2:0]	0x0	H0	R/W		–	✓
0x4312	P2UPMUX1 (P22–23 Universal Port Multiplexer Setting Register)	15–13	P23PPFNC[2:0]	0x0	H0	R/W	–	–	✓
		12–11	P23PERICH[1:0]	0x0	H0	R/W		–	✓
		10–8	P23PERISEL[2:0]	0x0	H0	R/W		–	✓
		7–5	P22PPFNC[2:0]	0x0	H0	R/W		–	✓
		4–3	P22PERICH[1:0]	0x0	H0	R/W		–	✓
		2–0	P22PERISEL[2:0]	0x0	H0	R/W		–	✓
0x4314	P2UPMUX2 (P24–25 Universal Port Multiplexer Setting Register)	15–13	P25PPFNC[2:0]	0x0	H0	R/W	–	–	✓
		12–11	P25PERICH[1:0]	0x0	H0	R/W		–	✓
		10–8	P25PERISEL[2:0]	0x0	H0	R/W		–	✓
		7–5	P24PPFNC[2:0]	0x0	H0	R/W		–	✓
		4–3	P24PERICH[1:0]	0x0	H0	R/W		–	✓
		2–0	P24PERISEL[2:0]	0x0	H0	R/W		–	✓
0x4316	P2UPMUX3 (P26–27 Universal Port Multiplexer Setting Register)	15–13	P27PPFNC[2:0]	0x0	H0	R/W	–	–	✓
		12–11	P27PERICH[1:0]	0x0	H0	R/W		–	✓
		10–8	P27PERISEL[2:0]	0x0	H0	R/W		–	✓
		7–5	P26PPFNC[2:0]	0x0	H0	R/W		–	✓
		4–3	P26PERICH[1:0]	0x0	H0	R/W		–	✓
		2–0	P26PERISEL[2:0]	0x0	H0	R/W		–	✓
0x4318	P3UPMUX0 (P30–31 Universal Port Multiplexer Setting Register)	15–13	P31PPFNC[2:0]	0x0	H0	R/W	–	–	✓
		12–11	P31PERICH[1:0]	0x0	H0	R/W		–	✓
		10–8	P31PERISEL[2:0]	0x0	H0	R/W		–	✓
		7–5	P30PPFNC[2:0]	0x0	H0	R/W		–	✓
		4–3	P30PERICH[1:0]	0x0	H0	R/W		–	✓
		2–0	P30PERISEL[2:0]	0x0	H0	R/W		–	✓
0x431a	P3UPMUX1 (P32–33 Universal Port Multiplexer Setting Register)	15–13	P33PPFNC[2:0]	0x0	H0	R/W	–	–	✓
		12–11	P33PERICH[1:0]	0x0	H0	R/W		–	✓
		10–8	P33PERISEL[2:0]	0x0	H0	R/W		–	✓
		7–5	P32PPFNC[2:0]	0x0	H0	R/W		–	✓
		4–3	P32PERICH[1:0]	0x0	H0	R/W		–	✓
		2–0	P32PERISEL[2:0]	0x0	H0	R/W		–	✓
0x431c	P3UPMUX2 (P34–35 Universal Port Multiplexer Setting Register)	15–13	P35PPFNC[2:0]	0x0	H0	R/W	–	–	✓
		12–11	P35PERICH[1:0]	0x0	H0	R/W		–	✓
		10–8	P35PERISEL[2:0]	0x0	H0	R/W		–	✓
		7–5	P34PPFNC[2:0]	0x0	H0	R/W		–	✓
		4–3	P34PERICH[1:0]	0x0	H0	R/W		–	✓
		2–0	P34PERISEL[2:0]	0x0	H0	R/W		–	✓
0x431e	P3UPMUX3 (P36–37 Universal Port Multiplexer Setting Register)	15–13	P37PPFNC[2:0]	0x0	H0	R/W	–	–	✓
		12–11	P37PERICH[1:0]	0x0	H0	R/W		–	✓
		10–8	P37PERISEL[2:0]	0x0	H0	R/W		–	✓
		7–5	P36PPFNC[2:0]	0x0	H0	R/W		–	✓
		4–3	P36PERICH[1:0]	0x0	H0	R/W		–	✓
		2–0	P36PERISEL[2:0]	0x0	H0	R/W		–	✓

0x4380–0x4390 UART (UART3) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4380	UA0CLK (UART3 Ch.0 Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	
		7–6	–	0x0	–	R	
		5–4	CLKDIV[1:0]	0x0	H0	R/W	
		3–2	–	0x0	–	R	
		1–0	CLKSRC[1:0]	0x0	H0	R/W	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4382	UA0MOD (UART3 Ch.0 Mode Register)	15–13	–	0x0	–	R	–
		12	PECAR	0	H0	R/W	
		11	CAREN	0	H0	R/W	
		10	BRDIV	0	H0	R/W	
		9	INVRX	0	H0	R/W	
		8	INVTX	0	H0	R/W	
		7	–	0	–	R	
		6	PUEN	0	H0	R/W	
		5	OUTMD	0	H0	R/W	
		4	IRMD	0	H0	R/W	
		3	CHLN	0	H0	R/W	
		2	PREN	0	H0	R/W	
		1	PRMD	0	H0	R/W	
		0	STPB	0	H0	R/W	
0x4384	UA0BR (UART3 Ch.0 Baud-Rate Register)	15–12	–	0x0	–	R	–
		11–8	FMD[3:0]	0x0	H0	R/W	
		7–0	BRT[7:0]	0x00	H0	R/W	
0x4386	UA0CTL (UART3 Ch.0 Control Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1	SFTRST	0	H0	R/W	
		0	MODEN	0	H0	R/W	
0x4388	UA0TXD (UART3 Ch.0 Transmit Data Register)	15–8	–	0x00	–	R	–
		7–0	TXD[7:0]	0x00	H0	R/W	
0x438a	UA0RXD (UART3 Ch.0 Receive Data Register)	15–8	–	0x00	–	R	–
		7–0	RXD[7:0]	0x00	H0	R	
0x438c	UA0INTF (UART3 Ch.0 Status and Interrupt Flag Register)	15–10	–	0x00	–	R	–
		9	RBSY	0	H0/S0	R	
		8	TBSY	0	H0/S0	R	
		7	–	0	–	R	
		6	TENDIF	0	H0/S0	R/W	Cleared by writing 1.
		5	FEIF	0	H0/S0	R/W	Cleared by writing 1 or reading the UA0RXD register.
		4	PEIF	0	H0/S0	R/W	
		3	OEIF	0	H0/S0	R/W	Cleared by writing 1.
		2	RB2FIF	0	H0/S0	R	Cleared by reading the UA0RXD register.
		1	RB1FIF	0	H0/S0	R	
		0	TBEIF	1	H0/S0	R	Cleared by writing to the UA0TXD register.
0x438e	UA0INTE (UART3 Ch.0 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7	–	0	–	R	
		6	TENDIE	0	H0	R/W	
		5	FEIE	0	H0	R/W	
		4	PEIE	0	H0	R/W	
		3	OEIE	0	H0	R/W	
		2	RB2FIE	0	H0	R/W	
		1	RB1FIE	0	H0	R/W	
		0	TBEIE	0	H0	R/W	
0x4390	UA0CAWF (UART3 Ch.0 Carrier Waveform Register)	15–8	–	0x00	–	R	–
		7–0	CRPER[7:0]	0x00	H0	R/W	

