Data Sheet

Mixed Signal MCU
MD6603

MD6603-DSE Rev.1.3
Jul. 14, 2021
SANKEN ELECTRIC CO., LTD.
https://www.sanken-ele.co.jp/en

Contents

1. Product Overview

- **Clock Control**

- Internal reference clock
- Built-in PLL
- Sleep mode/standby mode function

- **Reset Control**

- Pin reset
- Built-in power on reset circuit (POR)
- Watchdog timer reset

- **Reference Voltage (VREF)**

- 1.2 V

- **Low Voltage Detection (LVD)**

- LVD interrupt generation

- **8-bit CPU**

- Compatible with 8052 instruction
- 3-stage to 5-stage pipeline
- 60 MHz, 1 cycle per byte instruction execution
- 256-byte dedicated RAM

- **GPIO**

- Digital/analog function
- Pull-up/pull-down control
- Selectable interrupt sources

- **Event Processing Unit (EPU)**

- Multi-task 16-bit processor (switching zero-time task)
- 1 unit
- 6 threads
- 16 input/output events per thread
- 2-stage or 3-stage pipeline
- Program memory: 16 bits × 256 words

- **Interrupt Controller (INTC)**

- 32 interrupt sources
- Selectable priorities (2 levels)

- **Direct SFR Access Controller (DSAC)**

- Automatic data transfer between SFRs (SFR: Special Function Register)
- 16 channels
- 32 transfer request events
- Selectable transfer addressing
- 8-bit/16-bit data transfer
- Priority: DSAC > EPU > CPU

- **Built-in Flash Memory**

- 32 KB
- 8 bits per cycle access
- Protect functions
- Data flash memory support function

- **Built-in RAM**

- 1.75 KB

- **TinyDSP**

- 2 units
- 16-bit fixed point calculation
- Program memory: 48 steps
- Sixteen 16-bit data registers
- Eight 16-bit constant registers
- One 36-bit accumulator
- Instructions: Multiplication, division, multiply-accumulate calculation, shift calculation, move, jump, and minimum/maximum saturation
- Built-in divider (each unit)
- Sequence control synchronized with events
- An event is generated with arbitrary instruction
- 3-pole, 2-zero (3P2Z) IIR filter calculation: 10 cycles

- **High-resolution PWM**

- 4 channels (8 PWM outputs)
- 16-bit up/down counter control (each channel)
- Minimum resolution: 1.04 ns
- Duty setting range: 0% to 100%
- Selectable dead time
- Re-trigger operation by internal/external event

- **Watchdog Timer (WDT)**

- 8-bit counter
- Reset output or interrupt generation

● **16-bit Timer (TMR)**

- 4 channels
- 16-bit counter
- An event is generated by compare match
- Synchronized with the PWM
- A counter is cleared by an external pin event
- 4 phase count modes
- Buffer mode
- Input capture/compare match output

● **Serial Peripheral Interface (SPI)**

- 1 unit
- Master mode/subordinate mode communication
- Dedicated baud rate generator

● **I^2C/SMBUS**

- 1 unit
- Master mode/subordinate mode communication
- Dedicated baud rate generator
- GCA is supported
  (GCA: General Call Address)

● **UART**

- 1 unit
- Dedicated baud rate generator

● **12-bit SAR ADC**

- 2 units
- 12 analog inputs
- Conversion speed: 4 MSPS (max.)
- Sequence conversion
- Conversion start trigger is selectable
- Offset is set to the conversion result

● **Op amp (OPAMP)**

- 2 units
- Standalone mode/unity mode
  (×1 or ×4)

● **Comparator**

- 6 units
- Digital noise filter
- Event generation
- 8-bit DAC for reference voltage generation
  (each unit)

● **Temperature Sensor (TEMP)**

- 1 unit

● **PWM Output Controller (POC)**

- PWM output is controlled by the selected event such as comparator
- An output is controlled by the CPU
- Coupled operation with the PWM cycle

● **Comparator Lookup Table (CMPLUT)**

- 2 units
- Input signal: Comparator output (6 units)
- 6 inputs 1 output
- POC event input
- Pin output (LUT0/1)

● **Serial Communication Interface with Debugger**

- UART mode
  - Half-duplex asynchronous communication
- OCD mode
  - Debug function support
  - Flash memory program/erasing

● **Power Supply Voltage**

- DVCC and AVCC: 3.3 V
- Built-in regulator for core power supply

● **Package Dimension**

- QFN40
  (6 mm × 6 mm, pitch 0.5 mm)

2. Block Diagram

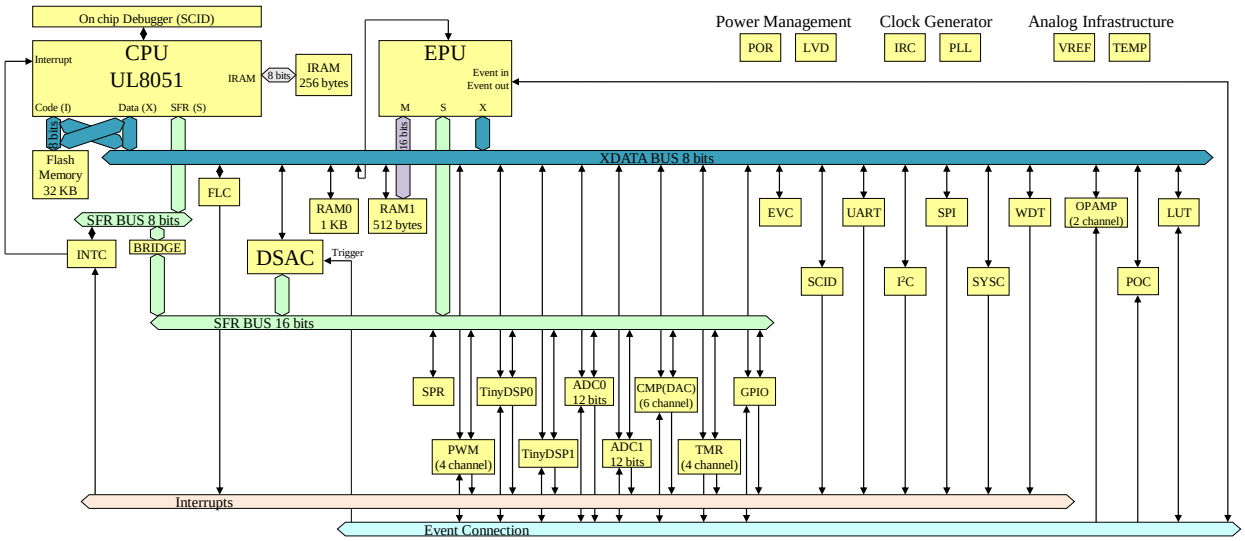


Figure 2-1. Block Diagram

3. Pin Configuration Definitions

3.1. Pin Configuration

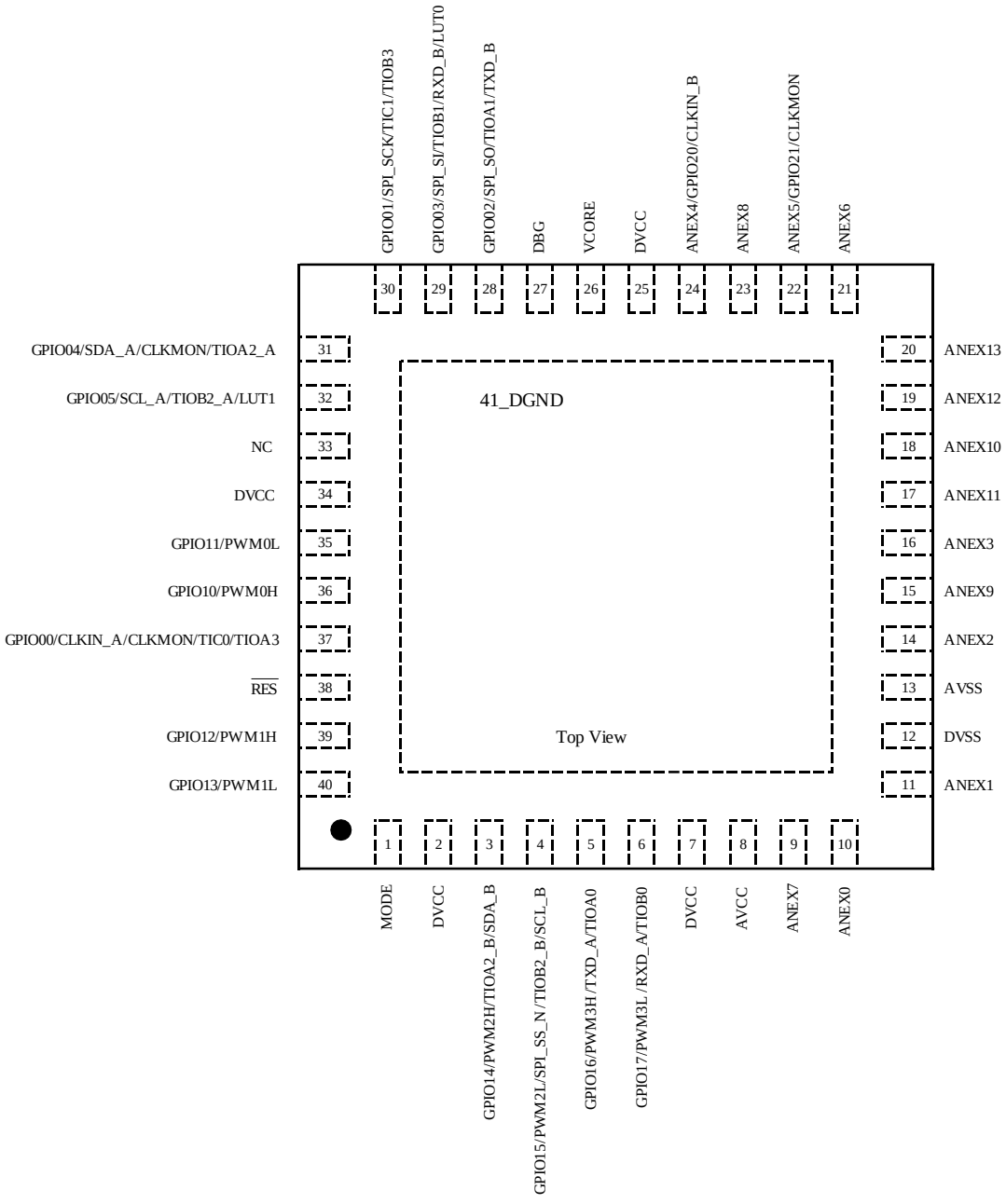


Figure 3-1. Pin Configuration of QFN40 (QFN41)

3.2. Pin Definitions (QFN40)

Classification	Pin No.	Pin Name	Input/ Output	Pull-up/ Pull-down	Description	Interrupt	5 V Tolerant	Schmitt	Logic Level	3.3 V I_{OUT}
Digital Power Supply	2 7 25 34	DVCC	—	—	Digital 3.3 V	—	—	—	—	—
	12 41	DVSS	—	—	Digital 0 V	—	—	—	—	—
Analog Power Supply	8	AVCC	—	—	Analog 3.3 V	—	—	—	—	—
	13	AVSS	—	—	Analog 0 V	—	—	—	—	—
System	1	MODE	Input	—	Chip mode (Fixed to 0)	—	—	Yes	LVTTL	—
	26	VCORE	—	—	Internal Power Supply (Stable capacitor connection)	—	—	—	—	—
	33	NC	—	—	—	—	—	—	—	—
	38	$\overline{\text{RES}}$	Input	Pull-up	Reset input	—	—	Yes	LVTTL	—
Debug	27	DBG	Input/ Output	Pull-up	1-wire OCD pin (Open drain)	—	Yes	—	LVTTL	4 mA
Serial/ TMR	37	GPIO00/ CLKIN_A/ CLKMON/ TIC0/ TIOA3	Input/ Output	Pull-up	GPIO, CLKIN, CLKMON, TIC0, TIOA3	Yes	Yes	—	LVTTL	4 mA
	30	GPIO01/ SPI_SCK/ TIC1/ TIOB3	Input/ Output	Pull-up	GPIO, SPI_SCK, TIC1, TIOB3	Yes	Yes	—	LVTTL	4 mA
	28	GPIO02/ SPI_SO/ TIOA1/ TXD_B	Input/ Output	Pull-up	GPIO, SPI_SO, TIOA1, TXD	Yes	Yes	—	LVTTL	4 mA
	29	GPIO03/ SPI_SI/ TIOB1/ RXD_B/ LUT0	Input/ Output	Pull-up	GPIO, SPI_SI, TIOB1, RXD, LUT0	Yes	Yes	—	LVTTL	4 mA
	31	GPIO04/ SDA_A/ CLKMON/ TIOA2_A	Input/ Output	Pull-up	GPIO, I^2C_SDA, CLKMON, TIOA2	Yes	Yes	—	LVTTL	4 mA
	32	GPIO05/ SCL_A/ TIOB2_A/ LUT1	Input/ Output	Pull-up	GPIO, I^2C_SCL, TIOB2, LUT1	Yes	Yes	—	LVTTL	4 mA
PWM/ Serial/ TMR	36	GPIO10/ PWM0H	Input/ Output	Pull-down	GPIO, PWM0H	Yes	Yes	—	LVTTL	16 mA
	35	GPIO11/ PWM0L	Input/ Output	Pull-down	GPIO, PWM0L	Yes	Yes	—	LVTTL	16 mA
	39	GPIO12/ PWM1H	Input/ Output	Pull-down	GPIO, PWM1H	Yes	Yes	—	LVTTL	16 mA
	40	GPIO13/ PWM1L	Input/ Output	Pull-down	GPIO, PWM1L	Yes	Yes	—	LVTTL	16 mA
	3	GPIO14/ PWM2H/ TIOA2_B/ SDA_B	Input/ Output	Pull-down	GPIO, PWM2H, TIOA2, I^2C_SDA	Yes	Yes	—	LVTTL	16 mA
	4	GPIO15/ PWM2L/ SPI_SS_N/ TIOB2_B/ SCL_B	Input/ Output	Pull-down	GPIO, PWM2L, SPI_SS_N, TIOB2, I^2C_SCL	Yes	Yes	—	LVTTL	16 mA
	5	GPIO16/ PWM3H/ TXD_A/ TIOA0	Input/ Output	Pull-down	GPIO, PWM3H, UART_TXD, TIOA0	Yes	Yes	—	LVTTL	16 mA
	6	GPIO17/ PWM3L/ RXD_A/ TIOB0	Input/ Output	Pull-down	GPIO, PWM3L, UART_RXD, TIOB0	Yes	Yes	—	LVTTL	16 mA
Analog/ TMR	24	ANEX4/ GPIO20/ CLKIN_B	Input/ Output	Pull-up	Analog pin 4, GPIO, CLKIN	Yes	—	—	LVTTL	4 mA
	22	ANEX5/ GPIO21/ CLKMON	Input/ Output	Pull-up	Analog pin 5, GPIO, CLKMON	Yes	—	—	LVTTL	4 mA

Classification	Pin No.	Pin Name	Input/ Output	Pull-up/ Pull-down	Description	Interrupt	5 V Tolerant	Schmitt	Logic Level	3.3 V I_{OUT}
Analog	10	ANEX0	Input/ Output	—	Analog pin 0	—	—	—	—	—
	11	ANEX1	Input/ Output	—	Analog pin 1	—	—	—	—	—
	14	ANEX2	Input/ Output	—	Analog pin 2	—	—	—	—	—
	16	ANEX3	Input/ Output	—	Analog pin 3	—	—	—	—	—
	21	ANEX6	Input/ Output	—	Analog pin 6	—	—	—	—	—
	9	ANEX7	Input/ Output	—	Analog pin 7	—	—	—	—	—
	23	ANEX8	Input/ Output	—	Analog pin 8	—	—	—	—	—
	15	ANEX9	Input/ Output	—	Analog pin 9	—	—	—	—	—
	18	ANEX10	Input/ Output	—	Analog pin 10	—	—	—	—	—
	17	ANEX11	Input/ Output	—	Analog pin 11	—	—	—	—	—
	19	ANEX12	Input/ Output	—	Analog pin 12	—	—	—	—	—
	20	ANEX13	Input/ Output	—	Analog pin 13	—	—	—	—	—

4. System Controller (SYSC)

4.1. Overview

The system controller (SYSC) controls all operation states of the chip. Clocks, reset operations, internal regulators, and the power consumption mode are integrally controlled.

Table 4-1. SYSC Functional Descriptions

Item		Description
Clock	Clock Source	● Internal reference clock (IRC): 12 MHz (max.) ● CLKIN: 12 MHz (max.) ● CLKSRC: 12 MHz (max.), IRC or CLKIN ● PLL: Multiply reference clock by 80 Maximum frequency: 960 MHz Reference clock: CLKSRC Reference clock frequency divided by 1 or 2 is selectable.
	Distribution Clock	● CLKFAST: 60 MHz (max.) Clock source: CLKSRC or PLL clock divided by 16; with 1, 2, 4, and 8 clock divider. ● CLKPWM: 120 MHz (max.) Clock source: CLKSRC or PLL clock divided by 8, 1 to 32 (2^n) clock divider for PLL clock.
Reset		● Power-on reset (POR) ● $\overline{\text{RES}}$ pin reset ● Watchdog timer (WDT) reset ● On chip debugger (OCD) reset
Regulator		● Internal regulator ● Reference voltage (VREF) ● Reference voltage for low voltage detection (LVD)
Power Consumption Control	Mode	● Sleep mode (returned by interrupt) ● Standby mode (with wakeup counter; returned by GPIO interrupt, CMP level interrupt,or LVD)
	Clock Enable/Disable	● Enable/Disable control of clock signal to each module
Analog Module Control		● IBIAS control for OPAMP (normal mode or low power consumption mode) ● VREF output control for low voltage detection (enabled or disabled) ● VREF output control for mesurement by ADC (enabled or disabled) ● GPIO20/GPIO21 extended pull-up resistor control
CPU BUS Buffer Control		● Buffer state clear for 16-bit XBUS register ● Buffer state clear for 16-bit SFR BUS register ● Access counter clear for 16-bit SFR BUS register

4.2. Clock System

Figure 4-1 shows the clock system of the LSI.

After a reset is released, the CLKIRC is used for the CLKSRC. The programmable CLKSRC can be changed to the clock that is input to the CLKIN pin.

The LSI has 2 types of clock (CLKFAST and CLKPWM). The CLKFAST is used for the peripheral modules other than the PWM. The CLKPWM is used for the high-resolution PWM. The maximum operation frequencies of the CLKFAST and the CLKPWM are 60 MHz and 120 MHz, respectively.

The IRC is also used for the baud rate generation of the one-wire OCD. The CLKSRC is used for the count clock of the standby mode wakeup counter.

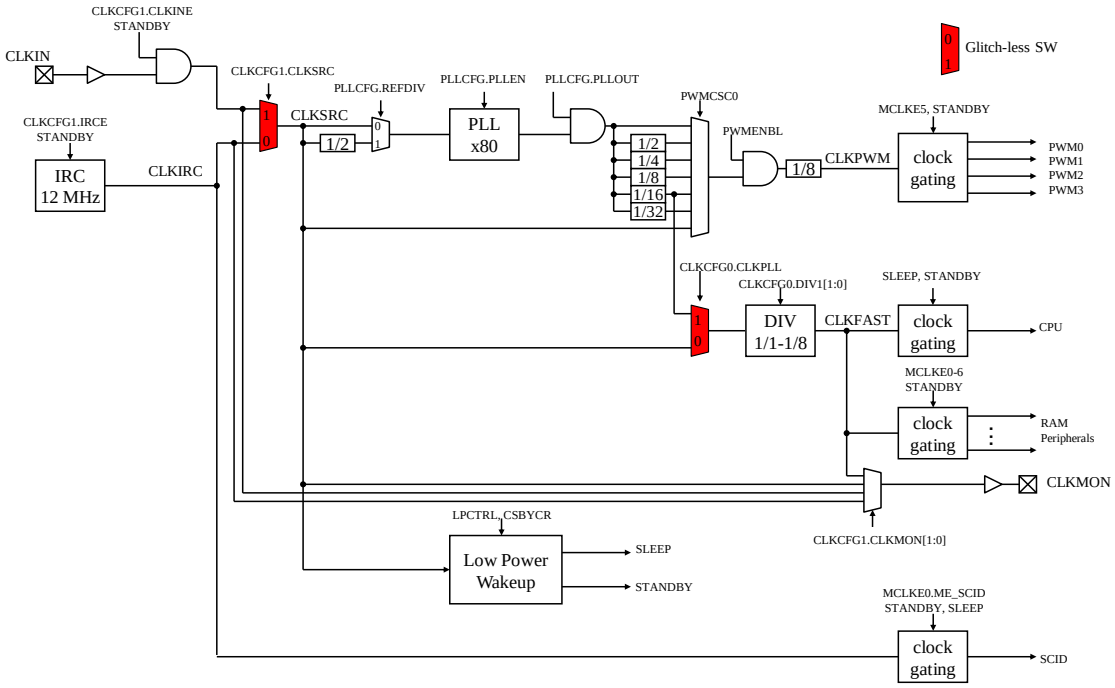


Figure 4-1. Clock System

4.2.1. Clock Sources

- **CLKIN**

This clock is supplied from the CLKIN pin, and is used for a system clock. The maximum input frequency to the CLKIN is 12 MHz. To supply the CLKIN from the CLKIN pin, set CLKCFG1.CLKINE = 1. To stop it, set CLKCFG1.CLKINE = 0.

- **CLKIRC**

This clock is supplied from the IRC, and is used for the system clock and the baud rate generation clock of the OCD. The maximum frequency of the CLKIRC is 12 MHz.

- **CLKSRC**

This clock is defined the CLKIN or the CLKIRC by the CLKCFG1.CLKSRC bit. After a reset is released, the CLKIRC is set for the CLKSRC. The CLKSRC is used for an internal clock source, the reference of the PLL clock, and the count clock of the standby mode wakeup counter. The selection of the CLKSRC can be dynamically changed by a glitchless swhich.

4.2.2. PLL Clock

The PLL clock generates the clock that is multiply reference clock source by 80. The reference clock source is the CLKSRC or the CLKSRC divided by 2. The maximum frequency of the PLL clock is 960 MHz. Enabling/disabling the PLL is determined by the PLLCFG.PLLEN bit. The divide rate of the reference clock is defined by the PLLCFG.REFDIV bit. The reference clock must be changed during PLLCFG.PLLEN = 0 (i.e., the PLL is disabled).

4.2.3. Distribution Clocks

● **CLKFAST**

This is used for the CPU and the peripheral modules other than the PWM. The clock source of the CLKFAST is determined by the CLKCFG0.CLKPLL bit, which defines the CLKSRC or the PLL clock divided by 16. In addition, the clock determined by the CLKCFG0.CLKPLL is divided by the selectable divide rate, and is supplied to the CLKFAST. The divide rate is determined by CLKCFG0.DIV1 bits, which defines 1, 2, 4, or 8. The maximum frequency of the CLKFAST is 60 MHz. The settings of the CLKCFG0.CLKPLL and CLKCFG0.DIV1 bits can be changed by a glitchless switch in operation.

● **CLKPWM**

This is used for the PWM. The CLKPWM is determined by the PWMCSC0.PWMCCA bits, which defines the CLKSRC or the PLL clock divided by 1, 2, 4, 8, 16, or 32. The maximum frequency of the CLKPWM is 120 MHz.
A PWM resolution is defined as the reciprocal of CLKPWM × 8; e.g., if CLKPWM = 120 MHz, the PWM resolution is 1.04 ns. To change the frequency of the CLKPWM while the clock is supplied to the PWM, the PWMENBL.PWMAE bit must be disabled (i.e., PWMENBL.PWMAE = 0).

4.2.4. Module Clocks

Table 4-2. List of Module Clocks

Clock Type	Initial State	Enable Control	Sleep Mode	Remarks
CPU	Enabled	No	Disabled	
BUS	Enabled	No	Disabled	
RAM0	Disabled	Yes	Enabled/Disabled	
RAM1	Disabled	Yes	Enabled/Disabled	
DSAC	Disabled	Yes	Enabled/Disabled	
EPU	Disabled	Yes	Enabled/Disabled	
GPIO	Disabled	Yes	Enabled/Disabled	
UART	Disabled	Yes	Enabled/Disabled	
I^2C	Disabled	Yes	Enabled/Disabled	
SPI	Disabled	Yes	Enabled/Disabled	
TMR0/1	Disabled	Yes	Enabled/Disabled	
TMR2/3	Disabled	Yes	Enabled/Disabled	
SCID	Enabled	Yes	Enabled/Disabled	Forcibly enabled by debugger connection. Without standby control*
TinyDSP0	Disabled	Yes	Enabled/Disabled	
TinyDSP1	Disabled	Yes	Enabled/Disabled	
ADC0	Disabled	Yes	Enabled/Disabled	
ADC1	Disabled	Yes	Enabled/Disabled	
SPR	Disabled	Yes	Enabled/Disabled	
CMP0	Disabled	Yes	Enabled/Disabled	
CMP1	Disabled	Yes	Enabled/Disabled	
CMP2	Disabled	Yes	Enabled/Disabled	
CMP3	Disabled	Yes	Enabled/Disabled	
CMP4	Disabled	Yes	Enabled/Disabled	
CMP5	Disabled	Yes	Enabled/Disabled	
AMP0	Disabled	Yes	Enabled/Disabled	
AMP1	Disabled	Yes	Enabled/Disabled	
CMPLUT	Disabled	Yes	Enabled/Disabled	
PWM0	Disabled	Yes	Enabled/Disabled	Without standby control*
PWM1	Disabled	Yes	Enabled/Disabled	Without standby control*
PWM2	Disabled	Yes	Enabled/Disabled	Without standby control*
PWM3	Disabled	Yes	Enabled/Disabled	Without standby control*
POC0	Disabled	Yes	Enabled/Disabled	Without standby control*
POC1	Disabled	Yes	Enabled/Disabled	Without standby control*
POC2	Disabled	Yes	Enabled/Disabled	Without standby control*
POC3	Disabled	Yes	Enabled/Disabled	Without standby control*
EVC	Disabled	Yes	Enabled/Disabled	

* The module clock is not automatically disabled at the transit to standby mode. The module clock must be set to disable by the MCLKEn register before the transit to the standby mode. To return from the standby mode, the module clock should be set to enable by the MCLKEn register.

4.2.5. Clock Setting Procedure

The following are the states in which the clocks of the LSI go into immediately after releasing the reset:

- IRC: Enabled, CLKIN pin input: Disabled, CLKSRC: The IRC is selected
- PLL: Disabled
- PLL reference input: CLKSRC/1
- CLKFAST: The CLKSRC divided by 8
- CLKPWM: Disabled

The clock setting procedure is as follows:

(1) To enable the CLKIN input, set CLKCFG1.CLKINE = 1.
(2) Set the CLKSRC to the IRC or the CLKIN by the PLLCFG.REFDIV bit.
(3) Set the devided rate of the reference clock for the PLL by PLLCFG.REFDIV bit.
(4) To enable the PLL, set PLLCFG.PLLEN = 1.
(5) Wait for the PLL Oscillation Stable Period. Then, to enable the PLL clock, set PLLCFG.PLLOUT = 1.
(6) Set the clock source of the CLKFAST to the CLKSRC or the PLL by the CLKCFG0.CLKPLL bit.
(7) Set the devided rate of the CLKFAST by the CLKCFG0.DIV1 bits.
(8) Set the clock type and the devided rate of the CLKPWM by the PWMCSC0 register.
(9) To enable the CLKPWM, set PWMENBL.PWMAE = 1.
(10) To supply the clock for peripheral functions, set the MCLKEn register.

When the CLKIN is not used, the settings of (1) and (2) are unnecessary. When the PLL is not used, the settings of (3), (4), and (5) are unnecessary. When the PWM is not used, the settings of (8) and (9) are unnecessary.

4.3. Low Power Consumption Modes

The LSI has 2 low power consumption modes: the sleep and standby modes.

4.3.1. Sleep Mode

The CPU clock stops during the sleep mode. To transit to the sleep mode, set LPCTRL.LPSE = 0, and then LPCTRL.GOTOLPM = 1. After the CPU completes accessing the flash memory, the LSI stops the CPU clock, and transits to the sleep mode. In the sleep mode, the clocks of peripheral functions operate according to the MCLKEn register setting. The following sources cause returning from the sleep mode: an interrupt and a reset. The sleep mode can be returned by all interrupts that is set INTENAn = 1 (i.e., enabled). Thus, the INTENAn register must be set before transition to the sleep mode.

The LSI does not transit to the sleep mode after the OCD communication access. The clock continues its operation and the processing.

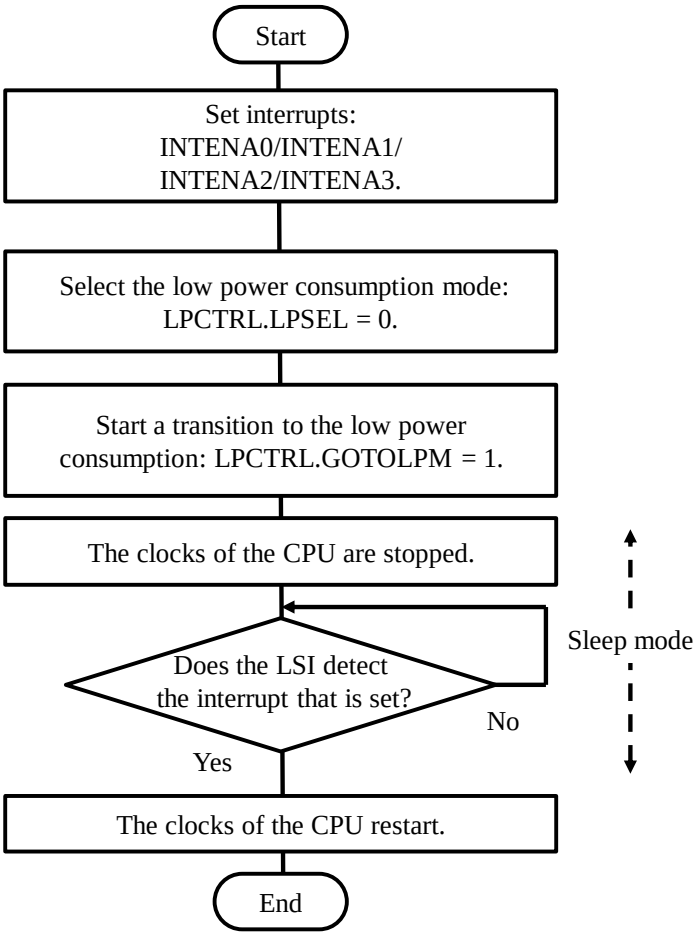


Figure 4-2. Sleep Mode Sequence

4.3.2. Standby Mode

All clocks (IRC, CLKIN, and PLL) in the LSI stop during the standby mode. To transit to the standby mode, set LPCTRL.LPSE = 1, and then LPCTRL.GOTOLPM = 1. After the CPU completes accessing the flash memory, the LSI stops all clocks in the LSI, and transits to the standby mode. The following sources cause returning from the standby mode: GPIO interrupt, CMP level interrupt, and low voltage detection (LVD). Thus, the INTENAn register must be set before transition to the standby mode. The sources by the CMP level interrupt and the LVD are defined by the CSBYCR register. When the sources to return the standby mode are detected, the IRC, CLKIN, and PLL are returned to the setting state before standby mode operation. Then, the wakeup counter that is counted by the CLKSRC operates. When the wakeup counter finishes counting, the clocks supply to the CPU and the peripheral functions again. The wakeup counter is defined by the LPCTRL.WUPTM bits.

The LSI does not transit to the standby mode after the OCD communication access. The clock continues its operation and the processing.

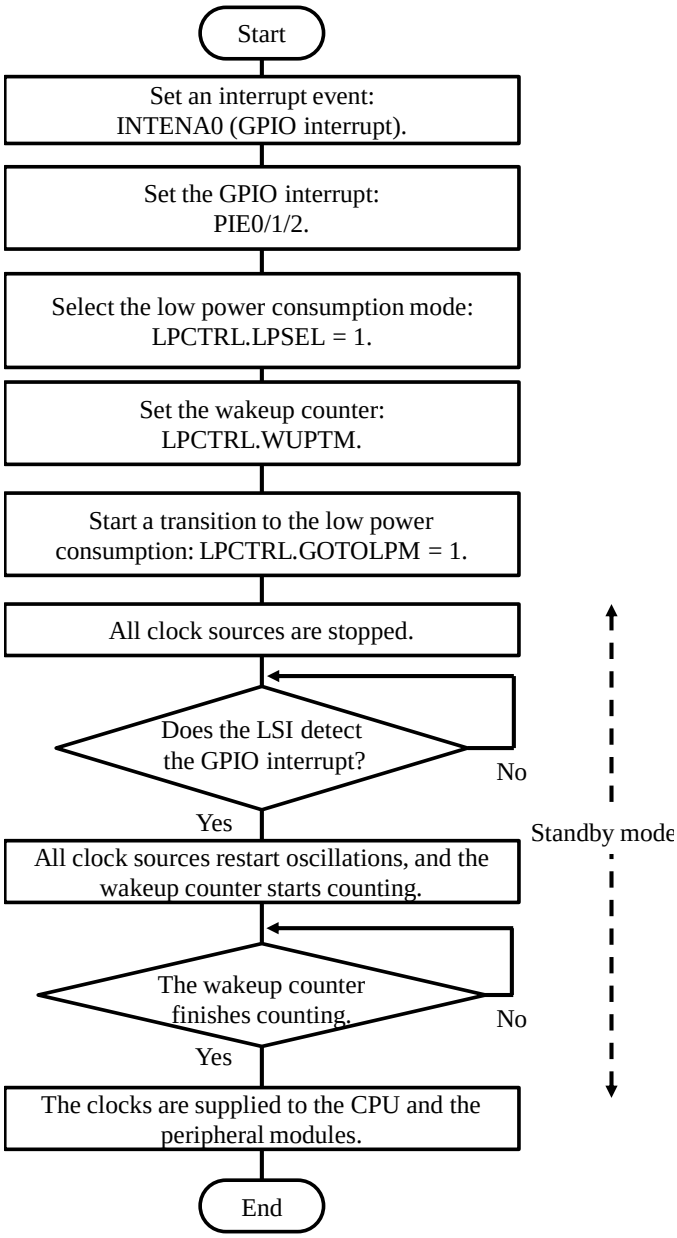


Figure 4-3. Standby Mode Sequence (Returned by GPIO Interrupt)

4.3.3. Inserting NOP Instruction at Transition to Sleep and Standby Modes

Insert 16 NOP instructions immediately after setting LPCTRL.GOTOLPM = 1 as follows:

```
LPCTRL |= 0x01;  // Go to STBY/SLEEP mode
__asm
                 nop
                 nop
                 nop
                 nop
                 nop
                 nop
                 nop
                 nop
                 nop
                 nop
                 nop
                 nop
                 nop
                 nop
                 nop
                 nop
__endasm;
```

4.4. Reset Circuit

Figure 4-4 shows the reset system.

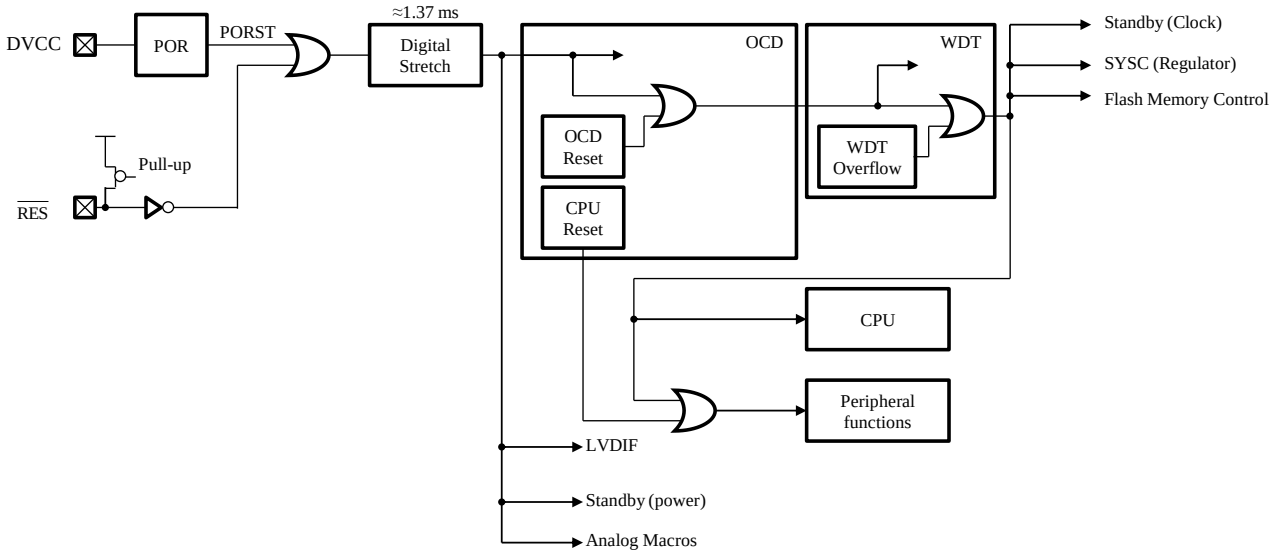


Figure 4-4. Reset System.

The LSI has the following reset types:

- **$\overline{\text{RES}}$ Pin**
  To reset all of the LSI setting, set the $\overline{\text{RES}}$ pin to a low state.

- **POR**
  The power-on reset, POR, is generated according to the voltage level of the DVCC pin. When a DVCC pin voltage decreases to the POR Detection Voltage, V_{PORL}, or less, the POR is generated. For release the POR, the DVCC pin voltage should increase to the POR Detection Voltage, V_{PORH}, or more.

- **OCD Reset**
  All the modules other than the OCD are reset by a command from the OCD.

- **WDT Reset**
  This reset is the generated at overflowing the watchdog timer, WDT. The period of the WDT reset is 64 cycles defined by the system clock.

- **CPU Reset**
  All the modules including the CPU other than the OCD and the WDT are reset by a command from the OCD.

The internal reset period is extended by a digital stretch for about 1.37 ms (IRC = 12 MHz) after releasing both of the $\overline{\text{RES}}$ pin reset and POR. The DVCC pin voltage and the IRC frequency must be stability in this extended period.

When the reset of the $\overline{\text{RES}}$ pin is not used, Open the $\overline{\text{RES}}$ pin, or fix the $\overline{\text{RES}}$ pin to the level of the DVCC pin voltage.

Table 4-3 shows the reset type and the reset area.

Table 4-3. Reset Type and Reset Area

Reset Type	OCD	WDT[1]	LVD Flag[2]	SYSC	The Others
$\overline{\text{RES}}$ Pin	Reset	Reset	Reset	Reset	Reset
POR	Reset	Reset	Reset	Reset	Reset
OCD Reset (All LSI Setting)	Not reset	Reset	Reset	Reset	Reset
WDT Reset	Not reset	Not reset	Not reset	Reset	Reset
CPU Reset	Not reset	Not reset	Not reset	Not reset	Reset

[1] The WTCNT register and the WTCSR.WOVF bit are reset. The other registers for WDT are reset by the WDT reset.
[2] The LVDCTRL.LVDIF bit is reset. The other bits in the LVDCTRL register is reset same as the SYSC.

4.5. Low Voltage Detection (LVD)

The low voltage detection, LVD, generates an interrupt request to the CPU when a DVCC pin voltage drop is detected. To use the LVD, set REFCTRL.VREF120AEN= 1 (i.e., the VREF voltage for the LVD is enabled), and then set LVDCTRL.LVDE = 1. When the DVCC pin voltage decreases below the LVD voltage, the LVDCTRL.LVDIF bit becomes 1. In addition, to generate the interrupt request to the CPU, set LVDCTRL.LVDIE = 1. To clear the LVDCTRL.LVDIF bit, set the LVDCTRL.LVDIF = 1 while the DVCC pin voltage exceeds the LVD voltage. If the LVDCTRL.LVDIF bit is set to 1 while the DVCC pin voltage is below the LVD voltage, the LVDCTRL.LVDIF bit is not cleared.

The LSI can be returned from the standby mode by the LVD. To generate the interrupt request to the CPU at return from the standby mode, set LVDCTRL.LVDIE = 1.

It is required to wait at least 3 cycles before reading the LVDCTRL.LVDIF bit immediately after the LVDCTRL.LVDIF bit is cleared.

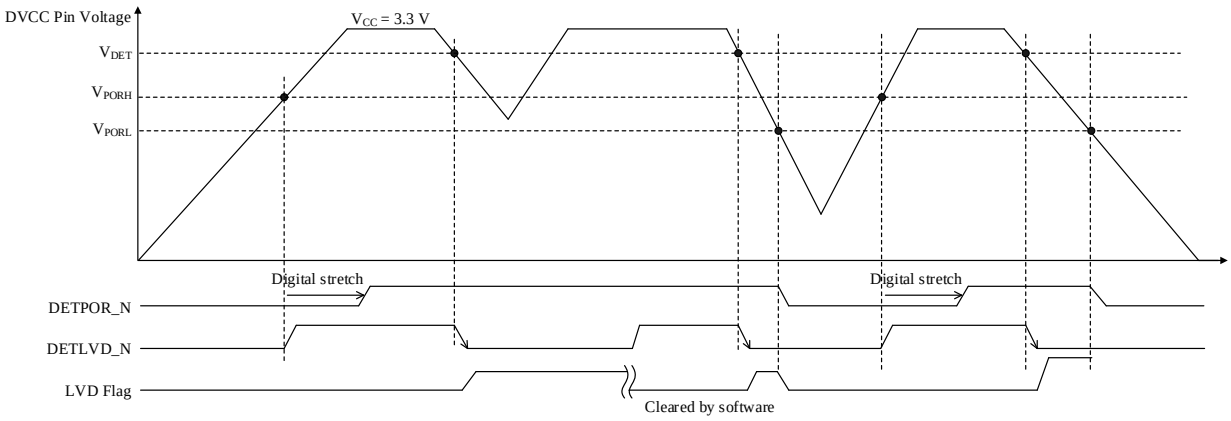


Figure 4-5. POR and LVD

4.6. Analog Infrastructure Control

Reference regulators and extended pull-up resistors of the pins are controlled by the SYSC.

- **IBIAS for OPAMP**

This controls the operation mode of the IBIAS for the OPAMP. To operate the OPAMP with the low power consumption mode, set REFCTRL.IBIAS_AMP_LPW = 1.

- **VREF Output for Low Voltage Detection (LVD)**

This controls the VREF output for the LVD. To use the LVD, set REFCTRL.VREF120AEN = 1 before the LVD is enabled. When the LVD is not used, turn off the VREF output by setting REFCTRL.VREF120AEN = 0. This results in the power consumption reduction.

- **VREF Output for Measurement by ADC**

This controls the VREF output for a measurement by the ADC. To measure the VREF voltage using the ADC, set REFCTRL.VREF120BEN = 1 before the AD conversion. When the VREF output measurement by the ADC is not used, turn off this VREF output by setting REFCTRL.VREF120BEN = 0. This results in the power consumption reduction.

- **Extended Pull-up Resistor for GPIO20**

To connect extended pull-up resistor to the GPIO20 (ANEX4), set RESCTRL.GPIO20E = 1. For reducing the pull-up resistance of the GPIO20, use the GPIO20 extended pull-up resistor in combination with the I/O buffer pull-up resistor.

- **Extended Pull-up Resistor for GPIO21**

To connect extended pull-up resistor to the GPIO21 (ANEX5), set RESCTRL.GPIO21E = 1. For reducing the pull-up resistance of the GPIO21, use the GPIO21 extended pull-up resistor in combination with the I/O buffer pull-up resistor.

4.7. Register Descriptions

Table 4-4. List of Registers

Symbol	Name	Address	Initial Value
CLKCFG0	Clock Configuration0 Register	0xFF80	0x00
CLKCFG1	Clock Configuration1 Register	0xFF81	0x01
PLLCFG	PLL Configuration Register	0xFF82	0x00
MCLKE0	Module Clock Enable0 Register	0xFF84	0x80
MCLKE1	Module Clock Enable1 Register	0xFF85	0x00
MCLKE2	Module Clock Enable2 Register	0xFF86	0x00
MCLKE3	Module Clock Enable3 Register	0xFF87	0x00
MCLKE4	Module Clock Enable4 Register	0xFF88	0x00
MCLKE5	Module Clock Enable5 Register	0xFF89	0x00
MCLKE6	Module Clock Enable6 Register	0xFF8A	0x00
LVDCTRL	LVD Control Register	0xFF90	0x00
REFCTRL	Reference Voltage Control Register	0xFF91	0x00
RESCTRL	Resistor Control Register	0xFF92	0x00
PWMENBL	PWM Clock Enable Control Register	0xFF98	0x00
PWMCSC0	PWM Clock Source Control0 Register	0xFF99	0x08
LPCTRL	Low Power Control Register	0xFFA0	0x00
CSBYCR	CMP Standby Control Register	0xFFA1	0x00
DEVER	Device Version and Revision Register	0xFFB0	0x30
REMAP	Remap Control Register	0xFFC0	0x00
TEMP*	Temperature Sensor Control Register	0xFFC1	0x00
BUSBUFCR	BUS Buffer Control Register	0xFFC2	0x80
LINECTRL	DBG Line Control Register	0xFFE0	0x00
TMR2INCR	TMR2 Input Control Register	0xFFE1	0x00

* For more details on the TEMP register, see Section 25.

4.7.1. CLKCFG0 (Clock Configuration0 Register)

Register	CLKCFG0	Clock Configuration0 Register	Address	0xFF80

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	CLKPLL	R/W	0	PLL clock 0: CLKSRC is the clock source of CLKFAST 1: PLL clock divided by 16 is the clock source of CLKFAST Writing 0 to the bit is invalid when the PLLCFG.PLLEN bit is 0, or the PLLCFG.PLLOUT bit is 0. Switching the clocks requires up to 6 cycles of the CLKSRC.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	DIV1	R/W	0	Main clock divider 00: Divide-by-8 divider is selected 01: Divide-by-4 divider is selected 10: Divide-by-2 divider is selected 11: Divide-by-1 divider is selected Switching the clocks requires up to 6 cycles of the CLKFAST.	
2		R/W	0		
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

4.7.2. CLKCFG1 (Clock Configuration1 Register)

The following settings are prohibited: disabling both the CLKIN and IRC; switching to a clock that has been stopped.

In the standby mode, the CLKIN and IRC will be stopped regardless of the settings of the CLKINE and IRCE bits. For switching between the CLKIN and IRC (i.e., from the IRC to the CLKIN, or vice versa) the CLKINE and IRCE bits must be set to 1 beforehand.

Register	CLKCFG1	Clock Configuration1 Register	Address	0xFF81

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	CLKMON	R/W	0	Clock monitor 00: CLKSRC is selected for CLKMON 01: CLKFAST is selected for CLKMON 10: CLKIRC is selected for CLKMON 11: CLKIN is selected for CLKMON	
3		R/W	0		
2	CLKSRC	R/W	0	Clock source 0: CLKIRC is selected for CLKSRC 1: CLKIN pin input is selected for CLKSRC Changing the value of the bit is valid only when the CLKINE and IRCE bits are 1. Switching the clocks requires up to 6 cycles of the clock whichever is slower, either the IRC or CLKIN.	
1	CLKINE	R/W	0	Clock input enable 0: CLKIN input is disabled 1: CLKIN input is enabled	
0	IRCE	R/W	1	IRC enable 0: IRC is disabled 1: IRC is enabled The bit is forcibly set to 1 when the OCD is connected. Setting the bit to 0 is prohibited. The bit is set to 0 when DTSTCR.IRCKILL = 1.	

4.7.3. PLLCFG (PLL Configuration Register)

Register	PLLCFG	PLL Configuration Register	Address	0xFF82

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	PLLOUT	R/W	0	PLL clock output 0: PLL clock output is disabled 1: PLL clock output is enabled Writing 0 to the bit is invalid when CLKCFG0.CLKPLL = 1.	
1	PLLEN	R/W	0	PLL enable 0: PLL is disabled 1: PLL is enabled Writing 0 to the bit is invalid when CLKCFG0.CLKPLL = 1.	
0	REFDIV	R/W	0	PLL reference clock divider 0: CLKSRC/1 1: CLKSRC/2	

4.7.4. MCLKE0 (Module Clock Enable0 Register)

Register	MCLKE0	Module Clock Enable0 Register	Address	0xFF84

Bit	Bit Name	R/W	Initial	Description	Remarks
7	ME_SCID	R/W	1	SCID clock enable 0: SCID clock is disabled 1: SCID clock is enabled Once the debugger communication starts, a clock for the SCID will be supplied regardless of the setting of the bit.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	ME_TIM23	R/W	0	TMR2/3 clock enable 0: TMR2/3 clock is disabled 1: TMR2/3 clock is enabled	
4	ME_TIM01	R/W	0	TMR0/1 clock enable 0: TMR0/1 clock is disabled 1: TMR0/1 clock is enabled	
3	ME_SPI	R/W	0	SPI clock enable 0: SPI clock is disabled 1: SPI clock is enabled	
2	ME_I2C	R/W	0	I^2C clock enable 0: I^2C clock is disabled 1: I^2C clock is enabled	
1	ME_UART	R/W	0	UART clock enable 0: UART clock is disabled 1: UART clock is enabled	
0	ME_GPIO	R/W	0	GPIO clock enable 0: GPIO clock is disabled 1: GPIO clock is enabled	

4.7.5. MCLKE1 (Module Clock Enable1 Register)

Register	MCLKE1		Module Clock Enable1 Register	Address	0xFF85
Bit	Bit Name	R/W	Initial	Description	Remarks
7	ME_EVC	R/W	0	EVC clock enable 0: EVC clock is disabled 1: EVC clock is enabled	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	ME_DSP1	R/W	0	TinyDSP1 clock enable 0: TinyDSP1 clock is disabled 1: TinyDSP1 clock is enabled	
4	ME_DSP0	R/W	0	TinyDSP1 clock enable 0: TinyDSP1 clock is disabled 1: TinyDSP1 clock is enabled	
3	ME_EPU	R/W	0	EPU clock enable 0: EPU clock is disabled 1: EPU clock is enabled	
2	ME_DSAC	R/W	0	DSAC clock enable 0: DSAC clock is disabled 1: DSAC clock is enabled	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

4.7.6. MCLKE2 (Module Clock Enable2 Register)

Register	MCLKE2		Module Clock Enable2 Register	Address	0xFF86
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	ME_ADC1	R/W	0	ADC1 clock enable 0: ADC1 clock is disabled 1: ADC1 clock is enabled	
4	ME_ADC0	R/W	0	ADC0 clock enable 0: ADC0 clock is disabled 1: ADC0 clock is enabled	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

4.7.7. MCLKE3 (Module Clock Enable3 Register)

Register	MCLKE3	Module Clock Enable3 Register	Address	0xFF87

Bit	Bit Name	R/W	Initial	Description	Remarks
7	ME_SPR	R/W	0	SPR clock enable 0: SPR clock is disabled 1: SPR clock is enabled	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	ME_AMP1	R/W	0	AMP1 clock enable 0: AMP1 clock is disabled 1: AMP1 clock is enabled	
4	ME_AMP0	R/W	0	AMP0 clock enable 0: AMP0 clock is disabled 1: AMP0 clock is enabled	
3	ME_CMP3	R/W	0	CMP3 clock enable 0: CMP3 clock is disabled 1: CMP3 clock is enabled	
2	ME_CMP2	R/W	0	CMP2 clock enable 0: CMP2 clock is disabled 1: CMP2 clock is enabled	
1	ME_CMP1	R/W	0	CMP1 clock enable 0: CMP1 clock is disabled 1: CMP1 clock is enabled	
0	ME_CMP0	R/W	0	CMP0 clock enable 0: CMP0 clock is disabled 1: CMP0 clock is enabled	

4.7.8. MCLKE4 (Module Clock Enable4 Register)

Register	MCLKE4	Module Clock Enable4 Register	Address	0xFF88

Bit	Bit Name	R/W	Initial	Description	Remarks
7	ME_CMPLUT	R/W	0	LUT clock enable 0: LUT clock is disabled 1: LUT clock is enabled	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	ME_CMP5	R/W	0	CMP5 clock enable 0: CMP5 clock is disabled 1: CMP5 clock is enabled	
0	ME_CMP4	R/W	0	CMP4 clock enable 0: CMP4 clock is disabled 1: CMP4 clock is enabled	

4.7.9. MCLKE5 (Module Clock Enable5 Register)

Register	MCLKE5	Module Clock Enable5 Register	Address	0xFF89

Bit	Bit Name	R/W	Initial	Description	Remarks
7	ME_POC3	R/W	0	POC3 clock enable 0: POC3 clock is disabled 1: POC3 clock is enabled	
6	ME_POC2	R/W	0	POC2 clock enable 0: POC2 clock is disabled 1: POC2 clock is enabled	
5	ME_POC1	R/W	0	POC1 clock enable 0: POC1 clock is disabled 1: POC1 clock is enabled	
4	ME_POC0	R/W	0	POC0 clock enable 0: POC0 clock is disabled 1: POC0 clock is enabled	
3	ME_PWM3	R/W	0	PWM3 clock enable 0: PWM3 clock is disabled 1: PWM3 clock is enabled	
2	ME_PWM2	R/W	0	PWM2 clock enable 0: PWM2 clock is disabled 1: PWM2 clock is enabled	
1	ME_PWM1	R/W	0	PWM1 clock enable 0: PWM1 clock is disabled 1: PWM1 clock is enabled	
0	ME_PWM0	R/W	0	PWM0 clock enable 0: PWM0 clock is disabled 1: PWM0 clock is enabled	

4.7.10. MCLKE6 (Module Clock Enable6 Register)

Register	MCLKE6	Module Clock Enable6 Register	Address	0xFF8A

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	ME_RAM1	R/W	0	RAM1 (addresses 0x0400 to 0x05FF) clock enable 0: RAM1 clock is disabled 1: RAM1 clock is enabled	
0	ME_RAM0	R/W	0	RAM0 (addresses 0x0000 to 0x03FF) clock enable 0: RAM0 clock is disabled 1: RAM0 clock is enabled	

4.7.11. PWMENBL (PWM Clock Enable Control Register)

Register	PWMENBL		PWM Clock Enable Control Register	Address	0xFF98
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	PWMAE	R/W	0	PWM clock source enable 0: PWM clock source is disabled 1: PWM clock source is enabled	

4.7.12. PWMCSC0 (PWM Clock Source Control0 Register)

Register	PWMCSC0		PWM Clock Source Control0 Register	Address	0xFF99
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	PWMCCA	R/W	1	Clock source 0000: PLLOUT/1 0001: PLLOUT/2 0010: PLLOUT/4 0011: PLLOUT/8 0100: PLLOUT/16 0101: PLLOUT/32 1000: CLKSRC (default) Other than above: Setting prohibited	
2		R/W	0		
1		R/W	0		
0		R/W	0		

4.7.13. LVDCTRL (Low Voltage Detector Control Register)

Register	LVDCTRL		LVD Control	Address	0xFF90
Bit	Bit Name	R/W	Initial	Description	Remarks
7	LVDE	R/W	0	Low voltage detection (LVD) enable 0: LVD is disabled 1: LVD is enabled	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	LVDIE	R/W	0	Low voltage detection (LVD) interrupt request enable 0: LVD interrupt request is disabled 1: LVD interrupt request is enabled	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	LVDIF	R/C	0	Low voltage detection (LVD) interrupt flag Read 0: LVD interrupt is not detected Read 1: LVD interrupt is detected Write 0: No change Write 1: The bit is cleared The bit is set by the LVD regardless of the setting of the LVDIE bit.	

4.7.14. REFCTRL (Reference Control Register)

Register	REFCTRL	Reference Control Register	Address	0xFF91

Bit	Bit Name	R/W	Initial	Description	Remarks
7	IBIAS_AMP_LPW	R/W	0	IBIAS operation mode for OPAMP 0: Operation in normal mode 1: Operation in low power consumption mode	
6	VREF120AEN	R/W	0	VREF output enable for LVD 0: VREF output is disabled 1: VREF output is enabled Before using the LDV, enable the VREF output for the LVD.	
5	VREF120BEN	R/W	0	VREF output enable for ADC 0: VREF output is disabled 1: VREF output is enabled When measuring the VREF voltage by the ADC, enable the VREF output for the ADC.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

4.7.15. RESCTRL (Resistor Control Register)

Register	RESCTRL	Resistor Control Register	Address	0xFF92

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	GPIO21E	R/W	0	Connecting the extended pull-up resistor to GPIO21 (ANEX5) 0: Extended pull-up resistor is not connected 1: Extended pull-up resistor is connected	
0	GPIO20E	R/W	0	Connecting the extended pull-up resistor to GPIO20 (ANEX4) 0: Extended pull-up resistor is not connected 1: Extended pull-up resistor is connected	

4.7.16. LPCTRL (Low Power Control Register)

Register	LPCTRL	Low Power Control Register	Address	0xFFA0

Bit	Bit Name	R/W	Initial	Description	Remarks
7	WUPTM	R/W	0	Setting the count number of wakeup counter 00: 512 counts (42.7 µs) 01: 768 counts (64.0 µs) 10: 1024 counts (85.3 µs) 11: 4096 counts (341.3 µs) The counter clock is the CLKSRC (12 MHz).	
6		R/W	0		
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	LPSEL	R/W	0	Selection of low power consumption mode 0: Sleep mode operation (only the CPU is stopped) 1: Standby mode operation (the chip is stopped)	
0	GOTOLPM	W	0	Transition to low power consumption mode 0: Operation in normal mode (no effect) 1: Operation in low power consumption mode	

4.7.17. CSBYCR (CMP Standby Control Register)

Register	CSBYCR	CMP Standby Control Register	Address	0xFFA1

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	LVDE	R/W	0	Return from standby mode by low voltage detection (LVD) 0: The bit is not returned 1: The bit is returned	
5	CMP5E	R/W	0	Return from standby mode by CMP5 level interrupt 0: The bit is not returned 1: The bit is returned	
4	CMP4E	R/W	0	Return from standby mode by CMP4 level interrupt 0: The bit is not returned 1: The bit is returned	
3	CMP3E	R/W	0	Return from standby mode by CMP3 level interrupt 0: The bit is not returned 1: The bit is returned	
2	CMP2E	R/W	0	Return from standby mode by CMP2 level interrupt 0: The bit is not returned 1: The bit is returned	
1	CMP1E	R/W	0	Return from standby mode by CMP1 level interrupt 0: The bit is not returned 1: The bit is returned	
0	CMP0E	R/W	0	Return from standby mode by CMP0 level interrupt 0: The bit is not returned 1: The bit is returned	

4.7.18. DEVER (Device Version and Revision Register)

Even if the protection level is 2, the DEVER register can be read by the OCD.

Register	DEVER	Device Version and Revision Register	Address	0xFFB0

Bit	Bit Name	R/W	Initial	Description	Remarks
7	VER	R	0	Device version These bits can read 3.	
6		R	0		
5		R	1		
4		R	1		
3	REV	R	x	Device revision These bits can read values according to the device revision.	
2		R	x		
1		R	x		
0		R	x		

4.7.19. REMAP (Remap Control Register)

Register	REMAP	Remap Control Register	Address	0xFFC0

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	REMAP	R/W	0	REMAP control 0: Operation in normal mode 1: Operation in REMAP mode For more details on the REMAP mode, see Section 5.2.	

4.7.20. BUSBUFCR (BUS Buffer Control Register)

Register	BUSBUFCR	BUS Buffer Control Register	Address	0xFFC2

Bit	Bit Name	R/W	Initial	Description	Remarks
7	SFRBUFE	R/W	1	SFR buffer enable 0: SFR buffer is disabled 1: SFR buffer is enabled	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	CLR_SCNT	W	0	Initialization of SFR buffer state 0: SFR buffer state remains unchanged 1: SFR buffer state is initialized	
4	CLR_SSTM	W	0	Initialization of the internal state of SFR BUS buffer 0: Internal state of SFR BUS buffer remains unchanged 1: Internal state of SFR BUS buffer is initialized	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	CLR_XSTM	W	0	Initialization of the internal state of XBUS buffer 0: Internal state of XBUS buffer remains unchanged 1: Internal state of XBUS buffer is initialized	

4.7.21. LINECTRL (DBG Line Control Register)

Register	LINECTRL	DBG Line Control Register	Address	0xFFE0

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
6	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
5	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
4	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
3	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
2	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
1	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
0	LINECTRL	R/W	0	DBG pin control 0: SCID controls DBG pin 1: UART controls DBG pin Setting the bit to 1 allows the UART to control the DBG pin.	

4.7.22. TMR2INCR (TMR2 Input Control Register)

Register	TMR2INCR	TMR2 Input Control Register	Address	0xFFE1

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
6	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
5	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
4	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
3	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
2	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
1	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
0	RXDSEL	R/W	0	TMR2 input 0: TIOA2/TIOB2 pin input is selected 1: RXD pin put is selected By setting the bit to 1, the RXD input for the UART, defined by the PFS and SIS registers, are input to TIOA and TIOB of TMR2.	

4.8. Usage Notes and Restrictions

When the LSI falls into the operation states listed below, the CPU does not return from the sleep mode.

(1) The condition in which the CPU receives a high-priority interrupt signal before or after writing an instruction enabling the sleep mode to a register.
(2) The condition in which the CPU writes an instruction enabling the sleep mode to a register while a high-priority interrupt is being processed.

Under the operation states listed below, the CPU returns from the sleep mode only when it receives a high-priority interrupt. However, the CPU does not return from the sleep mode when it receives a low-priority interrupt.

(1) The condition in which the CPU receives a high-priority interrupt signal before or after writing an instruction enabling the sleep mode to a register.
(2) The condition in which the CPU writes an instruction enabling the sleep mode to a register while a low-priority interrupt is being processed.

In a normal operation state (i.e., no interrupt is being processed), when the CPU receives no interrupt signal before or after writing a sleep-mode-enabling instruction to a register, it returns from the sleep mode by an interrupt signal regardless of the signal's priority level.

A high-priority interrupt is an interrupt whose bit corresponding to the INTLVLn register of the INTC is set to 1. A low-priority interrupt is an interrupt whose bit corresponding to the INTLVLn register of the INTC is set to 0.

The following workarounds should be implemented when the sleep mode is used.

(1) Workaround before CPU entering into sleep mode
Before the CPU enters the sleep mode, disable interrupts whose generation timing is non-CPU-controllable, such as those generated by external signals input to the comparators and GPIO. When using interrupts whose generation timing is CPU-controllable, set them not to be generated during the execution of a sleep-mode-enabling instruction.
(2) Workaround for sources used when CPU returning from sleep mode
Use interrupts whose generation timing is CPU-controllable as sources for returning from the sleep mode. When using interrupts whose timing is uncontrollable as recovery sources, enable interrupts by the EPU after the CPU enters the sleep mode. The EPU cannot access the registers of the INTC; therefore, the interrupt functions of the peripheral modules which issue interrupts should be enabled or disabled individually. It is recommended to define an interrupt used for a recovery source from the sleep mode as a high-priority interrupt.

5. 8051 CPU

The LSI has an 8051 core (UL8051) with an instruction set architecture compatible with the Intel MCS-51 (8051 family). The UL8051 is compatible with the 8052 instruction set.

The UL8051 is extended from the original 8052. Most instructions have a pipeline structure that operates in one cycle per instruction. The instruction code has 2 functions: to prefetch from the flash memory by 4 bytes, and to improve the throughput of instruction execution. In addition, the LSI has the one-wire OCD which can minimize the pin counts used for a debug.

5.1. Overview

Table 5-1 shows the UL8051 functional descriptions.

Table 5-1. UL8051 Functional Descriptions

Item	Description
Instruction Set	Compatible with MCS−51 (8052)
Execution Cycle	1 cycle per byte fetch (1T core)
Structure	Pipeline
Instruction bus	8-bit width IBUS (XPROG BUS)
Data bus	8-bit width XBUS (XDATA BUS)
SFR BUS	8-bit width with bit-write strobe
Interrupt Source	32 sources (max.)
OCD (On Chip Debugger)	Full debug function with 1-wire interface

5.2. CPU Peripheral System Configurations

Figure 5-1 shows the CPU peripheral system configuration.

- IBUS (XPROG BUS)
  For instructionfetch from the built-in flash memory.
  Access method: Instruction fetch or MOVC instruction (read only)
  Data width: 8 bits
  Address space: 64 KB

- XBUS
  For read from/write to the built-in RAM and peripheral function register.
  Access method: MOVX instruction
  Data width: 8 bits
  Address space: 64 KB

- SFR BUS
  For read from/write to the peripheral function SFR (Special Function Register).
  Access method: MOV instruction of direct addressing mode
  Data width: 8 bits
  Address space: 128 bytes (0x80 to 0xFF)

- IRAM BUS
  For read from/write to the built-in RAM
  Data width: 8 bits
  Address space: 256 bytes (0x00 to 0xFF)

In the REMAP mode, the data in the address area 0x0000 to 0x05FF of the RAM0 and the RAM1 can be read by reading the address area 0x0000 to 0x05FF in the flash memory (main block) by the instruction fetch or the MOVC instruction.

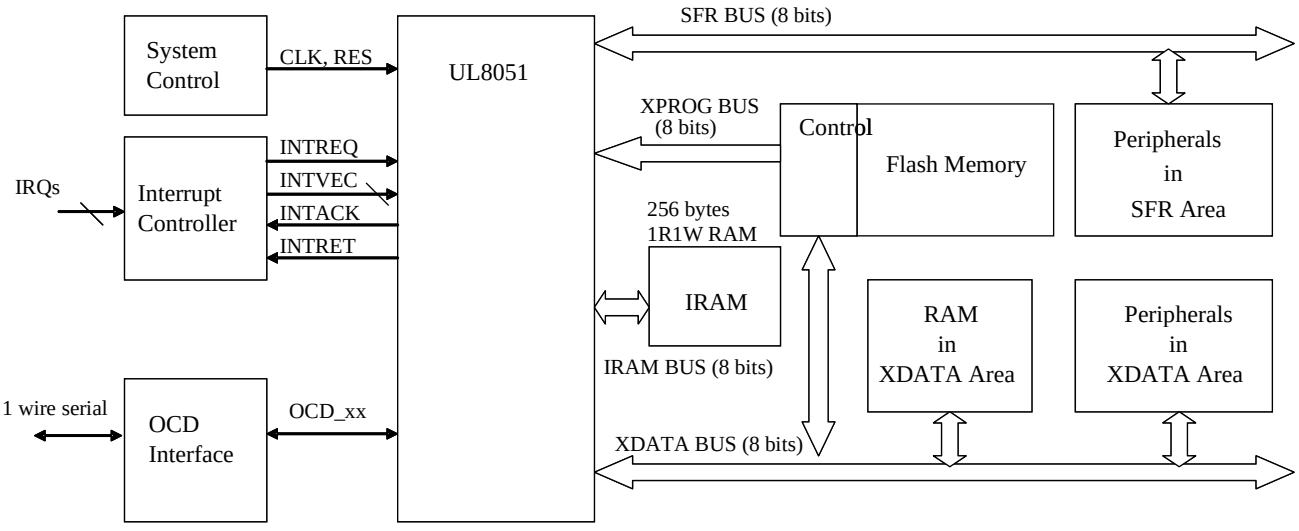


Figure 5-1. System Configuration

5.3. Memory Map

Figure 5-2 shows the system address map.

● **Internal Data Memory of 8052 Architecture**
The internal data memory is divided into 3 blocks.

- Internal Data Memory 1
  Address space: 0x00 to 0x7F (128 bytes)
  General-purpose register: 8 registers × 4 banks (32 bytes in total)
  Bit addressable area: 16 bytes
  General-purpose IRAM_1 area: 80 bytes
- Internal Data Memory 2
  Address space: 0x80 to 0xFF (128 bytes)
  General-purpose IRAM_2 area: 128 bytes
  The internal data memory 2 is accessed by the indirect addressing instruction.
- Internal Data Memory 3
  Address space: 0x80 to 0xFF (128 bytes)
  The internal data memory 3 is accessed by the direct addressing instruction in area where the CPU and the peripheral function SFR are assigned. Address 0xX0 or 0xX8 (X = 8 to F) is bit addressable access. For example, when the bit 3 of the address 0x80 is accessed, the bit address becomes 0x83.

● **Program Memory Area**
Address space: 0x0000 to 0xFFFF (64 KB)
The flash memory is assigned to the lower bits of the address (0x0000 to 0x7FFF) in program area. The data placed in the flash memory is read by the MOVC instruction.

● **Data Memory Area**
Address space: 0x0000 to 0xFFFF (64 KB)
The XDATA BUS area connected to the built-in RAM and the peripheral register is assigned. The data memory area is accessed by the MOVX instruction.

● **Peripheral Function Registers**
The peripheral function registers are assigned to the SFR and the data memory area.

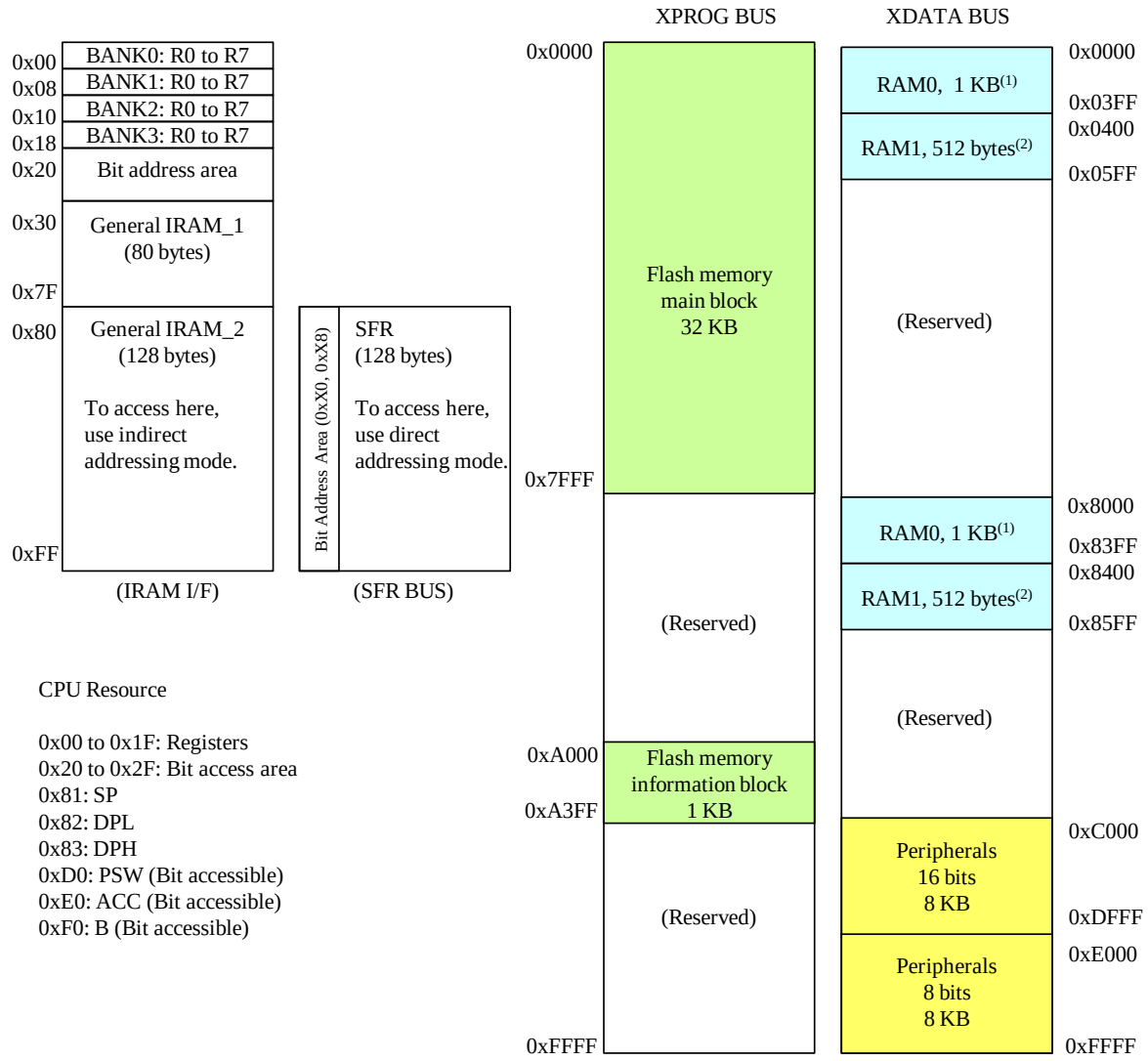


Figure 5-2. System Address Map

(1) The RAM0 area (0x8000 to 0x83FF) is the shadow memory of 0x0000 to 0x3FF.
(2) The RAM1 area (0x8400 to 0x85FF) is the shadow memory of 0x0400 to 0x5FF.

5.4. Instruction Code Map

Table 5-2. Instruction Code Map

Lower / Higher	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	Lower / Higher
0000	NOP	AJMP addr11	LJMP addr16	RR A	INC A	INC direct	INC @Ri		INC Rn								0000
0001	JBC bit, rel	ACALL addr11	LCALL addr16	RRC A	DEC A	DEC direct	DEC @Ri		DEC Rn								0001
0010	JB bit, rel	AJMP addr11	RET	RL A	ADD A, #data	ADD A, direct	ADD A, @Ri		ADD A, Rn								0010
0011	JNB bit rel	ACALL addr11	RETI	RLC A	ADDC A, #data	ADDC A, direct	ADDC A, @Ri		ADDC A, Rn								0011
0100	JC rel	AJMP addr11	ORL direct, A	ORL direct, #	ORL A, #data	ORL A, direct	ORL A, @Ri		ORL A, Rn								0100
0101	JNC rel	ACALL addr11	ANL direct, A	ANL direct, #	ANL A, #data	ANL A, direct	ANL A, @Ri		ANL A, Rn								0101
0110	JZ rel	AJMP addr11	XRL direct, A	XRL direct, #	XRL A, #data	XRL A, direct	XRL A, @Ri		XRL A, Rn								0110
0111	JNZ rel	ACALL addr11	ORL C, bit	JMP @A+DP	MOV A, #data	MOV d, #data	MOV @Ri, #data		MOV Rn, #data								0111
1000	SJMP rel	AJMP addr11	ANL C, bit	MOVC A, @A+P	DIV AB	MOV dD, dS	MOV direct, @Ri		MOV direct, Rn								1000
1001	MOV DPTR, #	ACALL addr11	MOV bit, C	MOVC A, @A+D	SUB A, #data	SUB A, direct	SUB A, @Ri		SUB A, Rn								1001
1010	ORL C, /bit	AJMP addr11	MOV C, bit	INC DPTR	MUL AB	(SBRK 0x0103)	MOV @Ri, direct		MOV Rn, direct								1010
1011	ANL C, /bit	ACALL addr11	CPL bit	CPL C	CJNE A, #, rel	CJNE A, d, rel	CJNE @Ri, #data, rel		CJNE Rn, #data, rel								1011
1100	PUSH direct	AJMP addr11	CLR bit	CLR C	SWAP A	XCH A, direct	XCH A, @Ri		XCH A, Rn								1100
1101	POP direct	ACALL addr11	SETB bit	SETB C	DA A	DJNZ d, rel	XCHD A, @Ri		DJNZ Rn rel								1101
1110	MOVX A, @DP	AJMP addr11	MOVX A, @Ri		CLR A	MOV A, direct	MOV A, @Ri		MOV A, Rn								1110
1111	MOVX @DP,A	ACALL addr11	MOVX @Ri, A		CPL A	MOV direct, A	MOV @Ri, A		MOV Rn, A								1111

5.4.1. Notes on CPU Instruction

- **Operation of Undefined Instruction Code (0xA5)**

The instruction code (0xA5) operates as “Software break: SBRK 0x0103”. The operation is similar to “LCALL 0x103”. Note that the return address (the address stored in the stack) is the address that the SBRK is placed.

- **Division-by-zero in DIV AB Instruction**

When $B = 0(A/0)$, the quotient A is 255 and the remainder B is the initial value of A. Then, the OV flag is set.

- **Stack Pointer (SP)**

The initial value of the stack pointer (SP) is 0x07. Therefore, care must be taken not to interfere with the stack area and the area used by the program. If the initial value of SP is 0x07, there is a possibility the following areas are used for the stack area: area of R0 to R7 (BANK1 to BANK3), bit addressable area, and other IRAM areas.

5.4.2. Execution Cycle Counts per Instruction

The CPU operates at 60 MHz, whereas the access frequency of the flash memory is 30 MHz (2 cycle access). The compensation system for this speed difference is as follows. The CPU instruction fetch width is 8 bits, whereas the flash memory data width stored the program is 32 bits. The flash memory is fetched the quadruple data at once. The instructions fetched from the flash memory are stored in a small capacity instruction buffer. While the instruction to be executed exists in the instruction buffer, the CPU can fetch the instruction in one cycle. When a branch is caused by a branch instruction, 2 cycles may be required because the necessary instruction should be fetched again from the flash memory.

In addition, 2 cycles are required for the access to the peripheral function registers connected to the XDATA BUS. (i.e., the execution cycle of the MOVX instruction and added one execution cycle). When the RAM is accessed by the MOVX instruction, the execution cycle is not added.

Conditional Jump: Taken/Not Taken

OPCODE			Mnemo-nic	Operand	UL8051 Cycles	Original Cycles
aaa10001	aaaaaaaa		ACALL	addr11	4	24
00100100	iiiiiiii		ADD	A, #imm	2	12
00100101	dddddddd		ADD	A, direct	3	12
0010011m			ADD	A, @Rm	2	12
00101nnn			ADD	A, Rn	1	12
00110100	iiiiiiii		ADDC	A, #imm	2	12
00110101	dddddddd		ADDC	A, direct	3	12
0011011m			ADDC	A, @Rm	2	12
00111nnn			ADDC	A, Rn	1	12
aaa00001	aaaaaaaa		AJMP	addr11	3	24
01010010	dddddddd		ANL	direct, A	3	12
01010011	dddddddd	iiiiiiii	ANL	direct, #imm	3	24
01010100	iiiiiiii		ANL	A, #imm	2	12
01010101	dddddddd		ANL	A, direct	3	12
0101011m			ANL	A, @Rm	2	12
01011nnn			ANL	A, Rn	1	12
10000010	bbbbbbbb		ANL	C, bit	3	24
10110000	bbbbbbbb		ANL	C, /bit	3	24
10110100	iiiiiiii	rrrrrrrr	CJNE	A, #imm, rel	4/4	24/24
10110101	dddddddd	rrrrrrrr	CJNE	A, direct, rel	5/5	24/24
1011011m	iiiiiiii	rrrrrrrr	CJNE	@Rm, #imm, rel	4/4	24/24
10111nnn	iiiiiiii	rrrrrrrr	CJNE	Rn, #imm, rel	4/4	24/24
11000010	bbbbbbbb		CLR	bit	3	12
11000011			CLR	C	1	12
11100100			CLR	A	1	12
10110010	bbbbbbbb		CPL	bit	3	12
10110011			CPL	C	1	12
11110100			CPL	A	1	12
11010100			DA	A	1	12
00010100			DEC	A	1	12
00010101	dddddddd		DEC	direct	3	12
0001011m			DEC	@Rm	2	12
00011nnn			DEC	Rn	1	12
10000100			DIV	AB	10	48
11010101	dddddddd	rrrrrrrr	DJNZ	direct, rel	5/5	24/24
11011nnn	rrrrrrrr		DJNZ	Rn, rel	3/3	24/24
00000100			INC	A	1	12
00000101	dddddddd		INC	direct	3	12
0000011m			INC	@Rm	2	12
00001nnn			INC	Rn	1	12
10100011			INC	DPTR	1	24
00100000	bbbbbbbb	rrrrrrrr	JB	bit, rel	5/5	24/24
00010000	bbbbbbbb	rrrrrrrr	JBC	bit, rel	5/5	24/24
01000000	rrrrrrrr		JC	rel	3/2	24/24
01110011			JMP	@A+DPTR	3	24
00110000	bbbbbbbb	rrrrrrrr	JNB	bit, rel	5/5	24/24
01010000	rrrrrrrr		JNC	rel	3/2	24/24
01110000	rrrrrrrr		JNZ	rel	3/2	24/24
01100000	rrrrrrrr		JZ	rel	3/2	24/24
00010010	aaaaaaaa	aaaaaaaa	LCALL	addr16	4	24
00000010	aaaaaaaa	aaaaaaaa	LJMP	addr16	4	24
01110100	iiiiiiii		MOV	A, #imm	2	12
01110101	dddddddd	iiiiiiii	MOV	direct, #imm	3	24
0111011m	iiiiiiii		MOV	@Rm, #imm	2	12
01111nnn	iiiiiiii		MOV	Rn, #imm	2	12
10000101	ddd(src)	ddd(dst)	MOV	dir(dst), dir(src)	3	24

Conditional Jump: Taken/Not Taken

OPCODE			Mnemo-nic	Operand	UL8051 Cycles	Original Cycles
1000011m	dddddddd		MOV	direct, @Rm	2	24
10001nnn	dddddddd		MOV	direct, Rn	2	24
10010000	iiiiiiii	iiiiiiii	MOV	DPTR, #imm16	3	24
10010010	bbbbbbbb		MOV	bit, C	3	24
10100010	bbbbbbbb		MOV	C, bit	3	12
1010011m	dddddddd		MOV	@Rm, direct	3	24
10101nnn	dddddddd		MOV	Rn, direct	3	24
11100101	dddddddd		MOV	A, direct	3	12
1110011m			MOV	A, @Rm	2	12
11101nnn			MOV	A, Rn	1	12
11110101	dddddddd		MOV	direct, A	2	12
1111011m			MOV	@Rm, A	1	12
11111nnn			MOV	Rn, A	1	12
10000011			MOVC	A, @A+PC	4	24
10010011			MOVC	A, @A+DPTR	4	24
11100000			MOVX	A, @DPTR	3	24
1110001m			MOVX	A, @Rm	3	24
11110000			MOVX	@DPTR, A	1	24
1111001m			MOVX	@Rm, A	1	24
10100100			MUL	AB	1	48
00000000			NOP		1	12
01000010	dddddddd		ORL	direct, A	3	12
01000011	dddddddd	iiiiiiii	ORL	direct, #imm	3	24
01000100	iiiiiiii		ORL	A, #imm	2	12
01000101	dddddddd		ORL	A, direct	3	12
0100011m			ORL	A, @Rm	2	12
01001nnn			ORL	A, Rn	1	12
01110010	bbbbbbbb		ORL	C, bit	3	24
10100000	bbbbbbbb		ORL	C, /bit	3	24
11010000	dddddddd		POP	direct	2	24
11000000	dddddddd		PUSH	direct	3	24
00100010			RET		5	24
00110010			RETI		5	24
00100011			RL	A	1	12
00110011			RLC	A	1	12
00000011			RR	A	1	12
00010011			RRC	A	1	12
10100101			SBRK	0x0103	4	Undefined
11010010	bbbbbbbb		SETB	bit	3	12
11010011			SETB	C	1	12
10000000	rrrrrrrr		SJMP	rel	3	24
10010100	iiiiiiii		SUBB	A, #imm	2	12
10010101	dddddddd		SUBB	A, direct	3	12
1001011m			SUBB	A, @Rm	2	12
10011nnn			SUBB	A, Rn	1	12
11000100			SWAP	A	1	12
11000101	dddddddd		XCH	A, direct	3	12
1100011m			XCH	A, @Rm	2	12
11001nnn			XCH	A, Rn	1	12
1101011m			XCHD	A, @Rm	2	12
01100010	dddddddd		XRL	direct, A	3	12
01100011	dddddddd	iiiiiiii	XRL	direct, #imm	3	24
01100100	iiiiiiii		XRL	A, #imm	2	12
01100101	dddddddd		XRL	A, direct	3	12
0110011m			XRL	A, @Rm	2	12
01101nnn			XRL	A, Rn	1	12

5.5. Bus Configurations

Table 5-3 shows the bus functional description. Figure 5-3 shows the system bus configuration.

Table 5-3. Bus Functional Descriptions

Item	Description
System bus	- XDATA BUS 8-bit width, 64-KB space Access cycle: 2 cycles (CLKFAST) - SFR BUS 16-bit width, 256-byte space 8-bit/16-bit access mode Access cycle: 1 cycle (CLKFAST)
Bus Master	- CPU - EPU (Event Processing Unit) - DSAC (Direct SFR Access Controller)
CPU BUS	- IBUS (XPROG BUS) 8-bit width, 64-KB space - XBUS 8-bit width, 64-KB space - SBUS 8-bit width, 128-byte space - IRAM BUS 8-bit width, 256-byte space
EPU BUS	- MBUS 16-bit width, 512-byte space - XBUS 8-bit width, 64-KB space - SBUS 16-bit width, 256-byte space
DSAC BUS	- SFR BUS 16-bit width, 256-byte space
Arbitration Circuit	- Flash memory arbiter - Peripheral function arbiter - XDATA BUS arbiter - SFR BUS arbiter - RAM arbiter
16-bit Access Buffer	- XDATA BUS buffer - SFR BUS buffer

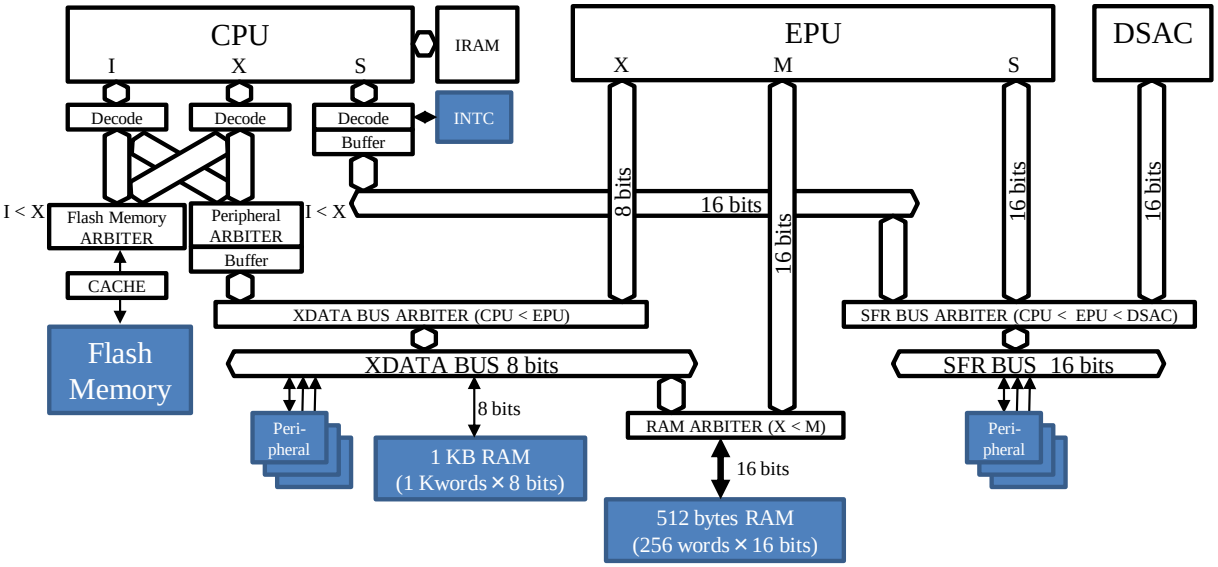


Figure 5-3. System Bus Configuration

5.6. Bus Operational Descriptions

5.6.1. System Bus

The XDATA and SFR BUSes are system buses. They are connected to the peripheral functions and the memories (RAM0 and RAM1). The XDATA and SFR BUSes operate independently.

● **XDATA BUS**
The XDATA BUS is connected to the peripheral function registers (low speed) and the memories (RAM0 and RAM1).

- Read/write access cycle: 2 cycles (CLKFAST)
- Data width: 8 bits
- Address space: 64 KB

● **SFR BUS**
The SFR BUS is connected to the peripheral function registers (high speed).

- Read/write access cycle: 1 cycle (CLKFAST)
- Data width: 16 bits
- Address space: 256 bytes
- Access mode: 8-bit/16-bit access

The 16-bit access mode operates to access the 16-bit register connected to the SFR BUS in one cycle using the 16-bit data bus. The 8-bit access mode operates to access the 8-bit register connected to the SFR BUS using the lower 8 bits in the 16-bit data bus. In addition, there are the peripheral functions that can access 16-bit registers by accessing twice in the 8-bit access mode.

5.6.2. Bus Master

● **CPU**

The CPU has the following bus interfaces: IBUS, XBUS, SBUS.

The flash memory and the XDATA BUS are accessed from the IBUS and the XBUS. The XDATA BUS is accessed from the CPU via a buffer. This buffer guarantees the atomicity of the 16-bit register access (sequential addresses: 2n and 2n + 1).

To access the 16-bit SFR from the CPU, access the same address twice. The SFR is accessed from the CPU via the buffer. This buffer combines 2 accesses from the CPU at once, and executes the 16-bit access at a time. Then, this buffer guarantees the atomicity of the 16-bit SFR access.

- IBUS: Used for the data reading by the instruction fetch or the MOVC instruction
  Data bus width: 8 bits
  Address space: 64 KB
- XBUS: Used for data access by the MOVX instruction
  Data bus width: 8 bits
  Address space: 64 KB
- SBUS: Used for the SFR access by the MOV instruction
  Data bus width: 8 bits
  Address space: 128 bytes (0x80 to 0xFF), 0x00 to 0x7F is an access prohibited area.

● **DSAC (Direct SFR Access Controller)**

The DSAC has the SFR BUS interface to access the SFR BUS.
Data bus width: 16 bits
Address space: 256 bytes

● **EPU (Event Processing Unit)**

The EPU has the following bus interfaces: MBUS, XBUS, SBUS.

- MBUS: For the instruction fetch and MOVM instruction accesses
  Data bus width: 16 bits
  Address space: 512 bytes
  The MBUS is connected to RAM1, and is sharing with the XDATA BUS.
- XBUS: For the MOVX instruction data accesses, and accessible to the XDATA BUS
  Data bus width: 8 bits
  Address space: 64 KB
- SFR BUS: For the SFR access of the MOVS instruction
  Data bus width: 16 bits
  Address space: 256 bytes

5.6.3. Arbitration Circuit

● **Flash Memory Arbiter**

This is the arbiter for the accesses to the flash memory of the IBUS and the XBUS in the CPU. When the IBUS access conflicts with the XBUS access, the XBUS access is prioritized, then the IBUS access is waited. When there is no subsequent XBUS access after the XBUS access completes, the waiting IBUS access is executed. When the other bus accesses while one bus is accessing the flash memory, the subsequent access waits until the access to the flash memory completes. After the first access completes, the subsequent access is arbitrated according to the priority order.

● **Peripheral System Arbiter**

This is the arbiter for the accesses to the XDATA BUS of the IBUS and the XBUS in the CPU. When the IBUS access conflicts with the XBUS access, the XBUS access is prioritized, then the IBUS access is waited. When there is no subsequent XBUS access after the XBUS access completes, the waiting IBUS access is executed. When the other bus accesses while one bus is accessing the XDATA BUS, the subsequent access waits until the access to the XDATA BUS completes. After the first access completes, the subsequent access is arbitrated according to the priority order.

● **XDATA BUS Arbiter**

This is the arbiter for the accesses to the XDATA BUS of the CPU and the EPU. When the CPU access conflicts with the EPU access, the EPU access is prioritized, then the CPU access is waited. When there is no subsequent EPU access after the EPU access completes, the waiting CPU access is executed. When the other bus accesses while one bus is accessing the XDATA BUS, the subsequent access waits until the access to the XDATA BUS completes. After the first access completes, the subsequent access is arbitrated according to the priority order.

● **SFR BUS Arbiter**

This is the arbiter for the SFR BUS accesses of the CPU, the DSAC, and the EPU. When the two or more accesses conflict, the highest priority access is selected, and then the unselected accesses are waited. The priority order is DSAC > EPU > CPU. After the selected access completes, the subsequent access is selected again according to the priority order. When the other bus accesses while one bus is accessing the SFR BUS, the subsequent access waits until the access to the SFR BUS completes. After the first access completes, the subsequent access is arbitrated according to the priority order.

● **RAM Arbiter**

This is the arbiter for the access to the RAM1 of the XDATA BUS and the MBUS (EPU). When the XDATA BUS access conflicts with the MBUS access, the MBUS access is prioritized, then the XDATA BUS access is waited. When there is no subsequent MBUS access after the MBUS access completes, the waiting XDATA BUS access is executed.

5.6.4. 16-bit Access Buffer

The 16-bit register exists on the XDATA and SFR BUSes. 16-bit data must be read from or written to the 16-bit register. The read or write of data is completed by accessing with the 16-bit data. Since the CPU is accessed by 8-bit units, 2 accesses are required for the 16-bit register. In the CPU, note that the interrupt and the OCD access may be generated during the CPU access of 16-bit data. In addition, the access to register of the same module during an interrupted access processing may cause the following operation: The write access before being interrupted is invalidated. The unintended operation such as the first access during the interrupt processing is recognized as the second access.

For solution of this problem, the 16-bit access buffer is integrated. In the first access to the 16-bit register, buffering processing is performed. Then, the atomicity of 2 accesses is guaranteed. The buffer has 4 statuses (uninterrupted status, low-level interrupt, high-level interrupt, and OCD access). To solve the failure of 16-bit access by interrupt processing, use the proper buffer depending on the each status.

In the write access, the first write access is buffered, and is not transmitted to the XDATA BUS or the SFR BUS. At the second write access, the first and second writes are atomically generated.

In the read access, the first and second reads are generated on the XDATA BUS or the SFR BUS at the first read access to obtain the 16-bit data at once. Note that, only the first read value is obtained at the first access, and the second read value is buffered. The second read value is obtained from the buffer at the second read.

(1) XDATA BUS Buffer

The XDATA BUS buffer is to guarantee the atomicity when the CPU accesses the 16-bit register on the XDATA BUS. The peripheral function register is assigned to the address 0xE000 to 0xFFFF. To enable the XDATA BUS buffer, access the 16-bit register space that is the mirror area of the peripheral function register space (0xC000 to 0xDFFF).. Also, the peripheral function register space can be accessed from the 16-bit register space (for example, the 16-bit register address of 0xFF80 is 0xDF80). To access the 16-bit register, access 2 addresses, 2n and 2n + 1, in the sequential order. The accesses to address 2n and 2n + 1 are regarded as the first access (lower byte) and the second access (higher byte), respectively. To reset the control state of the 16-bit register access buffer, set BUSBUFCR.CLR_XSTM = 1.

(2) SFR BUS Buffer

The SFR BUS buffer is to guarantee the atomicity when the CPU accesses the 16-bit register on the SFR BUS. To enable the buffer processing at the access to the 16-bit register on the SFR BUS, set BUSBUFCR.SFRBUFE = 1. To access the 16-bit register on the SFR BUS, 2 accesses are required. The SFR BUS access after processing the buffer operates in the 16-bit access mode. This guarantees the atomicity. To stop the buffer processing, set BUSBUFCR.SFRBUFE = 0. To clear the counter that counts the register access is the first or second, set BUSBUFCR.CLR_SCNT = 1. To reset the control state of the 16-bit register access buffer, set BUSBUFCR.CLR_SSTM =1.

5.7. Usage Notes and Restrictions

5.7.1. Restrictions on XDATA BUS Buffer

While a program in a RAM on a bus is being executed, the functions of the XDATA BUS buffer cannot be used.

5.7.2. Conflict between 16-bit Register Write and Interrupt

Writing to the 16-bit registers must not conflict with any interrupt acceptance; therefore, before writing to the higher bytes of the 16-bit registers (SFR BUS, XDATA BUS), set the INTMST.INTME bit to 0 to disable any interrupt acceptance. Then, insert two NOP instructions and write to the higher bytes. After writing to the higher bytes, set the INTMST.INTME bit to 1 to re-enable interrupt acceptance.

5.7.3. Access to RAM1 Area

Do not access the 512-byte RAM1 area (0x0400 to 0x05FF) from the CPU during the EPU operation.

5.7.4. Access to XDATA Space

Do not access the XDATA space addresses (0xC000 to 0xDFF) from the CPU during the EPU operation.

5.7.5. MOVX Instructions to Peripheral Registers of XDATA Space

During the EPU operation, the MOVX instructions by the CPU must not be executed consecutively, as in the examples below.

Example 1: Write the same data consecutively to the same address of the XDATA peripheral register from the CPU.

```
movx @dptr, a  (Write)
movx @dptr, a  (Write)
```

Example 2: Write to/read from the same address of the XDATA peripheral register consecutively by the CPU.

```
movx @dptr, a  (Write)
movx a, @dptr  (Read)
```

Inserting two or more execution cycles of a different instruction other than MOVX (e.g., NOP) between the two MOVX instructions can avoid the malfunction mentioned above. When the program is written in C, implement the workaround as follows.

During the EPU operation, when the CPU attempts two consecutive accesses to a peripheral register of the XDATA space with the MOVX instruction, insert two NOP instructions, "`__asm__("nop");`", between the consecutive accesses by the CPU as in the examples 1 and 2.

5.7.6. EPU Operation While CPU Is Accessing UART Register

5.7.6.1. EPU Access Failure by XDATA BUS Conflict

If the EPU attempts to access the XDATA space while the CPU is accessing the UART register, the buses will have conflicts thus the EPU access fails. To avoid such conflicts, implement the following workarounds.

- **Workaround**

(1) Access the UART by the EPU (alternative access for the CPU; see Table 5-4)

This workaround is to permit the EPU to access the UART registers instead of the CPU. Handshaking with the CPU can control one thread of the EPU, thus permitting an alternative access to the UART registers by the EPU. Flags for handshaking, data addresses to be accessed are then read/written by the CPU and EPU from/to the following: SPR on the SFR BUS; the RAM0 (0x0000 to 0x03FF).

(2) Access the UART by the CPU (see Table 5-4)
This workaround is to perform two consecutive accesses to the XDATA space by the EPU, thus avoiding access fails due to bus conflicts. The first access succeeds or fails, but the second access succeeds. In this description, "fail" means that no access by the EPU is issued to the XDATA space. If an access fails, no writing will be executed when a write access is attempted, whereas undefined values will be read when a read access is attempted.
When writing the instructions which activate the hardware (e.g., ADC trigger, EPU activation, UART transmission and reception, DSAC trigger, PWM re-trigger, etc.) to the peripheral function registers, write an ineffective value at the first time or implement the workaround (1) above.

(2)-1 In the case where thread switching occurs while an EPU thread is being processed:

- Perform byte access to the XDATA space by the EPU
  Read/write bytes by using the word access instruction W_SA mode (i.e., two consecutive accesses to the same address). For the usage notes on the word access instruction, see Section 5.7.6.2 below.
- Perform word access to the XDATA space by the EPU
  Accessing the 16-bit-width peripheral function registers (i.e. consecutive accessing to the lower and higher bytes of the registers) is prohibited. When accessing the 16-bit-width peripheral function registers, implement the workaround (1) above.

(2)-2 In the case where no thread switching occurs while an EPU thread is being processed:
When either of the following two conditions is met, perform two consecutive accesses to the XDATA space regardless of the byte or word access instruction: the condition in which only one thread is processed at once; or the condition in which no round-robin thread switching occurs while multiple threads are processed.

Table 5-4. List of Workarounds

Workaround		Instructions to Be Used	
		Byte Access Instruction by EPU	Word Access Instruction by EPU
(1)	Access by EPU to UART (Alternative Access)	Available	Available
(2)-1	Access by CPU to UART (Thread Switching Occurs)	Substitute the word access instruction (see Section 5.7.6.2)	Prohibited
(2)-2	Access by CPU to UART (No Thread Switching Occurs)	Execute two consecutive byte instructions	Execute two consecutive word instructions

5.7.6.2. Usage Notes on Two Consecutive Accesses with Word Access Instruction

When you perform two consecutive accesses to the same address by using the word access instruction, each access will end up with two different results according to which of the accesses, read (LOADX) or write (STOREX), has been attempted.

Read Access: `LOADX.W_SA Rn, @AddrX`

- The first and second accesses suceed
- The first access fails (a LOAD value is undefined); the second access succeeds

Write Access: `STOREX.W_SA @AddrX, Rn`

- The first and second accesses suceed
- The first access fails (access is lost); the second access succeeds

If both the first and second accesses succeeded, the first write access may cause malfunctions. Therefore, the following workarounds must be implemented, for proper writing to the registers by the EPU.

- Writing to the setting register of the peripheral fuction:
  Write the same values at the first and second accesses.
- Clearing the flags of the peripheral fuction registers:
  Write 0 at the first time, then clear at the second time.
- Writing the instructions which activate the hardware (e.g., ADC trigger, EPU activation, UART transmission and reception, DSAC trigger, PWM re-trigger, etc.) to the peripheral function registers:
  Write an ineffective value, or implement the workaround (1) in Section 5.7.6.1.

5.7.7. CPU and EPU Simultaneous Access to XDATA Space

● **Details**

Malfunctions may occur when the CPU and EPU attempt simultaneous access to the XDATA space, and then the EPU accesses the XDATA space after a single-cycle instruction. This is applicable to all of the the XDATA BUS access instructions by the EPU: STOREX (B_LO, B_HI, W, W_SA) and LOADX (B_SE, B_ZE, W, W_SA). In the descriptions hereafter, the term "STOREX/LOADX instruction" represents these XDATA BUS access instructions by the EPU. The term "MOVX instruction" represents all of the XDATA BUS access instructions by the CPU (MOVX @DPTR, A; MOVX @Rm, A; MOVX A, @DPTR; MOVX A, @Rm), which are also subject to the malfunction mentioned above.

The failure example below explains that the EPU may hang up, or the CPU or EPU may lose its access after a single-cycle instruction according to the EPU and CPU access destinations in the XDATA space (Table 5-5). Losing an access means that no writing will be executed when a write access is attempted, and undefined values will be read when a read access is attempted.

[Failure Example]

When the STOREX instruction (E-1) and the MOVX instruction (C-1) attempt simultaneous access to any of the peripheral registers of the XDATA space, the XDATA BUS access of the MOVX instruction (C-3) will be lost.

The EPU attempts the second access to a peripheral register of the XDATA space after a single-cycle instruction.

```
(E-1) STOREX.B_LO @addr, Rm
(E-2) MOV Rn, Rm
(E-3) STOREX.B_LO @addr, Rm
```

At the same time, the CPU also attempts the second access to a peripheral register of the XDATA space after a single-cycle instruction.

```
(C-1) movx @dptr, a
(C-2) nop
(C-3) movx @dptr, a
```

Table 5-5. Access Destinations and Phenomena to Be Occurred

EPU Instruction (E-1) Access Destination	EPU Instruction (E-3) Access Destination	CPU Instruction (C-1) (Access Destinations of 1st Instruction)		
		Peripheral Register	UART	RAM0/1
Peripheral register (incl. UART)	Peripheral register (incl. UART)	(C-3) Lost(1)	(E-3) Lost(3)	Access succeeded
	RAM0/1	EPU hanged up(2)	(E-3) Lost(3)	Access succeeded
RAM0/1	XDATA space (Peripheral register, UART, RAM0/1)	Access succeeded	(E-3) Lost(3)	Access succeeded

(1) Refers that the CPU's second XDATA BUS access (C-3) after a single-cycle instruction is lost.
(2) Refers that the EPU remains hanged up until the CPU's next XDATA BUS access occurs from the EPU's second XDATA BUS access ((E-3); to the RAM0/1) after a single-cycle instruction.
(3) Refers that the EPU's second XDATA BUS access (E-3) after a single-cycle instruction is lost (i.e., the

phenomenon described in Section 2.6).
When the CPU's first XDATA BUS access destination is the UART register, the workaround for the EPU's two consecutive accesses (see Section 5.7.6) does not prevent losing the CPU's second XDATA BUS access (C-3) after a single-cycle instruction, as shown in the following failure example. Therefore, implement the workarounds listed in this section in addition to the workarounds for the UART listed in Section 5.7.6.

[Failure Example]

When the STOREX instruction (E-1) and the MOVX instruction (C-1) attempt simultaneous access to the XDATA space, the XDATA BUS accesses of the STOREX instruction (E-3) and the MOVX instruction (C-3) will be lost.

The EPU attempts two consecutive accesses to the XDATA space after a single-cycle instruction.

```
(E-1) STOREX.B_LO @addr, Rm
(E-2) MOV Rn, Rm
(E-3) STOREX.B_LO @addr, Rm
(E-4) STOREX.B_LO @addr, Rm
```

At the same time, the CPU attempts the second access to the XDATA space after a single-cycle instruction.

```
(C-1) movx @dptr, a  (Access destination is the UART register.)
(C-2) nop
(C-3) movx @dptr, a
```

To avoid the failure examples above, implement the following workarounds.

(1) Workaround by the EPU

As indicated in the instructions (1) to (3) in the failure examples 1 to 4, the following execution order may cause the malfunction mentioned above: (1) the STOREX/LOADX instruction, (2) a single-cycle instruction other than the STOREX/LOADX instruction, then (3) the STOREX/LOADX instruction. To avoid the malfunction, make the EPU execute the STOREX/LOADX instructions consecutively, or insert one or more instructions between the MOVX instructions so that there are at least two consecutive bus-idle-state cycles between the XDATA BUS accesses (i.e., insert two single-cycle instructions or one double-cycle instruction). For the execution cycle counts defined for each instruction, see the section describing the EPU operations.

[Failure Examples]

Example 1:

```
(1) STOREX.B_LO @addr, Rm
(2) MOV Rn, Rm
(3) STOREX.B_LO @addr, Rm
```

Example 2:

```
(1) STOREX.B_LO @addr, Rm
(2) ADD Rn, Rm
(3) LOADX.B_SE Rn, @addr
```

Example 3:

```
(1) LOADX.B_SE Rn, @addr
(2) OR Rn, Rm
(3) STOREX.W @addr, Rm
```

Example 4:

```
(1) LOADX.W_SA Rn, @addr
(2) INC Rn, Rm
(3) LOADX.B_SE Rn, @addr
```

(1)-1 Workround for the fixed priority setting

Do not put the following in the same thread: the STOREX/LOADX instruction, a single-cycle instruction other than the STOREX/LOADX instruction, and a sequence of the STOREX/LOADX instructions.