0x43a0–0x43ac 16-bit Timer (T16) Ch.1

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x43a0	T16_1CLK (T16 Ch.1 Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	
		7–4	CLKDIV[3:0]	0x0	H0	R/W	
		3–2	–	0x0	–	R	
		1–0	CLKSRC[1:0]	0x0	H0	R/W	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x43a2	T16_1MOD (T16 Ch.1 Mode Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	TRMD	0	H0	R/W	
0x43a4	T16_1CTL (T16 Ch.1 Control Register)	15–9	–	0x00	–	R	–
		8	PRUN	0	H0	R/W	
		7–2	–	0x00	–	R	
		1	PRESET	0	H0	R/W	
		0	MODEN	0	H0	R/W	
0x43a6	T16_1TR (T16 Ch.1 Reload Data Register)	15–0	TR[15:0]	0xffff	H0	R/W	–
0x43a8	T16_1TC (T16 Ch.1 Counter Data Register)	15–0	TC[15:0]	0xffff	H0	R	–
0x43aa	T16_1INTF (T16 Ch.1 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIF	0	H0	R/W	Cleared by writing 1.
0x43ac	T16_1INTE (T16 Ch.1 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIE	0	H0	R/W	

0x43b0–0x43ba Synchronous Serial Interface (SPIA) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x43b0	SPI0MOD (SPIA Ch.0 Mode Register)	15–12	–	0x0	–	R	–
		11–8	CHLN[3:0]	0x7	H0	R/W	
		7–6	–	0x0	–	R	
		5	PUEN	0	H0	R/W	
		4	NOCLKDIV	0	H0	R/W	
		3	LSBFST	0	H0	R/W	
		2	CPHA	0	H0	R/W	
		1	CPOL	0	H0	R/W	
		0	MST	0	H0	R/W	
0x43b2	SPI0CTL (SPIA Ch.0 Control Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1	SFTRST	0	H0	R/W	
		0	MODEN	0	H0	R/W	
0x43b4	SPI0TXD (SPIA Ch.0 Transmit Data Register)	15–0	TXD[15:0]	0x0000	H0	R/W	–
0x43b6	SPI0RXD (SPIA Ch.0 Receive Data Register)	15–0	RXD[15:0]	0x0000	H0	R	–
0x43b8	SPI0INTF (SPIA Ch.0 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7	BSY	0	H0	R	
		6–4	–	0x0	–	R	
		3	OEIF	0	H0/S0	R/W	Cleared by writing 1.
		2	TENDIF	0	H0/S0	R/W	
		1	RBFIF	0	H0/S0	R	Cleared by reading the SPI0RXD register.
		0	TBEIF	1	H0/S0	R	Cleared by writing to the SPI0TXD register.
0x43ba	SPI0INTE (SPIA Ch.0 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3	OEIE	0	H0	R/W	
		2	TENDIE	0	H0	R/W	
		1	RBFIE	0	H0	R/W	
		0	TBEIE	0	H0	R/W	

0x43c0–0x43d2 I²C (I2C) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x43c0	I2C0CLK (I2C Ch.0 Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	
		7–6	–	0x0	–	R	
		5–4	CLKDIV[1:0]	0x0	H0	R/W	
		3–2	–	0x0	–	R	
		1–0	CLKSRC[1:0]	0x0	H0	R/W	
0x43c2	I2C0MOD (I2C Ch.0 Mode Register)	15–8	–	0x00	–	R	–
		7–3	–	0x00	–	R	
		2	OADR10	0	H0	R/W	
		1	GCEN	0	H0	R/W	
		0	–	0	–	R	
0x43c4	I2C0BR (I2C Ch.0 Baud-Rate Register)	15–8	–	0x00	–	R	–
		7	–	0	–	R	
		6–0	BRT[6:0]	0x7f	H0	R/W	
0x43c8	I2C0OADR (I2C Ch.0 Own Address Register)	15–10	–	0x00	–	R	–
		9–0	OADR[9:0]	0x000	H0	R/W	
0x43ca	I2C0CTL (I2C Ch.0 Control Register)	15–8	–	0x00	–	R	–
		7–6	–	0x0	–	R	
		5	MST	0	H0	R/W	
		4	TXNACK	0	H0/S0	R/W	
		3	TXSTOP	0	H0/S0	R/W	
		2	TXSTART	0	H0/S0	R/W	
		1	SFTRST	0	H0	R/W	
		0	MODEN	0	H0	R/W	
0x43cc	I2C0TXD (I2C Ch.0 Transmit Data Register)	15–8	–	0x00	–	R	–
		7–0	TXD[7:0]	0x00	H0	R/W	
0x43ce	I2C0RXD (I2C Ch.0 Receive Data Register)	15–8	–	0x00	–	R	–
		7–0	RXD[7:0]	0x00	H0	R	
0x43d0	I2C0INTF (I2C Ch.0 Status and Interrupt Flag Register)	15–13	–	0x0	–	R	–
		12	SDALOW	0	H0	R	
		11	SCLLOW	0	H0	R	
		10	BSY	0	H0/S0	R	
		9	TR	0	H0	R	
		8	–	0	–	R	
		7	BYTEENDIF	0	H0/S0	R/W	Cleared by writing 1.
		6	GCIF	0	H0/S0	R/W	
		5	NACKIF	0	H0/S0	R/W	
		4	STOPIF	0	H0/S0	R/W	
		3	STARTIF	0	H0/S0	R/W	
		2	ERRIF	0	H0/S0	R/W	
		1	RBFIF	0	H0/S0	R	Cleared by reading the I2C0RXD register.
		0	TBEIF	0	H0/S0	R	Cleared by writing to the I2C0TXD register.
0x43d2	I2C0INTE (I2C Ch.0 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7	BYTEENDIE	0	H0	R/W	
		6	GCIE	0	H0	R/W	
		5	NACKIE	0	H0	R/W	
		4	STOPIE	0	H0	R/W	
		3	STARTIE	0	H0	R/W	
		2	ERRIE	0	H0	R/W	
		1	RBFIE	0	H0	R/W	
		0	TBEIE	0	H0	R/W	