As the workaround example 1 shows, put the STOREX/LOADX instructions consecutively, or insert two single-cycle instructions or one double-cycle instruction. The workaround example 2 is for inserting two single-cycle instructions: insert an instruction that has no operational influence, or change the instruction order if no operational problem is found.

[Workaround Examples]

Example 1:

```
(1) STOREX.B_LO @addr, R0
(2) LOADX.B_SE R1, @addr
```

Example 2:

```
(1) STOREX.B_LO @addr, R1
(2) MOV R0, R0
(3) MOV R1, R1
(4) STOREX.B_LO @addr, R0
```

When thread switching occurs, be sure to pay attention to the following condition: a certain thread contains any of the instruction sequences listed in the following failure examples and a thread with lower priority than the certain thread contains the STOREX/LOADX instruction. If the upper-priority thread is executed immediately after the STOREX/LOADX instruction in the lower-priority thread is executed, an instruction sequence leading to malfunctions may occur depending on a release timing of the WAIT instruction. To avoid such failure, check operation timings and insert either of the following between the EVTWAIT/TIMWAIT instruction and the STOREX/LOADX instruction as needed: two or more single-cycle instructions; one or more double-cycle instructions.

[Failure Examples]

Example 1:

```
(1) EVTWAIT #evt
(2) STOREX.B_LO @addr, Rm
```

Example 2:

```
(1) EVTWAIT #evt
(2) MOV Rn, Rm
(3) STOREX.B_LO @addr, Rm
```

Example 3:

```
(1) TIMWAIT #time   (Failure occurs even when #time = 0.)
(2) STOREX.B_LO @addr, Rm
```

Example 4:

```
(1) TIMWAIT #time   (Failure occurs even when #time = 0.)
(2) MOV Rn, Rm
(3) STOREX.B_LO @addr, Rm
```

[Workaround Examples]

Example 1:

```
(1) EVTWAIT #evt
(2) MOV R0, R0
(3) MOV R0, R0
(4) STOREX.B_LO @addr, R1
```

Example 2:

```
(1) TIMWAIT #time
(2) MOV R1, R1
(3) MOV R1, R1
(4) STOREX.B_LO @addr, R0
```

(1)-2 Workaround for the round-robin setting

Do not use the round-robin method when the STOREX/LOADX instruction is used in multiple threads.

(2) Workaround by the CPU

Insert one or more instructions between the MOVX instructions so that there are at least two consecutive bus-idle-state cycles between the XDATA BUS accesses when the following execution order exists in the CPU: (1) the MOVX instruction, (2) a single-cycle instruction, then (3) the MOVX instruction. Note that when the workaround by the EPU is already implemented, no workaround by the CPU is required. For the execution cycle counts defined for each instruction, see Section 5.4.2 that provides the 8051 CPU instruction code map. However, note that the workaround by the EPU must be implemented when the EPU may hang up (as described in the footnote (2), Table 5-5) due to the following execution order existing in the EPU: (1) the STOREX/LOADX instruction, (2) a single-cycle instruction other than the STOREX/LOADX instruction, then (3) the STOREX/LOADX instruction (with the access destination to the RAM0/1).

[Workaround Example]

```
movx @dptr,a
nop            (Insert two single-cycle instructions b/w the MOVX instructions, e.g., two NOP instructions.)
nop
movx @dptr,a
```

6. Register Mapping

6.1. Peripheral Address on XDATA BUS

For more details of register, see the corresponding section.

Table 6-1. Peripheral Address on XDATA BUS

Module	Address (8 bits)		Address (16 bits)	
	Start	End	Start	End
EPU	E000	E07F	C000	C07F
—	E080	E0FF	C080	C0FF
—	E100	E17F	C100	C17F
—	E180	E1FF	C180	C1FF
SCID	E200	E27F	C200	C27F
—	E280	E2FF	C280	C2FF
EVC	E300	E37F	C300	C37F
—	E380	E3FF	C380	C3FF
—	E400	E47F	C400	C47F
—	E480	E4FF	C480	C4FF
—	E500	E57F	C500	C57F
—	E580	E5FF	C580	C5FF
—	E600	E67F	C600	C67F
—	E680	E6FF	C680	C6FF
—	E700	E77F	C700	C77F
—	E780	E7FF	C780	C7FF
—	E800	E87F	C800	C87F
—	E880	E8FF	C880	C8FF
—	E900	E97F	C900	C97F
—	E980	F9FF	C980	D9FF
—	EA00	EA7F	CA00	CA7F
—	EA80	EAFF	CA80	CAFF
—	EB00	EB7F	CB00	CB7F
—	EB80	EBFF	CB80	CBFF
—	EC00	EC7F	CC00	CC7F
CMPLUT	EC80	ECFF	CC80	CCFF
—	ED00	ED7F	CD00	CD7F
CMP4	ED80	EDFF	CD80	CDFF
CMP5	EE00	EE7F	CE00	CE7F
—	EE80	EEFF	CE80	CEFF
—	EF00	EF7F	CF00	CF7F
—	EF80	EFFF	CF80	CFFF
ADC0	F000	F07F	D000	D07F
ADC1	F080	F0FF	D080	D0FF
—	F100	F17F	D100	D17F
—	F180	F1FF	D180	D1FF
—	F200	F27F	D200	D27F
—	F280	F2FF	D280	D2FF
—	F300	F37F	D300	D37F
CMP0	F380	F3FF	D380	D3FF
CMP1	F400	F47F	D400	D47F
CMP2	F480	F4FF	D480	D4FF
CMP3	F500	F57F	D500	D57F
—	F580	F5FF	D580	D5FF

Module	Address (8 bits)		Address (16 bits)	
	Start	End	Start	End
OPAMP0	F600	F67F	D600	D67F
OPAMP1	F680	F6FF	D680	D6FF
—	F700	F77F	D700	D77F
TinyDSP0	F780	F7FF	D780	D7FF
TinyDSP1	F800	F87F	D800	D87F
DSAC	F880	F8FF	D880	D8FF
PWM0/ 1	F900	F97F	D900	D97F
PWM2/ 3	F980	F9FF	D980	D9FF
TIMER	FA00	FA7F	DA00	DA7F
—	FA80	FAFF	DA80	DAFF
—	FB00	FB7F	DB00	DB7F
SPI	FB80	FBFF	DB80	DBFF
I2C	FC00	FC7F	DC00	DC7F
UART	FC80	FCFF	DC80	DCFF
TRIM	FD00	FD7F	DD00	DD7F
POC	FD80	FDFF	DD80	DDFF
GPIO	FE00	FE7F	DE00	DE7F
WDT	FE80	FEFF	DE80	DEFF
FLC	FF00	FF7F	DF00	DF7F
SYSC	FF80	FFFF	DF80	DFFF

6.2. Peripheral Address on SFR BUS

The CPU system and frequently accessed peripheral registers are assigned to the SFR area. The CPU can read from/write to the SFR in 1 cycle. The PSW, ACC, and B registers are bit-addressable registers. Do not access the unassigned addresses, which are the gray area as shown in Table 6-2 and Table 6-3.

Table 6-2. Peripheral Register on SFR BUS

80		SP	DPL	DPH					87
88		ADIF0	ADIF1			MIXDA4 L/H	MIXDA5 L/H	SPR4	8F
90	PDR0	SPR0	MIXDA2 L/H	MIXDA3 L/H	SPR5	MIXDA1 L/H	MIXDA0 L/H		97
98	PDR1	AD00L/H	AD01L/H		INTMST	CMI1	SPR6	SPR7	9F
A0		AD10L/H	AD11L/H		INTENA0	INTENA1	INTENA2	INTENA3	A7
A8		AD20L/H	AD21L/H		INTLVL0	INTLVL1	INTLVL2	INTLVL3	AF
B0	PDR2	AD30L/H	AD31L/H		INTCFG0	INTCFG1	INTCFG2	INTCFG3	B7
B8		AD40L/H	AD41L/H		INTFLG0	INTFLG1	INTFLG2	INTFLG3	BF
C0		AD50L/H	AD51L/H		DSP0 _R0_L/H	DSP0 _R1_L/H	DSP0 _R2_L/H	DSP0 _R3_L/H	C7
C8	PIF0	AD60L/H	AD61L/H		DSP0 _R4_L/H	DSP0 _R5_L/H	DSP0 _R6_L/H	DSP0 _R7_L/H	CF
D0	PSW	AD70L/H	AD71L/H		DSP1 _R0_L/H	DSP1 _R1_L/H	DSP1 _R2_L/H	DSP1 _R3_L/H	D7
D8	PIF1	AD80L/H	AD81L/H		DSP1 _R4_L/H	DSP1 _R5_L/H	DSP1 _R6_L/H	DSP1 _R7_L/H	DF
E0	ACC	AD90L/H	AD91L/H		BUF_A0 _L/H	BUF_B0 _L/H	BUF_C0 _L/H	BUF_D0 _L/H	E7
E8	PIF2	ADA0L/H	ADA1L/H		BUF_A1 _L/H	BUF_B1 _L/H	BUF_C1 _L/H	BUF_D1 _L/H	EF
F0	B	ADB0L/H	ADB1L/H	CMI0	BUF_A2 _L/H	BUF_B2 _L/H	BUF_C2 _L/H	BUF_D2 _L/H	F7
F8		SPR1	SPR2	SPR3	BUF_A3 _L/H	BUF_B3 _L/H	BUF_C3 _L/H	BUF_D3 _L/H	FF

Table 6-3. Peripheral Register on SFR BUS Accessed from DSAC and EPU

00	ADO00 L/H	ADO01 L/H			TCMPA0 L/H	TCMPB0 L/H	TBUFA0 L/H	TBUFB0 L/H	07
08	ADO10 L/H	ADO11 L/H			TCMPA1 L/H	TCMPB1 L/H	TBUFA1 L/H	TBUFB1 L/H	0F
10	ADO20 L/H	ADO21 L/H			TCMPA2 L/H	TCMPB2 L/H	TBUFA2 L/H	TBUFB2 L/H	17
18	ADO30 L/H	ADO31 L/H			TCMPA3 L/H	TCMPB3 L/H	TBUFA3 L/H	TBUFB3 L/H	1F
20	ADO40 L/H	ADO41 L/H			DSP0 MIN0L/H	DSP0 MAX0L/H	DSP1 MIN0L/H	DSP1 MAX0L/H	27
28	ADO50 L/H	ADO51 L/H			DSP0 MIN1L/H	DSP0 MAX1L/H	DSP1 MIN1L/H	DSP1 MAX1L/H	2F
30	ADO60 L/H	ADO61 L/H			DSP0 MIN2L/H	DSP0 MAX2L/H	DSP1 MIN2L/H	DSP1 MAX2L/H	37
38	ADO70 L/H	ADO71 L/H							3F
40	ADO80 L/H	ADO81 L/H							47
48	ADO90 L/H	ADO91 L/H							4F
50	ADOA0 L/H	ADOA1 L/H			SPR0	SPR1	SPR2	SPR3	57
58	ADOB0 L/H	ADOB1 L/H			SPR4	SPR5	SPR6	SPR7	5F
60	BUF_MIN0 L/H	BUF_MAX0 L/H	CMP_MIN0 L/H	CMP_MAX0 L/H	CMP_A0 L/H	CMP_B0 L/H	CMP_C0 L/H	CMP_D0 L/H	67
68	BUF_MIN1 L/H	BUF_MAX1 L/H	CMP_MIN1 L/H	CMP_MAX1 L/H	CMP_A1 L/H	CMP_B1 L/H	CMP_C1 L/H	CMP_D1 L/H	6F
70	BUF_MIN2 L/H	BUF_MAX2 L/H	CMP_MIN2 L/H	CMP_MAX2 L/H	CMP_A2 L/H	CMP_B2 L/H	CMP_C2 L/H	CMP_D2 L/H	77
78	BUF_MIN3 L/H	BUF_MAX3 L/H	CMP_MIN3 L/H	CMP_MAX3 L/H	CMP_A3 L/H	CMP_B3 L/H	CMP_C3 L/H	CMP_D3 L/H	7F

The addresses on the SFR, x00 to 0x7F, (see Table 6-3) cannot be accessed from the CPU, but can be accessed from the DSAC and the EPU. For detail of the DSAC and the EPU, see Section 12 and Section 10, respectively.

16-bit SFR registers of the following modules are composed of the lower and higher bytes:

- TinyDSP
- 12-bit ADC
- DAC
- PWM

The lower and higher bytes are assigned to the same address. The 16-bit SFR register is accessed to the lower and higher bytes, in that order. It is required to access continuously to the lower and higher bytes.

If one or more of following statuses occur at the continuously access of CPU to the 16-bit SFR register, the access to the 16-bit SFR register may fail.

(1) When main and interrupt (low or high level) routines access to the same module, SFR, at once.
(2) When low and high level interrupt routines access to the same module, SFR, at once.
(3) When the OCD accesses to the 16-bit SFR during the CPU operation.

To avoid (1) and (2) listed in above, it is required to program as follows:

- Before the 16-bit SFR register is accessed, the INTC must be disabled (INTMST.INTME = 0).
- After the access to the 16-bit SFR register completes, the INTC must be enabled again (INTMST.INTME = 1).

The LSI cannot avoid (3) listed in above. Therefore, the CPU must be stopped in before and after when the 16-bit SFR is accessed by the OCD. It is recommended to stop the CPU using the brake function.

6.3. SPRn (Scratch Pad Register n) (n = 0 to 7)

The scratch pad register (SPR) is for the temporary storage of preliminary data. The SPR has 8 registers (SPR0 to SPR7) that are assigned to the SFR area. The DSAC can access the SPR. Table 6-4 lists the detail of the scratch pad registers.

Table 6-4. Scratch Pad Register

Register	SPR0	Scratch Pad Register0	Address	0x54	0x91
Register	SPR1	Scratch Pad Register1	Address	0x55	0xF9
Register	SPR2	Scratch Pad Register2	Address	0x56	0xFA
Register	SPR3	Scratch Pad Register3	Address	0x57	0xFB
Register	SPR4	Scratch Pad Register4	Address	0x5C	0x8F
Register	SPR5	Scratch Pad Register5	Address	0x5D	0x94
Register	SPR6	Scratch Pad Register6	Address	0x5E	0x9E
Register	SPR7	Scratch Pad Register7	Address	0x5F	0x9F

Bit	Bit Name	R/W	Initial	Description	Remarks
7	SPR	R/W	0	For the temporary storage of preliminary data.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

7. GPIO

The LSI has a general-purpose input/output (GPIO) function. The function of each pin can be defined independently. The signal direction also can be defined independently on the GPIO function setting. Each pin has the pull-up (PUP) or pull-down (PDN) functions that can be set on/off by software. The signal pin settings (function and input/output direction) and the pull-up and pull-down settings are completely independent.

7.1. GPIO Structure

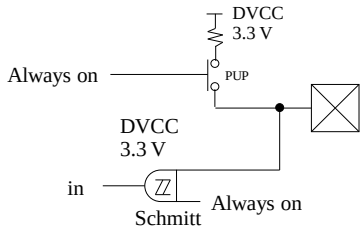
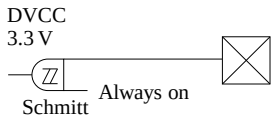
Pin	Description	Signal	Signal Level
$\overline{\text{RES}}$		in	DVCC (3.3 V)
MODE		in	DVCC (3.3 V)

Figure 7-1. GPIO/PIN Structure (Input Pin)

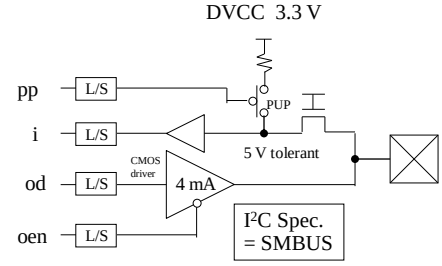
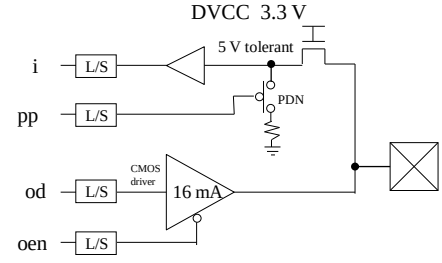
Pin	Description	Signal	Signal Level
DBG GPIO00 ⋮ GPIO05		od pp oen i	DVCC_CORE (1.8 V) DVCC_CORE (1.8 V) DVCC_CORE (1.8 V) DVCC_CORE (1.8 V)
GPIO10 ⋮ GPIO17		od pp oen i	DVCC_CORE (1.8 V) DVCC_CORE (1.8 V) DVCC_CORE (1.8 V) DVCC_CORE (1.8 V)

Figure 7-2. GPIO/PIN Structure (5 V Tolerant Input/Output Pin)

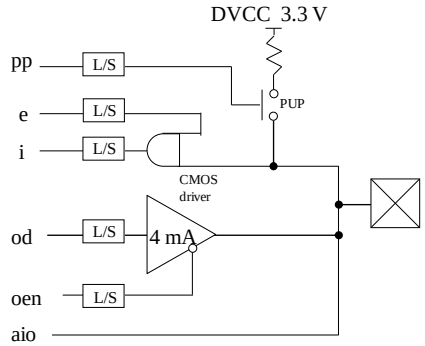
Pin	Description	Signal	Signal Level
GPIO21 GPIO20		od pp oen i e aio	DVCC_CORE (1.8 V) DVCC_CORE (1.8 V) DVCC_CORE (1.8 V) DVCC_CORE (1.8 V) DVCC_CORE (1.8 V) DVCC (3.3 V)

Figure 7-3. GPIO/PIN Structure (Input/Output Pin)

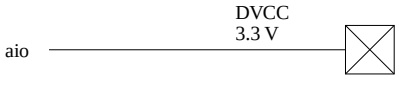
Pin	Description	Signal	Signal Level
ANEX13 ⋮ ANEX6 ANEX3 ⋮ ANEX0		aio	DVCC (3.3 V)

Figure 7-4. GPIO/PIN Structure (Analog Input Pin)

Note that there may be periods of weak driving at the level of about 2 V from the LSI at the falling edge of the input signal as shown in Figure 7-5. This occurs when the signal of the driver that has poor driving capability (i.e., the output impedance of the driver is high) is input to the 5 V tolerant structure pins (DBG, GPIO00 to GPIO05, GPIO10 to GPIO17) set in the input direction of Figure 7-2. The LSI recognizes the period that the 5 V tolerant pin voltage is about 2 V as high level input. This phenomenon does not occur when VIH is less than DVCC + 0.3 V. When VIH is DVCC + 0.3 V or more, check the waveform of the corresponding input pin. If this phenomenon occurs, input the signal of the driver that has high driving capability.

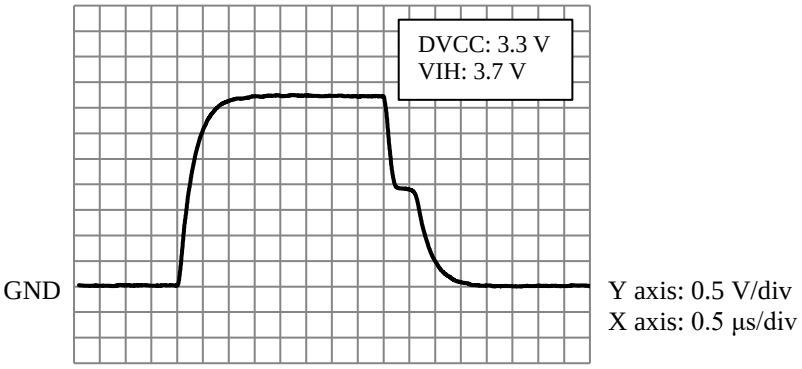


Figure 7-5. Waveform Example of 5 V Tolerant Pin

7.2. Register Descriptions

When a bit is unassigned to the corresponding pin physically, the bit is preliminary (reserved) bit. The read value of the reserved bit is 0. In addition, the write value to the reserved bit must always be 0.

Table 7-1. List of Registers

Symbol	Name	Address	Initial Value
PFS0	Pin Function Select for GPIO0	0xFE00	0x00
PFS1	Pin Function Select for GPIO1	0xFE01	0x00
PFS2	Pin Function Select for GPIO2	0xFE02	0x00
PDD0	Pin Data Direction for GPIO0	0xFE04	0x00
PDD1	Pin Data Direction for GPIO1	0xFE05	0x00
PDD2	Pin Data Direction for GPIO1	0xFE06	0x00
PPU0	Pin Pull Up Control for GPIO0	0xFE08	0x3F
PPU2	Pin Pull Up Control for GPIO2	0xFE0A	0x03
PPD1	Pin Pull Down Control for GPIO1	0xFE0D	0xFF
PIE0	Pin Interrupt Enable for GPIO0	0xFE10	0x00
PIE1	Pin Interrupt Enable for GPIO1	0xFE11	0x00
PIE2	Pin Interrupt Enable for GPIO2	0xFE12	0x00
PIS0	Pin Interrupt Sense for GPIO0	0xFE14	0x00
PIS1	Pin Interrupt Sense for GPIO1	0xFE15	0x00
PIS2	Pin Interrupt Sense for GPIO2	0xFE16	0x00
PIL0	Pin Interrupt Level for GPIO0	0xFE18	0x00
PIL1	Pin Interrupt Level for GPIO1	0xFE19	0x00
PIL2	Pin Interrupt Level for GPIO2	0xFE1A	0x00
PIB0	Pin Interrupt Both Edge for GPIO0	0xFE1C	0x00
PIB1	Pin Interrupt Both Edge for GPIO1	0xFE1D	0x00
PIB2	Pin Interrupt Both Edge for GPIO2	0xFE1E	0x00
PEADC0	ADC Event Select from GPIO0	0xFE20	0x00
PEADC1	ADC Event Select from GPIO1	0xFE21	0x00
PEADC2	ADC Event Select from GPIO2	0xFE22	0x00
PEPWM0	PWM Event Select from GPIO0	0xFE24	0x00
PEPWM1	PWM Event Select from GPIO1	0xFE25	0x00
PEPWM2	PWM Event Select from GPIO2	0xFE26	0x00
PEMETHOD	PWM and ADC Event Gathering Method	0xFE28	0x00
PFSH0	Pin Function Select High for GPIO0	0xFE29	0x00
PFSE1	Pin Function Extend Select for GPIO1	0xFE2A	0x00
PFSE2	Pin Function Extend Select for GPIO2	0xFE2D	0x00
PELUT0	CMPLUT Event Select from GPIO0	0xFE30	0x00
PELUT1	CMPLUT Event Select from GPIO1	0xFE31	0x00
PELUT2	CMPLUT Event Select from GPIO2	0xFE32	0x00
LEMETHOD	CMPLUT Event Gathering Method	0xFE38	0x00
SIS	Serial Input Select	0xFE70	0x00
I2CIS	I^2C Input Select	0xFE71	0x00
TMRIS	TMR Input/Output Select	0xFE72	0x00
CLKIS	Clock Input Select	0xFE73	0x00

Table 7-2. List of SFR Registers

Symbol	Name	Address	Initial Value
PDR0	Pin Data for GPIO0	0x90	0x00
PDR1	Pin Data for GPIO1	0x98	0x00
PDR2	Pin Data for GPIO2	0xB0	0x00
PIF0	Pin Interrupt Flag for GPIO0	0xC8	0x00
PIF1	Pin Interrupt Flag for GPIO1	0xD8	0x00
PIF2	Pin Interrupt Flag for GPIO2	0xE8	0x00

7.2.1. PFS0 (Pin Function Select for GPIO0)

The direction of a pin signal changes according to the function selected.

Register	PFS0		Pin Function Select for GPIO0		Address	0xFE00
Bit	Bit Name	R/W	Initial	Description		Remarks
7	PF3	R/W	0	GPIO03 function selection 00: GPIO03, SPI_SI, TIOB1 input, RXD_B 01: Reserved 10: CMPLUT0 output 11: TIOB1 output		
6		R/W	0			
5	PF2	R/W	0	GPIO02 function selection 00: GPIO02, TIOA1 input 01: TXD_B 10: SPI_SO 11: TIOA1 output		
4		R/W	0			
3	PF1	R/W	0	GPIO01 function selection 00: GPIO01, TIOB3 input, TIC1 01: Reserved 10: SPI_SCK 11: TIOB3 output		
2		R/W	0			
1	PF0	R/W	0	GPIO00 function selection 00: GPIO00, CLKIN_A, TIOA3 input, TIC0 01: Reserved 10: CLKMON 11: TIOA3 output		
0		R/W	0			

7.2.2. PFSH0 (Pin Function Select High for GPIO0)

The direction of a pin signal changes according to the function selected.

Register	PFSH0		Pin Function Select High for GPIO0		Address	0xFE29
Bit	Bit Name	R/W	Initial	Description		Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.		
6	Reserved	R	0	The read value is 0. The write value must always be 0.		
5	Reserved	R	0	The read value is 0. The write value must always be 0.		
4	Reserved	R	0	The read value is 0. The write value must always be 0.		
3	PF5	R/W	0	GPIO05 function selection 00: GPIO05, TIOB2_A input 01: SCL_A (open drain) 10: CMPLUT1 output 11: TIOB2_A output		
2		R/W	0			
1	PF4	R/W	0	GPIO04 function selection 00: GPIO04, TIOA2_A input 01: SDA_A (open drain) 10: CLKMON 11: TIOA2_A output		
0		R/W	0			

7.2.3. PFS1 (Pin Function Select for GPIO1)

The direction of a pin signal changes according to the function selected.

Register	PFS1	Pin Function Select for GPIO1	Address	0xFE01

Bit	Bit Name	R/W	Initial	Description	Remarks
7	PF7	R/W	0	GPIO17 function selection 0: GPIO17, RXD_A, TIOB0 input 1: Other functions When the bit is 1, the function and signal direction of the pin change according to the PFSE1 register setting.	
6	PF6	R/W	0	GPIO16 function selection 0: GPIO16, TIOA0 input 1: Other functions When the bit is 1, the function and signal direction of the pin change according to the PFSE1 register setting.	
5	PF5	R/W	0	GPIO15 function selection 0: GPIO15, SPI_SS_N, TIOB2_B input 1: Other functions When the bit is 1, the function and signal direction of the pin change according to the PFSE1 register setting.	
4	PF4	R/W	0	GPIO14 function selection 0: GPIO14, TIOA2_B input 1: Other functions When the bit is 1, the function and signal direction of the pin change according to the PFSE1 register setting.	
3	PF3	R/W	0	GPIO13 function selection 0: GPIO13 1: PWM1L	
2	PF2	R/W	0	GPIO12 function selection 0: GPIO12 1: PWM1H	
1	PF1	R/W	0	GPIO11 function selection 0: GPIO11 1: PWM0L	
0	PF0	R/W	0	GPIO10 function selection 0: GPIO10 1: PWM0H	

7.2.4. PFSE1 (Pin Function Extend Select for GPIO1)

The direction of a pin signal changes according to the function selected.

Register	PFSE1	Pin Function Extend Select for GPIO1	Address	0xFE2A

Bit	Bit Name	R/W	Initial	Description	Remarks
7	PF7	R/W	0	GPIO17 function selection 00: PWM3L 01: TIOB0 output 10: Reserved 11: Reserved	
6		R/W	0		
5	PF6	R/W	0	GPIO16 function selection 00: PWM3H 01: TXD_A 10: Reserved 11: TIOA0 output	
4		R/W	0		
3	PF5	R/W	0	GPIO15 function selection 00: PWM2L 01: TIOB2_B output 10: SCL_B 11: Reserved	
2		R/W	0		
1	PF4	R/W	0	GPIO14 function selection 00: PWM2H 01: Reserved 10: SDA_B 11: TIOA2_B output	
0		R/W	0		

7.2.5. PFS2 (Pin Function Select for GPIO2)

The direction of a pin signal changes according to the function selected.

Register	PFS2	Pin Function Select for GPIO2	Address	0xFE02

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	PF1	R/W	0	GPIO21 function selection 0: GPIO21 1: Other functions When the bit is 1, the function and signal direction of the pin change according to the PFSE2 register setting.	
0	PF0	R/W	0	GPIO20 function selection 0: GPIO20, CLKIN_B 1: Analog input	

7.2.6. PFSE2 (Pin Function Extend Select for GPIO2)

The direction of a pin signal changes according to the function selected.

Register	PFSE2		Pin Function Extend Select for GPIO2	Address	0xFE2D
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	PF1	R/W	0	GPIO21 function selection 00: Analog input 01: CLKMON 10: Reserved 11: Reserved	
2		R/W	0		
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

7.2.7. PDD0 (Pin Data Direction for GPIO0)

Register	PDD0		Pin Data Direction for GPIO0	Address	0xFE04
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
6	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
5	DD5	R/W	0	Selection of the direction of a pin signal 0: Input 1: Output Only when the pin is set to have the GPIO function, each bit setting is valid.	
4	DD4	R/W	0		
3	DD3	R/W	0		
2	DD2	R/W	0		
1	DD1	R/W	0		
0	DD0	R/W	0		

7.2.8. PDD1 (Pin Data Direction for GPIO1)

Register	PDD1		Pin Data Direction for GPIO1	Address	0xFE05
Bit	Bit Name	R/W	Initial	Description	Remarks
7	DD7	R/W	0	Selection of the direction of a pin signal 0: Input 1: Output Only when the pin is set to have the GPIO function, each bit setting is valid.	
6	DD6	R/W	0		
5	DD5	R/W	0		
4	DD4	R/W	0		
3	DD3	R/W	0		
2	DD2	R/W	0		
1	DD1	R/W	0		
0	DD0	R/W	0		

7.2.9. PDD2 (Pin Data Direction for GPIO2)

Register	PDD2		Pin Data Direction for GPIO2	Address	0xFE06
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	DD1	R/W	0	Selection of the direction of a pin signal 0: Input 1: Output Only when the pin is set to have the GPIO function, each bit setting is valid.	
0	DD0	R/W	0		

7.2.10. PDR0 (Pin Data for GPIO0)

Register	PDR0		Pin Data for GPIO0		Address	0x90
Bit	Bit Name	R/W	Initial	Description		Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.		
6	Reserved	R	0	The read value is 0. The write value must always be 0.		
5	PD5	R/W	0	States of pin signals Read 0: The pin signal is at a low level Read 1: The pin signal is at a high level Write 0: The pin signal is set to low when the pin is set as a GPIO output Write 1: The pin signal is set to high when the pin is set as a GPIO output When the pin is set as a GPIO input or a digital function other than the GPIO, the read value indicates the level of an external pin, and writing to these bits has no effect. When the pin is set as a GIPIO output, the write value is an output level of the pin, and the read value indicates the level of an external pin. For example, if an external signal conflicts with the output level of the pin, the level of the read value may not become that of the write value. When the pin is set as an analog function, the read value is always 0, and writing to these bits has no effect. Further, the write value is stored into an internal register. Thus, any changes to the function and the signal direction of the pin may cause an internal register value to affect an output pin level.		
4	PD4	R/W	0			
3	PD3	R/W	0			
2	PD2	R/W	0			
1	PD1	R/W	0			
0	PD0	R/W	0			

7.2.11. PDR1 (Pin Data for GPIO1)

Register	PDR1		Pin Data for GPIO1		Address	0x98
Bit	Bit Name	R/W	Initial	Description		Remarks
7	PD7	R/W	0	States of pin signals Read 0: The pin signal is at a low level Read 1: The pin signal is at a high level Write 0: The pin signal is set to low when the pin is set as a GPIO output Write 1: The pin signal is set to high when the pin is set as a GPIO output When the pin is set as a GPIO input or a digital function other than the GPIO, the read value indicates the level of an external pin, and writing to these bits has no effect. When the pin is set as a GIPIO output, the write value is an output level of the pin, and the read value indicates the level of an external pin. For example, if an external signal conflicts with the output level of the pin, the level of the read value may not become that of the write value. When the pin is set as an analog function, the read value is always 0, and writing to these bits has no effect. Further, the write value is stored into an internal register. Thus, any changes to the function and the signal direction of the pin may cause an internal register value to affect an output pin level.		
6	PD6	R/W	0			
5	PD5	R/W	0			
4	PD4	R/W	0			
3	PD3	R/W	0			
2	PD2	R/W	0			
1	PD1	R/W	0			
0	PD0	R/W	0			

7.2.12. PDR2 (Pin Data for GPIO2)

Register	PDR2		Pin Data for GPIO2	Address	0xB0
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	PD1	R/W	0	States of pin signals Read 0: The pin signal is at a low level Read 1: The pin signal is at a high level Write 0: The pin signal is set to low when the pin is set as a GPIO output Write 1: The pin signal is set to high when the pin is set as a GPIO output When the pin is set as a GPIO input or a digital function other than the GPIO, the read value indicates the level of an external pin, and writing to these bits has no effect. When the pin is set as a GIPIO output, the write value is an output level of the pin, and the read value indicates the level of an external pin. For example, if an external signal conflicts with the output level of the pin, the level of the read value may not become that of the write value. When the pin is set as an analog function, the read value is always 0, and writing to these bits has no effect. Further, the write value is stored into an internal register. Thus, any changes to the function and the signal direction of the pin may cause an internal register value to affect an output pin level.	
0	PD0	R/W	0		

7.2.13. PPU0 (Pin Pull Up Control for GPIO0)

Register	PPU0		Pin Pull Up Control for GPIO0	Address	0xFE08
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	PPU5	R/W	1	Pin pull-up enable 0: Pull-up MOSFET is disabled 1: Pull-up MOSFET is enabled	
4	PPU4	R/W	1		
3	PPU3	R/W	1		
2	PPU2	R/W	1		
1	PPU1	R/W	1		
0	PPU0	R/W	1		

7.2.14. PPU2 (Pin Pull Up Control for GPIO2)

Register	PPU2		Pin Pull Up Control for GPIO2	Address	0xFE0A
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	PPU1	R/W	1	Pin pull-up enable 0: Pull-up MOSFET is disabled 1: Pull-up MOSFET is enabled	
0	PPU0	R/W	1		

7.2.15. PPD1 (Pin Pull Down Control for GPIO1)

Register	PPD1		Pin Pull Down Control for GPIO1	Address	0xFE0D
Bit	Bit Name	R/W	Initial	Description	Remarks
7	PPD7	R/W	1	Pin pull-down enable 0: Pull-down MOSFET is disabled 1: Pull-down MOSFET is enabled	
6	PPD6	R/W	1		
5	PPD5	R/W	1		
4	PPD4	R/W	1		
3	PPD3	R/W	1		
2	PPD2	R/W	1		
1	PPD1	R/W	1		
0	PPD0	R/W	1		

7.2.16. PIE0 (Pin Interrupt Enable for GPIO0)

Register	PIE0		Pin Interrupt Enable for GPIO0	Address	0xFE10
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	PIE5	R/W	0	Pin interrupt enable 0: Pin interrupt is disabled 1: Pin interrupt is enabled When the pin is set as an analog function, the PIE0 to PIE5 bits are ignored. When the pin is set as a digital function, the PIE0 to PIE5 bits are enabled regardless the function and signal direction of the pin.	
4	PIE4	R/W	0		
3	PIE3	R/W	0		
2	PIE2	R/W	0		
1	PIE1	R/W	0		
0	PIE0	R/W	0		

7.2.17. PIE1 (Pin Interrupt Enable for GPIO1)

Register	PIE1		Pin Interrupt Enable for GPIO1	Address	0xFE11
Bit	Bit Name	R/W	Initial	Description	Remarks
7	PIE7	R/W	0	Pin interrupt enable 0: Pin interrupt is disabled 1: Pin interrupt is enabled When the pin is set as an analog function, the PIE0 to PIE7 bits are ignored. When the pin is set as a digital function, the PIE0 to PIE7 bits are enabled regardless the function and signal direction of the pin. This means that pin interrupts can be generated by PWM toggle operation.	
6	PIE6	R/W	0		
5	PIE5	R/W	0		
4	PIE4	R/W	0		
3	PIE3	R/W	0		
2	PIE2	R/W	0		
1	PIE1	R/W	0		
0	PIE0	R/W	0		

7.2.18. PIE2 (Pin Interrupt Enable for GPIO2)

Register	PIE2		Pin Interrupt Enable for GPIO2	Address	0xFE12
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	PIE1	R/W	0	Pin interrupt enable 0: Pin interrupt is disabled 1: Pin interrupt is enabled When the pin is set as an analog function, the PIE0 to PIE1 bits are ignored. When the pin is set as a digital function, the PIE0 to PIE1 bits are enabled regardless the function and signal direction of the pin.	
0	PIE0	R/W	0		

7.2.19. PIF0 (Pin Interrupt Flag for GPIO0)

Register	PIF0		Pin Interrupt Flag for GPIO0	Address	0xC8
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	PIF5	R/C	0	Pin interrupt flag setting Read 0: No interrupt request Read 1: An interrupt event is generated Write 0: No change Write 1: The corresponding bit is cleared The PIFx bit is asserted regardless of the value of the PIEx bit.	
4	PIF4	R/C	0		
3	PIF3	R/C	0		
2	PIF2	R/C	0		
1	PIF1	R/C	0		
0	PIF0	R/C	0		

7.2.20. PIF1 (Pin Interrupt Flag for GPIO1)

Register	PIF1		Pin Interrupt Flag for GPIO1	Address	0xD8
Bit	Bit Name	R/W	Initial	Description	Remarks
7	PIF7	R/C	0	Pin interrupt flag setting Read 0: No interrupt request Read 1: An interrupt event is generated Write 0: No change Write 1: The corresponding bit is cleared The PIFx bit is asserted regardless of the value of the PIEx bit.	
6	PIF6	R/C	0		
5	PIF5	R/C	0		
4	PIF4	R/C	0		
3	PIF3	R/C	0		
2	PIF2	R/C	0		
1	PIF1	R/C	0		
0	PIF0	R/C	0		

7.2.21. PIF2 (Pin Interrupt Flag for GPIO2)

Register	PIF2		Pin Interrupt Flag for GPIO2	Address	0xE8
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	PIF1	R/C	0	Pin interrupt flag setting Read 0: No interrupt request Read 1: An interrupt event is generated Write 0: No change Write 1: The corresponding bit is cleared The PIFx bit is asserted regardless of the value of the PIEx bit.	
0	PIF0	R/C	0		

7.2.22. PIS0 (Pin Interrupt Sense for GPIO0)

Register	PIS0		Pin Interrupt Sense for GPIO0	Address	0xFE14
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	PIS5	R/W	0	Selection of pin interrupt detection 0: Level detection 1: Edge detection For ensuring the proper operation of the wakeup function, an analog delay circuit with an appropriate noise filter is implemented for the edge detection.	
4	PIS4	R/W	0		
3	PIS3	R/W	0		
2	PIS2	R/W	0		
1	PIS1	R/W	0		
0	PIS0	R/W	0		

7.2.23. PIS1 (Pin Interrupt Sense for GPIO1)

Register	PIS1		Pin Interrupt Sense for GPIO1	Address	0xFE15
Bit	Bit Name	R/W	Initial	Description	Remarks
7	PIS7	R/W	0	Selection of pin interrupt detection 0: Level detection 1: Edge detection For the proper operation of the wakeup function, an analog delay circuit with an appropriate noise filter is implemented for the edge detection.	
6	PIS6	R/W	0		
5	PIS5	R/W	0		
4	PIS4	R/W	0		
3	PIS3	R/W	0		
2	PIS2	R/W	0		
1	PIS1	R/W	0		
0	PIS0	R/W	0		

7.2.24. PIS2 (Pin Interrupt Sense for GPIO2)

Register	PIS2		Pin Interrupt Sense for GPIO2	Address	0xFE16
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	PIS1	R/W	0	Selection of pin interrupt detection 0: Level detection 1: Edge detection For the proper operation of the wakeup function, an analog delay circuit with an appropriate noise filter is implemented for the edge detection.	
0	PIS0	R/W	0		

7.2.25. PIL0 (Pin Interrupt Level for GPIO0)

Register	PIL0		Pin Interrupt Level for GPIO0	Address	0xFE18
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	PIL5	R/W	0	Selection of pin interrupt levels 0: Low level, falling edge 1: High level, rising edge While the edge detection is selected, the PIBx bit has a higher priority.	
4	PIL4	R/W	0		
3	PIL3	R/W	0		
2	PIL2	R/W	0		
1	PIL1	R/W	0		
0	PIL0	R/W	0		

7.2.26. PIL1 (Pin Interrupt Level for GPIO1)

Register	PIL1		Pin Interrupt Level for GPIO1	Address	0xFE19
Bit	Bit Name	R/W	Initial	Description	Remarks
7	PIL7	R/W	0	Selection of pin interrupt levels 0: Low level, falling edge 1: High level, rising edge While the edge detection is selected, the PIBx bit has a higher priority.	
6	PIL6	R/W	0		
5	PIL5	R/W	0		
4	PIL4	R/W	0		
3	PIL3	R/W	0		
2	PIL2	R/W	0		
1	PIL1	R/W	0		
0	PIL0	R/W	0		

7.2.27. PIL2 (Pin Interrupt Level for GPIO2)

Register	PIL2		Pin Interrupt Level for GPIO2	Address	0xFE1A
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	PIL1	R/W	0	Selection of pin interrupt levels 0: Low level, falling edge 1: High level, rising edge While the edge detection is selected, the PIBx bit has a higher priority.	
0	PIL0	R/W	0		

7.2.28. PIB0 (Pin Interrupt Both Edge for GPIO0)

Register	PIB0		Pin Interrupt Both Edge for GPIO0	Address	0xFE1C
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	PIB5	R/W	0	Selection of both edges for pin interrupt 0: Falling or rising edge set by the PILx bit 1: Both edges that ignore the setting of the PILx bit Only while the edge detection is selected, the settings of these bits are valid.	
4	PIB4	R/W	0		
3	PIB3	R/W	0		
2	PIB2	R/W	0		
1	PIB1	R/W	0		
0	PIB0	R/W	0		

7.2.29. PIB1 (Pin Interrupt Both Edge for GPIO1)

Register	PIB1		Pin Interrupt Both Edge for GPIO1	Address	0xFE1D
Bit	Bit Name	R/W	Initial	Description	Remarks
7	PIB7	R/W	0	Selection of both edges for pin interrupt 0: Falling or rising edge set by the PILx bit 1: Both edges that ignore the setting of the PILx bit Only while the edge detection is selected, the settings of these bits are valid.	
6	PIB6	R/W	0		
5	PIB5	R/W	0		
4	PIB4	R/W	0		
3	PIB3	R/W	0		
2	PIB2	R/W	0		
1	PIB1	R/W	0		
0	PIB0	R/W	0		

7.2.30. PIB2 (Pin Interrupt Both Edge for GPIO2)

Register	PIB2		Pin Interrupt Both Edge for GPIO2	Address	0xFE1E
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
6	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
5	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
4	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
3	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
2	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
1	PIB1	R/W	0	Selection of both edges for pin interrupt 0: Falling or rising edge set by the PILx bit 1: Both edges that ignore the setting of the PILx bit Only while the edge detection is selected, the settings of these bits are valid.	
0	PIB0	R/W	0		

7.2.31. PEADC0 (ADC Event Select from GPIO0)

Register	PEADC0		ADC Event Select from GPIO0	Address	0xFE20
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	EVTADC5	R/W	0	ADC event selection 0: An input signal is not used for ADC event selection 1: An input signal is used for ADC event selection A GPIO input signal is selected by the EVTADC0 to EVTADC5 bits. The LSI computes the selected input signal and then generates an event signal. The generated event signal is used as a trigger for all of ADC events. Computing methods include the logical OR and the logical AND; the PEMETHOD.MPEADC0 bit defines which method to be employed.	
4	EVTADC4	R/W	0		
3	EVTADC3	R/W	0		
2	EVTADC2	R/W	0		
1	EVTADC1	R/W	0		
0	EVTADC0	R/W	0		

7.2.32. PEADC1 (ADC Event Select from GPIO1)

Register	PEADC1		ADC Event Select from GPIO1	Address	0xFE21
Bit	Bit Name	R/W	Initial	Description	Remarks
7	EVTADC7	R/W	0	ADC event selection 0: An input signal is not used for ADC event selection 1: An input signal is used for ADC event selection A GPIO input signal is selected by the EVTADC0 to EVTADC7 bits. The LSI computes the selected input signal and then generates an event signal. The generated event signal is used as a trigger for all of ADC events. Computing methods include the logical OR and the logical AND; the PEMETHOD.MPEADC1 bit defines which method to be employed.	
6	EVTADC6	R/W	0		
5	EVTADC5	R/W	0		
4	EVTADC4	R/W	0		
3	EVTADC3	R/W	0		
2	EVTADC2	R/W	0		
1	EVTADC1	R/W	0		
0	EVTADC0	R/W	0		

7.2.33. PEADC2 (ADC Event Select from GPIO2)

Register	PEADC2		ADC Event Select from GPIO2	Address	0xFE22
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	EVTADC1	R/W	0	ADC event selection 0: An input signal is not used for ADC event selection 1: An input signal is used for ADC event selection A GPIO input signal is selected by the EVTADC0 to EVTADC1 bits. The LSI computes the selected input signal and then generates an event signal. The generated event signal is used as a trigger for all of ADC events. Computing methods include the logical OR and the logical AND; the PEMETHOD.MPEADC2 bit defines which method to be employed.	
0	EVTADC0	R/W	0		

7.2.34. PEPWM0 (PWM Event Select from GPIO0)

Register	PEPWM0		PWM Event Select from GPIO0	Address	0xFE24
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
6	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
5	EVTPWM5	R/W	0	PWM event selection 0: An input signal is not used for PWM event selection 1: An input signal is used for PWM event selection A GPIO input signal is selected by the EVTPWM0 to EVTPWM5 bits. The LSI computes the selected input signal and then generates an event signal. The LSI-generated event signals are all used for PWM event signals. Computing methods include the logical OR and the logical AND; the PEMETHOD.MPEPWM0 bit defines which method to be employed.	
4	EVTPWM4	R/W	0		
3	EVTPWM3	R/W	0		
2	EVTPWM2	R/W	0		
1	EVTPWM1	R/W	0		
0	EVTPWM0	R/W	0		

7.2.35. PEPWM1 (PWM Event Select from GPIO1)

Register	PEPWM1		PWM Event Select from GPIO1	Address	0xFE25
Bit	Bit Name	R/W	Initial	Description	Remarks
7	EVTPWM7	R/W	0	PWM event selection 0: An input signal is not used for PWM event selection 1: An input signal is used for PWM event selection A GPIO input signal is selected by the EVTPWM0 to EVTPWM7 bits. The LSI computes the selected input signal and then generates an event signal. The LSI-generated event signals are all used for PWM event signals. Computing methods include the logical OR and the logical AND; the PEMETHOD.MPEPWM1 bit defines which method to be employed.	
6	EVTPWM6	R/W	0		
5	EVTPWM5	R/W	0		
4	EVTPWM4	R/W	0		
3	EVTPWM3	R/W	0		
2	EVTPWM2	R/W	0		
1	EVTPWM1	R/W	0		
0	EVTPWM0	R/W	0		

7.2.36. EPWM2 (PWM Event Select from GPIO2)

Register	PEPWM2		PWM Event Select from GPIO2	Address	0xFE26
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	EVTPWM1	R/W	0	PWM event selection 0: An input signal is not used for PWM event selection 1: An input signal is used for PWM event selection A GPIO input signal is selected by the EVTPWM0 to EVTPWM1 bits. The LSI computes the selected input signal and then generates an event signal. The LSI-generated event signals are all used for PWM event signals. Computing methods include the logical OR and the logical AND; the PEMETHOD.MPEPWM2 bit defines which method to be employed.	
0	EVTPWM0	R/W	0		

7.2.37. PELUT0 (LUT Event Select from GPIO0)

Register	PELUT0		LUT Event Select from GPIO0	Address	0xFE30
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
6	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
5	EVTLUT5	R/W	0	LUT event selection 0: An input signal is not used for LUT event selection 1: An input signal is used for LUT event selection A GPIO input signal is selected by the EVTLUT0 to EVTLUT5 bits. The LSI computes the selected input signal and then generates an event signal. The LSI-generated event signals are all used for LUT event signals. Computing methods include the logical OR and the logical AND; the LEMETHOD.MPELUT0 bit defines which method to be employed.	
4	EVTLUT4	R/W	0		
3	EVTLUT3	R/W	0		
2	EVTLUT2	R/W	0		
1	EVTLUT1	R/W	0		
0	EVTLUT0	R/W	0		

7.2.38. PELUT1 (LUT Event Select from GPIO1)

Register	PELUT1		LUT Event Select from GPIO1	Address	0xFE31
Bit	Bit Name	R/W	Initial	Description	Remarks
7	EVTLUT7	R/W	0	LUT event selection 0: An input signal is not used for LUT event selection 1: An input signal is used for LUT event selection A GPIO input signal is selected by the EVTLUT0 to EVTLUT7 bits. The LSI computes the selected input signal and then generates an event signal. The LSI-generated event signals are all used for LUT event signals. Computing methods include the logical OR and the logical AND; the LEMETHOD.MPELUT1 bit defines which method to be employed.	
6	EVTLUT6	R/W	0		
5	EVTLUT5	R/W	0		
4	EVTLUT4	R/W	0		
3	EVTLUT3	R/W	0		
2	EVTLUT2	R/W	0		
1	EVTLUT1	R/W	0		
0	EVTLUT0	R/W	0		

7.2.39. PELUT2 (LUT Event Select from GPIO2)

Register	PELUT2		LUT Event Select from GPIO2	Address	0xFE32
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	EVTLUT1	R/W	0	LUT event selection 0: An input signal is not used for LUT event selection 1: An input signal is used for LUT event selection A GPIO input signal is selected by the EVTLUT0 to EVTLUT1 bits. The LSI computes the selected input signal and then generates an event signal. The LSI-generated event signals are all used for LUT event signals. Computing methods include the logical OR and the logical AND; the LEMETHOD.MPELUT2 bit defines which method to be employed.	
0	EVTLUT0	R/W	0		

7.2.40. PEMETHOD (PWM and ADC Event Gathering Method)

Register	PEMETHOD		PWM Event Gathering Method		Address	0xFE28
Bit	Bit Name	R/W	Initial	Description		Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.		
6	MPEPWM2	R/W	0	Computing method for PWM event generation 0: Logical OR 1: Logical AND Each bit corresponds to the PEPWM0 to PEPWM2 registers.		
5	MPEPWM1	R/W	0			
4	MPEPWM0	R/W	0			
3	Reserved	R	0	The read value is 0. The write value must always be 0.		
2	MPEADC2	R/W	0	Computing method for ADC event generation 0: Logical OR 1: Logical AND Each bit corresponds to the PEADC0 to PEADC2 registers.		
1	MPEADC1	R/W	0			
0	MPEADC0	R/W	0			

7.2.41. LEMETHOD (CMPLUT Event Gathering Method)

Register	LEMETHOD		CMPLUT Event Gathering Method		Address	0xFE38
Bit	Bit Name	R/W	Initial	Description		Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.		
6	Reserved	R	0	The read value is 0. The write value must always be 0.		
5	Reserved	R	0	The read value is 0. The write value must always be 0.		
4	Reserved	R	0	The read value is 0. The write value must always be 0.		
3	Reserved	R	0	The read value is 0. The write value must always be 0.		
2	MPELUT2	R/W	0	Computing method for LUT event generation 0: Logical OR 1: Logical AND Each bit corresponds to the PELUT0 to PELUT2 registers.		
1	MPELUT1	R/W	0			
0	MPELUT0	R/W	0			

7.2.42. SIS (Serial Input Select)

Register	SIS		Serial Input Select Register	Address	0xFE70
Bit	Bit Name	R/W	Initial	Description	Remarks
7	RXDS	R/W	0	Selection of RXD 00: RXD_A 01: RXD_B 10: Reserved 11: Reserved	
6		R/W	0		
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

7.2.43. I2CIS (I^2C Input Select)

Register	I2CIS		I^2C Input Select Register	Address	0xFE71
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	SDAS	R/W	0	Selection of SDA 00: SDA_A 01: SDA_B 10: Reserved 11: Reserved	
2		R/W	0		
1	SCLS	R/W	0	Selection of SCL 00: SCL_A 01: SCL_B 10: Reserved 11: Reserved	
0		R/W	0		

7.2.44. TMRIS (TMR Input/Output Select)

Register	TMRIS		TMR Input/Output Select Register	Address	0xFE72
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	TMR2S_B	R	0	Selection of TIOB2 I/O 00: TIOB2_A 01: TIOB2_B 10: Reserved 11: Reserved	
2		R/W	0		
1	TMR2S_A	R	0	Selection of TIOA2 I/O 00: TIOA2_A 01: TIOA2_B 10: Reserved 11: Reserved	
0		R/W	0		

7.2.45. CLKIS (Clock Input Select)

Register	CLKIS		CLKIN Input Select Register	Address	0xFE73
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	CLKS	R/W	0	Selection of CLKIN 00: CLKIN_A 01: CLKIN_B 10: Reserved 11: Reserved	
0		R/W	0		

7.3. Usage Notes and Restrictions

7.3.1. Reading from PDRx Register after Writing to PDRx Register

When reading out the PDRx register after writing to the PDRx register, two or more wait cycles must be inserted between the read and write instructions. The following is the sample code of this operation.

```
mov PDR1, #0xaa  ; Writing to PDR1
nop              ; Wait cycle
nop              ; Wait cycle
mov a, PDR1      ; Reading from PDR1
```

7.3.2. Setting the Pin Functions: GPIO14, GPIO15, GPIO16, GPIO17, and GPIO21

To define the functions for the GPIO14, GPIO15, GPIO16, and GPIO17 pins, follow the steps below to set the PFSE1 and PFS1 registers.

(1) The PWM, communication interface (SPI or UART), and TMR functions are allocated to the above-mentioned pins. To use thsese functions, set the PFSE1 register. When the GPIO function is selected, the PFSE1 register setting is not needed.
(2) The GPIO function or the function configured by the PFSE1 register is selected by the PFS1 register.

To define the functions for the GPIO21, follow the steps below to set the PFSE2 and PFS2 registers.

(1) Either an analog input or the CLKMON register function is allocated to the GPIO21 pin. To use the function, set the PFSE2 register. When the GPIO function is selected, the PFSE2 register setting is not needed.
(2) The GPIO function or the function configured by the PFSE2 register is selected by the PFS2 register.

8. Event Connection

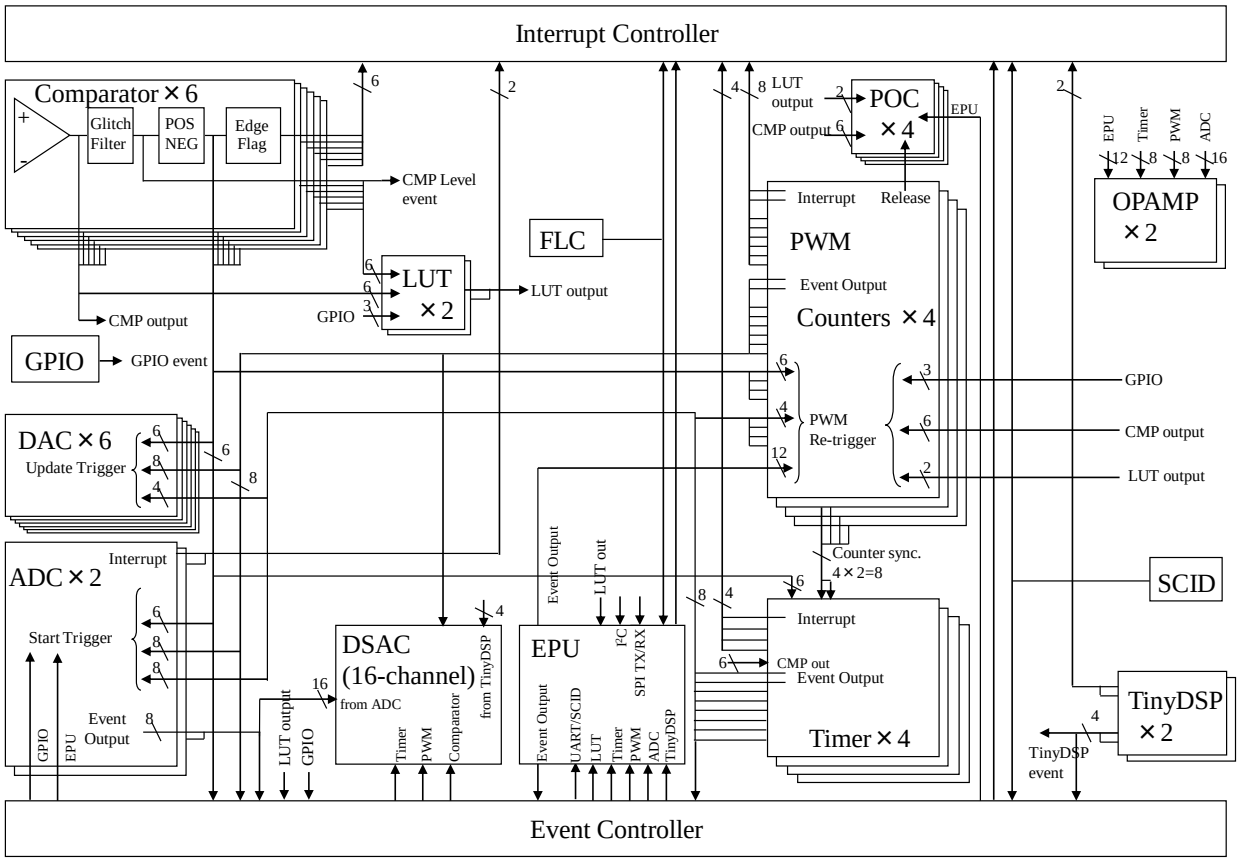


Figure 8-1. LSI Internal Event Connection

9. Event Controller (EVC)

9.1. Overview

The event controller (EVC) has the following functions.

- Sets the event connection between functional modules.
- Generates the interrupt to the CPU from the output event of EPU.
- Generates the GPIO0/1/2 edge event for the ADC0/1.

Table 9-1 EVC Functional Descriptions

Item	Description
Interrupt Generation	- EPU output event 14 per thread
Event Selection	Selects the input event to functional modules. - EPU - ADC - DSAC
Event Integration	Selects multiple events and integrates them into one signal (OR). - POC

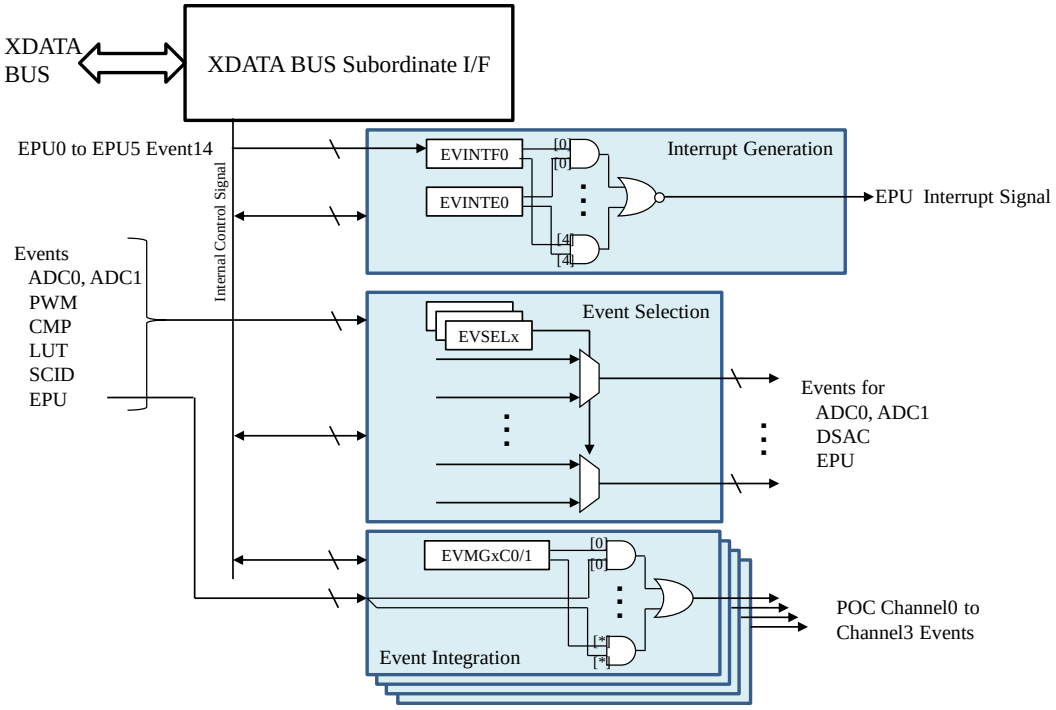


Figure 9-1. EVC Block Diagram

9.2. Operation

9.2.1. Interrupt Generation

The interrupts to the CPU are generated using the output event 14 of each EPU thread. When the event is detected, the corresponding bit in the EVINTF0 register is set. When the corresponding bit in the EVINTE0 register is 1 (i.e., enabled) during EVINTF0 = 1, the interrupt to the CPU is generated.

9.2.2. Event Selection

The event input to the EPU, ADC, and DSAC is selected by the EVSELn register. Table 9-2, Table 9-3, Table 9-4 show the input selection tables of the EPU, ADC, and DSAC, respectively. In these tables, an event type name with a slash represents an event which has 2 event types. An event type name before the slash is the event type when EVSELn.EPUmSEL = 0; an event type name after the slash is the event type when EVSELn.EPUmSEL = 1. Taking Table 9-2 as an example, if EVSEL0.EPU0SEL = 0, the event type is TinyDSP0_0; if EVSEL0.EPU0SEL = 1, the event type is TinyDSP0_1.

Table 9-2. EPU Event Input

Event No.	EPU Thread0		EPU Thread1		EPU Thread2	
	Event Type	Setting Register	Event Type	Setting Register	Event Type	Setting Register
0	TinyDSP0_0/ TinyDSP0_1	EVSEL0.EPU0SEL	TinyDSP0_0/ TinyDSP0_1	EVSEL0.EPU1SEL	TinyDSP0_0/ TinyDSP0_1	EVSEL0.EPU2SEL
1	TinyDSP1_0/ TinyDSP1_1		TinyDSP1_0/ TinyDSP1_1		TinyDSP1_0/ TinyDSP1_1	
2	ADC0_0/ADC0_1	EVSEL1.EPU0SEL	ADC0_0/ADC0_1	EVSEL1.EPU1SEL	ADC0_0/ADC0_1	EVSEL1.EPU2SEL
3	ADC0_2/ADC0_3		ADC0_2/ADC0_3		ADC0_2/ADC0_3	
4	ADC0_4/ADC0_5		ADC0_4/ADC0_5		ADC0_4/ADC0_5	
5	ADC0_6/ADC0_7		ADC0_6/ADC0_7		ADC0_6/ADC0_7	
6	ADC1_0/ADC1_1		ADC1_0/ADC1_1		ADC1_0/ADC1_1	
7	ADC1_2/ADC1_3		ADC1_2/ADC1_3		ADC1_2/ADC1_3	
8	ADC1_4/ADC1_5		ADC1_4/ADC1_5		ADC1_4/ADC1_5	
9	ADC1_6/ADC1_7		ADC1_6/ADC1_7		ADC1_6/ADC1_7	
10	PWM0_0/PWM0_1	EVSEL2.EPU0SEL	PWM0_0/PWM0_1	EVSEL2.EPU1SEL	PWM0_0/PWM0_1	EVSEL2.EPU2SEL
11	PWM1_0/PWM1_1		PWM1_0/PWM1_1		PWM1_0/PWM1_1	
12	PWM2_0/PWM2_1		PWM2_0/PWM2_1		PWM2_0/PWM2_1	
13	PWM3_0/PWM3_1		PWM3_0/PWM3_1		PWM3_0/PWM3_1	
14	LUT0	—	LUT0/LUT1	EVSEL3.EPU1SEL	LUT0/LUT1	EVSEL3.EPU2SEL
15	LUT1	—	TMR0A/TMR0B	EVSEL4.EPU1SEL	TMR1A/TMR1B	EVSEL4.EPU2SEL

Event No.	EPU Thread3		EPU Thread4		EPU Thread5	
	Event Type	Setting Register	Event Type	Setting Register	Event Type	Setting Register
0	TinyDSP0_0/ TinyDSP0_1	EVSEL0.EPU3SEL	TinyDSP0_0/ TinyDSP0_1	EVSEL0.EPU4SEL	TinyDSP0_0/ TinyDSP0_1	EVSEL0.EPU5SEL
1	TinyDSP1_0/ TinyDSP1_1		TinyDSP1_0/ TinyDSP1_1		TinyDSP1_0/ TinyDSP1_1	
2	ADC0_0/ADC0_1	EVSEL1.EPU3SEL	ADC0_0/ADC0_1	EVSEL1.EPU4SEL	ADC0_0/ADC0_1	EVSEL1.EPU5SEL
3	ADC0_2/ADC0_3		ADC0_2/ADC0_3		ADC0_2/ADC0_3	
4	ADC0_4/ADC0_5		ADC0_4/ADC0_5		ADC0_4/ADC0_5	
5	ADC0_6/ADC0_7		ADC0_6/ADC0_7		ADC0_6/ADC0_7	
6	ADC1_0/ADC1_1		ADC1_0/ADC1_1		ADC1_0/ADC1_1	
7	ADC1_2/ADC1_3		ADC1_2/ADC1_3		ADC1_2/ADC1_3	
8	ADC1_4/ADC1_5		ADC1_4/ADC1_5		ADC1_4/ADC1_5	
9	ADC1_6/ADC1_7		ADC1_6/ADC1_7		ADC1_6/ADC1_7	
10	PWM0_0/PWM0_1	EVSEL2.EPU3SEL	PWM0_0/PWM0_1	EVSEL2.EPU4SEL	PWM0_0/PWM0_1	EVSEL2.EPU5SEL
11	PWM1_0/PWM1_1		PWM1_0/PWM1_1		PWM1_0/PWM1_1	
12	PWM2_0/PWM2_1		PWM2_0/PWM2_1		PWM2_0/PWM2_1	
13	PWM3_0/PWM3_1		PWM3_0/PWM3_1		PWM3_0/PWM3_1	
14	SPI_TX | SPI_RX	—	I^2C	—	UART/SCID	EVSEL3.EPU5SEL
15	TMR2A/TMR2B	EVSEL4.EPU3SEL	TMR3A/TMR3B	EVSEL4.EPU4SEL	FLASH	—

Table 9-3. ADC Event Input

Event No.	ADC Unit0		ADC Unit1	
	Event Type	Setting Register	Event Type	Setting Register
1	GPIO0 rise	—	GPIO0 rise	—
2	GPIO0 fall	—	GPIO0 fall	—
3	GPIO0 both	—	GPIO0 both	—
4	GPIO1 rise	—	GPIO1 rise	—
5	GPIO1 fall	—	GPIO1 fall	—
6	GPIO1 both	—	GPIO1 both	—
7	GPIO2 rise	—	GPIO2 rise	—
8	GPIO2 fall	—	GPIO2 fall	—
9	GPIO2 both	—	GPIO2 both	—
10	CMP0	—	CMP0	—
11	CMP1	—	CMP1	—
12	CMP2	—	CMP2	—
13	CMP3	—	CMP3	—
14	CMP4	—	CMP4	—
15	CMP5	—	CMP5	—
16	—	—	—	—
17	—	—	—	—
18	TMR0_CMA	—	TMR0_CMA	—
19	TMR0_CMB	—	TMR0_CMB	—
20	TMR1_CMA	—	TMR1_CMA	—
21	TMR1_CMB	—	TMR1_CMB	—
22	TMR2_CMA	—	TMR2_CMA	—
23	TMR2_CMB	—	TMR2_CMB	—
24	TMR3_CMA	—	TMR3_CMA	—
25	TMR3_CMB	—	TMR3_CMB	—
26	PWM0_0	—	PWM0_0	—
27	PWM0_1	—	PWM0_1	—
28	PWM1_0	—	PWM1_0	—
29	PWM1_1	—	PWM1_1	—
30	PWM2_0	—	PWM2_0	—
31	PWM2_1	—	PWM2_1	—
32	PWM3_0	—	PWM3_0	—
33	PWM3_1	—	PWM3_1	—
34	EPU0_2/EPU1_2	EVSEL5.AD00SEL	EPU0_2/EPU1_2	EVSEL7.AD10SEL
35	EPU0_3/EPU1_3		EPU0_3/EPU1_3	
36	EPU0_4/EPU1_4	EVSEL5.AD01SEL	EPU0_4/EPU1_4	EVSEL7.AD11SEL
37	EPU0_5/EPU1_5		EPU0_5/EPU1_5	
38	EPU0_6/EPU1_6	EVSEL5.AD02SEL	EPU0_6/EPU1_6	EVSEL7.AD12SEL
39	EPU0_7/EPU1_7		EPU0_7/EPU1_7	
40	EPU0_8/EPU1_8	EVSEL5.AD03SEL	EPU0_8/EPU1_8	EVSEL7.AD13SEL
41	EPU0_9/EPU1_9		EPU0_9/EPU1_9	
42	EPU2_2/EPU3_2	EVSEL5.AD04SEL	EPU2_2/EPU3_2	EVSEL7.AD14SEL
43	EPU2_3/EPU3_3		EPU2_3/EPU3_3	
44	EPU2_4/EPU3_4	EVSEL5.AD05SEL	EPU2_4/EPU3_4	EVSEL7.AD15SEL
45	EPU2_5/EPU3_5		EPU2_5/EPU3_5	
46	EPU2_6/EPU3_6	EVSEL5.AD06SEL	EPU2_6/EPU3_6	EVSEL7.AD16SEL
47	EPU2_7/EPU3_7		EPU2_7/EPU3_7	
48	EPU2_8/EPU3_8	EVSEL5.AD07SEL	EPU2_8/EPU3_8	EVSEL7.AD17SEL
49	EPU2_9/EPU3_9		EPU2_9/EPU3_9	
50	EPU4_2/EPU5_2	EVSEL6.AD08SEL	EPU4_2/EPU5_2	EVSEL8.AD18SEL
51	EPU4_3/EPU5_3		EPU4_3/EPU5_3	
52	EPU4_4/EPU5_4	EVSEL6.AD09SEL	EPU4_4/EPU5_4	EVSEL8.AD19SEL
53	EPU4_5/EPU5_5		EPU4_5/EPU5_5	
54	EPU4_6/EPU5_6	EVSEL6.AD0ASEL	EPU4_6/EPU5_6	EVSEL8.AD1ASEL
55	EPU4_7/EPU5_7		EPU4_7/EPU5_7	
56	EPU4_8/EPU5_8	EVSEL6.AD0BSEL	EPU4_8/EPU5_8	EVSEL8.AD1BSEL
57	EPU4_9/EPU5_9		EPU4_9/EPU5_9	

Table 9-4. DSAC Event Input

Event No.	DSAC Channel0 to Channel7		DSAC Channel8 to Channel15	
	Event Type	Setting Register	Event Type	Setting Register
0	CMP0/CMP1	EVSEL9.DSAC0SEL	CMP0/CMP1	EVSEL9.DSAC1SEL
1	CMP2/CMP3		CMP2/CMP3	
2	CMP4/CMP5		CMP4/CMP5	
3	ADC0_0	—	ADC0_0	—
4	ADC0_1	—	ADC0_1	—
5	ADC0_2	—	ADC0_2	—
6	ADC0_3	—	ADC0_3	—
7	ADC0_4	—	ADC0_4	—
8	ADC0_5	—	ADC0_5	—
9	ADC0_6	—	ADC0_6	—
10	ADC0_7	—	ADC0_7	—
11	ADC1_0	—	ADC1_0	—
12	ADC1_1	—	ADC1_1	—
13	ADC1_2	—	ADC1_2	—
14	ADC1_3	—	ADC1_3	—
15	ADC1_4	—	ADC1_4	—
16	ADC1_5	—	ADC1_5	—
17	ADC1_6	—	ADC1_6	—
18	ADC1_7	—	ADC1_7	—
19	PWM0_0/PWM0_1	EVSEL10.DSAC0SEL	PWM0_0/PWM0_1	EVSEL10.DSAC1SEL
20	PWM1_0/PWM1_1		PWM1_0/PWM1_1	
21	PWM2_0/PWM2_1		PWM2_0/PWM2_1	
22	PWM3_0/PWM3_1		PWM3_0/PWM3_1	
23	TinyDSP0_0	—	TinyDSP0_0	—
24	TinyDSP0_1	—	TinyDSP0_1	—
25	TinyDSP1_0	—	TinyDSP1_0	—
26	TinyDSP1_1	—	TinyDSP1_1	—
27	TMR0 CMA/CMB	EVSEL11.DSAC0SEL	TMR0 CMA/CMB	EVSEL11.DSAC1SEL
28	TMR1 CMA/CMB		TMR1 CMA/CMB	
29	TMR2 CMA/CMB		TMR2 CMA/CMB	
30	TMR3 CMA/CMB		TMR3 CMA/CMB	
31	(CPU)	—	(CPU)	—

9.2.3. Event Integration

In this block, the events of EPU are selected and are integrated by an OR circuit. The integrated signal is output to each channel of the POC. There are 4 integrated groups, A, B, C, and D. An event for the Channel0, Channel1, Channel2, or Channel3 of the POC is generated in the Group A, Group B, Group C, or Group D, respectively.

When the MGENx bit of the EVMGxC0/1 register (where, x is A, B, C, or D) is 1, the corresponding event is output to the POC. When the MGENx bit is 0, the corresponding bit is not output to the POC.

Table 9-5. POC Event Input

Event No.	POC Channel0		POC Channel1		POC Channel2		POC Channel3	
	Event Type	Setting Register	Event Type	Setting Register	Event Type	Setting Register	Event Type	Setting Register
0	CMP0 (level)	—	CMP0 (level)	—	CMP0 (level)	—	CMP0 (level)	—
1	CMP1 (level)	—	CMP1 (level)	—	CMP1 (level)	—	CMP1 (level)	—
2	CMP2 (level)	—	CMP2 (level)	—	CMP2 (level)	—	CMP2 (level)	—
3	CMP3 (level)	—	CMP3 (level)	—	CMP3 (level)	—	CMP3 (level)	—
4	CMP4 (level)	—	CMP4 (level)	—	CMP4 (level)	—	CMP4 (level)	—
5	CMP5 (level)	—	CMP5 (level)	—	CMP5 (level)	—	CMP5 (level)	—
6	LUT0 (level)	—	LUT0 (level)	—	LUT0 (level)	—	LUT0 (level)	—
7	LUT1 (level)	—	LUT1 (level)	—	LUT1 (level)	—	LUT1 (level)	—
EPU	EPU0_10	EVMGAC0.MGEN0	EPU0_10	EVMGBC0.MGEN0	EPU0_10	EVMGCC0.MGEN0	EPU0_10	EVMGDC0.MGEN0
	EPU0_11	EVMGAC0.MGEN1	EPU0_11	EVMGBC0.MGEN1	EPU0_11	EVMGCC0.MGEN1	EPU0_11	EVMGDC0.MGEN1
	EPU1_10	EVMGAC0.MGEN2	EPU1_10	EVMGBC0.MGEN2	EPU1_10	EVMGCC0.MGEN2	EPU1_10	EVMGDC0.MGEN2
	EPU1_11	EVMGAC0.MGEN3	EPU1_11	EVMGBC0.MGEN3	EPU1_11	EVMGCC0.MGEN3	EPU1_11	EVMGDC0.MGEN3
	EPU2_10	EVMGAC0.MGEN4	EPU2_10	EVMGBC0.MGEN4	EPU2_10	EVMGCC0.MGEN4	EPU2_10	EVMGDC0.MGEN4
	EPU2_11	EVMGAC0.MGEN5	EPU2_11	EVMGBC0.MGEN5	EPU2_11	EVMGCC0.MGEN5	EPU2_11	EVMGDC0.MGEN5
	EPU3_10	EVMGAC0.MGEN6	EPU3_10	EVMGBC0.MGEN6	EPU3_10	EVMGCC0.MGEN6	EPU3_10	EVMGDC0.MGEN6
	EPU3_11	EVMGAC0.MGEN7	EPU3_11	EVMGBC0.MGEN7	EPU3_11	EVMGCC0.MGEN7	EPU3_11	EVMGDC0.MGEN7
	EPU4_10	EVMGAC1.MGEN8	EPU4_10	EVMGBC1.MGEN8	EPU4_10	EVMGCC1.MGEN8	EPU4_10	EVMGDC1.MGEN8
	EPU4_11	EVMGAC1.MGEN9	EPU4_11	EVMGBC1.MGEN9	EPU4_11	EVMGCC1.MGEN9	EPU4_11	EVMGDC1.MGEN9
	EPU5_10	EVMGAC1.MGENA	EPU5_10	EVMGBC1.MGENA	EPU5_10	EVMGCC1.MGENA	EPU5_10	EVMGDC1.MGENA
	EPU5_11	EVMGAC1.MGENB	EPU5_11	EVMGBC1.MGENB	EPU5_11	EVMGCC1.MGENB	EPU5_11	EVMGDC1.MGENB

9.2.4. GPIO Event Edge Detection

The GPIO event edge for the ADC is detected. When the GPIO event is selected in the ADC, it is required to enable the clock of the EVC module.

9.3. Register Descriptions

Table 9-6. List of Registers

Symbol	Name	Address	Initial Value
EVINTE0	EVC Interrupt Enable0	0xE300	0x00
EVINTF0	EVC Interrupt Flag0	0xE308	0x00
EVMGAC0	EVC Event Merge A Configuration0	0xE310	0x00
EVMGAC1	EVC Event Merge A Configuration1	0xE311	0x00
EVMGBC0	EVC Event Merge B Configuration0	0xE312	0x00
EVMGBC1	EVC Event Merge B Configuration1	0xE313	0x00
EVMGCC0	EVC Event Merge C Configuration0	0xE314	0x00
EVMGCC1	EVC Event Merge C Configuration1	0xE315	0x00
EVMGDC0	EVC Event Merge D Configuration0	0xE316	0x00
EVMGDC1	EVC Event Merge D Configuration1	0xE317	0x00
EVSEL0	EVC Select0	0xE330	0x00
EVSEL1	EVC Select1	0xE331	0x00
EVSEL2	EVC Select2	0xE332	0x00
EVSEL3	EVC Select3	0xE333	0x00
EVSEL4	EVC Select4	0xE334	0x00
EVSEL5	EVC Select5	0xE335	0x00
EVSEL6	EVC Select6	0xE336	0x00
EVSEL7	EVC Select7	0xE337	0x00
EVSEL8	EVC Select8	0xE338	0x00
EVSEL9	EVC Select9	0xE339	0x00
EVSEL10	EVC Select10	0xE33A	0x00
EVSEL11	EVC Select11	0xE33B	0x00

9.3.1. EVINTE0 (EVC Interrupt Enable0)

Register	EVINTE0	EVC Interrupt Enable0	Address	0xE300

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	EPU5IE	R/W	0	EPU Thread5 event interrupt enable 0: EPU Thread5 event interrupt is disabled 1: EPU Thread5 event interrupt is enabled If the bit is set to 1, an interrupt is notified to the CPU when EPU5IF = 1.	
4	EPU4IE	R/W	0	EPU Thread4 event interrupt enable 0: EPU Thread4 event interrupt is disabled 1: EPU Thread4 event interrupt is enabled If the bit is set to 1, an interrupt is notified to the CPU when EPU4IF = 1.	
3	EPU3IE	R/W	0	EPU Thread3 event interrupt enable 0: EPU Thread3 event interrupt is disabled 1: EPU Thread3 event interrupt is enabled If the bit is set to 1, an interrupt is notified to the CPU when EPU3IF = 1.	
2	EPU2IE	R/W	0	EPU Thread2 event interrupt enable 0: EPU Thread2 event interrupt is disabled 1: EPU Thread2 event interrupt is enabled If the bit is set to 1, an interrupt is notified to the CPU when EPU2IF = 1.	
1	EPU1IE	R/W	0	EPU Thread1 event interrupt enable 0: EPU Thread1 event interrupt is disabled 1: EPU Thread1 event interrupt is enabled If the bit is set to 1, an interrupt is notified to the CPU when EPU1IF = 1.	
0	EPU0IE	R/W	0	EPU Thread0 event interrupt enable 0: EPU Thread0 event interrupt is disabled 1: EPU Thread0 event interrupt is enabled If the bit is set to 1, an interrupt is notified to the CPU when EPU0IF = 1.	

9.3.2. EVINTF0 (EVC Interrupt Flag0)

Register	EVINTF0	EVC Interrupt Flag0	Address	0xE308

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	EPU5IF	R/C	0	EPU Thread5 event interrupt flag Read 0: No interrupt source generated Read 1: An interrupt source generated Write 0: No change Write 1: The bit is cleared The bit is set when the EPU Thread5 Event14 is detected. When EPU5IF = 1 and EPU5IE = 1, an interrupt is notified to the CPU. When 1 is written to the bit, the bit is cleared to 0. In the case where clearing the bit to 0 occurred simultaneously with an EPU5 event setting, the clearing operation has higher priority; therefore, the setting operation is ignored.	
4	EPU4IF	R/C	0	EPU Thread4 event interrupt flag Read 0: No interrupt source generated Read 1: An interrupt source generated Write 0: No change Write 1: The bit is cleared The bit is set when the EPU Thread4 Event14 is detected. When EPU4IF = 1 and EPU4IE = 1, an interrupt is notified to the CPU. When 1 is written to the bit, the bit is cleared to 0. In the case where clearing the bit to 0 occurred simultaneously with an EPU4 event setting, the clearing operation has higher priority; therefore, the setting operation is ignored.	
3	EPU3IF	R/C	0	EPU Thread3 event interrupt flag Read 0: No interrupt source generated Read 1: An interrupt source generated Write 0: No change Write 1: The bit is cleared The bit is set when the EPU Thread3 Event14 is detected. When EPU3IF = 1 and EPU3IE = 1, an interrupt is notified to the CPU. When 1 is written to the bit, the bit is cleared to 0. In the case where clearing the bit to 0 occurred simultaneously with an EPU3 event setting, the clearing operation has higher priority; therefore, the setting operation is ignored.	

Register	EVINTF0	EVC Interrupt Flag0	Address	0xE308

Bit	Bit Name	R/W	Initial	Description	Remarks
2	EPU2IF	R/C	0	EPU Thread2 event interrupt flag Read 0: No interrupt source generated Read 1: An interrupt source generated Write 0: No change Write 1: The bit is cleared The bit is set when the EPU Thread2 Event14 is detected. When EPU2IF = 1 and EPU2IE = 1, an interrupt is notified to the CPU. When 1 is written to the bit, the bit is cleared to 0. In the case where clearing the bit to 0 occurred simultaneously with an EPU2 event setting, the clearing operation has higher priority; therefore, the setting operation is ignored.	
1	EPU1IF	R/C	0	EPU Thread1 event interrupt flag Read 0: No interrupt source generated Read 1: An interrupt source generated Write 0: No change Write 1: The bit is cleared The bit is set when the EPU Thread1 Event14 is detected. When EPU1IF = 1 and EPU1IE = 1, an interrupt is notified to the CPU. When 1 is written to the bit, the bit is cleared to 0. In the case where clearing the bit to 0 occurred simultaneously with an EPU1 event setting, the clearing operation has higher priority; therefore, the setting operation is ignored.	
0	EPU0IF	R/C	0	EPU Thread0 event interrupt flag Read 0: No interrupt source generated Read 1: An interrupt source generated Write 0: No change Write 1: The bit is cleared The bit is set when the EPU Thread0 Event14 is detected. When EPU0IF = 1 and EPU0IE = 1, an interrupt is notified to the CPU. When 1 is written to the bit, the bit is cleared to 0. In the case where clearing the bit to 0 occurred simultaneously with an EPU0 event setting, the clearing operation has higher priority; therefore, the setting operation is ignored.	

9.3.3. EVMGAC0 (EVC Event Merge A Configuration0)

Register	EVMGAC0	EVC Event Merge A Configuration0	Address	0xE310

Bit	Bit Name	R/W	Initial	Description	Remarks
7	MGEN7	R/W	0	Integration of EPU Thread3 Event11 into POC0 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC0.	
6	MGEN6	R/W	0	Integration of EPU Thread3 Event10 into POC0 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC0.	
5	MGEN5	R/W	0	Integration of EPU Thread2 Event11 into POC0 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC0.	
4	MGEN4	R/W	0	Integration of EPU Thread2 Event10 into POC0 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC0.	
3	MGEN3	R/W	0	Integration of EPU Thread1 Event11 into POC0 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC0.	
2	MGEN2	R/W	0	Integration of EPU Thread1 Event10 into POC0 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC0.	
1	MGEN1	R/W	0	Integration of EPU Thread0 Event11 into POC0 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC0.	
0	MGEN0	R/W	0	Integration of EPU Thread0 Event10 into POC0 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC0.	

9.3.4. EVMGAC1 (EVC Event Merge A Configuration1)

Register	EVMGAC1	EVC Event Merge A Configuration1	Address	0xE311

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0..	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	MGENB	R/W	0	Integration of EPU Thread5 Event11 into POC0 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC0.	
2	MGENA	R/W	0	Integration of EPU Thread5 Event10 into POC0 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC0.	
1	MGEN9	R/W	0	Integration of EPU Thread4 Event11 into POC0 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC0.	
0	MGEN8	R/W	0	Integration of EPU Thread4 Event10 into POC0 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC0.	

9.3.5. EVMGBC0 (EVC Event Merge B Configuration0)

Register	EVMGBC0	EVC Event Merge B Configuration0	Address	0xE312

Bit	Bit Name	R/W	Initial	Description	Remarks
7	MGEN7	R/W	0	Integration of EPU Thread3 Event11 into POC1 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC1.	
6	MGEN6	R/W	0	Integration of EPU Thread3 Event10 into POC1 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC1.	
5	MGEN5	R/W	0	Integration of EPU Thread2 Event11 into POC1 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC1.	
4	MGEN4	R/W	0	Integration of EPU Thread2 Event10 into POC1 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC1.	
3	MGEN3	R/W	0	Integration of EPU Thread1 Event11 into POC1 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC1.	
2	MGEN2	R/W	0	Integration of EPU Thread1 Event10 into POC1 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC1.	
1	MGEN1	R/W	0	Integration of EPU Thread0 Event11 into POC1 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC1.	
0	MGEN0	R/W	0	Integration of EPU Thread0 Event10 into POC1 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC1.	

9.3.6. EVMGBC1 (EVC Event Merge B Configuration1)

Register	EVMGBC1	EVC Event Merge B Configuration1	Address	0xE313

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	MGENB	R/W	0	Integration of EPU Thread5 Event11 into POC1 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC1.	
2	MGENA	R/W	0	Integration of EPU Thread5 Event10 into POC1 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC1.	
1	MGEN9	R/W	0	Integration of EPU Thread4 Event11 into POC1 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC1.	
0	MGEN8	R/W	0	Integration of EPU Thread4 Event10 into POC1 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC1.	

9.3.7. EVMGCC0 (EVC Event Merge C Configuration0)

Register	EVMGCC0	EVC Event Merge C Configuration0	Address	0xE314

Bit	Bit Name	R/W	Initial	Description	Remarks
7	MGEN7	R/W	0	Integration of EPU Thread3 Event11 into POC2 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC2.	
6	MGEN6	R/W	0	Integration of EPU Thread3 Event10 into POC2 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC2.	
5	MGEN5	R/W	0	Integration of EPU Thread2 Event11 into POC2 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC2.	
4	MGEN4	R/W	0	Integration of EPU Thread2 Event10 into POC2 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC2.	
3	MGEN3	R/W	0	Integration of EPU Thread1 Event11 into POC2 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC2.	
2	MGEN2	R/W	0	Integration of EPU Thread1 Event10 into POC2 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC2.	
1	MGEN1	R/W	0	Integration of EPU Thread0 Event11 into POC2 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC2.	
0	MGEN0	R/W	0	Integration of EPU Thread0 Event10 into POC2 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC2.	

9.3.8. EVMGCC1 (EVC Event Merge C Configuration1)

Register	EVMGCC1	EVC Event Merge C Configuration1	Address	0xE315

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	MGENB	R/W	0	Integration of EPU Thread5 Event11 into POC2 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC2.	
2	MGENA	R/W	0	Integration of EPU Thread5 Event10 into POC2 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC2.	
1	MGEN9	R/W	0	Integration of EPU Thread4 Event11 into POC2 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC2.	
0	MGEN8	R/W	0	Integration of EPU Thread4 Event10 into POC2 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC2.	

9.3.9. EVMGDC0 (EVC Event Merge D Configuration0)

Register	EVMGDC0	EVC Event Merge D Configuration0	Address	0xE316

Bit	Bit Name	R/W	Initial	Description	Remarks
7	MGEN7	R/W	0	Integration of EPU Thread3 Event11 into POC3 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC3.	
6	MGEN6	R/W	0	Integration of EPU Thread3 Event10 into POC3 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC3.	
5	MGEN5	R/W	0	Integration of EPU Thread2 Event11 into POC3 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC3.	
4	MGEN4	R/W	0	Integration of EPU Thread2 Event10 into POC3 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC3.	
3	MGEN3	R/W	0	Integration of EPU Thread1 Event11 into POC3 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC3.	
2	MGEN2	R/W	0	Integration of EPU Thread1 Event10 into POC3 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC3.	
1	MGEN1	R/W	0	Integration of EPU Thread0 Event11 into POC3 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC3.	
0	MGEN0	R/W	0	Integration of EPU Thread0 Event10 into POC3 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC3.	

9.3.10. EVMGDC1 (EVC Event Merge D Configuration1)

Register	EVMGDC1	EVC Event Merge D Configuration1	Address	0xE317

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	MGENB	R/W	0	Integration of EPU Thread5 Event11 into POC3 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC3.	
2	MGENA	R/W	0	Integration of EPU Thread5 Event10 into POC3 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC3.	
1	MGEN9	R/W	0	Integration of EPU Thread4 Event11 into POC3 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC3.	
0	MGEN8	R/W	0	Integration of EPU Thread4 Event10 into POC3 event 0: Event is not integrated (OR) 1: Event is integrated (OR) If the bit is set to 1, the event is added to the EPU event of POC3.	

9.3.11. EVSEL0 (EVC Select0)

Register	EVSEL0	EVC Select0	Address	0xE330

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	EPU5SEL	R/W	0	TinyDSP event selection to EPU Thread5 0: TinyDSP0/1, Event0 1: TinyDSP0/1, Event1	
4	EPU4SEL	R/W	0	TinyDSP event selection to EPU Thread4 0: TinyDSP0/1, Event0 1: TinyDSP0/1, Event1	
3	EPU3SEL	R/W	0	TinyDSP event selection to EPU Thread3 0: TinyDSP0/1, Event0 1: TinyDSP0/1, Event1	
2	EPU2SEL	R/W	0	TinyDSP event selection to EPU Thread2 0: TinyDSP0/1, Event0 1: TinyDSP0/1, Event1	
1	EPU1SEL	R/W	0	TinyDSP event selection to EPU Thread1 0: TinyDSP0/1, Event0 1: TinyDSP0/1, Event1	
0	EPU0SEL	R/W	0	TinyDSP event selection to EPU Thread0 0: TinyDSP0/1, Event0 1: TinyDSP0/1, Event1	

9.3.12. EVSEL1 (EVC Select1)

Register	EVSEL1	EVC Select1	Address	0xE331

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	EPU5SEL	R/W	0	ADC0/1 event selection to EPU Thread5 0: Even numbers of ADC0/1 group 1: Odd numbers of ADC0/1 group	
4	EPU4SEL	R/W	0	ADC0/1 event selection to EPU Thread4 0: Even numbers of ADC0/1 group 1: Odd numbers of ADC0/1 group	
3	EPU3SEL	R/W	0	ADC0/1 event selection to EPU Thread3 0: Even numbers of ADC0/1 group 1: Odd numbers of ADC0/1 group	
2	EPU2SEL	R/W	0	ADC0/1 event selection to EPU Thread2 0: Even numbers of ADC0/1 group 1: Odd numbers of ADC0/1 group	
1	EPU1SEL	R/W	0	ADC0/1 event selection to EPU Thread1 0: Even numbers of ADC0/1 group 1: Odd numbers of ADC0/1 group	
0	EPU0SEL	R/W	0	ADC0/1 event selection to EPU Thread0 0: Even numbers of ADC0/1 group 1: Odd numbers of ADC0/1 group	

9.3.13. EVSEL2 (EVC Select2)

Register	EVSEL2	EVC Select2	Address	0xE332

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	EPU5SEL	R/W	0	PWM event selection to EPU Thread5 0: PWM0/1/2/3, Event0 1: PWM0/1/2/3, Event1	
4	EPU4SEL	R/W	0	PWM event selection to EPU Thread4 0: PWM0/1/2/3, Event0 1: PWM0/1/2/3, Event1	
3	EPU3SEL	R/W	0	PWM event selection to EPU Thread3 0: PWM0/1/2/3, Event0 1: PWM0/1/2/3, Event1	
2	EPU2SEL	R/W	0	PWM event selection to EPU Thread2 0: PWM0/1/2/3, Event0 1: PWM0/1/2/3, Event1	
1	EPU1SEL	R/W	0	PWM event selection to EPU Thread1 0: PWM0/1/2/3, Event0 1: PWM0/1/2/3, Event1	
0	EPU0SEL	R/W	0	PWM event selection to EPU Thread0 0: PWM0/1/2/3, Event0 1: PWM0/1/2/3, Event1	

9.3.14. EVSEL3 (EVC Select3)

Register	EVSEL3	EVC Select3	Address	0xE333

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	EPU5SEL	R/W	0	Serial module event selection to EPU Thread5 0: UART 1: SCID	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	EPU2SEL	R/W	0	LUT event selection to EPU Thread2 0: LUT0 1: LUT1	
1	EPU1SEL	R/W	0	LUT event selection to EPU Thread1 0: LUT0 1: LUT1	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

9.3.15. EVSEL4 (EVC Select4)

Register	EVSEL4	EVC Select4	Address	0xE334

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	EPU4SEL	R/W	0	TMR3 event selection to EPU Thread4 0: CMA of TMR3 1: CMB of TMR3	
3	EPU3SEL	R/W	0	TMR2 event selection to EPU Thread3 0: CMA of TMR2 1: CMB of TMR2	
2	EPU2SEL	R/W	0	TMR1 event selection to EPU Thread2 0: CMA of TMR1 1: CMB of TMR1	
1	EPU1SEL	R/W	0	TMR0 event selection to EPU Thread1 0: CMA of TMR0 1: CMB of TMR0	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

9.3.16. EVSEL5 (EVC Select5)

Register	EVSEL5	EVC Select5	Address	0xE335

Bit	Bit Name	R/W	Initial	Description	Remarks
7	AD07SEL	R/W	0	EPU Event8/9 thread selection to ADC0 0: EPU Thread2, Event8/9 1: EPU Thread3, Event8/9	
6	AD06SEL	R/W	0	EPU Event6/7 thread selection to ADC0 0: EPU Thread2, Event6/7 1: EPU Thread3, Event6/7	
5	AD05SEL	R/W	0	EPU Event4/5 thread selection to ADC0 0: EPU Thread2, Event4/5 1: EPU Thread3, Event4/5	
4	AD04SEL	R/W	0	EPU Event2/3 thread selection to ADC0 0: EPU Thread2, Event2/3 1: EPU Thread3, Event2/3	
3	AD03SEL	R/W	0	EPU Event8/9 thread selection to ADC0 0: EPU Thread0, Event8/9 1: EPU Thread1, Event8/9	
2	AD02SEL	R/W	0	EPU Event6/7 thread selection to ADC0 0: EPU Thread0, Event6/7 1: EPU Thread1, Event6/7	
1	AD01SEL	R/W	0	EPU Event4/5 thread selection to ADC0 0: EPU Thread0, Event4/5 1: EPU Thread1, Event4/5	
0	AD00SEL	R/W	0	EPU Event2/3 thread selection to ADC0 0: EPU Thread0, Event2/3 1: EPU Thread1, Event2/3	

9.3.17. EVSEL6 (EVC Select6)

Register	EVSEL6	EVC Select6	Address	0xE336

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	AD0BSEL	R/W	0	EPU Event8/9 thread selection to ADC0 0: EPU Thread4, Event8/9 1: EPU Thread5, Event8/9	
2	AD0ASEL	R/W	0	EPU Event6/7 thread selection to ADC0 0: EPU Thread4, Event6/7 1: EPU Thread5, Event6/7	
1	AD09SEL	R/W	0	EPU Event4/5 thread selection to ADC0 0: EPU Thread4, Event4/5 1: EPU Thread5, Event4/5	
0	AD08SEL	R/W	0	EPU Event2/3 thread selection to ADC0 0: EPU Thread4, Event2/3 1: EPU Thread5, Event2/3	

9.3.18. EVSEL7 (EVC Select7)

Register	EVSEL7	EVC Select7	Address	0xE337

Bit	Bit Name	R/W	Initial	Description	Remarks
7	AD17SEL	R/W	0	EPU Event8/9 thread selection to ADC1 0: EPU Thread2, Event8/9 1: EPU Thread3, Event8/9	
6	AD16SEL	R/W	0	EPU Event6/7 thread selection to ADC1 0: EPU Thread2, Event6/7 1: EPU Thread3, Event6/7	
5	AD15SEL	R/W	0	EPU Event4/5 thread selection to ADC1 0: EPU Thread2, Event4/5 1: EPU Thread3, Event4/5	
4	AD14SEL	R/W	0	EPU Event2/3 thread selection to ADC1 0: EPU Thread2, Event2/3 1: EPU Thread3, Event2/3	
3	AD13SEL	R/W	0	EPU Event8/9 thread selection to ADC1 0: EPU Thread0, Event8/9 1: EPU Thread1, Event8/9	
2	AD12SEL	R/W	0	EPU Event6/7 thread selection to ADC1 0: EPU Thread0, Event6/7 1: EPU Thread1, Event6/7	
1	AD11SEL	R/W	0	EPU Event4/5 thread selection to ADC1 0: EPU Thread0, Event4/5 1: EPU Thread1, Event4/5	
0	AD10SEL	R/W	0	EPU Event2/3 thread selection to ADC1 0: EPU Thread0, Event2/3 1: EPU Thread1, Event2/3	

9.3.19. EVSEL8 (EVC Select8)

Register	EVSEL8	EVC Select8	Address	0xE338

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	A10BSEL	R/W	0	EPU Event8/9 thread selection to ADC1 0: EPU Thread4, Event8/9 1: EPU Thread5, Event8/9	
2	AD1ASEL	R/W	0	EPU Event6/7 thread selection to ADC1 0: EPU Thread4, Event6/7 1: EPU Thread5, Event6/7	
1	AD19SEL	R/W	0	EPU Event4/5 thread selection to ADC1 0: EPU Thread4, Event4/5 1: EPU Thread5, Event4/5	
0	AD18SEL	R/W	0	EPU Event2/3 thread selection to ADC1 0: EPU Thread4, Event2/3 1: EPU Thread5, Event2/3	

9.3.20. EVSEL9 (EVC Select9)

Register	EVSEL9		EVC Select9	Address	0xE339
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	DSAC1SEL	R/W	0	CMP event selection to DSAC Channel8 to Channel15 0: CMP Thread0/2/4 1: CMP Thread1/3/5	
0	DSAC0SEL	R/W	0	CMP event selection to DSAC Channel0 to Channel7 0: CMP Thread0/2/4 1: CMP Thread1/3/5	

9.3.21. EVSEL10 (EVC Select10)

Register	EVSEL10		EVC Select10	Address	0xE33A
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	DSAC1SEL	R/W	0	PWM event selection to DSAC Channel8 to Channel15 0: PWM0/1/2/3, Event0 1: PWM0/1/2/3, Event1	
0	DSAC0SEL	R/W	0	PWM event selection to DSAC Channel0 to Channel7 0: PWM0/1/2/3, Event0 1: PWM0/1/2/3, Event1	

9.3.22. EVSEL11 (EVC Select11)

Register	EVSEL11	EVC Select11	Address	0xE33B

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	DSAC1SEL	R/W	0	TMR event selection to DSAC Channel8 to Channel15 0: TMR0/1/2/3, CMA event 1: TMR0/1/2/3, CMB event	
0	DSAC0SEL	R/W	0	TMR event selection to DSAC Channel0 to Channel7 0: TMR0/1/2/3, CMA event 1: TMR0/1/2/3, CMB event	

10. Event Processing Unit (EPU)

10.1. Overview

The event processing unit (EPU) is an operation unit that performs the data transfer, data processing, and event processing instead of the CPU. The EPU can process up to 6 threads (channels) in parallel in one operation unit. Since the thread is activated in one cycle at least from the event input, the EPU is suitable for the processing required the high speed response. Each thread has a context such as a program counter, a register, and a thread activation timer.

Since the high speed context switching is simple, the switching between threads is performed in zero cycle. The EPU can process at high speed in one throughput cycle with switching threads. The thread generates an activation request by an external event. The activation request of each thread is selected according to the priority, and processes the selected thread processing by the operation unit. User can program each thread processing. The program is allocated in the shared memory with all threads. The EPU operates for the bus masters of the MBUS that the program memory is connected, SBUS, and XBUS. The EPU transfers the data between these buses. Also, the EPU generates its own event. Using this function, the events that skipped an input event and the computed multiple input events can be generated.

Table 10-1. EPU Functional Descriptions

Item	Description
Number of Threads	6 threads
Resource (Each Thread)	Program counter: 9 bits General-purpose register: 16 bits × 2 (R0 or R1) Dedicated timer counter: 12 bits × 1 Prescaler: 8 bits × 1 Flag register: T (true), C (carry)
Event	16 inputs per thread Dedicated timer in the thread 16 outputs per thread
Program Memory	16 bits × 256 words (connected to the MBUS)
Instruction	16-bit fixed-length code, instruction set based on RISC architecture Arithmetic operation: ADD, SUB, MUL, signed/unsigned comparison Logic operation: AND, OR, NOT, XOR Load/Store: Register−register, SBUS−register, XBUS−register, MBUS−register Branch: Conditional/unconditional absolute address branch Event: Input/wait/output/dedicated timer event wait
Execution Unit	2-stage or 3-stage pipeline
MBUS	Common program/data memory for all threads Address space: 256 words Data width: 16 bits per word Dedicated bus Access mode: Byte access mode
SBUS	Address space: 256 words Data width: 16 bits per word Access mode: Word/byte access mode Access cycle: One cycle (min.), cycle extension with the wait control Priority: DSAC > EPU > CPU
XBUS	Address space: 64 Kwords Data width: 8 bits per word Access cycle: One cycle (min.), cycle extension with the wait control Priority: EPU > CPU

10.2. Block Diagram

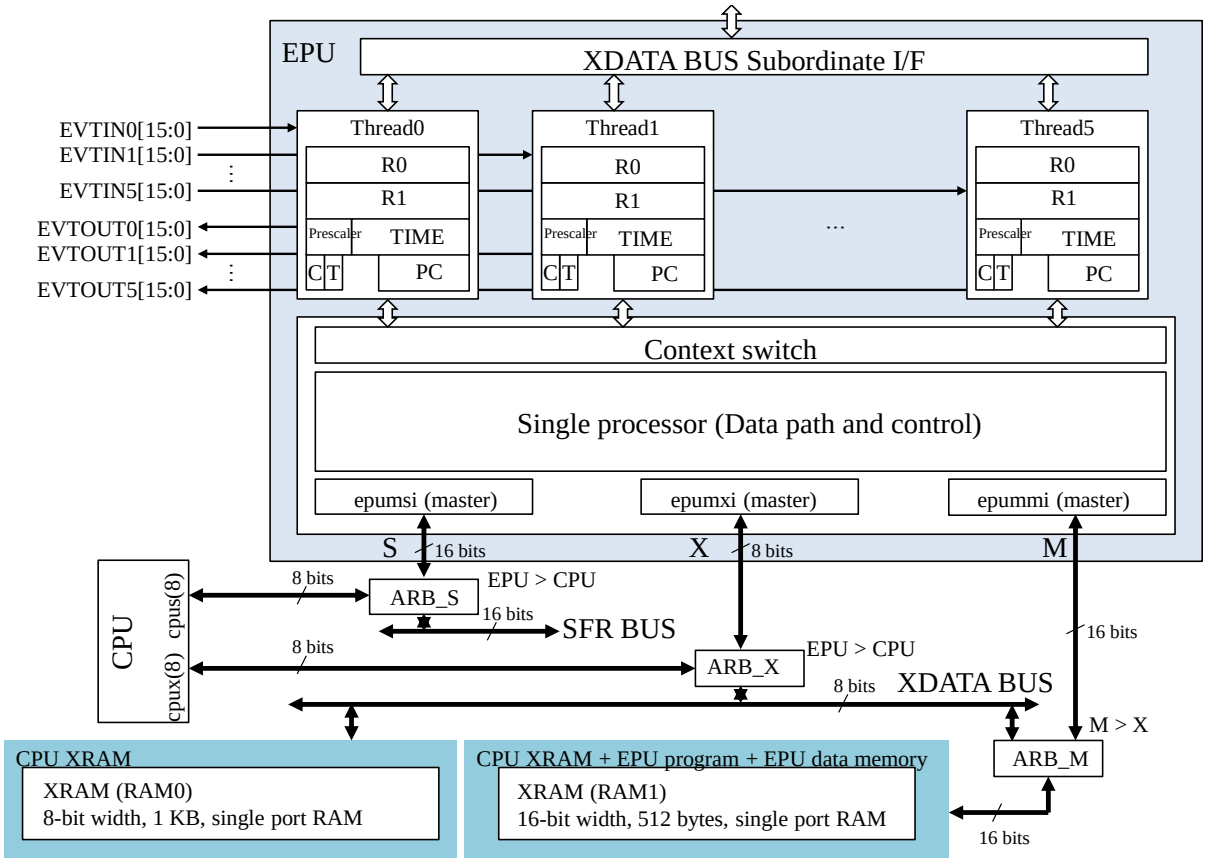


Figure 10-1. EPU Block Diagram

10.3. Common Resource

(1) Program Memory
This is a 256-word (max.) memory that allocates programs and data. The data width is 16 bits per word. The program memory is accessed by the EPU instruction fetch and the LOADM or STOREM instrcution.