0x5000–0x501a DMM 16-bit PWM Timer (T16B_DMM) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5000	T16B0CLK (T16B_DMM Ch.0 Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	
		7–4	CLKDIV[3:0]	0x0	H0	R/W	
		3	–	0	–	R	
		2–0	CLKSRC[2:0]	0x0	H0	R/W	
0x5002	T16B0CTL (T16B_DMM Ch.0 Counter Control Register)	15–9	–	0x00	–	R	–
		8	MAXBSY	0	H0	R	
		7–6	–	0x0	–	R	
		5–4	CNTMD[1:0]	0x0	H0	R/W	
		3	ONEST	0	H0	R/W	
		2	RUN	0	H0	R/W	
		1	PRESET	0	H0	R/W	
		0	MODEN	0	H0	R/W	
0x5004	T16B0MC (T16B_DMM Ch.0 Max Counter Data Register)	15–0	MC[15:0]	0xffff	H0	R/W	–
0x5006	T16B0TC (T16B_DMM Ch.0 Timer Counter Data Register)	15–0	TC[15:0]	0x0000	H0	R	–
0x5008	T16B0CS (T16B_DMM Ch.0 Counter Status Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3	CAPI1	0	H0	R	
		2	CAPI0	0	H0	R	
		1	UP_DOWN	1	H0	R	
		0	BSY	0	H0	R	
0x500a	T16B0INTF (T16B_DMM Ch.0 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–6	–	0x0	–	R	
		5	CAPOW1IF	0	H0	R/W	Cleared by writing 1.
		4	CMPCAP1IF	0	H0	R/W	
		3	CAPOW0IF	0	H0	R/W	
		2	CMPCAP0IF	0	H0	R/W	
		1	CNTMAXIF	0	H0	R/W	
		0	CNTZEROIF	0	H0	R/W	
0x500c	T16B0INTE (T16B_DMM Ch.0 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–6	–	0x0	–	R	
		5	CAPOW1IE	0	H0	R/W	
		4	CMPCAP1IE	0	H0	R/W	
		3	CAPOW0IE	0	H0	R/W	
		2	CMPCAP0IE	0	H0	R/W	
		1	CNTMAXIE	0	H0	R/W	
		0	CNTZEROIE	0	H0	R/W	
0x5010	T16B0CCCTL0 (T16B_DMM Ch.0 Compare/Capture 0 Control Register)	15	SCS	0	H0	R/W	–
		14–12	CBUFMD[2:0]	0x0	H0	R/W	
		11–10	CAPIS[1:0]	0x0	H0	R/W	
		9–8	CAPTRG[1:0]	0x0	H0	R/W	
		7	–	0	–	R	
		6	TOUTMT	0	H0	R/W	
		5	TOUTO	0	H0	R/W	
		4–2	TOUTMD[2:0]	0x0	H0	R/W	
		1	TOUTINV	0	H0	R/W	
		0	CCMD	0	H0	R/W	
0x5012	T16B0CCR0 (T16B_DMM Ch.0 Compare/Capture 0 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5018	T16B0CCCTL1 (T16B_DMM Ch.0 Compare/Capture 1 Control Register)	15	SCS	0	H0	R/W	–
		14–12	CBUFMD[2:0]	0x0	H0	R/W	
		11–10	CAPIS[1:0]	0x0	H0	R/W	
		9–8	CAPTRG[1:0]	0x0	H0	R/W	
		7	–	0	–	R	
		6	TOUTMT	0	H0	R/W	
		5	TOUTO	0	H0	R/W	
		4–2	TOUTMD[2:0]	0x0	H0	R/W	
		1	TOUTINV	0	H0	R/W	
		0	CCMD	0	H0	R/W	
0x501a	T16B0CCR1 (T16B_DMM Ch.0 Compare/Capture 1 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–

0x5040–0x505a DMM 16-bit PWM Timer (T16B_DMM) Ch.1

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5040	T16B1CLK (T16B_DMM Ch.1 Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	
		7–4	CLKDIV[3:0]	0x0	H0	R/W	
		3	–	0	–	R	
		2–0	CLKSRC[2:0]	0x0	H0	R/W	
0x5042	T16B1CTL (T16B_DMM Ch.1 Counter Control Register)	15–9	–	0x00	–	R	–
		8	MAXBSY	0	H0	R	
		7–6	–	0x0	–	R	
		5–4	CNTMD[1:0]	0x0	H0	R/W	
		3	ONEST	0	H0	R/W	
		2	RUN	0	H0	R/W	
		1	PRESET	0	H0	R/W	
		0	MODEN	0	H0	R/W	
0x5044	T16B1MC (T16B_DMM Ch.1 Max Counter Data Register)	15–0	MC[15:0]	0xffff	H0	R/W	–
0x5046	T16B1TC (T16B_DMM Ch.1 Timer Counter Data Register)	15–0	TC[15:0]	0x0000	H0	R	–
0x5048	T16B1CS (T16B_DMM Ch.1 Counter Status Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3	CAPI1	0	H0	R	
		2	CAPI0	0	H0	R	
		1	UP_DOWN	1	H0	R	
		0	BSY	0	H0	R	
0x504a	T16B1INTF (T16B_DMM Ch.1 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–6	–	0x0	–	R	
		5	CAPOW1IF	0	H0	R/W	Cleared by writing 1.
		4	CMPCAP1IF	0	H0	R/W	
		3	CAPOW0IF	0	H0	R/W	
		2	CMPCAP0IF	0	H0	R/W	
		1	CNTMAXIF	0	H0	R/W	
		0	CNTZEROIF	0	H0	R/W	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x504c	T16B1INTE (T16B_DMM Ch.1 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–6	–	0x0	–	R	
		5	CAPOW1IE	0	H0	R/W	
		4	CMPCAP1IE	0	H0	R/W	
		3	CAPOW0IE	0	H0	R/W	
		2	CMPCAP0IE	0	H0	R/W	
		1	CNTMAXIE	0	H0	R/W	
		0	CNTZEROIE	0	H0	R/W	
0x5050	T16B1CCCTL0 (T16B_DMM Ch.1 Compare/Capture 0 Control Register)	15	SCS	0	H0	R/W	–
		14–12	CBUFMD[2:0]	0x0	H0	R/W	
		11–10	CAPIS[1:0]	0x0	H0	R/W	
		9–8	CAPTRG[1:0]	0x0	H0	R/W	
		7	–	0	–	R	
		6	TOUTMT	0	H0	R/W	
		5	TOUTO	0	H0	R/W	
		4–2	TOUTMD[2:0]	0x0	H0	R/W	
		1	TOUTINV	0	H0	R/W	
		0	CCMD	0	H0	R/W	
0x5052	T16B1CCR0 (T16B_DMM Ch.1 Compare/Capture 0 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–
0x5058	T16B1CCCTL1 (T16B_DMM Ch.1 Compare/Capture 1 Control Register)	15	SCS	0	H0	R/W	–
		14–12	CBUFMD[2:0]	0x0	H0	R/W	
		11–10	CAPIS[1:0]	0x0	H0	R/W	
		9–8	CAPTRG[1:0]	0x0	H0	R/W	
		7	–	0	–	R	
		6	TOUTMT	0	H0	R/W	
		5	TOUTO	0	H0	R/W	
		4–2	TOUTMD[2:0]	0x0	H0	R/W	
		1	TOUTINV	0	H0	R/W	
		0	CCMD	0	H0	R/W	
0x505a	T16B1CCR1 (T16B_DMM Ch.1 Compare/Capture 1 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–

0x5080–0x509a DMM 16-bit PWM Timer (T16B_DMM) Ch.2

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5080	T16B2CLK (T16B_DMM Ch.2 Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	
		7–4	CLKDIV[3:0]	0x0	H0	R/W	
		3	–	0	–	R	
		2–0	CLKSRC[2:0]	0x0	H0	R/W	
0x5082	T16B2CTL (T16B_DMM Ch.2 Counter Control Register)	15–9	–	0x00	–	R	–
		8	MAXBSY	0	H0	R	
		7–6	–	0x0	–	R	
		5–4	CNTMD[1:0]	0x0	H0	R/W	
		3	ONEST	0	H0	R/W	
		2	RUN	0	H0	R/W	
		1	PRESET	0	H0	R/W	
		0	MODEN	0	H0	R/W	
0x5084	T16B2MC (T16B_DMM Ch.2 Max Counter Data Register)	15–0	MC[15:0]	0xffff	H0	R/W	–