(2) EPU Core
This is an EPU instrcution execution core that has 2-stage or 3-stage pipeline structure. The instruction is executed in one instruction (max.) per cycle.

(3) MBUS
This is a memory interface that allocates the program and data of the EPU. The bus width is 16 bits. The MBUS has 8 bits per word addressing. The program fetch is performed in 16-bit unit. The program must be allocated in 2-byte alignment. The LOADM or STOREM instruction can access the data in 16-bit or 8-bit unit. The memory is shared with the CPU XRAM. The access priority is MBUS > XBUS.

(4) SBUS
This is a high speed bus that can be accessed in one cycle (min.). The bus width is 16 bits. The SBUS is accessed by the CPU and the EPU. The priority is DSAC > EPU > CPU. The CPU can access with only 8-bit. The EPU can access with 8-bit or 16-bit, which can be specified. The SBUS has 16 bits per word addressing.

(5) XBUS
This is a bus that can be accessed in one cycle (min.). The bus width is 8 bits. The XBUS is accessed by the CPU and the EPU. The priority is EPU > CPU. The XBUS has 8 bits per word addressing.

10.4. Resource of Each Thread

● **General-purpose Registers (R0 and R1)**

These are 16-bit width general-purpose registers, and are used for storing the operands of arithmetic logic operation and the operation results. In the XBUS access instruction, these registers are used for the base address registers of the XBUS access address. These registers can be accessed as the XBUS registers from the CPU. Therefore, the initial value before the each EPU thread execution can be set from the CPU.

● **Program Counter (PC)**

This indicates the present program address, and is incremented by 2 per one instruction execution. The PC can be accessed as the XBUS register from the CPU. The PC initial value of each thread must be set from the CPU.

● **Dedicated Timer Counter (TIME)**

This is 12-bit timer counter. When the TIMWAIT instruction is executed, the executed thread cancels the instruction execution request; furthermore, the subsequent instruction execution is suspended. The value specified by the TIMWAIT instruction is loaded to the TIME. The TIME counts down until the value becomes zero. When TIME = 0, this thread outputs the execution request of the subsequent instruction. The TIME counts down according to the count signal obtained by dividing the system clock by the prescaler. The TIME can be accessed as the XBUS register from the CPU.

● **T Flag**

This is updated by a comparison instruction, and is used for a condition of a subsequent branch instruction. If the result of the comparison instruction is true, T = 1. If the result of the comparison instruction is false, T = 0. The T flag can be accessed as the XBUS register from the CPU.

● **C Flag**

This is a carry of the arithmetic logic operation result, and can be accessed from the CPU as the XBUS register.

● **Prescaler**

This generates the TIME count signal, and has an 8-bit counter configuration. The prescaler operates from issuing the TIMWAIT instruction until the TIME register count is completed by the TIMWAIT instruction. To clear the prescaler, set EPCTRLn.RESET = 0.

10.5. Instruction

10.5.1. Instruction Format

Operations when an undefined instruction code is executed are not guaranteed. The execution of any undefined instruction code is prohibited.

Table 10-2. Instructions

Instruction	Description	OPCODE (16 bits)
ALU Rn, Rm	ALU Operation, Rn ← Rn op Rm (16 bits)	0 0 0 0 0 0 N M 0 AM
MUL Rn	Rn ← Unsigned R0[7:0] * Unsigned R1[7:0]	0 0 0 0 0 0 N 0 1111
CMP Rn, Rm	Compare Operation, T ← Rn cmp Rm (16 bits)	0 0 0 0 0 0 N M 1 CM
EVTWAIT #EVT	Wait for Event #EVT (each thread owns 16x event-pending-flags)	0 0 0 0 0 1 0 0 EVT
EVTIN #EVT	T←Event #EVT (no wait; only get specified event-pending-flags)	0 0 0 0 0 1 0 1 EVT
EVTOUT #EVT	Output Event #EVT	0 0 0 0 0 1 1 0 EVT
EVTCLR #EVT	Clear Event Flag #EVT	0 0 0 0 0 1 1 1 EVT
JT @AddrM	If (T==1) PC ← {AddrM[8:1], 1'b0}, else PC ← PC+2	0 0 0 0 1 0 0 AddrM[8:1] 0
JF @AddrM	If (T==0) PC ← {AddrM[8:1], 1'b0}, else PC ← PC+2	0 0 0 0 1 0 1 AddrM[8:1] 0
JMP @AddrM	PC ← {AddrM[8:1], 1'b0}	0 0 0 0 1 1 AddrM[8:1] 0
LOADS Rn, @AddrS	Load Rn from S@AddrS	0 0 0 1 RM N 0 AddrS[7:0] (8 bits)
STORES @AddrS, Rm	Store Rn to S@Addrss	0 0 0 1 WM M 1 AddrS[7:0] (8 bits)
LOADM Rn, @AddrM	Load Rn from M@AddrM	0 0 1 0 RM N AddrM[8:0] (9 bits)
STOREM @AddrM, Rm	Store Rm to M@AddrM	0 0 1 1 WM M AddrM[8:0] (9 bits)
TIMWAIT #TIME	Wait for #TIME x prescaler cycles	0 1 0 0 TIME[11:0] (12 bits)
LOADX Rn, @(Rm+ZE(Offset))	Load Rn from X@(Rm+ZE(Offset))	1 0 N M RM Offset (10 bits)
STOREX @(Rn+ZE(Offset)), Rm	Store Rn to X@(Rn+ZE(Offset))	1 1 N M WM Offset (10 bits)

Read Mode	Read Operation	RM		Remarks
B_SE	8 bits, sign extended	0	0	X, M
B_ZE	8 bits, zero extended	0	1	X, M
W	16 bits, twice with address increment 1st read data is stored to Rn[7:0], 2nd read data is stored to Rn[15:8]	1	0	X
W_SA	16 bits, twice on same address 1st read data is stored to Rn[7:0], 2nd read data is stored to Rn[15:8]	1	1	X
B_LO	8-bit read mode (Byte access mode on DSAC), store to Rn[7:0]	0	0	S
B_HI	8-bit read mode (Byte access mode on DSAC), store to Rn[15:8]	0	1	S
W	16-bit read mode (Word access mode on DSAC)	1	0	S, M

Write Mode	Write Operation	WM		Remarks
B_LO	8 bits, lower side of operand	0	0	X, M
B_HI	8 bits, higher side of operand	0	1	X, M
W	16 bits, twice with address increment 1st write data is Rm[7:0], 2nd write data is Rm[15:8]	1	0	X
W_SA	16 bits, twice on same address 1st write data is Rm[7:0], 2nd write data is Rm[15:8]	1	1	X
B_LO	8-bit write mode (Byte access mode on DSAC), write Rn[7:0]	0	0	S
B_HI	8-bit write mode (Byte access mode on DSAC), write Rn[15:8]	0	1	S
W	16-bit write mode (Word access mode on DSAC)	1	0	S, M

ALU Mode	Description	AM Code			
MOV	Rn ← Rm	0	0	0	0
ADD	{C, Rn} ← Rn + Rm (16 bits)	0	0	0	1
ADDC	{C, Rn} ← Rn + Rm + {{15{0}}, C} (16 bits)	0	0	1	0
SUB	{C, Rn} ← Rn - Rm (16 bits)	0	0	1	1
SUBC	{C, Rn} ← Rn - Rm - {{15{0}}, C} (16 bits)	0	1	0	0
INC	Rn ← Rm + 1 (16 bits)	0	1	0	1
DEC	Rn ← Rm – 1 (16 bits)	0	1	1	0
AND	Rn ← Rn & Rm (16 bits)	0	1	1	1
OR	Rn ← Rn | Rm (16 bits)	1	0	0	0
XOR	Rn ← Rn ^ Rm (16 bits)	1	0	0	1
NOT	Rn ← ~Rm (16 bits)	1	0	1	0
SFTL	Rn ← (Rm << 1)	1	0	1	1
SFTR	Rn ←(Rm >> 1)	1	1	0	0
SFTLC	C ← Rm[15], Rn ← (Rm << 1), Rn[0] ← C	1	1	0	1
SFTRC	C ← Rm[0], Rn ← (Rm >> 1), Rn[15] ← C	1	1	1	0
MUL Rn	Rn ← Unsigned R0[7:0] * Unsigned R1[7:0]	1	1	1	1

Compare Mode	Description	CM Code			
EQ	T ← (Rn == Rm)	0	0	0	0
GTS	T ← (Rn > Rm), Signed	0	0	1	0
LTS	T ← (Rn < Rm), Signed	0	1	0	0
GTU	T ← (Rn > Rm), Unsigned	0	1	1	0
LTU	T ← (Rn < Rm), Unsigned	1	0	0	0
ZERO	T ← (Rn == 0)	1	0	0	1
CLRT	T ← 0	1	0	1	0
SETT	T ← 1	1	0	1	1
GETC	T ← C	1	1	0	0
PUTC	C ← T	1	1	0	1
CLRC	C ← 0	1	1	1	0
SETC	C ← 1	1	1	1	1

10.5.2. Instruction Set

In this section, the program counter and the dedicated timer counter are represented as the PC and the TIME, respectively. The R0 and R1 are general-purpose registers. For the details of each thread resource, see Section 10.4.

10.5.2.1. Instruction of Arithmetic Logic Unit (ALU)

- **MOV**

This is a data transfer instruction between registers. Rm is written to Rn. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2.

- **ADD**

The result of Rn + Rm is stored to Rn. The carry of the calculation result is indicted in the C flag. The T flag is not updated. After the instruction is executed, the PC is incremented by 2.

- **ADDC**

The result of Rn + Rm + C is stored to Rn. The carry of the calculation result is indicted in the C flag. The T flag is not updated. After the instruction is executed, the PC is incremented by 2.

- **SUB**

The result of Rn − Rm is stored to Rn. The borrow of the calculation result is indicted in the C flag. The T flag is not updated. After the instruction is executed, the PC is incremented by 2.

- **SUBC**

The result of Rn − Rm − C is stored to Rn. The borrow of the calculation result is indicted in the C flag. The T flag is not updated. After the instruction is executed, the PC is incremented by 2.

- **INC**

The result that Rm is incremented by 1 is stored to Rn. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2.

- **DEC**

The result that Rm is decremented by 1 is stored to Rn. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2.

- **AND**

The result of logical AND of Rn and Rm is stored to Rn. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2.

- **OR**

The result of logical OR of Rn and Rm is stored to Rn. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2.

- **XOR**

The result of exclusive logical OR of Rn and Rm is stored to Rn. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2.

- **NOT**

The bit inversion of Rm is stored to Rn. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2.

- **SFTL**
  The result that Rm is shifted left by 1 bit is stored to Rn. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2. The value to be stored to the LSB (least significant bit) is 0.

- **SFTR**
  The result that Rm is shifted right by 1 bit is stored to Rn. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2. The value to be stored to the MSB (most significant bit) is 0.

- **SFTLC**
  The result that Rm is shifted left by 1 bit is stored to Rn. At this time, the present C flag is stored to the LSB of Rn. Also, the MSB of Rm pushed out by left shifting is stored to the C flag. After the instruction is executed, the PC is incremented by 2.

- **SFTRC**
  The result Rm is shifted right by 1 bit is stored to Rn. At this time, the present C flag is stored to the MSB of Rn. Also, the LSB of Rm pushed out by right shifting is stored to the C flag. After the instruction is executed, the PC is incremented by 2.

- **MUL**
  The product of the lower 8 bits of R0 and R1 is stored to Rn. The R0 and R1 are regarded as unsigned values. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2.

10.5.2.2. Comparison Instruction and Flag Operation Instruction

- **EQ**
  Rn = Rm: The T flag is set to 1. The PC is incremented by 2.
  Not Rn = Rm: The T flag is set to 0. The PC is incremented by 2.

- **GTS**
  This regards Rn and Rm as the signed values.
  Rn > Rm: The T flag is set to 1. The PC is incremented by 2.
  Rn ≤ Rm: The T flag is set to 0. The PC is incremented by 2.

- **LTS**
  This regards Rn and Rm as the signed values.
  Rn < Rm: The T flag is set to 1. The PC is incremented by 2.
  Rn ≥ Rm: The T flag is set to 0. The PC is incremented by 2.

- **GTU**
  This regards Rn and Rm as the unsigned values.
  Rn > Rm: The T flag is set to 1. The PC is incremented by 2.
  Rn ≤ Rm: The T flag is set to 0. The PC is incremented by 2.

- **LTU**
  This regards Rn and Rm as the unsigned values.
  Rn < Rm: The T flag is set to 1. The PC is incremented by 2.
  Rn ≥ Rm: The T flag is set to 0. The PC is incremented by 2.

- **ZERO**
  Rn = 0: The T flag is set to 1. The PC is incremented by 2.
  Other than Rn = 0: The T flag is set to 0. The PC is incremented by 2.

- **CLRT**
  The T flag is set to 0. After the instruction is executed, the PC is incremented by 2.

- **SETT**
  The T flag is set to 1. After the instruction is executed, the PC is incremented by 2.

- **GETC**
  The C flag is transferred to the T flag. After the instruction is executed, the PC is incremented by 2. The C flag is not changed.

- **PUTC**
  The T flag is transferred to the C flag. After the instruction is executed, the PC is incremented by 2.The T flag is not changed.

- **CLRC**
  The C flag is set to 0. After the instruction is executed, the PC is incremented by 2.

- **SETC**
  The C flag is set to 1. After the instruction is executed, the PC is incremented by 2.

10.5.2.3. Branch Instruction

- **JMP**
  This is an unconditional branch instruction. The specified address is set to the PC.

- **JT**
  The T flag is 1: The specified address is set to the PC. The instruction is taken.
  The T flag is 0: The PC is incremented by 2. The instruction is not taken.

- **JF**
  The T flag is 0: The specified address is set to the PC. The instruction is taken.
  The T flag is 1: The PC is incremented by 2. The instruction is not taken.

10.5.2.4. Data Transfer Instruction

- **LOADS Rn, @AddrS**
  The SBUS address, AddrS, data is stored to Rn. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2. The instruction has 3 access modes (B_LO, B_HI, and W).

- **STORES @AddrS, Rm**
  Rm is written to the SBUS address, AddrS. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2. The instruction has 3 access modes (B_LO, B_HI, and W).

- **LOADM Rn, @AddrM**
  The MBUS address, AddrM, data is stored to Rn. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2. The instruction has 3 access modes (B_SE, B_ZE, and W).

- **STOREM @AddrM, Rm**
  Rm is written to the MBUS address, AddrM. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2. The instruction has 3 access modes (B_LO, B_HI, and W).

- **LOADX Rn, @(Rm+ZE(offset))**
  The XBUS address, Rm + offset (register relative), data is stored to Rn. The offset is unsigned 10-bit width. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2. The XBUS address has 16

bits. The 17th bit of the address calculation result is ignored. The instruction has 4 access modes (B_SE, B_ZE, W, and W_SA). In the word access mode of W or W_SA, the XBUS access occurs twice.

- **STOREX @(Rn+ZE(offset)), Rm**

Rm is stored to the XBUS address, Rn + offset (register relative). The offset is unsigned 10-bit width. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2. The XBUS address has 16 bits. The 17th bit of the address calculation result is ignored. The instruction has 4 access modes (B_LO, B_HI, W, and W_SA). In the word access mode of W or W_SA, the XBUS access occurs twice.

10.5.2.5. Event Instruction

- **EVTIN**

This stores 1 or 0 to the T flag when the specified event exists or does not exist, respectively. When the bit in the EPEISLn or EPEISHn register corresponding to the event status of the specified event number is 1, T = 1, and when it is 0, T = 0. The corresponding bit of the EPEISLn or EPEISHn register is not changed by the EVTIN instruction. The C flag is not updated. After the instruction is executed, the PC is incremented by 2.

- **EVTOUT**

This outputs the event corresponding to the specified event number. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2.

- **EVTCLR**

The corresponding bit of the event status in the EPEISLn or EPEISHn register is cleared. When the clearing operation conflicts with the setting operation, the clearing operation is ignored. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2.

- **EVTWAIT**

The specified event is waited. The thread execution is stopped (i.e., thread execution right is abandoned) until the corresponding bit of the event status in the EPEISLn or EPEISHn register becomes 1. When the specified event is input, and the corresponding bit in the EPEISLn or EPEISHn register becomes 1, the thread execution right is required, and the thread execution is restarted (i.e., thread execution right is acquired). When the EVTWAIT instruction is issued while the specified event exists (when the corresponding bit in the EPEISLn or EPEISHn register is 1), the execution of thread subsequent instruction is continued because the thread execution is not stopped. When the thread execution is restarted, the corresponding bit in the EPEISLn or EPEISHn register is cleared. The C and T flags are not updated. After the instruction is executed, the PC is incremented by 2. While the event is being waited, the thread execution is stopped.

- **TIMWAIT**

For the specified period, the thread execution is stopped (i.e., the thread execution right is abandoned). The C and T flags are not updated. When the instruction is executed, the value specified by this instruction is loaded to the TIME of each thread. The TIME counts down according to the count signal divided by the prescaler. When the TIME becomes 0, the suspended thread requires the thread execution right, and the thread execution is restarted. Even if 0 is specified for the TIME, the thread execution is stopped once, and the thread execution right is required at the timing of the next count signal. After the instruction is executed, the PC is incremented by 2.

10.6. Operation

10.6.1. Program Allocation

The EPU program can be allocated in 512-bytes area (RAM1) of 1.5-KB XRAM on the XBUS. This area is shared with the CPU. The addresses when accessing RAM1 from the CPU are 0x0400 to 0x05FF. The EPU program has 16-bit fixed-length. In the CPU address, the lower 8 bits and the higher 8 bits of the instruction codes are assigned to 2n (even address) and 2n + 1 (odd address), respectively.

10.6.2. Thread Resource Setting

Each thread has the resources such as general-purpose registers (R0 and R1), program counter (PC), dedicated timer counter (TIME), prescaler, T flag, and C flag. These resources can be accessed from the CPU as the registers on the XBUS. The initial value can be set to these resources before the thread activation. The program execution start address must be set to the PC.

10.6.3. Thread Activation and Stop

To activate the thread, set EPCTRLn.EN = 1. When the thread acquires the execution right, the instruction execution is started. To stop the thread at completing the instruction that is being executed, set EPCTRLn.EN = 0. If the TIME counts, the count is suspended. The prescaler just stops and not be reset. The event input can be accepted even while the thread is stopped. When the event input is detected, the EPEISLn and EPEISHn registers are changed. If the thread is stopped with the state waiting for the execution right, the thread execution right request is canceled. When EN = 0, the instructions already fetched or being executed are executed as they are. The stop states by the TIMWAIT and EVTWAIT instructions are held.

To reset the thread internal state (the EPSTSn.THSTS bit) and the prescaler to the initial state, set EPCTRLn.RESET = 1. The register values of other threads without EPCTRLn.RESET = 1 are not reset and are held. The thread state is initialized while the thread is disabled (i.e., EPCTRLn.EN = 0). Also, using this reset function, the TIMWAIT or EVTWAIT instruction forcibly cancels the thread execution wait state while the thread is enabled (i.e., EPCTRLn.EN = 1).

10.6.4. Thread Selection (Context Switch)

The thread to be executed is selected according to the thread priority from each thread execution request. For high speed switching of the threads to be executed, the contexts are switched with zero latency by the context switch. The normal thread priority is as follows: the highest priority when the thread number is low; the lowest priority when the thread number is high. The thread with the highest priority is selected from each thread outputs the execution request.

The priority can also be controlled by grouping. The EPCTRLn.PRI bits determine whether each thread belongs to one of 3 groups (A, B, or C) or not belong to any group. If EPCTRLn.PRI = 0b00, thread n does not belong to any priority control group, and the thread is selected with the normal thread priority. When the group A (EPCTRLn.PRI = 0b01), B (EPCTRLn.PRI = 0b10), and C (EPCTRLn.PRI = 0b11) are set, the priority is controlled according to a round-robin algorithm in the same group. In the initial state, Thread0 is the highest priority, followed by, Thread1, Thread2, ⋯, Thread5. For example, when the Thread3 is selected, Thread4 is the highest priority in the next cycle, followed by Thread5, Thread0, Thread1, ⋯, Thread3. The thread selected in the previous time is the lowest priority. The priorities between groups are determined by the normal priority from the threads selected by the individual group. Assigning the threads with the same frequency and the timing of operations to the same group makes it easier for the threads to operate in parallel. If the multiple threads that always operate exist, these threads must be assigned to the same group.

The thread that executed the TIMWAIT or EVTWAIT instruction temporarily cancels the thread execution request. Then, the execution right is transferred to other threads that the context switch outputs the execution request. The EPU stops the instruction execution during no execution request from all threads. The thread suspended by the TIMWAIT instruction outputs the thread execution request when the TIME counts down to 0, and tries to restart the thread execution. The thread suspended by the EVTWAIT instruction outputs the thread execution request by the event input specified by the EVTWAIT instruction, and tries to restart the thread execution.

10.6.5. Event Input

Each thread has a function that detects 16 event inputs. The event acceptance is defined by the EPEICLn and EPEICHn registers. While the event is input when the thread is enabled (EPCTRL.EN = 1) and the corresponding bit of the EPEICLn or EPEICHn register is 1, the event input is accepted, and then the corresponding bits of the EPEISLn and EPEISHn registers set to 1. When the event input specified by the EVTWAIT instruction is accepted (i.e., the thread execution right is acquired), the corresponding bit of the EPEISLn or EPEISHn register is cleared. When the setting operation conflicts with the clearing operation, the setting operation has the highest priority. When the event input is set the level signal, this event is regarded as "continuous event." While the event is input, the corresponding bit of the EPEISLn or EPEISHn register holds the state of 1. Even if the event is lost, the corresponding bit of the EPEISLn or EPEISHn register is not cleared. The state of 1 is held unless the event is accepted by the EVTWAIT instruction. For the selection of event input, see Section 9.

10.6.6. Event Output

Each thread outputs the 16 events by the EVTOUT instruction.

Table 10-3. Event Output

Event No.	Thread0	Thread1	Thread2
0	PWM0, PWM1, PWM2, PWM3 re-trigger	PWM0, PWM1, PWM2, PWM3 re-trigger	PWM0, PWM1, PWM2, PWM3 re-trigger
1	PWM0, PWM1, PWM2, PWM3 re-trigger	PWM0, PWM1, PWM2, PWM3 re-trigger	PWM0, PWM1, PWM2, PWM3 re-trigger
2	ADC0, ADC1	ADC0, ADC1	ADC0, ADC1
3	ADC0, ADC1	ADC0, ADC1	ADC0, ADC1
4	ADC0, ADC1	ADC0, ADC1	ADC0, ADC1
5	ADC0, ADC1	ADC0, ADC1	ADC0, ADC1
6	ADC0, ADC1	ADC0, ADC1	ADC0, ADC1
7	ADC0, ADC1	ADC0, ADC1	ADC0, ADC1
8	ADC0, ADC1	ADC0, ADC1	ADC0, ADC1
9	ADC0, ADC1	ADC0, ADC1	ADC0, ADC1
10	POC0, POC1, POC2, POC3	POC0, POC1, POC2, POC3	POC0, POC1, POC2, POC3
11	POC0, POC1, POC2, POC3	POC0, POC1, POC2, POC3	POC0, POC1, POC2, POC3
12	AMPON0, AMPON1	AMPON0, AMPON1	AMPON0, AMPON1
13	AMPOFF0, AMPOFF1	AMPOFF0, AMPOFF1	AMPOFF0, AMPOFF1
14	CPU INT	CPU INT	CPU INT
15	Reserve	Reserve	Reserve

Event No.	Thread3	Thread4	Thread5
0	PWM0, PWM1, PWM2, PWM3 re-trigger	PWM0, PWM1, PWM2, PWM3 re-trigger	PWM0, PWM1, PWM2, PWM3 re-trigger
1	PWM0, PWM1, PWM2, PWM3 re-trigger	PWM0, PWM1, PWM2, PWM3 re-trigger	PWM0, PWM1, PWM2, PWM3 re-trigger
2	ADC0, ADC1	ADC0, ADC1	ADC0, ADC1
3	ADC0, ADC1	ADC0, ADC1	ADC0, ADC1
4	ADC0, ADC1	ADC0, ADC1	ADC0, ADC1
5	ADC0, ADC1	ADC0, ADC1	ADC0, ADC1
6	ADC0, ADC1	ADC0, ADC1	ADC0, ADC1
7	ADC0, ADC1	ADC0, ADC1	ADC0, ADC1
8	ADC0, ADC1	ADC0, ADC1	ADC0, ADC1
9	ADC0, ADC1	ADC0, ADC1	ADC0, ADC1
10	POC0, POC1, POC2, POC3	POC0, POC1, POC2, POC3	POC0, POC1, POC2, POC3
11	POC0, POC1, POC2, POC3	POC0, POC1, POC2, POC3	POC0, POC1, POC2, POC3
12	AMPON0, AMPON1	AMPON0, AMPON1	AMPON0, AMPON1
13	AMPOFF0, AMPOFF1	AMPOFF0, AMPOFF1	AMPOFF0, AMPOFF1
14	CPU INT	CPU INT	CPU INT
15	Reserve	Reserve	Reserve

10.6.7. Bus Access

The EPU is the bus master of the MBUS. The MBUS is used for the instruction fetch and the data accesses of the LOADM or STOREM instruction. In the LOADM or STOREM instruction, the EPU accesses by the absolute address.

The EPU also operates as the bus master of the XBUS and the SBUS. The access to the XBUS is performed by an indirect addressing with the address that the offset specified to the Rn register. The SBUS is accessed by the LOADS or STORES instruction. The access to the SBUS is performed by a direct addressing that specifies the absolute address during the instruction. The XBUS is accessed by the LOADX or STOREX instruction.

10.6.8. MBUS Access

The MBUS is for the data accesses of the instruction fetch and the LOADM or STOREM instruction. The bus width is 16 bits (2 bytes). The EPU can access up to 2 byte-data at the same time. The instruction fetch is performed in 2-byte unit of a word alignment. The instruction allocation address must be the word alignment. The data access by the LOADM or STOREM instruction can be performed in 1-byte or 2-byte (word) unit. In the word access mode, only word alignment address can be used. In the data access mode, the read mode (RM) or the write mode (WM) during the LOADM or STOREM instruction is specified. In the LOADM or STOREM instruction, the data access and the instruction fetch of the subsequent instruction occur simultaneously; as a result, the resource competition hazard of the MBUS occurs. At this time, the data access is processed first. After that, the instruction fetch access of the subsequent instruction occurs. While the data is being accessed, the instruction execution is waited.

10.6.9. XBUS Access

The XBUS is for the data access of the LOADX or STOREX instruction. The bus width is 8 bits (1 byte). The XBUS can access the peripheral function register and the XRAM shared with the CPU. The access mode is specified from the read mode (RM) or the write mode (WM) of the LOADX or STOREX instruction. The EPU can access in 1-byte or 2-byte unit. The XBUS access occurs twice in 2-byte access. The address of the LOADX or STOREX instruction is the address that the offset specified to the Rn register is added.

10.6.10. SBUS Access

The SBUS is for the data access of the LOADS or STORES instruction. The bus width is 16 bits (2 bytes). The EPU can access the SFR of the peripheral function, and cannot access the SFR related to the CPU and the INTC. The access mode is specified from the read mode (RM) or the write mode (WM) of the LOADS or STORES instruction. The EPU can access in 1-byte or 2-byte unit. In the SBUS, up to 2-byte data is assigned to one address.

10.6.11. XBUS Subordinate Register Access

The EPU register access is performed via the XBUS. The XBUS has 8-bit data width. On the other hand, since R0 and R1 are 16-bit width, the program counter (PC) is 9-bit width, and the dedicated timer counter (TIME) is 12-bit width, the value cannot be set in one write operation from the XBUS; therefore the value must be set in 2 write operations (lower 8 bits and higher 8 bits). Be sure to access these registers lower bits, higher bits, in the sequential order.

The XBUS register access to the above registers is as follows: the write data to the lower bits are once stored in the buffer. The stored values are reflected concurrently with writing to the higher bits. This guarantees the atomicity when accessing 16-bit register.

Likewise, the higher bit is held in the buffer when reading the lower bit. Regarding the read data of the higher bit, the value held in the buffer is read. The buffer is mounted independently for the CPU access and the EPU access. As a result, the atomicity is guaranteed even if the accesses of the EPU and the CPU occur alternately.

10.6.12. Pipeline Operation of Each Instruction

In the EPU, the memory read instructions (i.e., instructions with the write back stage such as LOADS, LOADM, and LOADX) operate with a 3-stage pipeline. Other instructions operate with a 2-stage pipeline. For the pipeline operation of each instruction, see Figure 10-2.

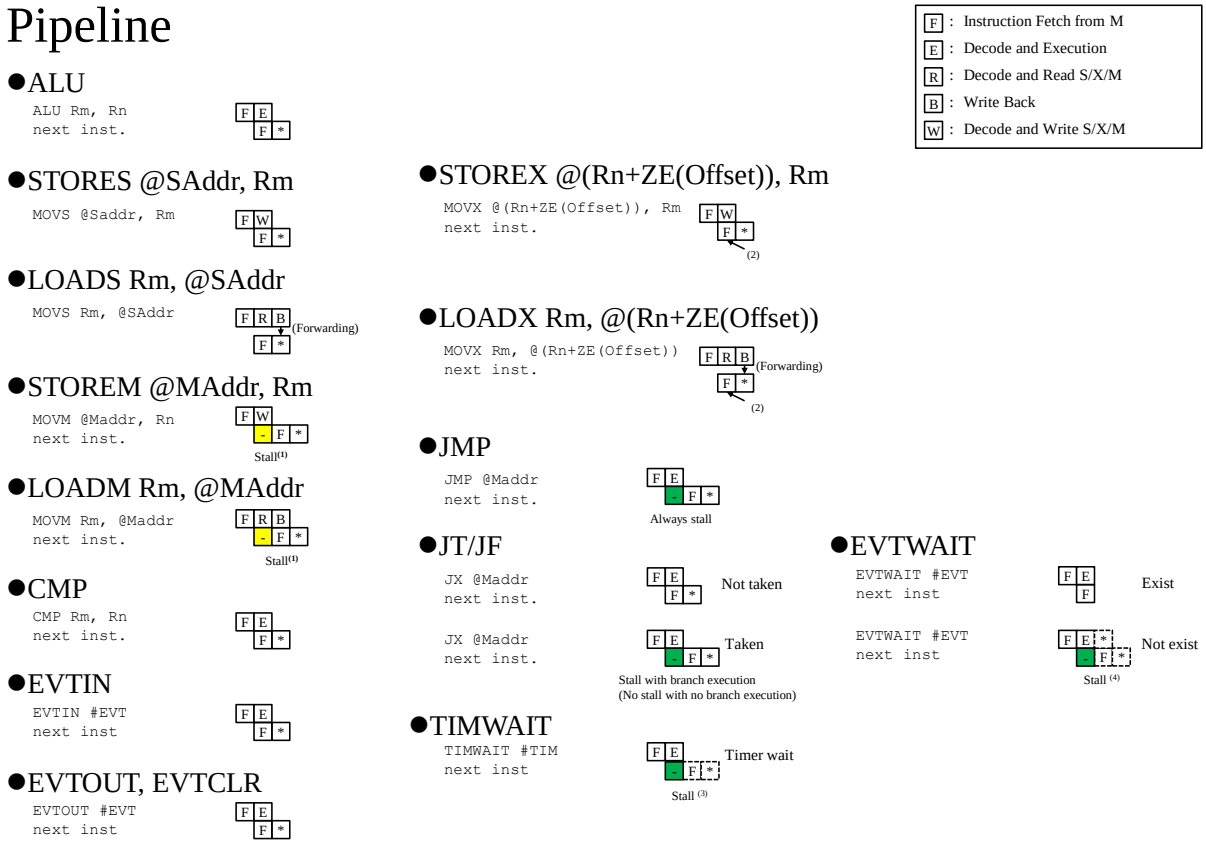


Figure 10-2. Pipeline Operation of Each Instruction

(1) The data access and the instruction fetch of the subsequent instruction occur simultaneously to the RAM1 (memory where the program is allocated). As a result, the resource conflict hazard of the MBUS access occurs. For this reason, the subsequent instruction is stalled.
(2) Since the data conflicts with the instruction fetch when accessing to the RAM1, the subsequent instruction is stalled.
(3) The subsequent instruction is stalled. Note that the instruction is fetched if the activation request of the other thread exists. Even if 0 is specified for the TIME, the thread execution is stopped once. The next count signal requires the thread execution right.
(4) The subsequent instruction is stalled. Note that the instruction is fetched if the activation request of the other thread exists.

10.6.13. Execution Time of Each Instruction

The EPU operates in 1 cycle per instruction except for the following instructions.

- **Branch Instruction**
  - Unconditional branch instruction: 2 cycles
  - Conditional branch instruction is taken: 2 cycles (1 cycle when the conditional branch instruction is not taken)

- **LOADM or STOREM Instruction**
  2 cycles (to the RAM1, the instruction fetch and the data access conflict)

- **LOADX or STOREX Instruction**
  1 cycle, 2 cycles when accessing to the RAM1

- **LOADS or STORES Instruction and LOADX or STOREX Instruction**
  When the wait cycle exists in the bus cycle of the data access of these instructions, the cycles of the wait cycle are added to the original instruction execution cycle.
  - For the SBUS of the LOADS or STORES instruction, the DSAC has the highest priority. The wait cycles are added according to the DSAC transfer count: 2 cycles (max.) for 1 count, 4 cycles (max.) for 2 counts, 8 cycles (max.) for 4 counts, and 16 cycles (max.) for 8 counts.
  - Since the XBUS of the LOADX or STOREX instruction is accessed with 2-cycle, 1-cycle wait is added.

- **EVTWAIT Instruction**
  - The corresponding event already exists: 1 cycle
  - No corresponding event exists: 2 cycles after the corresponding event is detected

- **Subsequent Instruction of TIMWAIT Instruction**
  2 cycles

10.7. Register Descriptions

Table 10-4. List of Registers

Symbol*	Name*	Address*	Initial Value
EPMCR	EPU Master Control Register	0xE000	0x00
EPCTRLn	EPU Control Register for Thread n	$0xE000 + 0x10 \times (n + 1)$	0x00
EPSTSn	EPU Status Register for Thread n	$0xE001 + 0x10 \times (n + 1)$	0x00
EPR0Ln	EPU R0 Register Lower Side for Thread n	$0xE002 + 0x10 \times (n + 1)$	0x00
EPR0Hn	EPU R0 Register Higher Side for Thread n	$0xE003 + 0x10 \times (n + 1)$	0x00
EPR1Ln	EPU R1 Register Lower Side for Thread n	$0xE004 + 0x10 \times (n + 1)$	0x00
EPR1Hn	EPU R1 Register Higher Side for Thread n	$0xE005 + 0x10 \times (n + 1)$	0x00
EPPCLn	EPU Program Counter Register Lower Side for Thread n	$0xE006 + 0x10 \times (n + 1)$	0x00
EPPCHn	EPU Program Counter Register Higher Side for Thread n	$0xE007 + 0x10 \times (n + 1)$	0x00
EPTIMELn	EPU Timer Counter Register Lower Side for Thread n	$0xE008 + 0x10 \times (n + 1)$	0x00
EPTIMEHn	EPU Timer Counter Register Higher Side for Thread n	$0xE009 + 0x10 \times (n + 1)$	0x00
EPEICLn	EPU Event Input Control Register Lower Side for Thread n	$0xE00A + 0x10 \times (n + 1)$	0x00
EPEICHn	EPU Event Input Control Register Higher Side for Thread n	$0xE00B + 0x10 \times (n + 1)$	0x00
EPEISLn	EPU Event Input Status Register Lower Side for Thread n	$0xE00C + 0x10 \times (n + 1)$	0x00
EPEISHn	EPU Event Input Status Register Higher Side for Thread n	$0xE00D + 0x10 \times (n + 1)$	0x00
EPPSPn	EPU Prescaler Period Register for Thread n	$0xE00E + 0x10 \times (n + 1)$	0x00
EPEISCn	EPU Event Input Status Control Register for Thread n	$0xE00F + 0x10 \times (n + 1)$	0x00

* The arbitrary letter “n” represents a thread number.

10.7.1. EPMCR (EPU Master Control Register)

Register	EPMCR	EPU Master Control Register	Address	0xE000

Bit	Bit Name	R/W	Initial	Description	Remarks
7	PRIRST	W	0	Resetting the priority control mechanism Write 0: No change Write 1: Reset The read value is always 0. The bit resets the state of the priority control mechanism.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

10.7.2. EPCTRLn (EPU Control Register for Thread n) (n = 0 to 5)

Register	EPCTRLn	EPU Control Register for Thread n	Address	See Table 10-4

Bit	Bit Name	R/W	Initial	Description	Remarks
7	EN	R/W	0	Enabling the thread n 0: Thread n is disabled (stop) 1: Thread n is enabled (execution) The bit enables or disables the thread n. When the bit is enabled, the corresponding thread sends an instruction execution request to the EPU core. When the bit is disabled, the corresponding thread cancels the instruction execution request.	
6	RESET	W	0	Resetting the status of the thread n Writing 1 to the bit clears the status of thread n (the EPSTSn.THSTS bit) and the prescaler, and initializes the status of the thread n.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	PRI	R/W	0	Selecting the group for priority control 00: Ungrouped (fixed priority control) 01: Group A 10: Group B 11: Group C	
0		R/W	0		

10.7.3. EPSTSn (EPU Status Register for Thread n) (n = 0 to 5)

Register	EPSTSn	EPU Status Register for Thread n	Address	See Table 10-4

Bit	Bit Name	R/W	Initial	Description	Remarks
7	THSTS	R	0	Thread status 00: Thread n is stopped 01: Thread n is active 10: Waiting for a specified event 11: Waiting for the timer count to be finished	
6		R	0		
5	SETC	W	0	Setting the C flag Writing 1 to the bit sets the C bit to 1. The read value is 0.	
4	SETT	W	0	Setting the T flag Writing 1 to the bit sets the T bit to 1. The read value is 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	C	R/C	0	C flag Read: A value of the C flag Write 0: No change Write 1: The bit is cleared The priority is defined as follows: clearing the bit by writing 1 to the bit > setting the bit by the STEC bit > updating the bit by the EPU.	
0	T	R/C	0	T flag Read: A value of T flag Write 0: No change Write 1: The bit is cleared The priority is defined as follows: clearing the bit by writing 1 to the bit > setting the bit by the SETT bit > updating the bit by the EPU.	

10.7.4. EPR0Ln (EPU R0 Register Lower Side for Thread n) (n = 0 to 5)

Register	EPR0Ln	EPU R0 Register Lower Side for Thread n		Address	See Table 10-4
Bit	Bit Name	R/W	Initial	Description	Remarks
7	R0	R/W	0	The lower bits of the R0 register Write: The bits must be sequentially written by the CPU in low-to-high order. The write data to the lower bits is once stored in the buffer; then, the stored values are written to the register concurrently with writing to the higher bits. Read: The bits must be sequentially read by the CPU in low-to-high order. When the lower bits are read, the values of the higher bits are stored in the buffer; then, the stored values are read at a time of reading the higher bits. In the case where a write-by-CPU conflicts with an update-by-EPU, the write-by-CPU takes precedence.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

10.7.5. EPR0Hn (EPU R0 Register Higher Side for Thread n) (n = 0 to 5)

Register	EPR0Hn	EPU R0 Register Higher Side for Thread n		Address	See Table 10-4
Bit	Bit Name	R/W	Initial	Description	Remarks
7	R0	R/W	0	The higher bits of the R0 register Write: The bits must be sequentially written by the CPU in low-to-high order. Read: The bits must be sequentially read by the CPU in low-to-high order. In the case where a write-by-CPU conflicts with an update-by-EPU, the write-by-CPU takes precedence.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

10.7.6. EPR1Ln (EPU R1 Register Lower Side for Thread n) (n = 0 to 5)

Register	EPR1Ln	EPU R1 Register Lower Side for Thread n		Address	See Table 10-4
Bit	Bit Name	R/W	Initial	Description	Remarks
7	R1	R/W	0	The lower bits of the R1 register Write: The bits must be sequentially written by the CPU in low-to-high order. The write data to the lower bits is once stored in the buffer; then, the stored values are written to the register concurrently with writing to the higher bits. Read: The bits must be sequentially read by the CPU in low-to-high order. When the lower bits are read, the values of the higher bits are stored in the buffer; then, the stored values are read at a time of reading the higher bits. In the case where a write-by-CPU conflicts with an update-by-EPU, the write-by-CPU takes precedence.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

10.7.7. EPR1Hn (EPU R1 Register Higher Side for Thread n) (n = 0 to 5)

Register	EPR1Hn	EPU R1 Register Higher Side for Thread n		Address	See Table 10-4
Bit	Bit Name	R/W	Initial	Description	Remarks
7	R1	R/W	0	The higher bits of the R1 register Write: The bits must be sequentially written by the CPU in low-to-high order. Read: The bits must be sequentially read by the CPU in low-to-high order. In the case where a write-by-CPU conflicts with an update-by-EPU, the write-by-CPU takes precedence.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

10.7.8. EPPCLn (EPU Program Counter Register Lower Side for Thread n) (n = 0 to 5)

Register	EPPCLn	EPU Program Counter Register Lower Side for Thread n		Address	See Table 10-4
Bit	Bit Name	R/W	Initial	Description	Remarks
7	PC	R/W	0	The lower bits of the program counter Write: The bits must be sequentially written by the CPU in low-to-high order. The write data to the lower bits is once stored in the buffer; then, the stored values are written to the register concurrently with writing to the higher bits. Read: The bits must be sequentially read by the CPU in low-to-high order. When the lower bits are read, the values of the higher bits are stored in the buffer; then, the stored values are read at a time of reading the higher bits. In the case where a write-by-CPU conflicts with an update-by-EPU, the write-by-CPU takes precedence.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R	0		

10.7.9. EPPCHn (EPU Program Counter Register Higher Side for Thread n) (n = 0 to 5)

Register	EPPCHn	EPU Program Counter Register Higher Side for Thread n		Address	See Table 10-4
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	PC	R/W	0	The higher bits of the program counter Write: The bits must be sequentially written by the CPU in low-to-high order. Read: The bits must be sequentially read by the CPU in low-to-high order. In the case where a write-by-CPU conflicts with an update-by-EPU, the write-by-CPU takes precedence.	

10.7.10. EPTIMELn (EPU Timer Counter Register Lower Side for Thread n) (n = 0 to 5)

Register	EPTIMELn	EPU Timer Counter Register Lower Side for Thread n		Address	See Table 10-4
Bit	Bit Name	R/W	Initial	Description	Remarks
7	TIME	R/W	0	The lower bits of the dedicated timer counter Write: The bits must be sequentially written by the CPU in low-to-high order. The write data to the lower bits is once stored in the buffer; then, the stored values are written to the register concurrently with writing to the higher bits. Read: The bits must be sequentially read by the CPU in low-to-high order. When the lower bits are read, the values of the higher bits are stored in the buffer; then, the stored values are read at a time of reading the higher bits. In the case where a write-by-CPU conflicts with an update-by-EPU, the write-by-CPU takes precedence.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

10.7.11. EPTIMEHn (EPU Timer Counter Register Higher Side for Thread n) (n = 0 to 5)

Register	EPTIMEHn	EPU Timer Counter Register Higher Side for Thread n		Address	See Table 10-4
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	TIME	R/W	0	The higher bits of the dedicated timer counter Write: The bits must be sequentially written by the CPU in low-to-high order. Read: The bits must be sequentially read by the CPU in low-to-high order. In the case where a write-by-CPU conflicts with an update-by-EPU, the write-by-CPU takes precedence.	
2		R/W	0		
1		R/W	0		
0		R/W	0		

10.7.12. EPEICLn (EPU Event Input Control Register Lower Side for Thread n) (n = 0 to 5)

Register	EPEICLn	EPU Event Input Control Register Lower Side for Thread n		Address	See Table 10-4
Bit	Bit Name	R/W	Initial	Description	Remarks
7	EVE[7]	R/W	0	Enabling the acceptance of events 0: Acceptance of events is disabled 1: Acceptance of events is enabled Setting the bit to 1 allows the corresponding event to be accepted. When an event applicable to the bit is input, the corresponding bit of the EPEISLn register is set. Setting the bit to 0 prohibits the corresponding event from being accepted. Even when an event applicable to the bit is input, the event is ignored, thus ineffective to the corresponding bit of the EPEISLn register.	
6	EVE[6]	R/W	0		
5	EVE[5]	R/W	0		
4	EVE[4]	R/W	0		
3	EVE[3]	R/W	0		
2	EVE[2]	R/W	0		
1	EVE[1]	R/W	0		
0	EVE[0]	R/W	0		

10.7.13. EPEICHn (EPU Event Input Control Register Higher Side for Thread n) (n = 0 to 5)

Register	EPEICHn	EPU Event Input Control Register Higher Side for Thread n		Address	See Table 10-4
Bit	Bit Name	R/W	Initial	Description	Remarks
7	EVE[15]	R/W	0	Enabling the acceptance of events 0: Acceptance of events is disabled 1: Acceptance of events is enabled Setting the bit to 1 allows the corresponding event to be accepted. When an event applicable to the bit is input, the corresponding bit of the EPEISHn register is set. Setting the bit to 0 prohibits the corresponding event from being accepted. Even when an event applicable to the bit is input, the event is ignored, thus ineffective to the corresponding bit of the EPEISHn register.	
6	EVE[14]	R/W	0		
5	EVE[13]	R/W	0		
4	EVE[12]	R/W	0		
3	EVE[11]	R/W	0		
2	EVE[10]	R/W	0		
1	EVE[9]	R/W	0		
0	EVE[8]	R/W	0		

10.7.14. EPEISLn (EPU Event Input Status Register Lower Side for Thread n) (n = 0 to 5)

Register	EPEISLn	EPU Event Input Status Register Lower Side for Thread n		Address	See Table 10-4
Bit	Bit Name	R/W	Initial	Description	Remarks
7	EVS[7]	R/C	0	Event input status Read 0: The corresponding event does not exist Read 1: The corresponding event exists Write 0: No change Write 1: The bit is cleared When an event applicable to the bit of the EPEICLn register, which is set to 1, is input, the corresponding bit of this register is set. When the EVTCLR or EVTWAIT instruction, whose event number agrees with that of the input event, is executed, the corresponding bit of this register is cleared. Writing 1 clears the bit. And the EPEISCn register defines the bit. A set operation by event input has the lowest priority, if it conflicts with either of a clear operation by writing 1 to the bit, or a set operation by the EPEISCn register. Writing by the CPU takes precedence over processing by the EVTCLR or EVTWAIT instruction.	
6	EVS[6]	R/C	0		
5	EVS[5]	R/C	0		
4	EVS[4]	R/C	0		
3	EVS[3]	R/C	0		
2	EVS[2]	R/C	0		
1	EVS[1]	R/C	0		
0	EVS[0]	R/C	0		

10.7.15. EPEISHn (EPU Event Input Status Register Higher Side for Thread n) (n = 0 to 5)

Register	EPEISHn	EPU Event Input Status Register Higher Side for Thread n		Address	See Table 10-4
Bit	Bit Name	R/W	Initial	Description	Remarks
7	EVS[15]	R/C	0	Event input status Read 0: The corresponding event does not exist Read 1: The corresponding event exists Write 0: No change Write 1: The bit is cleared When an event applicable to the bit of the EPEICHn register, which is set to 1, is input, the corresponding bit of this register is set. When the EVTCLR or EVTWAIT instruction, whose event number agrees with that of the input event, is executed, the corresponding bit of this register is cleared. Writing 1 clears the bit. And the EPEISCn register defines the bit. A set operation by event input has the lowest priority, if it conflicts with either of a clear operation by writing 1 to the bit, or a set operation by the EPEISCn register. Writing by the CPU takes precedence over processing by the EVTCLR or EVTWAIT instruction.	
6	EVS[14]	R/C	0		
5	EVS[13]	R/C	0		
4	EVS[12]	R/C	0		
3	EVS[11]	R/C	0		
2	EVS[10]	R/C	0		
1	EVS[9]	R C	0		
0	EVS[8]	R/C	0		

10.7.16. EPPSPn (EPU Prescaler Period Register for Thread n) (n = 0 to 5)

Register	EPPSPn	EPU Prescaler Period Register for Thread n	Address	See Table 10-4

Bit	Bit Name	R/W	Initial	Description	Remarks
7	PSP	R/W	0	The period of the prescaler counter The bit defines the period of the prescaler counter. The prescaler counter repeats a countdown, from the value defined by the EPPSPn register to 0, during the period from an issuance the TIMWAIT instruction to a completion of the countdown by the TIMWAIT instruction. The period of the dedicated timer counter can be calculated by the following equation: $\text{TIME Counter Period} = \frac{\text{PSP} + 1}{\text{EPU Clock Frequency}}$	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

10.7.17. EPEISCn (EPU Event Input Status Control Register for Thread n) (n = 0 to 5)

Register	EPEISCn	EPU Event Input Status Control Register for Thread n	Address	See Table 10-4

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	SEL	W	0	See the description of the SET bits, below.	
4		W	0		
3	SET[3]	W	0	Writing 1 to the bit can set the corresponding bit of the EPEISLn/EPEISHn register. The controllable status bits depend on which value to be written to the SEL bits. When SEL = 0b00 SET[3]: EPEISLn.EVS[3] SET[2]: EPEISLn.EVS[2] SET[1]: EPEISLn.EVS[1] SET[0]: EPEISLn.EVS[0] When SEL = 0b01 SET[3]: EPEISLn.EVS[7] SET[2]: EPEISLn.EVS[6] SET[1]: EPEISLn.EVS[5] SET[0]: EPEISLn.EVS[4] When SEL = 0b10 SET[3]: EPEISHn.EVS[3] SET[2]: EPEISHn.EVS[2] SET[1]: EPEISHn.EVS[1] SET[0]: EPEISHn.EVS[0] When SEL = 0b11 SET[3]: EPEISHn.EVS[7] SET[2]: EPEISHn.EVS[6] SET[1]: EPEISHn.EVS[5] SET[0]: EPEISHn.EVS[4]	
2	SET[2]	W	0		
1	SET[1]	W	0		
0	SET[0]	W	0		

10.8. Usage Notes and Restrictions

For the usage notes and restrictions on the conflicts between the EPU and CPU buses, see Section 5.7.

11. Interrupt Controller (INTC)

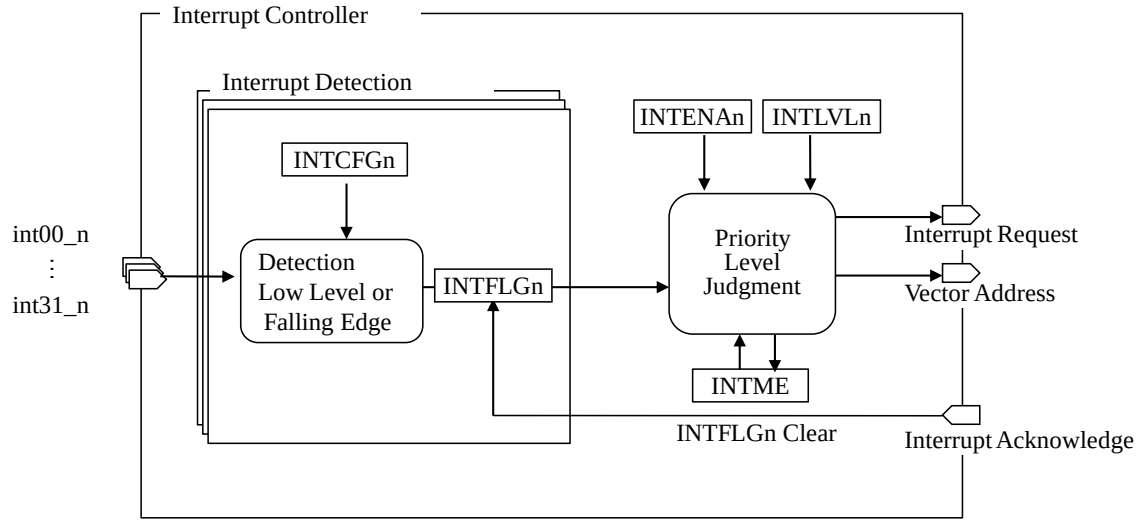
11.1. Overview

The interrupt controller (INTC) processes the interrupt requests from the peripheral functions, and requests the interrupt from the CPU.

Table 11-1 and Figure 11-1 show the INTC overview and the INTC block diagram, respectively. For the GPIO interrupt, see Section 7.

Table 11-1. INTC Functional Descriptions

Item	Description
Number of Source	32 sources
Interrupt Detection Type	Low level detection or falling edge detection (selectable per interrupt source) Low level detection must be selected in the LSI.
Interrupt Enable	Selectable per interrupt source
Interrupt Priority	High or low (selectable per interrupt source from 2 levels)
Vector Address	Fixed vector address is assigned per interrupt vector



Remarks
INTME: Interrupt Master Enable Signal
INTENAn: Interrupt Enable n Register
INTLVLn: Interrupt Level n Register
INTCFGn: Interrupt Configuration n Register
INTFLGn: Interrupt Flag n Register

Figure 11-1. INTC Block Diagram

11.2. Interrupt Vector

The INTC processes 32 interrupt sources. Table 11-2 lists the vector addresses and interrupt sources corresponding to the interrupt vectors. The interrupt handler (interrupt service routine) in small device C compiler (SDCC) is defined as follows.

```
void some_isr(void) __interrupt (5) __using (3)
{

}
```

"__interrupt (5)" means the interrupt service routine (ISR) corresponding to the interrupt vector number 5.
"__using (3)" means that the register bank is specified to 3. The SDCC automatically generates a vector table.

Table 11-2. Interrupt Vectors

Interrupt Vector No.	Vector Address	Interrupt Sources[(1)]	Default Priority	Remarks
0	0x0003	GPIO0	High	
1	0x000B	GPIO1		
2	0x0013	GPIO2		
3	0x001B	Reserved		
4	0x0023	LVD Interrupt		
5	0x002B	WDT Interrupt		
6	0x0033	Comparator0 Interrupt		
7	0x003B	Comparator1 Interrupt		
8	0x0043	Comparator2 Interrupt/Comparator4 Interrupt		
9	0x004B	Comparator3 Interrupt/Comparator5 Interrupt		
10	0x0053	ADC0 Interrupt		
11	0x005B	ADC1 Interrupt		
12	0x0063	Reserved		
13	0x006B	PWM0 Interrupt0		
14	0x0073	PWM0 Interrupt1		
15	0x007B	PWM1 Interrupt0		
16	0x0083	PWM1 Interrupt1		
17	0x008B	PWM2 Interrupt0		
18	0x0093	PWM2 Interrupt1		
19	0x009B	PWM3 Interrupt0		
20	0x00A3	PWM3 Interrupt1		
21	0x00AB	Timer0 Interrupt		
22	0x00B3	Timer1 Interrupt		
23	0x00BB	TinyDSP0 Interrupt		
24	0x00C3	TinyDSP1 Interrupt		
25	0x00CB	SPI Rx Interrupt		
26	0x00D3	SPI Tx/Timer2 Interrupt		
27	0x00DB	Tx or Rx Interrupt of I^2C		
28	0x00E3	SCID		
29	0x00EB	Tx or Rx Interrupt of UART		
30	0x00F3	EVC		
31	0x00FB	Flash Memory/Timer3	Low	

[(1)] The interrupt source that the 2 modules are connected with "/" operates as an interrupt request of corresponding vector number is generated when an interrupt signal is output from either of the 2 modules.

11.3. Register Descriptions

Table 11-3 lists the INTC registers. The INTC registers are assigned to the SFR area. Only the CPU can access the INTC registers. The DSAC and the EPU cannot access the INTC registers.

Table 11-3. List of Registers

Symbol	Name	Address	Initial Value
INTMST	Interrupt Master Control Register	0x9C	0x00
INTENA0	Interrupt Enable0 Register	0xA4	0x00
INTENA1	Interrupt Enable1 Register	0xA5	0x00
INTENA2	Interrupt Enable2 Register	0xA6	0x00
INTENA3	Interrupt Enable3 Register	0xA7	0x00
INTLVL0	Interrupt Level0 Register	0xAC	0x00
INTLVL1	Interrupt Level1 Register	0xAD	0x00
INTLVL2	Interrupt Level2 Register	0xAE	0x00
INTLVL3	Interrupt Level3 Register	0xAF	0x00
INTCFG0	Interrupt Configuration0 Register	0xB4	0x00
INTCFG1	Interrupt Configuration1 Register	0xB5	0x00
INTCFG2	Interrupt Configuration2 Register	0xB6	0x00
INTCFG3	Interrupt Configuration3 Register	0xB7	0x00
INTFLG0	Interrupt Flag0 Register	0xBC	0x00
INTFLG1	Interrupt Flag1 Register	0xBD	0x00
INTFLG2	Interrupt Flag2 Register	0xBE	0x00
INTFLG3	Interrupt Flag3 Register	0xBF	0x00

11.3.1. INTMST (Interrupt Master Control Register)

Register	INTMST	Interrupt Master Control Register	Address	0x9C

Bit	Bit Name	R/W	Initial	Description	Remarks
7	HIP	R	0	High-priority interrupt flag 0: A high-priority interrupt is not being executed 1: A high-priority interrupt is being executed	
6	LIP	R	0	Low-priority interrupt flag 0: A low-priority interrupt is not being executed 1: A low-priority interrupt is being executed, or being suspended due to a high-priority interrupt	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	INTME	R/W	0	Interrupt Master Enable 0: All interrupt requests are disabled 1: Interrupt requests are enabled	

11.3.2. INTENAn (Interrupt Enable n Register) (n = 0 to 3)

Register	INTENA0	Interrupt Enable0 Register	Address	0xA4
Register	INTENA1	Interrupt Enable1 Register	Address	0xA5
Register	INTENA2	Interrupt Enable2 Register	Address	0xA6
Register	INTENA3	Interrupt Enable3 Register	Address	0xA7

Bit	Bit Name	R/W	Initial	Description	Remarks
7	INTE7	R/W	0	Interrupt enable 0: The corresponding interrupt request is not accepted 1: The corresponding interrupt request is accepted The INTENAn.INTEx bit corresponds to an interrupt vector number (8 × n + x).	
6	INTE6	R/W	0		
5	INTE5	R/W	0		
4	INTE4	R/W	0		
3	INTE3	R/W	0		
2	INTE2	R/W	0		
1	INTE1	R/W	0		
0	INTE0	R/W	0		

11.3.3. INTLVLn (Interrupt Level n Register) (n = 0 to 3)

Register	INTLVL0	Interrupt Level0 Register	Address	0xAC
Register	INTLVL1	Interrupt Level1 Register	Address	0xAD
Register	INTLVL2	Interrupt Level2 Register	Address	0xAE
Register	INTLVL3	Interrupt Level3 Register	Address	0xAF

Bit	Bit Name	R/W	Initial	Description	Remarks
7	INTL7	R/W	0	Interrupt priority 0: Priority is set to low 1: Priority is set to high The INTLVLn.INTLx bit corresponds to an interrupt vector number (8 × n + x).	
6	INTL6	R/W	0		
5	INTL5	R/W	0		
4	INTL4	R/W	0		
3	INTL3	R/W	0		
2	INTL2	R/W	0		
1	INTL1	R/W	0		
0	INTL0	R/W	0		

11.3.4. INTCFGn (Interrupt Configuration n Register) (n = 0 to 3)

Register	INTCFG0	Interrupt Configuration0 Register	Address	0xB4
Register	INTCFG1	Interrupt Configuration1 Register	Address	0xB5
Register	INTCFG2	Interrupt Configuration2 Register	Address	0xB6
Register	INTCFG3	Interrupt Configuration3 Register	Address	0xB7

Bit	Bit Name	R/W	Initial	Description	Remarks
7	INTS7	R/W	0	Interrupt detection type 0: Low level detection is selected 1: Falling edge detection is selected The value must always be set to 0. The INTCFGn.INTSx bit corresponds to an interrupt vector number (8 × n + x).	
6	INTS6	R/W	0		
5	INTS5	R/W	0		
4	INTS4	R/W	0		
3	INTS3	R/W	0		
2	INTS2	R/W	0		
1	INTS1	R/W	0		
0	INTS0	R/W	0		

11.3.5. INTFLGn (Interrupt Flag n Register) (n = 0 to 3)

Although the interrupt detection type can be selected from the low level detection or the falling edge detection by the INTCFGn.INTSx bit, the low level detection must be set in the LSI. While the low level detection is selected (INTCFGn.INTSx = 0), the INTCFGn.INTSx bit indicates an interrupt request. To clear the INTFLGn.INTFx bit, it is necessary to clear the interrupt flag of the interrupt request generation source. If the falling edge detection is selected (INTCFGn.INTSx = 1), the INTFLGn.INTFx bit and interrupt request are cleared when 1 is written to the INTFLGn.INTFx bit.

Register	INTFLG0	Interrupt Flag0 Register	Address	0xBC
Register	INTFLG1	Interrupt Flag1 Register	Address	0xBD
Register	INTFLG2	Interrupt Flag2 Register	Address	0xBE
Register	INTFLG3	Interrupt Flag3 Register	Address	0xBF

Bit	Bit Name	R/W	Initial	Description	Remarks
7	INTF7	R/C	0	Interrupt flag Read 0: No interrupt is detected Read 1: An interrupt is detected Write 0: No change Write 1: The corresponding bit is cleared The INTFLGn.INTFx bit corresponds to an interrupt vector number (8 × n + x).	
6	INTF6	R/C	0		
5	INTF5	R/C	0		
4	INTF4	R/C	0		
3	INTF3	R/C	0		
2	INTF2	R/C	0		
1	INTF1	R/C	0		
0	INTF0	R/C	0		

11.4. Operational Descriptions

11.4.1. Initial Setting

The INTC initial setting procedure is as follows (see Figure 11-2):

(1) Set the priority (high or low) per interrupt source by the INTLVLn.INTLx bit. For the interrupt priority, see Section 11.4.3.
(2) Set the interrupt detection type (low level detection or falling edge detection) per interrupt source by the INTCFGn.INTSx bit. Always set the low level detection in the LSI.
(3) Set enable or disable of the interrupt per interrupt source by the INTENAn.INTEx bit.
(4) Set enable or disable of the interrupt master by the INTMST.INTME bit.

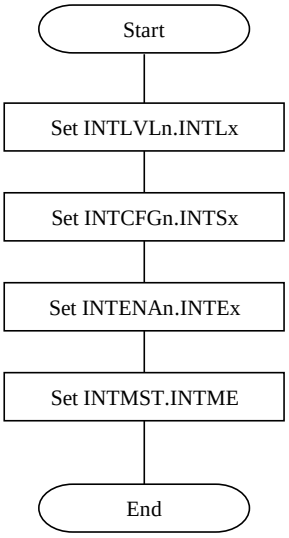


Figure 11-2. Procedure of INTC Initial Setting

11.4.2. Interrupt Flag

The INTFLGn register indicates that an interrupt request from the peripheral function is generated regardless of the INTENAn.INTEx bit setting.

Although the interrupt detection type can be selected from the low level detection or the falling edge detection by the INTCFGn.INTSx bit, the low level detection must be set in the LSI. While the low level detection is selected, the INTCFGn.INTSx bit indicates an interrupt request. To clear the INTFLGn.INTFx bit, it is necessary to clear the interrupt flag of interrupt request generation source. If the falling edge detection is selected, the INTFLGn.INTFx bit and the interrupt request are cleared when 1 is written to the INTFLGn.INTFx bit.

11.4.3. Interrupt Priority

The interrupt priority can be set high or low per interrupt. The interrupt is processed according to the priority as follows (see Figure 11-3):

- While the CPU processes a high priority interrupt, the CPU does not accept any interrupts including other high priority interrupts.
- When a high priority interrupt request is detected during a low priority interrupt processing by CPU, the CPU suspends the low priority interrupt processing, and accepts the high priority interrupt. Then, the CPU processes the high priority interrupt.
- After the high priority interrupt processing completes, the processing of the suspended low priority interrupt is restarted.
- While any interrupt is not accepted by the CPU, both low priority and high priority interrupts are accepted.
- When the interrupt requests of low priority and high priority occur simultaneously, the interrupt request of high priority is accepted.
- When the interrupt requests of the same priority occur simultaneously, the interrupt request of a smaller interrupt vector number is accepted.

To complete the ISR, clear the interrupt flag of interrupt source in the ISR. When another interrupt request is detected at the RETI instruction execution, the CPU returns an acknowledge indicating interrupt acceptance to the INTC, and accepts the subsequent interrupt request.

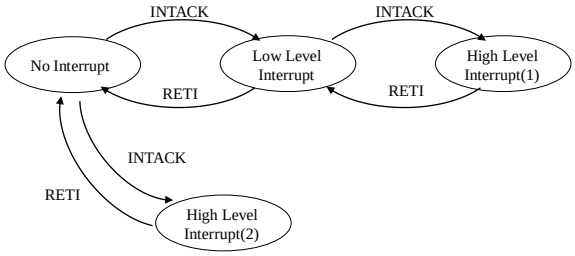


Figure 11-3. State Transition of Interrupt Priority Processing

The INTMST.HIP and INTMST.LIP bits indicate one of the following states.

- Processing the high priority and low priority interrupts
- Waiting for the acceptance of interrupt request
- Suspending

When the INTMST.HIP bit is 1, it indicates that the high priority interrupt is being processed. When the INTMST.LIP bit is 1, it indicates that the low priority interrupt is being processed or the low priority interrupt processing is suspended by the high priority interrupt processing.

Figure 11-4 shows the acceptance procedure of interrupt request in the INTC.

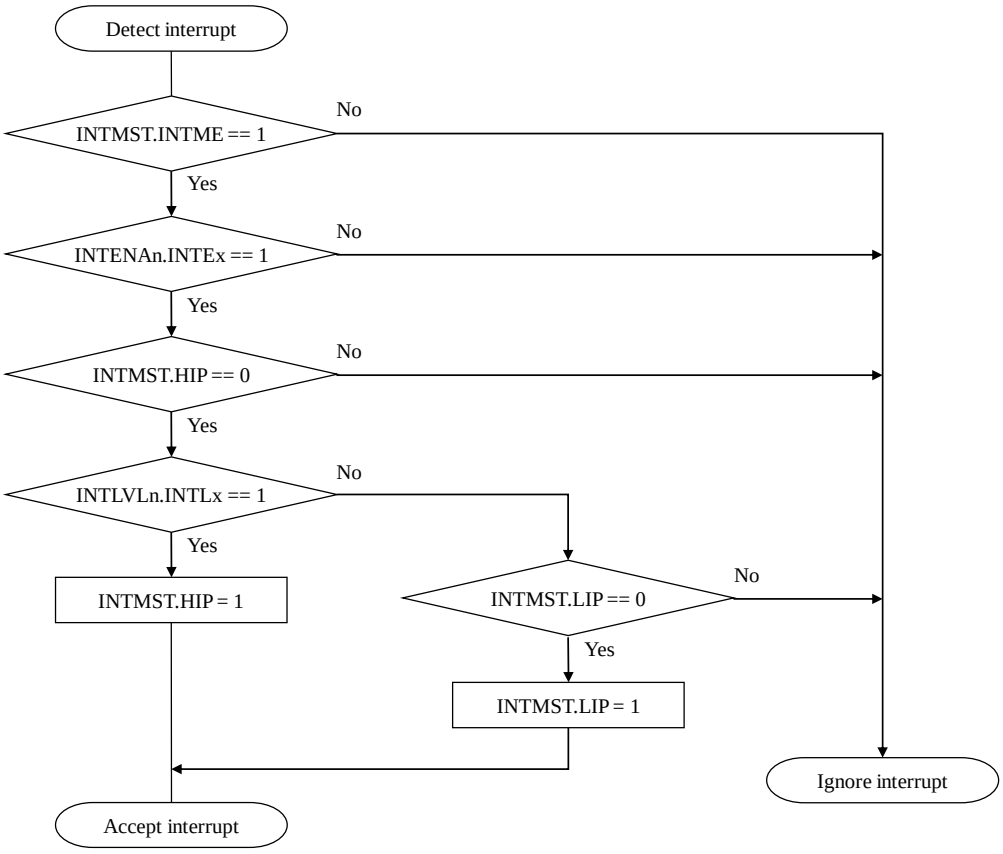


Figure 11-4. Acceptance Procedure of Interrupt Request

11.4.4. External Pin (GPIO) Interrupt

All GPIO pins are the interrupt inputs. The GPIO interrupt is defined by the registers in the GPIO. The interrupt of each pin is integrated in the unit of GPIOx (x = 0 to 2), the interrupt request is notified to the INTC. Figure 11-5 and Figure 11-6 show the logic diagram of GPIO interrupt generation and the generation timing of GPIO edge interrupt, respectively. For the detail of GPIO interrupt, see Section 7.

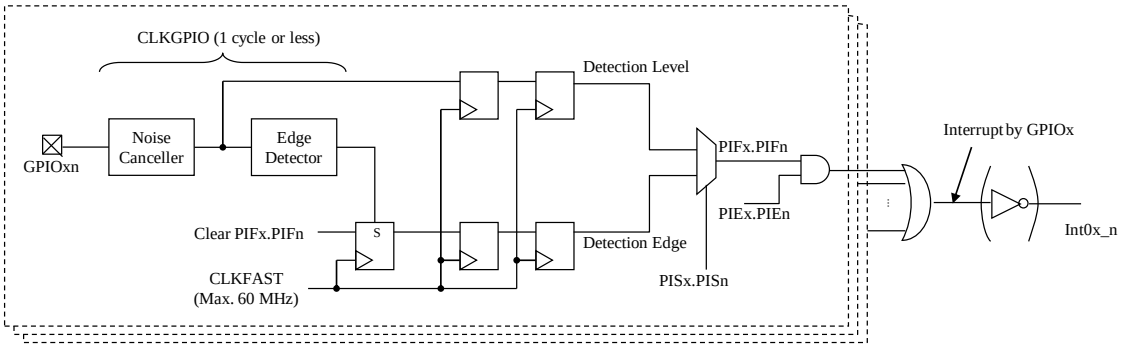


Figure 11-5. Logic Diagram of GPIO Interrupt Generation

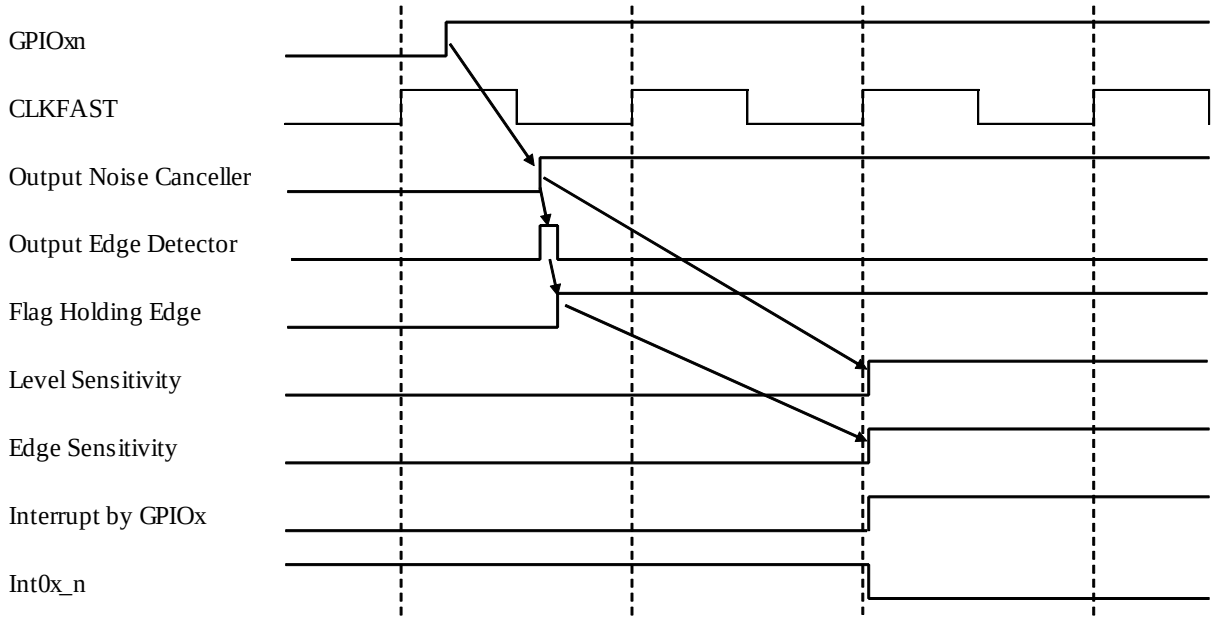


Figure 11-6. Generation Timing of GPIO Edge Interrupt

12. Direct SFR Access Controller (DSAC)

12.1. Overview

The direct SFR access controller (DSAC) can transfer the data directly between the SFRs without the CPU. The data transfer time can be greatly reduced by applying this function to the SFR of a peripheral function.

The DSAC cannot access the SFR related to the CPU or the INTC. When the DSAC reads these SFRs, the data becomes unstable. Writing from the DSAC to these SFRs is invalid. Figure 12-1 and Table 12-1 show the DSAC block diagram and the DSAC functional descriptions, respectively.

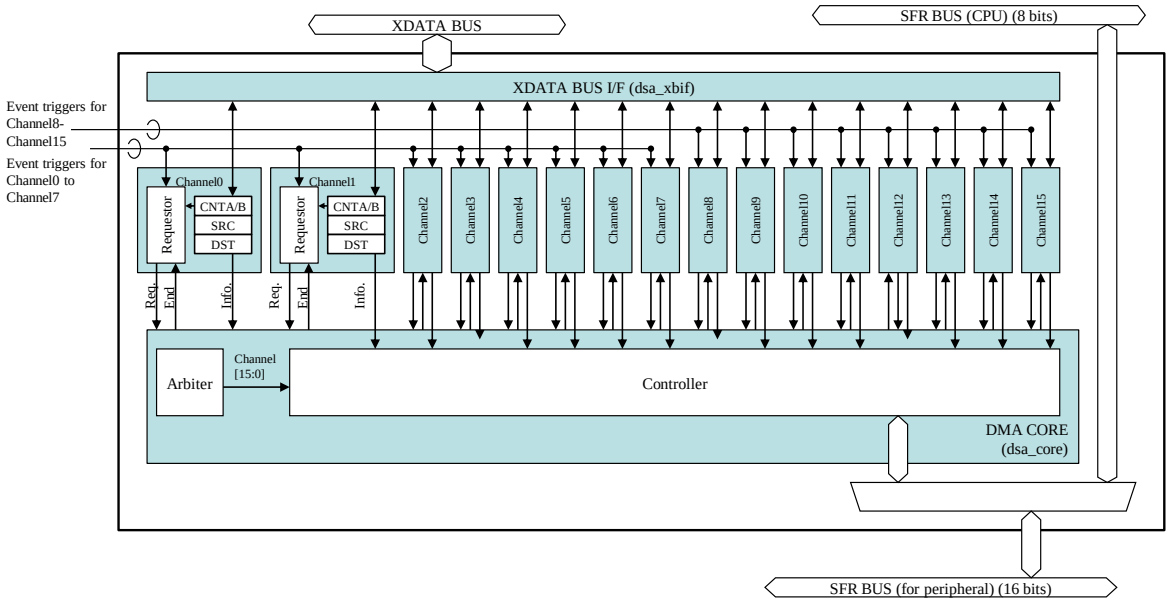


Figure 12-1. DSAC Block Diagram

Table 12-1. DSAC Functional Descriptions

Item	Description
Number of Channels	16 channels
Number of Transfer Request Events	32 events (max.) Selects one trigger event for each cannel by the corresponding register
Data Size	Transmits the selectable data (1 byte or 1 word) at once
Number of Data Transmissions	Selectable number of data transmissions per event (1 time, 2 times, 4 times, 8 times)
Address Mode	Fixed, +1, +8 Independently selectable increment modes for the source and destination addresses
Channel Priority	Channel0 > Channel1 > … > Channel14 > Channel15
SFR BUS Access Priority	DSAC > EPU > CPU
Transfer Mode	Cycle steal

12.2. Events

The DSAC is activated by the trigger event as shown in Table 12-2. The trigger event to activate the DSAC is defined by the DSACNTAn register and the event select register in the event controller (EVC). For the event selection, see the EVSELn register in Section 9.

Table 12-2. DSAC Events of Channel0 to Channel15

Event No.	Event Source	Trigger Event
0	Comparator0/1	Event generation
1	Comparator2/3	Event generation
2	Comparator4/5	Event generation
3	ADC Unit0 Group0	Event output from Group0 of ADC0
4	ADC Unit0 Group1	Event output from Group1 of ADC0
5	ADC Unit0 Group2	Event output from Group2 of ADC0
6	ADC Unit0 Group3	Event output from Group3 of ADC0
7	ADC Unit0 Group4	Event output from Group4 of ADC0
8	ADC Unit0 Group5	Event output from Group5 of ADC0
9	ADC Unit0 Group6	Event output from Group6 of ADC0
10	ADC Unit0 Group7	Event output from Group7 of ADC0
11	ADC Unit1 Group0	Event output from Group0 of ADC1
12	ADC Unit1 Group1	Event output from Group1 of ADC1
13	ADC Unit1 Group2	Event output from Group2 of ADC1
14	ADC Unit1 Group3	Event output from Group3 of ADC1
15	ADC Unit1 Group4	Event output from Group4 of ADC1
16	ADC Unit1 Group5	Event output from Group5 of ADC1
17	ADC Unit1 Group6	Event output from Group6 of ADC1
18	ADC Unit1 Group7	Event output from Group7 of ADC1
19	PWM0 Event0/1	Event generation
20	PWM1 Event0/1	Event generation
21	PWM2 Event0/1	Event generation
22	PWM3 Event0/1	Event generation
23	TinyDSP0 Event0	Event generation
24	TinyDSP0 Event1	Event generation
25	TinyDSP1 Event0	Event generation
26	TinyDSP1 Event1	Event generation
27	TMR0 Event A/B	Event generation
28	TMR1 Event A/B	Event generation
29	TMR2 Event A/B	Event generation
30	TMR3 Event A/B	Event generation
31	CPU Trigger	Activation by CPU trigger

12.3. Register Descriptions

Table 12-3. List of XDATA BUS Registers

Symbol	Name	Address	Initial Value
DSACNTA0	DSA Control A Channel0	0xF880	0x00
DSACNTB0	DSA Control B Channel0	0xF881	0x00
DSASRC0	DSA Source Address Channel0	0xF882	0x80
DSADST0	DSA Destination Address Channel0	0xF883	0x80
DSACNTA1	DSA Control A Channel1	0xF884	0x00
DSACNTB1	DSA Control B Channel1	0xF885	0x00
DSASRC1	DSA Source Address Channel1	0xF886	0x80
DSADST1	DSA Destination Address Channel1	0xF887	0x80
DSACNTA2	DSA Control A Channel2	0xF888	0x00
DSACNTB2	DSA Control B Channel2	0xF889	0x00
DSASRC2	DSA Source Address Channel2	0xF88A	0x80
DSADST2	DSA Destination Address Channel2	0xF88B	0x80
DSACNTA3	DSA Control A Channel3	0xF88C	0x00
DSACNTB3	DSA Control B Channel3	0xF88D	0x00
DSASRC3	DSA Source Address Channel3	0xF88E	0x80
DSADST3	DSA Destination Address Channel3	0xF88F	0x80
DSACNTA4	DSA Control A Channel4	0xF890	0x00
DSACNTB4	DSA Control B Channel4	0xF891	0x00
DSASRC4	DSA Source Address Channel4	0xF892	0x80
DSADST4	DSA Destination Address Channel4	0xF893	0x80
DSACNTA5	DSA Control A Channel5	0xF894	0x00
DSACNTB5	DSA Control B Channel5	0xF895	0x00
DSASRC5	DSA Source Address Channel5	0xF896	0x80
DSADST5	DSA Destination Address Channel5	0xF897	0x80
DSACNTA6	DSA Control A Channel6	0xF898	0x00
DSACNTB6	DSA Control B Channel6	0xF899	0x00
DSASRC6	DSA Source Address Channel6	0xF89A	0x80
DSADST6	DSA Destination Address Channel6	0xF89B	0x80
DSACNTA7	DSA Control A Channel7	0xF89C	0x00
DSACNTB7	DSA Control B Channel7	0xF89D	0x00
DSASRC7	DSA Source Address Channel7	0xF89E	0x80
DSADST7	DSA Destination Address Channel7	0xF89F	0x80
DSACNTA8	DSA Control A Channel8	0xF8A0	0x00
DSACNTB8	DSA Control B Channel8	0xF8A1	0x00

Symbol	Name	Address	Initial Value
DSASRC8	DSA Source Address Channel8	0xF8A2	0x80
DSADST8	DSA Destination Address Channel8	0xF8A3	0x80
DSACNTA9	DSA Control A Channel9	0xF8A4	0x00
DSACNTB9	DSA Control B Channel9	0xF8A5	0x00
DSASRC9	DSA Source Address Channel9	0xF8A6	0x80
DSADST9	DSA Destination Address Channel9	0xF8A7	0x80
DSACNTA10	DSA Control A Channel10	0xF8A8	0x00
DSACNTB10	DSA Control B Channel10	0xF8A9	0x00
DSASRC10	DSA Source Address Channel10	0xF8AA	0x80
DSADST10	DSA Destination Address Channel10	0xF8AB	0x80
DSACNTA11	DSA Control A Channel11	0xF8AC	0x00
DSACNTB11	DSA Control B Channel11	0xF8AD	0x00
DSASRC11	DSA Source Address Channel11	0xF8AE	0x80
DSADST11	DSA Destination Address Channel11	0xF8AF	0x80
DSACNTA12	DSA Control A Channel12	0xF8B0	0x00
DSACNTB12	DSA Control B Channel12	0xF8B1	0x00
DSASRC12	DSA Source Address Channel12	0xF8B2	0x80
DSADST12	DSA Destination Address Channel12	0xF8B3	0x80
DSACNTA13	DSA Control A Channel13	0xF8B4	0x00
DSACNTB13	DSA Control B Channel13	0xF8B5	0x00
DSASRC13	DSA Source Address Channel13	0xF8B6	0x80
DSADST13	DSA Destination Address Channel13	0xF8B7	0x80
SACNTA14	DSA Control A Channel14	0xF8B8	0x00
DSACNTB14	DSA Control B Channel14	0xF8B9	0x00
DSASRC14	DSA Source Address Channel14	0xF8BA	0x80
DSADST14	DSA Destination Address Channel14	0xF8BB	0x80
DSACNTA15	DSA Control A Channel15	0xF8BC	0x00
DSACNTB15	DSA Control B Channel15	0xF8BD	0x00
DSASRC15	DSA Source Address Channel 5	0xF8BE	0x80
DSADST15	DSA Destination Address Channel15	0xF8BF	0x80
DSATRG0	DSA Activation Trigger 0 for Channel0 to Channel7	0xF8F0	0x00
DSATRG1	DSA Activation Trigger 1 for Channel8 to Channel15	0xF8F1	0x00

12.3.1. DSACNTAn (DSA Control A Channel n) (n = 0 to 15)

Register	DSACNTA0	DSA Control A Channel0	Address	0xF880
Register	DSACNTA1	DSA Control A Channel1	Address	0xF884
Register	DSACNTA2	DSA Control A Channel2	Address	0xF888
Register	DSACNTA3	DSA Control A Channel3	Address	0xF88C
Register	DSACNTA4	DSA Control A Channel4	Address	0xF890
Register	DSACNTA5	DSA Control A Channel5	Address	0xF894
Register	DSACNTA6	DSA Control A Channel6	Address	0xF898
Register	DSACNTA7	DSA Control A Channel7	Address	0xF89C
Register	DSACNTA8	DSA Control A Channel8	Address	0xF8A0
Register	DSACNTA9	DSA Control A Channel9	Address	0xF8A4
Register	DSACNTA10	DSA Control A Channel10	Address	0xF8A8
Register	DSACNTA11	DSA Control A Channel11	Address	0xF8AC
Register	DSACNTA12	DSA Control A Channel12	Address	0xF8B0
Register	DSACNTA13	DSA Control A Channel13	Address	0xF8B4
Register	DSACNTA14	DSA Control A Channel14	Address	0xF8B8
Register	DSACNTA15	DSA Control A Channel15	Address	0xF8BC

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DSACHE	R/W	0	DSA channel enable 0: DSA channel is disabled 1: DSA channel is enabled	
6	DSATB	R/W	0	Setting of the number of DSA data transfers 00: 1 time 01: 2 times 10: 4 times 11: 8 times These bits set the number of Channel n data transfers per event. DSACNTBn.DSAWDACS bit determines the transfer data size.	
5		R/W	0		
4	DSAEV	R/W	0	DSA channel trigger 00000: Event0 is selected 00001: Event1 is selected ⋮ 11111: Event31 is selected	
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

12.3.2. DSACNTBn (DSA Control B Channel n) (n = 0 to 15)

Register	DSACNTB0	DSA Control B Channel0	Address	0xF881
Register	DSACNTB1	DSA Control B Channel1	Address	0xF885
Register	DSACNTB2	DSA Control B Channel2	Address	0xF889
Register	DSACNTB3	DSA Control B Channel3	Address	0xF88D
Register	DSACNTB4	DSA Control B Channel4	Address	0xF891
Register	DSACNTB5	DSA Control B Channel5	Address	0xF895
Register	DSACNTB6	DSA Control B Channel6	Address	0xF899
Register	DSACNTB7	DSA Control B Channel7	Address	0xF89D
Register	DSACNTB8	DSA Control B Channel8	Address	0xF8A1
Register	DSACNTB9	DSA Control B Channel9	Address	0xF8A5
Register	DSACNTB10	DSA Control B Channel10	Address	0xF8A9
Register	DSACNTB11	DSA Control B Channel11	Address	0xF8AD
Register	DSACNTB12	DSA Control B Channel12	Address	0xF8B1
Register	DSACNTB13	DSA Control B Channel13	Address	0xF8B5
Register	DSACNTB14	DSA Control B Channel14	Address	0xF8B9
Register	DSACNTB15	DSA Control B Channel15	Address	0xF8BD

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DSAWDACS	R/W	0	Transfer data size setting 0: 1 byte (8 bits) 1: 1 word (16 bits)	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	DSTINCMD	R/W	0	Destination address increment mode setting (Only when DSADSTC = 1, the DSTINCMD bit affects the operation.) 0: Destination address is incremented by 1 per transfer 1: Destination address is incremented by 8 per transfer	
2	SRCINCMD	R/W	0	Source address increment mode setting (Only when DSASRCC = 1, the SRCINCMD bit affects the operation.) 0: Source address is incremented by 1 per transfer 1: Destination address is incremented by 8 per transfer	
1	DSADSTC	R/W	0	Destination address setting 0: Fixed address 1: Incrementation according to the DSTINCMD bit setting	
0	DSASRCC	R/W	0	Source address setting 0: Fixed address 1: Incrementation according to the DSTINCMD bit setting	

12.3.3. DSASRCn (DSA Source Address Channel n) (n = 0 to 15)

Register	DSASRC0	DSA Source Address Channel0	Address	0xF882
Register	DSASRC1	DSA Source Address Channel1	Address	0xF886
Register	DSASRC2	DSA Source Address Channel2	Address	0xF88A
Register	DSASRC3	DSA Source Address Channel3	Address	0xF88E
Register	DSASRC4	DSA Source Address Channel4	Address	0xF892
Register	DSASRC5	DSA Source Address Channel5	Address	0xF896
Register	DSASRC6	DSA Source Address Channel6	Address	0xF89A
Register	DSASRC7	DSA Source Address Channel7	Address	0xF89D
Register	DSASRC8	DSA Source Address Channel8	Address	0xF8A2
Register	DSASRC9	DSA Source Address Channel9	Address	0xF8A6
Register	DSASRC10	DSA Source Address Channel10	Address	0xF8AA
Register	DSASRC11	DSA Source Address Channel11	Address	0xF8AE
Register	DSASRC12	DSA Source Address Channel12	Address	0xF8B2
Register	DSASRC13	DSA Source Address Channel13	Address	0xF8B6
Register	DSASRC14	DSA Source Address Channel14	Address	0xF8BA
Register	DSASRC15	DSA Source Address Channel15	Address	0xF8BE

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DSASA	R/W	1	SFR source address	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

12.3.4. DSADSTn (DSA Destination Address Channel n) (n = 0 to 15)

Register	DSADST0	DSA Destination Address Channel0	Address	0xF883
Register	DSADST1	DSA Destination Address Channel1	Address	0xF887
Register	DSADST2	DSA Destination Address Channel2	Address	0xF88B
Register	DSADST3	DSA Destination Address Channel3	Address	0xF88F
Register	DSADST4	DSA Destination Address Channel4	Address	0xF893
Register	DSADST5	DSA Destination Address Channel5	Address	0xF897
Register	DSADST6	DSA Destination Address Channel6	Address	0xF89B
Register	DSADST7	DSA Destination Address Channel7	Address	0xF89F
Register	DSADST8	DSA Destination Address Channel8	Address	0xF8A3
Register	DSADST9	DSA Destination Address Channel9	Address	0xF8A7
Register	DSADST10	DSA Destination Address Channel10	Address	0xF8AB
Register	DSADST11	DSA Destination Address Channel11	Address	0xF8AF
Register	DSADST12	DSA Destination Address Channel12	Address	0xF8B3
Register	DSADST13	DSA Destination Address Channel13	Address	0xF8B7
Register	DSADST14	DSA Destination Address Channel14	Address	0xF8BB
Register	DSADST15	DSA Destination Address Channel15	Address	0xF8BF

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DSADA	R/W	1	SFR destination address	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

12.3.5. DSATRGm (DSA Trigger m Channel0 to Channel7) (m = 0 to 1)

Register	DSATRG0	DSA Trigger m for Channel0 to Channel7	Address	0xF8F0

Bit	Bit Name	R/W	Initial	Description	Remarks
7	TRGCH7	R/W	0	Channel7 transfer trigger Read 0: Channel7 is in the idle state Read 1: Channel7 is in the transfer state, or is waiting for other channel transfers to be completed Write 0: No change Write 1: Channel7 activation	
6	TRGCH6	R/W	0	Channel6 transfer trigger Read 0: Channel6 is in the idle state Read 1: Channel6 is in the transfer state, or is waiting for other channel transfers to be completed Write 0: No change Write 1: Channel6 activation	
5	TRGCH5	R/W	0	Channel5 transfer trigger Read 0: Channel5 is in the idle state Read 1: Channel5 is in the transfer state, or is waiting for other channel transfers to be completed Write 0: No change Write 1: Channel5 activation	
4	TRGCH4	R/W	0	Channel4 transfer trigger Read 0: Channel4 is in the idle state Read 1: Channel4 is in the transfer state, or is waiting for other channel transfers to be completed Write 0: No change Write 1: Channel4 activation	
3	TRGCH3	R/W	0	Channel3 transfer trigger Read 0: Channel3 is in the idle state Read 1: Channel3 is in the transfer state, or is waiting for other channel transfers to be completed Write 0: No change Write 1: Channel3 activation	
2	TRGCH2	R/W	0	Channel2 transfer trigger Read 0: Channel2 is in the idle state Read 1: Channel2 is in the transfer state, or is waiting for other channel transfers to be completed Write 0: No change Write 1: Channel2 activation	
1	TRGCH1	R/W	0	Channel1 transfer trigger Read 0: Channel1 is in the idle state Read 1: Channel1 is in the transfer state, or is waiting for other channel transfers to be completed Write 0: No change Write 1: Channel1 activation	

Register	DSATRG0		DSA Trigger m for Channel0 to Channel7		Address	0xF8F0
Bit	Bit Name	R/W	Initial	Description		Remarks
0	TRGCH0	R/W	0	Channel0 transfer trigger Read 0: Channel0 is in the idle state Read 1: Channel0 is in the transfer state, or is waiting for other channel transfers to be completed Write 0: No change Write 1: Channel0 activation		

12.3.6. DSATRGm (DSA Trigger m Channel8 to Channel15) (m = 0 to 1)

Register	DSATRG1		DSA Trigger m for Channel8 to Channel15		Address	0xF8F1
Bit	Bit Name	R/W	Initial	Description		Remarks
7	TRGCH15	R/W	0	Channel15 transfer trigger Read 0: Channel15 is in the idle state Read 1: Channel15 is in the transfer state, or is waiting for other channel transfers to be completed Write 0: No change Write 1: Channel15 activation		
6	TRGCH14	R/W	0	Channel14 transfer trigger Read 0: Channel14 is in the idle state Read 1: Channel14 is in the transfer state, or is waiting for other channel transfers to be completed Write 0: No change Write 1: Channel14 activation		
5	TRGCH13	R/W	0	Channel13 transfer trigger Read 0: Channel13 is in the idle state Read 1: Channel13 is in the transfer state, or is waiting for other channel transfers to be completed Write 0: No change Write 1: Channel13 activation		
4	TRGCH12	R/W	0	Channel12 transfer trigger Read 0: Channel12 is in the idle state Read 1: Channel12 is in the transfer state, or is waiting for other channel transfers to be completed Write 0: No change Write 1: Channel12 activation		
3	TRGCH11	R/W	0	Channel11 transfer trigger Read 0: Channel11 is in the idle state Read 1: Channel11 is in the transfer state, or is waiting for other channel transfers to be completed Write 0: No change Write 1: Channel11 activation		

Register	DSATRG1		DSA Trigger m for Channel8 to Channel15		Address	0xF8F1
Bit	Bit Name	R/W	Initial	Description		Remarks
2	TRGCH10	R/W	0	Channel10 transfer trigger Read 0: Channel10 is in the idle state Read 1: Channel10 is in the transfer state, or is waiting for other channel transfers to be completed Write 0: No change Write 1: Channel10 activation		
1	TRGCH9	R/W	0	Channel9 transfer trigger Read 0: Channel9 is in the idle state Read 1: Channel9 is in the transfer state, or is waiting for other channel transfers to be completed Write 0: No change Write 1: Channel9 activation		
0	TRGCH8	R/W	0	Channel8 transfer trigger Read 0: Channel8is in the idle state Read 1: Channel8 is in the transfer state, or is waiting for other channel transfers to be completed Write 0: No change Write 1: Channel8 activation		

12.4. Operation

The DSAC is activated (started to transfer) by the trigger event as shown in Table 12-2. The trigger event to activate the DSAC is set by the DSACNTAn.DSAEV bit. The DSAC has 16 channels. The activation source of each channel is selected an event from 32 trigger events. The channel transfer operation is started when the trigger event defined by the DSACNTAn.DSAEV bit is detected. DSAC is also activated by writing 1 to the DSATRGm.TRGCHn bit. DSAC activation by this bit is separately from the setting of DSACNTAn.DSAEV bit.

The smaller the channel number, the higher the priority of the DSAC (i.e., Channel0 > Channel1 > … Channel15). When multiple channels are transferred at the same time, the transfer starts from the channel with the smallest number, and the channel with the large number waits until this transfer operation is completed. Even if a transfer trigger event is generated again on the same channel during transferring data in one channel, this event is ignored. The details are shown in Figure 12-2.

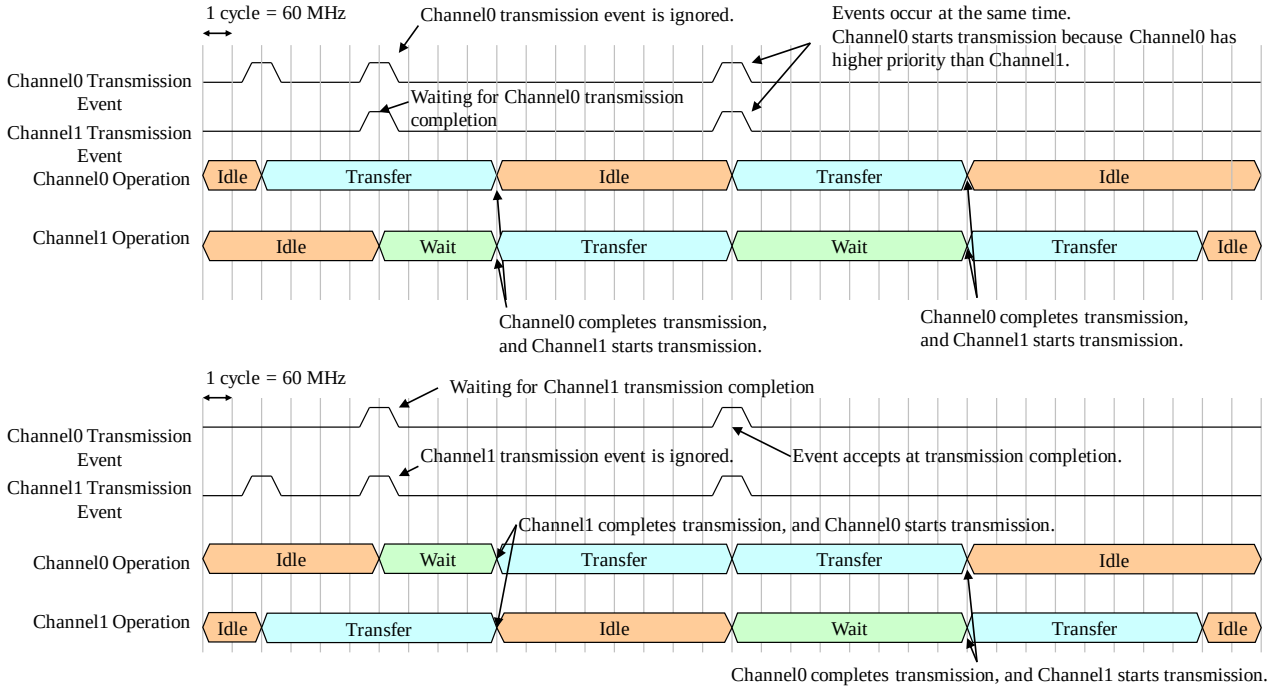


Figure 12-2. Event Trigger Priority

The source, destination, the number of data transfers, and address increment at transfer are determined by the following registers, respectively.

- DSASRCn: Source SFR address
- DSADSTn: Destination SFR address
- DSACNTAn: Number of data transfers per event (1 time, 2 times, 4 times, 8 times) determined by the DSATB bit
- DSACNTBn: Address increment at transfer (source/destination address is fixed, increment by 1, increment by 8)

The address is incremented according to the setting at each transfer, but the values of the DSASRCn and DSADSTn registers do not change. When the next event is accepted after the transfer, the transfer of the address defined by the DSASRCn and DSADSTn registers is started again.

The DSAC BUS access is processed with higher priority than other bus masters (CPU and EPU). When the accesses from the DSAC and other bus master are generated at the same time, the DSAC access is processed first. While the DSAC access is not generated, the access to the bus master that was waiting is processed. The details are shown in Figure 12-3.

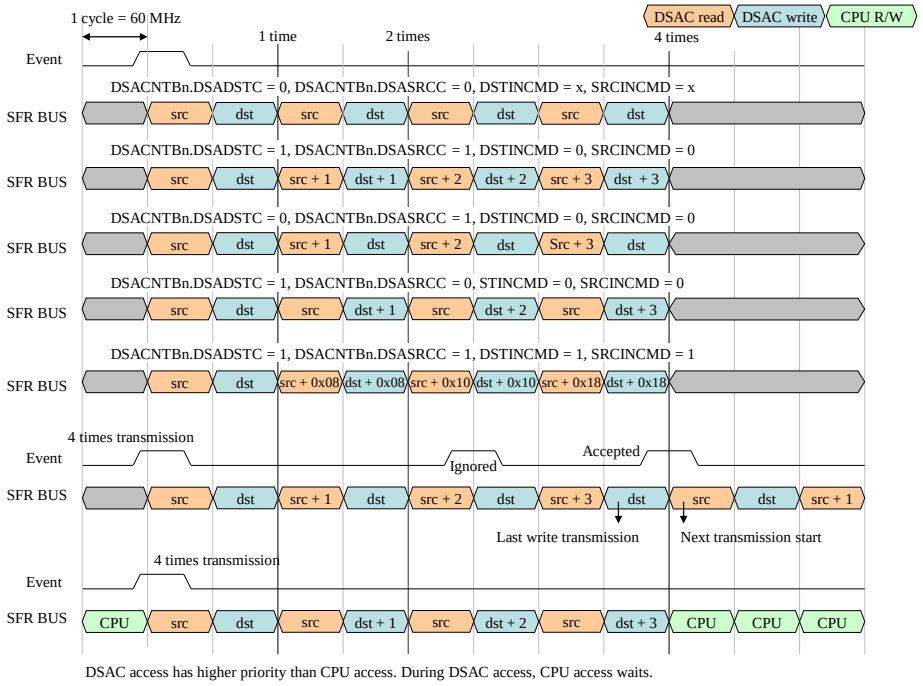


Figure 12-3. Example of Transfer Operation

Figure 12-4 shows the SFR BUS configuration. 8-bit width or 16-bit width register is connected to the SFR BUS. The SFR BUS of the CPU has 8 bits. The CPU is accessed one time with 8-bit width register. The 16-bit width registers of the modules such as PWM, TinyDSP, ADC, and DAC are accessed from the lower byte to the higher byte of the same address.

As for the method of accessing DSAC, there are also the following access methods using the BUS extended to 16 bits, in addition to the method of accessing two times with 8 bits.

- Method 1:
  DSAC transfer size: 8 bits (DSACNTBn.DSAWDACS = 0)
  SFR register width of source and destination: 8 bits
  The DSAC uses only the lower 8 bits of the SFR BUS.
- Method 2:
  DSAC transfer size: 8 bits (DSACNTBn.DSAWDACS = 0)
  SFR register width of source and destination: 16 bits
  The DSAC uses the lower 8 bits of the SFR BUS. The access method is the sames as the CPU (see Table 12-4). The 16-bit SFR register assigned to the same address is accessed in the order of lower byte to highr byte.
- Method 3: This setting is not recommended.
  DSAC transfer size: 16 bits (DSACNTBn.DSAWDACS = 1)
  SFR register width of source and destination: 8 bits
  The operation is almost the same as the method 1. When the DSAC reads the 8-bit SFR register with 16-bit transfer size, the lower 8 bits of the data read by the DSAC is the value of the read register, but the higher 8 bits become 0x00. When the DSAC writes the 8-bit SFR register with 16-bit transfer size, the lower 8 bits of the 16-bit write data output by the DSAC are actually written to the 8-bit SFR register, but the higher 8 bits are ignored.
- Method 4:
  DSAC transfer size: 16 bits (DSACNTBn.DSAWDACS = 1)
  SFR register width of source and destination: 16 bits
  The DSAC uses both higher and lower 8 bits of the SFR BUS to transfer 16-bit data.

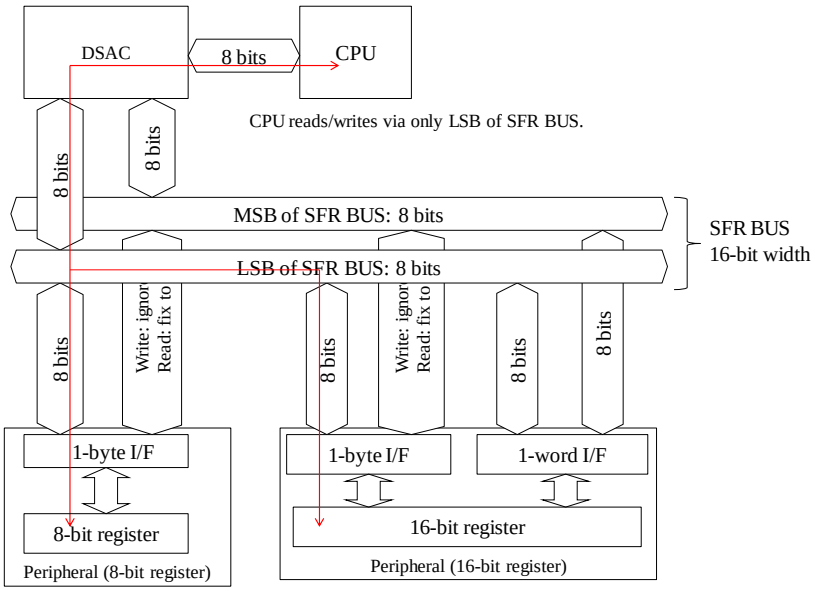


Figure 12-4. SFR BUS Configuration

Table 12-4. Access to 16-bit Register of SFR BUS

	1-byte Transmission (DSAWDACS = 0)	1-word Transmission (DSAWDACS = 1)
Read	DSAC 2nd access MSB_BUFFER 1st access 1st access MSB LSB 1-word register	DSAC 1-word register
Write	DSAC 1st access LSB_BUFFER 2nd access 2nd access MSB LSB 1-word register	DSAC 1-word register

12.5. Initial Setting Sequence

Figure 12-5 shows the initial setting sequence.

(1) Set the source start address.
(2) Set the destination start address.
(3) Set the transfer mode (1 word or 1 byte), the source address (fix or increment), and the destination address (fix or increment).
(4) Selet the increment units (1 or 8) of the source address and the destination address.
(5) Select the tranfer trigger event and the number of transfers.
(6) Enable the DSAC, and wait the trigger start.

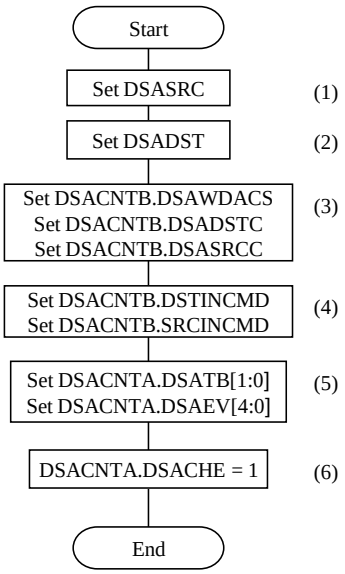


Figure 12-5. Initial Setting Sequence

12.6. Usage Notes and Restrictions

12.6.1. Invalidating the Channels

The following are the operational descriptions to disable Channel n.

Writing 0 to the DSACNTAn.DSACHE bit renders Channel n disabled. When 0 is written to the DSACNTAn.DSACHE bit during a transfer process, Channel n will be disabled after the transfer process completes. If 0 is written to the DSACNTAn.DSACHE bit even an unprocessed transfer request exists in Channel n, Channel n will be disabled after the corresponding transfer process completes. A transfer request received after writing 0 to the DSACNTAn.DSACHE bit will be ignored.

12.6.2. Restrictions when DSAC transfer data size is 8 bits

In case the following three conditions were met simultaneously, the DSAC will not transfer the data properly.

- Tranfer data size is 8 bits.
- Source and destination addresses are allocated to the same module.
- Source or destination register width is 16 bits.

13. Flash Memory Controller (FLC)

13.1. Overview

The flash memory controller (FLC) controls as follows: program fetch by the CPU, programming/erasing for the flash memory, and access security for the LSI.

Table 13-1. FLC Functional Descriptions

Item		Description
Flash Memory	Maximum Programming Cycles	20,000 cycles
	Main Block (Programming)	● Capacity: 32 KB (8 Kwords × 32 bits) ● Number of pages: 32 pages ● Number of rows: 8 rows per page ● Number of columns: 32 columns per row = 128 bytes per row (1 column = 1 word = 32 bits)
	Information Block	● Capacity: 1 KB (256 words × 32 bits) ● Number of pages: 1 page ● Number of rows: 8 rows per page ● Number of columns: 32 columns per row = 128 bytes per row (1 column = 1 word = 32 bits)
Program Fetch		● Fetch data width: 32 bits ● Instruction buffer: 32 bits × 2 lines ● Data buffer: 32 bits × 1 line ● Access mode: High-speed clock mode (2 cycles), low-speed clock mode (1 cycle) ● Prefetch: For instruction buffer only Generated by the fetch of the addresses, $4n + 2$ and $4n + 3$ (high-speed clock mode). Generated by the fetch of the address, $4n + 3$ (low-speed clock mode).
Flash Memory Operation Mode		● Programming: Row programming for main block or information block ● Erasing: Mass erase/page erase (erasing 1 page) for main block pages ● Reading: Row read for main block or information block ● Protection release: Decreasing of protection level ● Re-protection: Resetting of protection level ● Runtime flash memory operation: Programming, erasing, and reading for flash memory during program execution on flash memory by the CPU
Flash Memory Security Management		● Protection level: Level 1 or level 2 ● Protection code width: 32 bits ● Placed protection code in information block ● Programmable protection level by protection release and re-protection ● Protection level 2: Blocks flash memory accessing from programs on RAM; allows a part of register accesses from the OCD

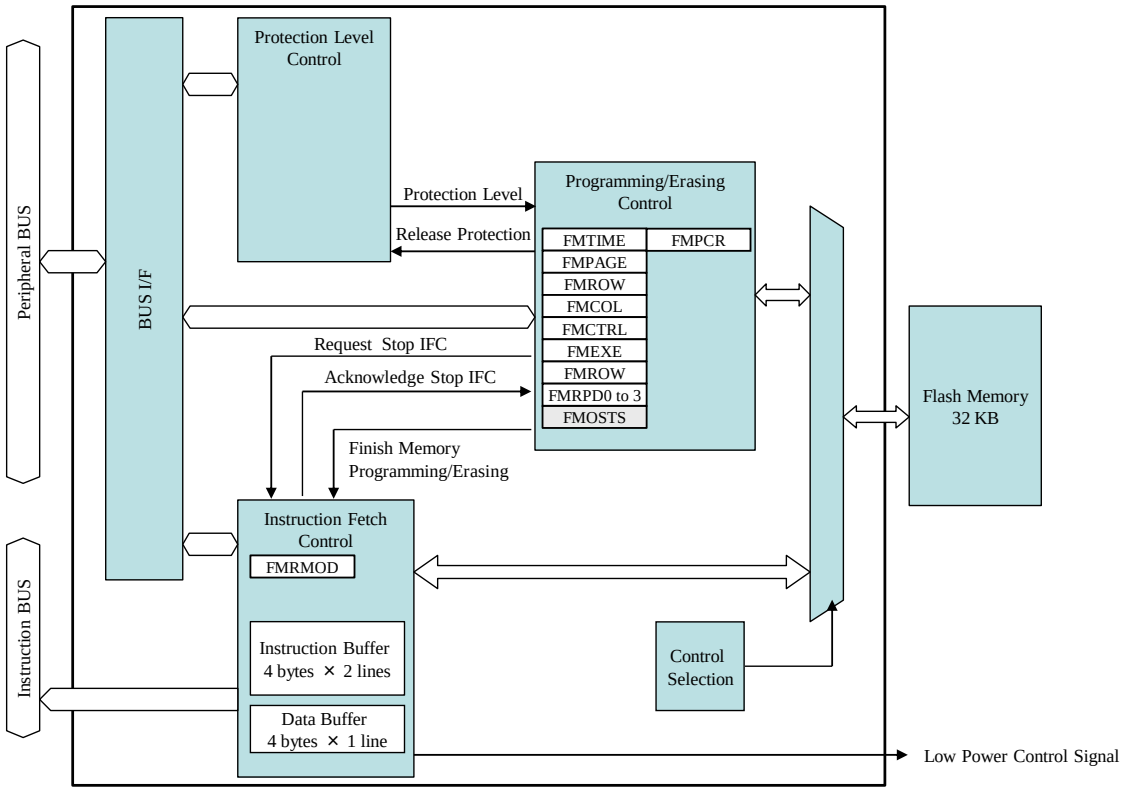


Figure 13-1. FLC Block Diagram

13.2. Register Descriptions

Table 13-2 lists the registers of FLC.

Table 13-2. List of Registers

Symbol	Name	Address	Initial Value
FMTIME	Flash Memory Control Time Register	0xFF00	0x0F
FMPAGE	Flash Memory Page Address Register	0xFF01	0x00
FMROW	Flash Memory Row Address Register	0xFF02	0x00
FMCOL	Flash Memory Column Address Register	0xFF03	0x00
FMCTRL	Flash Memory Control Register	0xFF04	0x00
FMEXE	Flash Memory Program Execution Register	0xFF05	0x00
FMRPD0	Flash Memory Row Program Data0 Register	0xFF10	0x00
FMRPD1	Flash Memory Row Program Data1 Register	0xFF11	0x00
FMRPD2	Flash Memory Row Program Data2 Register	0xFF12	0x00
FMRPD3	Flash Memory Row Program Data3 Register	0xFF13	0x00
FMRMOD	Flash Memory Read Mode Register	0xFF20	0x01
FMPCR	Flash Memory Program Control Register	0xFF23	0x00

13.2.1. FMTIME (Flash Memory Control Time Register)

Register	FMTIME		Flash Memory Control Time Register		Address	0xFF00
Bit	Bit Name	R/W	Initial	Description		Remarks
7	TIME	R/W	0	Flash memory control time A setting value of the counter for generating an internal timing signal of 1 µs. $\mathrm{FMTIME} = \frac{\mathrm{CLKFAST}}{1 \times 10^6} - 1$ Change the settings only when the FMEXE.FMEXE bit is 0.		
6		R/W	0			
5		R/W	0			
4		R/W	0			
3		R/W	1			
2		R/W	1			
1		R/W	1			
0		R/W	1			

13.2.2. FMPAGE (Flash Memory Page Address Register)

Register	FMPAGE		Flash Memory Page Address Register	Address	0xFF01
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	PAGE	R/W	0	Flash memory page address The bit specifies a page address in the following operation modes: row program, row read, and page erase. The bit is writable only when the FMEXE.FMEXE bit is 0.	
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

13.2.3. FMROW (Flash Memory Row Address Register)

When the protection level is 1, in the row read mode or row program mode for the information block, set FMROW.ROW[2] = 1. Setting the FMROW.ROW[2] bit to 0 allows the LSI to execute control sequence of the flash memory, but has no effect on the flash memory.

When the protection level is 2, set the FMROW.ROW bits to the ROW5, ROW6, or ROW7. Setting the FMROW.ROW bits to other rows allows the LSI to execute the control sequence of the flash memory, but has no effect on the flash memory.

Register	FMROW		Flash Memory Row Address Register	Address	0xFF02
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	ROW	R/W	0	Flash memory row address The bit specifies a row address in the following operation modes: row program and row read. The bit is writable only when the FMEXE.FMEXE bit is 0.	
1		R/W	0		
0		R/W	0		

13.2.4. FMCOL (Flash Memory Column Address Register)

Register	FMCOL		Flash Memory Column Address Register	Address	0xFF03
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	COL	R/W	0	Flash memory column address The bit specifies a column address in the following modes: row program and row read. At the completion of reading or writing one column, the bit is automatically incremented by 1 (except when COL = 31).	
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

13.2.5. FMCTRL (Flash Memory Control Register)

Register	FMCTRL		Flash Memory Control Register	Address	0xFF04
Bit	Bit Name	R/W	Initial	Description	Remarks
7	FMIF	R/C	0	Flash memory interrupt flag Read 0: Erasing/programming is not completed Read 1: Erasing/programming is completed Write 0: No change Write 1: The bit is cleared	
6	FMIE	R/W	0	Flash memory interrupt enable 0: Output of interrupt request is disabled 1: Output of interrupt request is enabled When FMIE = 1 and FMIF = 1, an interrupt request is output to the CPU.	
5	RFOMD	R/W	0	Runtime flash memory operation mode enable 0: Runtime flash memory operation is disabled 1: Runtime flash memory operation is enabled	
4	BWSEL	R/W	0	Bit width setting for writing to flash memory 0: 32 bits 1: 16 bits	
3	FMCMD	R/W	0	Flash memory operation mode 0000: Normal mode; instruction fetch by CPU is allowed 0100: Row read for the main block 0101: Row program for the main block 0110: Page erase for the main block 0111: Mass erase for the main block 1000: Row read for the information block 1001: Row program for the information block 1110: Re-protection 1111: Protection release Other than above: Setting prohibited The bit is writable only when the FMEXE.FMEXE bit is 0. Note that setting values differ according to the protection levels. For more details, see Table 13-3.	
2		R/W	0		
1		R/W	0		
0		R/W	0		

Table 13-3. Setting Values for Flash Memory Operation Modes

Mode	Setting Value	Level 1	Level 2
Normal Mode (Instruction/Data Fetch)	0b0000	Y	Y(2)
Row Read for Main Block	0b0100	Y	—
Row Program for Main Block	0b0101	Y	—
Page Erase for Main Block	0b0110	Y	—
Mass Erase for Main Block	0b0111	Y	—
Row Read for Information Block	0b1000	Y(1)	—
Row Program for Information Block	0b1001	Y(1)	—
Re-protection	0b1110	Y	—
Protection Release	0b1111	Y	Y

(1) Available to the user-releasable area (ROW4 to ROW7)
(2) Available only to the CPU operation and the OCD

13.2.6. FMEXE (Flash Memory Program Execution Register)

Register	FMEXE		Flash Memory Program Execution Register		Address	0xFF05
Bit	Bit Name	R/W	Initial	Description		Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.		
6	Reserved	R	0	The read value is 0. The write value must always be 0.		
5	Reserved	R	0	The read value is 0. The write value must always be 0.		
4	Reserved	R	0	The read value is 0. The write value must always be 0.		
3	Reserved	R	0	The read value is 0. The write value must always be 0.		
2	Reserved	R	0	The read value is 0. The write value must always be 0.		
1	Reserved	R	0	The read value is 0. The write value must always be 0.		
0	FMEXE	R/W	0	Execution of flash memory operation Read 0: Flash memory control is being stopped Read 1: Flash memory control is being executed Write 0: Flash memory control is cancelled Write 1: Flash memory control is started Writing 1 to the bit is prohibited when the FMCTRL.FMCMD bit is 0b0000.		

13.2.7. FMRPDn (Flash Memory Row Program Data n Register) (n = 0 to 3)

Register	FMRPD0	Flash Memory Row Program Data0 Register	Address	0xFF10
Register	FMRPD1	Flash Memory Row Program Data1 Register	Address	0xFF11
Register	FMRPD2	Flash Memory Row Program Data2 Register	Address	0xFF12
Register	FMRPD3	Flash Memory Row Program Data3 Register	Address	0xFF13

Bit	Bit Name	R/W	Initial	Description	Remarks
7	RPD	R/W	0	Flash memory data FMRPD0: Address 4n FMRPD1: Address 4n + 1 FMRPD2: Address 4n + 2 FMRPD3: Address 4n + 3 $n = 256 \times FMPAGE + 32 \times FMROW + FMCOL$ The bit specifies the data to be programmed in the row program mode. The read data is stored in the row read mode.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

13.2.8. FMRMOD (Flash Memory Read Mode Register)

Register	FMRMOD	Flash Memory Read Mode Register	Address	0xFF20

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	FAST	R/W	1	Fast clock read mode 0: Low-speed clock mode (1 cycle) 1: High-speed clock mode (2 cycles) The write value is reflected when the flash memory access is put into an idle state. After changing the bit, make sure that the setting value is readable. When setting the bit to the low-speed clock mode, specify an appropriate frequency for this mode beforehand.	

13.2.9. FMPCR (Flash Memory Program Control Register)

Register		FMPCR	Flash Memory Program Control Register		Address	0xFF23
Bit	Bit Name	R/W	Initial	Description		Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.		
6	Reserved	R	0	The read value is 0. The write value must always be 0.		
5	Reserved	R	0	The read value is 0. The write value must always be 0.		
4	PWAITS	R/C	0	Program wait status Read 0: Not waiting for the next program data Read 1: Waiting for the next program data Write 0: No change Write 1: The bit is cleared Writing 1 to the bit is enabled when the bit is waiting for the column data to be ready, or when the column writing is completed. The bit is automatically cleared by setting the FMRPD3 register during the column data wait, or by writing 0 to the FMEXE.FMEXE bit.		
3	Reserved	R	0	The read value is 0. The write value must always be 0.		
2	Reserved	R	0	The read value is 0. The write value must always be 0.		
1	Reserved	R	0	The read value is 0. The write value must always be 0.		
0	PWAITIE	R/W	0	Program wait interrupt enable 0: Output of interrupt request is disabled 1: Output of interrupt request is enabled The bit controls the interrupt requests of the PWAITS bit.		

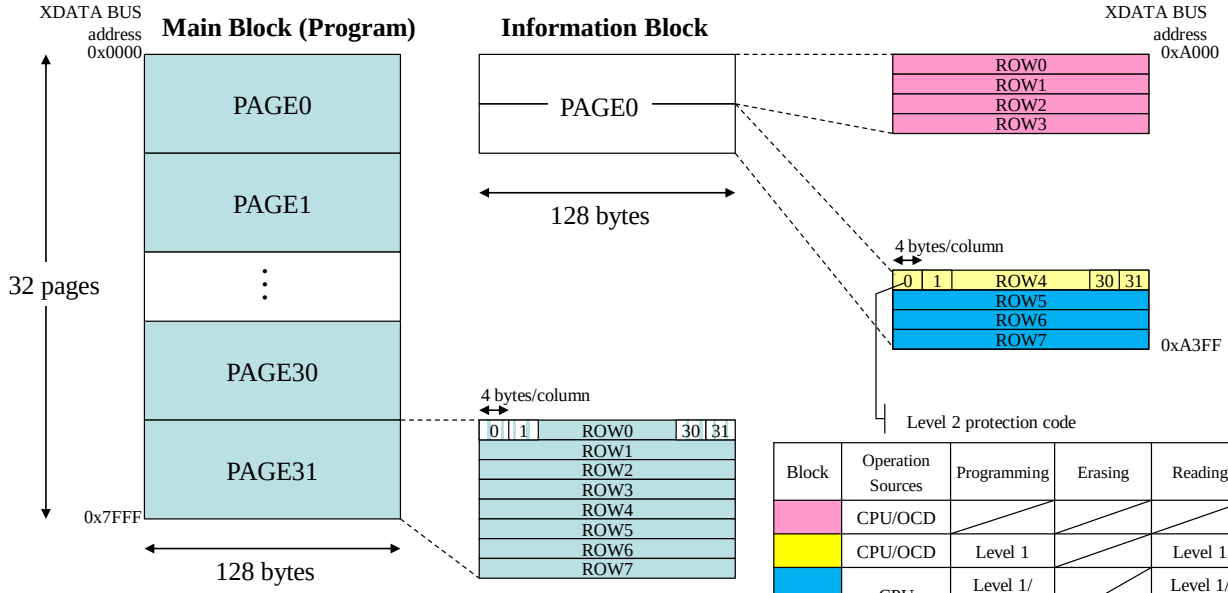
13.3. Flash Memory

Figure 13-2 shows the map of the flash memory. The flash memory has 32-KB main block and 1-KB information block. The main block has 32 pages. This configuration is 8 rows per page, 32 columns per row, and 4 bytes per column. The information block has 1 page. This configuration is 8 rows per page, 32 columns per row, and 4 bytes per column.

The CPU programs are stored in the main block. The protection code is stored in the information block. In addition, user data can be placed in the ROW5 to the ROW7 in the information block.

The protection level 1 allows the programming and erasing for the main block.

The protection level 2 allows the programming and erasing for the main block from the operating programs on the flash memory only, and does not allow the erasing for the information block. In addition, it allows the programming for the ROW5 to the ROW7, and does not allow the programming for the ROW0 to the ROW4. Write the protection code of the protection level 2 to the address 0xA200 (the COLUMN0 of the ROW4 in the information block).



Block	Operation Sources	Programming	Erasing	Reading
	CPU/OCD			
	CPU/OCD	Level 1		Level 1
	CPU	Level 1/ Level 2		Level 1/ Level 2
	OCD	Level 1		Level 1
	CPU	Level 1/ Level 2	Level 1/ Level 2	Level 1/ Level 2
	OCD	Level 1	Level 1	Level 1

Figure 13-2. Flash Memory Map

13.4. Operation

13.4.1. Instruction Fetch

The instruction fetch controller (IFC) controls instruction fetches to the CPU from the flash memory. The IFC has the instruction buffer (IBUF) that is 4 bytes × 2 lines. The instruction fetch operation of the CPU is as follows:

- When the IBUF has the instruction code of the adress requested by the CPU:
  The instruction code stored in the IBUF is fetched without wait cycles.
- When the IBUF does not have the instruction code of the adress requested by the CPU:
  The IFC fetches 1 column (4 bytes) of the flash memory. Then, the IFC returns necessary 1 byte to the CPU, and stores the fetched 1-column (4-byte) instruction code in the IBUF. The CPU waits until the flash memory accessing finishes.

In addition, the IFC has the data buffer (DBUF) that is 1 byte × 1 line. When the CPU reads the constant data from the flash memory by the MOVX or MOVC instruction, the reading operation is as follows:

- When the DBUF has the data of the adress requested by the CPU:
  The data stored in the DBUF is read without wait cycles.
- When the DBUF does not have the data of the adress requested by the CPU:
  The IFC fetches 1 column (4 bytes) of the flash memory. Then, the IFC returns necessary 1 byte to the CPU, and stores the fetched 1-column (4-byte) data in the DBUF. The CPU waits until the flash memory accessing finishes.

The IFC has programmable access modes to the flash memory, which are the high-speed and low-speed clock modes. The access mode to the flash memory is defined by the FMRMOD register. In the high-speed clock mode, the access cycle to the flash memory is 2 cycles. In the low-speed clock mode, the access cycle to the flash memory is 1 cycle. When an operation frequency of the CPU is more than 30 MHz, it is required to set to the high-speed clock mode. When the operation frequency of the CPU is 30 MHz or less, the low-speed clock mode can be used. The low-speed clock mode achieves an efficient operation because the wait cycles of the CPU are reduced.

The IBUF has a prefetch function to reduce the wait cycles at accessing the flash memory. In the high-speed clock mode, the prefetching starts after the CPU fetches the instruction of the address $4n + 2$ or $4n + 3$ ($n \geq 0$). In the low-speed clock mode, the prefetching starts after the CPU fetches the instruction of the address $4n + 3$ ($n \geq 0$). When the CPU fetches the prefetched code, the CPU executes the instruction without wait cycles. When the prefetched instruction is not executed by the instruction such as the JMP, the CPU waits while an instruction is fetched again from the flash memory.

13.4.2. Flash Memory Operation Mode

13.4.2.1. Mass Erase

The mass erase mode is the mode for erasing the all pages of the main block. To use the mass erase mode, set FMCTRL.FMCMD = 0b0111. In addition, to start the mass erase, set FMEXE.FMEXE = 1. During the mass erase execution, the FMEXE.FMEXE bit is kept to 1 that indicates the erasure is being executed. When the erasure completes, the FMEXE.FMEXE bit is cleared, and the FMCTRL.FMIF bit is set to 1. In addition, for outputting an interrupt request to the CPU, set FMCTRL.FMIE = 1. The FMCTRL.FMIF bit should be cleared in the interrupt routine.

Figure 13-3 shows the mass erase operation sequence. In addition, a part of the operation sequence is described below.

(1) Set FMCTRL.FMCMD = 0b0111 (mass erase mode in the main block).
(2) Wait for the erasure to complete. When the erasure completes, the FMCTRL.FMIF bit is set to 1. In addition, the FMEXE.FMEXE bit is automatically cleared. When FMCTRL.FMIE = 1 and FMCTRL.FMIF = 1, an interrupt request is output to the CPU.
(3) To clear the FMCTRL.FMIF bit, set FMCTRL.FMIF = 1.

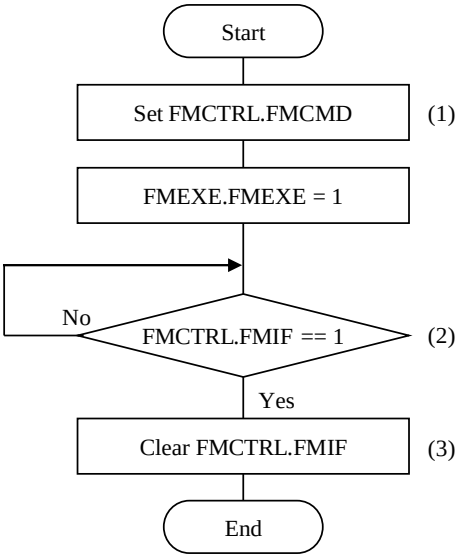


Figure 13-3. Mass Erase Operation Sequence

13.4.2.2. Page Erase

The page erase mode is the mode for erasing the specified one page of the main block. To use the page erase mode, set FMCTRL.FMCMD = 0b0110, and set the erased page to the FMPAGE register. Then, to start the page erase, set FMEXE.FMEXE = 1. During the page erase execution, the FMEXE.FMEXE bit is kept to 1 that indicates the erasure is being executed. When the erasure completes, the FMEXE.FMEXE bit is cleared, and the FMCTRL.FMIF bit is set to 1. In addition, for outputting an interrupt request to the CPU, set FMCTRL.FMIE = 1. The FMCTRL.FMIF bit should be cleared in the interrupt routine.

Figure 13-4 shows the page erase operation sequence. In addition, a part of the operation sequence is described below.

(1) Set FMCTRL.FMCMD = 0b0110 (page erase mode in the main block).
(2) Wait for the erasure to complete. When the erasure completes, the FMCTRL.FMIF bit is set to 1. In addition, the FMEXE.FMEXE bit is automatically cleared. When FMCTRL.FMIE =1 and FMCTRL.FMIF =1, an interrupt request is output to the CPU.
(3) To clear the FMCTRL.FMIF bit, set FMCTRL.FMIF = 1.

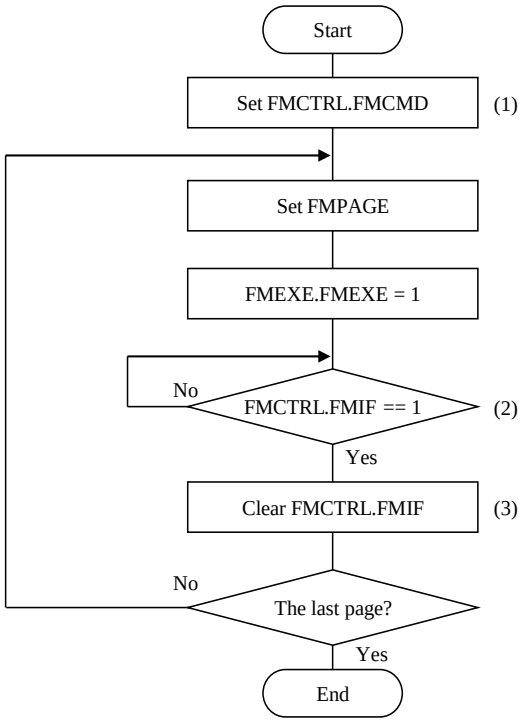


Figure 13-4. Page Erase Operation Sequence

13.4.2.3. Row Program

The row program mode has following 2 modes to program (i.e., to write) data to the specified address of the information block or the main block:

- Main block program mode: FMCTRL.FMCMD = 0b0101
- Information block program mode: FMCTRL.FMCMD = 0b1001

The programmed page, the programmed row, and programming start column are defined by the FMPAGE, FMROW, and FMCOL registers, respectively. To start the row program, set FMEXE.FMEXE = 1. Set the data to program to the FMRPD0 to FMRPD3 registers. When the FMRPD3 register is set (i.e., this setting is the programming trigger), the programming is performed to the specified current column. When the programming of the column completes, the FMCOL register is automatically incremented.

When the programming of the COLUMN31 completes, the FMEXE.FMEXE bit is cleared, and the FMCTRL.FMIF bit is set to 1. In this time, the FMCOL register is not updated. In addition, for outputting an interrupt request to the CPU, set FMCTRL.FMIE = 1. The FMCTRL.FMIF bit should be cleared in the interrupt routine.

To finish the row program midway, write the data to the FMRPD3 register and wait 46 µs, then clear the FMEXE.FMEXE bit. When the FMEXE.FMEXE bit becomes 0, the row program is finished, and the FMCTRL.FMIF bit is set to 1.

The flash memory on the LSI can be programmed by the half of one column (higher 2-byte unit or lower 2-byte unit). The programming in 2-byte unit reduces the programming period. For using the programming in 2-byte unit, set FMCTRL.BWSEL = 1. To program to higher 2 bytes in a column, set the writing values to the FMRPD0 and FMRPD1 registers, and set 0xFF to the FMRPD2 and FMRPD3 registers. On the other hand, to program to lower 2 bytes in a column, set 0xFF to the FMRPD0 and FMRPD1 registers, and set the writing values to the FMRPD2 and FMRPD3 registers.

When the column of one row is continuously programmed when FMCTRL.BWSEL = 1, note that the FMEXE.FMEXE bit is not automatically cleared at the completion of the COLUMN31 programming. Thus, to finish the programming, write the data to the FMRPD3 register and wait 23 µs, then clear the FMEXE.FMEXE bit.

In the row program mode, complete the programming (i.e., clear the FMEXE.FMEXE bit) within 8 ms after the programming is started.

Figure 13-5 shows the row program operation sequence. In addition, a part of the operation sequence is described below.

(1) Set the FMCTRL.FMCMD bits to 0b0101 (main block program mode) or 0b1001 (information block program mode).
(2) Set the bit width for writing to the flash memory, which is defined by the FMCTRL.BWSEL bit.
(3) Set the writing data to the flash memory. To start the programing operation, set the FMRPD3 register. When the FMEXE.FMEXE bit is set to 1 after writing the data to the FMRPD3 register, note that the programming operation is started after 17 µs.
(4) The period for writing completion is 46 µs. Wait 46 µs after the FMRPD3 register setting, and then, set the next data to the FMRPD3 register. The FMRPD0 to FMRPD2 registers can be set again while waiting. Since the accessing interval using the OCD is very long, need less attention for waiting period.
(5) When the writing to the last column (COLUMN31) completes, the FMEXE.FMEXE bit is automatically cleared.
(6) The FMCTRL.FMIF bit is automatically set to 1. When FMCTRL.FMIE = 1 and FMCTRL.FMIF = 1, an interrupt request is output to the CPU.
(7) To clear the FMCTRL.FMIF bit, set FMCTRL.FMIF = 1.

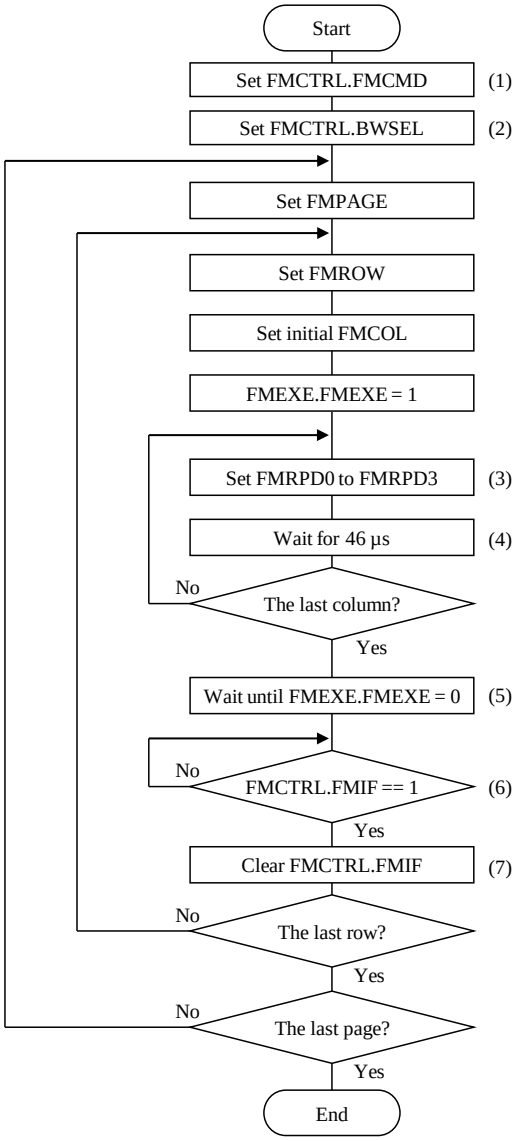


Figure 13-5. Row Program Operation Sequence

The FMPCR.PWAITS bit is cleared when the column programming starts, and is set to 1 when the column programming completes. To start the next column programming, be sure to check that FMPCR.PWAITS = 1, and then, set the FMRPD3 register. In addition, to set the next operation after the FMRPD3 register setting, be sure to check that FMPCR.PWAITS = 0. Since the FMPCR.PWAITS bit is not automatically cleared after the programming to the COLUMN31 completes, be sure to clear the FMPCR.PWAITS bit.

Figure 13-6 shows the row program operation sequence using the FMPCR.PWAITS bit. In addition, a part of the operation sequence is described below.

(1) Set the FMCTRL.FMCMD bits to 0b0101 (main block program mode) or 0b1001 (information block program mode).
(2) Set the bit width for writing to the flash memory, which is defined by the FMCTRL.BWSEL bit.
(3) Set the writing data to the flash memory. To start the programing operation, set the FMRPD3 register. When the FMEXE.FMEXE bit is set to 1 after writing the data to the FMRPD3 register, note that the programming operation is started after 17 µs.
(4) Check that FMPCR.PWAITS = 0.
(5) Wait for the completion of the writing to the column (FMPCR.PWAITS = 1). After checking the completion, set the next data to the FMRPD0 to FMRPD3 registers.
(6) When the writing to the last column (COLUMN31) completes, the FMEXE.FMEXE bit is automatically cleared.

(7) The FMCTRL.FMIF bit is automatically set to 1. When FMCTRL.FMIE = 1 and FMCTRL.FMIF = 1, an interrupt request is output to the CPU.
(8) To clear the FMCTRL.FMIF bit, set FMCTRL.FMIF = 1.
(9) To clear the FMPCR.PWAITS bit, set FMPCR.PWAITS = 1.

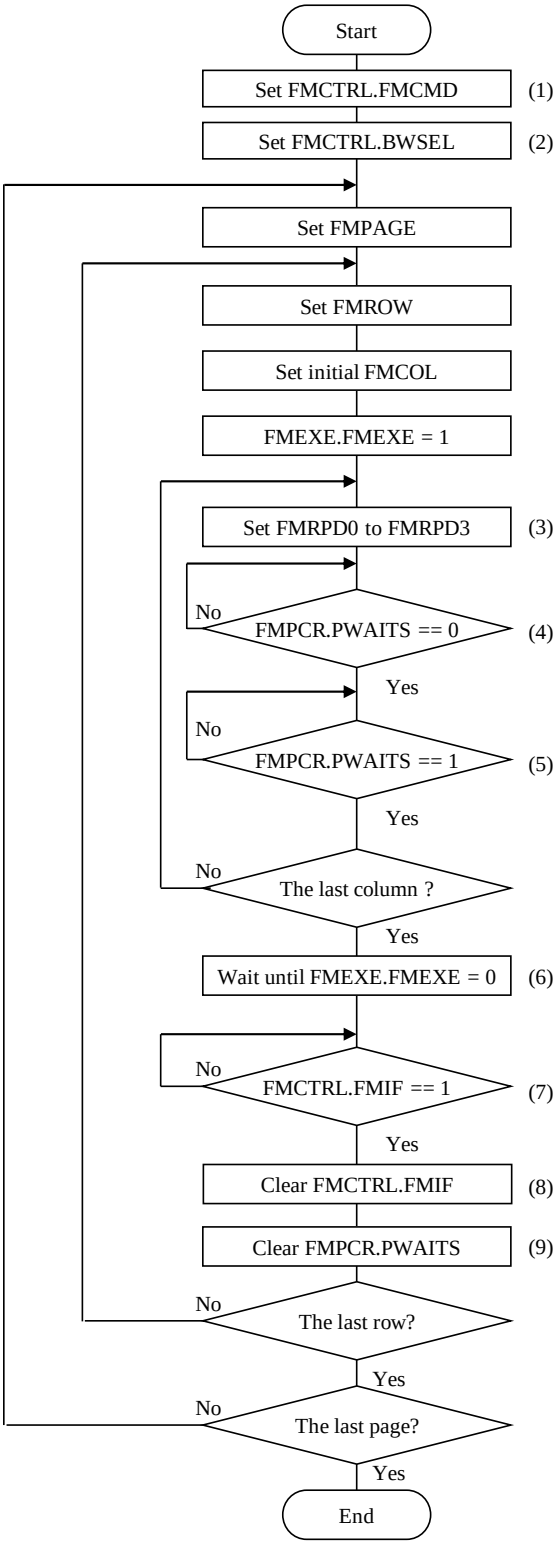


Figure 13-6. Row Program Operation Sequence Using FMPCR.PWAITS Bit

13.4.2.4. Row Read

The row read mode has following 2 modes to read data from the specified address of the information block or the main block:

- Main block read mode: FMCTRL.FMCMD = 0b0100
- Information block read mode: FMCTRL.FMCMD = 0b1000

The read page, the read row, and reading start column are defined by the FMPAGE, FMROW, and FMCOL registers, respectively. To start the row read, set FMEXE.FMEXE = 1. The read data is stored in the FMRPD0 to FMRPD3 registers. 6 cycles of the CLKFAST are required from setting the FMEXE.FMEXE bit to 1 to storing the read data to the FMRPD0 to FMRPD3 registers; wait with the instruction such as NOP.

When the reading of the FMRPD3 register completes, the FMCOL register is automatically incremented. 3 cycles of the CLKFAST are required from completing the FMRPD3 register read to storing the next read data to the FMRPD0 to FMRPD3 registers; wait with the instruction such as NOP.

When the reading from the COLUMN31 completes, the FMEXE.FMEXE bit is cleared, and the FMCTRL.FMIF bit is set to 1. In this time, the FMCOL register is not updated. In addition, for outputting an interrupt request to the CPU, set FMCTRL.FMIE = 1. The FMCTRL.FMIF bit should be cleared in the interrupt routine.

To finish the row read midway, clear the FMEXE.FMEXE bit. When the FMEXE.FMEXE bit becomes 0, the reading of the row data is finished, and the FMCTRL.FMIF bit is set to 1.

Figure 13-7 shows the row read operation sequence. In addition, a part of the operation sequence is described below.

(1) Set the FMCTRL.FMCMD bits to 0b0100 (main block read mode) or 0b1000 (information block read mode).
(2) To start the reading operation, set FMEXE.FMEXE = 1.
(3) When the reading of the FMRPD3 register is completed, the reading operation of the next column starts.
(4) When the reading from the last column (COLUMN31) completes, the FMEXE.FMEXE bit is automatically cleared.
(5) The FMCTRL.FMIF bit is automatically set to 1. When FMCTRL.FMIE = 1 and FMCTRL.FMIF = 1, an interrupt request is output to the CPU.
(6) To clear the FMCTRL.FMIF bit, set FMCTRL.FMIF = 1.

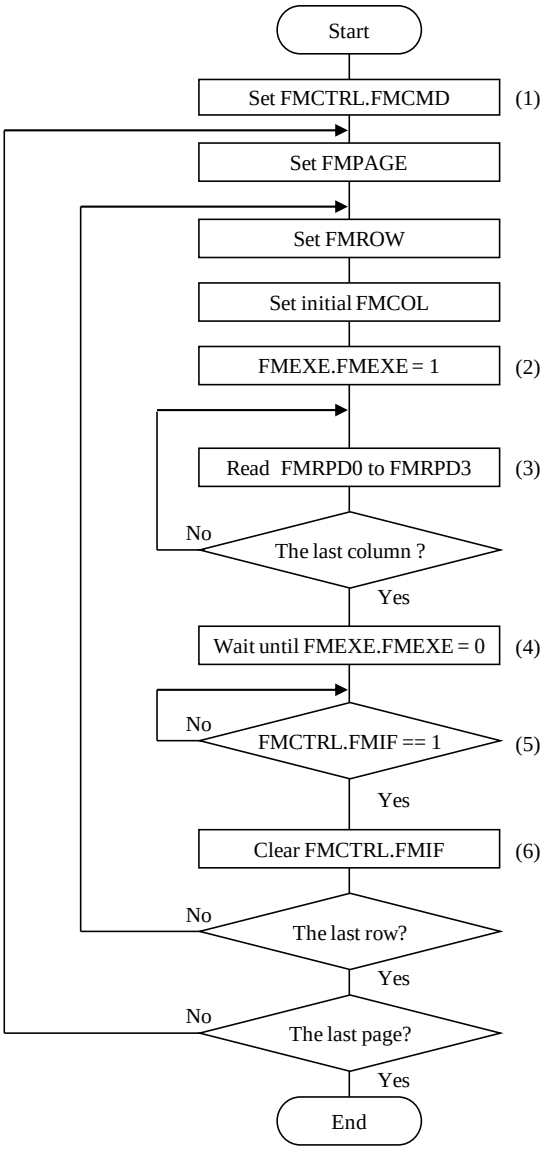


Figure 13-7. Row Read Operation Sequence

13.4.2.5. Protection Release

The protection release mode is the mode for decreasing the current protection level. The protection code that is set to enable the flash memory protection is necessary to down the protection level.

To use the protection release mode, set FMCTRL.FMCMD = 0b1111, and set the protection code to the FMRPD0 to FMRPD3 registers. Then, to execute the protection release, set the FMEXE.FMEXE = 1. When the protection release completes, the FMEXE.FMEXE bit is cleared, and the FMCTRL.FMIF bit is set to 1. In addition, for outputting an interrupt request to the CPU, set FMCTRL.FMIE = 1. The FMCTRL.FMIF bit should be cleared in the interrupt routine.

When the protection code written to the FMRPD0 to FMRPD3 registers matches the protection code written in the flash memory, the protection level is decreased from level 2 to level 1. Thus, resources in the LSI can be accessed using the OCD because the protection level is temporarily level 1.

Figure 13-8 shows the protection release operation sequence. In addition, a part of the operation sequence is described below.

(1) Set FMCTRL.FMCMD = 0b1111 (protection release).
(2) Set the protection code of level 2 to the FMRPD0 to FMRPD3 registers.
(3) To start the protection release, set FMEXE.FMEXE =1.
(4) Wait for the protection release to complete. When the protection release completes, the FMCTRL.FMIF bit is set to 1. When FMCTRL.FMIE =1 and FMCTRL.FMIF =1, an interrupt request is output to the CPU.
(5) To clear the FMCTRL.FMIF bit, set FMCTRL.FMIF = 1.

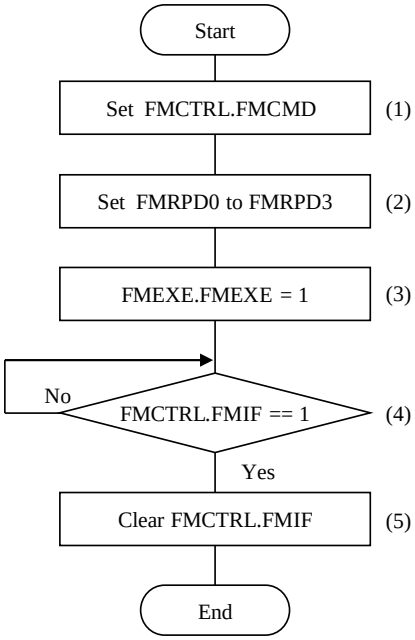


Figure 13-8. Protection Release Operation Sequence

13.4.2.6. Re-protection

The re-protection function is the mode for resetting the level that is decreased by the protection release mode. The protection level after re-protection is determined with or without the protection code of level 2 that is written in the flash memory.

To use the re-protection mode, set FMCTRL.FMCMD = 0b1110 and FMEXE.FMEXE = 1. When the re-protection completes, the FMEXE.FMEXE bit is cleared, and the FMCTRL.FMIF bit is set to 1. In addition, for outputting an interrupt request to the CPU, set FMCTRL.FMIE = 1. The FMCTRL.FMIF bit should be cleared in the interrupt routine.

Figure 13-9 shows the re-protection operation sequence. In addition, a part of the operation sequence is described below.

(1) Set FMCTRL.FMCMD = 0b1110 (re-protection).
(2) To start the re-protection, set FMEXE.FMEXE =1.
(3) Wait for the re-protection to complete. When the re-protection completes, the FMCTRL.FMIF bit is set to 1. When FMCTRL.FMIE =1 and FMCTRL.FMIF =1, an interrupt request is output to the CPU.
(4) To clear the FMCTRL.FMIF bit, set FMCTRL.FMIF = 1.

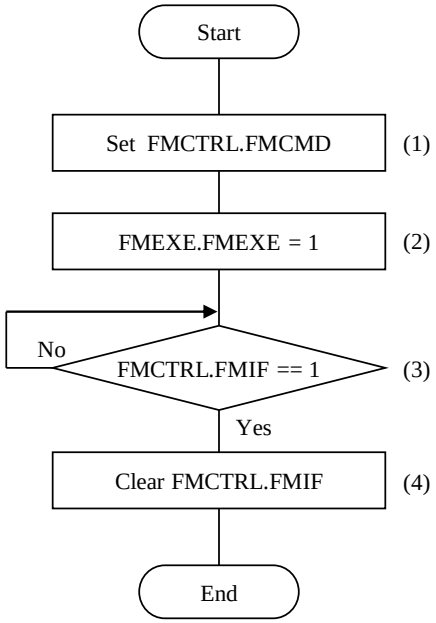


Figure 13-9. Re-protection Operation Sequence

13.5. Protection Level Control

It is required to control the accesses to the flash memory such as programming, erasing, and reading by the protection level control to protect programs from malicious attacks. In addition, this control limits to access resources in the LSI from the OCD.

- **Protection Level 1**
  The following instructions are blocked:
  - Mass erase of the information and main blocks in the flash memory
  - Erasing of the information block
  - Programming/Reading of ROW0 to ROW3 in the information block

- **Protection Level 2**
  - Blocks the flash memory accessing from programs on RAM.
  - Allows to program to/read from only ROW5 to ROW7 of the information block in the CPU operation (user controls).
  - Allows the execution of only the protection release mode from the OCD. In addition, accesses such as an OCD instruction and a data fetch are allowed for only following registers: DEVER, FMCTRL, and FMEXE.

Table 13-4 shows the relationship between protection levels and flash memory controls.

Table 13-4. Relationship between Protection Levels and Flash Memory Controls

Protection Level		Information Block		Main Block				Protection Release	Re-protection	Instruction/ Data Fetch
		Programming	Reading	Mass Erase	Erasing	Programming	Reading			
Level 1	Program on RAM (1)	Yes ROW4 to ROW7	Yes ROW4 to ROW7	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	CPU Operation									
	OCD									
Level 2	Program on RAM (1)	No	No	No	No	No	No	No	No	No
	CPU Operation	Yes ROW5 to ROW7	Yes ROW5 to ROW7	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	OCD	No	No	No	No	No	No	Yes	No	(2)

(1) Including the operand reading by a MOVC instruction.
(2) Allows reading from/writing to the FMCTRL and FMEXE registers. Accessing other than these registers is not allowed.

13.6. Runtime Flash Memory Operation

The LSI achieves programming, erasing, and reading of the flash memory during program execution on flash memory (runtime flash memory operation). This operation is used to write system log to the flash memory. While the flash memory is accessed by the runtime flash memory operation, the CPU is waited. When the runtime flash memory operation finishes, the CPU is released from the waiting, and restarts the program execution. The entries stored in the instruction buffer and the data buffer become invalid at the runtime flash memory operation start.

To enable the runtime flash memory operation, set FMCTRL.RFOMD = 1. The reading/writing in the runtime flash memory operation is executed the specified one column only. Note that the FMCOL register is not automatically incremented when this reading/writing operation completes.

Figure 13-10 shows the runtime flash memory operation sequence for one column writing. In addition, a part of the operation sequence is described below.

(1) Set the FMCTRL.FMCMD bits to 0b0101 (main block program mode) or 0b1001 (information block program mode).
(2) Set the bit width for writing to the flash memory, which is defined by the FMCTRL.BWSEL bit.
(3) Set FMCTRL.RFOMD = 1.
(4) Set writing data to flash memory to the FMRPD0 to FMRPD3 registers. Then, set FMEXE.FMEXE = 1.
(5) When the writing completes, the FMEXE.FMEXE bit is automatically cleared.
(6) The FMCTRL.FMIF bit is automatically set to 1. When FMCTRL.FMIE = 1 and FMCTRL.FMIF = 1, an interrupt request is output to the CPU.
(7) To clear the FMCTRL.FMIF bit, set FMCTRL.FMIF = 1.

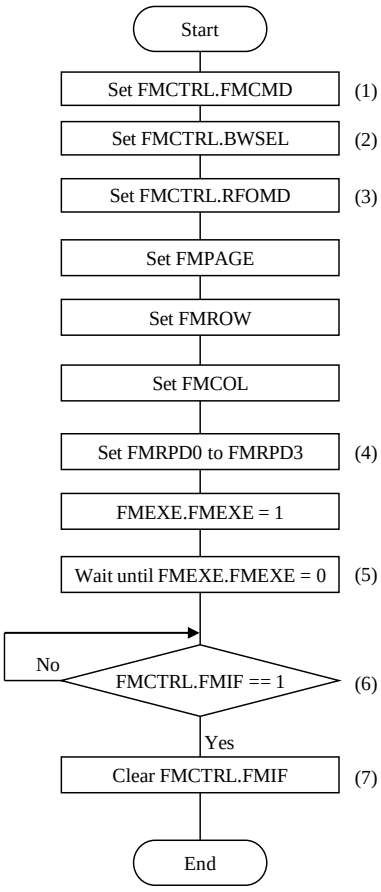


Figure 13-10. Runtime Flash Memory Operation Sequence (One Column Writing)

Figure 13-11 shows the runtime flash memory operation sequence for one column reading. In addition, a part of the operation sequence is described below.

(1) Set the FMCTRL.FMCMD bits to 0b0100 (main block read mode) or 0b1000 (information block read mode).
(2) Set FMCTRL.RFOMD = 1.
(3) To start the reading operation, set FMEXE.FMEXE = 1.
(4) Read from the FMRPD0 to FMRPD3 registers.
(5) When the reading completes, the FMEXE.FMEXE bit is automatically cleared.
(6) The FMCTRL.FMIF bit is automatically set to 1. When FMCTRL.FMIE = 1 and FMCTRL.FMIF = 1, an interrupt request is output to the CPU.
(7) To clear the FMCTRL.FMIF bit, set FMCTRL.FMIF = 1.

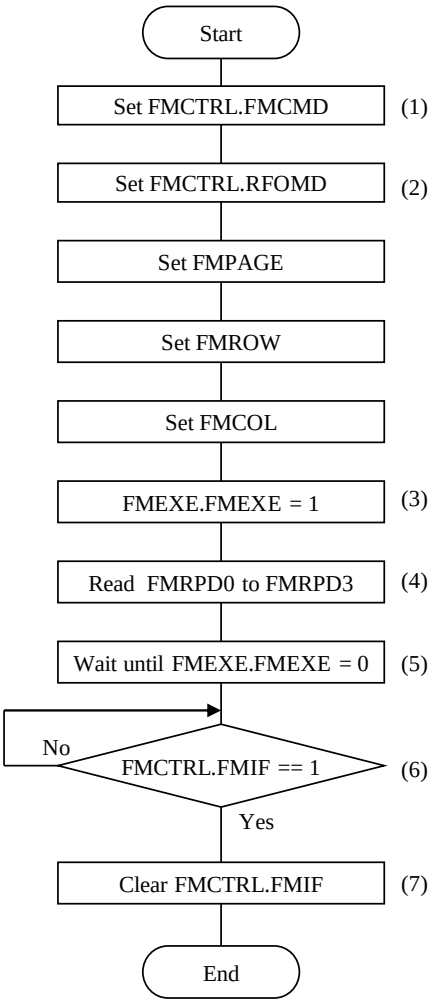


Figure 13-11. Runtime Flash Memory Operation Sequence (One Column Reading)

13.7. Usage Notes and Restrictions

13.7.1. Transition to Low Power Consumption Mode

For making a transition to the low power consumption mode, a NOP instruction must be inserted immediately after the write to the LPCTRL register. For more details, see Section 4.

13.7.2. Row Program Time

Set the program time of one row (i.e., a time during which the FMEXE.FMEXE bit is held at 1) to below 4 ms.

13.7.3. Usage Notes on Erasing or Programming (Writing) Immediately after Re-protection

Erasing or programming (writing) immediately after re-protection of the flash memory normally starts after a wait period. The duration of this wait period depends on the value* of the FMTIME register. For details on the FMTIME register setting, see Section 13.2.1. Even after a wait period, erasing and programming (writing) to the flash memory can be successfully executed. If not re-executing the re-protection, no wait period occurs before subsequent erasing or programming (writing) is performed.

The LSI can perform erasing and programming (writing) to the flash memory without protection release when operated according to programs in the flash memory itself (see Table 13-4). Consequently, for the operations executed by the programs in the flash memory, the LSI does not require any protection release or re-protection on the flash memory.

* If CLKFAST = 60 MHz and FMTIME = 59, the wait period is 32 ms.

14. TinyDSP

14.1. Overview

The TinyDSP is a dedicated processing unit for calculating a digital filter. The CPU and the TinyDSP can process each operation at the same time because the TinyDSP is separately from the CPU. The LSI has 2 TinyDSPs and these can be operated simultaneously.

The TinyDSP is calculated based on 16-bit fixed-point method. The program sequence can be composed of the simple instructions such as multiplication, division, multiply-accumulate calculation, shift calculation, move, jump, and minimum/maximum saturation. Each TinyDSP has sixteen 16-bit data registers (for storing coefficients, temporary data, and input/output data), eight 16-bit constant registers (for storing constants), and one 36-bit accumulator. Also, the TinyDSP supports the hardware division to improve the system performance.

The calculation sequence start is controlled by writing to the data registers from any of the units such as CPU, EPU, or DSAC. This calculation scheme can be configured by user.

For example, the user can configure the entire scheme to execute all operations without the CPU. The DSAC can transfer the data to the TinyDSP by the trigger that is the event generated in the LSI (such as AD conversion completion). Thus, the TinyDSP trigger can be transmitted from the internal hardware. When the calculation sequence is completed, the TinyDSP can generate the event that becomes an activation trigger for to the DSAC. The DSAC that receives the event can transfer the calculation result of the TinyDSP to the high-resolution PWM as a PWM duty cycle. As described above, the operations, such as from the trigger sending to the TinyDSP to the transferring of a calculation result, can be completely independent from the CPU.

Table 14-1. TinyDSP Functional Descriptions

Item	Description
Number of Units	2 units
Operation	16-bit fixed-point
Program Memory	48 steps (independent per unit)
Data Memory	Sixteen 16-bit data registers Eight 16-bit constant registers One 36-bit accumulator
Instruction	Multiplication, division, multiply-accumulate calculation, shift calculation, move, jump, and minimum/maximum saturation
Hardware Divider	Integrated
Sequence Control	Controlled by internal events
Event Output	Output at any time (with some exceptions)
Performance	3P2Z IIR: 10 cycles

14.2. Block Diagram

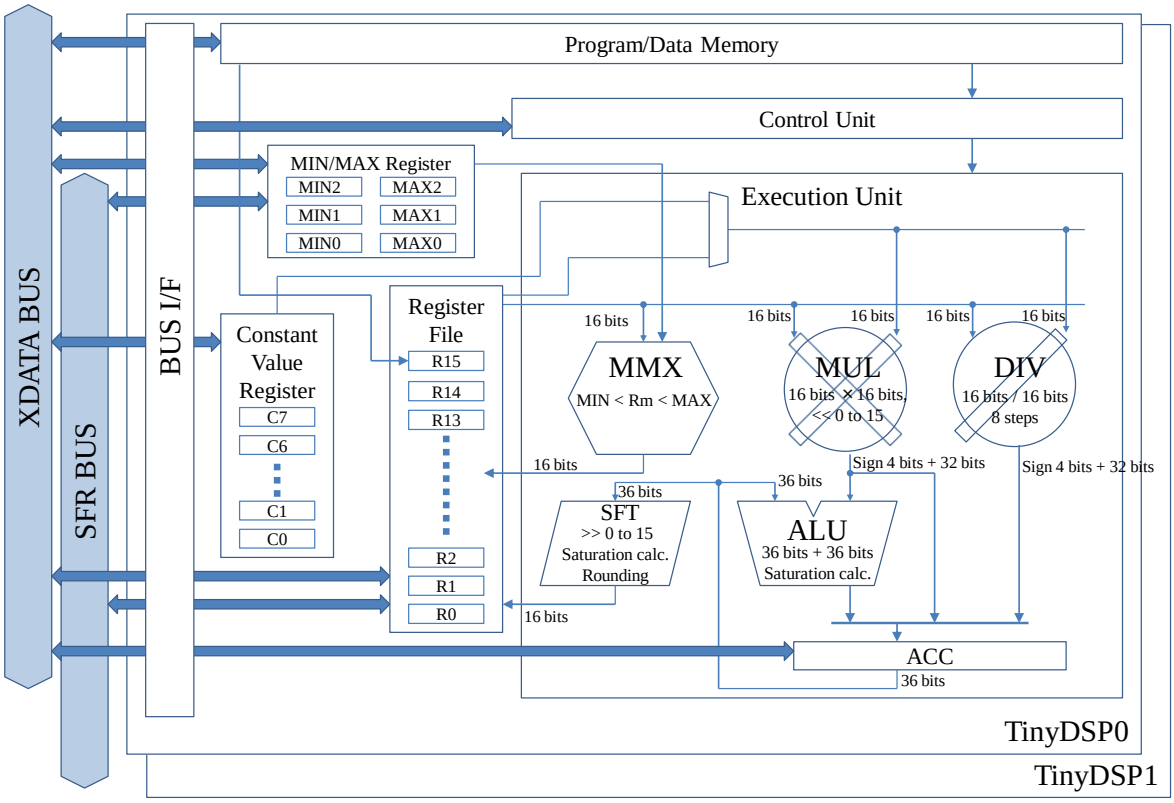


Figure 14-1. TinyDSP Block Diagram

14.3. Resources

- **Register File (Data Registers: R0 to R15)**

Each unit has one register file including sixteen 16-bit data registers, R0 to R15. The data registers are used for storing coefficients, or for internal memory (delay element of digital filter). R0 to R7 are accessible via the SFR BUS, whereas R8 to R15 are accessible via the XDATA BUS. Since the DSAC can access only R0 to R7, it is recommended to assign the input values from the ADC and the output values for the PWM duty cycle to R0 to R7 in order to validate the DSAC transfer. The register files are connected to input ports of the instructions such as MUL, DIV, or MMX.

Also the register files receive the data from the ACC via a shift instruction.

- **ACC (Accumulator)**

Each unit has one 36-bit accumulator register (ACC) for storing internal calculation results. The ACC receives the calculation results from the ALU, MUL, and DIV. The ACC is accessed from the XDATA BUS. When the overflowed calculation result is output to the ACC, the overflowed calculation result is saturated with the positive maximum value or the negative minimum value.

- **MUL (Multiplier)**

Each unit has one multiplier that multiplies 32 bits (16 bits × 16 bits). The MUL receives the data from R0 to R15, and outputs the results to the ALU or the ACC.

- **ALU (Arithmetic and Logical Unit)**

Each unit has one arithmetic logic unit that adds 36 bits (36 bits + 36 bits).

- **SFT (Shifter)**

Each unit has one shifter with only the right shift function. The SFT receives the 36-bit data from the ACC. The SFT cuts 16 bits out of the received data, and outputs the cut 16 bits to R0 to R15. In right shifting, an overflow may occur. When the overflow occurs, saturation processing is performed. Also, the result truncated on right shift (LSB) is rounded to the nearest value.

- **DIV (Divider)**

Each unit has one divider. The processing of this divider are as follows: The reciprocal of the divisor is calculated by the Newton-Raphson method, and is indicated with a 16-bit fixed-point number. Then, multiply the reciprocal of this divisor by the dividend and store the 32-bit result in the ACC.

Since DIV operation is processed using the resources such as MUL, ALU, and ACC, there is no DIV dedicated internal hardware.

- **MMX**

Each unit has one MMX unit that performs a saturation processing. When the MMX receives the data from R0 to R15, the MMS saturates with the predefined minimum value (MIN0 to MIN2) and maximum value (MAX0 to MAX2), and then stores the result in the same register. The TinyDSP has the registers that hold the maximum value and minimum value.

- **Program/Data Memory**

Each unit has one memory for storing 48 words of instructions of 16-bit length. The instructions (16 bits) for configuring the TinyDSP processing sequence and the constants for the LDR function are stored in this memory. Hereafter, this memory is called “Program/Data Memory”.

- **CVR (Constant Value Register, C0 to C7)**

Each unit has eight 16-bit constant registers, C0 to C7, for storing constants. When the CVR function is enabled, the TinyDSP can use the constant registers, as an argument for with the instructions of MUL, MAC, and DIV. For details, see Section 14.9.1.

- **LDR (Load Data Register)**

When the LDR function is enabled, the R15 register can be used as a window to read the constant data stored in the Program/Data Memory. Addresses of the Program/Data Memory accessible by the LDR function are 32 to 47. For details, see Section 14.9.2.

14.4. TinyDSP Instruction

Table 14-3 shows the TinyDSP instructions.

Prepare the sequence using these instructions in advance, and store them in the Program/Data Memory. The instructions other than MMX, MVS, and RSF have the trigger wait and event output functions.

14.4.1. TinyDSP Instruction Format

The TinyDSP instruction is configured as follows:

- **TRIG_WAIT Bit**

The trigger wait field is used to wait for a trigger before executing the instruction. When TRIG_WAIT = 1, the instruction is executed after receiving the trigger. When TRIG_WAIT = 0, the instruction is executed without waiting for the trigger.

- **TRIG_WHAT Bits**

The TinyDSP registers, R0 to R7, are selected in the trigger selection field (see Table 14-2). Writing to the selected register becomes the TinyDSP activation trigger.

Table 14-2. TRIG_WHAT Bits

Register	Bit 14	Bit 13	Bit 12
R0	0	0	0
R1	0	0	1
R2	0	1	0
R3	0	1	1
R4	1	0	0
R5	1	0	1
R6	1	1	0
R7	1	1	1

- **EVENT Bit**

The event field is used for the event output control. An event is output when EVENT = 1, whereas no event is output when EVENT = 0.

- **OPCODE Bits**

The OPCODE field specifies the instruction TinyDSP executes. For details, see Section 14.4.2.

- **FIELD A/B Bits**

The FIELD A/B field specifies the instruction Rn, Rm, # n. Rn and Rm are selected from the data registers, R0 to R15, respectively. #n is the size of the shift bit.

Table 14-3. TinyDSP Instructions

Instruction Format									Instruction	Operation	Execute Frequency
MSB (Bits 15 to 8)						LSB (Bits 7 to 0)					
TRIG_ WAIT	TRIG_ WHAT	EVENT	OPCODE			FIELD A	FIELD B				
0	001	0	0	0	0	Rm	#n	#k	0x0 MMX	If (Rm > MAXn) Rm ← MAXn; Else if (Rm < MINk) Rm ← MINk; #n > 2 → #n = 0	1
0	010	0				Rm	MAX/ MIN	#n	0x0 MVS	FIELD_B[3] = 0 MAXn ← Rm FIELD_B[3] = 1 MINn ← Rm #n > 2: No-operation (NOP)	1
T	WHAT	E				Don't care	Don't care		0x0 NOP	No-operation (NOP)	1
T	WHAT	E	0	0	1	0	Next PC		0x1 JMP	Jump	1
						1	LoadAddr		0x1 LDD	R15 ← Memory[LoadAddr] LDA = LoadAddr + 1	1
T	WHAT	E	0	1	0	Rm	Rn		0x2 MUL	ACC ← Rn × Rm	1
T	WHAT	E	0	1	1	Rm	Rn		0x3 MAC	ACC ← ACC + Rn × Rm	1
T	WHAT	E	1	0	0	Rm	Rn		0x4 DIV	ACC ← Rn/Rm	8
T	WHAT	E	1	0	1	Rm	#n		0x5 LSF	ACC ← Rm << #n	1
T/#n[4]	WHAT	E	1	1	0	Rm	#n[3:0]		0x6 RSF	Rm ← ACC >> #n	1
T	WHAT	E	1	1	1	Rm	Rn		0x7 MVC	Chain movement (delay element) Rm ← Rm-1 ←⋯← Rn+1 ← Rn Initial Rm is discarded. Rn is kept at the same value.	1

14.4.2. Instruction Set

- **0x0**

- **MMX**

When the MMX instruction is executed, Rm is saturated with both the maximum value and minimum value. When Rm is the maximum value or more, the maximum value is stored in Rm. When Rm is the minimum value or less, the minimum value is stored in Rm. There are 3 maximum values and 3 minimum values. Each of the maximum and minimum values, which are used in the MMX instruction, can be specified by the corresponding bits in the instruction code FIELD B: bits 3 to 2 for the maximum values, and bits 1 to 0 for the minimum values. Hereafter, the values of bits 3 to 2 and bits 1 to 0 are defined as #n and #k, respectively. When #n or #k is set to a value more than 2, the each value is treated as zero. The MMX instruction has no trigger wait and event output functions. To use the MMX instruction, set TRIG_WAIT and EVENT field to 0, and set TRIG_WHAT to 1.

- **MVS**

When the MVS instruction is executed, the DSPnMAX register or the DSPnMIN register is updated to the value of Rm. Bits 3 to 0 of the instruction code determine which register to update. To update the DSPnMAX register, set bit 3 = 0. To update the DSPnMIN register, set bit 3 = 1. Bits 2 to 0 specify the value of #n. When #n is set to a value more than 2, the instruction code is considered as the NOP instruction. To use the MVS instruction, set TRIG_WAIT and EVENT field to 0, and set TRIG_WHAT to 2.

- **NOP**

When the NOP instruction is executed, the program counter (PC) is started, and do not affect the others.. When the instruction code is neither the MMX instruction nor MVS instruction, whose settings are as follows, the instruction code is considered as the NOP instruction.

MMX instruction setting: TRIG_WAIT and EVENT field = 0, TRIG_WHAT = 1
MVS instruction setting: TRIG_WAIT and EVENT field = 0, TRIG_WHAT = 2

● **0x1**

- **JMP**

When the JMP instruction is executed, the program counter (PC) is changed to the specific position. Although the jump address is 7-bit length, its MSB is ignored. When the instruction bit 7 is 0 or the LDR function is disabled, the instruction code is considered as the JMP instruction.

- **LDD**

When the LDD instruction is executed, the TinyDSP reads the data from the Program/Data Memory, and stores it in R15. Although the source address of the LDD is 7-bit length, its MSB is ignored. When the LDR function is enabled and the instruction bit 7 is 1, the instruction code is considered as the LDD instruction. For the details of the LDD instruction, see Section 14.9.2.

● **0x2 MUL**

When the MUL instruction is executed, the 32-bit result of multiplying Rn (16 bits) by Rm (16 bits) is sign-extended to 36 bits, and then the sign-extended result is stored in the ACC.

● **0x3 MAC**

When the MAC instruction is executed, the 32-bit result obtained by multiplying Rn (16 bits) by Rm (16 bits) is sign-extended to 36 bits, and the sign-extended result is added to the ACC value of 36 bits, and then the added result is stored in the ACC. When overflowing to the positive direction in the adding process, the saturated value of 0x7_FFFF_FFFF is stored in the ACC. When overflowing to the negative direction in the adding process, the saturated value of 0x8_0000_0000 is stored in the ACC.

● **0x4 DIV**

When the DIV instruction is executed, the reciprocal of the 16-bit divisor Rm (i.e., 1/Rm) is approximately calculated firstly, and then the result is internally transformed to 16-bit fixed-point number. Next, multiply the transformed value by the dividend Rn (16 bits), and then the multiplying result is sign-extended to 36 bits. Finally, the sign-extended result is stored in the ACC. The DIV instruction has 2 calculation modes (normal mode and high accuracy mode). The calculation mode of the DIV instruction is determined by the DSPnCTRL2.HPDIV bit. For improvement of the accuracy of calculation result when the value of the divisor, Rm, is high, use the high accuracy mode. Both modes are executed in 8 cycles.

The decimal point exists only on user's assumption. When the decimal point of the divisor, Rm, is located between bit 1 and bit 0 (Q 1), the decimal point of 1/Rm is located between bit 14 and bit 13 (Q 14). When the decimal point of Rn is between bit 1 and bit 0 (Q1), the decimal point of the finally stored value in ACC is located between bit 15 and bit 14 (Q1 × Q14 = Q15). The value that is any16-bit field in the ACC selected by user can be stored in the register file (see Section 14.3) by the RSF instruction. Only the DIV instruction needs the multiple cycles for the execution.

● **0x5 LSF**

When the LSF instruction is executed, Rm is shifted left by n bits, and then the shifted value is stored in the ACC. When shifting left, the sign is extended using the sign of Rm. Lower bits are filled up with zeros.

● **0x6 RSF**

When the RSF instruction is executed, the ACC is shifted right by n bits, and then the shifted value is stored in Rm. The 16-bit data that is cut according to the amount shifting right from the 36-bit ACC is output to Rm. If an overflow occurs in right shifting, the saturated value is stored in Rm. Also, the result truncated on right shift (LSB) is rounded to the nearest value.

The RSF instruction has 2 modes, one can use the trigger function, and the other cannot. In the mode where the trigger function can be used, the right shift range is 0 to 15. In the mode where the trigger function cannot be used, the right shift range is 0 to 31. The DSPnCTRL2.EXRSF bit determines which mode to be used.

● **0x7 MVC**

When the MVC instruction is executed, the data in the register file is moved in a chain (chain movement), and then the delay element of the digital filter is implemented. The targets of the chain movement are consecutive number data registers. When m > n in the instruction field, Rm receives the value from Rm−1, Rm−1 receives the value from Rm−2, and finally Rn+1 receives the value from Rn. The value of Rm before executing the MVC instruction is discarded. The value of Rn does not change even after executing the MVC instruction. When m ≤ n, the MVC instruction operates in the same way as the NOP instruction.

14.5. Operations

The setting procedure and the operation of the TinyDSP are as follows:

(1) The following settings are required for the TinyDSP in advance.

- Set the program sequence in the Program/Data Memory.
- Set the initial value of R0 to R15 and C0 to C7 (such as coefficient or delay element of digital filter).

(2) To enable the TinyDSP,set the DSPnCTRL.DSPE bit.
The program sequence starts from PC = 0x00 usually, although the initial value of instruction sequence program counter (PC) can be set in advance.

(3) The TinyDSP instruction sequence is started. When the TRIG_WAIT flag of the TinyDSP instruction is set, the TinyDSP instruction is not executed until a trigger occurs. The trigger occurs when the data register is updated (data writing). The data register is selected from R0 to R7. R0 to R7 are defined by the TRIG_WHAT filed (see Table 14-2). When the trigger is detected, the TinyDSP executes the instruction that is set to the trigger, and then goes on to the next PC address processing.

(4) For the trigger to restart the TinyDSP instruction, see the description of the DSPnTRG register in Section 14.11.11. For example, the operations when TRIG_WAIT = 1 and TRIG_WHAT = 3 in the TinyDSP instruction field (see Table 14-3) are described as follows. This means that the TinyDSP, which is set a trigger, is waiting for R3 update by any of the CPU, EPU, or DSAC.

(5) When the DSPnTRG.SET_R3 bit is 0, the instruction is suspended before execution. When the new value is written to R3 by any of the CPU, EPU, or DSAC, the DSPnTRG.SET_R3 bit is automatically set, and the TinyDSP instruction execution is restarted. At this time, the DSPnTRG.SET_R3 bit is automatically cleared. When the CPU or the DSAC updates R3 to set DSPnTRG.SET_R3 = 1 before the TinyDSP instruction is executed, the TinyDSP instruction restarts execution immediately without suspending, and then the DSPnTRG.SET_R3 bit is cleared. Although the DSPnTRG register does not need to be accessed from the CPU during the TinyDSP operation, the DSPnTRG register can be accessed from the CPU for initializing again.

(6) When the PC reaches its final address (0x2F) during the TinyDSP instruction sequence, the PC is returned to 0x00, and then the TinyDSP continues the instruction sequence.

(7) When the instruction EVENT = 1 is completed, the TinyDSP interrupt flag (i.e., DSPnCTRL.DSPIF bit) is set to 1. Also, internal event pulses are generated for other modules. While the DSPnCTRL.DSPIE bit is set, the interrupt signal is asserted.

(8) When the saturation processing occurs during the TinyDSP instruction execution, the DSPnCTRL.DSP_SA bit (for the ALU) or the DSPnCTRL.DSP_SS bit (for the shifter) is set to 1. Only the occurrence of saturation processing is notified to the CPU. Each flag can be cleared by the CPU.

(9) While the DSPnDBG.DSP_DBG bit is 1, the TinyDSP operates with a debug mode. In the debug mode, the TinyDSP executes the program sequence with the step operation. The step operation is executed only when 1 is written to the DSPnDBG.DSP_STP bit during DSPnDBG.DSP_DBG = 1. When the DSPnDBG.DSP_DBG and DSPnDBG.DSP_STP bits are set at the same time, the step operation is not executed. Even if the TRIG_WAIT flag of the TinyDSP instruction is set, this instruction for the step operation is forced to execute. When the PC specifies the final address, the PC is returned to 0x00, and the step operation is repeated from the first address.

(10) Regarding the register or storage resource that can be accessed from the TinyDSP and the any of the CPU, EPU, or DSAC, when the TinyDSP and any of the CPU, EPU, or DSAC access the same resource simultaneously, the access by the CPU, EPU, or DSAC has the highest priority.

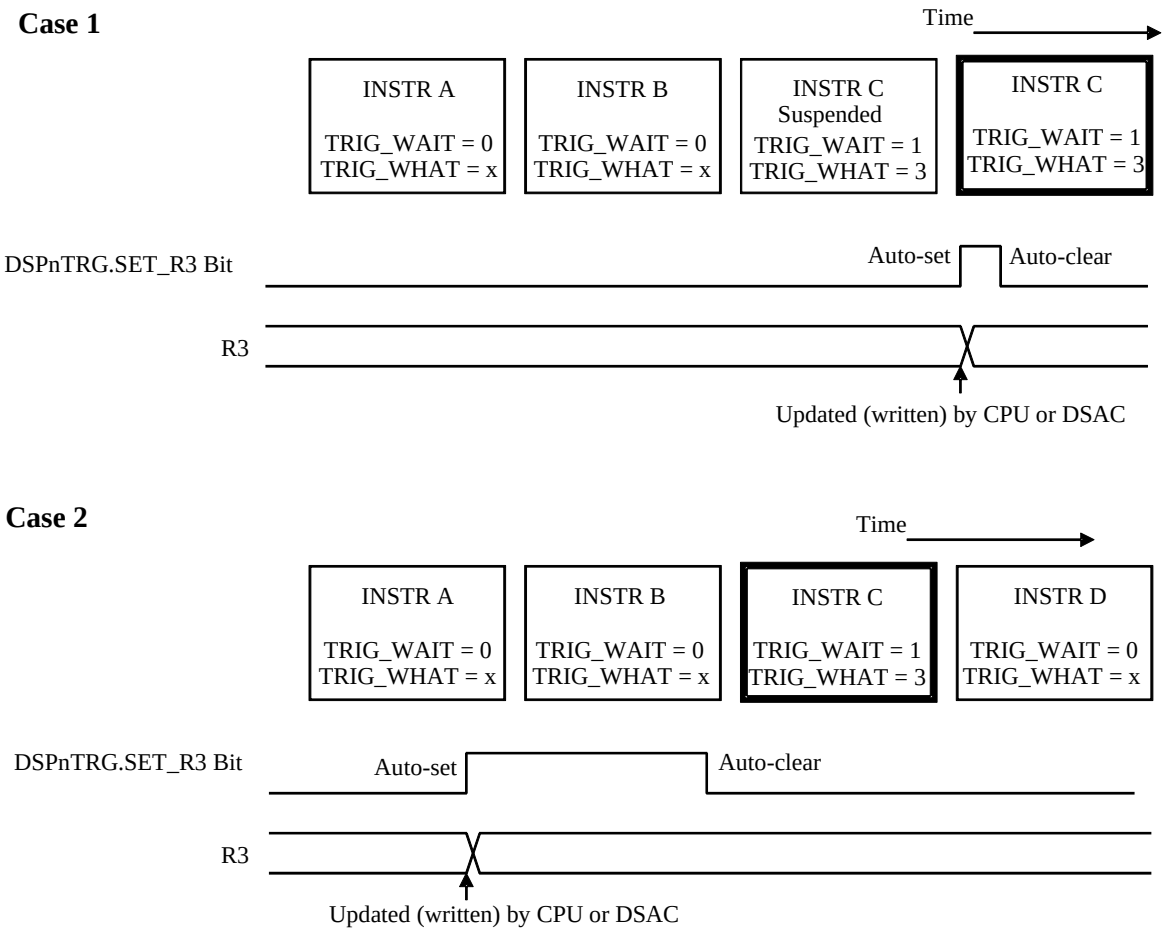


Figure 14-2. Trigger Operation of Instruction Sequence

14.6. 16-bit Register Access

All data registers (R8 to R15) in the register file and all constant registers (C0 to C7, MAX0 to MAX2, MIN0 to MIN2) are 16-bit width. These 16-bit width registers are composed of the lower byte (LSByte) register and higher byte (MSByte) register.

Data registers (R8 to R15) and all constant registers respectively have 3 addresses in the following addresses on the XDATA BUS area.

- TinyDSP0: 0xF788 to 0xF78F, 0xF798 to 0xF79F, 0xF7A8 to 0xF7AD
- TinyDSP1: 0xF808 to 0xF80F, 0xF818 to 0xF81F, 0xF828 to 0xF82D

LSByte and MSByte are assigned to one of the 3 addresses. The other 2 addresses are contiguous; LSByte and MSByte are assigned to the smaller and higher addresses, respectively.

Each data register (R0 to R7) and constant register (MAX0 to MAX2 and MIN0 to MIN2) has one address, which LSByte and MSByte are assigned, on the SFR area

The 16-bit register must be written in the order of LSByte and MSByte, sequentially. When LSByte is written first, the value of the written LSByte is stored in the temporary register for LSByte. When MSByte is written next, the value of the temporary register for LSByte and the value of MSByte are written to the 16-bit register simultaneously. The data must be read from the 16-bit register in the order of LSByte and MSByte, sequentially. When LSByte is read first, only the LSByte value out of the value of the 16-bit register is read, and the MSByte value is stored in a dedicated temporary register. At the next read, the MSByte value is read from the dedicated temporary register.

The DSPnMAXxL/H and DSPnMINxL/H registers have the addresses in both the XDATA BUS and SFR BUS areas. When the SFR BUS access conflicts with the XDATA BUS access, the writing of XDATA BUS has the highest priority.

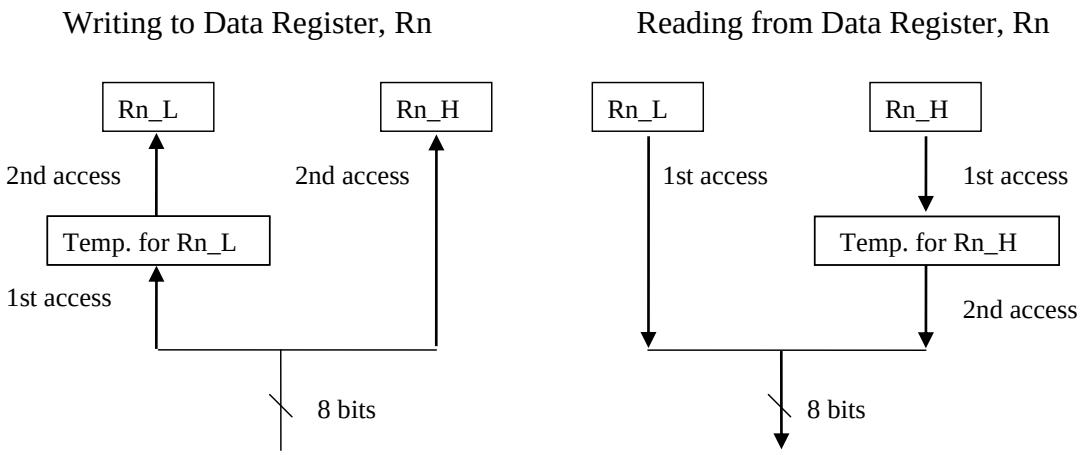


Figure 14-3. Data Register Access

The access states of LSByte and MSByte of the 16-bit register is managed by the access counter of the units such as CPU, DSAC, or EPU as follows, respectively.

● **CPU**

When the CPU reads the data from the 16-bit register, the CPU access counter is incremented. The access counter of the XDATA BUS is separated with its of the SFR BUS. When the DSPnRST.CPUACCLA bit is set to 1, both the SFR BUS CPU access counter and the XDATA BUS CPU access counter are cleared, and the CPU is ready to read LSByte.

● **DSAC**

When the DSAC reads the data from one of registers R0 to R7, MIN0 to MIN2, or MAX0 to MAX2, the DSAC access counter is incremented. When the DSPnRST.DSACACCLA bit is set to 1, the DSAC access counter is cleared, and the DSAC is ready to read LSByte. The 16-bit data can be directly read from and write to registers assigned to the SFR area by the DSAC. For the direct access to 16-bit data, see Section 11.

● **EPU**

When the EPU reads data from the 16-bit register via the XDATA BUS, the EPU access counter is incremented. When the DSPnRST.EPUCACCLA bit is set to 1, the EPU access counter is cleared, and the EPU is ready to read LSByte. When the EPU accesses the 16-bit register of the TinyDSP via the SFR BUS, the access length must be 16 bits because the EPU does not have a buffer in the SFR area.

14.7. Event Output

2 events (Event0 and Event1) are output from each TinyDSP to the DSAC.

- TinyDSP0 Event0
- TinyDSP0 Event1
- TinyDSP1 Event0
- TinyDSP1 Event1

The event outputs are controlled by the event field of each instruction code. The Event0 and Event1 are selected as follows. When the TinyDSP outputs the Event0 or Event1, the DSPnCTRL.DSPIF bit that is interrupt flag is set.

- Event0: EVENT = 1, TRIG_WHAT != 7
- Event1: EVENT = 1, TRIG_WHAT = 7

Where, the event is output according to the TinyDSP instruction as follows:

- When TRIG_WAIT = 1, EVENT = 1, and TRIG_WHAT != 7, the TinyDSP instruction waits for the writing to the register specified by the TRIG_WAIT. When the register is written (i.e., it is the trigger), the TinyDSP executes the instruction. Then,the Event0 is output after the instruction processing is completed.
- When EVENT = 1, TRIG_WAIT = 1, and TRIG_WHAT = 7, the TinyDSP waits for writing to R7. When the register is written (i.e., it is the trigger), the TinyDSP executes the instruction. Then,the Event1 is output after the instruction processing is completed.

14.8. Program/Data Memory

48 instructions can be stored in the Program/Data Memory that stores instructions and data (see Table 14-4 and Table 14-5). The Program/Data Memory has its own address space (0x00 to 0x2F) that 16 bits (one instruction, one data) are assigned to one address. To write data to the Program/Data Memory, write the address to be written to the DSPn_PRG_ADR register, first. Then, write the LSByte and MSByte of the data to be written to the DSPn_PRG_DATL and DSPn_PRG_DATH registers, respectively. There is no buffer function. Written data is reflected to the Program/Data Memory when the LSByte and MSByte are written, respectively. To read data from the Program/Data Memory, write the address to be read to the DSPn_PRG_ADR register. As a result, the LSByte and MSByte of the data to be read can be read from the DSPn_PRG_DATL and DSPn_PRG_DATH registers, respectively. As well as the writing, there is no buffer function.

Table 14-4. Program/Data Memory for Holding Instructions

	MSB (Bit 15)	LSB (Bit 0)
Address (PC) (DSPn_PRG_ADR)	Program/Data Memory for holding the instruction (16-bit width) (The DSPn_PRG_ADR register is assigned to the CPU data area)	
0x00 to 0x2F	DSPn_PRG_DATH	DSPn_PRG_DATL

Table 14-5. Program/Data Memory for Holding Data

	MSB (Bit 15)	LSB (Bit 0)
Address (LDA) (DSPn_PRG_ADR)	Constant data for LDR (16-bit width) (The DSPn_PRG_ADR register is assigned to the CPU data area)	
0x20 to 0x2F	DSPn_PRG_DATH	DSPn_PRG_DATL

14.9. Coefficient Hold Functions

The coefficient information of the TinyDSP is held in R0 to R15, C0 to C7, and the Program/Data Memory.

14.9.1. CVR Function

The CVR function is for holding coefficients in C0 to C7.

The FIELD B in the instruction code means R0 to R7 in the normal mode, but means C0 to C7 in the CVR mode. Enabling the CVR mode is defined by the DSPnCNSTEN register. The value of 0 to 15 can be specified to the FIELD B. The DSPnCTRL2.CNSTSEL bit determines whether 0 to 7 or 8 to 15 are assigned to C0 to C7. When the instructions of MUL, MAC, and DIV are executed, the CVR function operates.

When DSPnCTRL2.CNSTSEL = 0, 0 to 7 of the FIELD B are assigned to R0 to R7, and 8 to 15 of the FIELD B are assigned to R8 to R15 or C0 to C7. Using registers, R8 to R15 or C0 to C7, are defined by the DSPnCNSTEN register. For example, the DSPnCNSTEN register is 0x01, only 8 of the FIELD B are assigned to C0.

When DSPnCTRL2.CNSTSEL = 1, 8 to 15 of the FIELD B are assigned to R8 to R15, and 0 to 7 of the FIELD B are assigned to R0 to R7 or C0 to C7. Using the registers, R0 to R7 or C0 to C7, are defined by the DSPnCNSTEN register. For example, the DSPnCNSTEN register is 0x01, only 0 of the FIELD B are assigned to C0.

14.9.2. LDR Function

The LDR function is for holding coefficients in the Program/Data Memory.

In the LDR mode (DSPnCTRL2.LDEN = 1), the instructions of MUL, MAC, and DIV can reference constant data in the Program/Data Memory. To use the LDR mode, it is recommended to initialize the LDR function using the LDD instruction.

When the LDD instruction is executed, the value of the Program/Data Memory address indicated in the LoadAddr field is transferred to R15. As a result, the value of LoadAddr + 1 is set to the DSPnLDA register. The 16 registers (32 to 47) of the Program/Data Memory address can be used in the data area of the LDR function.

When FIELD B in the instruction codes of the MUL, MAC, and DIV is 15 (0b1111), each instruction uses a value of the Program/Data Memory stored in R15. After each instruction is executed, the program/data memory, which is stored in the address indicated by the DSPnLDA register, is transferred to R15, and the DSPnLDA register is incremented by 1. The LDR function has the higher priority than the CVR function.

Figure 14-4 and Figure 14-5 show the block diagrams of LDD instruction and LDR function, respectively.

14.9.3. Priority of CVR and LDR Funcitons

The LDR has the higher priority than the CVR. When the LDR function is enabled, and C7 is assigned to FIELD B = 15, the value stored in R15 is used because the LDR function has the highest priority.

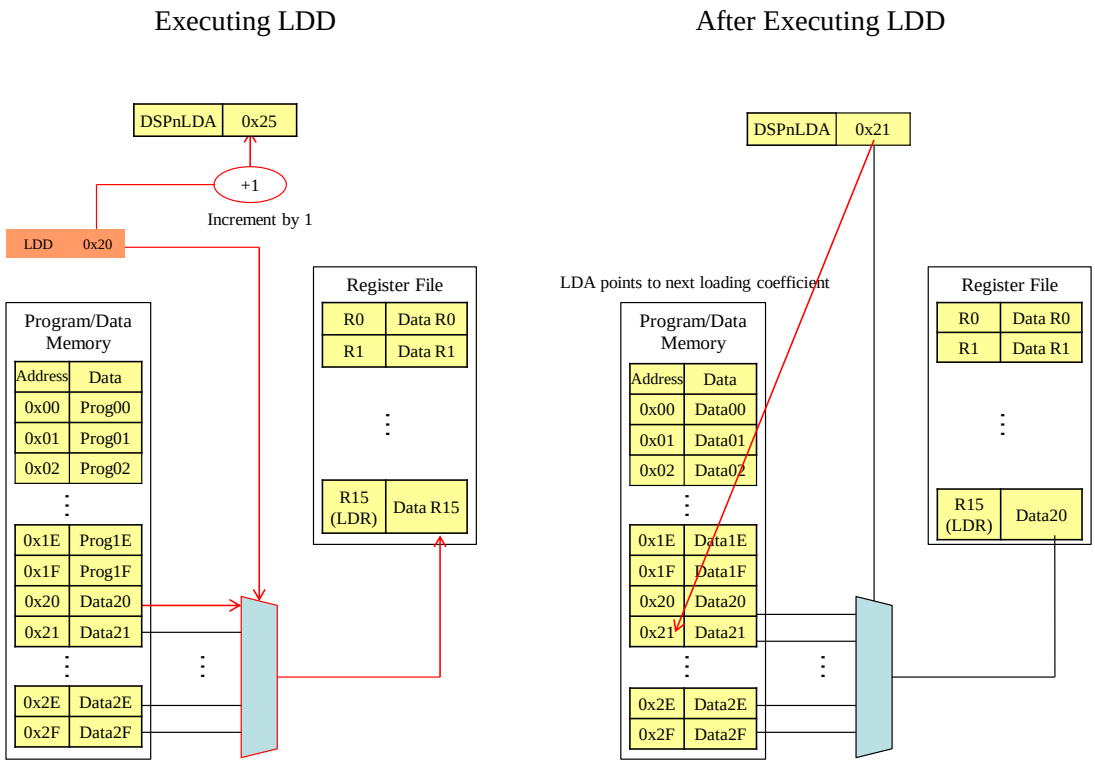


Figure 14-4. Block Diagram of LDD Instruction

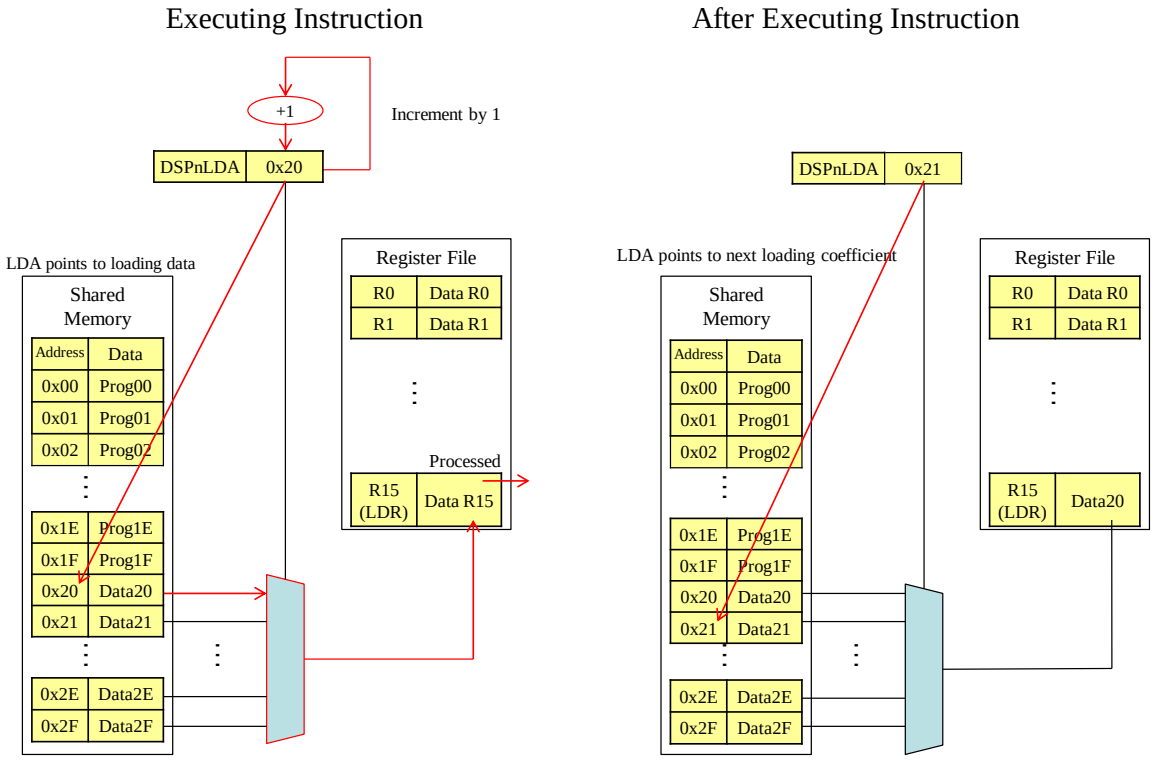


Figure 14-5. Block Diagram of LDR Function

14.10. Application Examples

Figure 14-6 and Figure 14-7 show the digital filter application examples. Although the filters of 2 applications are the same, Figure 14-6 is the application example without using the LDR function, and Figure 14-7 is the application example using the LDR function.

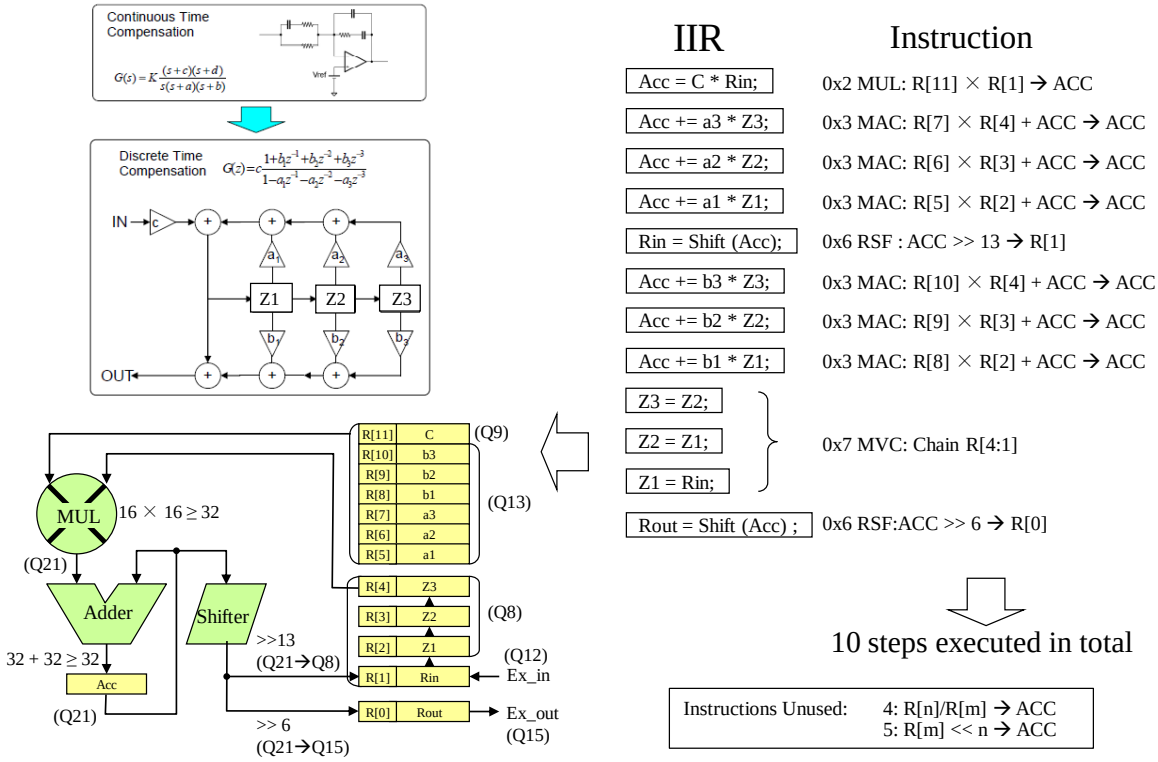


Figure 14-6. Application Example (LDR Function Disabled)

IIR Example When LDR Enabled

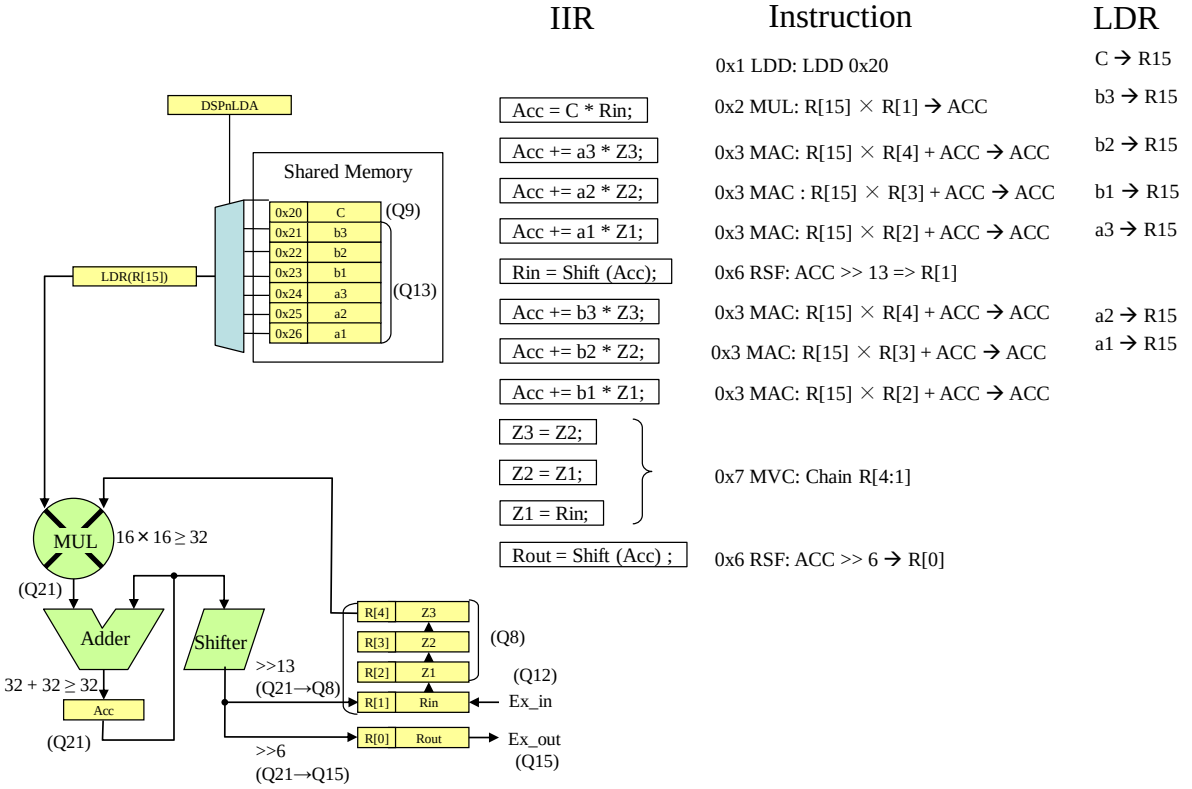


Figure 14-7. Application Example (LDR Function Enabled)

14.11. Register Descriptions

Table 14-6. List of XDATA BUS Registers

Symbol	Name	Address 1	Address 2	Initial Value
DSP0CTRL	TinyDSP0 Control Register	0xF780	—	0x00
DSP0EXEC	TinyDSP0 Execution Register	0xF781	—	0x00
DSP0TRG	TinyDSP0 Execution Trigger Status	0xF782	—	0x00
DSP0RST	TinyDSP0 Access Counter Clear Register	0xF783	—	0x00
DSP0DBG	TinyDSP0 Debug Register	0xF784	—	0x00
DSP0CTRL2	TinyDSP0 Control2 Register	0xF785	—	0x00
DSP0CNSTEN	TinyDSP0 CVR Enable Register	0xF786	—	0x00
DSP0_R8_L	TinyDSP0 R8 LSB Side	0xF788	0xF7C0	0x00
DSP0_R8_H	TinyDSP0 R8 MSB Side	0xF788	0xF7C1	0x00
DSP0_R9_L	TinyDSP0 R9 LSB Side	0xF789	0xF7C2	0x00
DSP0_R9_H	TinyDSP0 R9 MSB Side	0xF789	0xF7C3	0x00
DSP0_R10_L	TinyDSP0 R10 LSB Side	0xF78A	0xF7C4	0x00
DSP0_R10_H	TinyDSP0 R10 MSB Side	0xF78A	0xF7C5	0x00
DSP0_R11_L	TinyDSP0 R11 LSB Side	0xF78B	0xF7C6	0x00
DSP0_R11_H	TinyDSP0 R11 MSB Side	0xF78B	0xF7C7	0x00
DSP0_R12_L	TinyDSP0 R12 LSB Side	0xF78C	0xF7C8	0x00
DSP0_R12_H	TinyDSP0 R12 MSB Side	0xF78C	0xF7C9	0x00
DSP0_R13_L	TinyDSP0 R13 LSB Side	0xF78D	0xF7CA	0x00
DSP0_R13_H	TinyDSP0 R13 MSB Side	0xF78D	0xF7CB	0x00
DSP0_R14_L	TinyDSP0 R14 LSB Side	0xF78E	0xF7CC	0x00
DSP0_R14_H	TinyDSP0 R14 MSB Side	0xF78E	0xF7CD	0x00
DSP0_R15_L	TinyDSP0 R15 LSB Side	0xF78F	0xF7CE	0x00
DSP0_R15_H	TinyDSP0 R15 MSB Side	0xF78F	0xF7CF	0x00
DSP0_ACC_0	TinyDSP0 ACC[7:0]	0xF790	—	0x00
DSP0_ACC_1	TinyDSP0 ACC[15:8]	0xF791	—	0x00
DSP0_ACC_2	TinyDSP0 ACC[23:15]	0xF792	—	0x00
DSP0_ACC_3	TinyDSP0 ACC[31:24]	0xF793	—	0x00
DSP0_ACC_4	TinyDSP0 ACC[36:32]	0xF794	—	0x00
DSP0_C0_L	TinyDSP0 C0 LSB Side	0xF798	0xF7D0	0x00
DSP0_C0_H	TinyDSP0 C0 MSB Side	0xF798	0xF7D1	0x00
DSP0_C1_L	TinyDSP0 C1 LSB Side	0xF799	0xF7D2	0x00
DSP0_C1_H	TinyDSP0 C1 MSB Side	0xF799	0xF7D3	0x00
DSP0_C2_L	TinyDSP0 C2 LSB Side	0xF79A	0xF7D4	0x00
DSP0_C2_H	TinyDSP0 C2 MSB Side	0xF79A	0xF7D5	0x00

Symbol	Name	Address 1	Address 2	Initial Value
DSP0_C3_L	TinyDSP0 C3 LSB Side	0xF79B	0xF7D6	0x00
DSP0_C3_H	TinyDSP0 C3 MSB Side	0xF79B	0xF7D7	0x00
DSP0_C4_L	TinyDSP0 C4 LSB Side	0xF79C	0xF7D8	0x00
DSP0_C4_H	TinyDSP0 C4 MSB Side	0xF79C	0xF7D9	0x00
DSP0_C5_L	TinyDSP0 C5 LSB Side	0xF79D	0xF7DA	0x00
DSP0_C5_H	TinyDSP0 C5 MSB Side	0xF79D	0xF7DB	0x00
DSP0_C6_L	TinyDSP0 C6 LSB Side	0xF79E	0xF7DC	0x00
DSP0_C6_H	TinyDSP0 C6 MSB Side	0xF79E	0xF7DD	0x00
DSP0_C7_L	TinyDSP0 C7 LSB Side	0xF79F	0xF7DE	0x00
DSP0_C7_H	TinyDSP0 C7 MSB Side	0xF79F	0xF7DF	0x00
DSP0_PRG_DATL	TinyDSP0 Program Memory LSB Side	0xF7A0	—	0x00
DSP0_PRG_DATH	TinyDSP0 Program Memory MSB Side	0xF7A1	—	0x00
DSP0_PRG_ADR	TinyDSP0 Program Memory Address	0xF7A2	—	0x00
DSP0LDA	TinyDSP0 LDR Load Address Register	0xF7A3	—	0x20
DSP0MAX0L	TinyDSP0 MAX LSB Side	0xF7A8	0xF7E0	0x00
DSP0MAX0H	TinyDSP0 MAX MSB Side	0xF7A8	0xF7E1	0x00
DSP0MIN0L	TinyDSP0 MIN LSB Side	0xF7A9	0xF7E2	0x00
DSP0MIN0H	TinyDSP0 MIN MSB Side	0xF7A9	0xF7E3	0x00
DSP0MAX1L	TinyDSP0 MAX LSB Side	0xF7AA	0xF7E4	0x00
DSP0MAX1H	TinyDSP0 MAX MSB Side	0xF7AA	0xF7E5	0x00
DSP0MIN1L	TinyDSP0 MIN LSB Side	0xF7AB	0xF7E6	0x00
DSP0MIN1H	TinyDSP0 MIN MSB Side	0xF7AB	0xF7E7	0x00
DSP0MAX2L	TinyDSP0 MAX LSB Side	0xF7AC	0xF7E8	0x00
DSP0MAX2H	TinyDSP0 MAX MSB Side	0xF7AC	0xF7E9	0x00
DSP0MIN2L	TinyDSP0 MIN LSB Side	0xF7AD	0xF7EA	0x00
DSP0MIN2H	TinyDSP0 MIN MSB Side	0xF7AD	0xF7EB	0x00
DSP1CTRL	TinyDSP1 Control Register	0xF800	—	0x00
DSP1EXEC	TinyDSP1 Execution Register	0xF801	—	0x00
DSP1TRG	TinyDSP1 Execution Trigger Status	0xF802	—	0x00
DSP1RST	TinyDSP1 Access Counter Clear Register	0xF803	—	0x00
DSP1DBG	TinyDSP1 Debug Register	0xF804	—	0x00
DSP1CTRL2	TinyDSP1 Control2 Register	0xF805	—	0x00
DSP1CNSTEN	TinyDSP1 CVR Enable Register	0xF806	—	0x00
DSP1_R8_L	TinyDSP1 R8 LSB Side	0xF808	0xF840	0x00
DSP1_R8_H	TinyDSP1 R8 MSB Side	0xF808	0xF841	0x00
DSP1_R9_L	TinyDSP1 R9 LSB Side	0xF809	0xF842	0x00
DSP1_R9_H	TinyDSP1 R9 MSB Side	0xF809	0xF843	0x00

Symbol	Name	Address 1	Address 2	Initial Value
DSP1_R10_L	TinyDSP1 R10 LSB Side	0xF80A	0xF844	0x00
DSP1_R10_H	TinyDSP1 R10 MSB Side	0xF80A	0xF845	0x00
DSP1_R11_L	TinyDSP1 R11 LSB Side	0xF80B	0xF846	0x00
DSP1_R11_H	TinyDSP1 R11 MSB Side	0xF80B	0xF847	0x00
DSP1_R12_L	TinyDSP1 R12 LSB Side	0xF80C	0xF848	0x00
DSP1_R12_H	TinyDSP1 R12 MSB Side	0xF80C	0xF849	0x00
DSP1_R13_L	TinyDSP1 R13 LSB Side	0xF80D	0xF84A	0x00
DSP1_R13_H	TinyDSP1 R13 MSB Side	0xF80D	0xF84B	0x00
DSP1_R14_L	TinyDSP1 R14 LSB Side	0xF80E	0xF84C	0x00
DSP1_R14_H	TinyDSP1 R14 MSB Side	0xF80E	0xF84D	0x00
DSP1_R15_L	TinyDSP1 R15 LSB Side	0xF80F	0xF84E	0x00
DSP1_R15_H	TinyDSP1 R15 MSB Side	0xF80F	0xF84F	0x00
DSP1_ACC_0	TinyDSP1 ACC[7:0]	0xF810	—	0x00
DSP1_ACC_1	TinyDSP1 ACC[15:8]	0xF811	—	0x00
DSP1_ACC_2	TinyDSP1 ACC[23:15]	0xF812	—	0x00
DSP1_ACC_3	TinyDSP1 ACC[31:24]	0xF813	—	0x00
DSP1_ACC_4	TinyDSP1 ACC[36:32]	0xF814	—	0x00
DSP1_C0_L	TinyDSP1 C0 LSB Side	0xF818	0xF850	0x00
DSP1_C0_H	TinyDSP1 C0 MSB Side	0xF818	0xF851	0x00
DSP1_C1_L	TinyDSP1 C1 LSB Side	0xF819	0xF852	0x00
DSP1_C1_H	TinyDSP1 C1 MSB Side	0xF819	0xF853	0x00
DSP1_C2_L	TinyDSP1 C2 LSB Side	0xF81A	0xF854	0x00
DSP1_C2_H	TinyDSP1 C2 MSB Side	0xF81A	0xF855	0x00
DSP1_C3_L	TinyDSP1 C3 LSB Side	0xF81B	0xF856	0x00
DSP1_C3_H	TinyDSP1 C3 MSB Side	0xF81B	0xF857	0x00
DSP1_C4_L	TinyDSP1 C4 LSB Side	0xF81C	0xF858	0x00
DSP1_C4_H	TinyDSP1 C4 MSB Side	0xF81C	0xF859	0x00
DSP1_C5_L	TinyDSP1 C5 LSB Side	0xF81D	0xF85A	0x00
DSP1_C5_H	TinyDSP1 C5 MSB Side	0xF81D	0xF85B	0x00
DSP1_C6_L	TinyDSP1 C6 LSB Side	0xF81E	0xF85C	0x00
DSP1_C6_H	TinyDSP1 C6 MSB Side	0xF81E	0xF85D	0x00
DSP1_C7_L	TinyDSP1 C7 LSB Side	0xF81F	0xF85E	0x00
DSP1_C7_H	TinyDSP1 C7 MSB Side	0xF81F	0xF85F	0x00
DSP1_PRG_DATL	TinyDSP1 Program Memory LSB Side	0xF820	—	0x00
DSP1_PRG_DATH	TinyDSP1 Program Memory MSB Side	0xF821	—	0x00
DSP1_PRG_ADR	TinyDSP1 Program Memory Address	0xF822	—	0x00
DSP1LDA	TinyDSP1 LDR Load Address Register	0xF823	—	0x20

Symbol	Name	Address 1	Address 2	Initial Value
DSP1MAX0L	TinyDSP1 MAX0 LSB Side	0xF828	0xF860	0x00
DSP1MAX0H	TinyDSP1 MAX0 MSB Side	0xF828	0xF861	0x00
DSP1MIN0L	TinyDSP1 MIN0 LSB Side	0xF829	0xF862	0x00
DSP1MIN0H	TinyDSP1 MIN0 MSB Side	0xF829	0xF863	0x00
DSP1MAX1L	TinyDSP1 MAX1 LSB Side	0xF82A	0xF864	0x00
DSP1MAX1H	TinyDSP1 MAX1 MSB Side	0xF82A	0xF865	0x00
DSP1MIN1L	TinyDSP1 MIN1 LSB Side	0xF82B	0xF866	0x00
DSP1MIN1H	TinyDSP1 MIN1 MSB Side	0xF82B	0xF867	0x00
DSP1MAX2L	TinyDSP1 MAX2 LSB Side	0xF82C	0xF868	0x00
DSP1MAX2H	TinyDSP1 MAX2 MSB Side	0xF82C	0xF869	0x00
DSP1MIN2L	TinyDSP1 MIN2 LSB Side	0xF82D	0xF86A	0x00
DSP1MIN2H	TinyDSP1 MIN2 MSB Side	0xF82D	0xF86B	0x00

Table 14-7. List of SFR BUS Registers

Symbol	Name	Address	Initial Value
DSP0_R0_L	TinyDSP0 R0 LSB Side	0xC4	0x00
DSP0_R0_H	TinyDSP0 R0 MSB Side	0xC4	0x00
DSP0_R1_L	TinyDSP0 R1 LSB Side	0xC5	0x00
DSP0_R1_H	TinyDSP0 R1 MSB Side	0xC5	0x00
DSP0_R2_L	TinyDSP0 R2 LSB Side	0xC6	0x00
DSP0_R2_H	TinyDSP0 R2 MSB Side	0xC6	0x00
DSP0_R3_L	TinyDSP0 R3 LSB Side	0xC7	0x00
DSP0_R3_H	TinyDSP0 R3 MSB Side	0xC7	0x00
DSP0_R4_L	TinyDSP0 R4 LSB Side	0xCC	0x00
DSP0_R4_H	TinyDSP0 R4 MSB Side	0xCC	0x00
DSP0_R5_L	TinyDSP0 R5 LSB Side	0xCD	0x00
DSP0_R5_H	TinyDSP0 R5 MSB Side	0xCD	0x00
DSP0_R6_L	TinyDSP0 R6 LSB Side	0xCE	0x00
DSP0_R6_H	TinyDSP0 R6 MSB Side	0xCE	0x00
DSP0_R7_L	TinyDSP0 R7 LSB Side	0xCF	0x00
DSP0_R7_H	TinyDSP0 R7 MSB Side	0xCF	0x00
DSP0MIN0L	TinyDSP0 MIN0 LSB Side	0x24	0x00
DSP0MIN0H	TinyDSP0 MIN0 MSB Side	0x24	0x00
DSP0MAX0L	TinyDSP0 MAX0 LSB Side	0x25	0x00
DSP0MAX0H	TinyDSP0 MAX0 MSB Side	0x25	0x00
DSP0MIN1L	TinyDSP0 MIN1 LSB Side	0x2C	0x00
DSP0MIN1H	TinyDSP0 MIN1 MSB Side	0x2C	0x00
DSP0MAX1L	TinyDSP0 MAX1 LSB Side	0x2D	0x00
DSP0MAX1H	TinyDSP0 MAX1 MSB Side	0x2D	0x00
DSP0MIN2L	TinyDSP0 MIN2 LSB Side	0x34	0x00
DSP0MIN2H	TinyDSP0 MIN2 MSB Side	0x34	0x00
DSP0MAX2L	TinyDSP0 MAX2 LSB Side	0x35	0x00
DSP0MAX2H	TinyDSP0 MAX2 MSB Side	0x35	0x00
DSP1_R0_L	TinyDSP1 R0 LSB Side	0xD4	0x00
DSP1_R0_H	TinyDSP1 R0 MSB Side	0xD4	0x00
DSP1_R1_L	TinyDSP1 R1 LSB Side	0xD5	0x00
DSP1_R1_H	TinyDSP1 R1 MSB Side	0xD5	0x00
DSP1_R2_L	TinyDSP1 R2 LSB Side	0xD6	0x00
DSP1_R2_H	TinyDSP1 R2 MSB Side	0xD6	0x00
DSP1_R3_L	TinyDSP1 R3 LSB Side	0xD7	0x00
DSP1_R3_H	TinyDSP1 R3 MSB Side	0xD7	0x00

Symbol	Name	Address	Initial Value
DSP1_R4_L	TinyDSP1 R4 LSB Side	0xDC	0x00
DSP1_R4_H	TinyDSP1 R4 MSB Side	0xDC	0x00
DSP1_R5_L	TinyDSP1 R5 LSB Side	0xDD	0x00
DSP1_R5_H	TinyDSP1 R5 MSB Side	0xDD	0x00
DSP1_R6_L	TinyDSP1 R6 LSB Side	0xDE	0x00
DSP1_R6_H	TinyDSP1 R6 MSB Side	0xDE	0x00
DSP1_R7_L	TinyDSP1 R7 LSB Side	0xDF	0x00
DSP1_R7_H	TinyDSP1 R7 MSB Side	0xDF	0x00
DSP1MIN0L	TinyDSP1 MIN LSB Side	0x26	0x00
DSP1MIN0H	TinyDSP1 MIN MSB Side	0x26	0x00
DSP1MAX0L	TinyDSP1 MAX LSB Side	0x27	0x00
DSP1MAX0H	TinyDSP1 MAX MSB Side	0x27	0x00
DSP1MIN1L	TinyDSP1 MIN LSB Side	0x2E	0x00
DSP1MIN1H	TinyDSP1 MIN MSB Side	0x2E	0x00
DSP1MAX1L	TinyDSP1 MAX LSB Side	0x2F	0x00
DSP1MAX1H	TinyDSP1 MAX MSB Side	0x2F	0x00
DSP1MIN2L	TinyDSP1 MIN LSB Side	0x36	0x00
DSP1MIN2H	TinyDSP1 MIN MSB Side	0x36	0x00
DSP1MAX2L	TinyDSP1 MAX LSB Side	0x37	0x00
DSP1MAX2H	TinyDSP1 MAX MSB Side	0x37	0x00

14.11.1. DSPnCTRL (TinyDSP n Control Register) (n = 0 to 1)

Register	DSP0CTRL	TinyDSP0 Control Register	Address	0xF780
Register	DSP1CTRL	TinyDSP1 Control Register	Address	0xF800

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DSPE	R/W	0	TinyDSP enable 0: TinyDSP is disabled 1: TinyDSP is enabled Even while the TinyDSP is being disabled, all the registers are still accessible. However, a start trigger of TinyDSP sequence and such will be ignored. When the TinyDSP is enabled, it enters into a trigger wait state, i.e., waiting for a start trigger such as for writing to the Rn.	
6	DSPIE	R/W	0	TinyDSP interrupt enable 0: TinyDSP interrupt is disabled 1: TinyDSP interrupt is enabled	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	DSP_SS	R/C	0	Saturation detection on SFT Read 0: Saturation on SFT is not detected Read 1: Saturation on SFT is detected Write 0: No change Write 1: The bit is cleared	
1	DSP_SA	R/C	0	Saturation detection on ALU Read 0: Saturation on ALU is not detected Read 1: Saturation on ALU is detected Write 0: No change Write 1: The bit is cleared	
0	DSPIF	R/C	0	TinyDSP interrupt flag (event output) Read 0: No TinyDSP interrupt request is generated Read 1: A TinyDSP interrupt event is generated Write 0: No change Write 1: The bit is cleared If the event conditions are satisfied, an event output pulse occurs even when DSPIE = 0. The interrupt flag will be asserted, regardless of the TinyDSP interrupt signal being enabled or disabled.	