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5086	T16B2TC (T16B_DMM Ch.2 Timer Counter Data Register)	15–0	TC[15:0]	0x0000	H0	R	–
0x5088	T16B2CS (T16B_DMM Ch.2 Counter Status Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3	CAPI1	0	H0	R	
		2	CAPI0	0	H0	R	
		1	UP_DOWN	1	H0	R	
		0	BSY	0	H0	R	
0x508a	T16B2INTF (T16B_DMM Ch.2 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–6	–	0x0	–	R	
		5	CAPOW1IF	0	H0	R/W	Cleared by writing 1.
		4	CMPCAP1IF	0	H0	R/W	
		3	CAPOW0IF	0	H0	R/W	
		2	CMPCAP0IF	0	H0	R/W	
		1	CNTMAXIF	0	H0	R/W	
		0	CNTZEROIF	0	H0	R/W	
0x508c	T16B2INTE (T16B_DMM Ch.2 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–6	–	0x0	–	R	
		5	CAPOW1IE	0	H0	R/W	
		4	CMPCAP1IE	0	H0	R/W	
		3	CAPOW0IE	0	H0	R/W	
		2	CMPCAP0IE	0	H0	R/W	
		1	CNTMAXIE	0	H0	R/W	
		0	CNTZEROIE	0	H0	R/W	
0x5090	T16B2CCCTL0 (T16B_DMM Ch.2 Compare/Capture 0 Control Register)	15	SCS	0	H0	R/W	–
		14–12	CBUFMD[2:0]	0x0	H0	R/W	
		11–10	CAPIS[1:0]	0x0	H0	R/W	
		9–8	CAPTRG[1:0]	0x0	H0	R/W	
		7	–	0	–	R	
		6	TOUTMT	0	H0	R/W	
		5	TOUTO	0	H0	R/W	
		4–2	TOUTMD[2:0]	0x0	H0	R/W	
		1	TOUTINV	0	H0	R/W	
		0	CCMD	0	H0	R/W	
0x5092	T16B2CCR0 (T16B_DMM Ch.2 Compare/Capture 0 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–
0x5098	T16B2CCCTL1 (T16B_DMM Ch.2 Compare/Capture 1 Control Register)	15	SCS	0	H0	R/W	–
		14–12	CBUFMD[2:0]	0x0	H0	R/W	
		11–10	CAPIS[1:0]	0x0	H0	R/W	
		9–8	CAPTRG[1:0]	0x0	H0	R/W	
		7	–	0	–	R	
		6	TOUTMT	0	H0	R/W	
		5	TOUTO	0	H0	R/W	
		4–2	TOUTMD[2:0]	0x0	H0	R/W	
		1	TOUTINV	0	H0	R/W	
		0	CCMD	0	H0	R/W	
0x509a	T16B2CCR1 (T16B_DMM Ch.2 Compare/Capture 1 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–

0x5260–0x526c 16-bit Timer (T16) Ch.2

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5260	T16_2CLK (T16 Ch.2 Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	
		7–4	CLKDIV[3:0]	0x0	H0	R/W	
		3–2	–	0x0	–	R	
		1–0	CLKSRC[1:0]	0x0	H0	R/W	
0x5262	T16_2MOD (T16 Ch.2 Mode Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	TRMD	0	H0	R/W	
0x5264	T16_2CTL (T16 Ch.2 Control Register)	15–9	–	0x00	–	R	–
		8	PRUN	0	H0	R/W	
		7–2	–	0x00	–	R	
		1	PRESET	0	H0	R/W	
		0	MODEN	0	H0	R/W	
0x5266	T16_2TR (T16 Ch.2 Reload Data Register)	15–0	TR[15:0]	0xffff	H0	R/W	–
0x5268	T16_2TC (T16 Ch.2 Counter Data Register)	15–0	TC[15:0]	0xffff	H0	R	–
0x526a	T16_2INTF (T16 Ch.2 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIF	0	H0	R/W	Cleared by writing 1.
0x526c	T16_2INTE (T16 Ch.2 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIE	0	H0	R/W	

0x5300–0x530a Sound Generator (SNDA_DMM)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5300	SNDCLK (SNDA_DMM Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	
		7	–	0	–	R	
		6–4	CLKDIV[2:0]	0x0	H0	R/W	
		3–2	–	0x0	–	R	
		1–0	CLKSRC[1:0]	0x0	H0	R/W	
0x5302	SNDSEL (SNDA_DMM Select Register)	15–12	–	0x0	–	R	–
		11–8	STIM[3:0]	0x0	H0	R/W	
		7–4	–	0x0	–	R	
		3	CONT	0	H0	R/W	
		2	SINV	0	H0	R/W	
		1–0	MOSEL[1:0]	0x0	H0	R/W	
0x5304	SNDCTL (SNDA_DMM Control Register)	15–9	–	0x00	–	R	–
		8	SSTP	0	H0	R/W	
		7–1	–	0x00	–	R	
		0	MODEN	0	H0	R/W	
0x5306	SNDDAT (SNDA_DMM Data Register)	15	MDTI	0	H0	R/W	–
		14	MDRS	0	H0	R/W	
		13–8	SLEN[5:0]	0x00	H0	R/W	
		7–0	SFRQ[7:0]	0xff	H0	R/W	
0x5308	SNDINTF (SNDA_DMM Interrupt Flag Register)	15–9	–	0x00	–	R	–
		8	SBSY	0	H0	R	
		7–2	–	0x00	–	R	
		1	EMIF	1	H0	R	Cleared by writing to the SNDDAT register.
		0	EDIF	0	H0	R/W	Cleared by writing 1 or writing to the SNDDAT register.

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x530a	SNDINTE (SNDA_DMM Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1	EMIE	0	H0	R/W	
		0	EDIE	0	H0	R/W	

0x5400–0x5412 LCD Driver (LCD4B)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5400	LCD4CLK (LCD4B Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	1	H0	R/W	
		7	–	0	–	R	
		6–4	CLKDIV[2:0]	0x0	H0	R/W	
		3–2	–	0x0	–	R	
		1–0	CLKSRC[1:0]	0x0	H0	R/W	
0x5402	LCD4CTL (LCD4B Control Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1	LCDDIS	0	H0	R/W	
		0	MODEN	0	H0	R/W	
0x5404	LCD4TIM1 (LCD4B Timing Control Register 1)	15–12	–	0x0	–	R	–
		11–8	FRMCNT[3:0]	0x7	H0	R/W	
		7–2	–	0x00	–	R	
		1–0	LDUTY[1:0]	0x3	H0	R/W	
0x5406	LCD4TIM2 (LCD4B Timing Control Register 2)	15–10	–	0x00	–	R	–
		9–8	BSTC[1:0]	0x1	H0	R/W	
		7–2	–	0x00	–	R	
		1–0	NLINE[1:0]	0x0	H0	R/W	
0x5408	LCD4PWR (LCD4B Power Control Register)	15	EXVCSEL	1	H0	R/W	–
		14–13	RESISEL[1:0]	0x0	H0	R/W	
		12–8	LC[4:0]	0x00	H0	R/W	
		7–5	–	0x0	–	R	
		4	BSTEN	0	H0	R/W	
		3	–	0	–	R	
		2	HVLD	0	H0	R/W	
		1	VCSEL	0	H0	R/W	
		0	VCEN	0	H0	R/W	
0x540a	LCD4DSP (LCD4B Display Control Register)	15–8	–	0x00	–	R	–
		7	–	0	–	R	
		6	SEGREV	1	H0	R/W	
		5	COMREV	1	H0	R/W	
		4	DSPREV	1	H0	R/W	
		3	–	0	–	R	
		2	DSPAR	0	H0	R/W	
		1–0	DSPC[1:0]	0x0	H0	R/W	
0x540c	LCD4COMC0 (LCD4B COM Pin Control Register 0)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3	COM3DEN	1	H0	R/W	
		2	COM2DEN	1	H0	R/W	
		1	COM1DEN	1	H0	R/W	
		0	COM0DEN	1	H0	R/W	
0x5410	LCD4INTF (LCD4B Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	FRMIF	0	H0	R/W	Cleared by writing 1.
0x5412	LCD4INTE (LCD4B Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	FRMIE	0	H0	R/W	