14.11.2. DSPnEXEC (TinyDSP n Execution Register) (n = 0 to 1)

Register	DSP0EXEC	TinyDSP0 Execution Register	Address	0xF781
Register	DSP1EXEC	TinyDSP1 Execution Register	Address	0xF801

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	DSP_PC	R/W	0	TinyDSP program counter The program counter (PC) specifies the position of the instructions to be executed next. The CPU can forcibly change the PC at any time.	
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

14.11.3. DSPnDBG (TinyDSP n Debug Register) (n = 0 to 1)

Register	DSP0DBG	TinyDSP0 Debug Register	Address	0xF784
Register	DSP1DBG	TinyDSP1 Debug Register	Address	0xF804

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DSP_DBG	R/W	0	Debug mode 0: Normal mode 1: Debug mode The program sequences of the TinyDSP steps in the debug mode. The step execution will be performed only when 1 is written to the DSP_STP bit.	
6	DSP_STP	R/W	0	Step execution Write 0: No effect Write 1: Step execution The read value is always 0. The step execution will be performed only when 1 is written to the DSP_STP bit with the condition: DSP_DBG = 1. Even if an instruction is waiting for a trigger, the step execution will forcibly be performed. If the DSP_DBG and DSP_STP bits are set at the same time, the step execution will not be performed. When the program counter (PC) specifies address 0x2F during the step execution, the value of the PC will be reset to 0x00 after the instruction has been executed.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

14.11.4. DSPn_Rx_L (TinyDSP n Rx LSB Side) (n = 0 to 1, x = 0 to 7)

Register	DSP0_R0_L	TinyDSP0 R0 LSB Side	Address	0xC4
Register	DSP0_R1_L	TinyDSP0 R1 LSB Side	Address	0xC5
Register	DSP0_R2_L	TinyDSP0 R2 LSB Side	Address	0xC6
Register	DSP0_R3_L	TinyDSP0 R3 LSB Side	Address	0xC7
Register	DSP0_R4_L	TinyDSP0 R4 LSB Side	Address	0xCC
Register	DSP0_R5_L	TinyDSP0 R5 LSB Side	Address	0xCD
Register	DSP0_R6_L	TinyDSP0 R6 LSB Side	Address	0xCE
Register	DSP0_R7_L	TinyDSP0 R7 LSB Side	Address	0xCF
Register	DSP1_R0_L	TinyDSP1 R0 LSB Side	Address	0xD4
Register	DSP1_R1_L	TinyDSP1 R1 LSB Side	Address	0xD5
Register	DSP1_R2_L	TinyDSP1 R2 LSB Side	Address	0xD6
Register	DSP1_R3_L	TinyDSP1 R3 LSB Side	Address	0xD7
Register	DSP1_R4_L	TinyDSP1 R4 LSB Side	Address	0xDC
Register	DSP1_R5_L	TinyDSP1 R5 LSB Side	Address	0xDD
Register	DSP1_R6_L	TinyDSP1 R6 LSB Side	Address	0xDE
Register	DSP1_R7_L	TinyDSP1 R7 LSB Side	Address	0xDF

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DSP_REG	R/W	0	LSB side of the TinyDSP data register For access procedures to the bit, see Section 14.6.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

14.11.5. DSPn_Rx_H (TinyDSP n Rx MSB Side) (n = 0 to 1, x = 0 to 7)

Register	DSP0_R0_H	TinyDSP0 R0 MSB Side	Address	0xC4
Register	DSP0_R1_H	TinyDSP0 R1 MSB Side	Address	0xC5
Register	DSP0_R2_H	TinyDSP0 R2 MSB Side	Address	0xC6
Register	DSP0_R3_H	TinyDSP0 R3 MSB Side	Address	0xC7
Register	DSP0_R4_H	TinyDSP0 R4 MSB Side	Address	0xCC
Register	DSP0_R5_H	TinyDSP0 R5 MSB Side	Address	0xCD
Register	DSP0_R6_H	TinyDSP0 R6 MSB Side	Address	0xCE
Register	DSP0_R7_H	TinyDSP0 R7 MSB Side	Address	0xCF
Register	DSP1_R0_H	TinyDSP1 R0 MSB Side	Address	0xD4
Register	DSP1_R1_H	TinyDSP1 R1 MSB Side	Address	0xD5
Register	DSP1_R2_H	TinyDSP1 R2 MSB Side	Address	0xD6
Register	DSP1_R3_H	TinyDSP1 R3 MSB Side	Address	0xD7
Register	DSP1_R4_H	TinyDSP1 R4 MSB Side	Address	0xDC
Register	DSP1_R5_H	TinyDSP1 R5 MSB Side	Address	0xDD
Register	DSP1_R6_H	TinyDSP1 R6 MSB Side	Address	0xDE
Register	DSP1_R7_H	TinyDSP1 R7 MSB Side	Address	0xDF

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DSP_REG	R/W	0	MSB side of the TinyDSP data register For access procedures to the bit, see Section 14.6.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

14.11.6. DSPn_Rx_L (TinyDSP n Rx LSB Side) (n = 0 to 1, x = 8 to 15)

Register	DSP0_R8_L	TinyDSP0 R8 LSB Side	Address	0xF788	0xF7C0
Register	DSP0_R9_L	TinyDSP0 R9 LSB Side	Address	0xF789	0xF7C2
Register	DSP0_R10_L	TinyDSP0 R10 LSB Side	Address	0xF78A	0xF7C4
Register	DSP0_R11_L	TinyDSP0 R11 LSB Side	Address	0xF78B	0xF7C6
Register	DSP0_R12_L	TinyDSP0 R12 LSB Side	Address	0xF78C	0xF7C8
Register	DSP0_R13_L	TinyDSP0 R13 LSB Side	Address	0xF78D	0xF7CA
Register	DSP0_R14_L	TinyDSP0 R14 LSB Side	Address	0xF78E	0xF7CC
Register	DSP0_R15_L	TinyDSP0 R15 LSB Side	Address	0xF78F	0xF7CE
Register	DSP1_R8_L	TinyDSP1 R8 LSB Side	Address	0xF808	0xF840
Register	DSP1_R9_L	TinyDSP1 R9 LSB Side	Address	0xF809	0xF842
Register	DSP1_R10_L	TinyDSP1 R10 LSB Side	Address	0xF80A	0xF844
Register	DSP1_R11_L	TinyDSP1 R11 LSB Side	Address	0xF80B	0xF846
Register	DSP1_R12_L	TinyDSP1 R12 LSB Side	Address	0xF80C	0xF848
Register	DSP1_R13_L	TinyDSP1 R13 LSB Side	Address	0xF80D	0xF84A
Register	DSP1_R14_L	TinyDSP1 R14 LSB Side	Address	0xF80E	0xF84C
Register	DSP1_R15_L	TinyDSP1 R15 LSB Side	Address	0xF80F	0xF84E

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DSP_REG	R/W	0	LSB side of the TinyDSP data register For access procedures to the bit, see Section 14.6.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

14.11.7. DSPn_Rx_H (TinyDSP n Rx MSB Side) (n = 0 to 1, x = 8 to 15)

Register	DSP0_R8_H	TinyDSP0 R8 MSB Side	Address	0xF788	0xF7C1
Register	DSP0_R9_H	TinyDSP0 R9 MSB Side	Address	0xF789	0xF7C3
Register	DSP0_R10_H	TinyDSP0 R10 MSB Side	Address	0xF78A	0xF7C5
Register	DSP0_R11_H	TinyDSP0 R11 MSB Side	Address	0xF78B	0xF7C7
Register	DSP0_R12_H	TinyDSP0 R12 MSB Side	Address	0xF78C	0xF7C9
Register	DSP0_R13_H	TinyDSP0 R13 MSB Side	Address	0xF78D	0xF7CB
Register	DSP0_R14_H	TinyDSP0 R14 MSB Side	Address	0xF78E	0xF7CD
Register	DSP0_R15_H	TinyDSP0 R15 MSB Side	Address	0xF78F	0xF7CF
Register	DSP1_R8_H	TinyDSP1 R8 MSB Side	Address	0xF808	0xF841
Register	DSP1_R9_H	TinyDSP1 R9 MSB Side	Address	0xF809	0xF843
Register	DSP1_R10_H	TinyDSP1 R10 MSB Side	Address	0xF80A	0xF845
Register	DSP1_R11_H	TinyDSP1 R11 MSB Side	Address	0xF80B	0xF847
Register	DSP1_R12_H	TinyDSP1 R12 MSB Side	Address	0xF80C	0xF849
Register	DSP1_R13_H	TinyDSP1 R13 MSB Side	Address	0xF80D	0xF84B
Register	DSP1_R14_H	TinyDSP1 R14 MSB Side	Address	0xF80E	0xF84D
Register	DSP1_R15_H	TinyDSP1 R15 MSB Side	Address	0xF80F	0xF84F

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DSP_REG	R/W	0	MSB side of the TinyDSP data register For access procedures to the bit, see Section 14.6.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

14.11.8. DSPn_ACC_x (TinyDSP n ACC) (n = 0 to 1, x = 0 to 4)

Register	DSP0_ACC_0	TinyDSP0 ACC[7:0]	Address	0xF790
Register	DSP0_ACC_1	TinyDSP0 ACC[15:8]	Address	0xF791
Register	DSP0_ACC_2	TinyDSP0 ACC[23:15]	Address	0xF792
Register	DSP0_ACC_3	TinyDSP0 ACC[31:24]	Address	0xF793
Register	DSP0_ACC_4	TinyDSP0 ACC[36:32]	Address	0xF794
Register	DSP1_ACC_0	TinyDSP1 ACC[7:0]	Address	0xF810
Register	DSP1_ACC_1	TinyDSP1 ACC[15:8]	Address	0xF811
Register	DSP1_ACC_2	TinyDSP1 ACC[23:15]	Address	0xF812
Register	DSP1_ACC_3	TinyDSP1 ACC[31:24]	Address	0xF813
Register	DSP1_ACC_4	TinyDSP1 ACC[36:32]	Address	0xF814

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DSP_ACC	R/W	0	TinyDSP accumulator (ACC) The register configures the accumulator (ACC) of each TinyDSP. Each byte lane of the ACC is assigned to an independent address.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

14.11.9. DSPn_PRG_DATL/H (TinyDSP n Program Memory LSB/MSB Side) (n = 0 to 1)

Register	DSP0_PRG_DATL	TinyDSP0 Program Memory LSB Side	Address	0xF7A0
Register	DSP0_PRG_DATH	TinyDSP0 Program Memory MSB Side	Address	0xF7A1
Register	DSP1_PRG_DATL	TinyDSP1 Program Memory LSB Side	Address	0xF820
Register	DSP1_PRG_DATH	TinyDSP1 Program Memory MSB Side	Address	0xF821

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DSP_PRG_D	R/W	0	TinyDSP Program/Data Memory window Reading from/writing to the bit is based on how the DSPn_PRG_ADR register is set. For instance, when DSPn_PRG_ADR = 0x10, bits[7:0] of the LSB side of the data, which resides in address 0x10 of the Program/Data Memory, are read from the DSPn_PRG_DATL register.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

14.11.10. DSPn_PRG_ADR (TinyDSP n Program/Data Memory Address) (n = 0 to 1)

Register	DSP0_PRG_ADR	TinyDSP0 Program/Data Memory Address	Address	0xF7A2
Register	DSP1_PRG_ADR	TinyDSP1 Program/Data Memory Address	Address	0xF822

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	DSP_PRG_A	R/W	0	TinyDSP Program/Data Memory address The DSPn_PRG_DATL/H registers access the Program/Data Memory with using the value of the bits.	
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

14.11.11. DSPnTRG (TinyDSP n Execution Trigger Status) (n = 0 to 1)

A TinyDSP instruction with the TRIG_WAIT flag will be paused before its execution. During the pause, the SET_R0 to SET_R7 bits are monitored according to the TRIG_WHAT bit. When the corresponding bit of the DSPnTRG register is set (i.e., when an update to the data registers, R0 to R7, is detected), the TinyDSP instruction will be executed and move on to the next sequence. Then, the corresponding bit, SET_Rx, is automatically cleared.

Basically, the DSPnTRG register indicates only a trigger status; therefore, the CPU does not have to access this register. However, the CPU is able to access the register to write to/read from for debugging and re-initializing.

Register	DSP0TRG	TinyDSP0 Execution Trigger Status	Address	0xF782
Register	DSP1TRG	TinyDSP1 Execution Trigger Status	Address	0xF802

Bit	Bit Name	R/W	Initial	Description	Remarks
7	SET_R7	R/W	0	TinyDSP execution trigger status Each bit indicates whether its corresponding data register (R0 to R7) is written or not. If the CPU, DSA, or EPU writes a value to R0 to R7 when DSPnCTRL.DSPE = 1, the corresponding bit is automatically set. For more details, see Section 14.5.	
6	SET_R6	R/W	0		
5	SET_R5	R/W	0		
4	SET_R4	R/W	0		
3	SET_R3	R/W	0		
2	SET_R2	R/W	0		
1	SET_R1	R/W	0		
0	SET_R0	R/W	0		

14.11.12. DSPnRST (TinyDSP n Access Counter Clear Register) (n = 0 to 1)

Register	DSP0RST	TinyDSP0 Access Counter Clear Register	Address	0xF783
Register	DSP1RST	TinyDSP1 Access Counter Clear Register	Address	0xF803

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CPUACCLA	R/C	0	Clearing CPU access counters of SFR BUS and XDATA BUS Write 0: No change Write 1: CPU access counters of SFR BUS and XDATA BUS are cleared The read value is always 0.	
6	DSACACCLA	R/C	0	Clearing SFR BUS DSAC access counter Write 0: No change Write 1: SFR BUS DSAC access counter is cleared The read value is always 0.	
5	EPUCACCLA	R/C	0	Clearing SFR BUS EPU access counter Write 0: No change Write 1: SFR BUS EPU access counter is cleared The read value is always 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

14.11.13. DSPnCTRL2 (TinyDSP n Control2 Register) (n = 0 to 1)

Register	DSP0CTRL2	TinyDSP0 Control2 Register	Address	0xF785
Register	DSP1CTRL2	TinyDSP1 Control2 Register	Address	0xF805

Bit	Bit Name	R/W	Initial	Description	Remarks
7	HPDIV	R/W	0	DIV mode setting 0: Normal mode 1: High accuracy mode	
6	EXRSF	R/W	0	RSF instruction extension 0: Trigger function is enabled; the allowable range for shifting right is from 0 to 15 1: Trigger function is disabled; the allowable range for shifting right is from 0 to 31	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	LDEN	R/W	0	Enabling LDR function and LDD instruction 0: LDR function and LDD instruction are disabled 1: LDR function and LDD instruction are enabled	
0	CNSTSEL	R/W	0	Setting CVR range 0: C0 to C7 are assigned to FIELD B = 8 to 15 1: C0 to C7 are assigned to FIELD B = 0 to 7 Enabling the CVR mode is defined by the DSPnCNSTEN register. For more details, see Section 14.9.1.	

14.11.14. DSPnCNSTEN (TinyDSP n CVR Enable Register) (n = 0 to 1)

Register	DSP0CNSTEN	TinyDSP0 CVR Enable Register	Address	0xF786
Register	DSP1CNSTEN	TinyDSP1 CVR Enable Register	Address	0xF806

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CNSTEN7	R/W	0	Enabling C7 of CVR function 0: C7 of CVR function is disabled 1: C7 of CVR function is enabled	
6	CNSTEN6	R/W	0	Enabling C6 of CVR function 0: C6 of CVR function is disabled 1: C6 of CVR function is enabled	
5	CNSTEN5	R/W	0	Enabling C5 of CVR function 0: C5 of CVR function is disabled 1: C5 of CVR function is enabled	
4	CNSTEN4	R/W	0	Enabling C4 of CVR function 0: C4 of CVR function is disabled 1: C4 of CVR function is enabled	
3	CNSTEN3	R/W	0	Enabling C3 of CVR function 0: C3 of CVR function is disabled 1: C3 of CVR function is enabled	
2	CNSTEN2	R/W	0	Enabling C2 of CVR function 0: C2 of CVR function is disabled 1: C2 of CVR function is enabled	
1	CNSTEN1	R/W	0	Enabling C1 of CVR function 0: C1 of CVR function is disabled 1: C1 of CVR function is enabled	
0	CNSTEN0	R/W	0	Enabling C0 of CVR function 0: C0 of CVR function is disabled 1: C0 of CVR function is enabled	

14.11.15. DSPn_Cx_L (TinyDSP n Cx LSB Side) (n = 0 to 1, x = 0 to 7)

Register	DSP0_C0_L	TinyDSP0 C0 LSB Side	Address	0xF798	0xF7D0
Register	DSP0_C1_L	TinyDSP0 C1 LSB Side	Address	0xF799	0xF7D2
Register	DSP0_C2_L	TinyDSP0 C2 LSB Side	Address	0xF79A	0xF7D4
Register	DSP0_C3_L	TinyDSP0 C3 LSB Side	Address	0xF79B	0xF7D6
Register	DSP0_C4_L	TinyDSP0 C4 LSB Side	Address	0xF79C	0xF7D8
Register	DSP0_C5_L	TinyDSP0 C5 LSB Side	Address	0xF79D	0xF7DA
Register	DSP0_C6_L	TinyDSP0 C6 LSB Side	Address	0xF79E	0xF7DC
Register	DSP0_C7_L	TinyDSP0 C7 LSB Side	Address	0xF79F	0xF7DE
Register	DSP1_C0_L	TinyDSP1 C0 LSB Side	Address	0xF818	0xF850
Register	DSP1_C1_L	TinyDSP1 C1 LSB Side	Address	0xF819	0xF852
Register	DSP1_C2_L	TinyDSP1 C2 LSB Side	Address	0xF81A	0xF854
Register	DSP1_C3_L	TinyDSP1 C3 LSB Side	Address	0xF81B	0xF856
Register	DSP1_C4_L	TinyDSP1 C4 LSB Side	Address	0xF81C	0xF858
Register	DSP1_C5_L	TinyDSP1 C5 LSB Side	Address	0xF81D	0xF85A
Register	DSP1_C6_L	TinyDSP1 C6 LSB Side	Address	0xF81E	0xF85C
Register	DSP1_C7_L	TinyDSP1 C7 LSB Side	Address	0xF81F	0xF85E

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DSP_REG	R/W	0	LSB side of the TinyDSP constant register For access procedures to the bit, see Section 14.6.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

14.11.16. DSPn_Cx_H (TinyDSP n C x MSB Side) (n = 0 to 1, x = 0 to 7)

Register	DSP0_C0_H	TinyDSP0 C0 MSB Side	Address	0xF798	0xD7D1
Register	DSP0_C1_H	TinyDSP0 C1 MSB Side	Address	0xF799	0xD7D3
Register	DSP0_C2_H	TinyDSP0 C2 MSB Side	Address	0xF79A	0xD7D5
Register	DSP0_C3_H	TinyDSP0 C3 MSB Side	Address	0xF79B	0xD7D7
Register	DSP0_C4_H	TinyDSP0 C4 MSB Side	Address	0xF79C	0xD7D9
Register	DSP0_C5_H	TinyDSP0 C5 MSB Side	Address	0xF79D	0xD7DB
Register	DSP0_C6_H	TinyDSP0 C6 MSB Side	Address	0xF79E	0xD7DD
Register	DSP0_C7_H	TinyDSP0 C7 MSB Side	Address	0xF79F	0xD7DF
Register	DSP1_C0_H	TinyDSP1 C0 MSB Side	Address	0xF818	0xD851
Register	DSP1_C1_H	TinyDSP1 C1 MSB Side	Address	0xF819	0xD853
Register	DSP1_C2_H	TinyDSP1 C2 MSB Side	Address	0xF81A	0xD855
Register	DSP1_C3_H	TinyDSP1 C3 MSB Side	Address	0xF81B	0xD857
Register	DSP1_C4_H	TinyDSP1 C4 MSB Side	Address	0xF81C	0xD859
Register	DSP1_C5_H	TinyDSP1 C5 MSB Side	Address	0xF81D	0xD85B
Register	DSP1_C6_H	TinyDSP1 C6 MSB Side	Address	0xF81E	0xD85D
Register	DSP1_C7_H	TinyDSP1 C7 MSB Side	Address	0xF81F	0xD85F

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DSP_REG	R/W	0	MSB side of the TinyDSP constant register For access procedures to the bit, see Section 14.6.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

14.11.17. DSPnLDA (TinyDSP n LDR Load Address) (n = 0 to 1)

Register	DSP0LDA	TinyDSP0 LDR Load Address	Address	0xF7A3
Register	DSP1LDA	TinyDSP1 LDR Load Address	Address	0xF823

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	LDADR5	R	1	The read value is 1. The write value must always be 1.	
4	LDADR4	R	0	The read value is 0. The write value must always be 0.	
3	LDADR	R/W	0	Bits 0 to 3 of the pointer which stores the address of data to be loaded next in the LDR function.	
2		R/W	0		
1		R/W	0		
0		R/W	0		

14.11.18. DSPnMAXxL (TinyDSP n MAX x LSB Side) (n = 0 to 1, x = 0 to 2)

Register	DSP0MAX0L	TinyDSP0 MAX0 LSB Side	Address	0xF7A8	0xF7E0	0x25
Register	DSP0MAX1L	TinyDSP0 MAX1 LSB Side	Address	0xF7AA	0xF7E4	0x2D
Register	DSP0MAX2L	TinyDSP0 MAX2 LSB Side	Address	0xF7AC	0xF7E8	0x35
Register	DSP1MAX0L	TinyDSP1 MAX0 LSB Side	Address	0xF828	0xF860	0x27
Register	DSP1MAX1L	TinyDSP1 MAX1 LSB Side	Address	0xF82A	0xF864	0x2F
Register	DSP1MAX2L	TinyDSP1 MAX2 LSB Side	Address	0xF82C	0xF868	0x37

Bit	Bit Name	R/W	Initial	Description	Remarks
7	MAX	R/W	0	LSB side of the maximum value of the MMX instruction For access procedures to the bit, see Section 14.6.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

14.11.19. DSPnMAXxH (TinyDSP n MAX x MSB Side) (n = 0 to 1, x = 0 to 2)

Register	DSP0MAX0H	TinyDSP0 MAX0 MSB Side	Address	0xF7A8	0xF7E1	0x25
Register	DSP0MAX1H	TinyDSP0 MAX1 MSB Side	Address	0xF7AA	0xF7E5	0x2D
Register	DSP0MAX2H	TinyDSP0 MAX2 MSB Side	Address	0xF7AC	0xF7E9	0x35
Register	DSP1MAX0H	TinyDSP1 MAX0 MSB Side	Address	0xF828	0xF861	0x27
Register	DSP1MAX1H	TinyDSP1 MAX1 MSB Side	Address	0xF82A	0xF865	0x2F
Register	DSP1MAX2H	TinyDSP1 MAX2 MSB Side	Address	0xF82C	0xF869	0x37

Bit	Bit Name	R/W	Initial	Description	Remarks
7	MAX	R/W	0	MSB side of the maximum value of the MMX instruction For access procedures to the bit, see Section 14.6.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

14.11.20. DSPnMINxL (TinyDSP n MIN x LSB Side) (n = 0 to 1, x = 0 to 2)

Register	DSP0MIN0L	TinyDSP0 MIN0 LSB Side	Address	0xF7A9	0xF7E2	0x24
Register	DSP0MIN1L	TinyDSP0 MIN1 LSB Side	Address	0xF7AB	0xF7E6	0x2C
Register	DSP0MIN2L	TinyDSP0 MIN2 LSB Side	Address	0xF7AD	0xF7EA	0x34
Register	DSP1MIN0L	TinyDSP1 MIN0 LSB Side	Address	0xF829	0xF862	0x26
Register	DSP1MIN1L	TinyDSP1 MIN1 LSB Side	Address	0xF82B	0xF866	0x2E
Register	DSP1MIN2L	TinyDSP1 MIN2 LSB Side	Address	0xF82D	0xF86A	0x36

Bit	Bit Name	R/W	Initial	Description	Remarks
7	MIN	R/W	0	LSB side of the minimum value of the MMX instruction For access procedures to the bit, see Section 14.6.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

14.11.21. DSPnMINxH (TinyDSP n MIN x MSB Side) (n = 0 to 1, x = 0 to 2)

Register	DSP0MIN0H	TinyDSP0 MIN0 MSB Side	Address	0xF7A9	0xF7E3	0x24
Register	DSP0MIN1H	TinyDSP0 MIN1 MSB Side	Address	0xF7A9	0xF7E7	0x2C
Register	DSP0MIN2H	TinyDSP0 MIN2 MSB Side	Address	0xF7AB	0xF7EB	0x34
Register	DSP1MIN0H	TinyDSP1 MIN0 MSB Side	Address	0xF82B	0xF863	0x26
Register	DSP1MIN1H	TinyDSP1 MIN1 MSB Side	Address	0xF82D	0xF867	0x2E
Register	DSP1MIN2H	TinyDSP1 MIN2 MSB Side	Address	0xF82D	0xF86B	0x36

Bit	Bit Name	R/W	Initial	Description	Remarks
7	MIN	R/W	0	MSB side of the minimum value of MMX instruction For access procedures to the bit, see Section 14.6.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

14.12. Usage Notes and Restrictions

14.12.1. DSP_SS Bit Asserted by DIV Instruction

In a DIV instruction, the SFT is used for calculating the reciprocals of Rm. Therefore, when a divisor Rm is 1, the DSPnCTRL.DSP_SS bit is asserted.

14.12.2. Restrictions on Rewriting an Argument during DIV Instruction Execution

Do not rewrite the argument of a DIV instruction which is being executed. If such rewriting has occurred, the TinyDSPs cannot compute solutions as expected. When directly using a value written by the CPU or DSAC for the argument of a DIV instruction, execute an MVC instruction before the DIV instruction.

14.12.3. Setting the MMX Instruction

When using an MMX instruction, define the values of the DSPnMAX registers to be higher than the values of the DSPnMIN registers. If the DSPnMIN registers are higher than the DSPnMAX registers, Rn will not be compared with the DSPnMAX and DSPnMIN registers properly in using the MMX instruction.

15. High-resolution PWM

15.1. Overview

Table 15-1 shows the high-resolution PWM functional descriptions. The PWM module generates 8 (4 pairs) high-resolution pulse width modulation (PWM) signals. Each pair can generate PWM signals without high level overlapping. The PWM module receives internal event signals from other modules, and uses them for re-triggering the output signals or counter operation. In addition, the PWM module generates interrupts and LSI internal events according to the internal compare match state.

Table 15-1. High-resolution PWM Functional Descriptions

Item	Description
Number of Channels	4 channels
PWM Output	8 outputs (2 output per channel)
Minimum Resolution	1.04 ns
Minimum Cycle Resolution	8.32 ns
Minimum Cycle	16.64 ns
Duty Setting Range	0% to 100%
Waveform Generation Counter	Number of bits: 16 bits Counting direction: Up, and up-down Synchronization starts between the counters of different channels.
Compare Match Register	4 general-purpose registers (A, B, C, and D) 2 cycle setting registers (MIN and MAX)
Mode	2 modes Compare match or dead-time insertion
Number of Event Outputs	2 general-purpose events (for each channel) 1 TMR-dedicated event (for each channel)
Interrupt	2 interrupts (for each channel)
Event/Interrupt Output Source	Pin change event generated by compare match, or re-trigger generation More than one source can be set for one event.
Re-trigger Operation	Source events of re-trigger: 29 edge events (such as CMP, CPU, and EPU) 11 level events (CMP, CMPLUT, and GPIO) 5 operations (A, B, C, D, and re-trigger masking operation)

15.2. Block Diagram

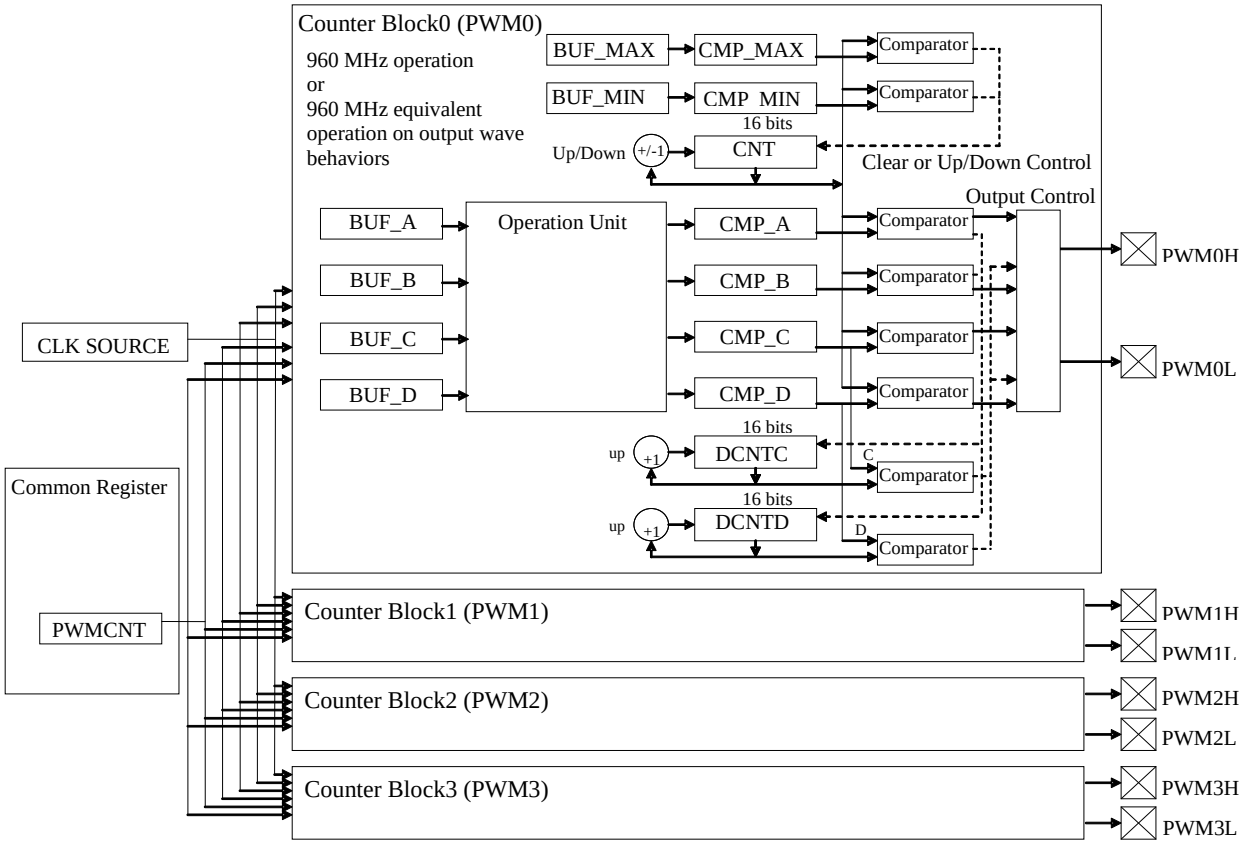


Figure 15-1. High-resolution PWM Block Diagram

15.3. Resources

As shown in Figure 15-1, the PWM module has 4 counter blocks. Each block has the following resources.

● **16-bit up/up-down Counter (CNT)**

The maximum frequency of each counter is 960 MHz. More than one counter between the counter blocks can be synchronized with each other. Counter operation has 2 modes (up mode and up-down mode). In the up mode, the counter always counts up. When the counter value reaches the value of the CMP_MAX register, the counter value is reloaded from the CMP_MIN register. In the up-down mode, the counter counts up until it reaches the value of the CMP_MAX register. When the counter value reaches the CMP_MAX value, the counter is counted down until it reaches the value of the CMP_MIN register. When the counter value reaches the CMP_MIN value, the counter is counted up again.

Since the values of the CMP_MIN and CMP_MAX registers exist, a user can regard a counter value not only as an unsigned value but also as a signed value.

The lower 3 bits of the CMP_MIN register must be set to 0b000, and the lower 3 bits of the CMP_MAX register must be set to 0b111. Moreover, the difference between the CMP-MIN and CMP_MAX values of all the channels must be 8 or more (i.e., $\mathrm{CMP\_MAX} - \mathrm{CMP\_MIN} \geq 8$). Therefore, the performances of the PWM module are as follows:

- Minimum PWM cycle: 16.64 ns
- Minimum resolution of PWM cycle: 8.32 ns
- Minimum resolution of PWM duty cycle (edge difference of PWMnL and PWMnH output signals): 1.04 ns

● **Compare Match Registers A and B (CMP_A and CMP_B)**

The CMP_A and CMP_B registers control the operation of the PWMnH/PWMnL signals. When the CNT matches the value of each compare match register, the output signal of the PWMnH or PWMnL is set/reset in accordance with the setting. The CMP_A and CMP_B registers can be rewritten at any time as needed; however, using the buffer mode is recommended to update the registers.

● **Compare Match Registers C and D (CMP_C and CMP_D)**

- In PWM mode 0: The CMP_C and CMP_D registers control the operation of the PWMnH/PWMnL signals.
- In PWM mode 1: The CMP_C and CMP_D registers are compared with the dead time counters (DCNTC and DCNTD). These dead time counters are used to insert dead time automatically.

● **Compare Match Registers MAX and MIN (CMP_MAX and CMP_MIN)**

The CMP_MAX and CMP_MIN registers define the period of the counters and the range of the counter values.

In the up mode, when the counter value matches the value of the CMP_MAX register, the counter value is reloaded from the CMP_MIN register. In the up-down mode, when the counter value reaches the value of the CMP_MAX register, the up mode changes to the down-count mode. When the counter value reaches the value of the CMP_MIN register, the down-count mode changes to the up mode again. Although the CMP_MAX and CMP_MIN registers can be rewritten at any time as needed, it is recommended to use the buffer mode to update the registers.

The lower 3 bits of the CMP_MAX register is fixed to 0b111, and the lower 3 bits of the CMP_MIN register is fixed to 0b000. The PWM cycle is the multiple numbers of 8.

● **Buffer Registers A, B, C, D, MAX, and MIN (BUF_A, BUF_B, BUF_C, BUF_D, BUF_MAX, and BUF_MIN)**

These registers are for the buffer mode. The value of the each buffer register is transferred to the compare match register at a specific timing (for details of the operation, see Section 15.4).

The lower 3 bits of the BUF_MAX register must be fixed to 0b111, and the lower 3 bits of the BUF_MIN register must be fixed to 0b000. The PWM cycle is the multiple numbers of 8.

● **16-bit dead-time Counters (DCNTC and DCNTD)**

The DCNTC and DCNTD counters are prepared to generate the automatic dead-time period when the output of PWMnH or PWMnL changes. The dead-time period is determined by the values of the CMP_C and CMP_D registers.

15.4. Operation

Table 15-2 shows the operation mode of the high-resolution PWM, and Table 15-3 shows the method to determine the next compare match register, CMP_xx, in the buffer mode. In addition, the overview of the operation mode of the high-resolution PWM is as follows:

- The PWM mode has 2 operation modes: PWM mode 0 and PWM mode 1
- Either the direct mode or the buffer mode can be selected by the compare match registers.
- The outline operation of the 2 PWM modes is as follows:

- PWM mode 0: The PWM signals are generated without using any dead-time counters.
- PWM mode 1: The PWM signals with dead time that is automatically inserted by the dead-time counter (DCNTC/DCNTD) are generated.

In the buffer mode, the constant period (3 CPU clock cycles + 40 PWM clock cycles or more) from the setting of the buffer register to the updating of the compare match register must be needed.

Table 15-2. Operation Mode of High-resolution PWM

Mode			Update Timing of CMP_xx	Next CMP_xx	Operation of DCNTC	Operation of DCNTD	Condition to Change Output Level by Setting Value	
							PWMnH	PWMnL
PWM Mode 0	Direct Mode	Up Mode	When CPU writes	Updated by CPU's writing value	Stop	Stop	VH0 (CNT==CMP_C) or VH1 (CNT==CMP_B)	VL0 (CNT==CMP_A) or VL1 (CNT==CMP_D)
		Up-down Mode			Stop	Stop	VH0 (CNT(UP) ==CMP_C) or VH1 (CNT(DN) ==CMP_C)	VL0 (CNT (UP) ==CMP_A) or VL1 (CNT (DN) ==CMP_A)
	Buffer Mode	Up Mode	When CNT reaches CMP_MAX (1)	See Table 15-3	Stop	Stop	VH0 (CNT==CMP_C) or VH1 (CNT==CMP_B)	VL0 (CNT==CMP_A) or VL1 (CNT==CMP_D)
		Up-down Mode	When changing counting direction (2)		Stop	Stop	VH0 (CNT(UP) ==CMP_C) or VH1 (CNT(DN) ==CMP_C)	VL0 (CNT(UP) ==CMP_A) or VL1 (CNT (DN) ==CMP_A)
PWM Mode 1	Direct Mode	Up Mode	When CPU writes	Updated by CPU's writing value	When starting: (CNT==CMP_A) When stopping/clearing: (DCNTC==CMP_C)	When starting: (CNT==CMP_B) When stopping/clearing: (DCNTD==CMP_D)	VH0 (DCNTC==CMP_C) or VH1 (CNT==CMP_B)	VL0 (CNT==CMP_A) or VL1 (DCNTD==CMP_D)
		Up-down Mode			When starting: (CNT (UP) ==CMP_A) When stopping/clearing: (DCNTC==CMP_C)	When starting: (CNT(DN) ==CMP_A) When stopping/clearing: (DCNTD==CMP_D)	VH0 (DCNTC==CMP_C) or VH1 (CNT(DN) ==CMP_A)	VL0 (CNT(UP) ==CMP_A) or VL1 (DCNTD==CMP_D)
	Buffer Mode	Up Mode	When CNT reaches CMP_MAX (1)	See Table 15-3	When starting: (CNT==CMP_A) When stopping/clearing: (DCNTC==CMP_C)	When starting: (CNT==CMP_B) When stopping/clearing: (DCNTD==CMP_D)	VH0 (DCNTC==CMP_C) or VH1 (CNT==CMP_B)	VL0 (CNT==CMP_A) or VL1 (DCNTD==CMP_D)
		Up-down Mode	When changing counting direction (2)		When starting: (CNT(UP) ==CMP_A) When stopping/clearing: (DCNTC==CMP_C)	When starting: (CNT(DN) ==CMP_A) When stopping/clearing: (DCNTD==CMP_D)	VH0 (DCNTC==CMP_C) or VH1 (CNT(DN) ==CMP_A)	VL0 (CNT(UP) ==CMP_A) or VL1 (DCNTD==CMP_D)

(1) At this timing, the CNT reloads the CMP_MIN register.
(2) Both or either of CNT==CMP_MAX or CNT==CMP_MIN can be selected.

Table 15-3. Determination of Next Compare Match Register in Buffer Mode in Each PWM Mode

PWM Mode	Next CMP_MAX/ CMP_MIN	Next CMP_A	Next CMP_B	Next CMP_C	Next CMP_D
0	BUF_MAX/ BUF_MIN	BUF_A	BUF_B	BUF_C	BUF_D
1	BUF_MAX/ BUF_MIN	BUF_A	BUF_B	BUF_C	BUF_D

15.4.1. Direct Mode and Buffer Mode

15.4.1.1. Direct Mode

The values of all the compare match registers must be updated directly by the CPU, EPU, or DSAC. When the compare match register is rewritten by the CPU, etc., the value is immediately compared with the CNT, DCNTC, or DCNTD. Therefore, attention must be paid since unintended operation may occur depending on the timing of rewriting. In the direct mode, the setting of the buffer register does not affect PWM's operation.

15.4.1.2. Buffer Mode

In the up mode, when the CNT matches the value of the CMP_MAX register and reloads the value of the CMP_MIN register, each value of all buffer registers is transferred to the corresponding compare match register.

In the up-down mode, when the CNT matches the value of the CMP_MAX register and starts counting down, or the CNT matches the value of the CMP_MIN register and starts counting up, each value of all buffer registers is transferred to the corresponding compare match register. The value of the buffer register is transferred to compare match register at 2 timings, and a user can select both or either of them.

It is normally recommended to use the buffer mode. The compare match register can be rewritten directly in the buffer mode. The value is immediately compared with the CNT, just as in direct mode.

15.4.2. PWM Mode 0

Set all the change timings of the PWM output waveform.

In the PWM mode 0, the dead-time counter is not used. In addition, the PWM mode 0 can be operated in both the direct mode and buffer mode.

15.4.2.1. PWM Mode 0 (Up Mode)

The CNT is counted up from the CNT's initial value to the CMP_MAX register value. When the CNT matches the CMP_MAX register value, the CNT loads the CMP_MIN register value and counts up from the CMP_MIN register value to the CMP_MAX register value.

The operations when the CNT matches the compare match register are as follows. Select each level (VH0, VH1, VL0, or VL1) from no change, low state, high state, or toggle operation.

- When the CNT matches the CMP_A register value, the PWMnL output is changed to the level determened by the PWMnLCR1.VL0 bits.
- When the CNT matches the CMP_C register value, the PWMnH output is changed to the level determened by the PWMnHCR1.VH0 bits.
- When the CNT matches the CMP_B register value, the PWMnH output is changed to the level determened by the PWMnHCR1.VH1 bits.
- When the CNT matches the CMP_D register value, the PWMnL output is changed to the level determened by the PWMnLCR1.VL1 bits.

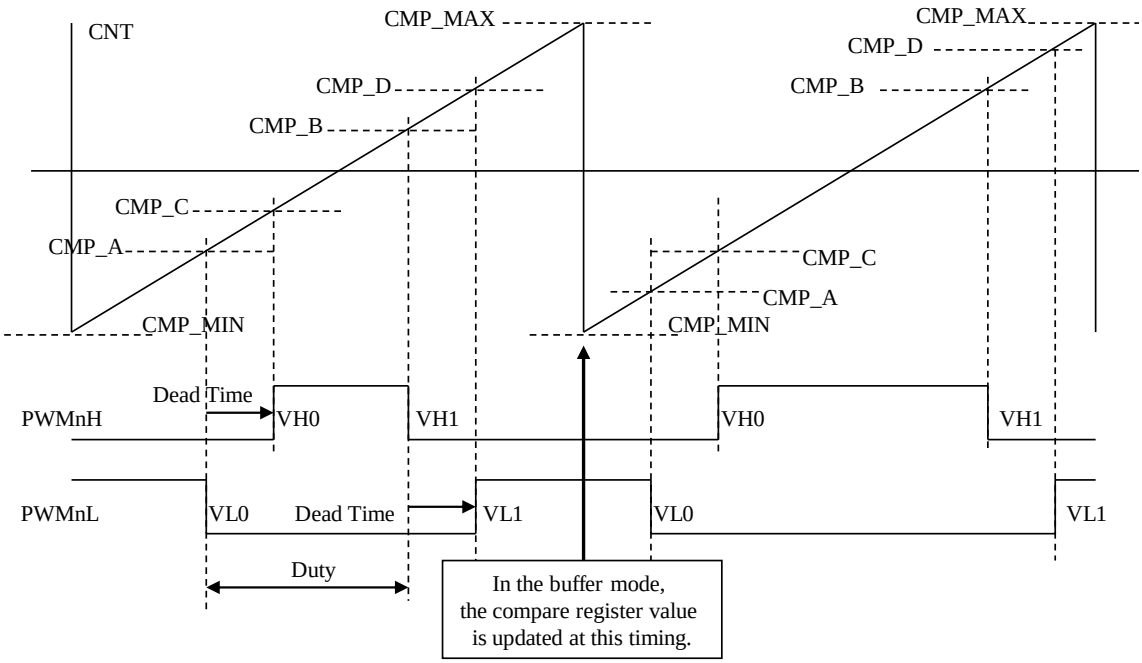


Figure 15-2. PWM Mode 0 (Up Mode)

15.4.2.2. PWM Mode 0 (Up-down Mode)

The CNT is counted up from the CNT's initial value to the CMP_MAX register value. When the CNT matches the CMP_MAX register value, the CNT is counted down to the CMP_MIN register value. When the CNT matches the CMP_MIN register value, the CNT is counted up to the CMP_MAX register value. When the PWMCNTS.PWMCSn bit is cleared to stop counting, the CNT when the counting is restarted counts up regardless of the direction of counting before the counter is stopped.

The operations when the CNT matches the value of the compare match register are shown below. Each level (VH0, VH1, VL0, or VL1) is selected from no change, low state, high state, or toggle operation.

- When the CNT matches the CMP_A register value during counting up, the PWMnL output is changed to the level specified by the PWMnLCR1.VL0 bits.
- When the CNT matches the CMP_C register value during counting up, the PWMnH output is changed to the level specified by the PWMnHCR1.VH0 bits.
- When the CNT matches the CMP_C register value during counting down, the PWMnH output is changed to the level specified by the PWMnHCR1.VH1 bits.
- When the CNT matches the CMP_A register value during counting down, the PWMnL output is changed to the level specified by the PWMnLCR1.VL1 bits.

When using the up-down mode in the buffer mode, the value of the buffer register is transferred to the compare match register at 2 timings. Both or either of those timings can be selected.

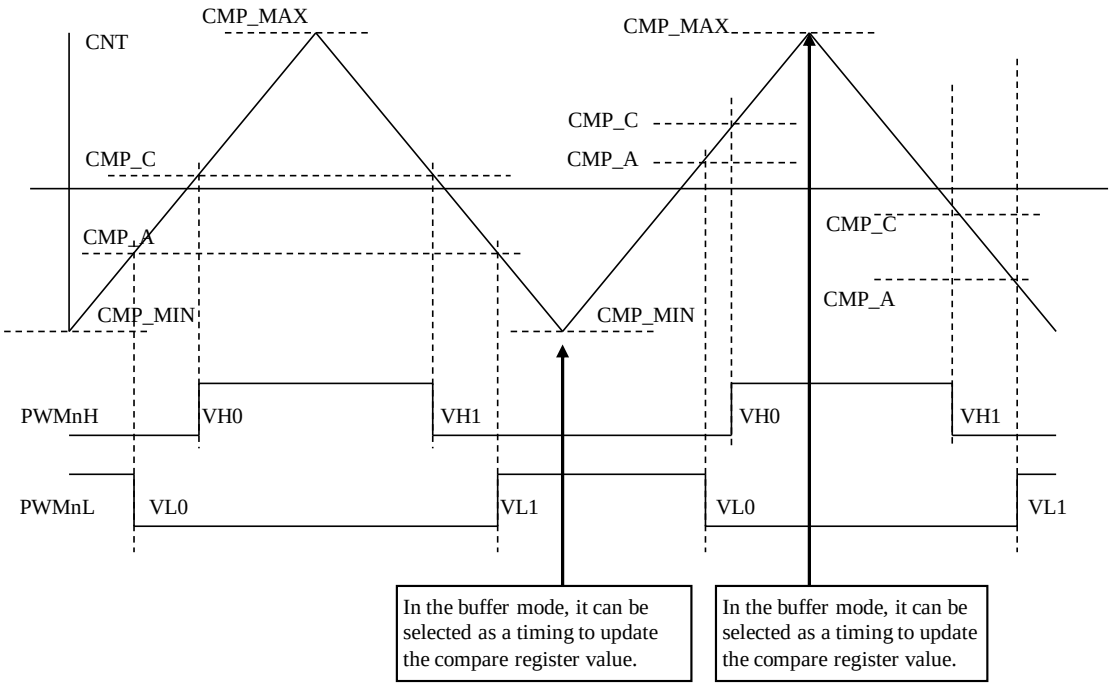


Figure 15-3. PWM Mode 0 (Up-down Mode)

15.4.3. PWM Mode 1 (Automatic Dead Time)

Dead time is set automatically by the dead-time counter (DCNTC and DCNTD). The dead-time value must be defined by both the CMP_C and CMP_D registers. The setting method of the CMP_C and CMP_D registers is different from that of the PWM mode 0.

15.4.3.1. PWM Mode 1 (Up Mode)

The CNT is counted up from the CNT's initial value to the CMP_MAX register value. When the CNT matches the CMP_MAX register value, the CNT loads the CMP_MIN register value and counts up from the CMP_MIN register value to the CMP_MAX register value.

The operations of the PWM mode 1 in the up mode are shown below.

- When the CNT matches the CMP_A register value, the PWMnL output is changed to the level determened by the PWMnLCR1.VL0 bits, and the DCNTC counts up from zero.
- When the DCNTC matches the CMP_C register value, the PWMnH output is changed to the level determened by the PWMnHCR1.VH0 bits. Then, the DCNTC is cleared, and counting is stopped.
- When the CNT matches the CMP_B register value, the PWMnH output is changed to the level determened by the PWMnHCR1.VH1 bits, and the DCNTD counts up from zero.
- When the DCNTD matches the CMP_D register value, the PWMnL output is changed to the level determened by the PWMnLCR1.VL1 bits. Then, the DCNTD is cleared, and counting is stopped.

In the direct mode, the period from the setting of the compare match register to the compare mach generation must be needed for a constant period (3 CPU clock cycles + 40 PWM counting cycles or more). In the buffer mode, each compare match register is changed when the CNT loads the CMP_MIN register value.

When the CNT matches the CMP_A/B register value during counting-up of the DCNTC/D, the DCNTC/D counts up again from zero. As a result, dead time becomes longer than the setting value. When CMP_C = 0 or CMP_D = 0, dead time is one PWM counter cycle.

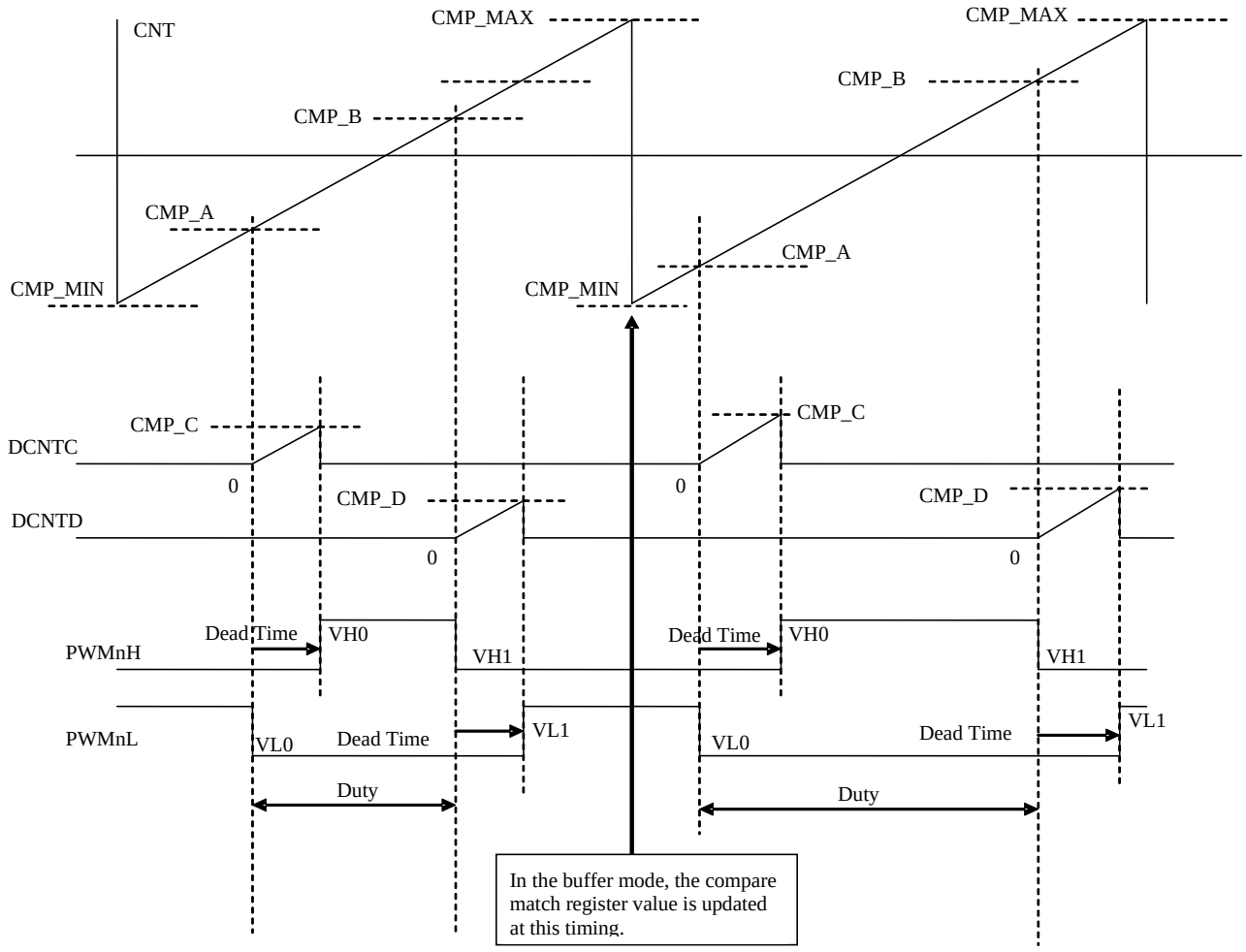


Figure 15-4. PWM Mode 1 (Up Mode)

15.4.3.2. PWM Mode 1 (Up-down Mode)

The CNT is counted up from the CNT's initial value to the CMP_MAX register value. When the CNT matches the CMP_MAX register value, the CNT is counted down to the CMP_MIN register value. When the CNT matches the CMP_MIN register value, the CNT is counted up to the CMP_MAX register value. When the PWMCNTS.PWMCSn bit is cleared to to stop counting, the CNT when the counting is restarted counts up regardless of the direction of counting before the counter is stopped.

The operations of the PWM mode 1 in the up-down mode are shown below. Each level (VH0, VH1, VL0, or VL1) is selected from no change, low state, high state, or toggle operation.

- When the CNT (upward counting) matches the CMP_A register value, the PWMnL output is changed to the level determened by PWMnLCR1.VL0 bits, and the DCNTC counts up from zero.
- When the DCNTC matches the CMP_C register value, the PWMnH output is changed to the level determened by PWMnHCR1.VH0 bits. Then, the DCNTC is cleared, and counting is stopped.
- When the CNT (downward counting) matches the CMP_A, the PWMnH output is changed to the level determened by PWMnHCR1.VH1 bits, and the DCNTD counts up from zero.
- When the DCNTD matches the CMP_D register value, the PWMnL output is changed to the level determened by PWMnLCR1.VL1 bits. Then, the DCNTD is cleared, and counting is stopped.

In the direct mode, the period from the setting of the compare match register to the compare mach generation must be needed for a constant period (3 CPU clock cycles + 40 PWM counting cycles or more). In the buffer mode, the value of the buffer register is transferred to the compare match register at 2 timings. A user can select both or either of them.

When the CNT matches the CMP_A register value during counting-up of the DCNTC/D, the DCNTC/D counts up again from zero. As a result, dead time becomes longer than the set value. When CMP_C = 0 or CMP_D = 0, dead-time is one PWM counter cycle.

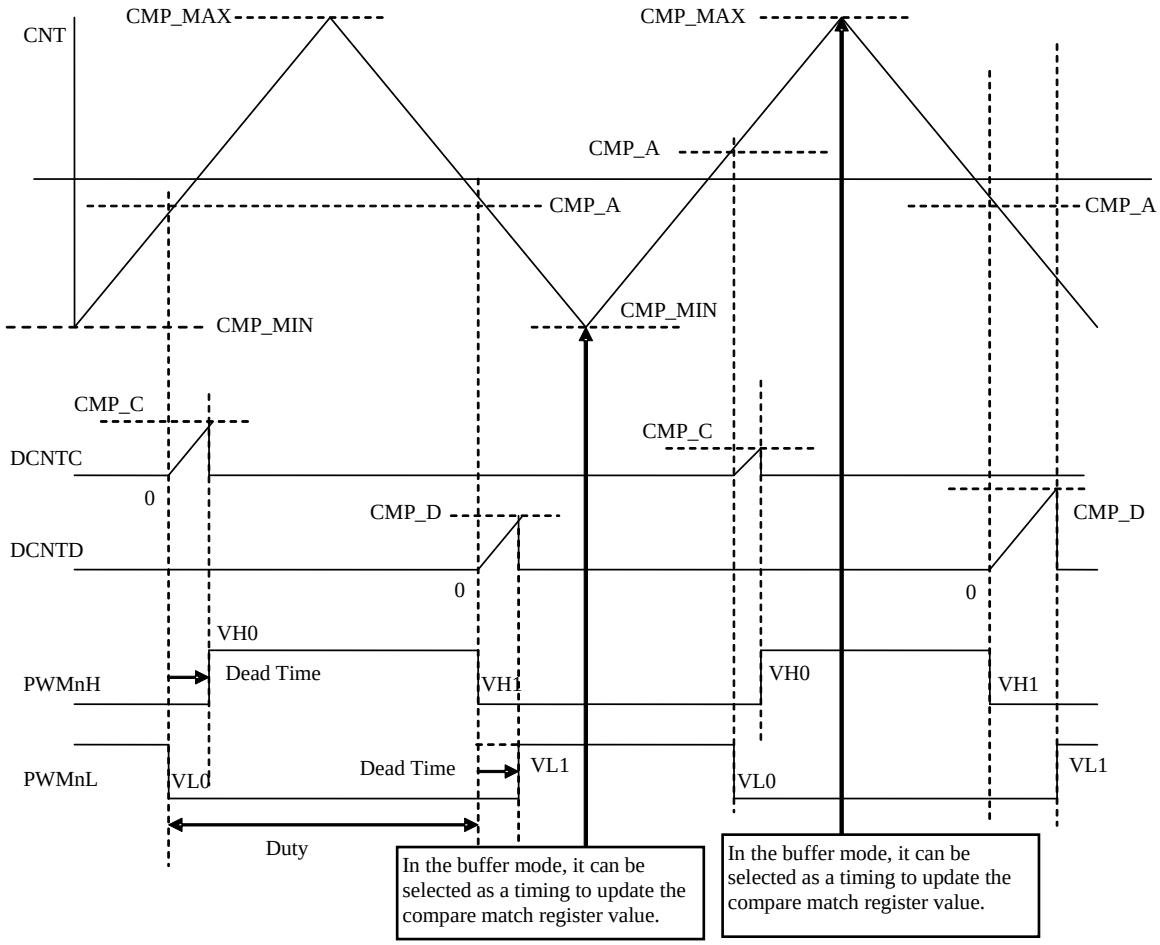


Figure 15-5. PWM Mode 1 (Up-down Mode)

15.5. Conflict or Output Control Condition

Table 15-4 shows the priority when the output control operation and more than one case of compare match is generated simultaneously. When such conflict occurs in each automatic dead-time mode, the DCNTC and DCNTD start according to the defined condition.

Table 15-4. Priority of Output Control Condition

Mode	Priority of PWMnH Output Control		Priority of PWMnL Output Control	
PWM Mode 0 (Up Mode)	High ↑ Low	VTH event	High ↑ Low	VTL event
		VH1 event		VL1 event
		VH0 event		VL0 event
		Compare mach event of CMP_MAX		Compare mach event of CMP_MAX
PWM Mode 0 (Up-down Mode)	High ↑ Low	VTH event	High ↑ Low	VTL event
		VH1 event		VL1 event
		VH0 event		VL0 event
		Compare mach event of CMP_MAX		Compare mach event of CMP_MAX
		Compare mach event of CMP_MIN		Compare mach event of CMP_MIN
PWM Mode 1 (Up Mode)	High ↑ Low	VTH event	High ↑ Low	VTL event
		VH1 event		VL1 event
		VH0 event		VL0 event
		Compare mach event of CMP_MAX		Compare mach event of CMP_MAX
PWM Mode 1 (Up-down Mode)	High ↑ Low	VTH event	High ↑ Low	VTL event
		VH1 event		VL1 event
		VH0 event		VL0 event
		Compare mach event of CMP_MAX		Compare mach event of CMP_MAX
		Compare mach event of CMP_MIN		Compare mach event of CMP_MIN

When the update of the compare match register or the CNT from the BUS conflicts with a compare match, the PWM operates as follows:

- When the compare match register is updated to another value by compare match generation, the PWM executes the operation by the compare match with the value before the update.
- When the compare match is generated with a post-update value by updating the compare match register, the PWM does not execute the operation by the compare match with the value after the update.
- When the CNT is rewritten to another value forcibly by compare match generation, the PWM executes the operation by the compare match with the value before the rewriting.
- When the CNT is rewritten to another value forcibly and the rewritten value matches the compare match register, the PWM does not execute the operation by the compare match with the value after the rewriting.

15.6. Operation Timing

15.6.1. Compare Match Timing

Figure 15-6 shows the timing of compare match. The condition of the example shown in Figure 15-6 is as follows:

- Module system clock: 960 MHz
- Count-up/count-down timing: 960 MHz

The CNT is updated at the count-up/count-down timing of 960 MHz. After the compare match register, CMP_xx, matches the CNT, a compare match is generated at the timing of updating the CNT next time.

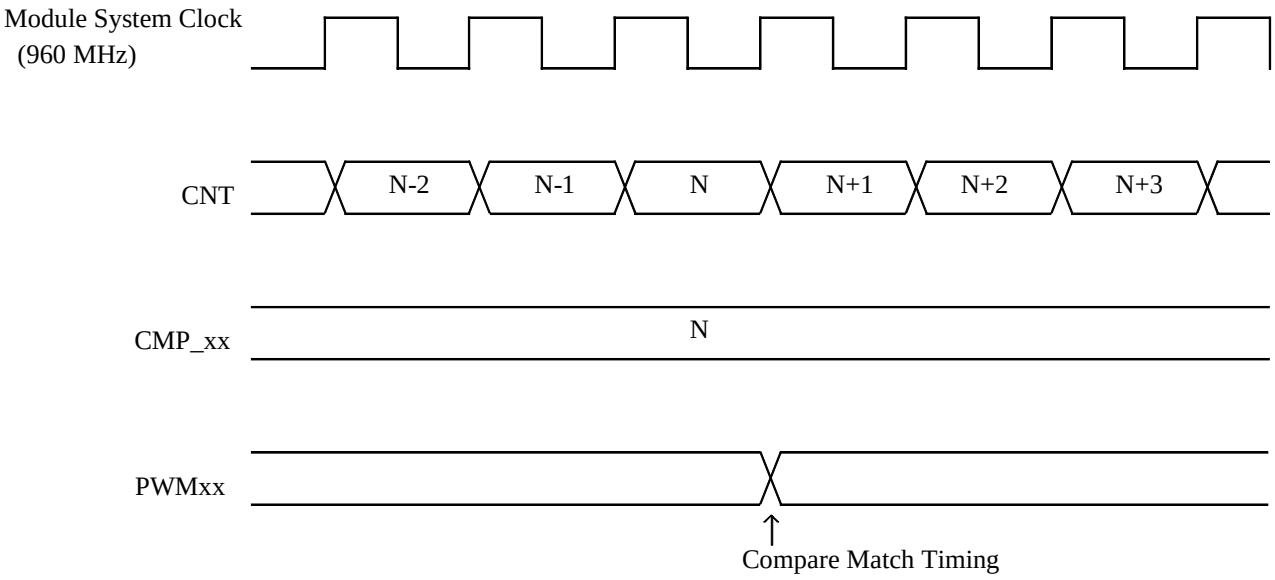


Figure 15-6. Compare Match Timing

15.6.2. CNT Clear Timing in Up Mode

Figure 15-7 shows the CNT clear timing in the up mode. The condition of the example shown in Figure 15-7 is as follows:

- Module system clock: 960 MHz
- Count-up timing: 960 MHz

After the value of the CMP_MAX register matches the CNT, the CNT reloads the value of the CMP_MIN register at the timing of updating the CNT next time.

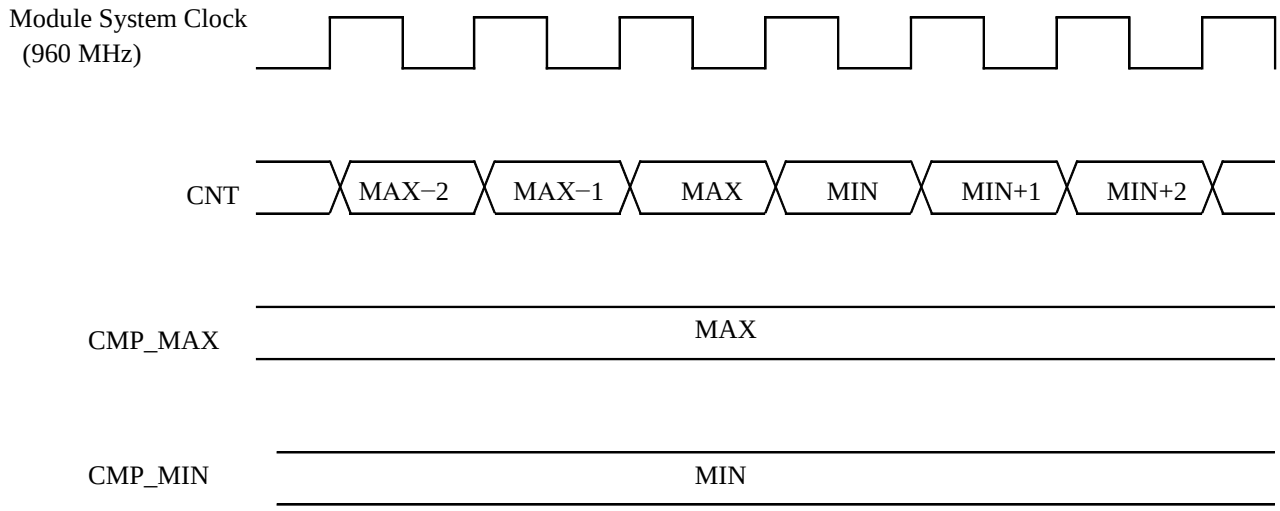


Figure 15-7. Counter Clear Timing (CNT's Reloading of CMP_MIN) in Up Mode

15.6.3. Up-down counter's Change from Count up to Count down in Up-down Mode

Figure 15-8 shows the timing to change from count up to count down. The condition of the example shown in Figure 15-8 is as follows:

- Module system clock: 960 MHz
- Count-up/count-down timing: 960 MHz

The CNT's first value in the count-down operation is the same as the CNT's last value in the count-up operation. This means that the value of the CMP_MAX register is kept for the 2 cycles between the CNT's count up and count down.

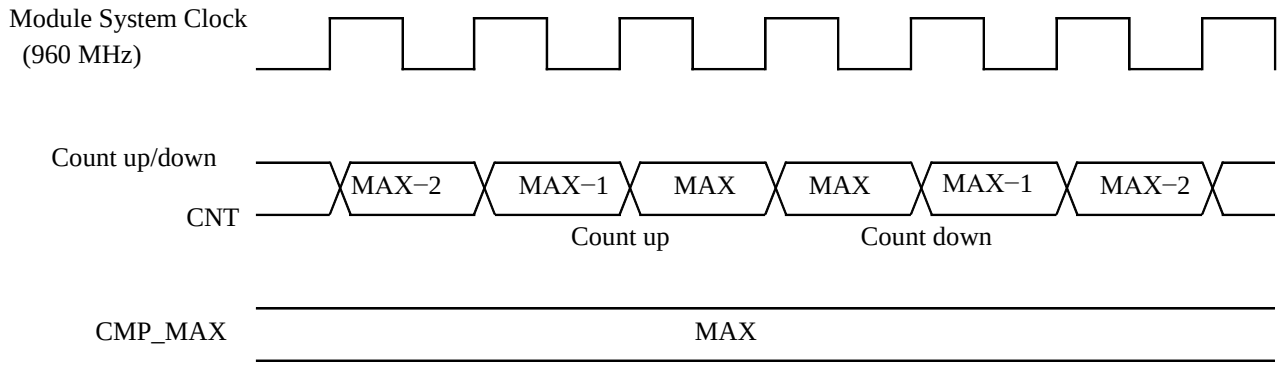


Figure 15-8. Counter Change Timing from Count Up to Count Down in Up-down Mode

15.6.4. Up-down counter's Change from Count down to Count up in the Up -down Mode

Figure 15-9 shows the timing to change from count-down to count-up. The condition of the example shown in Figure 15-9 is as follows:

- Module system clock: 960 MHz
- Count-up/count-down timing: 960 MHz

When the CNT is switched from count down to count up, its first value is the same as the last value in the count-down operation. This means that the value of the CMP_MIN register is kept for 2 cycles between the CNT's count down and count up.

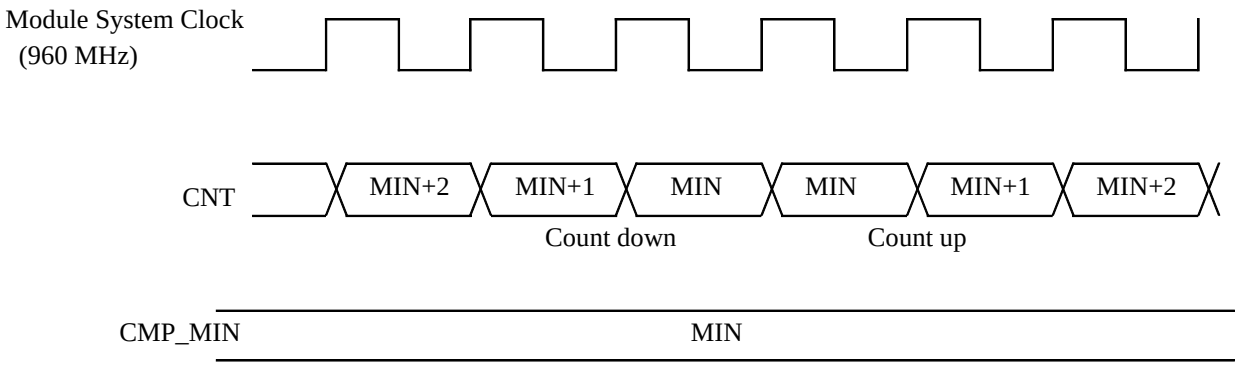


Figure 15-9. Counter Change Timing from Count Down to Count Up in Up-down Mode

The judgment of a compare match is executed every time even immediately after reloading of the CNT.

The minimum pulse width is the counter updating cycle (1.096 ns at 960 MHz). However, when the pulse width is very small, the pulse might be lost due to the rising/falling time of the pin.

There is an internal delay from the generation of a compare match event to the generation of the toggle operation of PWMnH and PWMnL.

15.7. Re-trigger Operation

The re-trigger operation mode can be set on each counter block. Each output signal or counter value can be set again by the specified event. The re-trigger operation has 5 modes (A, B, C, D, and re-trigger masking operation), and both the PWMnH and PWMnL output levels can be controlled in these 5 operation modes.

15.7.1. Re-trigger Events

The re-trigger operation can be specified by the edge event and level event. Table 15-5 shows the overview of the edge event, and Table 15-6 shows that of the level event. To set a re-trigger event of the PWM, write the number corresponding to the event to the PWMnRTRS.PWMRTS bit.

A setting example is as follows:

- When using event0 output from EPU's channel0 as re-trigger event: PWMnRTRS.PWMRTS = 0b010100
- When using the output from LUT0 as re-trigger event: PWMnRTRS.PWMRTS = 0b100110

The edge event of re-trigger is generated when the PWM receives an event pulse from a module or detects the edge of a specific signal. When the re-trigger is set as an edge event, the PWM manages only presence or absence of an event generation. To use the edge event of re-trigger, set the PWMnRTRG.RTRGPLS bits to 0b000.

For the level event of re-trigger, the PWM manages the signal level. The timing of event generation can be selected from one of following signals: rising edge, falling edge, both edges, high level, or low level. For the setting of the generation timing of level events, see the description on the PWMnRTRG register (Section 15.12.15). When the generation timing of the level event is set to low level or high level, the re-trigger operation mode C cannot be selected.

The PWM performs different processes in re-trigger masking operation between the edge event and level event. For details, see Section 15.7.8.

Table 15-5. Edge Events for Re-trigger

No.	Source	Remarks
0	CPU access	
1	Reserved	
2	Trigger pulse from Timer0_CMA	
3	Trigger pulse from Timer1_CMA	
4	Trigger pulse from CMP0	
5	Trigger pulse from CMP1	
6	Trigger pulse from CMP2	
7	Trigger pulse from CMP3	
8	Positive edge signal event form GPIO0	
9	Positive edge signal event form GPIO1	
10	Positive edge signal event form GPIO2	
11	Reserved	
12	Negative edge signal event form GPIO0	
13	Negative edge signal event form GPIO1	
14	Negative edge signal event form GPIO2	
15	Reserved	
16	Trigger pulse from Timer2_CMA	
17	Trigger pulse from Timer3_CMA	
18	Trigger pulse from CMP4	
19	Trigger pulse from CMP5	
20	Channel0 event0 of EPU	
21	Channel0 event1 of EPU	
22	Channel1 event0 of EPU	
23	Channel1 event1 of EPU	
24	Channel2 event0 of EPU	
25	Channel2 event1 of EPU	
26	Channel3 event0 of EPU	
27	Channel3 event1 of EPU	
28	Channel4 event0 of EPU	
29	Channel4 event1 of EPU	
30	Channel5 event0 of EPU	
31	Channel5 event1 of EPU	

Table 15-6. Level Events for Re-trigger

No.	Source	Remarks
32	CMP0 output	
33	CMP1 output	
34	CMP2 output	
35	CMP3 output	
36	CMP4 output	
37	CMP5 output	
38	LUT0 output	
39	LUT1 output	
40	GPIO0 event level	
41	GPIO1 event level	
42	GPIO2 event level	
43 to 63	Reserved	

15.7.2. Operation in Re-trigger Mode A

In re-trigger mode A, when the specified event is detected, each PWM output signal is changed to the specified level (for details of the changing method, see Section 15.7.7). This state is kept until the counter (CNT) is stopped. During this period, the PWM stops outputting events and changing of the PWM output by comparing the CNT with the compare match register. This period is called a non-comparison period.

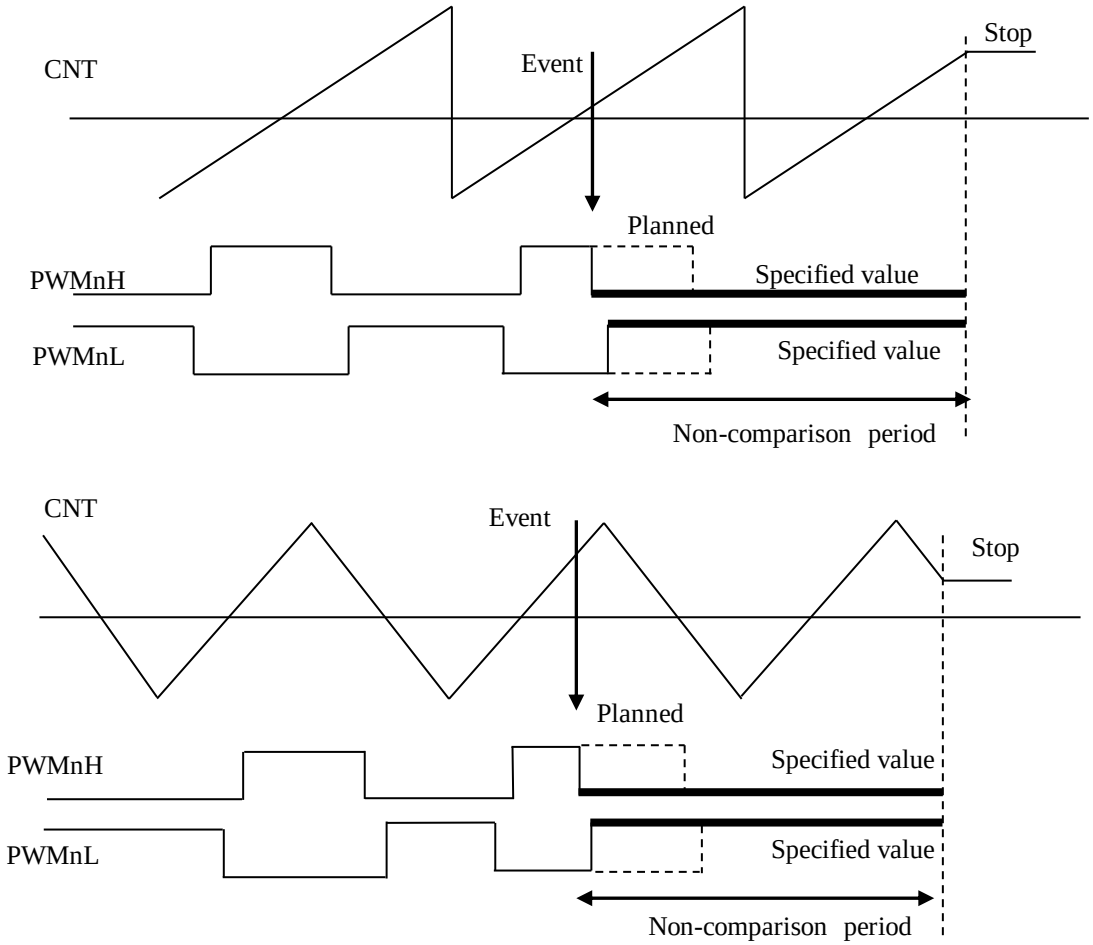


Figure 15-10. Re-trigger Mode A

15.7.3. Operation in Re-trigger Mode B

In re-trigger mode B, when the specified event is detected, each PWM output signal is changed to the specified level (for details of the changing method, see Section 15.7.7). This state is kept until the counter (CNT) loads the value of CMP_MIN register. The period from detection of an event to loading of the CMP_MIN value register is a non-comparison period.

When the CNT value becomes equal to the CMP_MIN value, the PWM starts the normal comparison operation again.

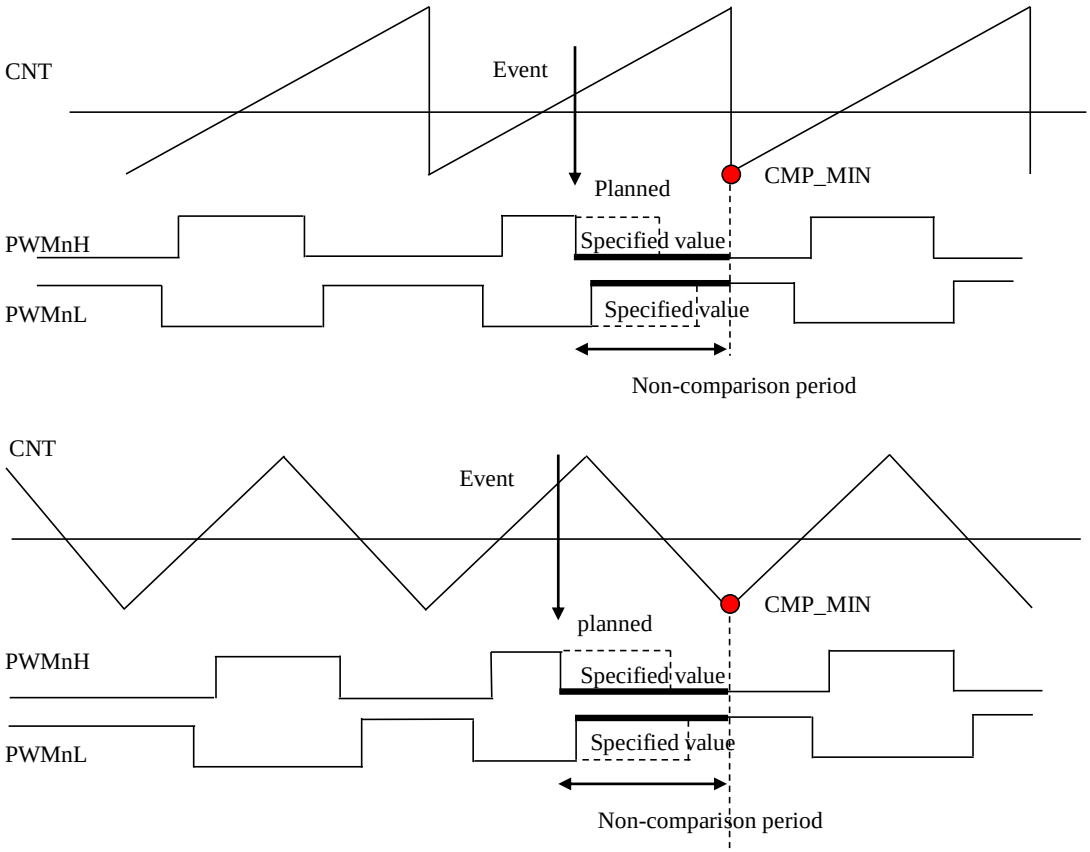


Figure 15-11. Re-trigger Mode B

15.7.4. Operation in Re-trigger Mode C

In re-trigger mode C, when the specified event is detected, each PWM output signal is changed to the specified level. Then, the counter (CNT) operates as follows:

- In the up mode: The CNT loads the value of the CMP_MIN register, and counts up from the loaded value.
- During count-up in the up-down mode: The CNT loads the value of the CMP_MAX register, and counts down from the loaded value.
- During count-down in the up-down mode: The CNT loads the value of the CMP_MIN register, and counts up from the loaded value.

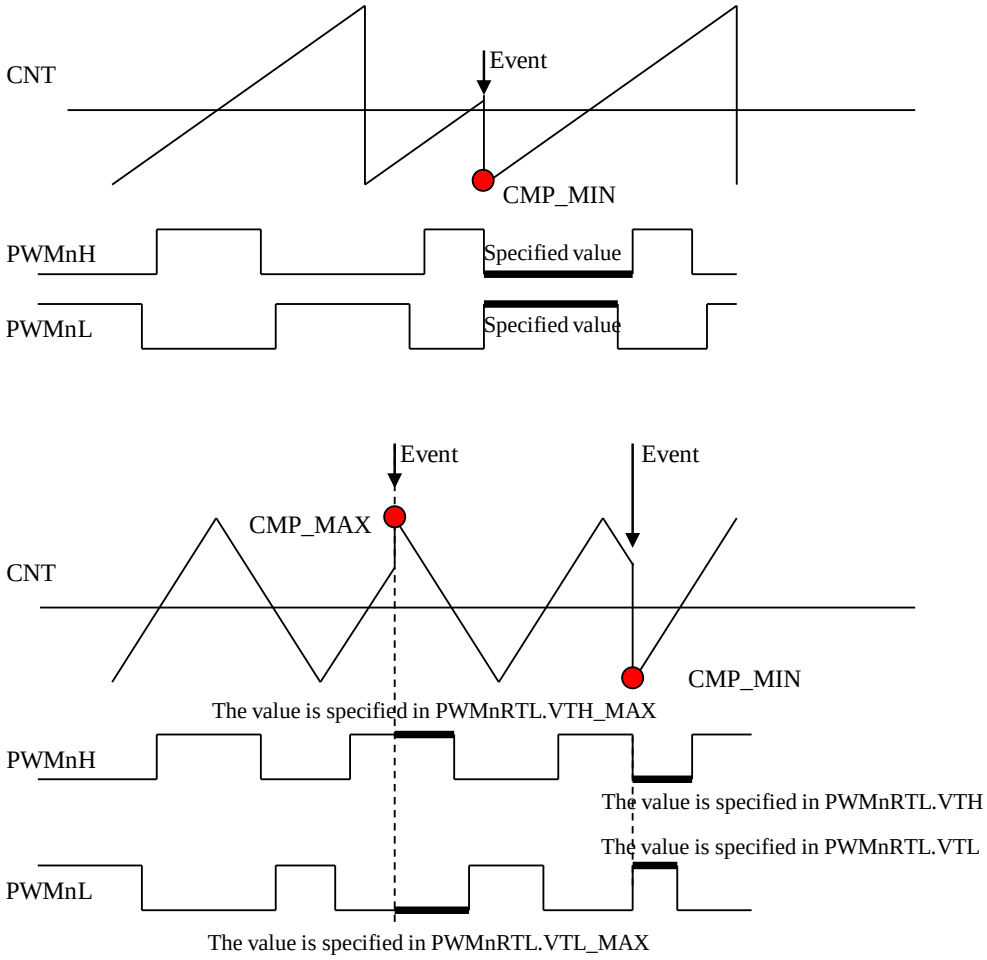


Figure 15-12. Re-trigger Mode C

15.7.5. Buffer Updating when Operation of Re-trigger Mode C Starts

In re-trigger mode C, the compare match register can be updated with the buffer register value using a re-trigger operation starting event as the trigger. The setting method is different depending on the counting direction of the PWM counter. For details, see bits 6 and 7 of the PWMnRTRS register.

15.7.6. Operation in Re-trigger Mode D

In re-trigger mode D, when the specified event is detected, the PWM enters a non-comparison period from the timing that the counter (CNT) loads the value of the CMP_MIN register after the event detection to the timing that the CNT loads the CMP_MIN value next time.

When the specified event is detected again in the non-comparison period, this period is extended to the next PWM cycle.

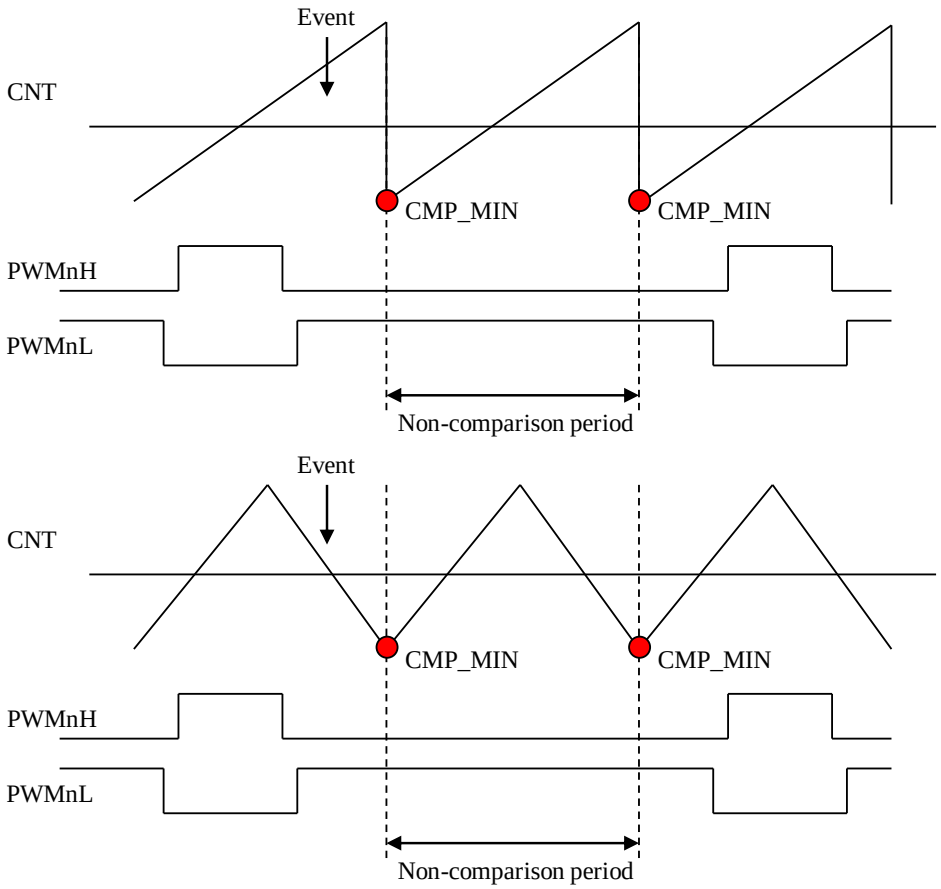


Figure 15-13. Re-trigger Mode D

15.7.7. Re-trigger Masking Operation

By setting to the re-trigger mask, re-trigger events can be masked (ignored). The start timing of the masking period can be selected from each edge of the PWM output signal (rising edge or falling edge of the PWMnH/PWMnL).

The division of the counter to measure a masking period can be selected from Count Clock Frequency/8, Count Clock Frequency/16, or Count Clock Frequency/32.

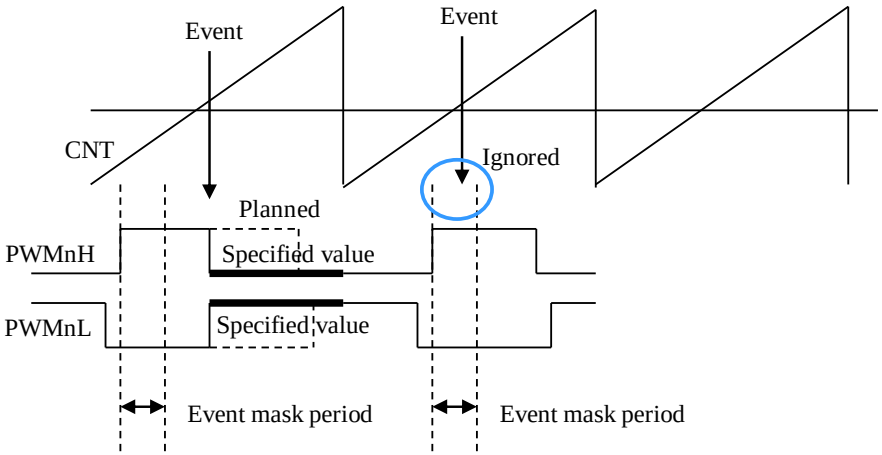


Figure 15-14. Re-trigger Masking Operation

15.7.8. Interpretation of Edge and Level Events at Masking Release

The PWM operation in the re-trigger masking is different according to the re-trigger event setting: edge event or level event.

When the edge event is selected, the PWM receives the event as a pulse. Thus, the edge events are ignored in the masking period.

When a level event is selected, the PWM manages the level of the event signal. When using event masking for level events, the level of the event signal is fixed to the negating side during the event operation. For example, if the falling edge of the level signal is set as the level event of re-trigger, and the level signal is low when the masking period ends: The level signal is fixed to high by the masking process during the masking period. When this period ends, the level signal becomes low. Then, the falling edge is generated. As a result, an event of re-trigger operation is generated, and the PWM performs the re-trigger operation.

Figure 15-15 shows the overview of the re-trigger detection circuit.

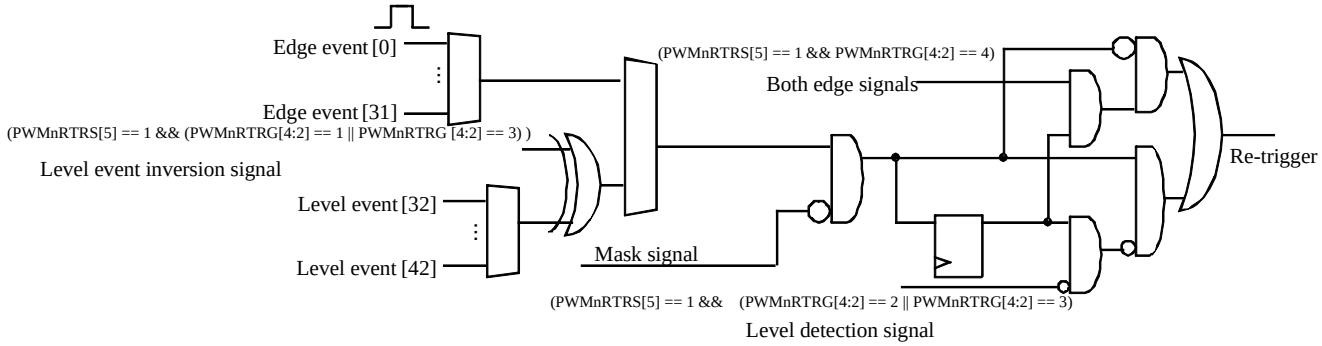


Figure 15-15. Re-trigger Detection Circuit

15.7.9. Method to Change Output Waveform by Re-trigger Operation

When the PWM executes the re-trigger operation, the output waveform is changed as shown in Table 15-7. The content of Table 15-7 is applied to both the up and up-down modes.

The output levels of the PWMnH and PWMnL are determined by the PWMnRTL.VTH and PWMnRTL.VTL bits, respectively.

Table 15-7. Output Waveform by Re-trigger Operation

Mode	Output Level of PWMnH	Output Level of PWMnL
PWM Mode 0	Changes to the specified value by the PWMnRTL.VTH bits immediately.	Changes to the specified value by the PWMnRTL.VTL bits immediately.
PWM Mode 1	Changes to the specified value by the PWMnRTL.VTH bits immediately, and the DCNTD counts up to the value of the CMP_D register simultaneously.	Changes to the specified value by the PWMnRTL.VTL bits at the timing when the compare mach of the DCNTD and DCNTD is generated, and clears the DCNTD simultaneously.

15.8. Event Output

Each counter block has the following 3 types of event output.

- PWMn Event0
- PWMn Event1
- PWMn TIMERSYNC (This event resets the counter value of the timer: see Section 17.)

The generation source of each event can be selected from the following PWM internal events. More than one PWM internal event can be selected per event. When more than one PWM internal event is selected, the PWM event is output when one or more PWM internal event is generated.

- EVT_MIN: Compare match event of CNT and CMP_MIN register value
- EVT_MAX: Compare match event of CNT and CMP_MAX register value
- EVT_VH1: Event to transit the PWMnH pin to pin state VH1.
- EVT_VH0: Event to transit the PWMnH pin to pin state VH0.
- EVT_VL1: Event to transit the PWMnL pin to pin state VL1.
- EVT_VL0: Event to transit the PWMnL pin to pin state VL0.
- EVT_T: Detection of the re-trigger operation starting event

In the up mode, the event of the EVT_MIN is not generated.

The event of the EVT_VH0/1 or the EVT_VL0/1 is generated only when the pin state transits to the VH0/1 or the VL0/1. For example, when the VL0 is set to low level and the PWMnL becomes low level by accessing the PWMnLCR0 register, the event of the EVT_VL0 is not generated. This is because the pin level matches the level specified by the VL0 by chance, so that the PWMnL pin does not transfer to the control state of the VL0. The events of the EVT_VH1, EVT_VH0, EVT_VL1, and EVT_VL0 are generated only when compare match is generated between the counter and the compare match register.

15.9. Interrupt Output

Each counter block has the following 2 types of interrupt output.

- PWMn INT0
- PWMn INT1

The generation source of each interrupt can be selected from the following PWM internal events. More than one PWM internal event can be selected per interrupt. When more than one PWM internal event is selected, the interrupt is generated when one or more PWM internal event is generated.

- INT_MIN: Compare match event of CNT and CMP_MIN register value
- INT_MAX: Compare match event of CNT and CMP_MAX register value
- INT_VH1: Event to transit the PWMnH pin to pin state VH1
- INT_VH0: Event to transit the PWMnH pin to pin state VH0
- INT_VL1: Event to transit the PWMnL pin to pin state VL1
- INT_VL0: Event to transit the PWMnL pin to pin state VL0
- INT_T: Detection of the re-trigger operation starting event

In the count-up mode, the event of the INT_MIN is not generated.

The event of the INT_VH0/1 or the INT_VL0/1 is generated only when the pin state transits to the VH0/1 or the VL0/1. For example, when the VL0 is set to low level and the PWMnL becomes low level by accessing the PWMnLCR0 register, the event of the INT_VL0 is not generated. This is because the pin level matches the level specified by the VL0 by chance, so that the PWMnL pin does not transfer to the control state of the VL0. The events of the INT_VH1, INT_VH0, INT_VL1, and INT_VL0 are generated only when compare match is generated between the counter and the compare match register.

15.10. Event and Interrupt Output in Re-trigger Operation

When the re-trigger A, B, or D is generated, a period in which compare match is not performed until the re-trigger operation is released occurs. Even in the default setting, any events or interrupts by compare match are not output.

For example, if the PWM is set so that the PWMn Event0 is output if the PWMnH becomes the VH0 in the up mode of PWM mode 0 (PWMnEVO0.EVT_VH0 = 1) when the CNT matches the value of the CMP_C register, the PWMnH changes to the VH0, and the PWMn Event0 is generated. When the PWM starts re-triggering before the CNT matches the CMP_C value, the PWMn Event0 is not generated. This is because the PWMnH pin does not transit to the control state even when the CNT matches the CMP_C value.

To output the event/interrupt generating at the compare mach timing of the CNT and the compare match register value during the re-trigger operation, set PWMnRTRG.RTMSKD = 1. By this setting, even during the re-trigger operation as shown in the example above, the PWMn Event0 is output when the CNT matches the CMP_C value.

15.11. Register Access

The PWM module has 16-bit registers such as the buffer register, compare match register, and counter (CNT). Write the register on the LSB side first when these registers are written from the CPU or the DSAC.

In a write access, the write data is stored in the temporary register on the LSB side by the first access. Then, the data is written to the MSB side by the second access. The data in the temporary register is transferred to the LSB side of the register simultaneously with the second access.

In a read access, the data on the LSB side is obtained by the first access. The data on the MSB side is stored in the temporary register simultaneously. After that, the data on the MSB side is obtained from the temporary register at the second time accessed.

For PWM's 16-bit width register allocated in the SFR area, the LSB side and MSB side registers of 16-bit are assigned to the same address.

For PWM's 16-bit width register allocated in the XDATA BUS area, the LSB side and MSB side registers of 16-bit are assigned to an independent address, respectively. These addresses are allocated by the little endian method (the LSB side is assigned to the lower address and the MSB side is to the higher address).

For the buffer register mapped in the SFR area, the DSAC can read/write 16-bit data at one time by the one-word access mode.

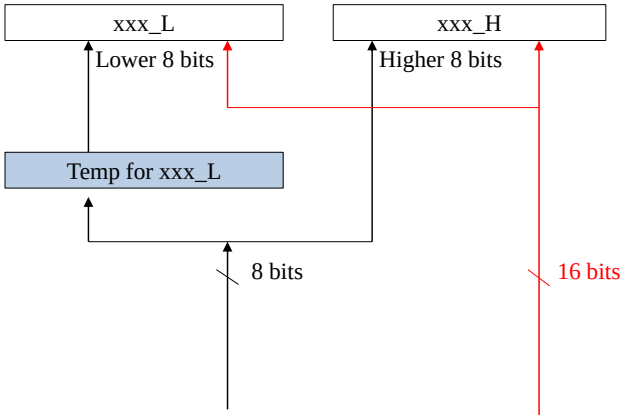


Figure 15-16. 16-bit Register Writing Method

The PWMnACSTS.XREGACS bit is the writing access flag for each block of the register connected with the XDATA BUS. This bit is asserted when writing to either register shown below:

- 8-bit register connected with XDATA BUS (other than PWMCNTS, PWMnEVO0/1/T, PWMnINTS0/1, or PWMnACCLR), or
- The data on the MSB side of the 16-bit width register connected with XDATA BUS.

PWMnACSTS.XREGACS = 1 indicates that writing to the above 8-bit registers connected with XDATA BUS is ongoing. New writing is not received while this writing is ongoing. When the writing operation finishes, the PWMnACSTS.XREGACS bit is negated so that the next writing can be received. Writing is received only when PWMnACSTS.XREGACS = 0.

The PWMnACSTS.SFRACS bit is the writing access flag of the SFR. PWMnACSTS.SFRACS = 1 indicates that the writing operation of the SFR is ongoing. When data is written in the MSB side of the BUF_A, B, C, or D register, the PWMnACSTS.SFRACS bit is set to 1. The BUF_A, B, C, or D register can be written continuously only once during PWMnACSTS.SFRACS = 1. For example, data can be written in order such as BUF_An_L -> BUF_An_H -> BUF_Bn_L -> BUF_Bn_H -> BUF_Cn_L -> BUF_Cn_H -> BUF_Dn_L -> BUF_Dn_H. Before writing to the BUF_A, B, C, or D register again, make sure that PWMnACSTS.SFRACS = 0.

The PWMnACSTS.CNTSACS bit is a flag to manage accessing to the PWMCNTS register. This bit is asserted when the PWMCNTS register is written Before writing to the PWMCNTS register again, make sure that PWMnACSTS.CNTSACS = 0.

When writing to the register whose access is managed by the PWMnACSTS register, 6 CPU clock cycles + 24 PWM count cycles are necessary.

Only writing is valid when accessing from the SFR BUS to the register assigned to x60 to 0x7F of the SFR BUS. When the data of the corresponding register is read from the SFR_BUS, all the read data becomes 0.

15.12. Register Descriptions

Table 15-8. XDATA BUS Common Registers

Symbol	Address	Initial Value
PWMCNTS	0xF903	0x00

Table 15-9. List of XDATA BUS Registers (Each Channel)

Symbol (Channel n)	Address (Channel0)	Address (Channel1)	Address (Channel2)	Address (Channel3)	Initial Value
PWMnEVO0	0xF905	0xF945	0xF985	0xF9C5	0x00
PWMnEVO1	0xF906	0xF946	0xF986	0xF9C6	0x00
PWMnEVOT	0xF907	0xF947	0xF987	0xF9C7	0x00
PWMnINTS0	0xF908	0xF948	0xF988	0xF9C8	0x00
PWMnINTS1	0xF909	0xF949	0xF989	0xF9C9	0x00
PWMnINTF	0xF90A	0xF94A	0xF98A	0xF9CA	0x00
PWMnACCLR	0xF90B	0xF94B	0xF98B	0xF9CB	0x00
PWMnACSTS	0xF90C	0xF94C	0xF98C	0xF9CC	0x00
CNTn_L	0xF910	0xF950	0xF990	0xF9D0	0x00
CNTn_H	0xF911	0xF951	0xF991	0xF9D1	0x00
CMP_An_L	0xF912	0xF952	0xF992	0xF9D2	0x00
CMP_An_H	0xF913	0xF953	0xF993	0xF9D3	0x00
CMP_Bn_L	0xF914	0xF954	0xF994	0xF9D4	0x00
CMP_Bn_H	0xF915	0xF955	0xF995	0xF9D5	0x00
CMP_Cn_L	0xF916	0xF956	0xF996	0xF9D6	0x00
CMP_Cn_H	0xF917	0xF957	0xF997	0xF9D7	0x00
CMP_Dn_L	0xF918	0xF958	0xF998	0xF9D8	0x00
CMP_Dn_H	0xF919	0xF959	0xF999	0xF9D9	0x00
CMP_MINn_L	0xF91A	0xF95A	0xF99A	0xF9DA	0x00
CMP_MINn_H	0xF91B	0xF95B	0xF99B	0xF9DB	0x00
CMP_MAXn_L	0xF91C	0xF95C	0xF99C	0xF9DC	0x00
CMP_MAXn_H	0xF91D	0xF95D	0xF99D	0xF9DD	0x00
PWMnCNTMD	0xF920	0xF960	0xF9A0	0xF9E0	0x00
PWMnHCR0	0xF921	0xF961	0xF9A1	0xF9E1	0x00
PWMnLCR0	0xF922	0xF962	0xF9A2	0xF9E2	0x00
PWMnHCR1	0xF923	0xF963	0xF9A3	0xF9E3	0x00
PWMnLCR1	0xF924	0xF964	0xF9A4	0xF9E4	0x00
PWMnMODE	0xF925	0xF965	0xF9A5	0xF9E5	0x00
PWMnRTRG	0xF926	0xF966	0xF9A6	0xF9E6	0x00
PWMnRTRS	0xF927	0xF967	0xF9A7	0xF9E7	0x00
PWMnRTGC	0xF928	0xF968	0xF9A8	0xF9E8	0x00
PWMnRTL	0xF929	0xF969	0xF9A9	0xF9E9	0x00
PWMnRTMC	0xF92A	0xF96A	0xF9AA	0xF9EA	0x00
PWMnRTMP	0xF92B	0xF96B	0xF9AB	0xF9EB	0x00
BUF_MINn_L	0xF92C	0xF96C	0xF9AC	0xF9EC	0x00
BUF_MINn_H	0xF92D	0xF96D	0xF9AD	0xF9ED	0x00
BUF_MAXn_L	0xF92E	0xF96E	0xF9AE	0xF9EE	0x00
BUF_MAXn_H	0xF92F	0xF96F	0xF9AF	0xF9EF	0x00

Table 15-10. List of SFR Register (Each Channel)

Symbol (Channel n)	Address (Channel0)	Address (Channel1)	Address (Channel2)	Address (Channel3)	Initial Value
BUF_An_L	0xE4	0xEC	0xF4	0xFC	0x00
BUF_An_H	0xE4	0xEC	0xF4	0xFC	0x00
BUF_Bn_L	0xE5	0xED	0xF5	0xFD	0x00
BUF_Bn_H	0xE5	0xED	0xF5	0xFD	0x00
BUF_Cn_L	0xE6	0xEE	0xF6	0xFE	0x00
BUF_Cn_H	0xE6	0xEE	0xF6	0xFE	0x00
BUF_Dn_L	0xE7	0xEF	0xF7	0xFF	0x00
BUF_Dn_H	0xE7	0xEF	0xF7	0xFF	0x00
CMP_An_L	0x64	0x6C	0x74	0x7C	0x00
CMP_An_H	0x64	0x6C	0x74	0x7C	0x00
CMP_Bn_L	0x65	0x6D	0x75	0x7D	0x00
CMP_Bn_H	0x65	0x6D	0x75	0x7D	0x00
CMP_Cn_L	0x66	0x6E	0x76	0x7E	0x00
CMP_Cn_H	0x66	0x6E	0x76	0x7E	0x00
CMP_Dn_L	0x67	0x6F	0x77	0x7F	0x00
CMP_Dn_H	0x67	0x6F	0x77	0x7F	0x00
BUF_MINn_L	0x60	0x68	0x70	0x78	0x00
BUF_MINn_H	0x60	0x68	0x70	0x78	0x00
BUF_MAXn_L	0x61	0x69	0x71	0x79	0x00
BUF_MAXn_H	0x61	0x69	0x71	0x79	0x00
CMP_MINn_L	0x62	0x6A	0x72	0x7A	0x00
CMP_MINn_H	0x62	0x6A	0x72	0x7A	0x00
CMP_MAXn_L	0x63	0x6B	0x73	0x7B	0x00
CMP_MAXn_H	0x63	0x6B	0x73	0x7B	0x00

15.12.1. PWMCNTS (PWM Counter Start)

The counter of each PWM channel can start simultaneously (in synchronization). When writing 1 to the PWMCSn bit, all the counters start counting.

Register	PWMCNTS	PWM Counter Start		Address	0xF903
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
4	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
3	PWMCS3	R/W	0	PWM3 counter start/stop 0: PWM3 counter stop 1: PWM3 counter start	
2	PWMCS2	R/W	0	PWM2 counter start/stop 0: PWM2 counter stop 1: PWM2 counter start	
1	PWMCS1	R/W	0	PWM1 counter start/stop 0: PWM1 counter stop 1: PWM1 counter start	
0	PWMCS0	R/W	0	PWM0 counter start/stop 0: PWM0 counter stop 1: PWM0 counter start	

15.12.2. PWMnEVO0/1/T (PWM Event0/1 Output/ to Timer for Block n) (n = 0 to 3)

Register	PWM0EVO0	PWM Event0 Output for Block0	Address	0xF905
Register	PWM1EVO0	PWM Event0 Output for Block1	Address	0xF945
Register	PWM2EVO0	PWM Event0 Output for Block2	Address	0xF985
Register	PWM3EVO0	PWM Event0 Output for Block3	Address	0xF9C5
Register	PWM0EVO1	PWM Event1 Output for Block0	Address	0xF906
Register	PWM1EVO1	PWM Event1 Output for Block1	Address	0xF946
Register	PWM2EVO1	PWM Event1 Output for Block2	Address	0xF986
Register	PWM3EVO1	PWM Event1 Output for Block3	Address	0xF9C6
Register	PWM0EVOT	PWM Event to Timer for Block0	Address	0xF907
Register	PWM1EVOT	PWM Event to Timer for Block1	Address	0xF947
Register	PWM2EVOT	PWM Event to Timer for Block2	Address	0xF987
Register	PWM3EVOT	PWM Event to Timer for Block3	Address	0xF9C7

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	EVT_T	R/W	0	Event output source To select a PWM internal event, set the bit corresponding to the event to 1. When more than one PWM internal event is selected, the PWM event is output when one or more PWM internal event is generated. The PWM event signal is one pulse.	
5	EVT_VH1	R/W	0		
4	EVT_VH0	R/W	0		
3	EVT_VL1	R/W	0		
2	EVT_VL0	R/W	0		
1	EVT_MAX	R/W	0		
0	EVT_MIN	R/W	0		

15.12.3. PWMnINTS0/1 (PWM Interrupt0/1 Select for Block n) (n = 0 to 3)

Register	PWM0INTS0	PWM Interrupt0 Select for Block0	Address 0xF908
Register	PWM1INTS0	PWM Interrupt0 Select for Block1	Address 0xF948
Register	PWM2INTS0	PWM Interrupt0 Select for Block2	Address 0xF988
Register	PWM3INTS0	PWM Interrupt0 Select for Block3	Address 0xF9C8
Register	PWM0INTS1	PWM Interrupt1 Select for Block0	Address 0xF909
Register	PWM1INTS1	PWM Interrupt1 Select for Block1	Address 0xF949
Register	PWM2INTS1	PWM Interrupt1 Select for Block2	Address 0xF989
Register	PWM3INTS1	PWM Interrupt1 Select for Block3	Address 0xF9C9

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	INT_T	R/W	0	Interrupt source To select a PWM internal event, set the bit corresponding to the event to 1. When more than one PWM internal event is selected, the interrupt flag is set when one or more PWM internal event is generated.	
5	INT_VH1	R/W	0		
4	INT_VH0	R/W	0		
3	INT_VL1	R/W	0		
2	INT_VL0	R/W	0		
1	INT_MAX	R/W	0		
0	INT_MIN	R/W	0		

15.12.4. PWMnINTF (PWM Interrupt Flag for Block n) (n = 0 to 3)

Register	PWM0INTF	PWM Interrupt Flag for Block0	Address	0xF90A
Register	PWM1INTF	PWM Interrupt Flag for Block1	Address	0xF94A
Register	PWM2INTF	PWM Interrupt Flag for Block2	Address	0xF98A
Register	PWM3INTF	PWM Interrupt Flag for Block3	Address	0xF9CA

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	PWMIE1	R/W	0	PWM interrupt1 enable 0: PWM interrupt1 is disabled 1: PWM interrupt1 is enabled When PWMIE1 = 1 and PEMIF1 = 1, an interrupt request is generated.	
4	PWMIE0	R/W	0	PWM interrupt0 enable 0: PWM interrupt0 is disabled 1: PWM interrupt0 is enabled When PWMIE0 = 1 and PEMIF0 = 1, an interrupt request is generated.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	PWMIF1	R/C	0	PWM interrupt flag1 Read 0: No change Read 1: Interrupt event defined by the PWMnINTS1 register is detected Write 0: No change Write 1: The bit is cleared	
0	PWMIF0	R/C	0	PWM interrupt flag0 Read 0: No change Read 1: Interrupt event defined by the PWMnINTS0 register is detected Write 0: No change Write 1: The bit is cleared	

15.12.5. PWMnACCLR (PWM Access Counter Clear Register for Block n) (n = 0 to 3)

Register	PWM0ACCLR	PWM Access Counter Clear Register for Block0	Address	0xF90B
Register	PWM1ACCLR	PWM Access Counter Clear Register for Block1	Address	0xF94B
Register	PWM2ACCLR	PWM Access Counter Clear Register for Block2	Address	0xF98B
Register	PWM3ACCLR	PWM Access Counter Clear Register for Block3	Address	0xF9CB

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CLRCPUACC	W	0	CPU's SFR access counter clearing Write 0: No change Write 1: CPU access counter register is cleared The read value is always 0.	
6	CLRDSAACC	W	0	DSAC's SFR access counter clearing Write 0: No change Write 1: DSAC access counter register is cleared The read value is always 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

15.12.6. PWMnACSTS (PWM Access Status Register for Block n) (n = 0 to 3)

Register	PWM0ACSTS	PWM Access Status Register for Block0	Address	0xF90C
Register	PWM1ACSTS	PWM Access Status Register for Block1	Address	0xF94C
Register	PWM2ACSTS	PWM Access Status Register for Block2	Address	0xF98C
Register	PWM3ACSTS	PWM Access Status Register for Block3	Address	0xF9CC

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	CNTSACS	R	0	Access status to the PWMCNTS register 0: No access to the PWMCNTS register 1: Writing to the PWMCNTS register Before writing to the PWMCNTS register, make sure that CNTSACS = 0.	
1	SFRACS	R	0	Access status to the SFR BUS 0: No access to the SFR BUS register 1: Writing to the SFR BUS register When the same SFR BUS register is written continuously make sure that SFRACS = 0 before the next writing.	
0	XREGACS	R	0	Access status to the XDATA BUS register 0: No access to the XDATA BUS register 1: Writing to the XDATA BUS register Before writing to the XDATA BUS register, make sure that XREGACS = 0.	

15.12.7. CNTn_L/H (Counter n LSB/MSB Side) (n = 0 to 3)

When PWMnACSTS.XREGACS = 1, do not write to this register.

Register	CNT0_L	Counter0 LSB Side	Address	0xF910
Register	CNT1_L	Counter1 LSB Side	Address	0xF950
Register	CNT2_L	Counter2 LSB Side	Address	0xF990
Register	CNT3_L	Counter3 LSB Side	Address	0xF9D0

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CNT	R/W	0	LSB side counter	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

Register	CNT0_H	Counter0 MSB Side	Address	0xF911
Register	CNT1_H	Counter1 MSB Side	Address	0xF951
Register	CNT2_H	Counter2 MSB Side	Address	0xF991
Register	CNT3_H	Counter3 MSB Side	Address	0xF9D1

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CNT	R/W	0	MSB side counter	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

15.12.8. CMP_xxxn_L/H (CMP_xxx for Block n LSB/MSB Side) (n = 0 to 3)

When PWMnACSTS.XREGACS = 1, do not write to this register from the XDATA BUS.

Register	CMP_MIN0_L	CMP_MIN for Block0 LSB Side	Address	0xF91A
Register	CMP_MIN1_L	CMP_MIN for Block1 LSB Side	Address	0xF95A
Register	CMP_MIN2_L	CMP_MIN for Block2 LSB Side	Address	0xF99A
Register	CMP_MIN3_L	CMP_MIN for Block3 LSB Side	Address	0xF9DA

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMP_MIN	R/W	0	CMP_MIN[7:3]	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R	0	CMP_MIN[2:0] The read value is 0. The write value must always be 0.	
1		R	0		
0		R	0		

Register	CMP_MIN0_L	CMP_MIN for Block0 LSB Side	Address	0x62
Register	CMP_MIN1_L	CMP_MIN for Block1 LSB Side	Address	0x6A
Register	CMP_MIN2_L	CMP_MIN for Block2 LSB Side	Address	0x72
Register	CMP_MIN3_L	CMP_MIN for Block3 LSB Side	Address	0x7A

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMP_MIN	W	0	CMP_MIN[7:3] The read value is always 0.	
6		W	0		
5		W	0		
4		W	0		
3		W	0		
2		W	0	CMP_MIN[2:0] The read value is 0. The write value must always be 0.	
1		W	0		
0		W	0		

When PWMnACSTS.XREGACS = 1, do not write to this register from the XDATA BUS.

Register	CMP_MIN0_H	CMP_MIN for Block0 MSB Side	Address	0xF91B
Register	CMP_MIN1_H	CMP_MIN for Block1 MSB Side	Address	0xF95B
Register	CMP_MIN2_H	CMP_MIN for Block2 MSB Side	Address	0xF99B
Register	CMP_MIN3_H	CMP_MIN for Block3 MSB Side	Address	0xF9DB

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMP_MIN	R/W	0	MSB side of CMP_MIN	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

Register	CMP_MIN0_H	CMP_MIN for Block0 MSB Side	Address	0x62
Register	CMP_MIN1_H	CMP_MIN for Block1 MSB Side	Address	0x6A
Register	CMP_MIN2_H	CMP_MIN for Block2 MSB Side	Address	0x72
Register	CMP_MIN3_H	CMP_MIN for Block3 MSB Side	Address	0x7A

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMP_MIN	W	0	MSB side of CMP_MIN The read value is always 0.	
6		W	0		
5		W	0		
4		W	0		
3		W	0		
2		W	0		
1		W	0		
0		W	0		

When PWMnACSTS.XREGACS = 1, do not write to this register from the XDATA BUS.

Register	CMP_MAX0_L	CMP_MAX for Block0 LSB Side	Address	0xF91C
Register	CMP_MAX1_L	CMP_MAX for Block1 LSB Side	Address	0xF95C
Register	CMP_MAX2_L	CMP_MAX for Block2 LSB Side	Address	0xF99C
Register	CMP_MAX3_L	CMP_MAX for Block3 LSB Side	Address	0xF9DC

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMP_MAX	R/W	0	CMP_MAX[7:3]	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R	1	CMP_MAX[2:0] The read value is 7. The write value must always be 7.	
1		R	1		
0		R	1		

Register	CMP_MAX0_L	CMP_MAX for Block0 LSB Side	Address	0x63
Register	CMP_MAX1_L	CMP_MAX for Block1 LSB Side	Address	0x6B
Register	CMP_MAX2_L	CMP_MAX for Block2 LSB Side	Address	0x73
Register	CMP_MAX3_L	CMP_MAX for Block3 LSB Side	Address	0x7B

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMP_MAX	W	0	CMP_MAX[7:3] The read value is always 0.	
6		W	0		
5		W	0		
4		W	0		
3		W	0		
2		W	1	CMP_MAX[2:0] The read value is 0. The write value must always be 7.	
1		W	1		
0		W	1		

When PWMnACSTS.XREGACS = 1, do not write to this register from the XDATA BUS.

Register	CMP_MAX0_H	CMP_MAX for Block0 MSB Side	Address	0xF91D
Register	CMP_MAX1_H	CMP_MAX for Block1 MSB Side	Address	0xF95D
Register	CMP_MAX2_H	CMP_MAX for Block2 MSB Side	Address	0xF99D
Register	CMP_MAX3_H	CMP_MAX for Block3 MSB Side	Address	0xF9DD

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMP_MAX	R/W	0	MSB side of CMP_MAX	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	1		
1		R/W	1		
0		R/W	1		

Register	CMP_MAX0_H	CMP_MAX for Block0 MSB Side	Address	0x63
Register	CMP_MAX1_H	CMP_MAX for Block1 MSB Side	Address	0x6B
Register	CMP_MAX2_H	CMP_MAX for Block2 MSB Side	Address	0x73
Register	CMP_MAX3_H	CMP_MAX for Block3 MSB Side	Address	0x7B

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMP_MAX	W	0	MSB side of CMP_MAX The read value is always 0.	
6		W	0		
5		W	0		
4		W	0		
3		W	0		
2		W	0		
1		W	0		
0		W	0		

When PWMnACSTS.XREGACS = 1, do not write to this register from the XDATA BUS.

Register	CMP_A0_L	CMP_A for Block0 LSB Side	Address	0xF912
Register	CMP_A1_L	CMP_A for Block1 LSB Side	Address	0xF952
Register	CMP_A2_L	CMP_A for Block2 LSB Side	Address	0xF992
Register	CMP_A3_L	CMP_A for Block3 LSB Side	Address	0xF9D2
Register	CMP_B0_L	CMP_B for Block0 LSB Side	Address	0xF914
Register	CMP_B1_L	CMP_B for Block1 LSB Side	Address	0xF954
Register	CMP_B2_L	CMP_B for Block2 LSB Side	Address	0xF994
Register	CMP_B3_L	CMP_B for Block3 LSB Side	Address	0xF9D4
Register	CMP_C0_L	CMP_C for Block0 LSB Side	Address	0xF916
Register	CMP_C1_L	CMP_C for Block1 LSB Side	Address	0xF956
Register	CMP_C2_L	CMP_C for Block2 LSB Side	Address	0xF996
Register	CMP_C3_L	CMP_C for Block3 LSB Side	Address	0xF9D6
Register	CMP_D0_L	CMP_D for Block0 LSB Side	Address	0xF918
Register	CMP_D1_L	CMP_D for Block1 LSB Side	Address	0xF958
Register	CMP_D2_L	CMP_D for Block2 LSB Side	Address	0xF998
Register	CMP_D3_L	CMP_D for Block3 LSB Side	Address	0xF9D8

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMP_xxx	R/W	0	LSB side of the compare match register	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

Register	CMP_A0_L	CMP_A for Block0 LSB Side	Address	0x64
Register	CMP_A1_L	CMP_A for Block1 LSB Side	Address	0x6C
Register	CMP_A2_L	CMP_A for Block2 LSB Side	Address	0x74
Register	CMP_A3_L	CMP_A for Block3 LSB Side	Address	0x7C
Register	CMP_B0_L	CMP_B for Block0 LSB Side	Address	0x65
Register	CMP_B1_L	CMP_B for Block1 LSB Side	Address	0x6D
Register	CMP_B2_L	CMP_B for Block2 LSB Side	Address	0x75
Register	CMP_B3_L	CMP_B for Block3 LSB Side	Address	0x7D
Register	CMP_C0_L	CMP_C for Block0 LSB Side	Address	0x66
Register	CMP_C1_L	CMP_C for Block1 LSB Side	Address	0x6E
Register	CMP_C2_L	CMP_C for Block2 LSB Side	Address	0x76
Register	CMP_C3_L	CMP_C for Block3 LSB Side	Address	0x7E
Register	CMP_D0_L	CMP_D for Block0 LSB Side	Address	0x67
Register	CMP_D1_L	CMP_D for Block1 LSB Side	Address	0x6F
Register	CMP_D2_L	CMP_D for Block2 LSB Side	Address	0x77
Register	CMP_D3_L	CMP_D for Block3 LSB Side	Address	0x7F

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMP_xxx	W	0	LSB side of the compare match register The read value is always 0.	
6		W	0		
5		W	0		
4		W	0		
3		W	0		
2		W	0		
1		W	0		
0		W	0		

When PWMnACSTS.XREGACS = 1, do not write to this register from the XDATA BUS.

Register	CMP_A0_H	CMP_A for Block0 MSB Side	Address	0xF913
Register	CMP_A1_H	CMP_A for Block1 MSB Side	Address	0xF953
Register	CMP_A2_H	CMP_A for Block2 MSB Side	Address	0xF993
Register	CMP_A3_H	CMP_A for Block3 MSB Side	Address	0xF9D3
Register	CMP_B0_H	CMP_B for Block0 MSB Side	Address	0xF915
Register	CMP_B1_H	CMP_B for Block1 MSB Side	Address	0xF955
Register	CMP_B2_H	CMP_B for Block2 MSB Side	Address	0xF995
Register	CMP_B3_H	CMP_B for Block3 MSB Side	Address	0xF9D5
Register	CMP_C0_H	CMP_C for Block0 MSB Side	Address	0xF917
Register	CMP_C1_H	CMP_C for Block1 MSB Side	Address	0xF957
Register	CMP_C2_H	CMP_C for Block2 MSB Side	Address	0xF997
Register	CMP_C3_H	CMP_C for Block3 MSB Side	Address	0xF9D7
Register	CMP_D0_H	CMP_D for Block0 MSB Side	Address	0xF919
Register	CMP_D1_H	CMP_D for Block1 MSB Side	Address	0xF959
Register	CMP_D2_H	CMP_D for Block2 MSB Side	Address	0xF999
Register	CMP_D3_H	CMP_D for Block3 MSB Side	Address	0xF9D9

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMP_xxx	R/W	0	MSB side of the compare match register	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

Register	CMP_A0_H	CMP_A for Block0 MSB Side	Address	0x64
Register	CMP_A1_H	CMP_A for Block1 MSB Side	Address	0x6C
Register	CMP_A2_H	CMP_A for Block2 MSB Side	Address	0x74
Register	CMP_A3_H	CMP_A for Block3 MSB Side	Address	0x7C
Register	CMP_B0_H	CMP_B for Block0 MSB Side	Address	0x65
Register	CMP_B1_H	CMP_B for Block1 MSB Side	Address	0x6D
Register	CMP_B2_H	CMP_B for Block2 MSB Side	Address	0x75
Register	CMP_B3_H	CMP_B for Block3 MSB Side	Address	0x7D
Register	CMP_C0_H	CMP_C for Block0 MSB Side	Address	0x66
Register	CMP_C1_H	CMP_C for Block1 MSB Side	Address	0x6E
Register	CMP_C2_H	CMP_C for Block2 MSB Side	Address	0x76
Register	CMP_C3_H	CMP_C for Block3 MSB Side	Address	0x7E
Register	CMP_D0_H	CMP_D for Block0 MSB Side	Address	0x67
Register	CMP_D1_H	CMP_D for Block1 MSB Side	Address	0x6F
Register	CMP_D2_H	CMP_D for Block2 MSB Side	Address	0x77
Register	CMP_D3_H	CMP_D for Block3 MSB Side	Address	0x7F

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMP_xxx	W	0	MSB side of the compare match register The read value is always 0.	
6		W	0		
5		W	0		
4		W	0		
3		W	0		
2		W	0		
1		W	0		
0		W	0		

15.12.9. PWMnCNTMD (PWM Counter Mode for Block n) (n = 0 to 3)

When PWMnACSTS.XREGACS = 1, do not write to this register.

Register	PWM0CNTMD	PWM Counter Mode for Block0	Address	0xF920
Register	PWM1CNTMD	PWM Counter Mode for Block1	Address	0xF960
Register	PWM2CNTMD	PWM Counter Mode for Block2	Address	0xF9A0
Register	PWM3CNTMD	PWM Counter Mode for Block3	Address	0xF9E0

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	PWMCM	R/W	0	Counter block mode 0/1 0: Counter is set to up mode 1: Counter is set to up-down mode	

15.12.10. PWMnHCR0 (PWMnH Output Control0) (n = 0 to 3)

When PWMnACSTS.XREGACS = 1, do not write to this register.

Register	PWM0HCR0	PWM0H Output Control0	Address	0xF921
Register	PWM1HCR0	PWM1H Output Control0	Address	0xF961
Register	PWM2HCR0	PWM2H Output Control0	Address	0xF9A1
Register	PWM3HCR0	PWM3H Output Control0	Address	0xF9E1

Bit	Bit Name	R/W	Initial	Description	Remarks
7	PWM_MAX	R/W	0	Controls output by matching between PWMnH and CMP_MAX value 00: No change 01: Output is set to low level 10: Output is set to high level 11: Output is toggled	
6		R/W	0		
5	PWM_MIN	R/W	0	Controls output by matching between PWMnH and CMP_MIN value 00: No change 01: Output is set to low level 10: Output is set to high level 11: Output is toggled Effective in the up-down mode.	
4		R/W	0		
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	PWM_SET	W	0	Initializes the output level of PWMnH 00: No change 01: Output is set to low level 10: Output is set to high level 11: Setting prohibited Changing of the output level by writing to the bit has higher priority than other sources such as compare match or re-triggering. The read value is always 0.	
0		W	0		

15.12.11. PWMnLCR0 (PWMnL Output Control0) (n = 0 to 3)

When PWMnACSTS.XREGACS = 1, do not write to this register.

Register	PWM0LCR0	PWM0L Output Control0	Address	0xF922
Register	PWM1LCR0	PWM1L Output Control0	Address	0xF962
Register	PWM2LCR0	PWM2L Output Control0	Address	0xF9A2
Register	PWM3LCR0	PWM3L Output Control0	Address	0xF9E2

Bit	Bit Name	R/W	Initial	Description	Remarks
7	PWM_MAX	R/W	0	Controls output by matching between PWMnL and CMP_MAX value 00: No change 01: Output is set to low level 10: Output is set to high level 11: Output is toggled	
6		R/W	0		
5	PWM_MIN	R/W	0	Controls output by matching between PWMnH and CMP_MIN value 00: No change 01: Output is set to low level 10: Output is set to high level 11: Output is toggled Effective in the up-down mode.	
4		R/W	0		
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	PWM_SET	W	0	Initializes the output level of PWMnL 00: No change 01: Output is set to low level 10: Output is set to high level 11: Setting prohibited Changing of the output level by writing to the bit has higher priority than other sources such as compare match or re-triggering. The read value is always 0.	
0		W	0		

15.12.12. PWMnHCR1 (PWMnH Output Control1) (n = 0 to 3)

When PWMnACSTS.XREGACS = 1, do not write to this register.

Register	PWM0HCR1	PWM0H Output Control1	Address	0xF923
Register	PWM1HCR1	PWM1H Output Control1	Address	0xF963
Register	PWM2HCR1	PWM2H Output Control1	Address	0xF9A3
Register	PWM3HCR1	PWM3H Output Control1	Address	0xF9E3

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	VH1	R/W	0	Controls the output level of VH1 00: No change 01: Output is set to low level 10: Output is set to high level 11: Output is toggled	
2		R/W	0		
1	VH0	R/W	0	Controls the output level of VH0 00: No change 01: Output is set to low level 10: Output is set to high level 11: Output is toggled	
0		R/W	0		

15.12.13. PWMnLCR1 (PWMnL Output Control1) (n = 0 to 3)

When PWMnACSTS.XREGACS = 1, do not write to this register.

Register	PWM0LCR1	PWM0L Output Control1	Address	0xF924
Register	PWM1LCR1	PWM1L Output Control1	Address	0xF964
Register	PWM2LCR1	PWM2L Output Control1	Address	0xF9A4
Register	PWM3LCR1	PWM3L Output Control1	Address	0xF9E4

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	VL1	R/W	0	Controls the output level of VH1 00: No change 01: Output is set to low level 10: Output is set to high level 11: Output is toggled	
2		R/W	0		
1	VL0	R/W	0	Controls the output level of VH0 00: No change 01: Output is set to low level 10: Output is set to high level 11: Output is toggled	
0		R/W	0		

15.12.14. PWMnMODE (PWM n Operation Mode) (n = 0 to 3)

When PWMnACSTS.XREGACS = 1, do not write to this register.

Register	PWM0MODE	PWM0 Operation Mode	Address	0xF925
Register	PWM1MODE	PWM1 Operation Mode	Address	0xF965
Register	PWM2MODE	PWM2 Operation Mode	Address	0xF9A5
Register	PWM3MODE	PWM3 Operation Mode	Address	0xF9E5

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
6	BUFM	R/W	0	Buffer mode 0: Direct mode is set 1: Buffer mode is set	
5	UDBM	R/W	0	Data transfer timing from buffer register to compare match register in the up-down mode 00: Reserved 01: Transferred at CMP_MAX 10: Transferred at CMP_MIN 11: Transferred at both CMP_MAX and CMP_MIN The bit is effective only in the up-down mode of the buffer mode. In the up mode, when the CNT matches the CMP_MAX register value and the CNT is cleared, data is transferred from the buffer register to the compare match register.	
4		R/W	0		
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	PWMMD	R/W	0	PWM mode setting 00: PWM mode 0 is set 01: PWM mode 1 is set Other than above: Setting prohibited	
0		R/W	0		

15.12.15. PWMnRTRG (PWM Re-trigger Mode for Block n) (n = 0 to 3)

When PWMnACSTS.XREGACS = 1, do not write to this register.

Register	PWM0RTRG	PWM Re-trigger Mode for Block0	Address	0xF926
Register	PWM1RTRG	PWM Re-trigger Mode for Block1	Address	0xF966
Register	PWM2RTRG	PWM Re-trigger Mode for Block2	Address	0xF9A6
Register	PWM3RTRG	PWM Re-trigger Mode for Block3	Address	0xF9E6

Bit	Bit Name	R/W	Initial	Description	Remarks
7	PWMRTE	R/W	0	Re-trigger enable 0: Re-trigger is disabled 1: Re-trigger is enabled	
6	RTMSKD	R/W	0	Re-trigger event mask disable 0: Re-trigger event mask is enabled 1: Re-trigger event mask is disabled When RTMSKD = 0, the PWM events are not generated by compare match in the non-comparison period. When RTMSKD = 1, the PWM events are generated by compare match even in the non-comparison period. Control signals do not vary by compare match.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	RTRGPLS	R/W	0	Re-trigger detection timing setting 000: Re-trigger event is detected by the rising edge of the specified signal 001: Re-trigger event is detected by the falling edge of the specified signal 010: Re-trigger event is detected by the high level of the specified signal 011: Re-trigger event is detected by the low level of the specified signal 100: Re-trigger event is detected by both the rising and falling edges of the specified signal Other: Setting prohibited The bit can be set only when selecting No. 32 to No. 63 for the re-trigger event.	
3		R/W	0		
2		R/W	0		
1	PWMRTM	R/W	0	Re-trigger mode setting 00: Re-trigger mode A is set 01: Re-trigger mode B is set 10: Re-trigger mode C is set 11: Re-trigger mode D is set	
0		R/W	0		

15.12.16. PWMnRTRS (PWM Re-trigger Select for Block n) (n = 0 to 3)

When PWMnACSTS.XREGACS = 1, do not write to this register.

Register	PWM0RTRS	PWM Re-trigger Select for Block0	Address	0xF927
Register	PWM1RTRS	PWM Re-trigger Select for Block1	Address	0xF967
Register	PWM2RTRS	PWM Re-trigger Select for Block2	Address	0xF9A7
Register	PWM3RTRS	PWM Re-trigger Select for Block3	Address	0xF9E7

Bit	Bit Name	R/W	Initial	Description	Remarks
7	RTCBM[1]	R/W	0	Re-trigger C buffer update setting bit 1 (The bit has different meanings depending on the setting of PWM count mode.) - Up mode 0: Compare match register is not updated by re-trigger C events 1: Compare match register is updated to the buffer register when receiving the input of a re-trigger C event - Up-down mode 0: Buffer register is not updated by re-trigger C events during counting down 1: Compare match register is updated to the value of the buffer register when receiving the input of a re-trigger C event during counting down	
6	RTCBM[0]	R/W	0	Re-trigger C buffer update setting bit 0 (The bit has different meanings depending on the setting of PWM count mode.) - Up mode The bit is not related to operation. - Up-down mode 0: Buffer register is not updated by re-trigger C events during counting up 1: Compare match register is updated to the value of the buffer register when receiving the input of a re-trigger C event during counting up	
5	PWMRTS	R/W	0	Re-trigger event selection 000000: Event No. 0 … 011111: Event No. 31 100000: Event No. 32 … 111111: Event No. 63 For details, see Table 15-5.	
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

15.12.17. PWMnRTGC (PWM Re-trigger by CPU for Block n) (n = 0 to 3)

When PWMnACSTS.XREGACS = 1, do not write to this register.

Register	PWM0RTGC	PWM Re-trigger by CPU for Block0	Address	0xF928
Register	PWM1RTGC	PWM Re-trigger by CPU for Block1	Address	0xF968
Register	PWM2RTGC	PWM Re-trigger by CPU for Block2	Address	0xF9A8
Register	PWM3RTGC	PWM Re-trigger by CPU for Block3	Address	0xF9E8

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	PWMRTGC	R/W	0	CPU re-trigger setting Write 0: No change Write 1: When the CPU is set for the re-trigger event that is defined by the PWMnRTRS register, a re-trigger event is generated. The read value is always 0.	

15.12.18. PWMnRTL (PWM n Re-trigger Output Control) (n = 0 to 3)

When PWMnACSTS.XREGACS = 1, do not write to this register.

Register	PWM0RTL	PWM0 Re-trigger Output Control	Address	0xF929
Register	PWM1RTL	PWM1 Re-trigger Output Control	Address	0xF969
Register	PWM2RTL	PWM2 Re-trigger Output Control	Address	0xF9A9
Register	PWM3RTL	PWM3 Re-trigger Output Control	Address	0xF9E9

Bit	Bit Name	R/W	Initial	Description	Remarks
7	VTH_MAX	R/W	0	PWMnH pin setting in re-trigger mode C (CMP_MAX) 00: No change 01: Output is set to low level 10: Output is set to high level 11: Setting prohibited The PWMnH output is controlled when the counter loads the CMP_MAX register value in re-trigger mode C.	
6		R/W	0		
5	VTL_MAX	R/W	0	PWMnL pin setting in re-trigger mode C (CMP_MAX) 00: No change 01: Output is set to low level 10: Output is set to high level 11: Setting prohibited The PWMnL output is controlled when the counter loads the CMP_MAX register value in re-trigger mode C.	
4		R/W	0		
3	VTH	R/W	0	PWMnH pin setting in the re-trigger mode 00: No change 01: Output is set to low level 10: Output is set to high level 11: Setting prohibited Re-trigger mode A and re-trigger mode B: The PWMnH output is controlled. Re-trigger mode C: The PWMnH output is controlled when the counter loads the CMP_MIN register value.	
2		R/W	0		
1	VTL	R/W	0	PWMnL pin setting in the re-trigger mode 00: No change 01: Output is set to low level 10: Output is set to high level 11: Setting prohibited Re-trigger mode A/B: The PWMnL output is controlled. Re-trigger mode C: The PWMnL output is controlled when the counter loads CMP_MIN register value.	
0		R/W	0		

15.12.19. PWMnRTMC (PWM n Re-trigger Mask Control) (n = 0 to 3)

When PWMnACSTS.XREGACS = 1, do not write to this register.

Register	PWM0RTMC	PWM0 Re-trigger Mask Control	Address	0xF92A
Register	PWM1RTMC	PWM1 Re-trigger Mask Control	Address	0xF96A
Register	PWM2RTMC	PWM2 Re-trigger Mask Control	Address	0xF9AA
Register	PWM3RTMC	PWM3 Re-trigger Mask Control	Address	0xF9EA

Bit	Bit Name	R/W	Initial	Description	Remarks
7	RTME	R/W	0	Re-trigger mask enable 0: Re-trigger mask is disabled 1: Re-trigger mask is enabled	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	RTMC	R/W	0	Clock source for the re-trigger period 00: Count Clock Frequency/8 is set 01: Count Clock Frequency/16 is set 10: Count Clock Frequency/32 is set 11: Setting prohibited	
2		R/W	0		
1	RTMS	R/W	0	Start point of re-trigger mask 00: Start masking from rising edge of PWMnH 01: Start masking from falling edge of PWMnH 10: Start masking from rising edge of PWMnL 11: Start masking from falling edge of PWMnL	
0		R/W	0		

15.12.20. PWMnRTMP (PWM n Re-trigger Mask Period) (n = 0 to 3)

When PWMnACSTS.XREGACS = 1, do not write to this register.

Register	PWM0RTMP	PWM0 Re-trigger Mask Period	Address	0xF92B
Register	PWM1RTMP	PWM1 Re-trigger Mask Period	Address	0xF96B
Register	PWM2RTMP	PWM2 Re-trigger Mask Period	Address	0xF9AB
Register	PWM3RTMP	PWM3 Re-trigger Mask Period	Address	0xF9EB

Bit	Bit Name	R/W	Initial	Description	Remarks
7	RTMP	R/W	0	Re-trigger masking period Period = Clock source cycle x (RTMP + 1) The clock source cycle is defined by the PWMnRTMC.RTMC bits.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

15.12.21. BUF_MIN/MAXn (BUF_MIN/MAX for Block n LSB/MSB Side) (n = 0 to 3)

When PWMnACSTS.XREGACS = 1, do not write to this register from the XDATA BUS.

Register	BUF_MIN0_L	BUF_MIN for Block0 LSB Side	Address	0xF92C
Register	BUF_MIN1_L	BUF_MIN for Block1 LSB Side	Address	0xF96C
Register	BUF_MIN2_L	BUF_MIN for Block2 LSB Side	Address	0xF9AC
Register	BUF_MIN3_L	BUF_MIN for Block3 LSB Side	Address	0xF9EC

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BUF_MIN	R/W	0	BUF_MIN[7:3]	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R	0	BUF_MIN[2:0] The read value is 0. The write value must always be 0.	
1		R	0		
0		R	0		

Register	BUF_MIN0_L	BUF_MIN for Block0 LSB Side	Address	0x60
Register	BUF_MIN1_L	BUF_MIN for Block1 LSB Side	Address	0x68
Register	BUF_MIN2_L	BUF_MIN for Block2 LSB Side	Address	0x70
Register	BUF_MIN3_L	BUF_MIN for Block3 LSB Side	Address	0x78

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BUF_MIN	W	0	BUF_MIN[7:3] The read value is always 0.	
6		W	0		
5		W	0		
4		W	0		
3		W	0		
2		W	0	BUF_MIN[2:0] The read value is 0. The write value must always be 0.	
1		W	0		
0		W	0		

When PWMnACSTS.XREGACS = 1, do not write to this register from the XDATA BUS.

Register	BUF_MIN0_H	BUF_MIN for Block0 MSB Side	Address	0xF92D
Register	BUF_MIN1_H	BUF_MIN for Block1 MSB Side	Address	0xF96D
Register	BUF_MIN2_H	BUF_MIN for Block2 MSB Side	Address	0xF9AD
Register	BUF_MIN3_H	BUF_MIN for Block3 MSB Side	Address	0xF9ED

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BUF_MIN	R/W	0	MSB side of BUF_MIN	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

Register	BUF_MIN0_H	BUF_MIN for Block0 MSB Side	Address	0x60
Register	BUF_MIN1_H	BUF_MIN for Block1 MSB Side	Address	0x68
Register	BUF_MIN2_H	BUF_MIN for Block2 MSB Side	Address	0x70
Register	BUF_MIN3_H	BUF_MIN for Block3 MSB Side	Address	0x78

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BUF_MIN	W	0	MSB side of BUF_MIN The read value is always 0.	
6		W	0		
5		W	0		
4		W	0		
3		W	0		
2		W	0		
1		W	0		
0		W	0		

When PWMnACSTS.XREGACS = 1, do not write to this register from the XDATA BUS.

Register	BUF_MAX0_L	BUF_MAX for Block0 LSB Side	Address	0xF92E
Register	BUF_MAX1_L	BUF_MAX for Block1 LSB Side	Address	0xF96E
Register	BUF_MAX2_L	BUF_MAX for Block2 LSB Side	Address	0xF9AE
Register	BUF_MAX3_L	BUF_MAX for Block3 LSB Side	Address	0xF9EE

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BUF_MAX	R/W	0	BUF_MAX[7:3]	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R	1	BUF_MAX[2:0] The read value is 7. The write value must always be 7.	
1		R	1		
0		R	1		

Register	BUF_MAX0_L	BUF_MAX for Block0 LSB Side	Address	0x61
Register	BUF_MAX1_L	BUF_MAX for Block1 LSB Side	Address	0x69
Register	BUF_MAX2_L	BUF_MAX for Block2 LSB Side	Address	0x71
Register	BUF_MAX3_L	BUF_MAX for Block3 LSB Side	Address	0x79

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BUF_MAX	W	0	BUF_MAX[7:3] The read value is always 0.	
6		W	0		
5		W	0		
4		W	0		
3		W	0		
2		W	1	BUF_MAX[2:0] The read value is 0. The write value must always be 7.	
1		W	1		
0		W	1		

When PWMnACSTS.XREGACS = 1, do not write to this register from the XDATA BUS.

Register	BUF_MAX0_H	BUF_MAX for Block0 MSB Side	Address	0xF92F
Register	BUF_MAX1_H	BUF_MAX for Block1 MSB Side	Address	0xF96F
Register	BUF_MAX2_H	BUF_MAX for Block2 MSB Side	Address	0xF9AF
Register	BUF_MAX3_H	BUF_MAX for Block3 MSB Side	Address	0xF9EF

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BUF_MAX	R/W	0	MSB side of BUF_MAX	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

Register	BUF_MAX0_H	BUF_MAX for Block0 MSB Side	Address	0x61
Register	BUF_MAX1_H	BUF_MAX for Block1 MSB Side	Address	0x69
Register	BUF_MAX2_H	BUF_MAX for Block2 MSB Side	Address	0x71
Register	BUF_MAX3_H	BUF_MAX for Block3 MSB Side	Address	0x79

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BUF_MAX	W	0	MAB side of BUF_MAX The read value is always 0.	
6		W	0		
5		W	0		
4		W	0		
3		W	0		
2		W	0		
1		W	0		
0		W	0		

15.12.22. BUF_A/B/C/Dn_L/H (BUF_A/B/C/D for Block n LSB/MSB Side) (n = 0 to 3)

The BUF_xn_L/H register is mapped to the same address. To access these addresses, access to the BUF_xn_L register first, and then access the BUF_xn_H register. Before data is written in the same register connected with SFR BUS again, make sure that PWMnACSTS.SFRACS = 0. When writing data to the BUF_xn_H register or the CMP_xn_H register, the PWMnACSTS.SFRACS bit is set to 1.

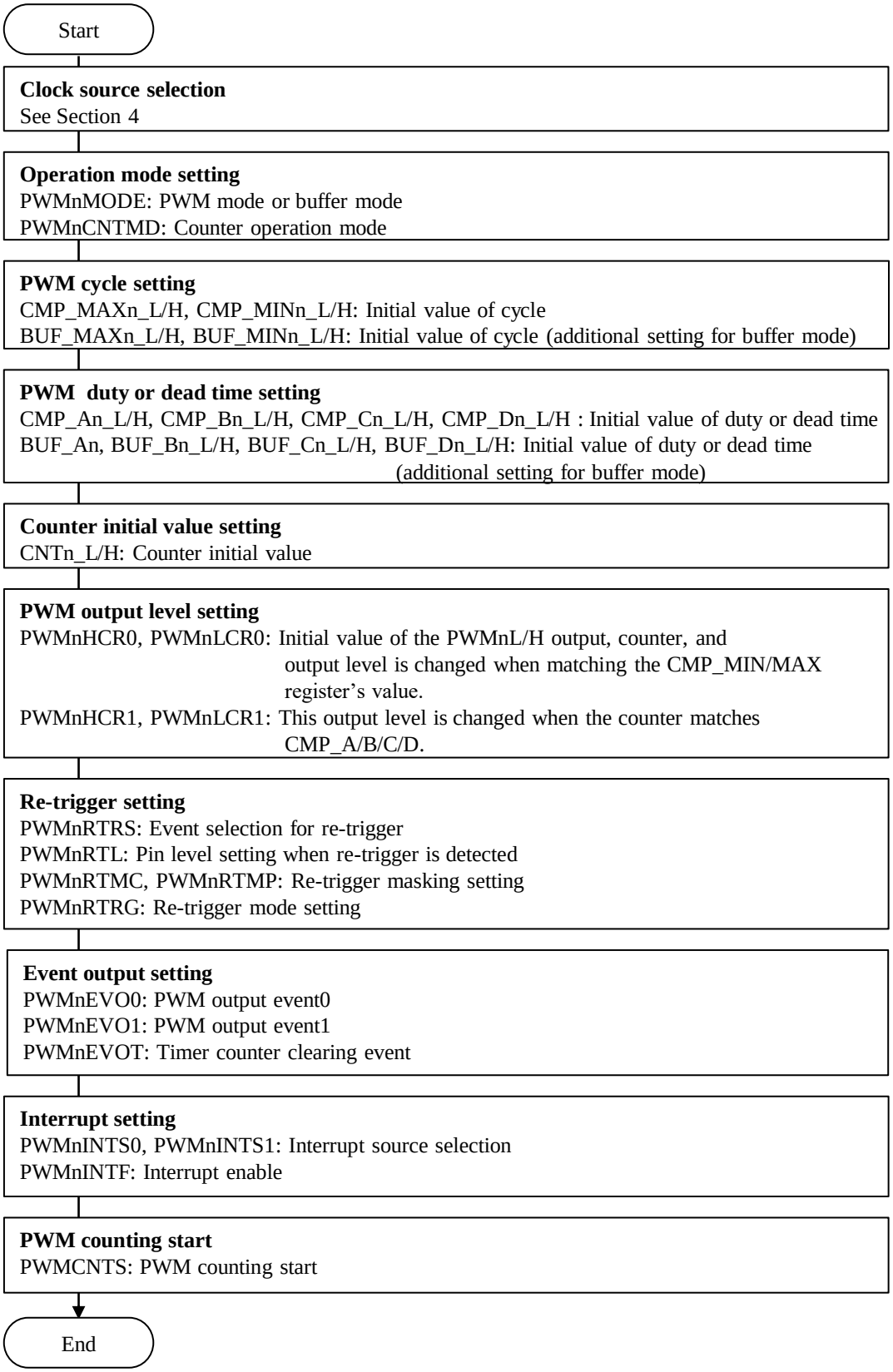
Register	BUF_A0_L	BUF_A for Block0 LSB Side	Address	0xE4
Register	BUF_A1_L	BUF_A for Block1 LSB Side	Address	0xEC
Register	BUF_A2_L	BUF_A for Block2 LSB Side	Address	0xF4
Register	BUF_A3_L	BUF_A for Block3 LSB Side	Address	0xFC
Register	BUF_B0_L	BUF_B for Block0 LSB Side	Address	0xE5
Register	BUF_B1_L	BUF_B for Block1 LSB Side	Address	0xED
Register	BUF_B2_L	BUF_B for Block2 LSB Side	Address	0xF5
Register	BUF_B3_L	BUF_B for Block3 LSB Side	Address	0xFD
Register	BUF_C0_L	BUF_C for Block0 LSB Side	Address	0xE6
Register	BUF_C1_L	BUF_C for Block1 LSB Side	Address	0xEE
Register	BUF_C2_L	BUF_C for Block2 LSB Side	Address	0xF6
Register	BUF_C3_L	BUF_C for Block3 LSB Side	Address	0xFE
Register	BUF_D0_L	BUF_D for Block0 LSB Side	Address	0xE7
Register	BUF_D1_L	BUF_D for Block1 LSB Side	Address	0xEF
Register	BUF_D2_L	BUF_D for Block2 LSB Side	Address	0xF7
Register	BUF_D3_L	BUF_D for Block3 LSB Side	Address	0xFF

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BUF_xxx	R/W	0	LSB side of buffer register	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

Register	BUF_A0_H	BUF_A for Block0 MSB Side	Address	0xE4
Register	BUF_A1_H	BUF_A for Block1 MSB Side	Address	0xEC
Register	BUF_A2_H	BUF_A for Block2 MSB Side	Address	0xF4
Register	BUF_A3_H	BUF_A for Block3 MSB Side	Address	0xFC
Register	BUF_B0_H	BUF_B for Block0 MSB Side	Address	0xE5
Register	BUF_B1_H	BUF_B for Block1 MSB Side	Address	0xED
Register	BUF_B2_H	BUF_B for Block2 MSB Side	Address	0xF5
Register	BUF_B3_H	BUF_B for Block3 MSB Side	Address	0xFD
Register	BUF_C0_H	BUF_C for Block0 MSB Side	Address	0xE6
Register	BUF_C1_H	BUF_C for Block1 MSB Side	Address	0xEE
Register	BUF_C2_H	BUF_C for Block2 MSB Side	Address	0xF6
Register	BUF_C3_H	BUF_C for Block3 MSB Side	Address	0xFE
Register	BUF_D0_H	BUF_D for Block0 MSB Side	Address	0xE7
Register	BUF_D1_H	BUF_D for Block1 MSB Side	Address	0xEF
Register	BUF_D2_H	BUF_D for Block2 MSB Side	Address	0xF7
Register	BUF_D3_H	BUF_D for Block3 MSB Side	Address	0xFF

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BUF_xxx	R/W	0	MSB side of buffer register	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

15.13. Example of PWM Setting



15.14. Usage Notes and Restrictions

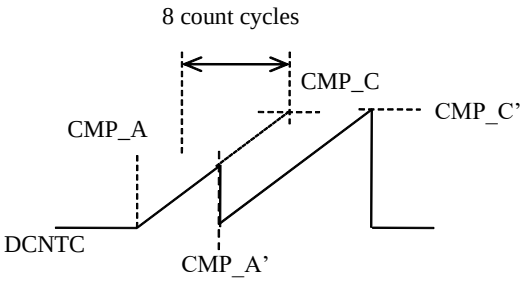
15.14.1. Restrictions on Automatic Dead Time Mode

● **Description**

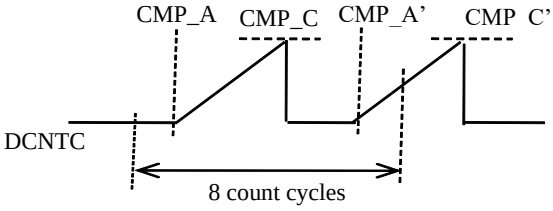
When the PWM operates in the automatic dead time mode (PWMnMODE.PWMMD = 0b01) and one of the following conditions is satisfied, the dead time counter does not operate correctly. Therefore, the compare match between the dead time counter and the compare match register is not detected correctly, and the PWM output signal, event, and interrupt are not generated correctly by compare match.

● **Conditions**

- Condition A: When another event which starts counting of dead time again is generated in one to 8 count cycles before the timing at which dead time counting ends (see Figure 15-17 (a)) while the dead time counter is operating.
- Condition B: When, the same dead time counter starts counting twice or more within 8 count cycles (see Figure 15-17 (b)) while the PWM counter is operating in the up mode (i.e., PWMnCNTMD.PWMCM = 0).



(a) Condition A



(b) Condition B

CMP_A: Starting source of dead time counting
CMP_C: End of dead time counting started by CMP_A
CMP_A': Starting source of new dead time counting
CMP_C': End of dead time counting started by CMP_A'

Figure 15-17. Prohibited Waveform of Dead Time Counter (DCNTC)

● **Workaround**

- Workaround against condition A:

  Apply one of the following 2 methods.

  ● Not start counting while the dead time counter is operating.
  ● When counting of dead time should be started again while the dead time counter operates, generate an event to start dead time counting earlier than over 8 count cycles before the end of dead time counting.

- Workaround against condition B:

  Ensure 8 or more count cycles of the dead time counter between the first start and second start of dead time counting.

15.14.2. Restrictions on Re-trigger Mode

15.14.2.1. Restrictions on Re-trigger Mode A/B/D in the Non-comparison Period

- Each pin level of the PWMnL/H is not changed by each compare match.
- When PWMnRTRG.RTMSKD = 0, events are not generated by the compare match. When events are used in a non-comparison period, write 1 to the PWMnRTRG.RTMSKD bit.
- When PWMnRTRG.RTMSKD = 0, the interrupt flag generated by compare match (PWMnINTF.PWMIF0/1) is not set.
- Even in the non-comparison period, each dead time counter is started by the normal method.
- In the non-comparison period, each compare match register of the buffer mode is updated correctly.

15.14.2.2. Restrictions on Re-trigger Mode B in the Non-comparison Period

- The end timing of the non-comparison period is delayed by up to 8 count cycles from the timing of the compare match between the CMP_MIN register value and CNT.
- When PWMnRTRG.RTMSKD = 0, the compare match event between the values of the CNT and the CMP_MIN register is not detected in the non-comparison period.

15.14.2.3. Restrictions on Re-trigger Mode D in the Non-comparison Period

- The start timing of the non-comparison period is delayed by up to 8 count cycles from the timing of compare match between the values of the CNT and the CMP_MIN register.
- The end timing of the non-comparison period is delayed by up to 8 count cycles from the timing of compare match between the values of the CNT and the CMP_MIN register.
- When the compare match event between the values of the CNT and the CMP_MIN register is detected at the start timing of the non-comparison period, this compare match event is not detected at the end timing of the non-comparison period.

Figure 15-18 shows the non-comparison periods of re-trigger modes B and D.

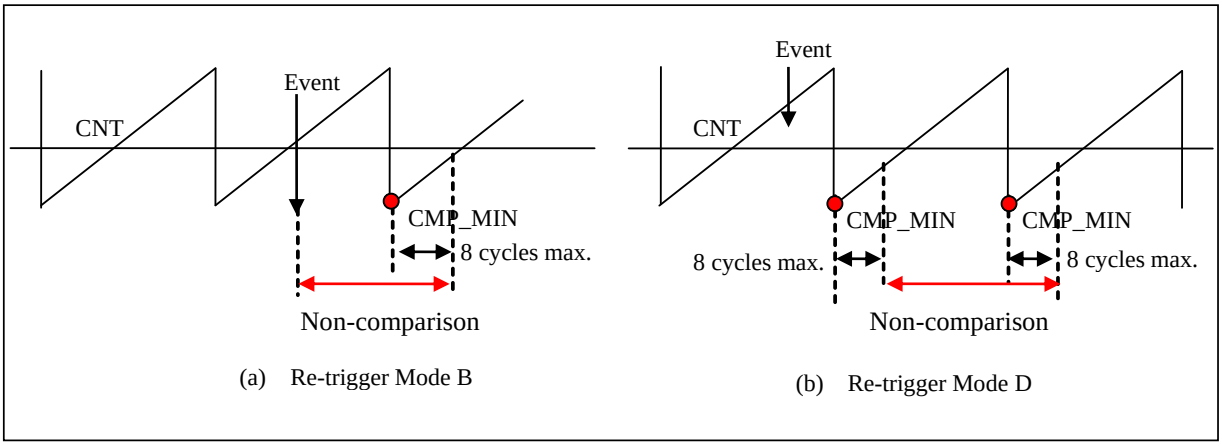


Figure 15-18. Non-comparison Period

15.14.2.4. Restrictions on Re-trigger Masking Operation

The Re-trigger mask starts before 4 to 20 count cycles from the toggle timing of the PWM output pin defined by the PWMxRTMC.RTMS bits.

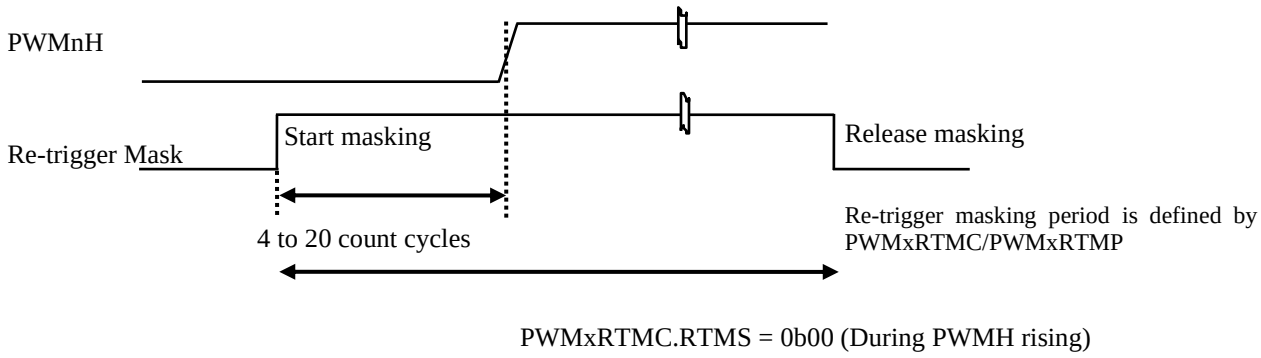


Figure 15-19. Re-trigger Masking Timing

The control logic to judge start of the re-trigger masking monitors the internal state of the PWM module; however, it does not directly monitor the toggling of the PWM output pin.

Internal state of the PWM is reflected to the PWM output pin after 4 count cycles or more.

The delay between the internal state controlling the toggling of the PWM output and the re-trigger masking is about 20 cycles (max.). This value varies depending on the semi-conductor process, power supply voltage, and ambient temperature. When setting re-trigger masking, these sources must be taken into account.

16. Watchdog Timer (WDT)

16.1. Overview

Table 16-1 shows the watchdog timer (WDT) module functional descriptions.

Table 16-1. WDT Functional Descriptions

Item	Description
Count Clock	Clock obtained by dividing CLKFAST 8 division ratios
Counter	8-bit counter × 1 channel Counter rewrite operation protection
Operation Mode	Watchdog timer mode and interval timer mode
Watchdog Timer Mode	Outputs the reset to internal module when the counter timer is overflowed
Interval Timer Mode	Generates the interval timer interrupt when the counter timer is overflowed

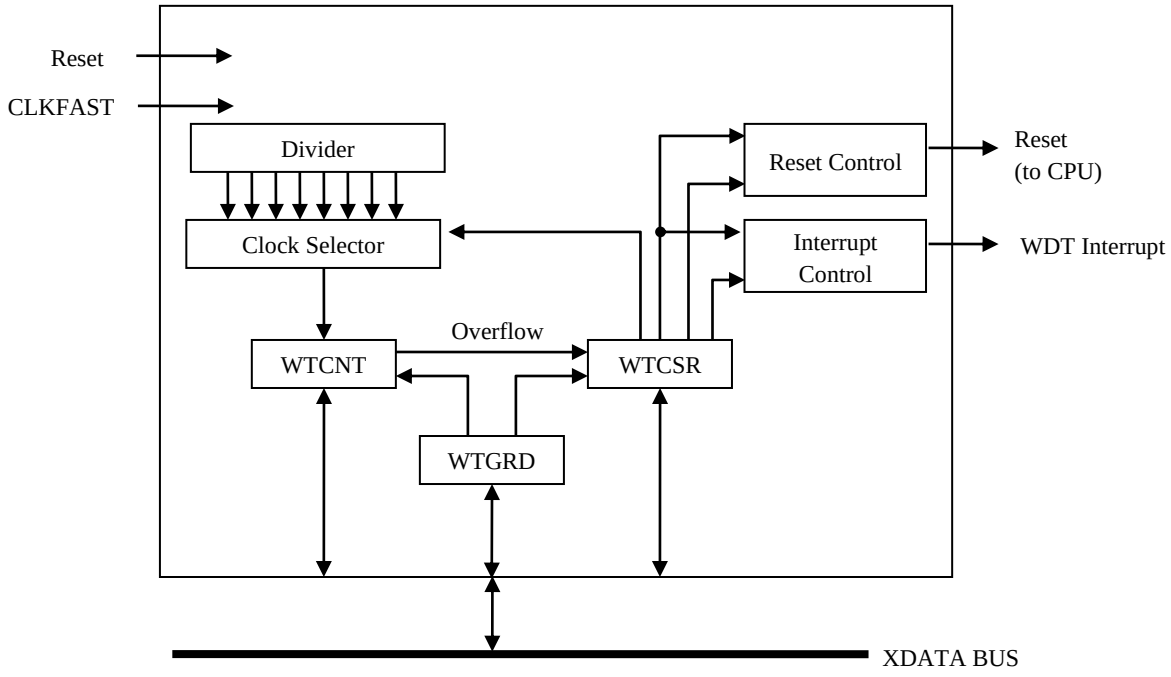


Figure 16-1. WDT Block Diagram

16.2. Register Descriptions

16.2.1. List of Registers

Table 16-2. List of Registers

Symbol	Name	Address	Initial Value
WTCNT	Watchdog Timer Counter	0xFE80	0x00
WTCSR	Watchdog Timer Control/Status	0xFE81	0x00
WTGRD	Watchdog Timer Register Access Guard	0xFE82	0x00

16.2.2. WTCNT (Watchdog Timer Counter)

When WTCSR.TME = 1, the WTCNT register starts a count by the internal clock determined by the WTCSR.CKS bit. When TME = 0, the WTCNT register holds the count value and stops the count.

While the OCD stops the CPU instruction execution, the watchdog timer stops the count.

Register	WTCNT	Watchdog Timer Counter	Address	0xFE80

Bit	Bit Name	R/W	Initial	Description	Remarks
7	WTCNT	R/W	0	Watchdog time counter When an overflow occurs, the watchdog timer operates as follows: ● Generates a reset in watchdog timer mode ● Generates an iterrupt in the interval timer mode	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

16.2.3. WTCSR (Watchdog Timer Control/Status)

Before setting the CKS bit, set the TME bit to 0, and stop the WTCNT register count.

When the WTCNT register count overflows, the WOVF and IOVF bits are not initialized by the watchdog timer reset. Therefore, the WOVF and IOVF bits must be cleared after the watchdog timer reset is released.

Register	WTCSR	Watchdog Timer Control/Status	Address	0xFE81

Bit	Bit Name	R/W	Initial	Description	Remarks
7	TME	R/W	0	Timer enable 0: The up-count is stopped; the value of the WTCNT register is held 1: The timer is enabled	
6	TM	R/W	0	Timer mode setting 0: Interval timer mode 1: Watchdog timer mode	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	WOVF	R/C	0	Watchdog timer overflow Read 0: No overflow Read 1: The WTCNT register count has overflowed in watchdog timer mode Write 0: No change Write 1: The bit is cleared	
3	IOVF	R/C	0	Interval timer overflow Read 0: No overflow Read 1: The WTCNT register count has overflowed in watchdog timer mode Write 0: No change Write 1: The bit is cleared	
2	CKS	R/W	0	WTCNT clock selection Division ratio — Counter period (CLKFAST = 60 MHz) 000 : $\frac{1}{2}^{13}$ — 137 µs 001 : $\frac{1}{2}^{14}$ — 273 µs 010 : $\frac{1}{2}^{15}$ — 546 µs 011 : $\frac{1}{2}^{16}$ — 1.09 ms 100 : $\frac{1}{2}^{17}$ — 2.18 ms 101 : $\frac{1}{2}^{18}$ — 4.37 ms 110 : $\frac{1}{2}^{19}$ — 8.74 ms 111 : $\frac{1}{2}^{20}$ — 17.5 ms	
1		R/W	0		
0		R/W	0		

16.2.4. WTGRD (Watchdog Timer Register Access Guard)

Register	WTGRD		Watchdog Timer Register Access Guard	Address	0xFE82
Bit	Bit Name	R/W	Initial	Description	Remarks
7	WTGRD	R/W	0	Set a key code to this register for writing to the WTCNT/WTCSR register. After writing to the WTCNT/WTCSR register, the WTGRD register is cleared to 0x00. Key code: WTCNT: 0x5A WTCSR: 0xA5	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

16.3. Reset Configuration

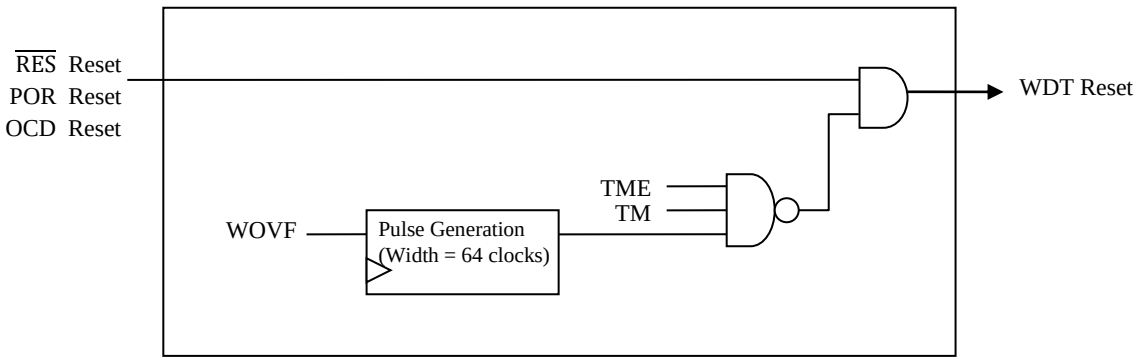


Figure 16-2. Reset Configuration

16.4. Interrupt Configuration

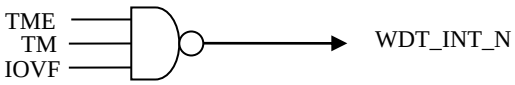


Figure 16-3. Interrupt Configuration

16.5. Prescaler

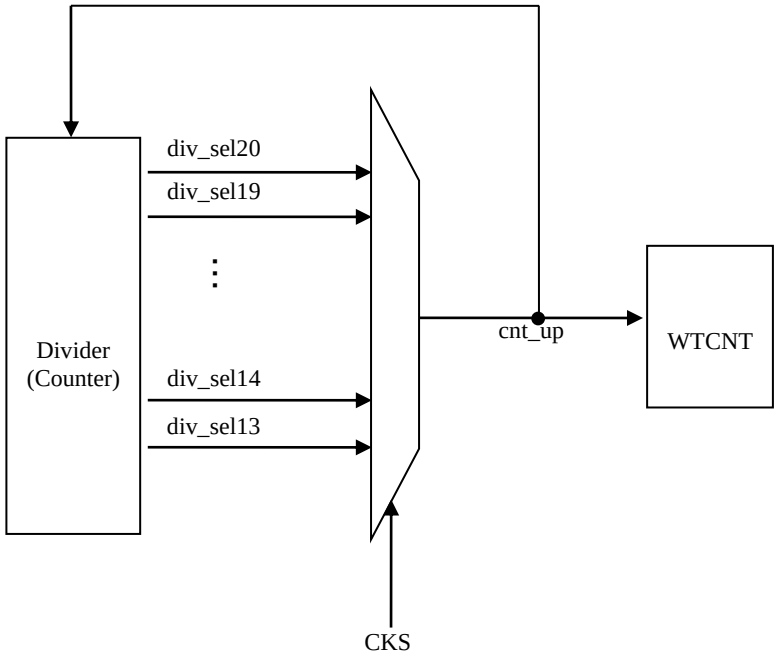


Figure 16-4. Prescaler Configuration

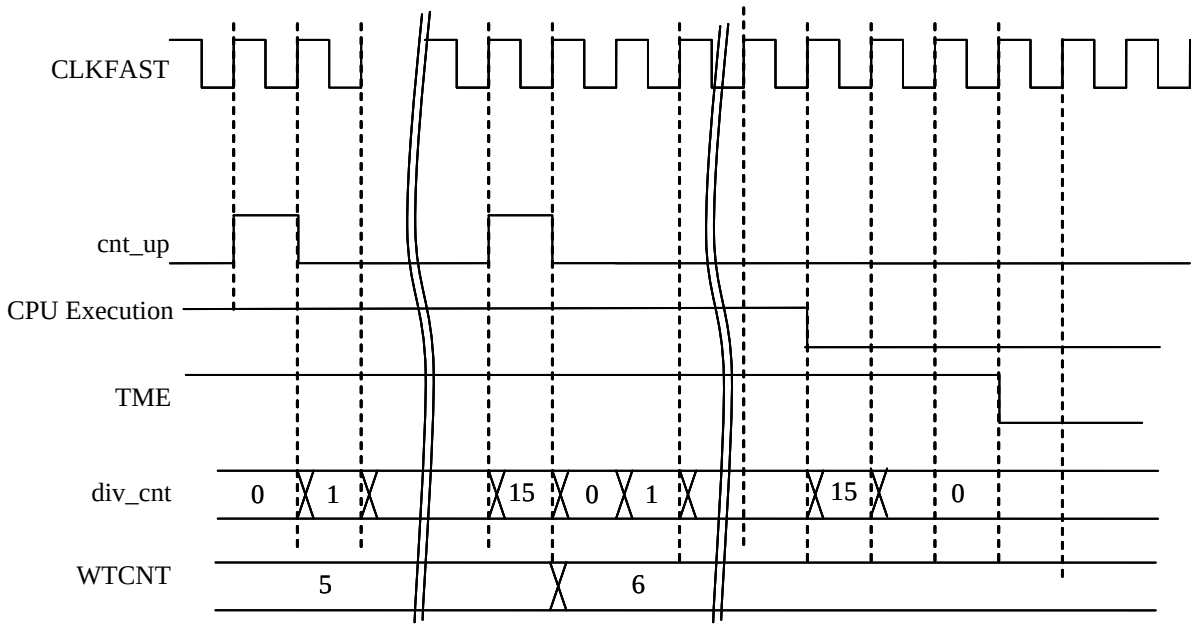


Figure 16-5. Prescaler Timing

16.6. Operation

16.6.1. Writing to WTCNT and WTCSR Registers

The following cares should be taken when writing to the WTCNT and WTCSR registers.

- When the value of the WTGRD register is other than 0x5A and 0xA5, the writing to the WTCNT and WTCSR registers cannot be performed.
- After 0x5A is written to the WTGRD register, write to the WTCNT register. Figure 16-6 shows the operation that is 0x00 is written to the WTCNT register.
- After 0xA5 is written to the WTGRD register, write to the WTCSR register. Figure 16-6 shows the operation that is 1 is written to the WTCSR.TM and WTCSR.TME bits.

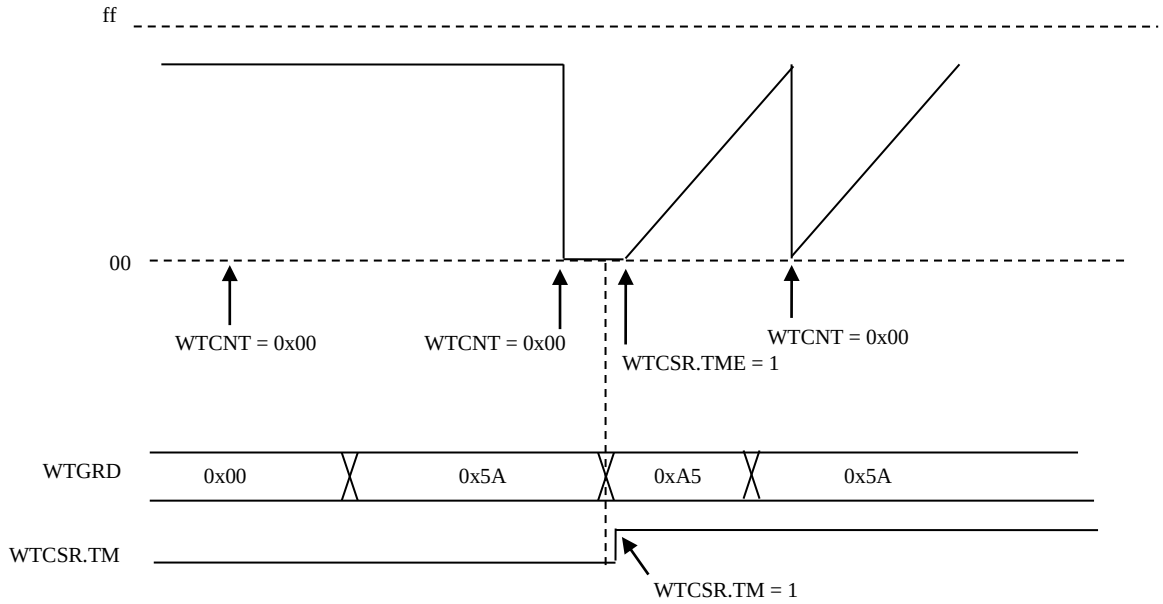


Figure 16-6. Writing to WTCNT and WTCSR Registers

16.6.2. Watchdog Timer Mode

To start the timer up-count operation, set WTCSR.TME = 1. During the normal operation, the value of the WTCNT register must be periodically set again so as not to overflow the WTCNT register count. When the WTCNT register count overflows, the watchdog timer reset is generated. At this point, the WTCNT register continues the count operation. The reset period is 64 clocks (1.07 μs in CLKFAST = 60 MHz). After the reset, write 1 to the WTCSR.WOVF bit to clear the flag. While the WTCSR.WOVF bit is not cleared, the next watchdog timer reset is not generated. To stop the count, set WTCSR.TME = 0; and then the value is held. To restart the count from the held value, set WTCSR.TME = 1. When the value is written to the WTCNT register during the count, the up-count operation starts from the written value.

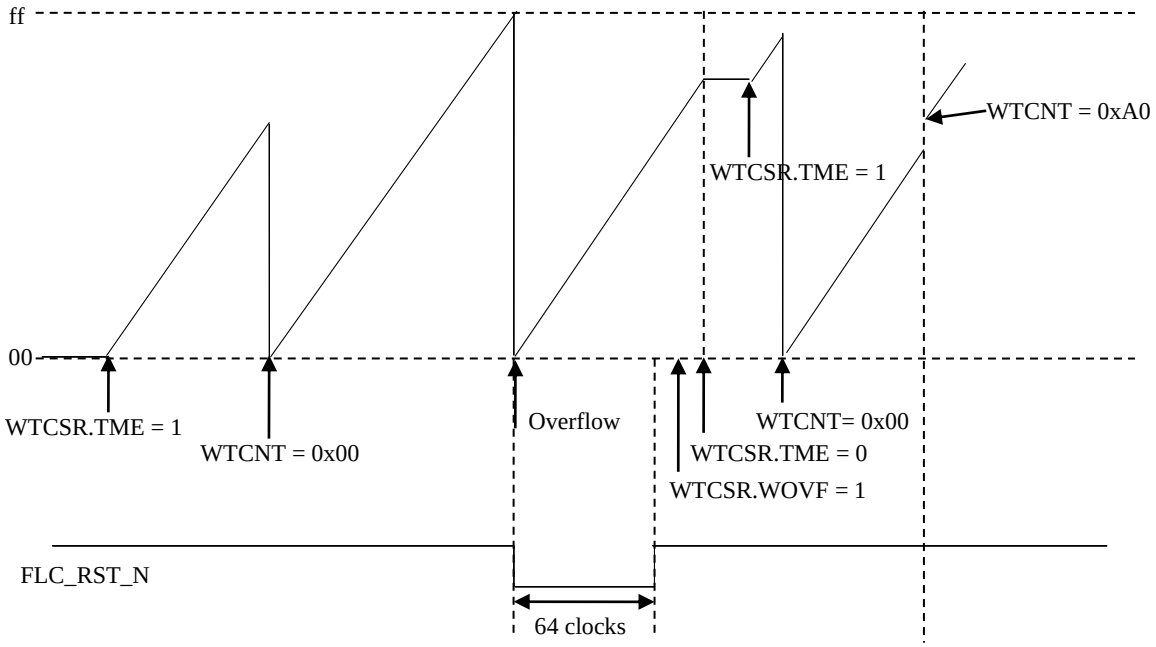


Figure 16-7. Operation in Watchdog Timer Mode

16.6.3. Interval Timer Mode

In the interval timer mode, an interrupt request for the interval interrupt is generated. The initial interval period is the WTCNT register up-count time from 0x00 to overflow. The initial interval period is determined by the WTCSR.CKS bit. To generate the interrupt in a shorter period than the initial interval period, set the value to the WTCNT register again in the interrupt routine. The writing to the WTCNT register is completed in a much shorter time than the count operation. To start the up-count operation of watchdog timer, set WTCSR.TME = 1. When the WTCNT register count is overflowed, the interrupt is generated. Write 1 to the WTCSR.IOVF bit in the interrupt routine to clear the interrupt flag.

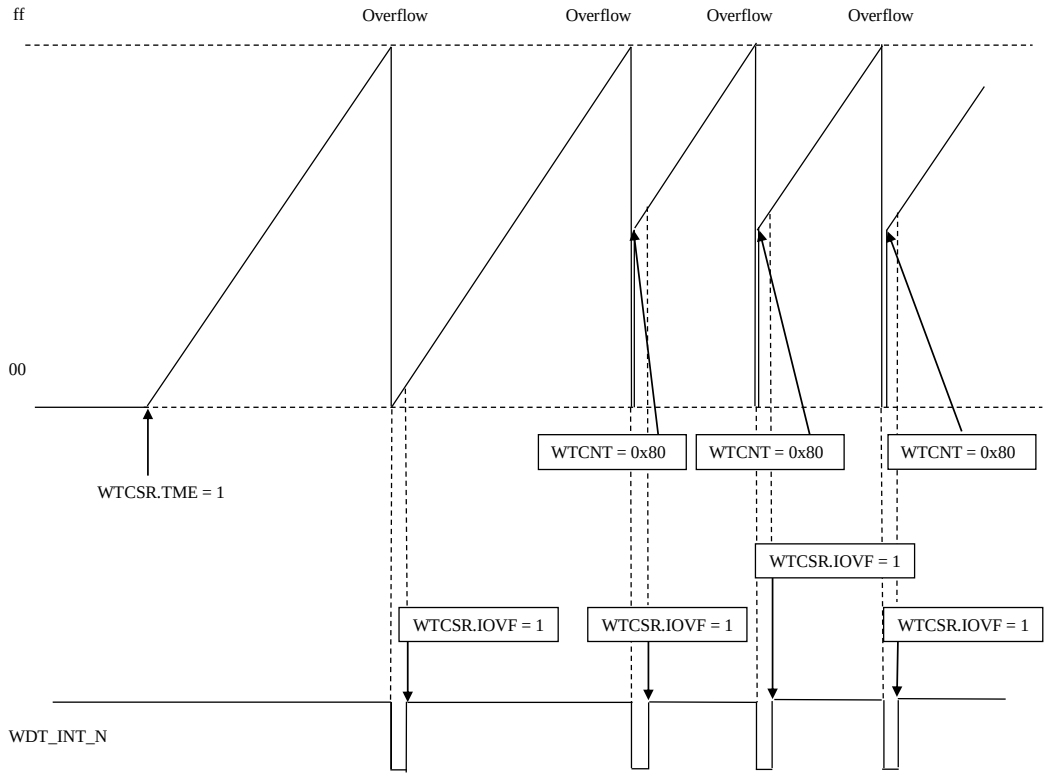


Figure 16-8. Operation in Interval Timer Mode

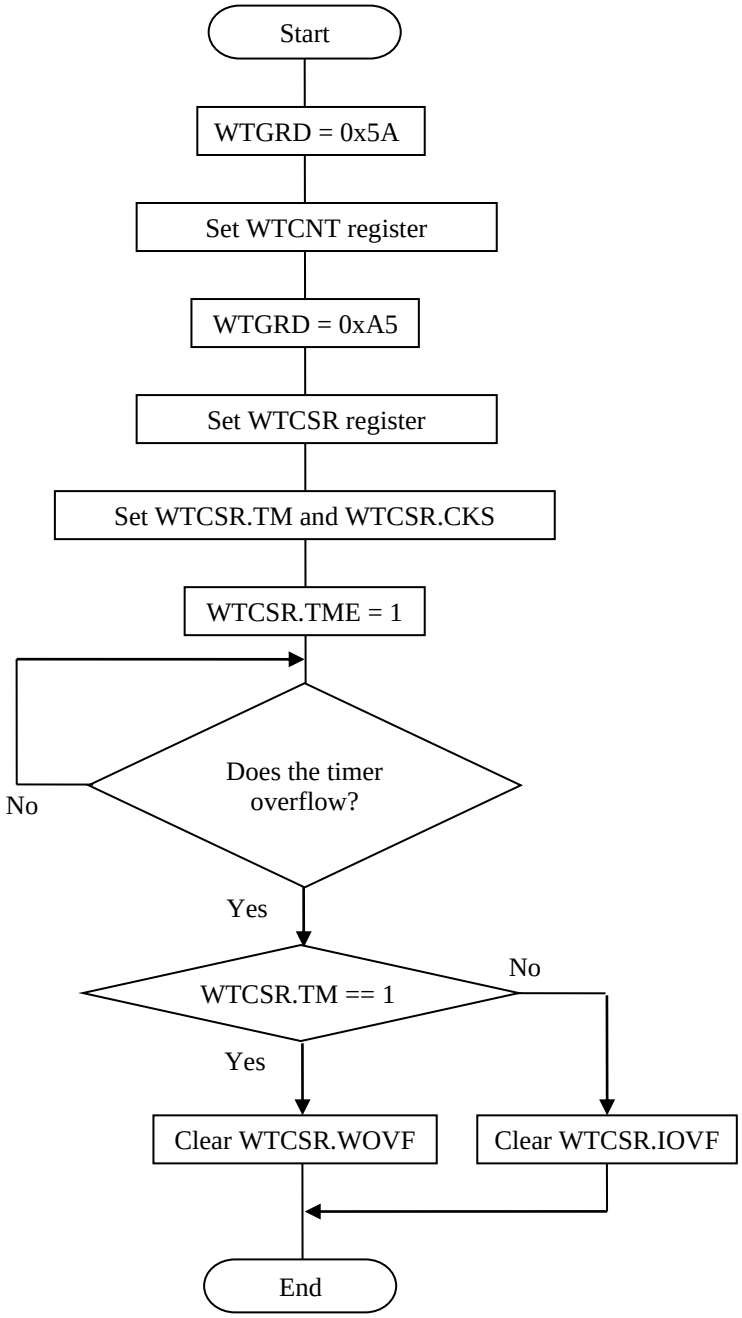


Figure 16-9. Flow Chart of Watchdog Timer Operation

17. 16-bit Timer (TMR)

17.1. Overview

The LSI has four 16-bit timer modules (TMR0, TMR1, TMR2, and TMR3). Table 17-1 shows the timer (TMR) functional descriptions, and Figure 17-1 shows the TMR block diagram.

Table 17-1. TMR Functional Descriptions

Item	Description
Number of Channels	4 channels Channel0 and Channel1: Operated by the 32-bit timer (cascade mode) Channel2 and Channel3: Operated by the 32-bit timer (cascade mode)
Input/Output	TIOAn (n = 0 to 3): CMPA input capture/phase A input for phase counting mode/compare match output A TIOBn (n = 0 to 3): CMPB input capture/phase B input for phase counting mode/compare match output B TICn (n = 2/3): Phase Z input for phase counting mode
Operation Mode	Normal mode (cascade mode, input capture, and compare match output) Phase counting mode (cascade mode and input capture)
Normal Mode	Counter clock: CLKFAST is divided by 1, 4, 16, 64, 256, 1024, 4096, and 16384 Capture events: CMPA: TIOAn input (rising edge, falling edge, and both edges) Compare match A/B between TMR0 and TMR1 Events of Comparator0 to Comparator5 CMPB: TIOBn input (rising edge, falling edge, both edges), Compare match A/B between TMR0 and TMR1 Events of Comparator0 to Comparator5 Counter clearing events: Compare match A/B TIOAn/TIOBn capturing Event clearing from PWM
Phase Counting Mode	Counter clock: TIOAn/TIOBn input or comparator output phase Capture events: CMPA/B: Compare match A/B between TMR0 and TMR1 Outputs of Comparator0 to Comparator5 Counter clearing events: TIOAn/TIOBn/TICn input event Event clearing from PWM
Data Transfer by DSAC	The TCMPAxn, TCMPBxn, TBUFAxn, and TBUFBxn registers are mapped to both the XDATA BUS and SFR BUS areas. The CPU can read from or write to the TCMPAxn/Bxn register by the MOVX instruction.
Interrupt	Compare match or input capture A/B Counter overflow Counter underflow (only in the phase counting mode) TIOAn/TIOBn/TICn input event

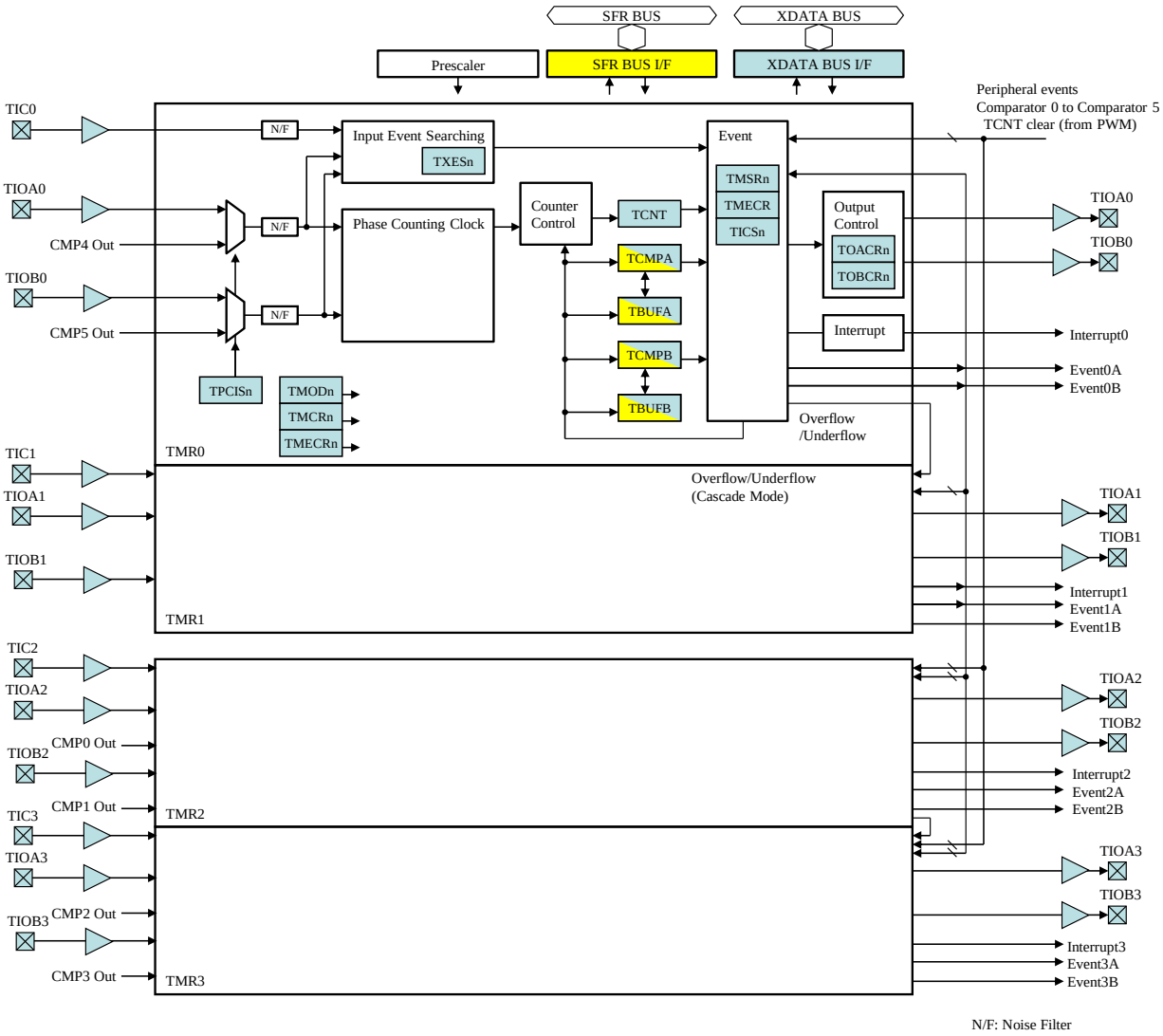


Figure 17-1. TMR Block Diagram

17.2. Register Descriptions

Table 17-2. List of XDATA BUS Registers

Symbol	Name	Address	Initial Value
TMOD0	Timer0 Control Mode Register	0xFA00	0x00
TMOD1	Timer1 Control Mode Register	0xFA01	0x00
TMSR0	Timer0 Status Register	0xFA02	0x00
TMSR1	Timer1 Status Register	0xFA03	0x00
TMCR0	Timer0 Control Register	0xFA04	0x00
TMCR1	Timer0 Control Register	0xFA05	0x00
TMECR0	Timer0 Event Clear Register	0xFA06	0x00
TMECR1	Timer1 Event Clear Register	0xFA07	0x00
TEMOD0	Timer0 Extend Mode Register	0xFA08	0x00
TEMOD1	Timer1 Extend Mode Register	0xFA09	0x00
TICS0	Timer0 Input Capture Select Register	0xFA0A	0x00
TICS1	Timer1 Input Capture Select Register	0xFA0B	0x00
TXES0	Timer0 External Event Select Register	0xFA0C	0x00
TXES1	Timer1 External Event Select Register	0xFA0D	0x00
TPSNF0	Timer0 Prescaler for Noise Filter Register	0xFA0E	0x00
TPSNF1	Timer1 Prescaler for Noise Filter Register	0xFA0F	0x00
TCMPAL0	Timer0 Compare Match A Low	0xFA10	0x00
TCMPAH0	Timer0 Compare Match A High	0xFA11	0x00
TCMPAL1	Timer1 Compare Match A Low	0xFA12	0x00
TCMPAH1	Timer1 Compare Match A High	0xFA13	0x00
TCMPBL0	Timer0 Compare Match B Low	0xFA14	0x00
TCMPBH0	Timer0 Compare Match B High	0xFA15	0x00
TCMPBL1	Timer1 Compare Match B Low	0xFA16	0x00
TCMPBH1	Timer1 Compare Match B High	0xFA17	0x00
TCNTL0	Timer0 Counter Low	0xFA18	0x00
TCNTH0	Timer0 Counter High	0xFA19	0x00
TCNTL1	Timer1 Counter Low	0xFA1A	0x00
TCNTH1	Timer1 Counter High	0xFA1B	0x00
TBUFAL0	Timer0 Buffer A Low	0xFA20	0x00
TBUFAH0	Timer0 Buffer A High	0xFA21	0x00
TBUFAL1	Timer1 Buffer A Low	0xFA22	0x00
TBUFAH1	Timer1 Buffer A High	0xFA23	0x00
TBUFBL0	Timer0 Buffer B Low	0xFA24	0x00
TBUFBH0	Timer0 Buffer B High	0xFA25	0x00
TBUFBL1	Timer1 Buffer B Low	0xFA26	0x00
TBUFBH1	Timer1 Buffer B High	0xFA27	0x00
TOACR0	Timer0 TIOA Output Control Register	0xFA30	0x00
TOACR1	Timer1 TIOA Output Control Register	0xFA31	0x00
TOBCR0	Timer0 TIOB Output Control Register	0xFA32	0x00
TOBCR1	Timer1 TIOB Output Control Register	0xFA33	0x00
TPCIS0	Timer0 Phase Counting Input Select Register	0xFA34	0x00

Symbol	Name	Address	Initial Value
TPCIS1	Timer1 Phase Counting Input Select Register	0xFA35	0x00
TMOD2	Timer2 Control Mode Register	0xFA40	0x00
TMOD3	Timer3 Control Mode Register	0xFA41	0x00
TMSR2	Timer2 Status Register	0xFA42	0x00
TMSR3	Timer3 Status Register	0xFA43	0x00
TMCR2	Timer2 Control Register	0xFA44	0x00
TMCR3	Timer3 Control register	0xFA45	0x00
TMECR2	Timer2 Event Clear Register	0xFA46	0x00
TMECR3	Timer3 Event Clear Register	0xFA47	0x00
TEMOD2	Timer2 Extend Mode Register	0xFA48	0x00
TEMOD3	Timer3 Extend Mode Register	0xFA49	0x00
TICS2	Timer2 Input Capture Select Register	0xFA4A	0x00
TICS3	Timer3 Input Capture Select Register	0xFA4B	0x00
TXES2	Timer2 External Event Select Register	0xFA4C	0x00
TXES3	Timer3 External Event Select Register	0xFA4D	0x00
TPSNF2	Timer2 Prescaler for Noise Filter Register	0xFA4E	0x00
TPSNF3	Timer3 Prescaler for Noise Filter Register	0xFA4F	0x00
TCMPAL2	Timer2 Compare Match A Low	0xFA50	0x00
TCMPAH2	Timer2 Compare Match A High	0xFA51	0x00
TCMPAL3	Timer3 Compare Match A Low	0xFA52	0x00
TCMPAH3	Timer3 Compare Match A High	0xFA53	0x00
TCMPBL2	Timer2 Compare Match B Low	0xFA54	0x00
TCMPBH2	Timer2 Compare Match B High	0xFA55	0x00
TCMPBL3	Timer3 Compare Match B Low	0xFA56	0x00
TCMPBH3	Timer3 Compare Match B High	0xFA57	0x00
TCNTL2	Timer2 Counter Low	0xFA58	0x00
TCNTH2	Timer2 Counter High	0xFA59	0x00
TCNTL3	Timer3 Counter Low	0xFA5A	0x00
TCNTH3	Timer3 Counter High	0xFA5B	0x00
TBUFAL2	Timer2 Buffer A Low	0xFA60	0x00
TBUFAH2	Timer2 Buffer A High	0xFA61	0x00
TBUFAL3	Timer3 Buffer A Low	0xFA62	0x00
TBUFAH3	Timer3 Buffer A High	0xFA63	0x00
TBUFBL2	Timer2 Buffer B Low	0xFA64	0x00
TBUFBH2	Timer2 Buffer B High	0xFA65	0x00
TBUFBL3	Timer3 Buffer B Low	0xFA66	0x00
TBUFBH3	Timer3 Buffer B High	0xFA67	0x00
TOACR2	Timer2 TIOA Output Control Register	0xFA70	0x00
TOACR3	Timer3 TIOA Output Control Register	0xFA71	0x00
TOBCR2	Timer2 TIOB Output Control Register	0xFA72	0x00
TOBCR3	Timer3 TIOB Output Control Register	0xFA73	0x00
TPCIS2	Timer2 Phase Counting Input Select Register	0xFA74	0x00
TPCIS3	Timer3 Phase Counting Input Select Register	0xFA75	0x00

Table 17-3. List of SFR BUS Registers

Symbol	Name	Address	Initial Value
TCMPAL0	Timer0 Compare Match A Low	0x04	0x00
TCMPAL1	Timer1 Compare Match A Low	0x0C	0x00
TCMPAH0	Timer0 Compare Match A High	0x04	0x00
TCMPAH1	Timer1 Compare Match A High	0x0C	0x00
TCMPBL0	Timer0 Compare Match B Low	0x05	0x00
TCMPBL1	Timer1 Compare Match B Low	0x0D	0x00
TCMPBH0	Timer0 Compare Match B High	0x05	0x00
TCMPBH1	Timer1 Compare Match B High	0x0D	0x00
TBUFAL0	Timer0 Buffer A Low	0x06	0x00
TBUFAL1	Timer1 Buffer A Low	0x0E	0x00
TBUFAH0	Timer0 Buffer A Low	0x06	0x00
TBUFAH1	Timer1 Buffer A Low	0x0E	0x00
TBUFBL0	Timer0 Buffer B Low	0x07	0x00
TBUFBL1	Timer1 Buffer B Low	0x0F	0x00
TBUFBH0	Timer0 Buffer B High	0x07	0x00
TBUFBH1	Timer1 Buffer B High	0x0F	0x00
TCMPAL2	Timer2 Compare Match A Low	0x14	0x00
TCMPAL3	Timer3 Compare Match A Low	0x1C	0x00
TCMPAH2	Timer2 Compare Match A High	0x14	0x00
TCMPAH3	Timer3 Compare Match A High	0x1C	0x00
TCMPBL2	Timer2 Compare Match B Low	0x15	0x00
TCMPBL3	Timer3 Compare Match B Low	0x1D	0x00
TCMPBH2	Timer2 Compare Match B High	0x15	0x00
TCMPBH3	Timer3 Compare Match B High	0x1D	0x00
TBUFAL2	Timer2 Buffer A Low	0x16	0x00
TBUFAL3	Timer3 Buffer A Low	0x1E	0x00
TBUFAH2	Timer2 Buffer A High	0x16	0x00
TBUFAH3	Timer3 Buffer A High	0x1E	0x00
TBUFBL2	Timer2 Buffer B Low	0x17	0x00
TBUFBL3	Timer3 Buffer B Low	0x1F	0x00
TBUFBH2	Timer2 Buffer B High	0x17	0x00
TBUFBH3	Timer3 Buffer B High	0x1F	0x00

17.2.1. TMOD0/2 (Timer0/2 Control Mode Register)

When TMOD1.CASMD = 1, TMR0 and TMR1 operate as a timer of 32 bits × 1 channel. TMR1 corresponds to the higher 16 bits and TMR0 corresponds to the lower 16 bits. Timer operation in this mode is controlled by the register of TMR0.

When TMOD3.CASMD = 1, TMR2 and TMR3 operate as a timer of 32 bits × 1 channel. TMR3 corresponds to the higher 16 bits and TMR2 corresponds to the lower 16 bits. Timer operation in this mode is controlled by the register of TMR2.

Register	TMOD0	Timer0 Control Mode Register	Address	0xFA00
Register	TMOD2	Timer2 Control Mode Register	Address	0xFA40

Bit	Bit Name	R/W	Initial	Description	Remarks
7	TMREN	R/W	0	Timer enable 0: Timer is disabled 1: Timer is enabled	
6	TMRIE	R/W	0	Timer interrupt master enable 0: Timer interrupt master is disabled 1: Timer interrupt master is enabled	
5	CMPAEN	R/W	0	Compare match/input capture A enable 0: Compare match/input capture A is disabled 1: Compare match/input capture A is enabled	
4	CMPBEN	R/W	0	Compare match/input capture B enable 0: Compare match/input capture B is disabled 1: Compare match/input capture B is enabled	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	PRSCL	R/W	0	Prescaler setting 000: 1/1 001: 1/4 010: 1/16 011: 1/64 100: 1/256 101: 1/1024 110: 1/4096 111: 1/16384	
1		R/W	0		
0		R/W	0		

17.2.2. TMOD1/3 (Timer1/3 Control Mode Register)

When TMOD1.CASMD = 1, TMR0 and TMR1 operate according to the setting value of the TMOD0 register. The TMOD1 register's setting does not affect the operation.

When TMOD3.CASMD = 1, TMR2 and TMR3 operate according to the setting value of the TMOD2 register. The TMOD3 register's setting does not affect the operation.

Register	TMOD1	Timer1 Control Mode Register	Address	0xFA01
Register	TMOD3	Timer3 Control Mode Register	Address	0xFA41

Bit	Bit Name	R/W	Initial	Description	Remarks
7	TMREN	R/W	0	Timer enable 0: Timer is disabled 1: Timer is enabled	
6	TMRIE	R/W	0	Timer interrupt master enable 0: Timer interrupt master is disabled 1: Timer interrupt master is enabled	
5	CMPAEN	R/W	0	Compare match A enable 0: Compare match A is disabled 1: Compare match A is enabled	
4	CMPBEN	R/W	0	Compare match B enable 0: Compare match B is disabled 1: Compare match B is enabled	
3	CASMD	R/W	0	Cascade mode enable 0: Cascade mode is disabled 1: Cascade mode is enabled	
2	PRSCL	R/W	0	Prescaler setting 000: 1/1 001: 1/4 010: 1/16 011: 1/64 100: 1/256 101: 1/1024 110: 1/4096 111: 1/16384	
1		R/W	0		
0		R/W	0		

17.2.3. TMSRn (Timer n Status Register) (n = 0 to 3)

When TMOD1.CASMD = 1, the internal status is indicated in both the TMSR0 and TMSR1 registers.
When TMOD3.CASMD = 1, the internal status is indicated in both the TMSR2 and TMSR3 registers.

Register	TMSR0	Timer0 Status Register	Address	0xFA02
Register	TMSR1	Timer1 Status Register	Address	0xFA03
Register	TMSR2	Timer2 Status Register	Address	0xFA42
Register	TMSR3	Timer3 Status Register	Address	0xFA43

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	TICLRF	R/C	0	TIC input interrupt flag Read 0: TIC input event does not occur Read 1: TIC input event occurs Write 0: No change Write 1: The bit is cleared	
3	UDF	R/C	0	Underflow flag Read 0: Underflow does not occur Read 1: Underflow occurs Write 0: No change Write 1: The bit is cleared	
2	OVF	R/C	0	Overflow flag Read 0: Overflow does not occur Read 1: Overflow occurs Write 0: No change Write 1: The bit is cleared	
1	CMBF	R/C	0	Compare match/input capture B flag Read 0: Compare match B/input capture B does not occur Read 1: Compare match B/input capture B occurs Write 0: No change Write 1: The bit is cleared	
0	CMAF	R/C	0	Compare match/input capture A flag Read 0: Compare match A/input capture A does not occur Read 1: Compare match A/input capture A occurs Write 0: No change Write 1: The bit is cleared	

17.2.4. TMCRn (Timer n Control Register) (n = 0 to 3)

When TMOD1.CASMD = 1, TMR0 and TMR1 operate according to the setting value of the TMCR0 register. The TMCR1 register's setting does not affect the operation.

When TMOD3.CASMD = 1, TMR0 and TMR1 operate according to the setting value of the TMCR2 register. The TMCR3 register's setting does not affect the operation.