0x5500–0x553e DMM Controller (DSADC16)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5500	DSADC16CLK (DSADC16 Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	
		7–6	–	0x0	–	R	
		5–4	CLKDIV[1:0]	0x0	H0	R/W	
		3–2	–	0	–	R	
		1–0	CLKSRC[1:0]	0x0	H0	R/W	
0x5502	DSADC16CONF (DSADC16 Configuration Register)	15–14	–	0x0	–	R	–
		13–12	AVE_NUM[1:0]	0x0	H0	R/W	
		11–8	OSR[3:0]	0x0	H0	R/W	
		7	TRUERMS_ON	0	H0	R/W	
		6	XHPF_LPF	0	H0	R/W	
		5	PEAK_ON	0	H0	R/W	
		4	XABS_SQUARE	0	H0	R/W	
		3–1	–	0	–	R	
		0	MODEN	0	H0	R/W	
0x5504	DSADC16CTL (DSADC16 Control Register)	15–8	–	0x00	–	R	–
		7	–	0	–	R	
		6–4	RANGESEL[2:0]	0x0	H0	R/W	
		3–0	FUNCSEL[3:0]	0x0	H0	R/W	
0x5506	DSADC16INIT (DSADC16 Initialize Control Register)	15–10	–	0x00	–	R	–
		9	PEAKRST	0	H0	R/W	
		8	FILTERRST	0	H0	R/W	
		7–2	–	0x00	–	R	
		1	RUN	0	H0	R/W	
		0	–	0	–	R	
0x5508	DSADC16IE (DSADC16 Interrupt Enable Register)	15–11	–	0x00	–	R	–
		10	CONTIE	0	H0	R/W	
		9	TRUERMS_OVRIE	0	H0	R/W	
		8	TRUERMSIE	0	H0	R/W	
		7	RMS_OVRIE	0	H0	R/W	
		6	HPF_OVRIE	0	H0	R/W	
		5	LPF_OVRIE	0	H0	R/W	
		4	COMB_OVRIE	0	H0	R/W	
		3	RMSIE	0	H0	R/W	
		2	HPFIE	0	H0	R/W	
		1	LPFIE	0	H0	R/W	
		0	COMBIE	0	H0	R/W	
0x550a	DSADC16IF (DSADC16 Interrupt Flag Register)	15	TRUERMS	0	H0	R	–
		14	RMS	0	H0	R	
		13	HPF	0	H0	R	
		12	LPF	0	H0	R	
		11	COMB	0	H0	R	
		10	CONTIF	0	H0	R/W	Cleared by writing 1.
		9	TRUERMS_OVRIF	0	H0	R/W	
		8	TRUERMSIF	0	H0	R/W	
		7	RMS_OVRIF	0	H0	R/W	
		6	HPF_OVRIF	0	H0	R/W	
		5	LPF_OVRIF	0	H0	R/W	
		4	COMB_OVRIF	0	H0	R/W	
		3	RMSIF	0	H0	R/W	
		2	HPFIF	0	H0	R/W	
		1	LPFIF	0	H0	R/W	
		0	COMBIF	0	H0	R/W	
0x550c	DSADC16COMB (DSADC16 Comb Filter Result Register)	15–0	COMB_RESULT[15:0]	0x0000	H0	R	–

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x550e	DSADC16LPF (DSADC16 Low Pass/High Pass Filter Result Register)	15–0	LPFHPF_RESULT[15:0]	0x0000	H0	R	–
0x5510	DSADC16RMS1 (DSADC16 RMS Result Register 1)	15–0	RMS_RESULT[15:0]	0x0000	H0	R	–
0x5512	DSADC16RMS2 (DSADC16 RMS Result Register 2)	15–0	RMS_RESULT[31:16]	0x0000	H0	R	–
0x5514	DSADC16 DCPEAKMAX (DSADC16 DC Peak Hold MAX Result Register)	15–0	DCPEAKMAX[15:0]	0x8000	H0	R	–
0x5516	DSADC16 DCPEAKMIN (DSADC16 DC Peak Hold MIN Result Register)	15–0	DCPEAKMIN[15:0]	0x7fff	H0	R	–
0x5518	DSADC16 ACPEAKMAX1 (DSADC16 AC Peak Hold MAX Result Register 1)	15–0	ACPEAKMAX[15:0]	0x0000	H0	R	–
0x551a	DSADC16 ACPEAKMAX2 (DSADC16 AC Peak Hold MAX Result Register 2)	15–0	ACPEAKMAX[31:16]	0x0000	H0	R	–
0x551c	DSADC16 ACPEAKMIN1 (DSADC16 AC Peak Hold MIN Result Register 1)	15–0	ACPEAKMIN[15:0]	0x0000	H0	R	–
0x551e	DSADC16 ACPEAKMIN2 (DSADC16 AC Peak Hold MIN Result Register 2)	15–0	ACPEAKMIN[31:16]	0x0000	H0	R	–
0x5520	VIRCTL (VIR Control Register)	15–12	(reserved)	0x0	H0	R/WP	Always set to 0x0.
		11	–	0	–	R	–
		10–9	(reserved)	0x0	H0	R/WP	Always set to 0x0.
		8	DMM_XRESET	0	H0	R/W	–
		7–6	(reserved)	0x3	H0	R/WP	Always set to 0x3.
		5–3	(reserved)	0x0	H0	R/WP	Always set to 0x0.
		2	(reserved)	1	H0	R/WP	Always set to 1.
		1	(reserved)	0	H0	R/WP	Always set to 0.
		0	VIR_EN	0	H0	R/W	–
0x5522	DMMSET1 (DMM Setting Register 1)	15	–	0	–	R	–
		14–8	DMM_SET_IODS[6:0]	0x00	H0	R/W	
		7	–	0	–	R	
		6–0	DMM_SET_IOPS[6:0]	0x00	H0	R/W	
0x5524	DMMSET2 (DMM Setting Register 2)	15	–	0	–	R	–
		14–8	DMM_SET_IOSS[6:0]	0x00	H0	R/W	
		7	–	0	–	R	
		6–0	DMM_SET_IOFS[6:0]	0x00	H0	R/W	
0x5526	SMODESET (DMM SMODE Setting Register)	15	–	0	–	R	–
		14–12	DMM_SET_CMPIN[2:0]	0x0	H0	R/W	
		11–8	DMM_SET_MAMPVR[3:0]	0x0	H0	R/W	
		7	–	0	–	R	
		6–0	DMM_SET_SMODE[6:0]	0x00	H0	R/W	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5528	AFENET1 (AFE Network Setting Register 1)	15	–	0	–	R	–
		14–12	DMM_SET_FLTINN[2:0]	0x0	H0	R/W	
		11–8	DMM_SET_FLTINP[3:0]	0x0	H0	R/W	
		7–4	DMM_SET_CMPRL[3:0]	0x0	H0	R/W	
		3–0	DMM_SET_CMPRH[3:0]	0x0	H0	R/W	
0x552a	AFENET2 (AFE Network Setting Register 2)	15	–	0	–	R	–
		14–12	DMM_SET_ADVRL[2:0]	0x0	H0	R/W	
		11	–	0	–	R	
		10–8	DMM_SET_ADVRH[2:0]	0x0	H0	R/W	
		7–3	–	0x00	–	R	
		2–0	DMM_SET_AGNDV[2:0]	0x0	H0	R/W	
0x552c	AFENET3 (AFE Network Setting Register 3)	15–14	–	0x0	–	R	–
		13–12	DMM_SET_ FLTMODE[1:0]	0x0	H0	R/W	
		11	DMM_SET_DIOSW	0	H0	R/W	
		10	DMM_EN_CMP	0	H0	R/W	
		9	DMM_EN_AGND	0	H0	R/W	
		8	DMM_EN_MAMP	0	H0	R/W	
		7–6	–	0x0	–	R	
		5–4	(reserved)	0x0	H0	R/WP	Always set to 0x0.
		3–2	–	0x0	–	R	–
		1–0	(reserved)	0x0	H0	R/WP	Always set to 0x0.
0x552e	CHPCTL (Chopper Amp Control Register)	15–13	CHP_SET_CHP[2:0]	0x0	H0	R/W	–
		12	(reserved)	0	H0	R/WP	Always set to 0.
		11	–	0	–	R	–
		10–8	CHP_SET_GAIN[2:0]	0x0	H0	R/W	
		7–5	CHP_SET_BIAS[2:0]	0x0	H0	R/W	
		4–1	(reserved)	0x0	H0	R/WP	Always set to 0x0.
		0	CHP_EN	0	H0	R/W	–
0x5530	DSMVCMCTL Sigma Delta Modulator & ADCVCM Control Register)	15	–	0	–	R	–
		14–12	(reserved)	0x0	H0	R/WP	Always set to 0x0.
		11	–	0	–	R	–
		10	VCM_SET_VRMD	0	H0	R/W	
		9	VCM_EN_REF	0	H0	R/W	
		8	VCM_EN_ACM	0	H0	R/W	
		7–6	–	0x0	–	R	
		5–1	(reserved)	0x00	H0	R/WP	Always set to 0x00.
		0	DSM_EN	0	H0	R/W	–
0x5532	TSRVRTEMP (TSRVR Temperature Correction Data Register)	15–0	TSRVR_TEMP[15:0]	0x0000	H0	R	–
0x5534	TSRVR (TSRVR Control Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1	(reserved)	0	H0	R/WP	Always set to 0.
		0	TSR_EN	0	H0	R/W	–
0x5536	CMPOUT (Comparator Output Status Register)	15–9	–	0	H0	R	–
		8	DMM_OUT_CMP	0	H0	R	
		7–1	–	0	H0	R	
		0	(reserved)	1	H0	R/WP	Always set to 1.
0x553c	DCDCCTL (DCDC Control Register)	15–11	(reserved)	0x00	H0	R/WP	Always set to 0x00.
		10	–	0	–	R	–
		9	(reserved)	1	H0	R/W	Always set to 1.
		8	LDO_CP_ON	1	H0	R/W	–
		7–6	–	0x0	–	R	
		5–1	(reserved)	0x00	H0	R/WP	Always set to 0x00.
		0	CHG_EN	0	H0	R/W	–

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x553e	AFESUB (AFE Sub-control Register)	15–9	–	0x00	H0	R	–
		8–5	OHMCTL[3:0]	0x0	H0	R/WP	
		4-2	CHP_SET_BIAS2[2:0]	0x0	H0	R/WP	
		1–0	(reserved)	0x0	H0	R/WP	Always set to 0x0.

0x5580–0x558c 16-bit Timer (T16) Ch.3

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5580	T16_3CLK (T16 Ch.3 Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	
		7–4	CLKDIV[3:0]	0x0	H0	R/W	
		3–2	–	0x0	–	R	
		1–0	CLKSRC[1:0]	0x0	H0	R/W	
0x5582	T16_3MOD (T16 Ch.3 Mode Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	TRMD	0	H0	R/W	
0x5584	T16_3CTL (T16 Ch.3 Control Register)	15–9	–	0x00	–	R	–
		8	PRUN	0	H0	R/W	
		7–2	–	0x00	–	R	
		1	PRESET	0	H0	R/W	
		0	MODEN	0	H0	R/W	
0x5586	T16_3TR (T16 Ch.3 Reload Data Register)	15–0	TR[15:0]	0xffff	H0	R/W	–
0x5588	T16_3TC (T16 Ch.3 Counter Data Register)	15–0	TC[15:0]	0xffff	H0	R	–
0x558a	T16_3INTF (T16 Ch.3 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIF	0	H0	R/W	Cleared by writing 1.
0x558c	T16_3INTE (T16 Ch.3 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIE	0	H0	R/W	

0xffff90 Debugger (DBG)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0xffff90	DBRAM (Debug RAM Base Register)	31–24	–	0x00	–	R	–
		23–0	DBRAM[23:0]	0x00 07c0	H0	R	

Appendix B Power Saving

Current consumption will vary dramatically, depending on CPU operating mode, operation clock frequency, peripheral circuits being operated, and power generator operating mode. Listed below are the control methods for saving power.

B.1 Operating Status Configuration Examples for Power Saving

Table B.1.1 lists typical examples of operating status configuration with consideration given to power saving.

Table B.1.1 Typical Operating Status Configuration Examples

Operating status configuration	Current consumption	V_{D1}	OSC1	IOSC/ OSC3/ EXOSC	RTCA	CPU	Current consumption listed in electrical characteristics
Standby	↑ Low	Economy	OFF	OFF	OFF	SLEEP	I_{SLP}
Clock counting			ON		ON	SLEEP or HALT	I_{HALT2}
Low-speed processing						OSC1 RUN	I_{RUN2}
Peripheral circuit operations		Normal		ON		SLEEP or HALT	I_{HALT1}
High-speed processing	High ↓					IOSC/OSC3/EXOSC RUN	I_{RUN1}

If the current consumption order by the operating status configuration shown in Table B.1.1 is different from one that is listed in "Electrical Characteristics," check the settings shown below.

PWGVD1CTL.PWGMOD[1:0] bits of the power generator

If the PWGVD1CTL.PWGMOD[1:0] bits of the power generator is 0x2 (normal mode) when the CPU enters SLEEP mode, current consumption in SLEEP mode will be larger than I_{SLP} that is listed in "Electrical Characteristics." Set the PWGVD1CTL.PWGMOD[1:0] bits to 0x3 (economy mode) or 0x0 (automatic mode) before executing the slp instruction.

CLGOSC.IOSCSLPC/OSC1SLPC/OSC3SLPC/EXOSCSLPC bits of the clock generator

Setting the CLGOSC.IOSCSLPC, OSC1SLPC, OSC3SLPC, or EXOSCSLPC bit of the clock generator to 0 disables the oscillator circuit stop control when the slp instruction is executed. To stop the oscillator circuits during SLEEP mode, set these bits to 1.

MODEN bits of the peripheral circuits

Setting the MODEN bit of each peripheral circuit to 1 starts supplying the operating clock enabling the peripheral circuit to operate. To reduce current consumption, set the MODEN bits of unnecessary peripheral circuits to 0. Note that the real-time clock has no MODEN bit, therefore, current consumption does not vary if it is counting or idle.

OSC1 oscillator circuit configurations

The OSC1 oscillator circuit provides some configuration items to support various crystal resonators with ranges from cylinder type through surface-mount type. These configurations trade off current consumption for performance as shown below.

- The lower oscillation inverter gain setting (CLGOSC1.INV1B[1:0]/INV1N[1:0] bits) decreases current consumption.
- The lower OSC1 internal gate capacitance setting (CLGOSC1.CGI1[2:0] bits) decreases current consumption.
- Using lower OSC1 external gate and drain capacitances decreases current consumption.
- Using a crystal resonator with lower C_L value decreases current consumption.

However, these configurations may reduce the oscillation margin and increase the frequency error, therefore, be sure to perform matching evaluation using the actual printed circuit board.

B.2 Other Power Saving Methods

Supply voltage detector configuration

Continuous operation mode (SVDCTL.SVDMD[1:0] bits = 0x0) always detects the power supply voltage, therefore, it increases current consumption. Set the supply voltage detector to intermittent operation mode or turn it on only when required.