Register	TMCR0	Timer0 Control Register	Address	0xFA04
Register	TMCR1	Timer1 Control Register	Address	0xFA05
Register	TMCR2	Timer2 Control Register	Address	0xFA44
Register	TMCR3	Timer3 Control Register	Address	0xFA45

Bit	Bit Name	R/W	Initial	Description	Remarks
7	UDFIEN	R/W	0	Underflow interrupt enable 0: Underflow interrupt is disabled 1: Underflow interrupt is enabled The bit is enabled in the phase counting mode.	
6	OVFIEN	R/W	0	Overflow interrupt enable 0: Overflow interrupt is disabled 1: Overflow interrupt is enabled	
5	CMAIEN	R/W	0	Compare match/input capture A interrupt enable 0: Compare match/input capture A interrupt is disabled 1: Compare match/input capture A interrupt is enabled	
4	CMBIEN	R/W	0	Compare match/input capture B interrupt enable 0: Compare match/input capture B interrupt is disabled 1: Compare match/input capture B interrupt is enabled	
3	EOAEN	R/W	0	Compare match/input capture A event generation enable 0: Compare match/input capture A event generation is disabled 1: Compare match/input capture A event generation is enabled	
2	EOBEN	R/W	0	Compare match/input capture B event generation enable 0: Compare match/input capture B event generation is disabled 1: Compare match/input capture B event generation is enabled	
1	ACLEN	R/W	0	Timer automatic clearing enable 0: Timer automatic clearing is disabled 1: Timer automatic clearing is enabled	
0	ACLSEL	R/W	0	Selects timer clearing 0: Compare match/input capture A 1: Compare match/input capture B When the ACLEN bit is set to 1, the counter is cleared at the timing set by the ACLSEL bit.	

17.2.5. TMECRn (Timer n Event Clear Register) (n = 0 to 3)

When TMOD1.CASMD = 1, TMR0 and TMR1 operate according to the setting value of the TMECR0 register. The TMECR1 register's setting does not affect the operation.

When TMOD3.CASMD = 1, TMR2 and TMR3 operate according to the setting value of the TMECR2 register. The TMECR3 register's setting does not affect the operation.

Register	TMECR0	Timer0 Event Clear Register	Address	0xFA06
Register	TMECR1	Timer1 Event Clear Register	Address	0xFA07
Register	TMECR2	Timer2 Event Clear Register	Address	0xFA46
Register	TMECR3	Timer3 Event Clear Register	Address	0xFA47

Bit	Bit Name	R/W	Initial	Description	Remarks
7	TICCLRS	R/W	0	TIC event counter clearing enable 0: Counter clearing is disabled 1: Counter clearing is enabled	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	P3CLRS	R/W	0	PWM3 event counter clearing enable 0: Counter clearing is disabled 1: Counter clearing is enabled	
2	P2CLRS	R/W	0	PWM2 event counter clearing enable 0: Counter clearing is disabled 1: Counter clearing is enabled	
1	P1CLRS	R/W	0	PWM1 event counter clearing enable 0: Counter clearing is disabled 1: Counter clearing is enabled	
0	P0CLRS	R/W	0	PWM0 event counter clearing enable 0: Counter clearing is disabled 1: Counter clearing is enabled	

17.2.6. TEMODn (Timer n Extend Mode Register) (n = 0 to 3)

When TMOD1.CASMD = 1, TMR0 and TMR1 operate according to the setting value of the TEMOD0 register. The TEMOD1 register's setting does not affect the operation.

When TMOD3.CASMD = 1, TMR2 and TMR3 operate according to the setting value of the TEMOD2 register. The TEMOD3 register's setting does not affect in the operation.

Register	TEMOD0	Timer0 Extend Mode Register	Address	0xFA08
Register	TEMOD1	Timer1 Extend Mode Register	Address	0xFA09
Register	TEMOD2	Timer2 Extend Mode Register	Address	0xFA48
Register	TEMOD3	Timer3 Extend Mode Register	Address	0xFA49

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	BUFMD	R/W	0	Buffer mode enable 0: Buffer mode is disabled 1: Buffer mode is enabled	
4	FILEN	R/W	0	Input filter enable 0: Input filter is disabled (synchronization of 2-stage flip-flop (F/F)) 1: Input filter is enabled (importing by matching of 3 sampling data)	
3	TICIE	R/W	0	TIC input interrupt enable 0: TIC input interrupt is disabled 1: TIC input interrupt is enabled	
2	EMOD	R/W	0	Timer extension mode setting 000: Normal mode 001: Phase counting mode 1 010: Phase counting mode 2 011: Phase counting mode 3 100: Phase counting mode 4 Other than above: Setting prohibited	
1		R/W	0		
0		R/W	0		

17.2.7. TICS0 (Timer0 Input Capture Select Register)

Register	TICS0	Timer0 Input Capture Select Register	Address	0xFA0A

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMPBCS	R/W	0	TCMPB input capture setting 0000: Compare match register 0001: Reserved 0010: Reserved 0011: Compare match A of TMR1* 0100: Compare match B of TMR1* 0101: TIOB input event 0110: Reserved 0111: Reserved 1000: Comparator0 event 1001: Comparator1 event 1010: Comparator2 event 1011: Comparator3 event 1100: Comparator4 event 1101: Comparator5 event 1110: Reserved 1111: Reserved	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3	CMPACS	R/W	0	TCMPA input capture setting 0000: Compare match register 0001: Reserved 0010: Reserved 0011: Compare match A of TMR1* 0100: Compare match B of TMR1* 0101: TIOA input event 0110: Reserved 0111: Reserved 1000: Comparator0 event 1001: Comparator1 event 1010: Comparator2 event 1011: Comparator3 event 1100: Comparator4 event 1101: Comparator5 event 1110: Reserved 1111: Reserved	
2		R/W	0		
1		R/W	0		
0		R/W	0		

* Do not select in the cascade mode (TMODn.CASMD = 1).

17.2.8. TICS1 (Timer1 Input Capture Select Register)

When TMOD1.CASMD = 1, TMR0 and TMR1 operate according to the setting value of the TICS0 register. The TICS1 register's setting does not affect the operation.

Register	TICS1		Timer1 Input Capture Select Register	Address	0xFA0B
Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMPBCS	R/W	0	TCMPB input capture setting 0000: Compare match register 0001: Compare match A of TMR0 0010: Compare match B of TMR0 0011: Reserved 0100: Reserved 0101: TIOB input event 0110: Reserved 0111: Reserved 1000: Comparator0 event 1001: Comparator1 event 1010: Comparator2 event 1011: Comparator3 event 1100: Comparator4 event 1101: Comparator5 event 1110: Reserved 1111: Reserved	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3	CMPACS	R/W	0	TCMPA input capture setting 0000: Compare match register 0001: Compare match A of TMR0 0010: Compare match B of TMR0 0011: Reserved 0100: Reserved 0101: TIOA input event 0110: Reserved 0111: Reserved 1000: Comparator0 event 1001: Comparator1 event 1010: Comparator2 event 1011: Comparator3 event 1100: Comparator4 event 1101: Comparator5 event 1110: Reserved 1111: Reserved	
2		R/W	0		
1		R/W	0		
0		R/W	0		

17.2.9. TICSn (Timer n Input Capture Select Register) (n = 2 to 3)

When TMOD3.CASMD = 1, TMR2 and TMR3 operate according to the setting value of the TICS2 register. The TICS3 register's setting does not affect the operation.

Register	TICS2	Timer2 Input Capture Select Register	Address	0xFA4A
Register	TICS3	Timer3 Input Capture Select Register	Address	0xFA4B

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMPBCS	R/W	0	TCMPB input capture setting 0000: Compare match register 0001: TMR0 compare match A 0010: TMR0 compare match B 0011: TMR1 compare match A 0100: TMR1 compare match B 0101: TIOB input event 0110: Reserved 0111: Reserved 1000: Comparator0 event 1001: Comparator1 event 1010: Comparator2 event 1011: Comparator3 event 1100: Comparator4 event 1101: Comparator5 event 1110: Reserved 1111: Reserved	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3	CMPACS	R/W	0	TCMPA input capture setting 0000: Compare match register 0001: TMR0 compare match A 0010: TMR0 compare match B 0011: TMR1 compare match A 0100: TMR1 compare match B 0101: TIOA input event 0110: Reserved 0111: Reserved 1000: Comparator0 event 1001: Comparator1 event 1010: Comparator2 event 1011: Comparator3 event 1100: Comparator4 event 1101: Comparator5 event 1110: Reserved 1111: Reserved	
2		R/W	0		
1		R/W	0		
0		R/W	0		

17.2.10. TXESn (Timer n External Event Select Register) (n = 0 to 3)

When TMOD1.CASMD = 1, TMR0 and TMR1 operate according to the setting value of the TXES0 register. The setting of the TXES1 register does not affect the operation.

When TMOD3.CASMD = 1, TMR2 and TMR3 operate according to the setting value of the TXES2 register. The TXES3 register's setting does not affect the operation.

Register	TXES0	Timer0 External Event Select Register	Address	0xFA0C
Register	TXES1	Timer1 External Event Select Register	Address	0xFA0D
Register	TXES2	Timer2 External Event Select Register	Address	0xFA4C
Register	TXES3	Timer3 External Event Select Register	Address	0xFA4D

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	TICEVS	R/W	0	TIC input event setting 00: No event is selected 01: Falling edge 10: Rising edge 11: Both edge	
4		R/W	0		
3	TIBEVS	R/W	0	TIOB input event setting 00: No event is selected 01: Falling edge 10: Rising edge 11: Both edge	
2		R/W	0		
1	TIAEVS	R/W	0	TIOA input event setting 00: No event is selected 01: Falling edge 10: Rising edge 11: Both edge	
0		R/W	0		

17.2.11. TPSNFn (Timer n Prescaler for Noise Filter Register) (n = 0 to 3)

When TMOD1.CASMD = 1, TMR0 and TMR1 operate according to the setting value of the TPSNF0 register. The setting of the TPSNF1 register does not affect in the operation.

When TMOD3.CASMD = 1, TMR2 and TMR3 operate according to the setting value of the TPSNF2 register. The TPSNF3 register's setting does not affect in the operation.

Register	TPSNF0	Timer0 Prescaler for Noise Filter Register	Address	0xFA0E
Register	TPSNF1	Timer1 Prescaler for Noise Filter Register	Address	0xFA0F
Register	TPSNF2	Timer2 Prescaler for Noise Filter Register	Address	0xFA4E
Register	TPSNF3	Timer3 Prescaler for Noise Filter Register	Address	0xFA4F

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	NFPSCLC	R/W	0	Prescaler setting of noise filter 000: 1/1 001: 1/16 010: 1/128 011: 1/1024 100: 1/8192 101: 1/65536 110: 1/262144 111: 1/1048576 The sampling frequency of the input noise filter is the product of the CLKFAST frequency multiplied by the above setting value. For example, if CLKFAST = 60 MHz and NFPSCLC = 0b111, the sampling cycle is approximately 17.5 ms.	
1		R/W	0		
0		R/W	0		

17.2.12. TCMPALn (Timer n Compare Match A Low) (n = 0 to 3)

Register	TCMPAL0	Timer0 Compare Match A Low	Address	0xFA10	0x04
Register	TCMPAL1	Timer1 Compare Match A Low	Address	0xFA12	0x0C
Register	TCMPAL2	Timer2 Compare Match A Low	Address	0xFA50	0x14
Register	TCMPAL3	Timer3 Compare Match A Low	Address	0xFA52	0x1C

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMPAL	R/W	0	Lower 8 bits of compare match A The CPU can read from/write to these bits using the MOVX instruction only.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

17.2.13. TCMPAHn (Timer n Compare Match A High) (n = 0 to 3)

Register	TCMPAH0	Timer0 Compare Match A High	Address	0xFA11	0x04
Register	TCMPAH1	Timer1 Compare Match A High	Address	0xFA13	0x0C
Register	TCMPAH2	Timer2 Compare Match A High	Address	0xFA51	0x14
Register	TCMPAH3	Timer3 Compare Match A High	Address	0xFA53	0x1C

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMPAH	R/W	0	Higher 8 bits of compare match A The CPU can read from/write to these bits using the MOVX instruction only.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

17.2.14. TCMPBLn (Timer n Compare Match B Low) (n = 0 to 3)

Register	TCMPBL0	Timer0 Compare Match B Low	Address	0xFA14	0x05
Register	TCMPBL1	Timer1 Compare Match B Low	Address	0xFA16	0x0D
Register	TCMPBL2	Timer2 Compare Match B Low	Address	0xFA54	0x15
Register	TCMPBL3	Timer3 Compare Match B Low	Address	0xFA56	0x1D

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMPBL	R/W	0	Lower 8 bits of compare match B The CPU can read from/write to these bits using the MOVX instruction only.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

17.2.15. TCMPBHn (Timer n Compare Match B High) (n = 0 to 3)

Register	TCMPBH0	Timer0 Compare Match B High	Address	0xFA15	0x05
Register	TCMPBH1	Timer1 Compare Match B High	Address	0xFA17	0x0D
Register	TCMPBH2	Timer2 Compare Match B High	Address	0xFA55	0x15
Register	TCMPBH3	Timer3 Compare Match B High	Address	0xFA57	0x1D

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMPBH	R/W	0	Higher 8 bits of compare match B The CPU can read from/write to these bits using the MOVX instruction only.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

17.2.16. TCNTLn (Timer n Counter Low) (n = 0 to 3)

Register	TCNTL0	Timer0 Counter Low	Address	0xFA18
Register	TCNTL1	Timer1 Counter Low	Address	0xFA1A
Register	TCNTL2	Timer2 Counter Low	Address	0xFA58
Register	TCNTL3	Timer3 Counter Low	Address	0xFA5A

Bit	Bit Name	R/W	Initial	Description	Remarks
7	TCNTL	R/W	0	Lower 8 bits of the counter value	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

17.2.17. TCNTHn (Timer n Counter High) (n = 0 to 3)

Register	TCNTH0	Timer0 Counter High	Address	0xFA19
Register	TCNTH1	Timer1 Counter High	Address	0xFA1B
Register	TCNTH2	Timer2 Counter High	Address	0xFA59
Register	TCNTH3	Timer3 Counter High	Address	0xFA5B

Bit	Bit Name	R/W	Initial	Description	Remarks
7	TCNTH	R/W	0	Higher 8 bits of the counter value	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

17.2.18. TBUFALn (Timer n Buffer A Low) (n = 0 to 3)

Register	TBUFAL0	Timer0 Buffer A Low	Address	0xFA20	0x06
Register	TBUFAL1	Timer1 Buffer A Low	Address	0xFA21	0x0E
Register	TBUFAL2	Timer2 Buffer A Low	Address	0xFA60	0x16
Register	TBUFAL3	Timer3 Buffer A Low	Address	0xFA61	0x1E

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BUFAL	R/W	0	Lower 8 bits of buffer A The CPU can read from/write to these bits using the MOVX instruction only.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

17.2.19. TBUFAHn (Timer n Buffer A High) (n = 0 to 3)

Register	TBUFAH0	Timer0 Buffer A High	Address	0xFA22	0x06
Register	TBUFAH1	Timer1 Buffer A High	Address	0xFA23	0x0E
Register	TBUFAH2	Timer2 Buffer A High	Address	0xFA62	0x16
Register	TBUFAH3	Timer3 Buffer A High	Address	0xFA63	0x1E

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BUFAH	R/W	0	Higher 8 bits of buffer A The CPU can read from/write to these bits using the MOVX instruction only.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

17.2.20. TBUFBLn (Timer n Buffer B Low) (n = 0 to 3)

Register	TBUFBL0	Timer0 Buffer B Low	Address	0xFA24	0x07
Register	TBUFBL1	Timer1 Buffer B Low	Address	0xFA25	0x0F
Register	TBUFBL2	Timer2 Buffer B Low	Address	0xFA64	0x17
Register	TBUFBL3	Timer3 Buffer B Low	Address	0xFA65	0x1F

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BUFBL	R/W	0	Lower 8 bits of buffer B The CPU can read from/write to these bits using the MOVX instruction only.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

17.2.21. TBUFBHn (Timer n Buffer B High) (n = 0 to 3)

Register	TBUFBH0	Timer0 Buffer B High	Address	0xFA26	0x07
Register	TBUFBH1	Timer1 Buffer B High	Address	0xFA27	0x0F
Register	TBUFBH2	Timer2 Buffer B High	Address	0xFA66	0x17
Register	TBUFBH3	Timer3 Buffer B High	Address	0xFA67	0x1F

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BUFBH	R/W	0	Higher 8 bits of buffer B The CPU can read from/write to these bits using the MOVX instruction only.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

17.2.22. TOACRn (Timer n TIOA Output Control Register) (n = 0 to 3)

When TMOD1.CASMD = 1, TMR0 and TMR1 operate according to the setting value of the TOACR0 register. The setting value of the TOACR1 register does not affect. In addition, TMR2 and TMR3 operate according to the setting value of the TOACR2 register. The setting value of the TOACR3 register does not affect.

Register	TOACR0	Timer0 TIOA Output Control Register	Address	0xFA30
Register	TOACR1	Timer1 TIOA Output Control Register	Address	0xFA31
Register	TOACR2	Timer2 TIOA Output Control Register	Address	0xFA70
Register	TOACR3	Timer3 TIOA Output Control Register	Address	0xFA71

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	TOCLR	R/W	0	TIOA output level setting when TCNT is cleared or overflows 00: No change 01: The output level is set to low 10: The output level is set to high 11: Toggle	
4		R/W	0		
3	TOCMPA	R/W	0	TIOA output level setting at TCMPA matching 00: No change 01: The output level is set to low 10: The output level is set to high 11: Toggle	
2		R/W	0		
1	TOINI*	W	0	TIOA initial output level setting 00: No change 01: The output level is set to low 10: The output level is set to high 11: Setting prohibited The read value is always 0. The setting of the bit has the highest priority over the pin settings of any other events.	
0		W	0		

* When TMR0 and TMR1 are in the cascade mode, the initial output levels of TIOA0 and TIOA1 are determined by the setting of the TOACR0.TOINI bit. In addition, when TMR2 and TMR3 are in the cascade mode, the initial output levels of TIOA2 and TIOA3 are determined by the setting of the TOACR2.TOINI bit.

17.2.23. TOBCRn (Timer n TIOB Output Control Register) (n = 0 to 3)

When TMOD1.CASMD = 1, TMR0 and TMR1 operate according to the setting value of the TOBCR0 register. The setting value of the TOBCR1 register does not affect. In addition, TMR2 and TMR3 operate according to the setting value of the TOBCR2 register. The setting value of the TOBCR3 register does not affect.

Register	TOBCR0	Timer0 TIOB Output Control Register	Address	0xFA32
Register	TOBCR1	Timer1 TIOB Output Control Register	Address	0xFA33
Register	TOBCR2	Timer2 TIOB Output Control Register	Address	0xFA72
Register	TOBCR3	Timer3 TIOB Output Control Register	Address	0xFA73

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	TOCLR	R/W	0	TIOB output level setting when TCNT is cleared or overflows 00: No change 01: The output level is set to low 10: The output level is set to high 11: Toggle	
4		R/W	0		
3	TOCMPB	R/W	0	TIOB output level setting at TCMPB matching 00: No change 01: The output level is set to low 10: The output level is set to high 11: Toggle	
2		R/W	0		
1	TOINI*	W	0	TIOB initial output level setting 00: No change 01: The output level is set to low 10: The output level is set to high 11: Setting prohibited The read value is always 0. The writing by change of the TIOB output level has a higher priority than any other compare match that the output level is changed.	
0		W	0		

* When TMR0 and TMR1 are in the cascade mode, the initial output levels of TIOB0 and TIOB1 are determined by the setting of the TOBCR0.TOINI bit. In addition, when TMR2 and TMR3 are in the cascade mode, the initial output levels of TIOB2 and TIOB3 are determined by the setting of the TOBCR2.TOINI bit.

17.2.24. TPCISn (Timer n Phase Counting Input Select Register) (n = 0 to 3)

When TMOD1.CASMD = 1, TMR0 and TMR1 operate according to the setting value of the TPCIS0 register. The setting value of the TPCIS1 register is ignored. In addition, TMR2 and TMR3 operate according to the setting value of the TPCIS2 register. The setting value of the TPCIS3 register is ignored.

Register	TPCIS0	Timer0 Phase Counting Input Select Register	Address	0xFA34
Register	TPCIS1	Timer1 Phase Counting Input Select Register	Address	0xFA35
Register	TPCIS2	Timer2 Phase Counting Input Select Register	Address	0xFA74
Register	TPCIS3	Timer3 Phase Counting Input Select Register	Address	0xFA75

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	TIBSEL	R/W	0	TIOB input selection 0: TIOB input pin 1: Comparator output The bit is referred to when the phase counting mode is used (see Table 17-4).	
0	TIASEL	R/W	0	TIOA input selection 0: TIOA input pin 1: Comparator output The bit is referred to when the phase counting mode is used (see Table 17-4).	

Table 17-4. Correspondence between Channel and Comparator Output in Phase Counting Mode

TMR Channel	TOIA Input (TIASEL = 1)	TIOB Input (TIBSEL = 1)
TMR0	Comparator4 output	Comparator5 output
TMR1	—	—
TMR2	Comparator0 output	Comparator1 output
TMR3	Comparator2 output	Comparator3 output

17.3. Operation

17.3.1. 16-bit Register Access

When reading from or writing to the TCNTxn, TCMPxxn, or TBUFxxn register, read from or write to the lower 8-bit register and the corresponding higher 8-bit register continuously.

● **TCNTxn Register**

- Writing:
  The data written to the TCNTLn register (lower 8 bits) is temporarily buffered. The buffered lower 8 bits is simultaneously written to the counter with writing to the TCNTHn register (higher 8 bits).
- Reading:
  When reading the TCNTLn register (lower 8 bits), the value of the TCNTHn register (higher 8 bits) is buffered. When reading the TCNTHn register, the buffered data is read.

● **TCMPxxn and TBUFxxn Registers**

The TCMPxxn and TBUFxxn registers are mapped to both the XDATA BUS and the SFR BUS areas. The TCMPxxn and TBUFxxn registers are 16-bit width. In the XDATA area, the lower 8-bit and higher 8-bit registers are mapped to continuous addresses. The mapping order is little endian. In the SFR BUS area, the lower 8-bit and higher 8-bit registers are mapped to the same address. These registers can be accessed from the DSAC or the EPU by the 16-bit access method only. The CPU can be read from or written to using the MOVX instruction only. When the CPU accesses to the TCMPxxn and TBUFxxn registers by the 8-bit access method, the following ways are used:

- Writing:
  The data written to the TCMPxLn or the TBUFxLn register (lower 8 bits) is temporarily buffered. The buffered lower 8-bit data is simultaneously written to the TCMPxLn or TBUFxLn register with writing to the TCMPxHn or the TBUFxHn register (higher 8 bits).
- Reading:
  When reading the TCMPxLn or TBUFxLn register (lower 8 bits), the value of the TCMPxHn or TBUFxHn register (higher 8 bits) is buffered. When reading from the TCMPxLn or the TBUFxHn register (higher 8 bits), the buffered data is read.

17.3.2. Counter Operation

When the TMODn.TMREN bit is set to 1, the 16-bit counter (TCNT) starts counting from the values set in the TCNTLn and TCNTHn registers. The initial value of the TCNT is 0x0000.

When the write access to the TCNT and the overflow of the TCNT simultaneously occur, the write access takes priority. The priority of the counter operation is as follows:

Writing to the TCNT by a program > clearing of the TCNT > counting up/down

17.3.3. Compare Match Operation

The compare match generates an event when the value of the 16-bit counter (TCNT) matches the setting value of the TCMPA or the TCMPB. The value of TCMPA is set to the TCMPALn and TCMPAHn registers. The value of TCMPB is set to the TCMPBLn and TCMPBHn registers.

When the TMODn.CMPAEN bit is set to 1, the operation of compare match A is enabled. When the TMODn.CMPBEN bit is set to 1, the operation of compare match B is enabled.

When compare match A is detected, the TMSRn.CMAF bit is set to 1. When compare match B is detected, the TMSRn.CMBF bit is set to 1. An event and an interrupt can be generated at the timing that compare match is detected (For details, see Section 17.3.14). When the writing to the compare match registers and the compare match simultaneously occur, the compare match takes priority.

17.3.4. Compare Match Output

Each channel of the TMR has 2 types of compare match output (TIOAn and TIOBn).

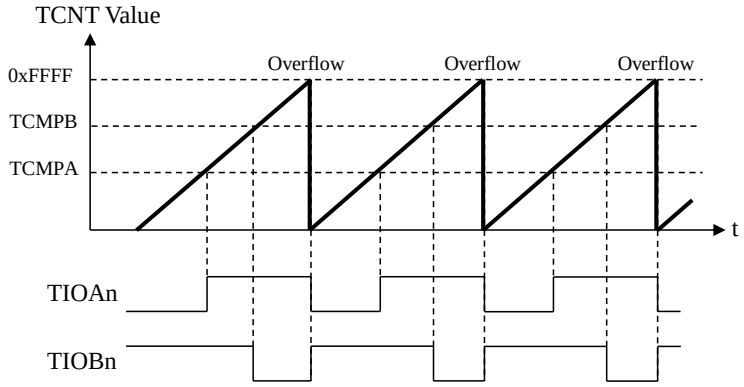
When the TCMPA register (the TCMPALn and TCMPAHn registers) is operated as a register for compare match, the output level of the TIOAn pin can be changed by the following events: The compare match of the TCMPA register, and the overflow or clearing of the 16-bit counter (TCNT). The initial value of the TIOAn pin output is defined by the TOACRn.TOINI bits. The output level of the TIOAn pin after the compare match is defined by the TOACRn.TOCMPA bits. The output level of the TIOAn pin after the overflow or clearing of TCNT is defined by the TOACRn.TOCLR bits.

When the TCMPB register (the TCMPBLn and TCMPBHn registers) is operated as a register for compare match, the output level of the TIOBn pin can be changed by the following events: The compare match of the TCMPB register, and the overflow or clearing of the TCNT. The initial value of the TIOBn pin output is defined by the TOBCRn.TOINI bits. The output level of the TIOBn pin after the compare match is defined by the TOBCRn.TOCMPB bits. The output level of the TIOBn pin after the overflow or clearing of TCNT is defined by the TOBCRn.TOCLR bits.

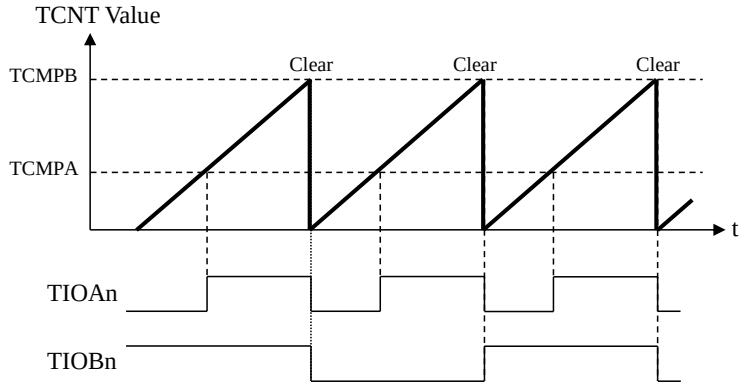
When a clearing event and a writing 0x0000 to the TCNT simultaneously occur, the TCNT is changed to 0x0000 by writing, and the levels of the TIOAn and TIOBn pins are also changed.

The priority of the output level setting is as follows:

- TIOAn: TOACRn.TOINI bits > overflow or clearing of TCNT > compare match A
- TIOBn: TOBCRn.TOINI bits > overflow or clearing of TCNT > compare match B



a) TMCRn.ACLEN = 0b0, TOACRn.TOCMPA = 0b10, TOACRn.TOCLR = 0b01, TOBCRn.TOCMPB = 0b01, TOBCRn.TOCLR = 0b10



b) TMCRn.ACLEN = 0b1, TMCRn.ACLSEL = 0b1, TOACRn.TOCMPA = 0b01, TOACRn.TOCLR = 0b10, TOBCRn.TOCMPB = 0b11, TOBCRn.TOCLR = 0b11

Figure 17-2. Compare Match Output Operation

17.3.5. Automatic Clearing

The 16-bit counter (TCNT) is cleared by generating compare match. When TMCRn.ACLEN bit is set to 1, the

clearing operation of the TCNT by compare match can be enabled. To automatically clear the TCNT by the compare match A or B, set the TMCRn.ACLSEL bit to 0 or 1, respectively. When automatic clearing is disabled, the TCNT counts up to 0xFFFF, and then it is changed to 0x0000. In addition, the TMSRn.OVF bit that indicates overflowing of the TCNT is set to 1.

17.3.6. PWM Event Clearing

To clear the 16-bit counter (TCNT) by the TMR clearing event of PWM0, PWM1, PWM2, and PWM3, set TMECRn.PxCLRS = 1 (x = 0 to 3: channel numbers of the PWM).

17.3.7. TIC Input Event Clearing

To clear The 16-bit counter (TCNT) by the TICn input event, set TMECRn.TICCLRS = 1. The TICn input event is defined by the TXESn.TICEVS bits. Clearing operation by the TICn input event is synchronized with the CLKFAST. This is not synchronized with the timing of counting up of the prescaler and the timing of counting up or down in the phase counting mode.

17.3.8. 32-bit Counter Mode (Cascade Mode)

In the cascade mode, the 2 counters connected in series operate as a 32-bit counter. The cascade mode can be used in the normal mode and the phase counting mode. When the TMOD1.CASMD bit is set to 1, the 16-bit counter (TCNT) of the TMR0 becomes the lower 16-bit counter, and the TCNT of the TMR1 becomes the higher 16-bit counter. When the TMOD3.CASMD bit is set to 1, the TCNT of the TMR2 becomes the lower 16-bit counter, and the TCNT of TMR3 becomes the higher 16-bit counter.

In the cascade mode, the timer operates according to the register settings of small number channels. The register settings of large number channels are ignored. For the cascade mode of the TMR0 and TMR1, the setting of the TMR0 is effective. For the cascade mode of the TMR2 and TMR3, the setting of the TMR2 is effective.

The TMSRn register that indicates the TMR status is reflected in the cascaded 2 channels. The flag must be cleared for each channel.

For the input pin, the pin of the smaller number channel out of the cascaded pair channels is used. The output pins of both the channels are set by the smaller number channel of the cascaded pair channels. The same signal is output from the output pins of both the channels.

An interrupt and an event are set by the smaller number channel of the cascaded pair channels, and output from both the channels.

17.3.9. Compare Match Timing

Figure 17-3 shows the timing of a compare match.

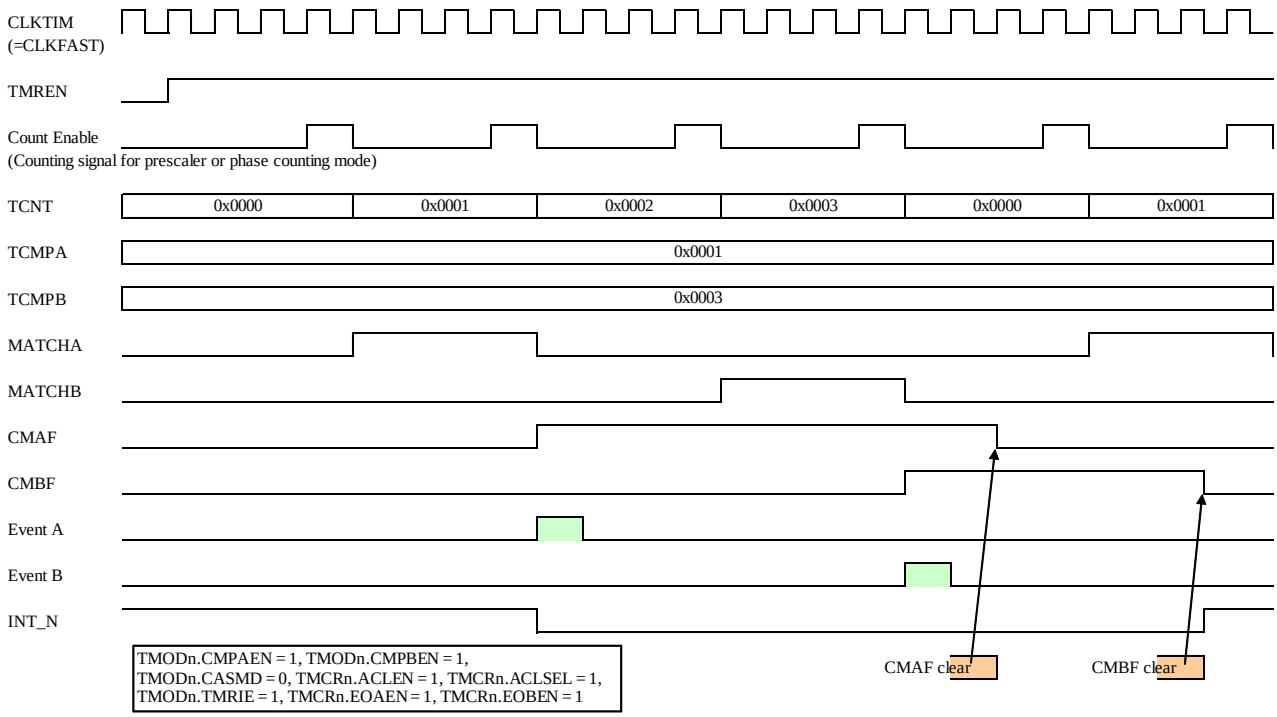


Figure 17-3. Compare Match Timing

When the TMODn.TMREN bit is set to 1, the 16-bit counter (TCNT) is counted up from a preset value. The initial value of the TCNT is 0x0000. The TMODn.PRSCL bits determine the count-up timing of the TCNT or the count-down timing of the TCNT in the phase counting mode.

Figure 17-3 shows an operation when dividing CLKFAST by 4 is selected. This operation is described as follows:

- When the TCNT matches the value of the TCMPA register (the TCMPALn and TCMPAHn registers), the TMSRn.CMAF bit is set at the next update timing of the TCNT.
- When TCNT matches the value of the TCMPB register (the TCMPBLn and TCMPBHn registers), the TMSRn.CMBF bit is set at the next update timing of the TCNT.
- The compare match interrupt (INT_N) occurs at the same time as the TMSRn.CMAF bit or the TMSRn.CMBF bit is set.
- When the TCNT matches the TCMPB register during TMCRn.ACLEN =1 and TMCRn.ACLSEL = 1, the TCNTLn/Hn register is cleared at the next update timing of the TCNT.
- When the TMCRn.EOAEN is set to 1, the Event A is output at the timing that the TMSRn.CMAF bit is set.
- When the TMCRn.EOBEN is set to 1, the Event B is output at the timing that the TMSRn.CMBF bit is set.

17.3.10. Input Capture Mode

In the input capture mode, the 16-bit counter (TCNT) value is imported to the TCMPA register (the TCMPALn and TCMPAHn registers) or TCMPB (the TCMPBLn and TCMPBHn registers) register by the selected event.

The operation mode of the TCMPA register is determined by the TICSn.CMPACS bits. When the TICSn.CMPACS bits are set to other than 0b000, the TCMPA register operates in the input capture mode. When the event selected by the TICSn.CMPACS bits is received, the TCNT value is imported to the TCMPA register. When TICSn.CMPACS = 0b011, the TCMPA register performs an input capture operation by the TIOAn input event. The TIOAn input event is defined by the TXESn.TIAEVS bits.

The operation mode of the TCMPB register is determined by the TICSn.CMPBCS bits. When the TICSn.CMPBCS bits are set to other than 0b000, the TCMPB register operates in the input capture mode. When the event selected by the TICSn.CMPBCS bits is received, the TCNT value is imported to the TCMPB register. When TICSn.CMPBCS = 0b011, the TCMPB register performs an input capture operation by the TIOBn input event. The TIOBn input event is defined by the TXESn.TIBEVS bits.

An interrupt or an event can be output at the input capture operation of the TCMPA/B register. Enabling or disabling of interrupts or events is determined by the TMCRn register.

To clear the TCNT at the input capture operation, set the TMCRn.ACLEN bit = 1. The event to clear TCNT is defined by the TMCRn.ACLSEL bit.

In the input capture function, the value is imported synchronized with the CLKFAST, and is without depending on the prescaler or the phase counting enable. In the same way, data transfers from the TCMPA/B register to the TBUFA/B register (the TBUFALn and TBUFAH registers, or the TBUFBLn and TBUFBHn registers) in the buffer mode is performed in synchronization with the CLKFAST.

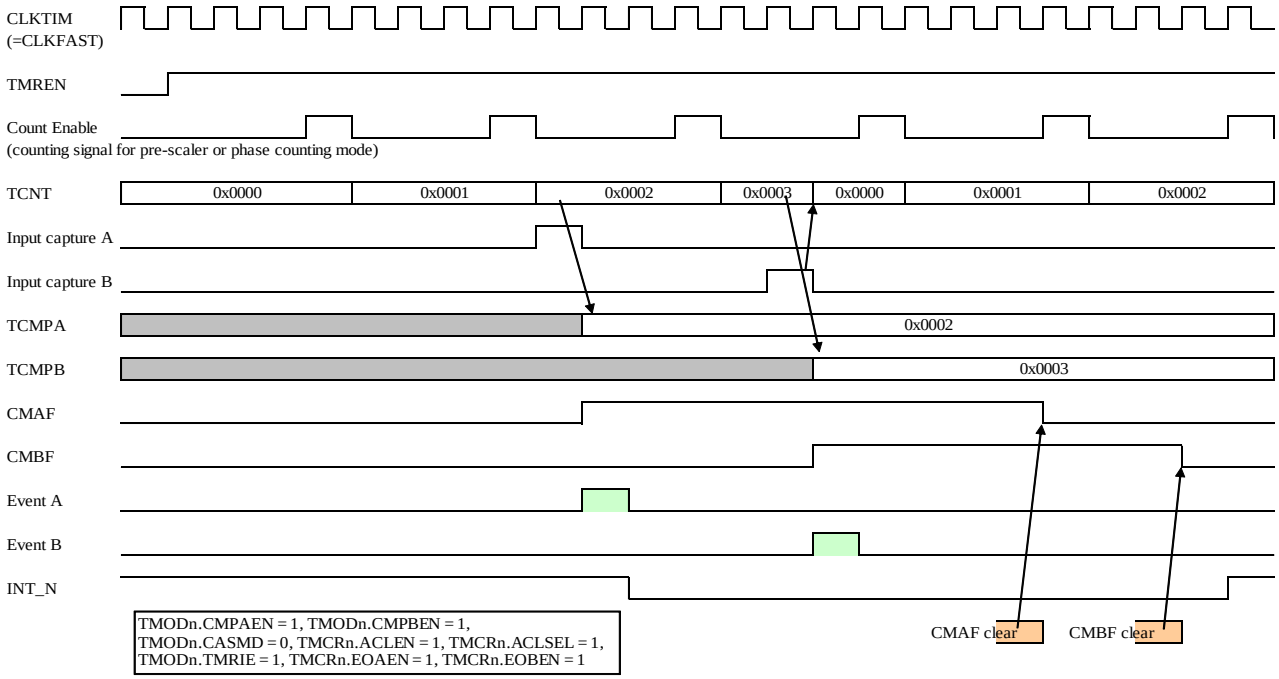


Figure 17-4. Input Capture Timing

17.3.11. Buffer Mode

The TBUFALn/Hn and TBUFBLn/Hn registers operate as a buffer for the TCMPALn/Hn and TCMPBLn/Hn registers, respectively. To operate the buffer mode, set TEMODn.BUFMD = 1. The operation in the buffer mode is as follows:

- When TCMPALn/Hn is the compare match register:
  When TCNTLn/Hn is cleared or overflows, the value of the TBUFAL/Hn register is transferred to the TCMPALn/Hn register.
- When TCMPALn/Hn is the input capture register:
  When an input capture event of TCMPALn/Hn is generated, the value of the TCMPAL/Hn register is transferred to the TBUFALn/Hn register.
- When TCMPBLn/Hn is the compare match register:
  When TCNTLn/Hn is cleared or overflows, the value of the TBUFBL/Hn register is transferred to the TCMPBLn/Hn register.
- When TCMPBLn/Hn is the input capture register:
  When an input capture event of TCMPBLn/Hn is generated, the value of the TCMPBLn/Hn register is transferred to the TBUFBLn/Hn register.

17.3.12. Phase Counting Mode

The phase counting mode is a mode to count up/down the 16 bit counter (TCNT) according to the input states of the TIOAn and TIOBn pins. This mode can be selected from the following 4 modes.

- Phase counting mode 1: TEMODn.EMOD = 0b001
- Phase counting mode 2: TEMODn.EMOD = 0b010
- Phase counting mode 3: TEMODn.EMOD = 0b011
- Phase counting mode 4: TEMODn.EMOD = 0b100

The output signal from the comparator can be used in place of the signal of the TIOAn and TIOBn input pins. To use the output of the comparator in place of the signal of the TIOAn input pin, set TPCISn.TIASEL = 1. To use the output of the comparator in place of the signal of the TIOBn input pin, set TPCISn.TIBSEL = 1. For the comparator output corresponding to the TMR channels, see Table 17-4.

Figure 17-5 to Figure 17-8 show the operations in phase counting mode 1 to 4, respectively. Table 17-5 to Table 17-8 shows the conditions for counting-up/down in these modes, respectively.

● Phase Counting Mode 1

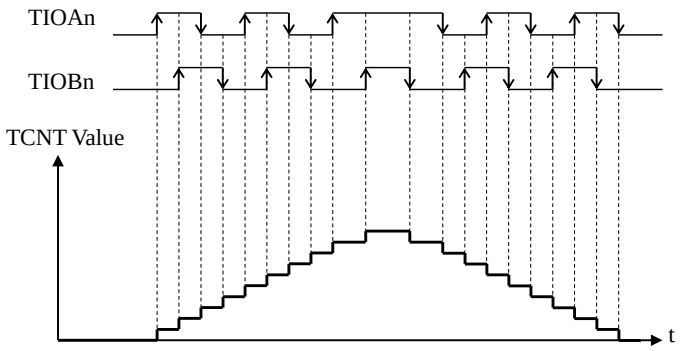


Figure 17-5. Operation in Phase Counting Mode 1

Table 17-5. Condition for Counting-up/down in Phase Counting Mode 1

TIOAn	TIOBn	Counter Operation
High level	Rising edge	+1
Low level	Falling edge	
Rising edge	Low level	
Falling edge	High level	
High level	Falling edge	−1
Low level	Rising edge	
Rising edge	High level	
Falling edge	Low level	

● Phase Counting Mode 2

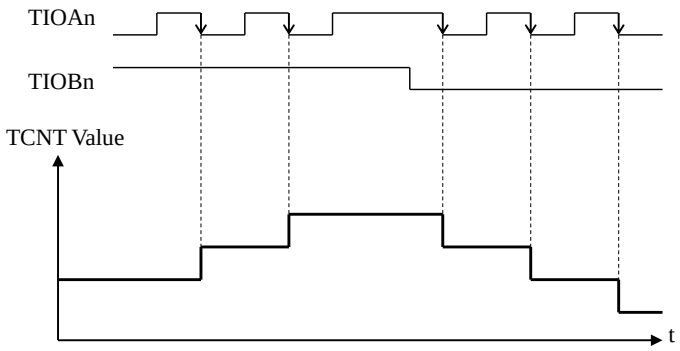


Figure 17-6. Operation of Phase Counting Mode 2

Table 17-6. Condition for Counting-up/down in Phase Counting Mode 2

TIOAn	TIOBn	Counter Operation
High level	Rising edge	No operation
Low level	Falling edge	
Rising edge	Low level	
Falling edge	High level	+1
High level	Falling edge	No operation
Low level	Rising edge	
Rising edge	High level	
Falling edge	Low level	-1

● Phase Counting Mode 3

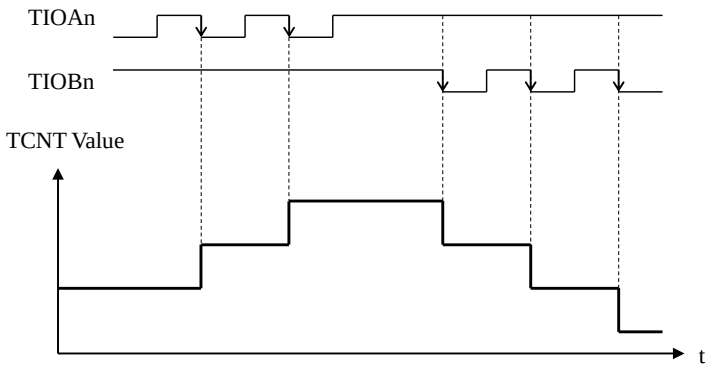


Figure 17-7. Operation in Phase Counting Mode 3

Table 17-7. Condition for Counting-up/down in Phase Counting Mode 3

TIOAn	TIOBn	Counter Operation
High level	Rising edge	No operation
Low level	Falling edge	
Rising edge	Low level	
Falling edge	High level	+1
High level	Falling edge	-1
Low level	Rising edge	No operation
Rising edge	High level	
Falling edge	Low level	

● Phase Counting Mode 4

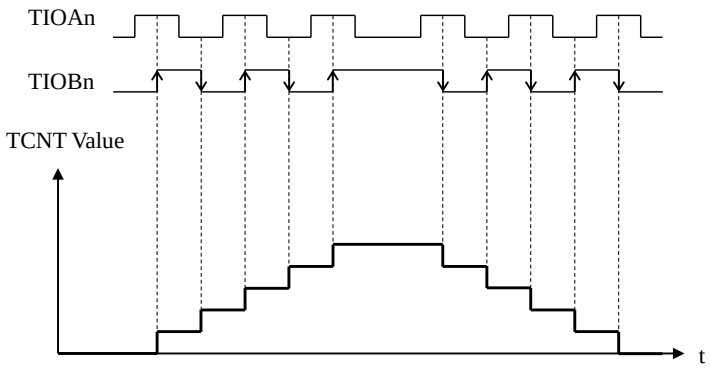


Figure 17-8. Operation in Phase Counting Mode 4

Table 17-8. Condition for Counting-up/down in Phase Counting Mode 4

TIOAn	TIOBn	Counter Operation
High level	Rising edge	+1
Low level	Falling edge	
Rising edge	Low level	No operation
Falling edge	High level	
High level	Falling edge	-1
Low level	Rising edge	
Rising edge	High level	No operation
Falling edge	Low level	

17.3.13. Noise Filter of Input Pins

Each input pin has a noise filter. To enable the noise filter, set TEMODn.FILEN = 1. The noise filter samples the input signal, and holds recent 3 values. The sampling frequency is determined by the TPSNFn register (see Section 17.2.11). When 3 samples are the same value, the input level is imported to the internal TMR logic. When the noise filter is disabled (TEMODn.FILEN = 0), the input signal is synchronized with the CLKFAST by 2 flip-flop circuits.

17.3.14. Events and Interrupts

The TMRn outputs one interrupt to the CPU and 2 events to the peripheral modules for each channel. When the TMODn.TMRIE bit is set to 1, an interrupt request is output to the CPU at the selected interrupt source generation. There are 5 types of interrupt sources as follows:

- When the TMCRn.CMAIEN bit is set to 1, an interrupt output is allowed at the compare match of the TCMPA or the input capture generation.
- When the TMCRn.CMBIEN bit is set to 1, an interrupt output is allowed at the compare match of the TCMPB or the input capture generation.
- When the TMCRn.OVFIEN bit is set to 1, an interrupt output is allowed at overflow of the TCNT.
- When the TMCRn.UDFIEN bit is set to 1, an interrupt output is allowed at underflow of the TCNT. This occurs only in the phase counting mode.
- When the TEMODn.TICIE bit is set to 1, an interrupt output is allowed at the TIC input event detection.

The TMSRn register is a status register indicating generation of each interrupt source. To clear this status, write 1 to the bit to be cleared.

Table 17-9 shows the conditions to use event outputs or interrupt outputs. The Event A and Event B are independent of each other; so, if the generation condition of the 2 events is simultaneously satisfied, these events are simultaneously output.

Table 17-9. Condition to Output Event or Interrupt

Condition	Event A	Event B	Interrupt
TCMPA Compare Match	Usable	—	Usable
TCMPAB Compare Match	—	Usable	Usable
TCMPA Input Capture	Usable	—	Usable
TCMPAB Input Capture	—	Usable	Usable
TCNT Overflow	—	—	Usable
TCNT Underflow	—	—	Usable
TCNT Clearing by Inputting TICn	—	—	Usable

17.3.15. Basic Setting

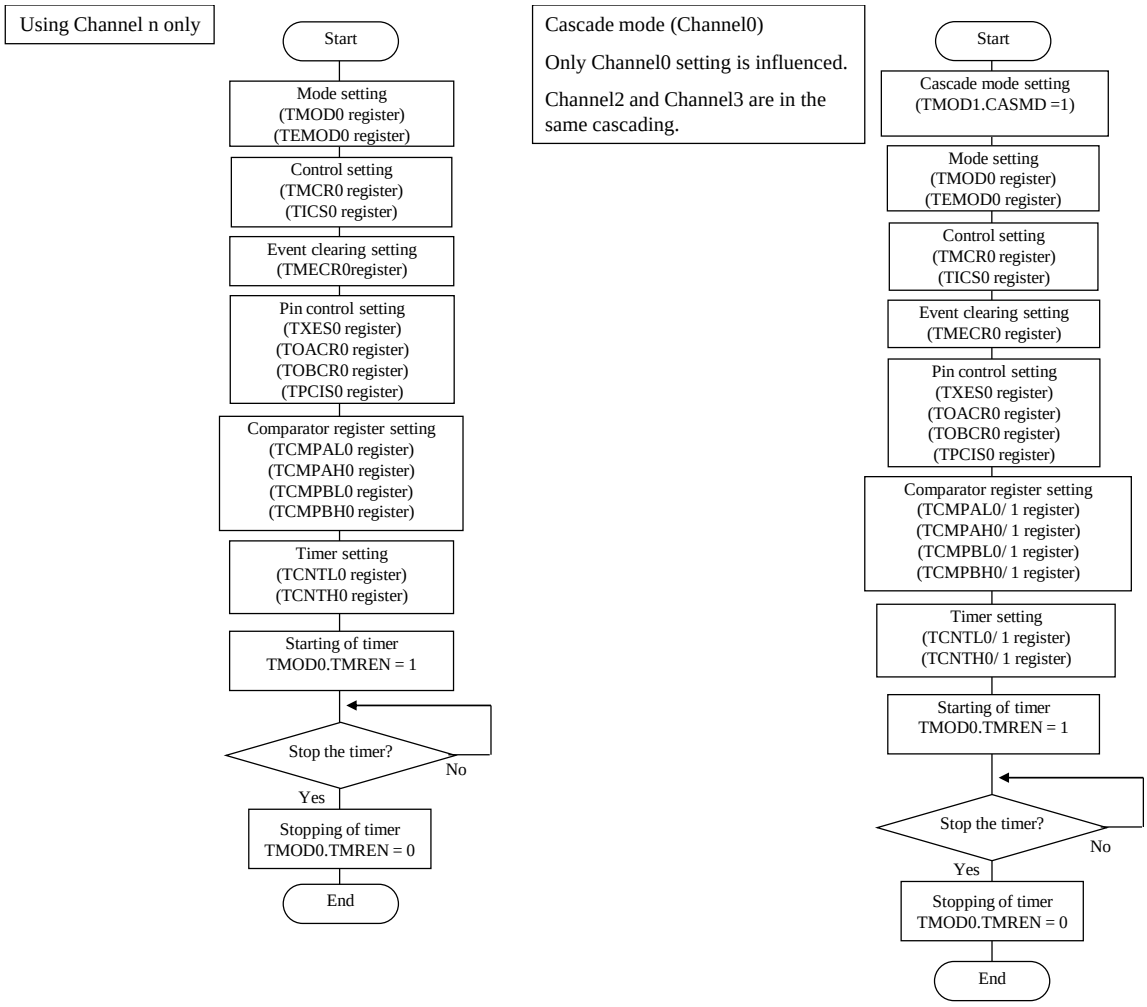


Figure 17-9. Flowchart

18. Serial Peripheral Interface (SPI)

18.1. Overview

The serial peripheral interface (SPI) operates in clock-synchronous and full-duplex serial communications. An external processor and a peripheral system can communicate via the SPI.

Table 18-1. SPI Functional Descriptions

Item	Description
TX and RX Functions	● Serial communications: Master mode or subordinate mode ● Signal of SPI serial communications: MOSI (master out subordinate in), MISO (master in subordinate out), SCK (SPI clock) ● Select (SS_N) signal: When the LSI is a subordinate device, the SS_N signal can be processed by a hardware. When the LSI is a master device, the SS_N signal is generated by the CPU. ● Clocks for the SPI data transmission/reception is enabled for SPI clock edges (both rising edge and falling edge). ● Selectable clock pin level in SPI idle state. ● TXFIFO and RXFIFO have 2 stages, respectively. ● Transmission data buffer: 16 bits × 2 lines ● Reception data buffer: 16 bits × 2 lines
Data Format	● Order of data transfer bit: MSB first or LSB first ● Transfer data length: 6 bits to 16 bits
SPI Clock	● SPI clock frequency range: f/4 to f/1024
Error Detection	● FIFO overrun error
Interrupt Source	● Independent interrupt for transmission or reception ● SPI reception interrupts: Reception data exists on buffer. FIFO errors (buffer underflow and buffer overflow) ● SPI transmission interrupts: Transmission buffer is not full. Transmission completion FIFO errors (buffer underflow and buffer overflow)
Others	● When the SPI is set to a master device, and is disabled to transmit, transmission data is not output (high-Z). ● When the SPI is set to a subordinate device, the SS_N pin is used for selecting SPI. Transmission data is output only during the SPI selection. In the other status, the SS_N pin is high-Z.

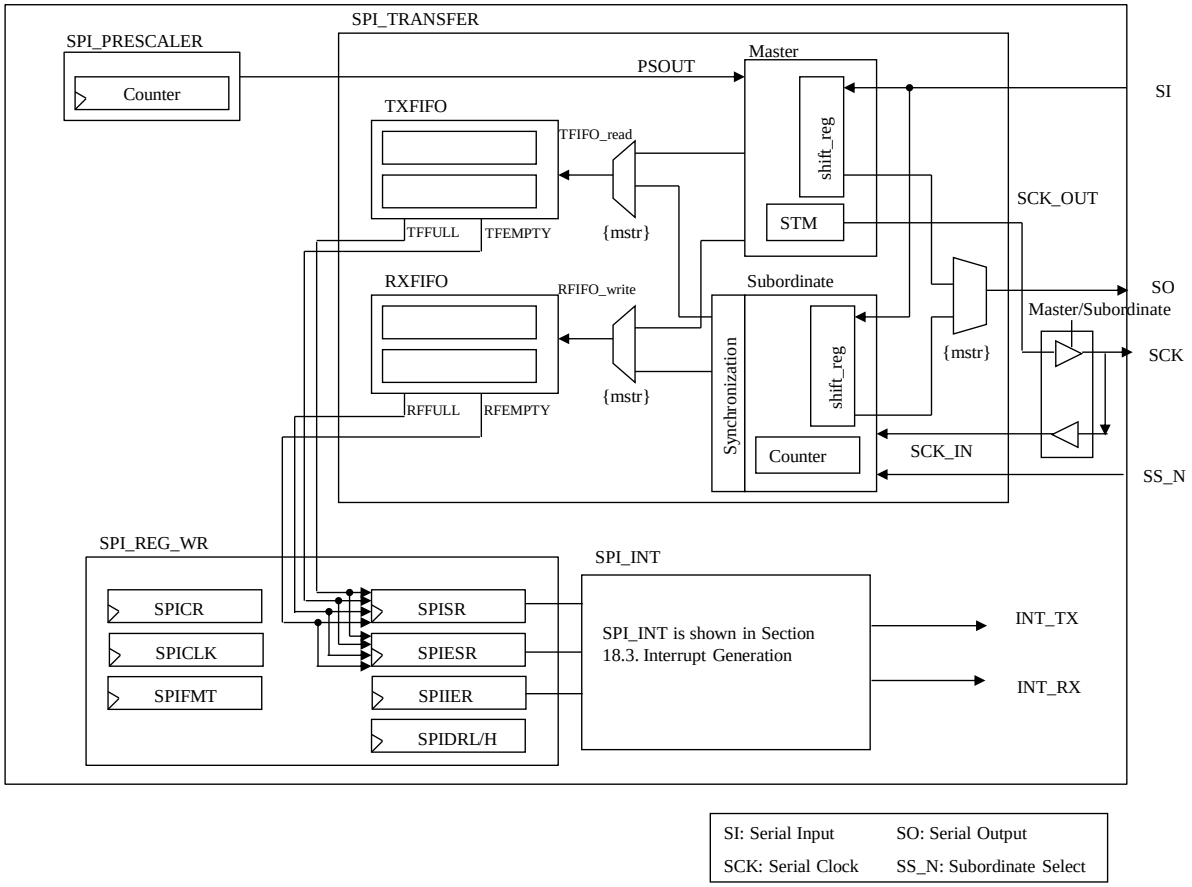


Figure 18-1. SPI Block Diagram

18.2. Register Descriptions

Table 18-2. List of Registers

Symbol	Name	Address	Initial Value
SPICR	SPI Control Register	0xFB80	0x00
SPICLK	SPI Clock Divider Register	0xFB81	0x00
SPIFMT	SPI Data Format Register	0xFB82	0x00
SPISR	SPI Status Register	0xFB84	0x05
SPIESR	SPI Error Status Register	0xFB85	0x00
SPIIER	SPI Interrupt Enable Register	0xFB86	0x00
SPIDRL	SPI Data Register Low	0xFB88	0x00
SPIDRH	SPI Data Register High	0xFB89	0x00

18.2.1. SPICR (SPI Control Register)

Register	SPICR	SPI Control Register	Address	0xFB80

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	SPE	R/W	0	SPI communication enable 0: SPI communication is disabled 1: SPI communication is enabled Data is transferred only when the bit is set to 1 and the SPI core is enabled.	
4	MSTR	R/W	0	Selection of master mode or salve mode 0: Subordinate mode 1: Master mode When the bit is set to 1, the SPI is a master device. When the bit is cleared, the SPI is a subordinate device.	
3	CPOL	R/W	0	Clock polarity The bit and the CPHA bit determine the transfer mode. For details, see Table 18-3.	
2	CPHA	R/W	0	Clock phase The bit and the CPOL bit determine the transfer mode. For details, see Table 18-3.	
1	TXEN	R/W	0	TX operation enable setting 0: TX operation is enabled. 1: TX operation is disabled.	
0	RXEN	R/W	0	RX operation enable setting 0: RX operation is enabled. 1: RX operation is disabled.	

18.2.1.1. The TXEN and RXEN Bits

In the master and subordinate modes, the TXEN and RXEN bits operate differently as follows:

- **Transmission**

- When SPI is Master Mode
  When TXEN =1 or RXEN =1, the SPI operates. However, in RXEN = 1, it is required to write a dummy transmission data in the TXFIFO to operate the SPI.
- When SPI is Subordinate Mode
  When a transmission starts at TXEN = 1 and TXFIFO = empty, the underflow flag of the TXFIFO is set to 1. If only RXEN bit is set to 1, the underflow flag of the TXFIFO is not set to 1 because the data is not transmitted from the TXFIFO.

- **Reception**

When the RXEN bit is set to 1 in both master and subordinate modes, a reception data is stored in the RXFIFO. When all transmission/reception to the RXFIFO are completed in RXEN = 1, the RXFIFO overflow flag is set to 1.

18.2.1.2. The CPOL and CPHA Bits

The clock polarity and clock phase are defined by the CPOL and CPHA bits, respectively.

Selectable 4 SPI modes have different combination of a data setup and a data sampling timings. Table 18-3 shows the settings of CPOL and CPHA bits for the SPI mode. Figure 18-2 shows the timing of each SPI mode. In the subordinate mode (SPICR.MSTR = 0), the SPI modes 0 and 2 cannot be used (i.e., only the SPI modes 1 and 3 can be used).

Table 18-3. Settings of CPOL and CPHA Bits

CPOL Bit	CPHA Bit	First Edge in One SCK Cycle	Second Edge in One SCK Cycle	SPI Mode
0	0	Input data sampling at rising edge	Data output at falling edge	0
0	1	Data output at rising edge	Input data sampling at falling edge	1
1	0	Input data sampling at falling edge	Data output at rising edge	2
1	1	Data output at falling edge	Input data sampling at rising edge	3

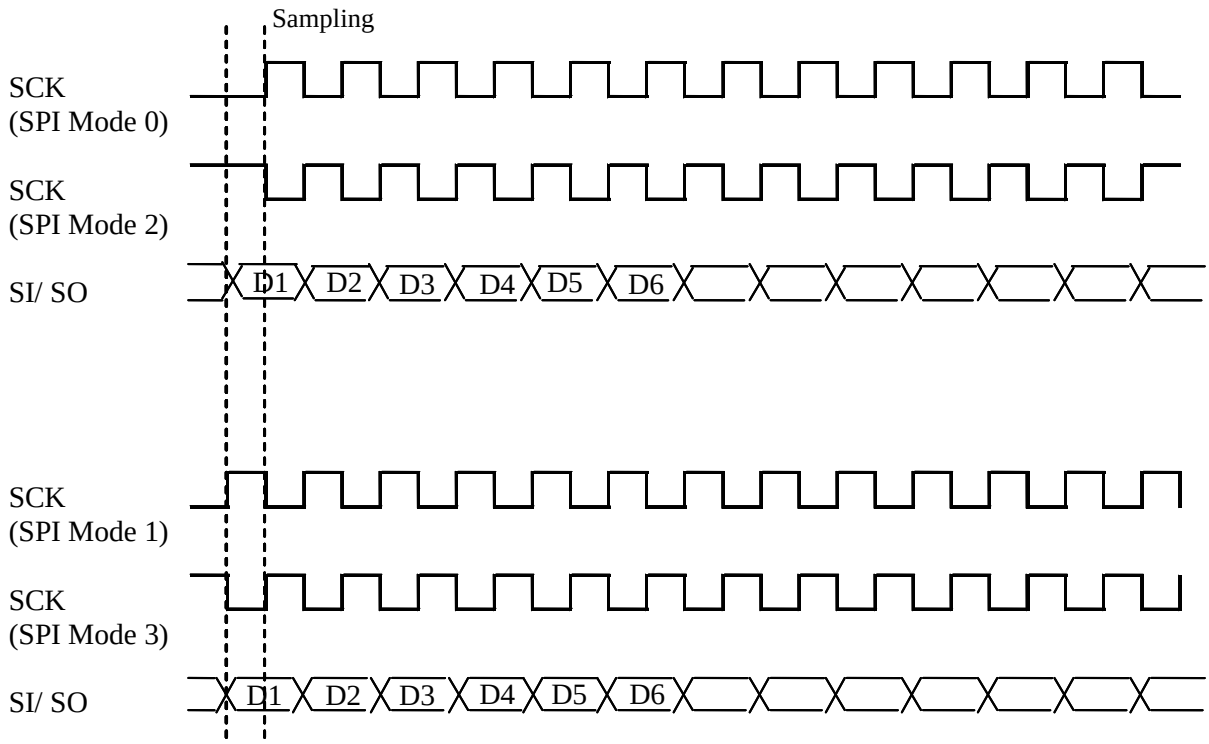


Figure 18-2. Timing of Each SPI Mode.

18.2.2. SPICLK (SPI Clock Divider Register)

Register	SPICLK	SPI Clock Divider Register	Address	0xFB81

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CLKDIV	R/W	0	SPI clock rate (only available in master mode) $SCK = CLK/4\,(CLKDIV + 1)$	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

18.2.3. SPIFMT (SPI Data Format Register)

Register	SPIFMT	SPI Data Format Register	Address	0xFB82

Bit	Bit Name	R/W	Initial	Description	Remarks
7	LSB	R/W	0	Order of data transfer bit 0: MSB first 1: LSB first	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	WORD	R/W	0	Transfer word length setting 000: 6 bits 001: 7 bits 010: 8 bits 011: 9 bits 100: 12 bits 101: 14 bits 110: 16 bits 111: Reserved	
1		R/W	0		
0		R/W	0		

18.2.4. SPISR (SPI Status Register)

Register	SPISR	SPI Status Register	Address	0xFB84

Bit	Bit Name	R/W	Initial	Description	Remarks
7	TEND	R/C	0	Transfer completion flag Read 0: Transfer is not completed Read 1: Transfer is completed Write 0: No change Write 1: The bit is cleared If TXFIFO = empty, the transfer completion flag is set when the transfer block is completed. When SPISR.TEND = 1 and SPIIER.TXENDIE = 1, an interrupt is generated. To clear the bit, write 1 to the bit.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	TFFULL	R	0	TXFIFO full status 0: TXFIFO is not full 1: TXFIFO is full	
2	TFEMPTY	R	1	TXFIFO empty status 0: TXFIFO is not empty 1: TXFIFO is empty	
1	RFFULL	R	0	RXFIFO full status 0: RXFIFO is not full 1: RXFIFO is full	
0	RFEMPTY	R	1	RXFIFO empty status 0: RXFIFO is not empty 1: RXFIFO is empty	

18.2.5. SPIESR (SPI Error Status Register)

Register	SPIESR	SPI Error Status Register	Address	0xFB85

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	TOVF	R/C	0	TXFIFO overflow flag Read 0: Overflow is not detected Read 1: Overflow is detected Write 0: No change Write 1: The bit is cleared When the SPIDR register is written during TXFIFO = full, the TXFIFO overflow flag is set. To clear the TXFIFO overflow flag, write 1 to the bit.	
2	TUDF	R/C	0	TXFIFO underflow flag Read 0: Underflow is not detected Read 1: Underflow is detected Write 0: No change Write 1: The bit is cleared When data is transferred during TXFIFO = empty, the TXFIFO underflow flag is set. To clear the TXFIFO underflow flag, write 1 to the bit.	
1	ROVF	R/C	0	RXFIFO overflow flag Read 0: Overflow is not detected Read 1: Overflow is detected Write 0: No change Write 1: The bit is cleared When data is received during RXFIFO = full, the RXFIFO overflow flag is set. To clear the RXFIFO overflow flag, write 1 to the bit.	
0	RUDF	R/C	0	RXFIFO underflow flag Read 0: Underflow is not detected Read 1: Underflow is detected Write 0: No change Write 1: The bit is cleared When the SPIDR register is read during RXFIFO = empty, the RXFIFO underflow flag is set. To clear the RXFIFO underflow flag, write 1 to the bit.	

18.2.6. SPIIER (SPI Interrupt Enable Register)

Register	SPIIER	SPI Interrupt Enable Register	Address	0xFB86

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	TXENDIE	R/W	0	TXEND interrupt enable 0: TXEND interrupt is disabled 1: TXEND interrupt is enabled When the bit is set to 1 and the SPISR.TEND bit is 1, the transmission interrupt is generated.	
3	TXERRIE	R/W	0	TX ERROR interrupt enable 0: TXERR interrupt is disabled 1: TXERR interrupt is enabled When the bit is set to 1 and the SPIESR.TOVF or SPIESR.TUDF bit is set, the transmission interrupt is generated.	
2	RXERRIE	R/W	0	RX ERROR interrupt enable 0: RXERR interrupt is disabled 1: RXERR interrupt is enabled When the bit is set to 1 and the SPIESR.ROVF or SPIESR.RUDF bit is set, the reception interrupt is generated.	
1	TXFIFOIE	R/W	0	TXFIFO interrupt enable 0: TXFIFO interrupt is disabled 1: TXFIFO interrupt is enabled When the bit is set to 1 and the SPISR.TFFULL bit is set, the transmission interrupt is generated.	
0	RXFIFOIE	R/W	0	RXFIFO interrupt enable 0: RXFIFO interrupt is disabled 1: RXFIFO interrupt is enabled When the bit is set to 1 and the SPISR.RFEMPTY bit is set, the reception interrupt is generated.	

18.2.7. SPIDRL (SPI Data Register Low)

Register	SPIDRL		SPI Data Register Low		Address	0xFB88
Bit	Bit Name	R/W	Initial	Description		Remarks
7	SPIDRL	R/W	0	The lower byte of the TX/RX data of the SPI		
6		R/W	0			
5		R/W	0			
4		R/W	0			
3		R/W	0			
2		R/W	0			
1		R/W	0			
0		R/W	0			

18.2.8. SPIDRH (SPI Data Register High)

When a data length is ≤ 8 bits, the SPI uses the SPIDRL register only, and does not use the SPIDRH register. When a data length is ≥ 9 bits, it is required to read or write in the order of the SPIDRL register to the SPIDRH register.

In a data length ≥ 9 bits, the FIFO status changes as follows:

- **TXFIFO**
  When writing to the SPIDRH register, the SPISR.TFEMPTY and SPISR.TFFULL bits are updated.

- **RXFIFO**
  When reading the SPIDRH register, the SPISR.RFFULL and SPISR.RFEMPTY bits are updated.

Register	SPIDRH		SPI Data Register High		Address	0xFB89
Bit	Bit Name	R/W	Initial	Description		Remarks
7	SPIDRH	R/W	0	The higher byte of the TX/RX data of the SPI		
6		R/W	0			
5		R/W	0			
4		R/W	0			
3		R/W	0			
2		R/W	0			
1		R/W	0			
0		R/W	0			

18.3. Interrupt Generation

18.3.1. Transmission Interrupt (INT_TX)

Figure 18-3 shows the transmission interrupt (INT_TX) generation logic.

- **SPISR.TFFULL Bit:**
  When TXFIFO = full, the SPISR.TFFULL bit is set to 1. The SPISR.TFFULL bit is cleared at a transmission start.

- **SPIESR.TUDF Bit:**
  When a transmission starts during TXFIFO = empty, the SPIESR.TUDF bit set to 1. To clear the TXFIFO underflow flag, write 1 to the SPIESR.TUDF bit.

- **SPIESR.TOVF Bit:**
  When a writing generates during TXFIFO = full, the SPIESR.TOVF bit set to 1. To clear the TXFIFO overflow flag, write 1 to the SPIESR.TOVF bit.

- **SPISR.TEND Bit:**
  When a transmission is completed during TXFIFO = empty, the SPISR.TEND bit set to 1. To clear the SPISR.TEND bit, write 1 to the SPISR.TEND bit.

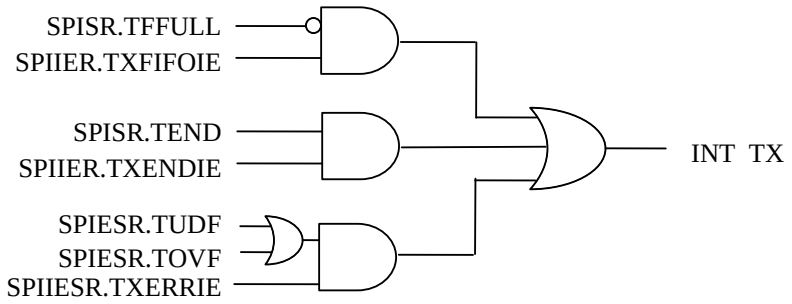


Figure 18-3. INT_TX Generation Logic

18.3.2. Reception Interrupt (INT_RX)

Figure 18-4 shows the reception interrupt (INT_RX) generation logic.

- **SPISR.RFEMPTY Bit:**

The SPISR.RFEMPTY bit is cleared while a reception data is stored in the RXFIFO. When all data stored in RXFIFO is read, the RXFIFO becomes empty, and the SPISR.RFEMPTY bit is set to 1.

- **SPIESR.RUDF Bit:**

When the SPIDRL register is more read when RXFIFO = empty, the RXFIFO underflow is generated, and the SPIESR.RUDF bit is set to 1. To clear the RXFIFO underflow flag, write 1 to the SPIESR.RUDF bit.

- **SPIESR.ROVF Bit:**

When data is more received and stored in the RXFIFO when RXFIFO = full, the RXFIFO overflow is generated, and the SPIESR.ROVF bit is set to 1. To clear the RXFIFO overflow flag, write 1 to the SPIESR.ROVF bit.

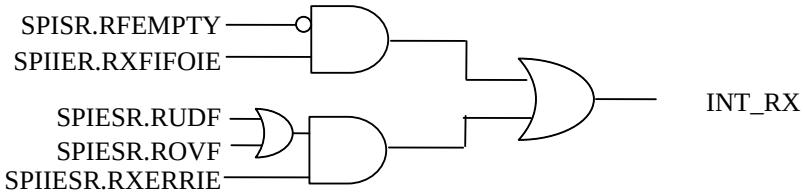


Figure 18-4. INT_RX Generation Logic

18.4. Timing and Connection

18.4.1. Master Mode

Figure 18-5 and Figure 18-6 shows the timing when the SPI mode is 0 and 1, respectively. Conditions for Figure 18-5 and Figure 18-6 are as follows:

- Transfer word length = 8 bits (SPIFMT.WORD = 0b010), and
- SPI clock rate = 1/2 (SPICLK.CLKDIV = 0x00).

Figure 18-7 and Figure 18-8 shows the connection between the master and subordinate devices. The master device is MD6603.

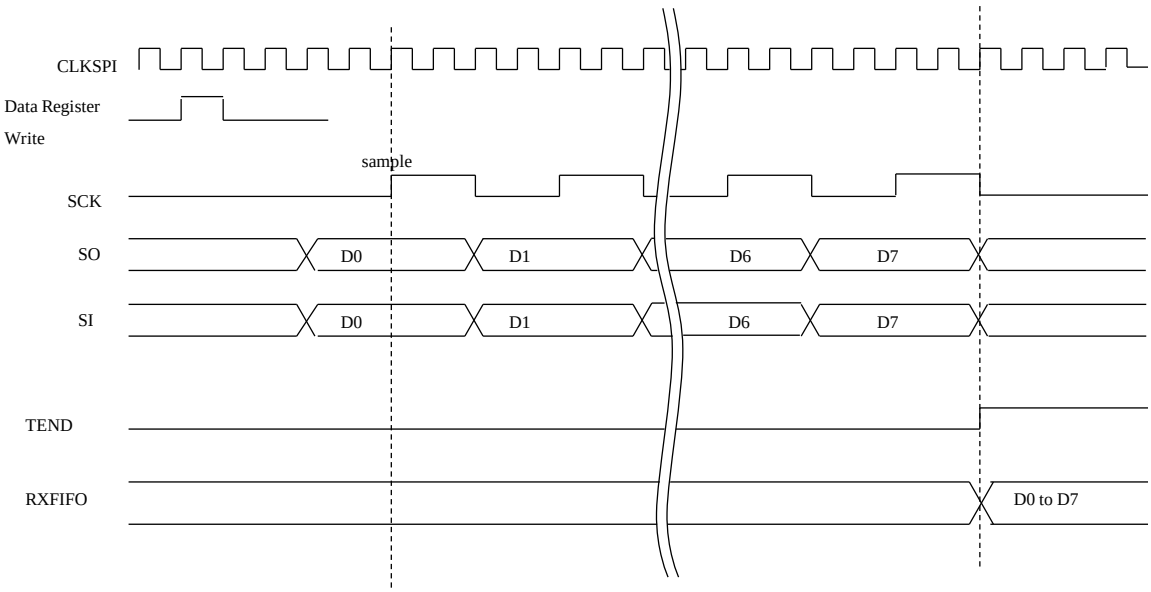


Figure 18-5. Timing in Master Mode (SPI Mode: 0)

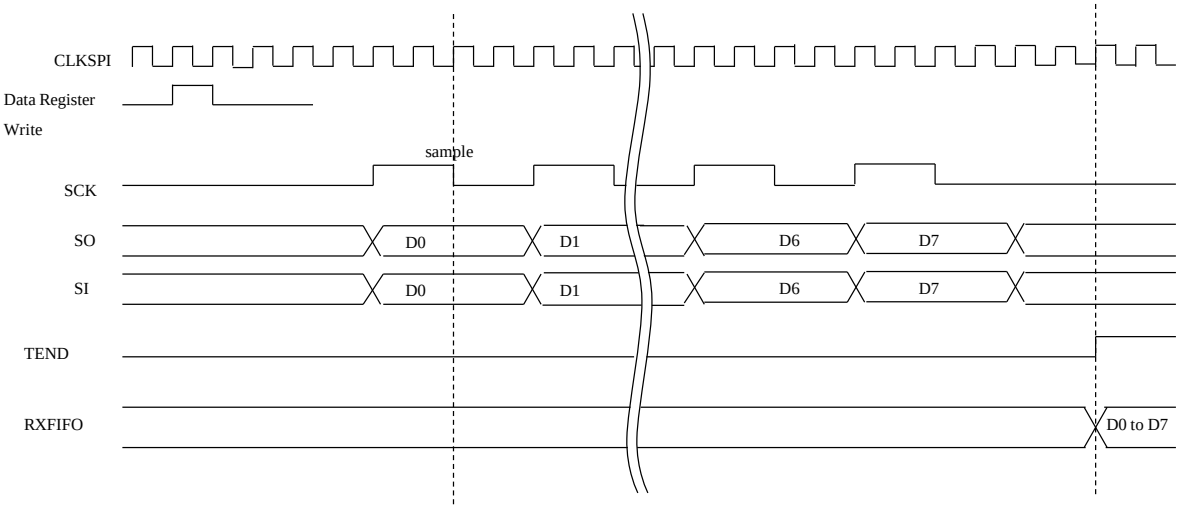


Figure 18-6. Timing in Master Mode (SPI Mode: 1)

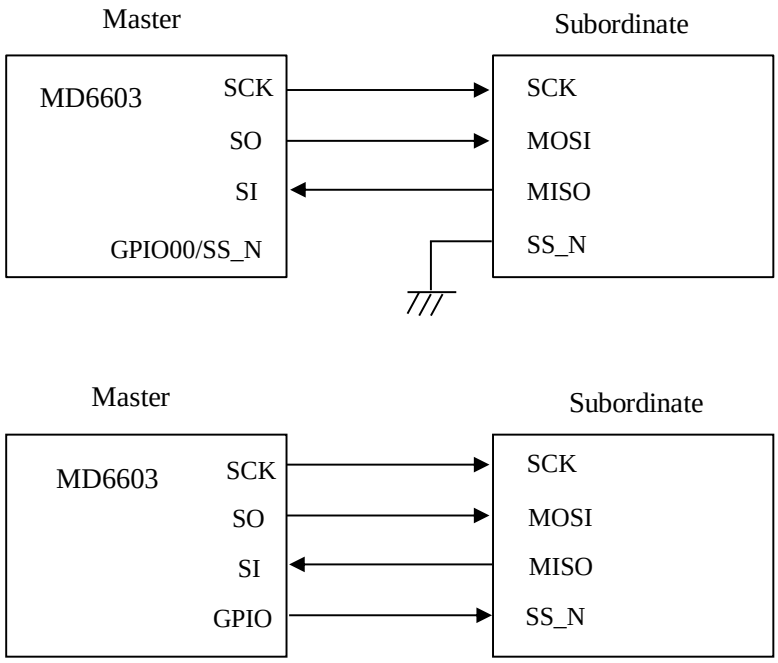


Figure 18-7. Connection between Single Master and Single Subordinate (Master Device: MD6603)

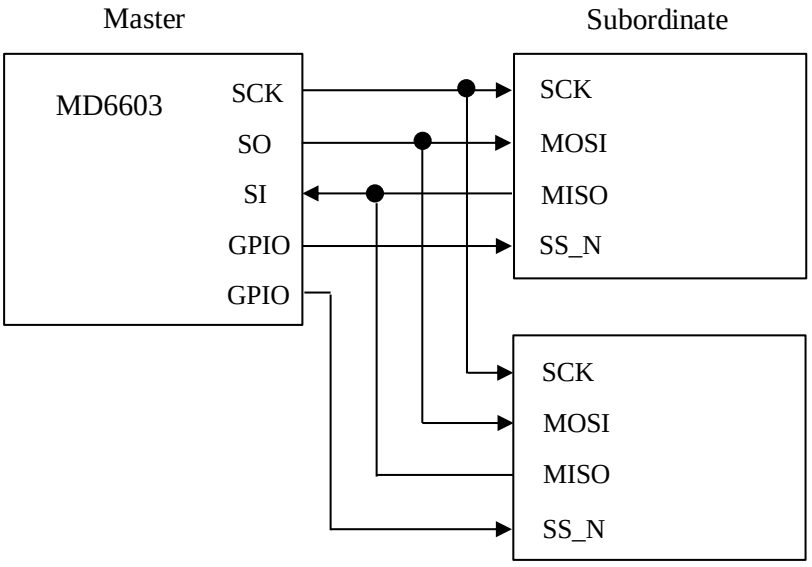


Figure 18-8. Connection between Single Master and Multi-subordinate (Master Device: MD6603)

If a subordinate device has the SS_N pin, the SS_N pin should be controlled by software via the GPIO pin. In multi-subordinate control, each SS_N pin of the subordinate device should be connected to the GPIO pin.

The SO and SI pins of the MD6603 should be connected to the MOSI and MISO pins of the subordinate device, respectively.

18.4.2. Subordinate Mode

Figure 18-9 shows the timing when the SPI mode is 1. The condition in Figure 18-9 is transfer word length = 8 bits (i.e., SPIFMT.WORD = 0b010).

Figure 18-10 shows the connection between the master and subordinate devices. The subordinate device is MD6603.

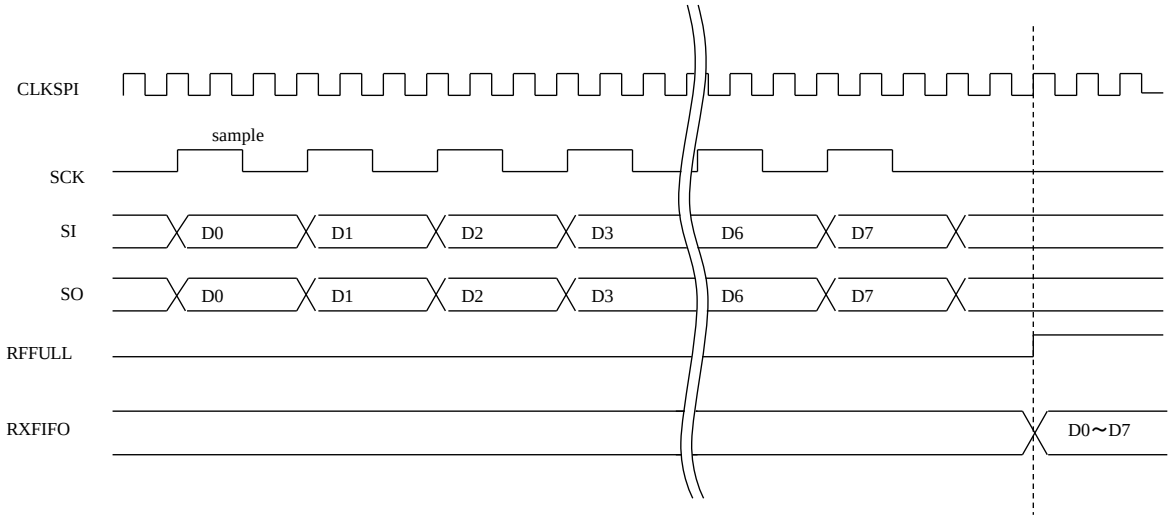


Figure 18-9. Timing in Subordinate Mode (SPI Mode: 1)

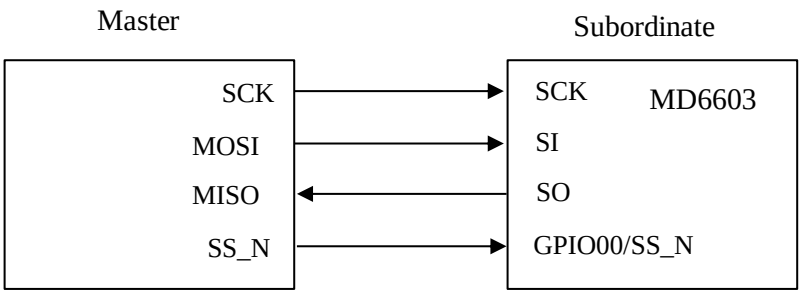


Figure 18-10. Connection in Subordinate Mode (Subordinate Device: MD6603)

The SS_N pin of the master device is connected to the SS_N pin of theMD6603.

The SI and SO pins of MD6603 should be connected to the MOSI and MISO pins of the master device, respectively. The SCK pin of MD6603 is an input pin.

18.5. Operation

18.5.1. Master Mode

● **Transmission Start**

When the transmission data[1] is written to the SPIDRL/H register while the TXFIFO bit is empty (i.e., SPISR.TFEMPTY = 1), the transmission data is transferred to the shift register via the TXFIFO, and then the transfer as shown in Figure 18-11 starts.

If data is written to the SPIDRL/H register consecutively in order to transmit multiple data, the values of 2 data are stored in the TXFIFO, and the TXFIFO status becomes full (i.e., SPISR.TFFULL = 1).

● **Reception Start**

When a dummy data is written to the SPIDRL/H register while the RXFIFO bit is empty (i.e., SPISR.RFEMPTY = 1), the SCK is generated to start reception. As shown in Figure 18-12, a reception data is sampled by the SCK, and is latched to the shift register.

● **Transmission Completion**

The SPI transfer completes at the SCK edge corresponding to the settings of the SPICR.CPHA and SPICR.CPOL bits, and then the SPISR.TEND is set to 1. The final sampling timing depends on a data bit length.

● **Reception Completion**

When the reception data is stored in the RXFIFO, the RXFIFO status becomes non-empty (i.e., SPISR.REMPTY = 0). To read the data in the RXFIFO, read the SPIDRL/H[2] register. The RXFIFO status indicates full or empty.

---

[1] Data length ≤ 8 bit: To update the TXFIFO status, write to the SPIDRL register.
Data length ≥ 9 bit: To update the TXFIFO status, write to the SPIDRH register.

[2] Data length ≤ 8 bit: To update the RXFIFO status, read the SPIDRL register.
Data length ≥ 9 bit: To update the RXFIFO status, read the SPIDRH register.

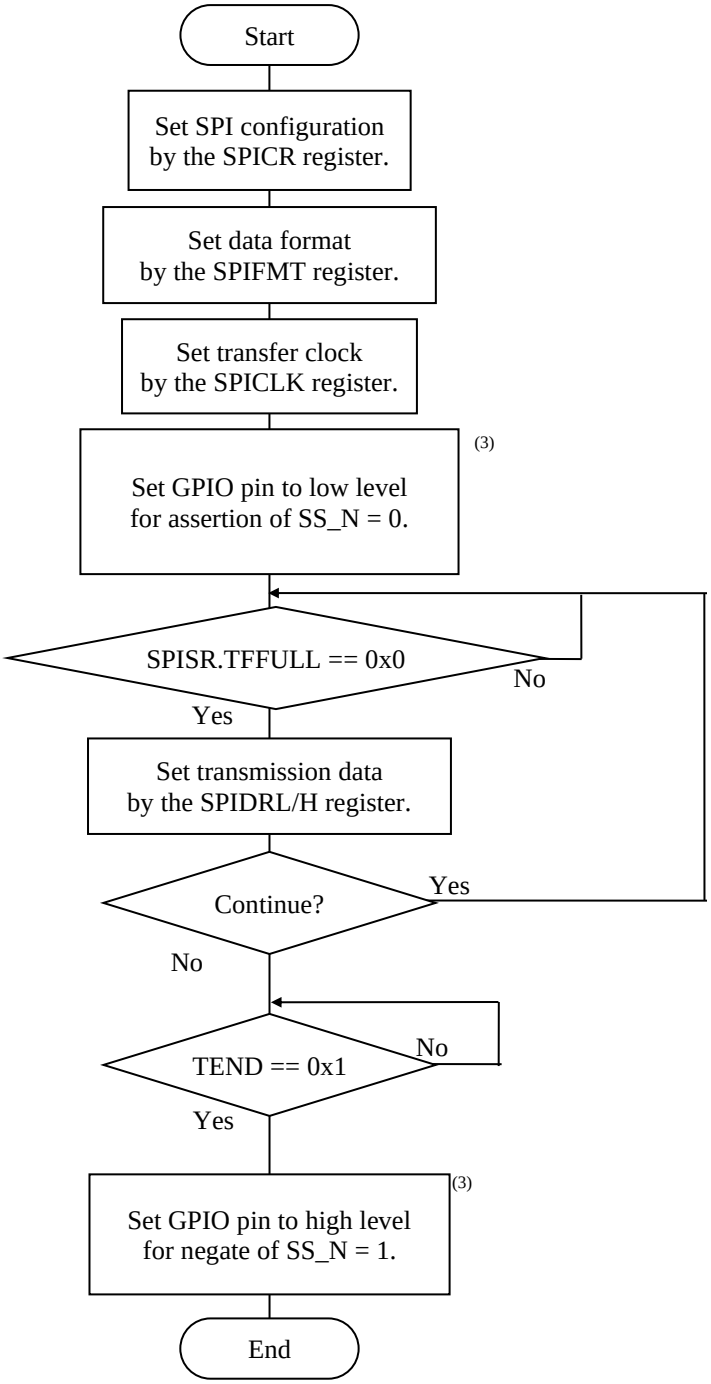


Figure 18-11. Master Mode (Transmission)

(3) When the SS_N pin can be asserted or negated, the GPIO pin can be used for any pins.

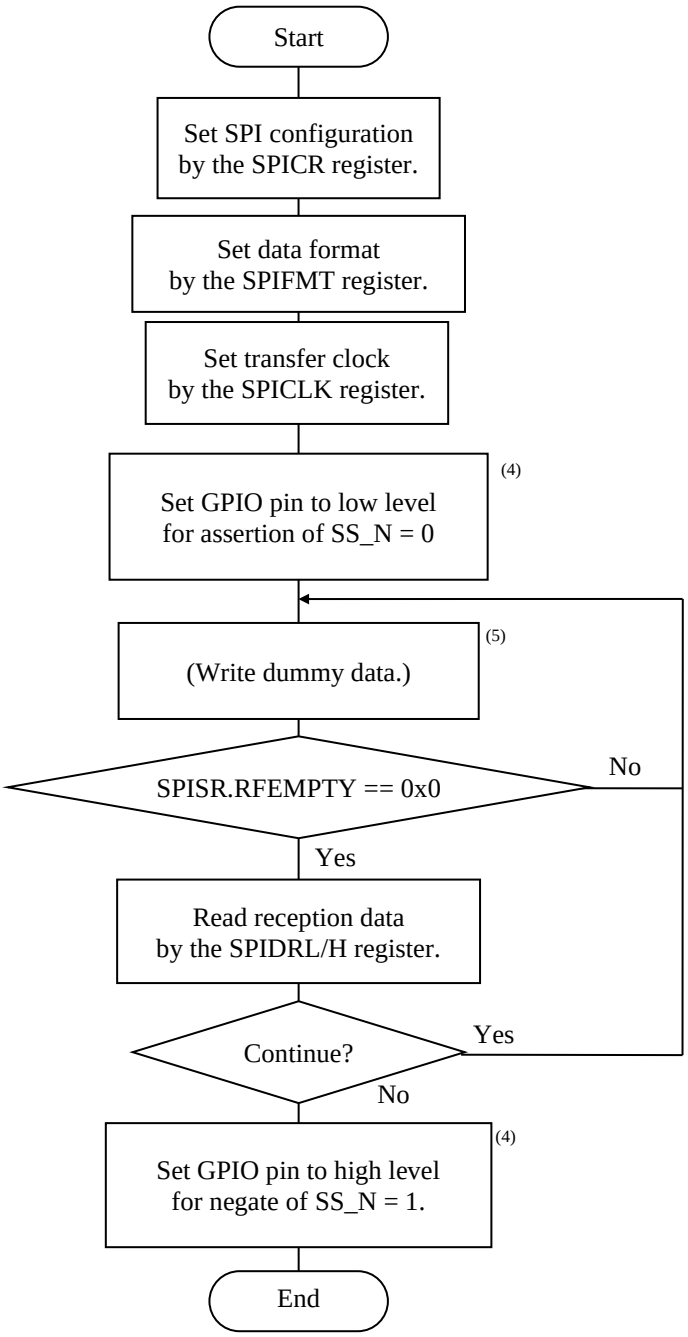


Figure 18-12. Master Mode (Reception)

(4) When the SS_N pin can be asserted or negated, the GPIO pin can be used for any pins.
(5) The SCK should be generated by a dummy data transmission in the master mode.

18.5.2. Subordinate Mode

● **Transmission Start**

The SPI operates with the subordinate mode while the SS_N pin is low level. Data is transferred at the sampling timing of the SCK input. When the data[(1)] is written to the SPIDRL/H register during TXFIFO = empty, the transmission data is transferred to the shift register via the TXFIFO. If the values of 2 data exist in the TXFIFO when the data is written to the SPIDRL/H register consecutively, the TXFIFO status becomes full (i.e., SPISR.TFFULL = 1).

● **Transmission Completion**

When the final SCK edge is received, the SPI completes a serial transmission. In addition, the SPISR.TEND bit is set to 1.

● **Reception Completion**

When the reception data is written to the RXFIFO, the RXFIFO status indicates non-empty (i.e., SPISR.REMPTY = 0). To read the data in the RXFIFO, read the SPIDRL/H[(2)] register.

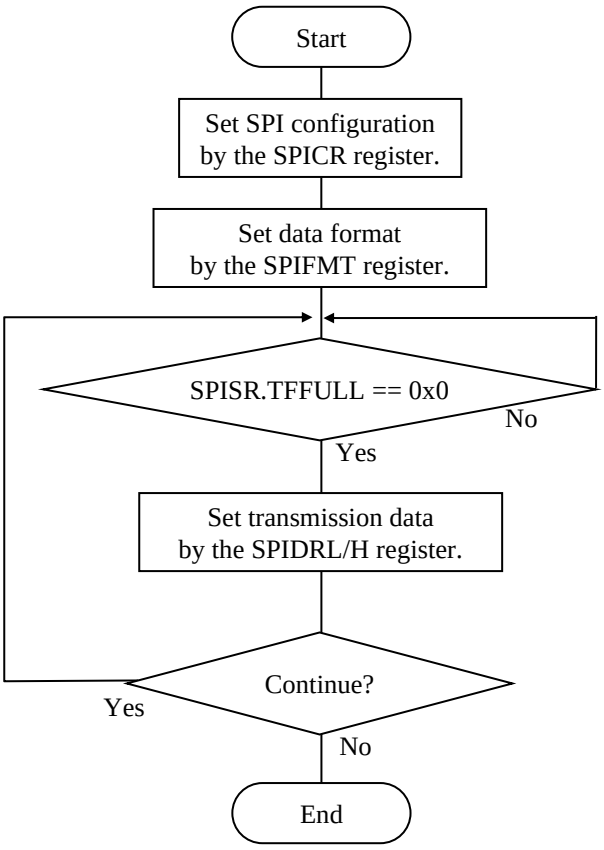


Figure 18-13. Subordinate Mode (Transmission)

(1) Data length ≤ 8 bit: To update the TXFIFO status, write to the SPIDRL register.
Data length ≥ 9 bit: To update the TXFIFO status, write to the SPIDRH register.
(2) Data length ≤ 8 bit: To update the RXFIFO status, read the SPIDRL register.
Data length ≥ 9 bit: To update the RXFIFO status, read the SPIDRH register.

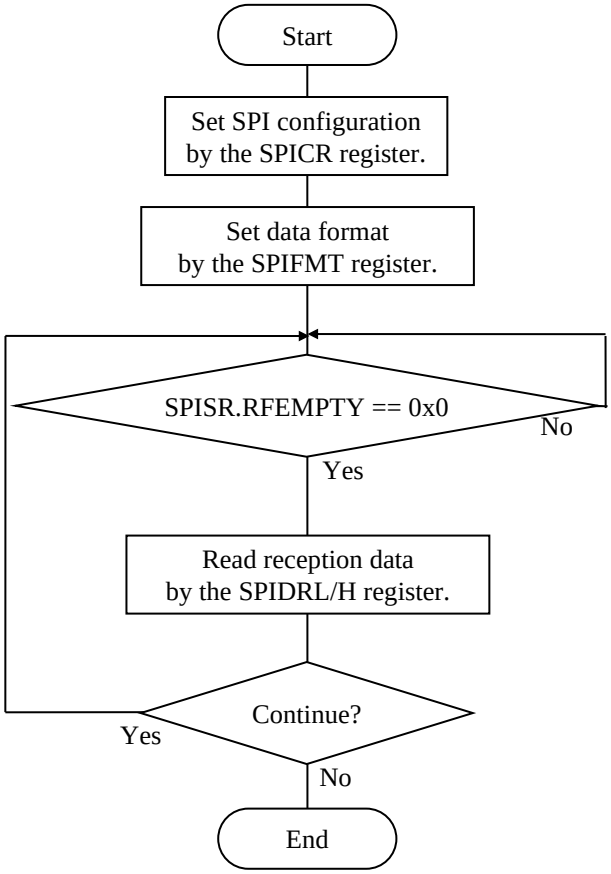


Figure 18-14. Subordinate Mode (Reception)

19. I²C/SMBUS

19.1. Overview

The LSI has the I²C communication module supporting both the master and subordinate modes.

Table 19-1. I²C Functional Descriptions

Item	Description
Communication Method	- I²C bus method and SMBUS method - Master or subordinate mode can be selected
Clock	CLKFAST/8 CLKFAST/32 CLKFAST/128 CLKFAST/512
Supported Function	Clock stretching GCA (General Call Address)
Interrupt	Single source

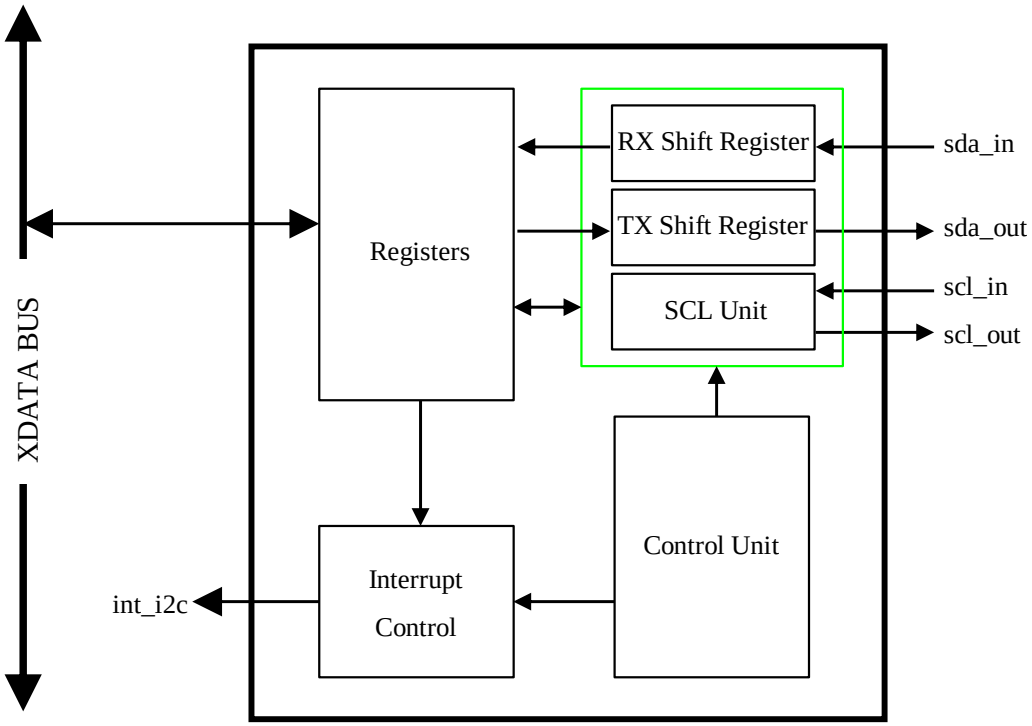


Figure 19-1. I²C Block Diagram

19.2. Register Descriptions

Table 19-2. List of Registers

Symbol	Name	Address	Initial Value
ICCR	I²C Bus Control Register	0xFC00	0x00
ICSR	I²C Bus Status Register	0xFC01	0x00
ICRXDR	I²C Bus Receive Data Register	0xFC02	0x00
ICTXDR	I²C Bus Transmit Data Register	0xFC03	0x00
ICTSAR	I²C Transmit Address Register	0xFC04	0x00
ICSAR	I²C Subordinate Address Register	0xFC05	0x00
ICCLK	I²C Clock Divide Register	0xFC06	0x03
ICCMD	I²C Command Register	0xFC07	0x00
ICSSTR	I²C Bus SDA Setup Time Register	0xFC08	0x01
ICSHTR	I²C Bus SDA Hold Time Register	0xFC09	0x00
ICHDSR0	I²C Bus Hardware Status Register0	0xFC0A	0xC0
ICHDSR1	I²C Bus Hardware Status Register1	0xFC0B	0x00
ICTIMER	I²C Time Base Register	0xFC10	0xFF
SMBINT	SMBUS INT Status Register	0xFC11	0x00
ICSAA	I²C Subordinate Alert Address Register	0xFC18	0x00
ICSAIR	I²C Subordinate Address Identifier Register	0xFC19	0x00

19.2.1. ICCR (I^2C Bus Control Register)

Register	ICCR	I^2C Bus Control Register	Address	0xFC00

Bit	Bit Name	R/W	Initial	Description	Remarks
7	ICE	R/W	0	I^2C Bus interface enable 0: I^2C Bus interface is disabled 1: I^2C Bus interface is enabled	
6	IEIC0	R/W	0	IRIC0 interrupt enable 0: IRIC0 interrupt is disabled 1: IRIC0 interrupt is enabled When the bit is set to 1 and the ICSR.IRIC0 bit is 1, the interrupt is generated.	
5	IEIC1	R/W	0	IRIC1 interrupt enable 0: IRIC1 interrupt is disabled 1: IRIC1 interrupt is enabled When the bit is set to 1 and the ICSR.IRIC1 bit is 1, the interrupt is generated.	
4	IEIC2	R/W	0	IRIC2 interrupt enable 0: IRIC2 interrupt is disabled 1: IRIC2 interrupt is enabled When the bit is set to 1 and the ICSR.IRIC2 bit is 1, the interrupt is generated.	
3	IEIC3	R/W	0	IRIC3 interrupt enable 0: IRIC3 interrupt is disabled 1: IRIC3 interrupt is enabled When the bit is set to 1 and the ICSR.IRIC3 bit is 1, the interrupt is generated.	
2	IEIC4	R/W	0	IRIC4 interrupt enable 0: IRIC4 interrupt is disabled 1: IRIC4 interrupt is enabled When the bit is set to 1 and the ICSR.IRIC4 bit is 1, the interrupt is generated.	
1	IEIC5	R/W	0	IRIC5 interrupt enable 0: IRIC5 interrupt is disabled 1: IRIC5 interrupt is enabled When the bit is set to 1 and the ICSR.IRIC5 bit is 1, the interrupt is generated.	
0	GCAE	R/W	0	GCA enable 0: GCA response is disabled 1: GCA response is enabled The bit controls the response to the GCA (General Call Address) in the subordinate mode.	

19.2.2. ICSR (I^2C Bus Status Register)

Register	ICSR	I^2C Bus Status Register		Address	0xFC01
Bit	Bit Name	R/W	Initial	Description	Remarks
7	BBSY	R	0	I^2C Bus busy detection flag 0: Bus released state 1: Bus occupied state The bit indicates whether the I^2C bus is occupied (bus busy) or released (bus free). When the start condition is detected, the bit enters the occupied state; when the stop condition is detected, the bit enters the released state.	
6	IRIC0	R/C	0	Subordinate access detection interrupt Read 0: Subordinate access is not detected Read 1: Subordinate access is detected Write 0: No change Write 1: The bit is cleared When the LSI is in the subordinate mode, if the GCA is detected by the subordinate address or GCAE = 1 and the ACK is sent, the bit is set to 1. The bit can be cleared by writing 1 to the bit from the CPU. For the setup timing of the bit, see Figure 19-8 and Figure 19-9.	
5	IRIC1	R/C	0	ACK/NACK reception interrupt 1 Read 0: ACK/NACK is not received Read 1: ACK/NACK is received Write 0: No change Write 1: The bit is cleared When the LSI completes to transmit an address or a command and receives the ACK or the NACK, the bit is set to 1. The bit can be cleared by writing 1 to the bit from the CPU. For the setup timing of the bit, see Figure 19-10 and Figure 19-11.	
4	IRIC2	R/C	0	ACK/NACK reception interrupt 2 Read 0: ACK/NACK is not received Read 1: ACK/NACK is received Write 0: No change Write 1: The bit is cleared When the LSI transmits data, if the transmission is completed and the ACK or the NACK is received, the bit is set to 1. The bit can be cleared by writing 1 to the bit from the CPU. For the setup timing of the bit, see Figure 19-9 and Figure 19-11.	

Register	ICSR	I^2C Bus Status Register		Address	0xFC01
Bit	Bit Name	R/W	Initial	Description	Remarks
3	IRIC3	R/C	0	Data reception completion interrupt Read 0: Data reception is not completed Read 1: Data reception is completed Write 0: No change Write 1: The bit is cleared When the data reception is completed, the bit is set to 1 before the ACK or the NACK is transmitted. The bit can be cleared by writing 1 to the bit from the CPU. For the setup timing of the bit, see Figure 19-8 and Figure 19-10.	
2	IRIC4	R	0	SMBUS interrupt 0: SMBUS interrupt is not detected 1: SMBUS interrupt is detected The logical OR of the interrupt sources of the SMBINT register is displayed on the bit. To clear the bit, clear the interrupt source of the SMBINT register.	
1	IRIC5	R/C	0	Stop condition interrupt Read 0: Stop condition is not detected Read 1: Stop condition is detected Write 0: No change Write 1: The bit is cleared When the stop condition is detected, the bit is set to 1. The bit can be cleared by writing 1 to the bit from the CPU.	
0	RXACK	R	0	Acknowledge bit 0: ACK is received 1: NACK is received In the master mode, information on the acknowledge bit received from the subordinate device is stored in the bit. In the subordinate mode, information on the acknowledge bit received from the master device is stored in the bit.	

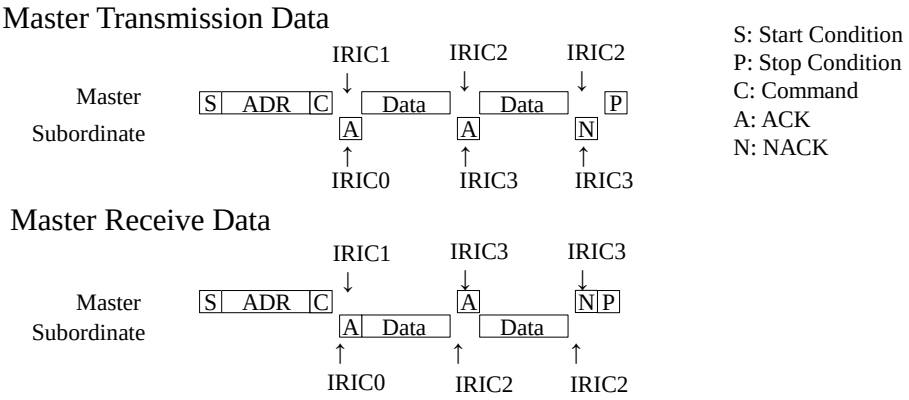


Figure 19-2. Interrupt Timings

19.2.3. ICRXDR (I^2C Bus Receive Data Register)

After the data reception is completed, the data of the ICRXDR register must be read. Whether the data reception is completed can be confirmed by the ICSR.IRIC3 bit.