LCD driver configurations

- Setting the LCD voltage regulator to operate with the V_{C1} reference voltage (LCD4PWR.VCSEL bit = 0) increases current consumption. If a desired LCD drive voltage can be obtained, operating with V_{C2} reference voltage (LCD4PWR.VCSEL bit = 1) is recommended.
- The lower booster clock frequency setting (LCD4TIM2.BSTC[1:0] bits) for the LCD voltage booster decreases current consumption. Note, however, that the load characteristic becomes worse.
- Setting the LCD voltage regulator into heavy load protection mode (LCD4PWR.HVLD bit = 1) increases current consumption. Heavy load protection mode should be set only when the display becomes unstable.

Appendix C Mounting Precautions

This section describes various precautions for circuit board design and IC mounting.

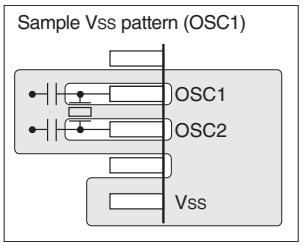
OSC1 oscillator circuit

- Oscillation characteristics depend on factors such as components used (resonator, C_G, C_D) and circuit board patterns. In particular, with crystal resonators, select the appropriate capacitors (C_G, C_D) only after fully evaluating components actually mounted on the circuit board.
- Oscillator clock disturbances caused by noise may cause malfunctions. To prevent such disturbances, consider the following points.

(1) Components such as a resonator, resistors, and capacitors connected to the OSC1 and OSC2 pins should have the shortest connections possible.

(2) Wherever possible, avoid locating digital signal lines within 3 mm of the OSC1 and OSC2 pins or related circuit components and wiring. Rapidly-switching signals, in particular, should be kept at a distance from these components. Since the spacing between layers of multi-layer printed circuit boards is a mere 0.1 mm to 0.2 mm, the above precautions also apply when positioning digital signal lines on other layers.
Never place digital signal lines alongside such components or wiring, even if more than 3 mm distance or located on other layers. Avoid crossing wires.

(3) Use V_{SS} to shield the OSC1 and OSC2 pins and related wiring (including wiring for adjacent circuit board layers). Layers wired should be adequately shielded as shown to the right. Fully ground adjacent layers, where possible. At minimum, shield the area at least 5 mm around the above pins and wiring.
Even after implementing these precautions, avoid configuring digital signal lines in parallel, as described in (2) above. Avoid crossing even on discrete layers, except for lines carrying signals with low switching frequencies.



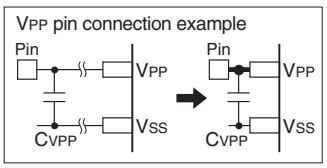
(4) After implementing these precautions, check the FOUT pin output clock waveform by running the actual application program within the product.
For the OSC1 waveform, enlarge the areas before and after the clock rising and falling edges and take special care to confirm that the regions approximately 100 ns to either side are free of clock or spiking noise.
Failure to observe precautions (1) to (3) adequately may lead to noise in OSC1CLK. Noise in the OSC1CLK will destabilize timers that use OSC1CLK as well as CPU Core operations.

#RESET pin

Components such as a switch and resistor connected to the #RESET pin should have the shortest connections possible to prevent noise-induced resets.

V_{PP} pin

Connect a capacitor C_{VPP} between the V_{SS} and V_{PP} pins to suppress fluctuations within V_{PP} ± 1 V. The C_{VPP} should be placed as close to the V_{PP} pin as possible and use a sufficiently thick wiring pattern that allows current of several tens of mA to flow.

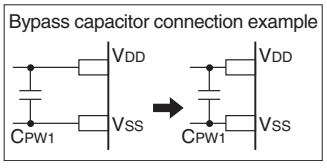


Power supply circuit

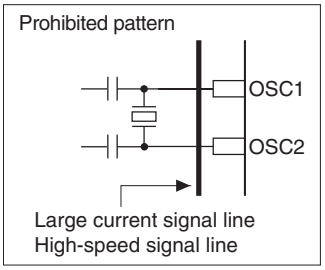
Sudden power supply fluctuations due to noise will cause malfunctions. Consider the following issues.

(1) Connections from the power supply to the V_{DD} and V_{SS} pins should be implemented via the shortest, thickest patterns possible.

(2) If a bypass capacitor is connected between V_{DD} and V_{SS}, connections between the V_{DD} and V_{SS} pins should be as short as possible.



Signal line location



- To prevent electromagnetically-induced noise arising from mutual induction, large-current signal lines should not be positioned close to pins susceptible to noise, such as oscillator and analog measurement pins.
- Locating signal lines in parallel over significant distances or crossing signal lines operating at high speed will cause malfunctions due to noise generated by mutual interference.

Unused pins

(1) I/O port (P) pins
Unused pins should be left open. The control registers should be fixed at the initial status.

(2) OSC1, OSC2, and EXOSC pins
If the OSC1 crystal oscillator circuit is not used, the OSC1 and OSC2 pins should be left open. If the EXOSC input circuit is not used, the pin should be configured as a general-purpose I/O port. The control registers should be fixed at the initial status (disabled).

(3) V_{C1-3}, C_{P1-2}, SEG*x*, and COM*x* pins
If the LCD driver is not used, the V_{C1-3} pins should be left open. The C_{P1-2} pins should be configured as general-purpose I/O ports. The control registers should be fixed at the initial status (display off). The unused SEG*x* and COM*x* pins that are not required to connect with the LCD panel should be configured as a general-purpose/peripheral circuit I/O port even if the LCD driver is used.

Miscellaneous

Minor variations over time may result in electrical damage arising from disturbances in the form of voltages exceeding the absolute maximum rating when mounting the product in addition to physical damage. The following factors can give rise to these variations:

(1) Electromagnetically-induced noise from industrial power supplies used in mounting reflow, reworking after mounting, and individual characteristic evaluation (testing) processes
(2) Electromagnetically-induced noise from a solder iron when soldering

In particular, during soldering, take care to ensure that the soldering iron GND (tip potential) has the same potential as the IC GND.

Appendix D Measures Against Noise

To improve noise immunity, take measures against noise as follows:

Noise Measures for V_{DD} and V_{SS} Power Supply Pins

When noise falling below the rated voltage is input, an IC malfunction may occur. If desired operations cannot be achieved, take measures against noise on the circuit board, such as designing close patterns for circuit board power supply circuits, adding noise-filtering decoupling capacitors, and adding surge/noise prevention components on the power supply line.
For the recommended patterns on the circuit board, see "Mounting Precautions" in Appendix.

Noise Measures for #RESET Pin

If noise is input to the #RESET pin, the IC may be reset. Therefore, the circuit board must be designed properly taking noise measures into consideration.
For the recommended patterns on the circuit board, see "Mounting Precautions" in Appendix.

Noise Measures for Oscillator Pins

The oscillator input pins must pass a signal of small amplitude, so they are hypersensitive to noise. Therefore, the circuit board must be designed properly taking noise measures into consideration.
For the recommended patterns on the circuit board, see "Mounting Precautions" in Appendix.

Noise Measures for Debug Pins

This product provides the input/output pins (DCLK, DST2, and DSIO) to connect ICDmini (S5U1C17001H) for debugging. If noise is input to these pins with the debugging function enabled, the S1C17 Core may enter DEBUG mode. To prevent unexpected transitions to DEBUG mode caused by extraneous noise, switch the DCLK, DST2, and DSIO pins to general-purpose I/O port pins within the initialization routine when the debug functions are not used.
For details of the pin functions and the function switch control, see the "I/O Ports" chapter.

Note: Do not perform the function switching shown above when the application is under development, as the debug functions must be used. The debugging cannot be performed after the pin function is switched. The above processing must be added after the application development has completed and debugging is no longer necessary.

The DSIO pin should be pulled up with a 10 kΩ resistor when using the debug pin functions.

Noise Measures for Interrupt Input Pins

This product is able to generate a port input interrupt when the input signal changes. The interrupt is generated when an input signal edge is detected, therefore, an interrupt may occur if the signal changes due to extraneous noise. To prevent occurrence of unexpected interrupts due to extraneous noise, enable the chattering filter circuit when using the port input interrupt.
For details of the port input interrupt and chattering filter circuit, see the "I/O Ports" chapter.

Noise Measures for UART Pins

This product includes a UART for asynchronous communications. The UART starts receive operation when it detects a low level input from the SIN*n* pin. Therefore, a receive operation may be started if the SIN*n* pin is set to low due to extraneous noise. In this case, a receive error will occur or invalid data will be received.
To prevent the UART from malfunction caused by extraneous noise, take the following measures:

- Stop the UART operations while asynchronous communication is not performed.
- Execute the resending process via software after executing the receive error handler with a parity check.

For details of the pin functions and the function switch control, see the "I/O Ports" chapter. For the UART control and details of receive errors, see the "UART" chapter.

Noise Measures for Input Pins Connected to Signal with High Driving Capability Such As Power Supply

There is a possibility of a large current flow into the pins that are directly connected to a power supply or an output of a device with high driving capability if noise is input to those pins. To prevent this, connect a 30 Ω or more pin protection resistor to the pins in series. The resistance value should be determined by evaluating it on the mounting board.

When connecting a power supply directly to the VREFA pin, insert a 100 Ω resistor in series. This resistance does not affect the A/D converter characteristics.

Appendix E Initialization Routine

The following lists typical vector tables and initialization routines:

boot.s

```
.org    0x8000
.section .rodata                                                          ...(1)

; ======================================================================
;       Vector table
; ======================================================================
                                ; interrupt  vector  interrupt
                                ; number     offset  source

.long BOOT                      ; 0x00       0x00    reset                ...(2)
.long unalign_handler           ; 0x01       0x04    unalign
.long nmi_handler               ; 0x02       0x08    NMI
.long int03_handler             ; 0x03       0x0c    -
.long svd4_handler              ; 0x04       0x10    SVD4
.long pport_handler             ; 0x05       0x14    PPORT
.long int06_handler             ; 0x06       0x18    -
.long clg_handler               ; 0x07       0x1c    CLG
.long int08_handler             ; 0x08       0x20    -
.long t16_0_handler             ; 0x09       0x24    T16 ch0
.long uart3_0_handler           ; 0x0a       0x28    UART3 ch0
.long t16_1_handler             ; 0x0b       0x2c    T16 ch1
.long spia_0_handler            ; 0x0c       0x30    SPIA ch0
.long i2c_0_handler             ; 0x0d       0x34    I2C ch0
.long t16b_dmm_0_handler        ; 0x0e       0x38    T16B_DMM ch0
.long t16b_dmm_1_handler        ; 0x0f       0x3c    T16B_DMM ch1
.long int10_handler             ; 0x10       0x40    -
.long snda_dmm_handler          ; 0x11       0x44    SNDA_DMM
.long int12_handler             ; 0x12       0x48    -
.long lcd4b_handler             ; 0x13       0x4c    LCD4B
.long int14_handler             ; 0x14       0x50    -
.long eepromc_handler           ; 0x15       0x54    EEPROMC
.long t16_2_handler             ; 0x16       0x58    T16 ch2
.long int17_handler             ; 0x17       0x5c    -
.long t16_3_handler             ; 0x18       0x60    T16 ch3
.long dsadc16_handler           ; 0x19       0x64    DSADC16
.long t16B_dmm_2_handler        ; 0x1a       0x68    T16B_DMM ch2
.long int1b_handler             ; 0x1b       0x6c    -
.long int1c_handler             ; 0x1c       0x70    -
.long int1d_handler             ; 0x1d       0x74    -
.long int1e_handler             ; 0x1e       0x78    -
.long int1f_handler             ; 0x1f       0x7c    -
; ======================================================================
;       Program code
; ======================================================================

.text                                                                     ...(3)
.align 1

BOOT:
        ; ===== Initialize ============================================

        ; ----- Stack pointer --------------------
        Xld.a   %sp, 0x7c0                                                ...(4)

        ; ----- Memory controller ----------------
        Xld.a   %r1, 0x41b0     ; FLASHC register address

        ; Flash read wait cycle
        Xld.a   %r0, 0x00       ; 0x00 = No wait
        ld.b    [%r1], %r0      ; [0x41b0] <= 0x00                        ...(5)

        ; ===== Main routine ==========================================
         ...
```

```
; ========================================================================
;       Interrupt handler
; ========================================================================
; ----- Address unalign --------------------------
unalign_handler:
        ...

; ----- NMI --------------------------------------
nmi_handler:
        ...
```

(1) A "`.rodata`" section is declared to locate the vector table in the "`.vector`" section.

(2) Interrupt handler routine addresses are defined as vectors.
"`int`*`XX`*`_handler`" can be used for software interrupts.

(3) The program code is written in the "`.text`" section.

(4) Sets the stack pointer.

(5) Sets the number of Flash memory read cycles.
(See the "Memory and Bus" chapter.)

Revision History

Code No.	Page	Contents
414282100	All	New establishment

EPSON

International Sales Operations

America

Epson America, Inc.

Headquarter:
3131 Katella Ave.,
Los Alamitos, CA 90720, USA
Phone: +1-800-463-7766

San Jose Office:
2860 Zanker Road Suite 204,
San Jose, CA 95134, USA
Phone: +1-800-463-7766

Europe

Epson Europe Electronics GmbH

Riesstrasse 15, 80992 Munich,
Germany
Phone: +49-89-14005-0 Fax: +49-89-14005-110

Asia

Epson (China) Co., Ltd.

4th Floor, Tower 1 of China Central Place, 81 Jianguo Road,
Chaoyang District, Beijing 100025, China
Phone: +86-10-8522-1199 Fax: +86-10-8522-1120

Shanghai Branch

Room 601-603, Building A One East, No. 325 East Longhua Road,
Shanghai 200023, China
Phone: +86-21-5330-4888 Fax: +86-21-5423-4677

Shenzhen Branch

Room 804-805, 8 Floor, Tower 2, Ali Center, No. 3331
Keyuan South RD (Shenzhen bay), Nanshan District,
Shenzhen 518054, China
Phone: +86-755-3299-0588 Fax: +86-755-3299-0560

Epson Taiwan Technology & Trading Ltd.

15F, No. 100, Songren Rd, Sinyi Dist, Taipei City 110, Taiwan
Phone: +886-2-8786-6688

Epson Singapore Pte., Ltd.

438B Alexandra Road,
Block B Alexandra TechnoPark, #04-01/04, Singapore 119968
Phone: +65-6586-5500 Fax: +65-6271-7066

Epson Korea Co., Ltd

10F Posco Tower Yeoksam, Teheranro 134 Gangnam-gu,
Seoul, 06235, Korea
Phone: +82-2-3420-6695

Seiko Epson Corp.
Sales & Marketing Division

MD Sales & Marketing Department

29th Floor, JR Shinjuku Miraina Tower, 4-1-6 Shinjuku,
Shinjuku-ku, Tokyo 160-8801, Japan

Document Code: 414282100
Issue March 2022 in JAPAN Ⓛ