Register	ICRXDR		I^2C Bus Receive Data Register	Address	0xFC02
Bit	Bit Name	R/W	Initial	Description	Remarks
7	ICRXDR	R	0	Reception data	
6		R	0		
5		R	0		
4		R	0		
3		R	0		
2		R	0		
1		R	0		
0		R	0		

19.2.4. ICTXDR (I^2C Bus Transmit Data Register)

When data is written to the ICTXDR register, the transmission is started.

Writing is prohibited until the data transmission from the ICTXDR register is completed. If data is written during the data transmission, the data will be destroyed.

Register	ICTXDR		I^2C Bus Transmit Data Register	Address	0xFC03
Bit	Bit Name	R/W	Initial	Description	Remarks
7	ICTXDR	R/W	0	Transmission data	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

19.2.5. ICTSAR (I^2C Transmit Address Register)

After setting the ICTSAR register, set the ICCMD register to start communications.

Register	ICTSAR	I^2C Transmit Address Register	Address	0xFC04

Bit	Bit Name	R/W	Initial	Description	Remarks
7	ADR	R/W	0	Transmission address	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0	CMD	R/W	0	Transmission command 0: Write command 1: Read command	

19.2.6. ICSAR (I^2C Subordinate Address Register)

When the received subordinate address matches the ICSAR.SVA bits, the LSI operates as the subordinate device specified by the master device. When the ICSAR.SVA bits = 0, the IC does not judge whether or not the subordinate address matches the ICSAR.SVA bits.

The ICSAR.CMD bit of the reception command is updated when the ICSAR.CMD bit matches with either of the following addresses: with the GCA when GCA = enabled; with the SAA when SAA = enabled.

Register	ICSAR	I^2C Subordinate Address Register	Address	0xFC05

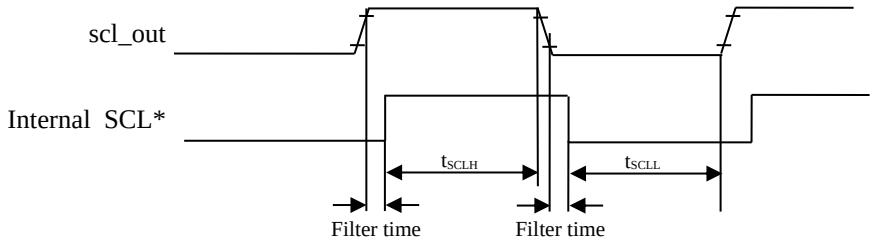
Bit	Bit Name	R/W	Initial	Description	Remarks
7	SVA	R/W	0	Subordinate address	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0	CMD	R	0	Reception command 0: Write command 1: Read command	

19.2.7. ICCLK (I^2C Clock Divider Register)

Register	ICCLK	I^2C Clock Divider Register	Address	0xFC06

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	DIV	R/W	1	SCL minimum pulse width 00: t_{SCLH}/t_{SCLL} is set to 9 cycles of CLKFAST 01: t_{SCLH}/t_{SCLL} is set to 32 cycles of CLKFAST 10: t_{SCLH}/t_{SCLL} is set to 128 cycles of CLKFAST 11: t_{SCLH}/t_{SCLL} is set to 512 cycles of CLKFAST	
0		R/W	1		

Set the minimum pulse width of SCL to the ICCLK.DIV bits. The SCL pulse width on the bus (t_{SCLH}/t_{SCLL}) is 5 cycles wider than the setting of the ICCLK.DIV bits due to synchronization with the noise filter.



*Internal SCL: SCL after noise filter

Figure 19-3. Definition of Timing of t_{SCLH}/t_{SCLL}

Table 19-3. Limitation of Minimum Clock Frequency

Item	Normal Mode	High-speed Mode
SCL Frequency	0 kHz to 100 kHz	0 kHz to 400 kHz
CLKFAST Frequency	1.74 MHz or more	6.67 MHz or more

19.2.8. ICCMD (I^2C Command Register)

The ICCMD register is initialized when it is reset or the ICCR.ICE bit is cleared.

Register	ICCMD	I^2C Command Register	Address	0xFC07

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	NACK	R/W	0	NACK response 0: NACK response is not returned 1: NACK response is returned In the subordinate mode, the NACK response is returned after data is received. When the NACK response is completed, the bit is cleared. The bit is not set in the following cases: ● When ACK = 1 ● When the ACK and NACK bits are simultaneously set ● In the master mode	
3	ACK	R/W	0	ACK response 0: ACK response is not returned 1: ACK response is returned In the subordinate mode, the ACK response is returned after data is received. When the ACK response is completed, the bit is cleared. The bit is not set when: ● NACK = 1 ● In the master mode	
2	RDCNT	R/W	0	Read continue 0: Continuous reading is not requested 1: Continuous reading is requested In the master mode, reading of the next data is requested while data is being read. The bit is cleared when receiving of the read data starts. In the subordinate mode, the bit is not set.	
1	END	R/W	0	Stop condition generation 0: Generating a condition to stop operation is not requested 1: Generating a condition to stop operation is requested When the bit is set to 1, the data transfer is finished and the generation of the stop condition is requested. Then, the LSI transits to the subordinate mode. The bit is cleared if the stop condition is detected. The bit is not set in the following cases: ● When GO = 1 ● When the bit is simultaneously set with the GO bit ● In the subordinate mode	

Register	ICCMD	I²C Command Register		Address	0xFC07
Bit	Bit Name	R/W	Initial	Description	Remarks
0	GO	R/W	0	Start condition generation 0: No start condition is generated 1: A start condition is generated When the bit is set to 1, the start condition is generated and the starting of address or command transmission is requested. Then the LSI transits to the master mode. The bit is cleared when the start condition is detected. In the subordinate mode, do not set the bit to 1. When the END bit is set to 1, the bit is not set.	

19.2.9. ICSSTR (I²C Bus SDA Setup Time Register)

Register	ICSSTR	I²C Bus SDA Setup Time Register		Address	0xFC08
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	ICSSTR	R/W	0	SDA setup time	
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	1		

The ICSSTR.ICSSTR bits define the setup time of the SDA output to the SCL rising. The minimum and maximum setup time values of the SDA output are 0 and 0x3F, respectively. The setup time of the SDA output, $t_{SU:DAT}$, can be calculated using the following equation.

$$t_{SU:DAT} = (\text{ICSSTR} + 1) \times \text{cycle of CLKFAST (ns)}$$

When CLKFAST is 60 MHz, the setup range of the setup time of the SDA output is 16 ns to 1.07 μs.

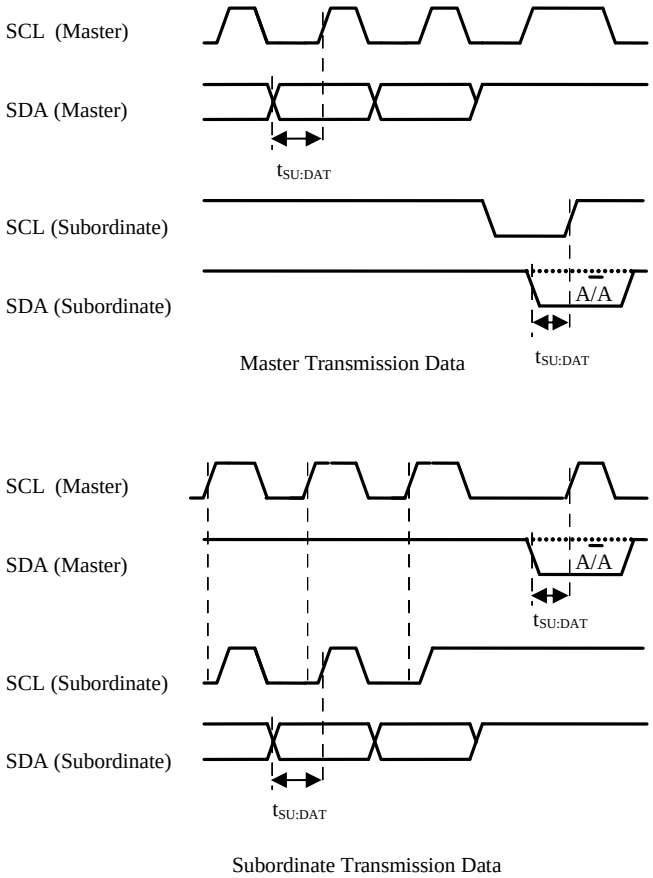


Figure 19-4. Relationship between SDA Output Setup Time and SCL Rising

19.2.10. ICSHTR (I^2C Bus SDA Hold Time Register)

Register	ICSHTR	I^2C Bus SDA Hold Time Register		Address	0xFC09
Bit	Bit Name	R/W	Initial	Description	Remarks
7	ICSHEXP	R/W	0	Extended SDA output holding 0: SDA output holding is not extended 1: SDA output holding is extended When using the LSI with CLKFAST ≤ 12.5 MHz or when not using the I^2C as the SMBUS, set the bit to 0. When using the LSI with CLKFAST > 12.5 MHz and the I^2C as the SMBUS, set the bit to 1. When the bit is set to 1, the SDA output is delayed by 3 cycles of the CLKFAST compared with setting the bit to 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	ICSHTR	R/W	0	Time setting of SDA holding	
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

The ICSHTR.ICSHTR bits define the delay time of the SDA to the SCL when receiving the subordinate address and subordinate data. This delay time is used to ensure holding time internally.

$$t_{HD:DAT} = \text{ICSHTR} \times \text{cycle of CLKFAST (ns)}$$

When CLKFAST is 60 MHz, the range of the delay time which can be set to the ICSHTR.ICSHTR bits is 0 ns to 516 ns. The setting time of the ICSSTR.ICSSTR and ICSHTR.ICSHTR bits must be set so as to satisfy the following relation equations.

$$(\text{ICSSTR.ICSSTR bits seting time}) > (\text{ICSHTR.ICSHTR bits setting time}) - (\text{Relative delay time of SDA to SCL})$$

$$(\text{Relative delay time of SDA to SCL}) = (\text{Delay time of SCL}) - (\text{Delay time of SDA})$$

Where, the relative delay time of SDA to SCL is the difference in delay time between the SDA and SCL signals on the I^2C bus. Pay attention to this difference in delay time because it depends on the actual operating environment of the I^2C bus.

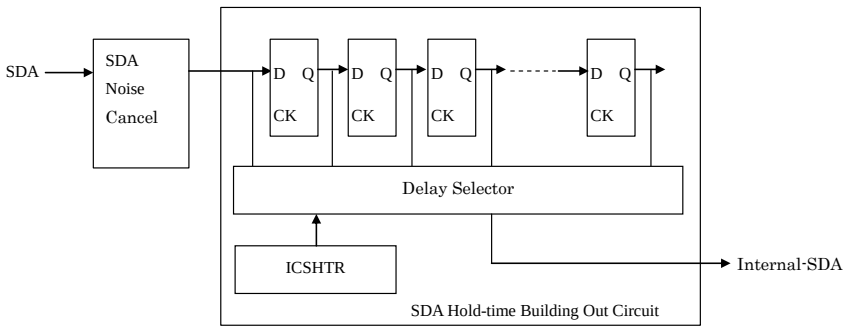


Figure 19-5. Internal SDA Generation Block Diagram

19.2.11. ICHDSR0 (I^2C Bus Hardware Status Register0)

Register	ICHDSR0	I^2C Bus Hardware Status Register0	Address	0xFC0A

Bit	Bit Name	R/W	Initial	Description	Remarks
7	SDAMON	R	1	SDA signal monitor 1 0: SDA signal is low level 1: SDA signal is high level	
6	SCLMON	R	1	SCL signal monitor 1 0: SCL signal is low level 1: SCL signal is high level	
5	SDACHG	R/C	0	SDA signal monitor 2 (SDAMON bit variation) Read 0: Variation of the SDAMON bit has not been detected Read 1: Variation of the SDAMON bit has been detected Write 0: No change Write 1: The bit is cleared The bit indicates whether or not the SDA signal has varied. The bit can be cleared by writing 1 to the bit from the CPU.	
4	SCLCHG	R/C	0	SCL signal monitor 2 (SCLMON bit variation) Read 0: Variation of the SCLMON bit has not been detected Read 1: Variation of the SCLMON bit has been detected Write 0: No change Write 1: The bit is cleared The bit indicates whether or not the SCL signal has varied. The bit can be cleared by writing 1 to the bit from the CPU.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

19.2.12. ICHDSR1 (I^2C Bus Hardware Status Register1)

Register	ICHDSR1	I^2C Bus Hardware Status Register1		Address	0xFC0B
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	I2C_ST	R	0	State of the state machine for controlling bus interface	
3		R	0		
2		R	0		
1		R	0		
0		R	0		

19.2.13. ICTIMER (I^2C Time Base Register)

Register	ICTIMER	I^2C Time Base Register		Address	0xFC10
Bit	Bit Name	R/W	Initial	Description	Remarks
7	TIME	R/W	1	To generate a timing signal of 1 ms, use the following equation. $\text{ICTIMER} = \left(\frac{\text{Frequency of CLKFAST (kHz)}}{128}\right) - 1$	
6		R/W	1		
5		R/W	1		
4		R/W	1		
3		R/W	1		
2		R/W	1		
1		R/W	1		
0		R/W	1		

The ICTIMER.TIME bits define a cycle of the reference timing signal for detecting the violation defined by the SMBUS standards. The setting value is used for the violation detection by the SMBINT register. The ICTIMER.TIME bits should be set so that a timing signal of 1 ms is generated.

The equation below defines the relationship between the timing signal cycle and the ICTIMER.TIME bits.

$$\text{Timing signal cycle (ms)} = \frac{128 \times (\text{TIME} + 1)}{\text{CLKFAST Frequency (kHz)}}$$

Compared to the judgement time when the timing signal cycle = 1 ms, the judgement time becomes half when the timing signal cycle = 0.5 ms, but becomes double when the timing signal cycle = 2 ms.

When the CLKFAST frequency > 32768 kHz, the timing signal cycle cannot be set to 1 ms. Even if the maximum value of 0xFF is set to the ICTIMER.TIME bits, the violation detection time is shorter than the time defined by the SMBUS standards.

19.2.14. SMBINT (SMBUS INT Status Register)

To detect the violation defined by the SMBUS standards using the IRSM0 to IRSM2 bits, the ICTIMER register must be set to 1 ms.

Register	SMBINT		SMBUS INT Status Register		Address	0xFC11
Bit	Bit Name	R/W	Initial	Description		Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.		
6	Reserved	R	0	The read value is 0. The write value must always be 0.		
5	Reserved	R	0	The read value is 0. The write value must always be 0.		
4	Reserved	R	0	The read value is 0. The write value must always be 0.		
3	Reserved	R	0	The read value is 0. The write value must always be 0.		
2	IRSM2	R/C	0	SMBUS interrupt monitor 2 Read 0: TIMEOUT violation is not detected Read 1: TIMEOUT violation is detected Write 0: No change Write 1: The bit is cleared When the TIMEOUT violation defined by the SMBUS standards is detected, the bit is set to 1. The bit can be cleared by writing 1 to the bit from the CPU.		
1	IRSM1	R/C	0	SMBUS interrupt monitor 1 Read 0: TLOW:SEXT violation is not detected Read 1: TLOW:SEXT violation is detected Write 0: No change Write 1: The bit is cleared When the TLOW:SEXT violation defined by the SMBUS standards is detected, the bit is set to 1. The bit can be cleared by writing 1 to the bit from the CPU.		
0	IRSM0	R/C	0	SMBUS interrupt monitor 0 Read 0: TLOW:MEXT violation is not detected Read 1: TLOW:MEXT violation is detected Write 0: No change Write 1: The bit is cleared When the TLOW:MEXT violation defined by the SMBUS standards is detected, the bit is set to 1. The bit can be cleared by writing 1 to the bit from the CPU.		

19.2.15. ICSAA (I^2C Subordinate Alert Address Register)

When the subordinate alert address defined by the ICSAIR.SAAEN bit is enabled and the received subordinate address matches the ICSAA.SAA bits, the LSI operates as the subordinate device specified by the master device. When ICSAA.SAA bits = 0, the IC does not judge whether or not the subordinate address matches the ICSAA.SAA bits.

The ICSAA.CMD bit of the reception command is updated when the ICSAR.CMD bit matches with either of the following addresses regardless of enabling/disabling the SAA: with the SVA; with the GCA when GCA = enabled; with the SSA when SSA = enabled.

Register	ICSAA	I^2C Subordinate Alert Address Register		Address	0xFC18
Bit	Bit Name	R/W	Initial	Description	Remarks
7	SAA	R/W	0	Subordinate alert address	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0	CMD	R	0	Reception command 0: Write command 1: Read command	

19.2.16. ICSAIR (I^2C Subordinate Address Identifier Register)

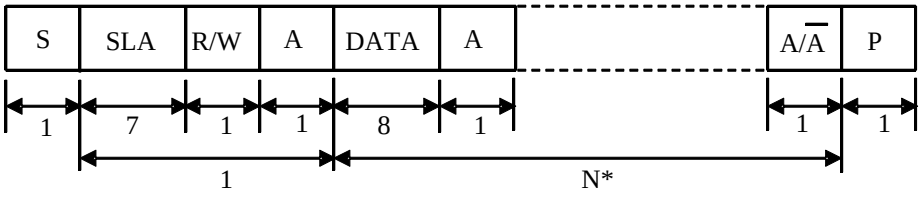
Register	ICSAIR	I^2C Subordinate Address Identifier Register	Address	0xFC19

Bit	Bit Name	R/W	Initial	Description	Remarks
7	SAAEN	R/W	0	Subordinate alert address enable 0: Subordinate alert address is disabled 1: Subordinate alert address is enabled When the bit is set to 0, if I^2C receives the address matching the ICSAA.SAA bits, it responds the NACK and does not reserve the next data. When the bit is set to 1, if I^2C receives the address matching the ICSAA.SAA bits, it responds the ACK and reserves the next data.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	SAI2	R	0	Subordinate address indicator 2 0: The last received address does not match the ICSAR.SVA bits 1: The last received address matches the ICSAR.SVA bits In the subordinate mode, the bit indicates whether or not the last received address matches the ICSAR.SVA bits.	
1	SAI1	R	0	Subordinate address indicator 1 0: The last received address does not match the ICSAA.SAA bits 1:The last received address matches the ICSAA.SAA bits In the subordinate mode, the bit indicates whether or not the last received address matches the ICSAA.SAA bits.	
0	SAI0	R	0	Subordinate address indicator 0 0: GCA does not match the last received address 1: GCA matches the last received address In the subordinate mode, the bit indicates whether or not GCA (General Call Address) matches the last received address.	

19.3. I²C Bus Data Format

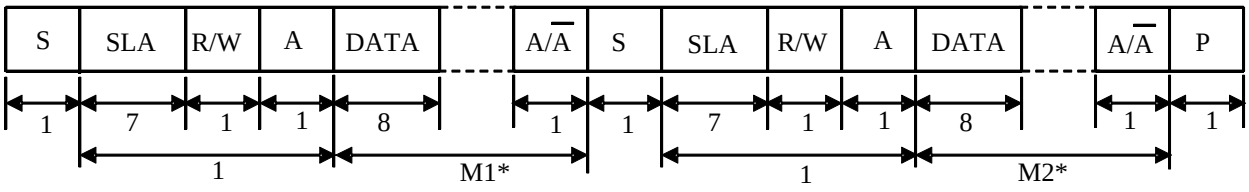
The I²C bus interface has 2 types of data format (Figure 19-6). The first byte after the start condition is always configured with 8 bits. Figure 19-7 shows the I²C bus timing.

(a) Transfer format



*N: Number of bytes transferred

(b) Transfer format (Retransmitting a start condition)



*M1, M2: Number of bytes transferred

Figure 19-6. I²C Bus Data Format

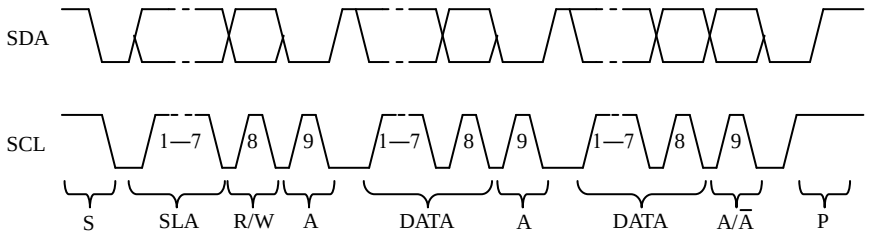


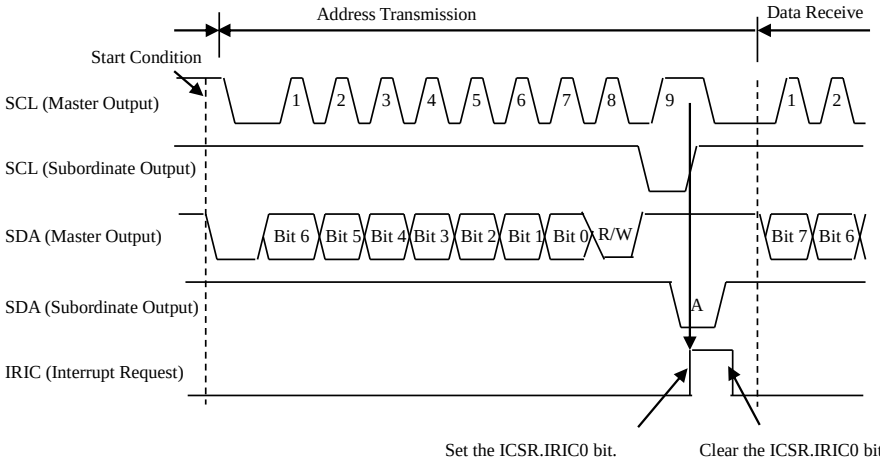
Figure 19-7. I²C Bus Timing

19.4. Subordinate Reception

In the subordinate reception, the master device outputs the subordinate address, transmission data, and clock, whereas the subordinate device transmits the acknowledge. This section describes the procedures and operations for subordinate reception.

(1) Set the following bits to 1: ICCR.ICE, ICCR.ICE0, ICCR.IEIC3, and ICCR.IEIC5. To perform the subordinate transmission, set the ICCR.IEIC2 bit to 1.
(2) Set the subordinate address to the ICSAR.SVA bits. When using the GCA, set the ICCR.GCAE bit to 1. When using the SAA, set the ICSAIR.SAAEN bit to 1 and the alert address to the ICSAA register. The GCA and SAA are judged in the same way as the ICSAR.SVA bits by enabling the GCA and SAA.
(3) The master device transmits the subordinate address and the write command after transmitting the start condition.
(4) The subordinate device compares the received subordinate address with the ICSAR.SVA bits.
(5) When the subordinate address matches the ICSAR.SVA bits, the subordinate device stores the write command in the ICSAR CMD bit, and automatically transmits the acknowledge to the master device.
(6) The ICSR.IRIC0 bit is simultaneously set to 1 with the transmission of the acknowledge, and an interrupt is generated.
(7) Clear the ICSR.IRIC0 bit. When using the GCA or SAA, check the ICSAIR.SAI0 to ICSAIR.SAI2 bits. In addition, check the ICSAR.CMD bit whether the write command or the read command is set. When the read command is set, start from step (8) in Section 19.5.
(8) The data is received from the master device.
(9) The subordinate device stores the reception data to the ICRXDR register, and sets the ICSR.IRIC3 bit to 1. Then, the interrupt is generated.
(10) To continue the reception, set 1 to the ICCMD.ACK bit. Then, an ACK response is output to the I^2C bus. Clear the ICSR.IRIC3 bit, and return to step (8).
(11) Otherwise, set the ICCMD.NACK bit to 1. Then, a NACK response is output to the I^2C bus. Clear the ICSR.IRIC3 bit.
(12) When the stop condition is detected, the ICSR.IRIC5 bit is set to 1, and an interrupt is generated. Clear the ICSR.IRIC5 bit in the interrupt service routine.

a) Address Transmission (Master) −> Data Receive (Subordinate)



b) Data Receive (Subordinate)

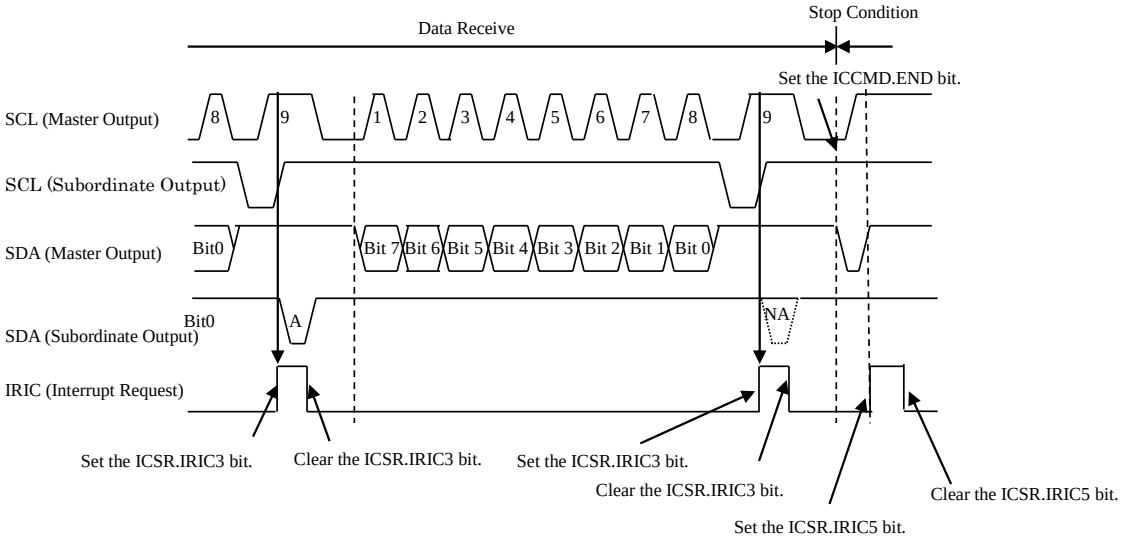


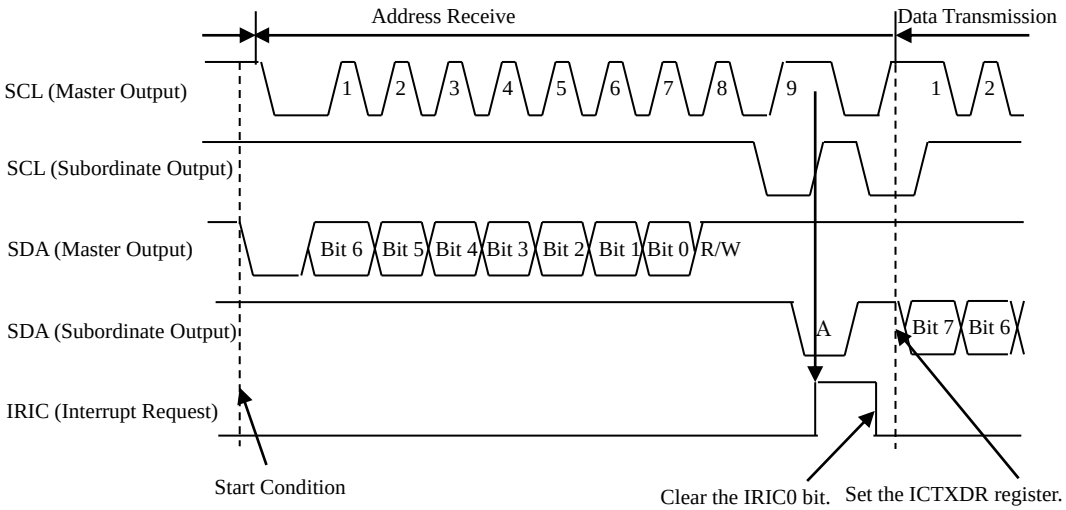
Figure 19-8. Subordinate Reception Timing

19.5. Subordinate Transmission

In the subordinate transmission, the subordinate device outputs transmission data, whereas the master device transmits the subordinate address and the acknowledge. This section describes the procedures and operations for subordinate transmission.

(1) Set the following bits to 1: ICCR.ICE, ICCR.IEIC0, ICCR.IEIC2, and ICCR.IEIC. To perform the subordinate reception, set the ICCR.IEIC3 bit to 1.
(2) Set the subordinate address to the ICSAR.SVA bits. When using the SAA, set the ICSAIR.SAAEN bit to 1 and the alert address to the ICSAA register.
(3) The master device transmits the subordinate address and read command after transmitting the start condition.
(4) The subordinate device compares the received subordinate address with the ICSAR.SVA bits.
(5) When the subordinate address matches the ICSAR.SVA bits, the subordinate device stores the read command in the ICSAR.CMD bit, and automatically transmits the acknowledge to the master device.
(6) The ICSR.IRIC0 bit is simultaneously set to 1 with the transmission of the acknowledge, and an interrupt is generated.
(7) Clear the ICSR.IRIC0 bit. When using the SAA, check the ICSAIR.SAI0 to ICSAIR.SAI2 bits. In addition, check the ICSAR.CMD bit whether the write command or the read command is set. When the write command is set, start from step (8) in Section 19.4.
(8) When the subordinate device sets the transmission data to the ICTXDR register, the transmission is started.
(9) When the subordinate device receives the ACK or the NACK from the master device after the transmission completes, the ICSR.IRIC2 bit is set to 1. Then, the interrupt is generated.
(10) Check the ICSR.RXACK bit whether the received acknowledge bit is the ACK or the NACK.
(11) Clear the ICSR.IRIC2 bit.
(12) To continue the transmission, go back to step (8).
(13) When the stop condition is detected, the ICSR.IRIC5 bit is set to 1, and an interrupt is generated. Clear the ICSR.IRIC5 bit in the interrupt service routine.

a) Address Receive −> Data Transmission



b) Data Transmission

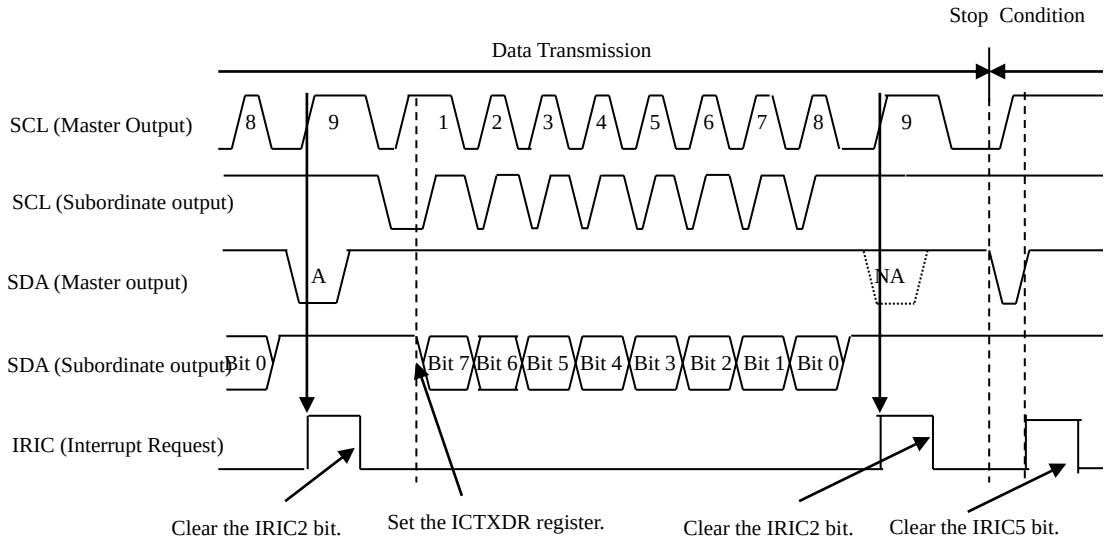


Figure 19-9. Subordinate Transmission Timing

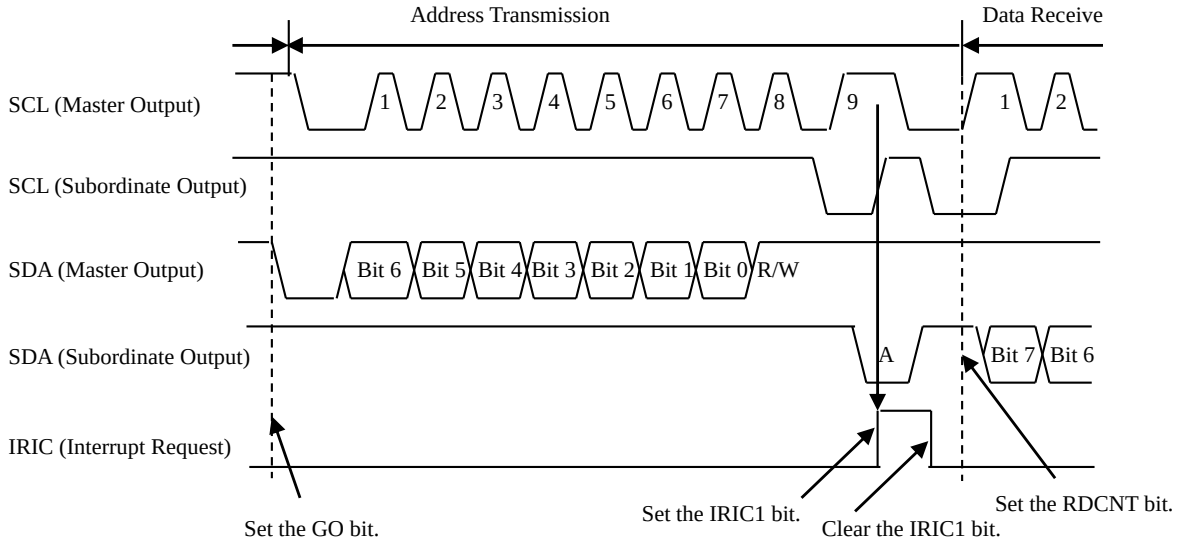
19.6. Master Reception

In the master reception, the subordinate device outputs transmission data, whereas the master device transmits the subordinate address and the acknowledge. This section describes the procedures and operations for master reception.

(1) Set the following bits to 1: ICCR.ICE, ICCR.IEIC1, ICCR.IEIC3, ICCR.IEIC5.
(2) The master device sets the subordinate address to the ICTSAR.ADR bits, and sets the ICTSAR.CMD bit to 1.
(3) For the master device to generate the start condition, set the ICCMD.GO bit to 1. When the start condition is detected, the ICCMD.GO bit is automatically cleared.
(4) The master device transmits the subordinate address and read command after transmitting the start condition.
(5) At the same time that the master device receives the ACK or the NACK from the subordinate device, the ICSR.IRIC1 bit is set to 1, and an interrupt is generated. Clear the ICSR.IRIC1 bit.
(6) Whether the ACK or the NACK has been received can be confirmed by the ICSR.RXACK bit. When the ACK is received, the master device keeps the SCL at the low level until 1 is written to one of the ICCMD register bits. When the NACK is received, the master device automatically generates the stop condition. When the stop condition generation is completed, the ICSR.IRIC5 bit is set to 1, and an interrupt is generated. Clear the ICSR.IRIC5 bit in the interrupt service routine, and go to step (12).
(7) Set the ICCMD.RDCNT bit to 1. The SCL is released, and then the data reception is started.
(8) When the master device receives the data of the 8th bit, the ICSR.IRIC3 bit is set to 1, and an interrupt is generated. The master device keeps the SCL at the low level until 1 is written to one of the ICCMD register bits.
(9) Clear the ICSR.IRIC3 bit.
(10) Read the reception data from the ICRXDR register.
(11) Make settings as follows according to the next operation executed:
- When continuing the reception:
  Set the ICCMD.RDCNT bit to 1. The ACK is automatically transmitted to the subordinate device, and the operations of steps (8) to (11) are repeated.
- When stopping the reception:
  Set the ICCMD.END bit to 1. The master device automatically transmits the NACK to the subordinate device, and generates the stop condition. When the stop condition generation is completed, the ICSR.IRIC5 bit is set to 1, and an interrupt is generated. Clear the ICSR.IRIC5 bit in the interrupt service routine.

(12) When starting the next operation, check that the ICSR.BBSY bit has detected the stop condition. Then, go back to step (2).

a) Address Transmission −> Data Receive



b) Data Receive

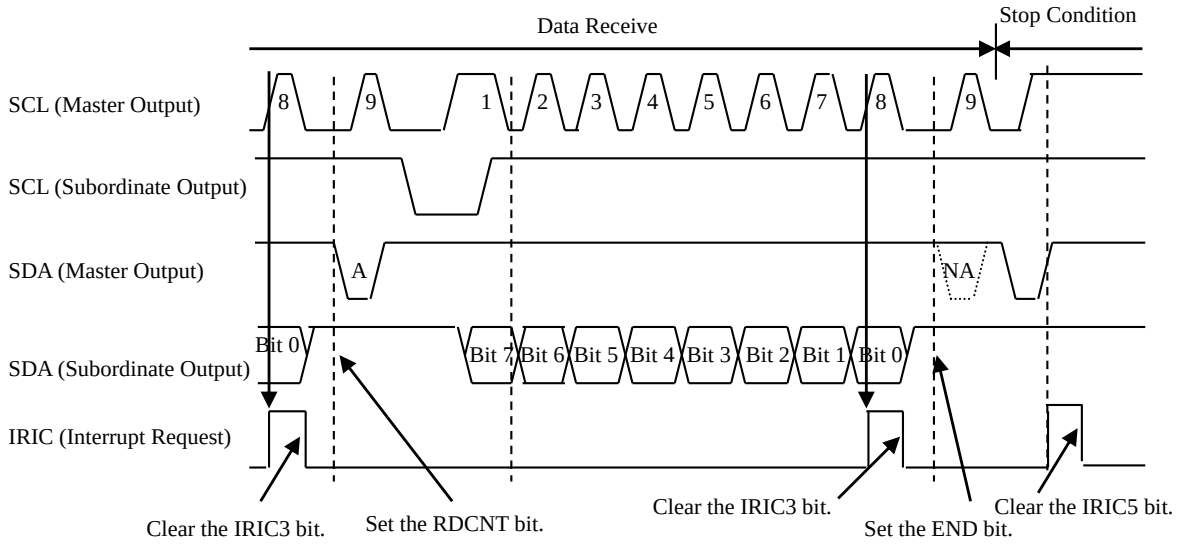


Figure 19-10. Master Reception Timing

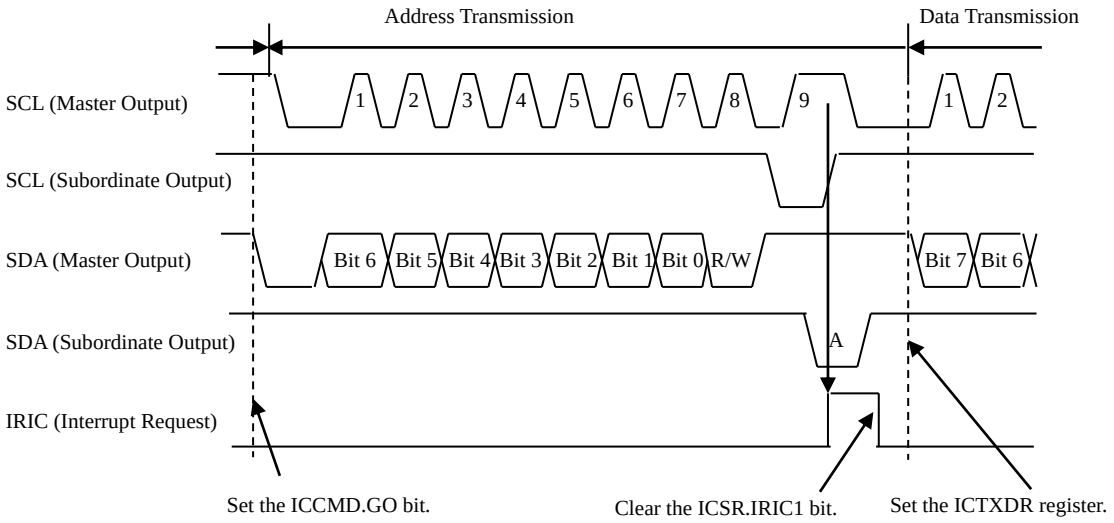
19.7. Master Transmission

In the master transmission, the master device outputs transmission data and transmission clock, whereas the subordinate device transmits the acknowledge. This section describes the procedures and operations for master transmission.

(1) Set the following bits to 1: ICCR.ICE, ICCR.IEIC1, ICCR.IEIC2, and ICCR.IEIC5.
(2) Set the subordinate address to the ICTSAR.ADR bits, and set the ICTSAR.CMD bit to 0.
(3) To generate the start condition, set the ICCMD.GO bit to 1. When the start condition is detected, the ICCMD.GO bit is automatically cleared.
(4) The master device transmits the subordinate address and the write command after transmitting the start condition.
(5) At the same time that the master device receives the ACK or the NACK from the subordinate device, the ICSR.IRIC1 bit is set to 1, and an interrupt is generated. Clear the ICSR.IRIC1 bit.
(6) Whether the ACK or the NACK has been received can be confirmed by the ICSR.RXACK bit. When the ACK is received, the master device keeps the SCL at the low level until 1 is written to one of the ICCMD register bits or transmission data is set to the ICTXDR register. When the NACK is received, the master device automatically generates the stop condition. When the stop condition generation is completed, the ICSR.IRIC5 bit is set to 1, and an interrupt is generated. Clear the ICSR.IRIC5 bit in the interrupt service routine, and go to step (12).
(7) Set the transmission data to the ICTXDR register.
(8) After the master device completes to transmit the data of the 8th bit, the master device waits for a response (ACK or NACK) from the subordinate device.
(9) At the same time that the master device receives the ACK or the NACK from the subordinate device, the ICSR.IRIC2 bit is set to 1, and an interrupt is generated. Whether the ACK or the NACK has been received can be confirmed by the ICSR.RXACK bit. The master device keeps the SCL at the low level until 1 is written to one of the ICCMD register bits or transmission data is set to the ICTXDR register.
(10) Clear the ICSR.IRIC2 bit.
(11) Make settings as follows according to the next operation executed:
- When continuing the transmission:
  Set the transmission data to the ICTXDR register. The operations of steps (8) to (11) are repeated.
- When generating the start condition again to switch to the master reception:
  When generating the start condition again without generating the stop condition to switch to the master reception, start from step (2) in Section 19.6. Be sure to preset the ICCR.IEIC3 bit to 1.
- When stopping the transmission:
  For the master device to generate the stop condition, set the ICCMD.END bit to 1. When the stop condition generation is completed, the ICSR.IRIC5 bit is set to 1, and an interrupt is generated. Clear the ICSR.IRIC5 bit in the interrupt service routine.

(12) When starting the next operation, check that the ICSR.BBSY bit has detected the stop condition. Then, go back to step (2).

a) Address Transmission −> Data Transmission



b) Data Transmission

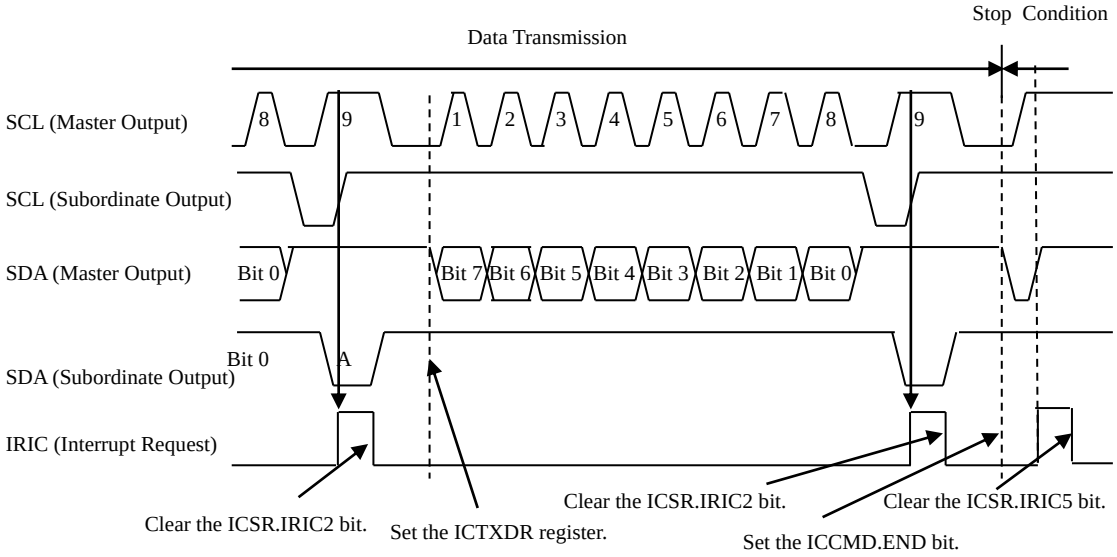


Figure 19-11. Master Transmission Timing

19.8. Subordinate Alert Address (SAA), General Call Address (GCA), and Reception Address Indicator

This I²C device has the following registers to set the subordinate address: the general call address (GCA), which is fixed to 0x00, the ICSAR register, which can set an arbitrary address, and the subordinate alert address (SAA) register.

In the subordinate mode, the following operation when the subordinate address matches the SAA or the GCA is the same as the operation when the subordinate address matches the ICSAR register. When the same subordinate address is set to the ICSAR register and the SAA, the priority is ICSAR > SAA.

When the ICSAR.SVA bits = 0, the IC does not judge whether or not the subordinate address matches the ICSAR.SVA bits. Thus, the GCA and the ICSAR.SVA bits will never match with the subordinate address at the same time as when the GCA is enabled.

When the ICSAA.SAA bits = 0, the IC does not judge whether or not the subordinate address matches the ICSAA.SAA bits. Thus, the SAA and the ICSAA.SAA bits will never match with the subordinate address at the same time as when the SAA is enabled.

When not using the SAA, set the ICSAIR.SAAEN bit to 0. When not using the GCA, set the ICCR.GCAE bit to 0. The lower 3 bits of the ICSAIR register indicate the last address that was accessed as the subordinate device. Whether the command for the subordinate address is writing or reading can be confirmed by the ICSAR.CMD bit.

19.9. Noise Filter

The states of the SCL and SDA pins are stored to inside through the noise filter circuit. Figure 19-12 shows the block diagram of the noise filter circuit.

The noise filter is configured with 2 flip-flop circuits connected in series and a matching detection circuit. The SCL and SDA signals are sampled by the CLKFAST. When the output levels of the 2 flip-flops match, the level is output from the matching detection circuit. Otherwise, the output of the matching detection circuit is kept at the previous value.

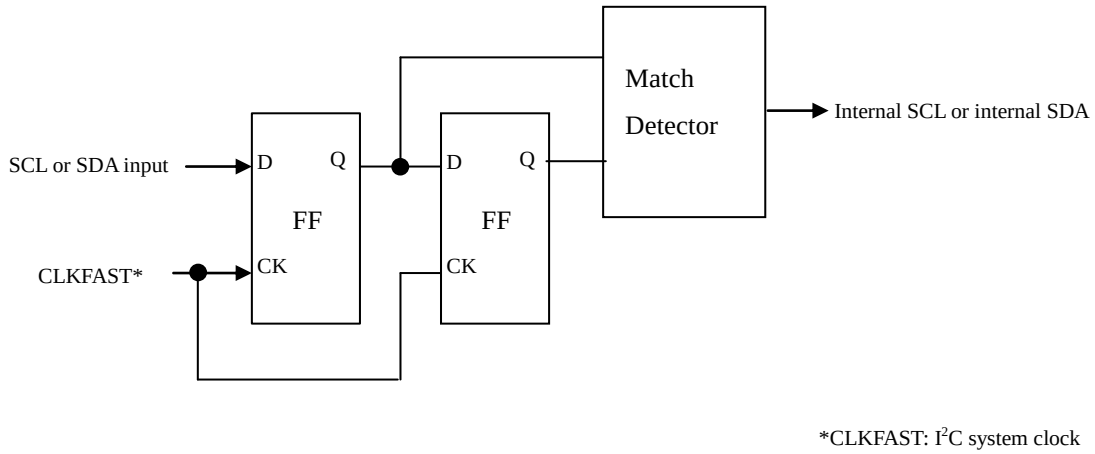


Figure 19-12. Noise Filter Block Diagram

20. UART

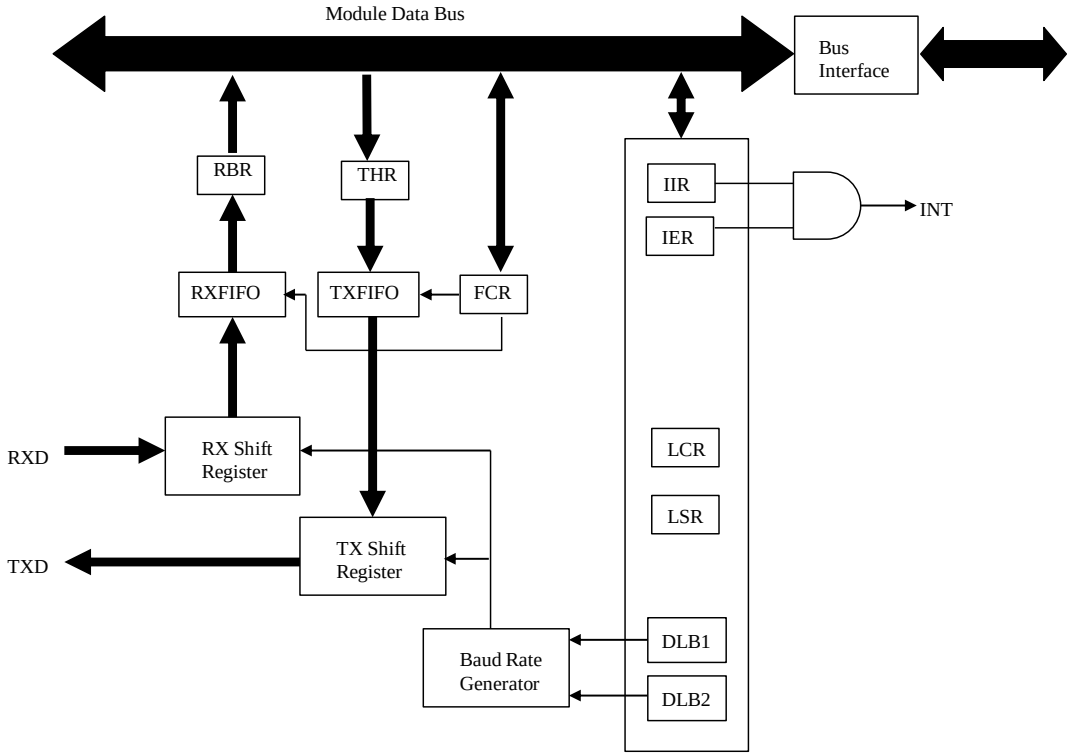
The LSI does not support a modem pin control.

20.1. Overview

The universal asynchronous receiver transmitter (UART) supports an asynchronous serial input/output mode. The UART generates a transfer clock, and has a timer for a baud rate generation.

Table 20-1. UART Functional Descriptions

Item	Description
Character Format	Data length: 5 bits, 6 bits, 7 bits, or 8 bits Start bit: 1 bit Parity bit: Odd, even, or none Stop bit: 1 bit or 2 bits
Baud Rate	The baud rate is determined by the following equation, which is based on a frequency divider register setting. $\text{Baud rate (bps)} = \frac{\text{CLKUART}}{16 \times \text{n}}$, where: CLKUART is CLKFAST, and n is setting value in the DLB1 and DLB2 registers.
FIFO	TX (transmitter) FIFO: 16 bytes RX (receiver) FIFO: 16 bytes
TX Interrupt	Transmission buffer empty Transfer completion
RX Interrupt	Reception buffer full Parity error, overrun error, or framing error



RXFIFO: Receiver FIFO
TXFIFO: Transmitter FIFO

RXD: Reception data
TXD: Transmission data

Figure 20-1. UART Block Diagram

20.2. External Connection Pins

Pin Name	Input/Output	Description
TXD	Output	Transmission data
RXD	Input	Reception data

20.3. Register Descriptions

The clock frequency divider registers that are listed in Table 20-3 are 16 bits, and are mapped on the same addresses as listed in Table 20-2. While the LCR.DLAB bit is set to 1, the clock frequency divider registers (the DLAB1 and DLAB2 registers) can be accessed. In this state, the RBR, THR, and IER registers that are mapped on the same address cannot be accessed.

Table 20-2. List of Control Registers

Symbol	Name	Address	Initial Value
RBR	Receiver Buffer Register	0xFC80	Undefined
THR	Transmitter Holding Register	0xFC80	Undefined
IER	Interrupt Enable Register	0xFC81	0x00
IIR	Interrupt Identification Register	0xFC82	0xC1
FCR	FIFO Control Register	0xFC82	0xC0
LCR	Line Control Register	0xFC83	0x03
LSR	Line Status Register	0xFC85	0x60

Table 20-3. List of Clock Frequency Divider Registers

Symbol	Name	Address	Initial Value
DLB1	Divisor Latch Byte1	0xFC80	0x00
DLB2	Divisor Latch Byte2	0xFC81	0x00

20.3.1. RBR (Receiver Buffer Register)/THR (Transmitter Holding Register)

Register	RBR	Receiver Buffer Register	Address	0xFC80

Bit	Bit Name	R/W	Initial	Description	Remarks
7	RBR	R	Undefined	Receiver FIFO output	
6		R	Undefined		
5		R	Undefined		
4		R	Undefined		
3		R	Undefined		
2		R	Undefined		
1		R	Undefined		
0		R	Undefined		

Register	THR	Transmitter Holding Register	Address	0xFC80

Bit	Bit Name	R/W	Initial	Description	Remarks
7	THR	W	Undefined	Transmitter FIFO input	
6		W	Undefined		
5		W	Undefined		
4		W	Undefined		
3		W	Undefined		
2		W	Undefined		
1		W	Undefined		
0		W	Undefined		

20.3.2. IER (Interrupt Enable Register)

The UART interrupt is enabled or disabled by the IER register setting.

Register	IER	Interrupt Enable Register	Address	0xFC81

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
6	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
5	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
4	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
3	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
2	IER2	R/W	0	Reception status interrupt enable 0: Reception status interrupt is disabled 1: Reception status interrupt is enabled	
1	IER1	R/W	0	Empty interrupt enable of the transmitter holding register 0: Transmitter holding register empty interrupt is disabled 1:Transmitter holding register empty interrupt is enabled	
0	IER0	R/W	0	Reception data interrupt enable 0: Reception data interrupt is disabled 1: Reception data interrupt is enabled	

20.3.3. IIR (Interrupt Identification Register)

The IIR register indicates the interrupt source with the highest priority in interrupts currently waiting for processing. When the waiting interrupt processing exists, the NOPEND bit is 0. When the waiting interrupt processing does not exist, the NOPEND bit is 1.

Register	IIR		Interrupt Identification Register	Address	0xFC82
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	1	The read value is 1. The write value must always be 1.	
6	Reserved	R	1	The read value is 1. The write value must always be 1.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	1	The read value is 0. The write value must always be 0.	
3	IIR3	R	0	Setting of Interrupt type (3) For the interrupt type, see Table 20-4.	
2	IIR2	R	0	Setting of Interrupt type (2) For the interrupt type, see Table 20-4.	
1	IIR1	R	0	Setting of Interrupt type (1) For the interrupt type, see Table 20-4.	
0	NOPEND	R	1	The bit indicates whether a pending interrupt exists or not exist 0: Pending interrupt exists 1: No pending interrupt	

The following table shows the relationship between each bit and interrupt, and the priority.

Table 20-4. Relationship between Each Bit and Interrupt, and the Priority

IIR3	IIR2	IIR1	Priority	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	1	1	1 (Highest)	Reception communication status	Break interrupt, or parity, overrun, and framing errors.	Communication status register is read.
0	1	0	2	Completion of reception data acquisition	Reaches the FIFO trigger level.	The data in the FIFO is less than the trigger level.
1	1	0	2	Time out	One of the following state: - Data is not received to the FIFO although one or more data exist in the FIFO. - The FIFO is not read during 4 data transmission.	The data is read from the FIFO (reception buffer register).
0	0	1	3 (Lowest)	Transmission buffer empty	Transmission buffer empty	Writing in the transmission buffer or reading the IIR register.

20.3.4. FCR (FIFO Control Register)

The number of bytes of the receiver FIFO that generates a reception data interrupt is defined by the FCR register. The FIFO can be cleared by the FCR register.

Register	FCR	FIFO Control Register	Address	0xFC82

Bit	Bit Name	R/W	Initial	Description	Remarks
7	FTL	W	1	Setting of the number of bytes of the receiver FIFO that generates a reception data interrupt 00: 1 byte 01: 4 bytes 10: 8 bytes 11: 14 bytes	
6		W	1		
5	Reserved	W	0	The read value is 0. The write value must always be 0.	
4	Reserved	W	0	The read value is 0. The write value must always be 0.	
3	Reserved	W	0	The read value is 0. The write value must always be 0.	
2	TFCLR	C	0	When 1 is written to the bit, the transmitter FIFO is cleared (TXFIFO = 0).	
1	RFCLR	C	0	When 1 is written to the bit, the receiver FIFO is cleared (RXFIFO = 0).	
0	Reserved	W	0	The read value is 0. The write value must always be 0.	

20.3.5. LCR (Line Control Register)

The asynchronous data communication method is determined by the LCR register. The LCR register accesses the frequency divider register to define a baud rate.

Register	LCR	Line Control Register	Address	0xFC83

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DLAB	R/W	0	The transmission/reception register and the frequency divider register for the baud rate setting are mapped in the same address (0xFC80). The bit determines which address is accessed. 0: Access the transmission/reception register 1: Access the frequency divider register	
6	BRK	R/W	0	Break state setting 0: Break state is disabled 1: TXD pin is fixed to 0 (break state)	
5	STICK	R/W	0	Stick parity setting 0: Stick parity is disabled 1: The operation in STICK = 1 depends on the states of the PARE and EVPAR bits, and is as follows: - When PARE = 1 and EVPAR = 1: The parity bit is cleared in a transmission. The parity bit is interpreted as 0 in a reception. - When PARE = 1 and EVPAR = 0: The parity bit is set to 1 in a transmission. The parity bit is interpreted as 1 in a reception.	
4	EVPAR	R/W	0	Even parity setting 0: Odd parity The parity bit is transmitted so that the number of 1 bits among the data and parity bits becomes odd number. The reception data is checked whether the number of 1 bits is odd. If the number of "1" in the reception data is even, the parity is set to 1. 1: Even parity The parity bit is transmitted so that the number of 1 bits among the data and parity bits becomes even number. The reception data is checked whether the number of 1 bits is even.	
3	PARE	R/W	0	Parity enable 0: Parity is disabled 1: Parity is enabled	
2	NSTP	R/W	0	Setting of the number of stop bits 0: 1 stop bit 1: 2 stop bits when the character length is selected 6 bits, 7 bits, or 8 bits 1.5 stop bits when the character length is selected 5 bits. The first stop bit is always checked when the data is received.	
1	NBCHAR	R/W	1	Setting of the number of bits for each character 00: 5 bits 01: 6 bits 10: 7 bits 11: 8 bits	
0		R/W	1		

20.3.6. LSR (Line Status Register)

Register	LSR	Line Status Register	Address	0xFC85

Bit	Bit Name	R/W	Initial	Description	Remarks
7	RXERR	R	0	Transmission error indicator 0: No RXFIFO error 1: One or more parity error, overrun error, framing error, brake interrupt exist in RXFIFO When the LSR register is read, the bit is cleared.	
6	TXEMP	R	1	Transmission empty indicator 0: Not empty 1: TXFIFO and TX shift register are empty When data is written to the TXFIFO, the bit is cleared.	
5	TFEMP	R	1	TXFIFO empty 0: TXFIFO is not empty 1: TXFIFO is empty When the TXFIFO is empty, the TXFIFO empty interrupt is generated. When data is written to the TXFIFO, the bit is cleared.	
4	BRKI	R	0	Break interrupt (BI) indicator 0: The latest reception data is not in break state. 1: The latest reception data is in break state When the data that is received during receiving of one character string (sum of start bit, data, parity, and stop bit) is all zero, the UART interprets as a break state is received. When the UART receives the break state, the UART executes the following processing. - Stores 0x00 in the RXFIFO - Generates the reception communication status interrupt - Waits for the next start bit The bit is cleared by reading the LSR register.	
3	FERI	R	0	Framing error (FE) indicator 0: No framing error in the latest reception data 1: A framing error exists in the latest reception data When a stop bit is not detected, the UART interprets as a framing error is received. When the UART receives the framing error, the UART generates the reception communication status interrupt. The bit is cleared by reading the LSR register.	
2	PERI	R	0	Parity error (PE) indicator 0: No parity error in the latest reception data 1: A parity error exists in the latest reception data When the parity error is detected, the UART generates the reception communication status interrupt. The bit is cleared by reading the LSR register.	

Register	LSR	Line Status Register	Address	0xFC85

Bit	Bit Name	R/W	Initial	Description	Remarks
1	OERI	R	0	Overrun error (OE) indicator 0: RXFIFO is not in overrun state 1: An overrun error occurs in RXFIFO When the next data is received even though the RXFIFO is full, the UART interprets as the RXFIFO is in an overrun error state. When the overrun error is detected, the UART generates the reception communication status interrupt. The bit is cleared by reading the LSR register.	
0	DRDYI	R	0	Data ready (DR) indicator 0: No character in RXFIFO 1: One or more data are received, and are stored in RXFIFO The bit is cleared by reading all the data in the RXFIFO.	

20.3.7. DLB1/2 (Divisor Latch Byte1/2)

Register	DLB1		Divisor Latch Byte1	Address	0xFC80
Bit	Bit Name	R/W	Initial	Description	Remarks
7	DLB1	R/W	0	LSB of the frequency divider register	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

Register	DLB2		Divisor Latch Byte2	Address	0xFC81
Bit	Bit Name	R/W	Initial	Description	Remarks
7	DLB2	R/W	0	MSB of the frequency divider register	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

To access a frequency divider register, set 1 to the LCR.DLAB bit. In the initial setting of the UART, the frequency divider value is set to the DLB1 and DLB2 registers after 1 is written to the LCR.DLAB bit. The frequency divider value is 16 bits (2 bytes). When the DLB1 register is written, the internal counter in the baud rate generator operates. Thus, it is required to write the DLB2 and DLB1 registers in that order when setting the frequency divider.

The initial values of the DLB1 and DLB2 registers are 0, which disable all serial input/output operations. After the frequency divider value is set, to access the RBR and THR registers, set the LCR.DLAB bit to 0.

The baud rate is determined by the following equation based on a frequency divider value.

$$\text{Baud rate (bps)} = \frac{\text{CLKUART}}{16 \times \text{n}},$$

where:
CLKUART is CLKFAST, and
n is setting value in the DLB1 and DLB2 registers.

Table 20-5 shows a relationship between a baud rate and a frequency divider value as an example.

20.3.8. Baud Rate

Table 20-5 shows the example of the frequency divider values in decimal to the baud rate when some crystals are used. The baud rate is calculated by equation in Section 20.3.7. The baud rate accuracy depends on the characteristics of an oscillator. Note that the error values shown in Table 20-5 are the result obtained by the calculation and are not guaranteed.

Table 20-5. Example of Frequency Divider Values (CLKFAST = 60MHz, 30 MHz, and 12 MHz) for Baud Rate

Baud Rate (bps)	CLKFAST = 60 MHz		CLKFAST = 30 MHz		CLKFAST = 12 MHz	
	Frequency Divider Value in Decimal	Error (%)	Frequency Divider Value in Decimal	Error (%)	Frequency Divider Value in Decimal	Error (%)
2400	1563	0.03	781	0.03	313	0.16
4800	781	0.03	391	0.10	156	0.16
9600	391	0.10	195	0.16	78	0.16
19200	195	0.16	98	0.35	39	0.16
38400	98	0.35	49	0.35	20	2.40
76800	49	0.35	24	1.70	10	2.40
96000	39	0.16	20	2.40	8	2.40
115200	33	1.38	16	1.70	7	7.52
128000	29	1.01	15	2.40	6	2.40
256000	15	2.40	7	4.43	3	2.40
384000	10	2.40	5	2.40	2	2.40
512000	7	4.43	4	9.23	1	31.73
768000	5	2.40	2	18.08	1	2.40
1152000	3	7.84	2	22.88	1	53.60
1536000	2	18.08	1	18.08	-	-

20.4. Operation

The UART is based on the 16550 UART standard chips except a modem pin control.

21. Analog Interconnect

21.1. Overview

Figure 21-1 shows the analog interconnect. The LSI configures the analog interconnect that switches connect between internal analog modules. All switch states in Figure 21-1 are set by user setting of the corresponding register. The configuration examples are as follows:

(1) An ADC input is connected to an external pin directly, or is connected to an external pin via an OPAMP.
(2) An OPAMP can be used for a standalone amplifier or a gain amplifier (×1 or ×4). A comparator is used for standalone, or is connected to an external pin via an OPAMP.

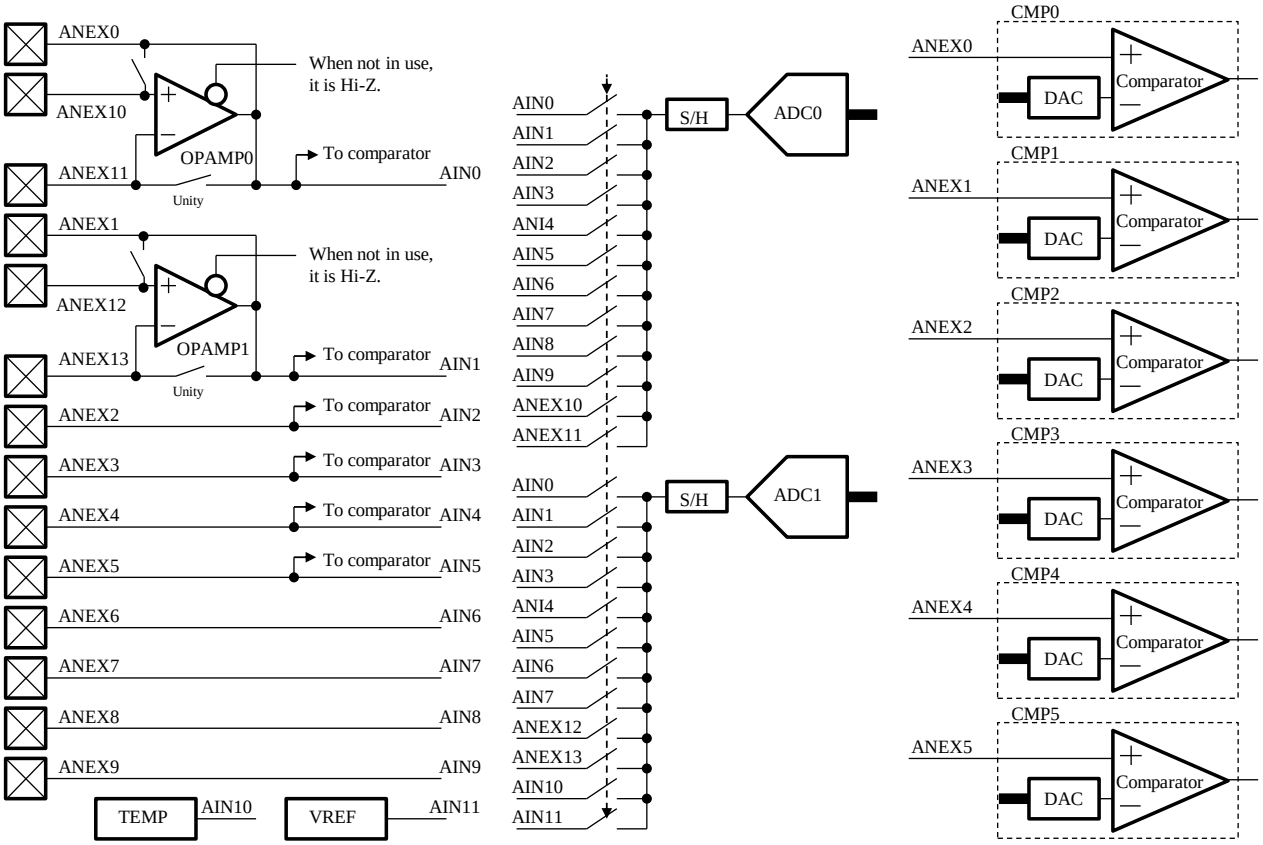


Figure 21-1. Analog Interconnect

22. 12-bit SAR ADC

22.1. Overview

Table 22-1 shows the ADC functional descriptions.

Table 22-1. ADC Functional Descriptions

Item	Description
Resolution	12 bits
Conversion Method	Successive approximation register (SAR)
Conversion Speed	4 MSPS (max.) with clock frequency 60 MHz
Number of Units	2 units
Analog Input	12 channels per unit
Features	Registration of 8 conversion groups (max.) Synchronous operations between the units (ADC0 and ADC1)

The LSI has 2 ADCs of a successive approximation register (SAR) method. The resolution of the ADC is 12 bits. The maximum conversion speed of each ADC is 4 MSPS, where the clock frequency is 60 MHz. The ADC is based on the CLKFAST clock. Each ADC has 12 analog inputs. The specific channel numbers (0 to 11) are assigned to each analog input for identification. These channel numbers are used for the settings. The ADC has 3 registers: a register to store a conversion result in each channel, a register to set the offset value that is added to the conversion result, and a register to set a sampling period. The sampling period is the time that is the ADC internal capacitor is charged with the analog voltage applied the channel. For the sample period, 1 cycle to 256 cycles can be specified. The result converted to the digital value is stored in the registers for storing the conversion result after the offset value set to each channel is added.

After the internal capacitor is charged with the analog voltage, the ADC separates the analog input and the capacitor, and converts the capacitor voltage to the digital value with the SAR method by a binary search. A conversion process of each channel is a series of processing of charging to the capacitor (sample), separating the capacitor (hold), and storing the result, which is obtained by converting the capacitance to the digital value and adding the offset, to the register.

One ADC can convert the analog value of one channel to the digital value in one conversion process. When one ADC needs to convert multiple channels, the ADC performs the channel conversion process continuously. The ADC manages the channels by groups. One ADC has 8 groups. An arbitrary channel can be set for each group. The ADC instructs the group to start conversion instead of specifying the channels directly. The event that is the trigger of the conversion start is set for each group. When receiving the instruction of the conversion start from the CPU or the events, the ADC converts continuously all channels registered in the instructed group.

The following describes the processing of the ADC group and channels. First, the ADC selects one group to be processed from the received trigger. If multiple groups to be the candidates for processing exist, the group with the smallest number is selected. After the group to be processed is determined, the ADC converts all conversion channels set to the group from the smaller channel number in sequence. When the conversion process of the group that is the event output is specified is completed, the ADC event is generated. The ADC event type is determined by the group, and is not related to the channel number. If an interrupt enable is set when the conversion process of all channels is completed, the ADC interrupt signal is generated, and the group processing is completed. The interrupt signal once generated by the ADC is generated continuously unless user clears it. If the unprocessed items such as an unselected group or a trigger received during the processing exist when the group processing is completed, one group to be processed again is determined, and then group processing is started. Otherwise, it enters the idle state until receiving the next activation trigger.

The group is activated by the triggers from the GPIO, comparator, PWM, timer, DSAC, EPU, and CPU. Each group has 3 activation triggers: 2 predetermined CPU triggers (ADT trigger and ADLOOP trigger) and one trigger specified by user (selected from GPIO trigger, comparator trigger, PWM trigger, timer trigger, DSAC trigger, and EPU trigger). Although 2 ADCs operate independently, 2 ADC conversions can be started simultaneously in the Group0. This is valid when 2 analog values require to be converted simultaneously.

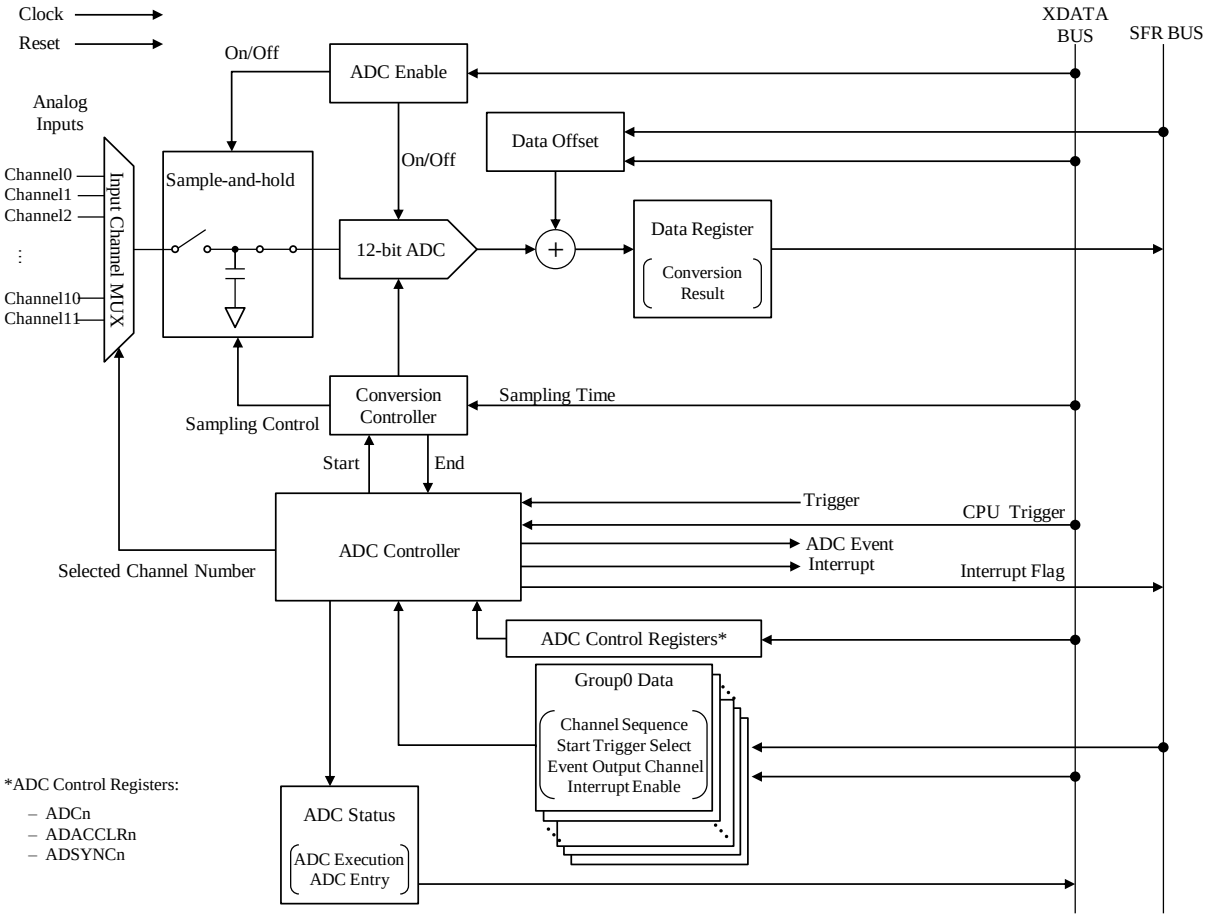


Figure 22-1. 12-bit ADC Block Diagram

22.2. Register Descriptions

The ADC uses 2 registers: the XDATA BUS and SFR BUS registers. Table 22-2 and Table 22-3 show the list of the XDATA BUS and SFR BUS registers, respectively. The letter “n” means the unit number.

Table 22-2. List of XDATA BUS Register

Symbol	Name	Address (Unit0)	Address (Unit1)	Initial Value
ADCn	ADCn Configuration Register	0xF000	0xF080	0x00
ADENn	ADCn Enable Register	0xF001	0xF081	0x00
ADTn	ADCn CPU Trigger Register	0xF003	0xF083	0x00
ADLOOPn	ADCn CPU Loop Trigger Register	0xF008	0xF088	0x00
ADEXEn	ADCn Group Execution Status Register	0xF006	0xF086	0x00
ADENTRYn	ADCn Group Entry Status Register	0xF004	0xF084	0x00
ADIENn	ADCn Group Interrupt Enable Register	0xF007	0xF087	0x00
ADSYNCn	ADCn Synchronous Control Register	0xF005	0xF085	0x00
ADCHSELTESTLn	ADCn Channel Select Test Low Register	0xF00E	0xF08E	0x00
ADCHSELTESTHn	ADCn Channel Select Test High Register	0xF00F	0xF08F	0x00
ADNSMP0n	ADCn Channel0 Sampling Time Register	0xF010	0xF090	0x03
ADNSMP1n	ADCn Channel1 Sampling Time Register	0xF011	0xF091	0x03
ADNSMP2n	ADCn Channel2 Sampling Time Register	0xF012	0xF092	0x03
ADNSMP3n	ADCn Channel3 Sampling Time Register	0xF013	0xF093	0x03
ADNSMP4n	ADCn Channel4 Sampling Time Register	0xF014	0xF094	0x03
ADNSMP5n	ADCn Channel5 Sampling Time Register	0xF015	0xF095	0x03
ADNSMP6n	ADCn Channel6 Sampling Time Register	0xF016	0xF096	0x03
ADNSMP7n	ADCn Channel7 Sampling Time Register	0xF017	0xF097	0x03
ADNSMP8n	ADCn Channel8 Sampling Time Register	0xF018	0xF098	0x03
ADNSMP9n	ADCn Channel9 Sampling Time Register	0xF019	0xF099	0x03
ADNSMPAn	ADCn Channel10 Sampling Time Register	0xF01A	0xF09A	0x03
ADNSMPBn	ADCn Channel11 Sampling Time Register	0xF01B	0xF09B	0x03
ADO0Ln	ADCn Channel0 Data Offset Low Register	0xF020	0xF0A0	0x00
ADO0Hn	ADCn Channel0 Data Offset High Register	0xF021	0xF0A1	0x00
ADO1Ln	ADCn Channel1 Data Offset Low Register	0xF022	0xF0A2	0x00
ADO1Hn	ADCn Channel1 Data Offset High Register	0xF023	0xF0A3	0x00
ADO2Ln	ADCn Channel2 Data Offset Low Register	0xF024	0xF0A4	0x00
ADO2Hn	ADCn Channel2 Data Offset High Register	0xF025	0xF0A5	0x00
ADO3Ln	ADCn Channel3 Data Offset Low Register	0xF026	0xF0A6	0x00
ADO3Hn	ADCn Channel3 Data Offset High Register	0xF027	0xF0A7	0x00
ADO4Ln	ADCn Channel4 Data Offset Low Register	0xF028	0xF0A8	0x00
ADO4Hn	ADCn Channel4 Data Offset High Register	0xF029	0xF0A9	0x00

Symbol	Name	Address (Unit0)	Address (Unit1)	Initial Value
ADO5Ln	ADCn Channel5 Data Offset Low Register	0xF02A	0xF0AA	0x00
ADO5Hn	ADCn Channel5 Data Offset High Register	0xF02B	0xF0AB	0x00
ADO6Ln	ADCn Channel6 Data Offset Low Register	0xF02C	0xF0AC	0x00
ADO6Hn	ADCn Channel6 Data Offset High Register	0xF02D	0xF0AD	0x00
ADO7Ln	ADCn Channel7 Data Offset Low Register	0xF02E	0xF0AE	0x00
ADO7Hn	ADCn Channel7 Data Offset High Register	0xF02F	0xF0AF	0x00
ADO8Ln	ADCn Channel8 Data Offset Low Register	0xF030	0xF0B0	0x00
ADO8Hn	ADCn Channel8 Data Offset High Register	0xF031	0xF0B1	0x00
ADO9Ln	ADCn Channel9 Data Offset Low Register	0xF032	0xF0B2	0x00
ADO9Hn	ADCn Channel9 Data Offset High Register	0xF033	0xF0B3	0x00
ADOALn	ADCn Channel10 Data Offset Low Register	0xF034	0xF0B4	0x00
ADOAHn	ADCn Channel10 Data Offset High Register	0xF035	0xF0B5	0x00
ADOBLn	ADCn Channel11 Data Offset Low Register	0xF036	0xF0B6	0x00
ADOBHn	ADCn Channel11 Data Offset High Register	0xF037	0xF0B7	0x00
ADACCLRn	ADCn Data Read Access Counter Clear Register	0xF002	0xF082	0x00
ADS0Ln	ADCn Group0 Channel Sequence Low Register	0xF040	0xF0C0	0x00
ADS0Hn	ADCn Group0 Channel Sequence High Register	0xF041	0xF0C1	0x00
ADS1Ln	ADCn Group1 Channel Sequence Low Register	0xF048	0xF0C8	0x00
ADS1Hn	ADCn Group1 Channel Sequence High Register	0xF049	0xF0C9	0x00
ADS2Ln	ADCn Group2 Channel Sequence Low Register	0xF050	0xF0D0	0x00
ADS2Hn	ADCn Group2 Channel Sequence High Register	0xF051	0xF0D1	0x00
ADS3Ln	ADCn Group3 Channel Sequence Low Register	0xF058	0xF0D8	0x00
ADS3Hn	ADCn Group3 Channel Sequence High Register	0xF059	0xF0D9	0x00
ADS4Ln	ADCn Group4 Channel Sequence Low Register	0xF060	0xF0E0	0x00
ADS4Hn	ADCn Group4 Channel Sequence High Register	0xF061	0xF0E1	0x00
ADS5Ln	ADCn Group5 Channel Sequence Low Register	0xF068	0xF0E8	0x00
ADS5Hn	ADCn Group5 Channel Sequence High Register	0xF069	0xF0E9	0x00
ADS6Ln	ADCn Group6 Channel Sequence Low Register	0xF070	0xF0F0	0x00
ADS6Hn	ADCn Group6 Channel Sequence High Register	0xF071	0xF0F1	0x00
ADS7Ln	ADCn Group7 Channel Sequence Low Register	0xF078	0xF0F8	0x00
ADS7Hn	ADCn Group7 Channel Sequence High Register	0xF079	0xF0F9	0x00
ADSTSEL0n	ADCn Group0 Start Trigger Select Register	0xF042	0xF0C2	0x00
ADSTSEL1n	ADCn Group1 Start Trigger Select Register	0xF04A	0xF0CA	0x00
ADSTSEL2n	ADCn Group2 Start Trigger Select Register	0xF052	0xF0D2	0x00
ADSTSEL3n	ADCn Group3 Start Trigger Select Register	0xF05A	0xF0DA	0x00
ADSTSEL4n	ADCn Group4 Start Trigger Select Register	0xF062	0xF0E2	0x00
ADSTSEL5n	ADCn Group5 Start Triggesr Select Register	0xF06A	0xF0EA	0x00

Symbol	Name	Address (Unit0)	Address (Unit1)	Initial Value
ADSTSEL6n	ADCn Group6 Start Trigger Select Register	0xF072	0xF0F2	0x00
ADSTSEL7n	ADCn Group7 Start Trigger Select Register	0xF07A	0xF0FA	0x00
ADEVT0Ln	ADCn Group0 Event Output Channel Low Register	0xF044	0xF0C4	0x00
ADEVT0Hn	ADCn Group0 Event Output Channel High Register	0xF045	0xF0C5	0x00
ADEVT1Ln	ADCn Group1 Event Output Channel Low Register	0xF04C	0xF0CC	0x00
ADEVT1Hn	ADCn Group1 Event Output Channel High Register	0xF04D	0xF0CD	0x00
ADEVT2Ln	ADCn Group2 Event Output Channel Low Register	0xF054	0xF0D4	0x00
ADEVT2Hn	ADCn Group2 Event Output Channel High Register	0xF055	0xF0D5	0x00
ADEVT3Ln	ADCn Group3 Event Output Channel Low Register	0xF05C	0xF0DC	0x00
ADEVT3Hn	ADCn Group3 Event Output Channel High Register	0xF05D	0xF0DD	0x00
ADEVT4Ln	ADCn Group4 Event Output Channel Low Register	0xF064	0xF0E4	0x00
ADEVT4Hn	ADCn Group4 Event Output Channel High Register	0xF065	0xF0E5	0x00
ADEVT5Ln	ADCn Group5 Event Output Channel Low Register	0xF06C	0xF0EC	0x00
ADEVT5Hn	ADCn Group5 Event Output Channel High Register	0xF06D	0xF0ED	0x00
ADEVT6Ln	ADCn Group6 Event Output Channel Low Register	0xF074	0xF0F4	0x00
ADEVT6Hn	ADCn Group6 Event Output Channel High Register	0xF075	0xF0F5	0x00
ADEVT7Ln	ADCn Group7 Event Output Channel Low Register	0xF07C	0xF0FC	0x00
ADEVT7Hn	ADCn Group7 Event Output Channel High Register	0xF07D	0xF0FD	0x00

Table 22-3. List of SFR BUS Registers

Symbol (Unit n)	Name	Address (Unit0)	Address (Unit1)	Initial Value
ADO0n	ADCn Channel0 Data Offset Register	0x00	0x01	0x00
ADO1n	ADCn Channel1 Data Offset Register	0x08	0x09	0x00
ADO2n	ADCn Channel2 Data Offset Register	0x10	0x11	0x00
ADO3n	ADCn Channel3 Data Offset Register	0x18	0x19	0x00
ADO4n	ADCn Channel4 Data Offset Register	0x20	0x21	0x00
ADO5n	ADCn Channel5 Data Offset Register	0x28	0x29	0x00
ADO6n	ADCn Channel6 Data Offset Register	0x30	0x31	0x00
ADO7n	ADCn Channel7 Data Offset Register	0x38	0x39	0x00
ADO8n	ADCn Channel8 Data Offset Register	0x40	0x41	0x00
ADO9n	ADCn Channel9 Data Offset Register	0x48	0x49	0x00
ADOAn	ADCn Channel10 Data Offset Register	0x50	0x51	0x00
ADOBn	ADCn Channel11 Data Offset Register	0x58	0x59	0x00
AD0n	ADCn Channel0 Data Register	0x99	0x9A	0x00
AD1n	ADCn Channel1 Data Register	0xA1	0xA2	0x00
AD2n	ADCn Channel2 Data Register	0xA9	0xAA	0x00
AD3n	ADCn Channel3 Data Register	0xB1	0xB2	0x00
AD4n	ADCn Channel4 Data Register	0xB9	0xBA	0x00
AD5n	ADCn Channel5 Data Register	0xC1	0xC2	0x00
AD6n	ADCn Channel6 Data Register	0xC9	0xCA	0x00
AD7n	ADCn Channel7 Data Register	0xD1	0xD2	0x00
AD8n	ADCn Channel8 Data Register	0xD9	0xDA	0x00
AD9n	ADCn Channel9 Data Register	0xE1	0xE2	0x00
ADAn	ADCn Channel10 Data Register	0xE9	0xEA	0x00
ADBn	ADCn Channel11 Data Register	0xF1	0xF2	0x00
ADIFn	ADCn Interrupt Flag Register	0x89	0x8A	0x00

22.2.1. ADCn (ADCn Configuration Register) (n = 0 to 1)

Register	ADC0	ADC0 Configuration Register	Address	0xF000
Register	ADC1	ADC1 Configuration Register	Address	0xF080

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CHCONMODE	R/W	0	Channel connection mode 0: The channel is set to Mode 0 1: The channel is set to Mode 1 For more details on the channel connection modes, see Section 22.4.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	COMPASYNC	R/W	0	Synchronization in 2-cycle mode 0: Comparison-data-creating cycle and comparison-operating cycle are synchronized between other units and ADCs 1: Not synchronized In the case where 2 ADCs are operated simultaneously in the 2-cycle mode, ADC-induced noise will occur during a comparison-data-creating cycle. And the ADC-induced noise may affect ADC operation (i.e., conversion accuracy) during a comparison-operating cycle. To avoid such noise and subsequent negative effect, set the bit to 0.	
3	MODE2	R/W	0	Mode 2 (selection of operation cycle mode) 0: 1-cycle mode Comparison data creation and comparison are operated in one cycle 1: 2-cycle mode Comparison data creation and comparison are operated in 2 different cycles The 2-cycle mode can lower the negative effects of power-supply noise on conversion accuracy. However, its conversion time takes two times longer than the 1-cycle mode.	
2	MODE3	R/W	0	Mode 3 (selection of noise-suppressing operation mode) 0: Noise-suppressing operation mode is disabled 1: Noise-suppressing operation mode The noise-suppressing operation mode can lower the negative effects of power-supply noise on conversion accuracy. Setting the bit has restrictions. Table 22-4 lists the available settings when combined with the ADCn.MODE1 bit. Note that the settings represented “OK” can only be used. Any combination other than those listed cannot obtain proper converted values. In addition, be sure to set the ADC0.MODE3 and ADC1.MODE3 bits to the same value.	

Register	ADC0	ADC0 Configuration Register	Address	0xF000
Register	ADC1	ADC1 Configuration Register	Address	0xF080

Bit	Bit Name	R/W	Initial	Description	Remarks
1	MODE1	R/W	0	Mode 1 (selection of comparison timing mode) 0: Fast comparison-timing mode Speeds up the timing of comparison operation 1: Slow comparison-timing mode Slows down the timing of comparison operation The slow comparison timing mode can reduce the negative effects of power-supply noise on conversion accuracy. Setting the bit has restrictions. Table 22-4 lists the available settings when combined with the ADCn.MODE3 bit. Note that the settings represented "OK" can only be used. Any combination other than those listed cannot obtain proper converted values.	
0	MODE4	R/W	0	Mode 4 (ADC characteristics evaluation mode) Do not set the bit to 1 (proper converted values cannot be obtained).	

Table 22-4. Restricted Setting Conditions for ADCn.MODE1 and ADCn.MODE3

Conditions of CLKFAST and ADC				Setting Conditions of ADCn.MODE1 and ADCn.MODE3		
CLKFAST/ ADC	PLLCFG.REFDIV	LKCFG0.DIV1	LKCFG0.DIV1	ADCn.MODE3	ADCn.MODE1	Setting
60 MHz/ 1-cycle Mode	0	0b11	0	0	0	OK
					1	—
				1	0	—
					1	—
60 MHz/ 2-cycle Mode	0	0b11	1	0	0	OK
					1	—
				1	0	OK
					1	—
≤30 MHz/ 1-cycle Mode	1	—	0	0	0	OK
					1	—
				1	0	OK
					1	OK
	—	Other than 0b11	0	0	0	OK
					1	—
				1	0	OK
					1	OK
≤30 MHz/ 2-cycle Mode	1	—	1	0	0	OK
					1	OK
				1	0	OK
					1	OK
	—	Other than 0b11	1	0	0	OK
					1	OK
				1	0	OK
					1	OK

22.2.2. ADENn (ADCn Enable Register) (n = 0 to 1)

Register	ADEN0	ADC0 Enable Register	Address	0xF001
Register	ADEN1	ADC1 Enable Register	Address	0xF081

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	ADENABLE	R/W	0	ADCn enable 0: ADCn is disabled 1: ADCn is enabled	

22.2.3. ADTn (ADCn CPU Trigger Register) (n = 0 to 1)

Register	ADT0	ADC0 CPU Trigger Register	Address	0xF003
Register	ADT1	ADC1 CPU Trigger Register	Address	0xF083

Bit	Bit Name	R/W	Initial	Description	Remarks
7	TRIGGER7	W	0	CPU trigger of Group7 Write 0: No change Write 1: Group7 is activated	
6	TRIGGER6	W	0	CPU trigger of Group6 Write 0: No change Write: Group6 is activated	
5	TRIGGER5	W	0	CPU trigger of Group5 Write 0: No change Write 1: Group5 is activated	
4	TRIGGER4	W	0	CPU trigger of Group4 Write 0: No change Write 1: Group4 is activated	
3	TRIGGER3	W	0	CPU trigger of Group3 Write 0: No change Write 1: Group3 is activated	
2	TRIGGER2	W	0	CPU trigger of Group2 Write 0: No change Write 1: Group2 is activated	
1	TRIGGER1	W	0	CPU trigger of Group1 Write 0: No change Write 1: Group1 is activated	
0	TRIGGER0	W	0	CPU trigger of Group0 Write 0: No change Write 1: Group0 is activated	

22.2.4. ADLOOPn (ADCn CPU Loop Trigger Register) (n = 0 to 1)

An activation trigger from the CPU is not stopped even if 0 is written to each bit. To stop the activation trigger, set DLOOPn.LOOPm = 0. For details of the ADLOOP trigger, see Section 22.5.6.3.

Register	ADLOOP0	ADC0 CPU Loop Trigger Register	Address	0xF008
Register	ADLOOP1	ADC1 CPU Loop Trigger Register	Address	0xF088

Bit	Bit Name	R/W	Initial	Description	Remarks
7	LOOP7	R/W	0	CPU trigger of Group7 Read 0: ADLOOP trigger has not been issued Read 1: ADLOOP trigger has been issued Write 0: ADLOOP trigger is cancelled Write 1: ADLOOP trigger is issued	
6	LOOP6	R/W	0	CPU trigger of Group6 Read 0: ADLOOP trigger has not been issued Read 1: ADLOOP trigger has been issued Write 0: ADLOOP trigger is cancelled Write 1: ADLOOP trigger is issued	
5	LOOP5	R/W	0	CPU trigger of Group5 Read 0: ADLOOP trigger has not been issued Read 1: ADLOOP trigger has been issued Write 0: ADLOOP trigger is cancelled Write 1: ADLOOP trigger is issued	
4	LOOP4	R/W	0	CPU trigger of Group4 Read 0: ADLOOP trigger has not been issued Read 1: ADLOOP trigger has been issued Write 0: ADLOOP trigger is cancelled Write 1: ADLOOP trigger is issued	
3	LOOP3	R/W	0	CPU trigger of Group3 Read 0: ADLOOP trigger has not been issued Read 1: ADLOOP trigger has been issued Write 0: ADLOOP trigger is cancelled Write 1: ADLOOP trigger is issued	
2	LOOP2	R/W	0	CPU trigger of Group2 Read 0: ADLOOP trigger has not been issued Read 1: ADLOOP trigger has been issued Write 0: ADLOOP trigger is cancelled Write 1: ADLOOP trigger is issued	
1	LOOP1	R/W	0	CPU trigger of Group1 Read 0: ADLOOP trigger has not been issued Read 1: ADLOOP trigger has been issued Write 0: ADLOOP trigger is cancelled Write 1: ADLOOP trigger is issued	
0	LOOP0	R/W	0	CPU trigger of Group0 Read 0: ADLOOP trigger has not been issued Read 1: ADLOOP trigger has been issued Write 0: ADLOOP trigger is cancelled Write 1: ADLOOP trigger is issued	

22.2.5. ADEXEn (ADCn Group Execution Status Register) (n = 0 to 1)

Register	ADEXE0	ADC0 Group Execution Status Register	Address	0xF006
Register	ADEXE1	ADC1 Group Execution Status Register	Address	0xF086

Bit	Bit Name	R/W	Initial	Description	Remarks
7	EXECUTION7	R	0	Execution status of Group7 0: Stopping 1: Executing	
6	EXECUTION6	R	0	Execution status of Group6 0: Stopping 1: Executing	
5	EXECUTION5	R	0	Execution status of Group5 0: Stopping 1: Executing	
4	EXECUTION4	R	0	Execution status of Group4 0: Stopping 1: Executing	
3	EXECUTION3	R	0	Execution status of Group3 0: Stopping 1: Executing	
2	EXECUTION2	R	0	Execution status of Group2 0: Stopping 1: Executing	
1	EXECUTION1	R	0	Execution status of Group1 0: Stopping 1: Executing	
0	EXECUTION0	R	0	Execution status of Group0 0: Stopping 1: Executing	

22.2.6. ADENTRYn (ADCn Group Entry Status Register) (n = 0 to 1)

Register	ADENTRY0	ADC0 Group Entry Status Register	Address	0xF004
Register	ADENTRY1	ADC1 Group Entry Status Register	Address	0xF084

Bit	Bit Name	R/W	Initial	Description	Remarks
7	ENTRY7	R	0	Entry status of Group7 0: Unentered 1: Entered	
6	ENTRY6	R	0	Entry status of Group6 0: Unentered 1: Entered	
5	ENTRY5	R	0	Entry status of Group5 0: Unentered 1: Entered	
4	ENTRY4	R	0	Entry status of Group4 0: Unentered 1: Entered	
3	ENTRY3	R	0	Entry status of Group3 0: Unentered 1: Entered	
2	ENTRY2	R	0	Entry status of Group3 0: Unentered 1: Entered	
1	ENTRY1	R	0	Entry status of Group1 0: Unentered 1: Entered	
0	ENTRY0	R	0	Entry status of Group0 0: Unentered 1: Entered	

22.2.7. ADIENn (ADCn Group Interrupt Enable Register) (n = 0 to 1)

Register	ADIEN0	ADC0 Group Interrupt Enable Register	Address	0xF007
Register	ADIEN1	ADC1 Group Interrupt Enable Register	Address	0xF087

Bit	Bit Name	R/W	Initial	Description	Remarks
7	IEN7	R/W	0	Enabling the interrupt signal of Group7 0: Interrupt signal is disabled 1: Interrupt signal is enabled	
6	IEN6	R/W	0	Enabling the interrupt signal of Group6 0: Interrupt signal is disabled 1: Interrupt signal is enabled	
5	IEN5	R/W	0	Enabling the interrupt signal of Group5 0: Interrupt signal is disabled 1: Interrupt signal is enabled	
4	IEN4	R/W	0	Enabling the interrupt signal of Group4 0: Interrupt signal is disabled 1: Interrupt signal is enabled	
3	IEN3	R/W	0	Enabling the interrupt signal of Group3 0: Interrupt signal is disabled 1: Interrupt signal is enabled	
2	IEN2	R/W	0	Enabling the interrupt signal of Group2 0: Interrupt signal is disabled 1: Interrupt signal is enabled	
1	IEN1	R/W	0	Enabling the interrupt signal of Group1 0: Interrupt signal is disabled 1: Interrupt signal is enabled	
0	IEN0	R/W	0	Enabling the interrupt signal of Group0 0: Interrupt signal is disabled 1: Interrupt signal is enabled	

22.2.8. ADSYNCn (ADCn Synchronous Control Register) (n = 0 to 1)

Register	ADSYNC0	ADC0 Synchronous Control Register	Address	0xF005
Register	ADSYNC1	ADC1 Synchronous Control Register	Address	0xF085

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	SYNCHRONOUS	R/W	0	Synchronous execution 0: Not synchronously executed 1: Synchronously executed	

22.2.9. ADCHSELTESTLn (ADCn Channel Select Test Low Register) (n = 0 to 1)

Register	ADCHSELTESTL0	ADC0 Channel Select Test Low Register	Address	0xF00E
Register	ADCHSELTESTL1	ADC1 Channel Select Test Low Register	Address	0xF08E

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

22.2.10. ADCHSELTESTHn (ADCn Channel Select Test High Register) (n = 0 to 1)

Register	ADCHSELTESTH0	ADC0 Channel Select Test High Register	Address	0xF00F
Register	ADCHSELTESTH1	ADC1 Channel Select Test High Register	Address	0xF08F

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

22.2.11. ADNSMPmn (ADCn Channel m Sampling Time Register) (n = 0 to 1) (m = 0 to 11)

Register	ADNSMP00	ADC0 Channel0 Sampling Time Register	Address	0xF010
Register	ADNSMP01	ADC1 Channel0 Sampling Time Register	Address	0xF090
Register	ADNSMP10	ADC0 Channel1 Sampling Time Register	Address	0xF011
Register	ADNSMP11	ADC1 Channel1 Sampling Time Register	Address	0xF091
Register	ADNSMP20	ADC0 Channel2 Sampling Time Register	Address	0xF012
Register	ADNSMP21	ADC1 Channel2 Sampling Time Register	Address	0xF092
Register	ADNSMP30	ADC0 Channel3 Sampling Time Register	Address	0xF013
Register	ADNSMP31	ADC1 Channel3 Sampling Time Register	Address	0xF093
Register	ADNSMP40	ADC0 Channel4 Sampling Time Register	Address	0xF014
Register	ADNSMP41	ADC1 Channel4 Sampling Time Register	Address	0xF094
Register	ADNSMP50	ADC0 Channel5 Sampling Time Register	Address	0xF015
Register	ADNSMP51	ADC1 Channel5 Sampling Time Register	Address	0xF095
Register	ADNSMP60	ADC0 Channel6 Sampling Time Register	Address	0xF016
Register	ADNSMP61	ADC1 Channel6 Sampling Time Register	Address	0xF096
Register	ADNSMP70	ADC0 Channel7 Sampling Time Register	Address	0xF017
Register	ADNSMP71	ADC1 Channel7 Sampling Time Register	Address	0xF097
Register	ADNSMP80	ADC0 Channel8 Sampling Time Register	Address	0xF018
Register	ADNSMP81	ADC1 Channel8 Sampling Time Register	Address	0xF098
Register	ADNSMP90	ADC0 Channel9 Sampling Time Register	Address	0xF019
Register	ADNSMP91	ADC1 Channel9 Sampling Time Register	Address	0xF099
Register	ADNSMPA0	ADC0 Channel10 Sampling Time Register	Address	0xF01A
Register	ADNSMPA1	ADC1 Channel10 Sampling Time Register	Address	0xF09A
Register	ADNSMPB0	ADC0 Channel11 Sampling Time Register	Address	0xF01B
Register	ADNSMPB1	ADC1 Channel11 Sampling Time Register	Address	0xF09B

Bit	Bit Name	R/W	Initial	Description	Remarks
7	SHTIME	R/W	0	Sampling cycle 00000000: 256 cycles (never becomes 0) 00000001: 1 cycle 00000010: 2 cycles 00000011: 3 cycles 00000100: 4 cycles ⋮ 11111111: 255 cycles The bit specifies a sampling cycle to be performed (1 to 256). For more details, see Section 22.4.1. Do not use the sampling cycles listed below because proper converted values are not obtainable. 00000101: 5 cycles 00100101: 37 cycles 01000101: 69 cycles 01100101: 101 cycles 10000101: 133 cycles 10100101: 165 cycles 11000101: 197 cycles 11100101: 229 cycles	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	1		
0		R/W	1		

22.2.12. ADOmLn (ADCn Channel m Data Offset Low Register) (n = 0 to 1) (m = 0 to 11)

Register	ADO0L0	ADC0 Channel0 Data Offset Low Register	Address	0xF020
Register	ADO0L1	ADC1 Channel0 Data Offset Low Register	Address	0xF0A0
Register	ADO1L0	ADC0 Channel1 Data Offset Low Register	Address	0xF022
Register	ADO1L1	ADC1 Channel1 Data Offset Low Register	Address	0xF0A2
Register	ADO2L0	ADC0 Channel2 Data Offset Low Register	Address	0xF024
Register	ADO2L1	ADC1 Channel2 Data Offset Low Register	Address	0xF0A4
Register	ADO3L0	ADC0 Channel3 Data Offset Low Register	Address	0xF026
Register	ADO3L1	ADC1 Channel3 Data Offset Low Register	Address	0xF0A6
Register	ADO4L0	ADC0 Channel4 Data Offset Low Register	Address	0xF028
Register	ADO4L1	ADC1 Channel4 Data Offset Low Register	Address	0xF0A8
Register	ADO5L0	ADC0 Channel5 Data Offset Low Register	Address	0xF02A
Register	ADO5L1	ADC1 Channel5 Data Offset Low Register	Address	0xF0AA
Register	ADO6L0	ADC0 Channel6 Data Offset Low Register	Address	0xF02C
Register	ADO6L1	ADC1 Channel6 Data Offset Low Register	Address	0xF0AC
Register	ADO7L0	ADC0 Channel7 Data Offset Low Register	Address	0xF02E
Register	ADO7L1	ADC1 Channel7 Data Offset Low Register	Address	0xF0AE
Register	ADO8L0	ADC0 Channel8 Data Offset Low Register	Address	0xF030
Register	ADO8L1	ADC1 Channel8 Data Offset Low Register	Address	0xF0B0
Register	ADO9L0	ADC0 Channel9 Data Offset Low Register	Address	0xF032
Register	ADO9L1	ADC1 Channel9 Data Offset Low Register	Address	0xF0B2
Register	ADOAL0	ADC0 Channel10 Data Offset Low Register	Address	0xF034
Register	ADOAL1	ADC1 Channel10 Data Offset Low Register	Address	0xF0B4
Register	ADOBL0	ADC0 Channel11 Data Offset Low Register	Address	0xF036
Register	ADOBL1	ADC1 Channel11 Data Offset Low Register	Address	0xF0B6

Bit	Bit Name	R/W	Initial	Description	Remarks
7	OFFSET	R/W	0	The lower bits of offset An offset value used for each channel can be set by signed 13-bit values (−4096 to 4096). The ADOmHn.OFFSTSIGN bit defines which sign is to be set.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

22.2.13. ADOmHn (ADCn Channel m Data Offset High Register) (n = 0 to 1) (m = 0 to 11)

Register	ADO0H0	ADC0 Channel0 Data Offset High Register	Address	0xF021
Register	ADO0H1	ADC1 Channel0 Data Offset High Register	Address	0xF0A1
Register	ADO1H0	ADC0 Channel1 Data Offset High Register	Address	0xF023
Register	ADO1H1	ADC1 Channel1 Data Offset High Register	Address	0xF0A3
Register	ADO2H0	ADC0 Channel2 Data Offset High Register	Address	0xF025
Register	ADO2H1	ADC1 Channel2 Data Offset High Register	Address	0xF0A5
Register	ADO3H0	ADC0 Channel3 Data Offset High Register	Address	0xF027
Register	ADO3H1	ADC1 Channel3 Data Offset High Register	Address	0xF0A7
Register	ADO4H0	ADC0 Channel4 Data Offset High Register	Address	0xF029
Register	ADO4H1	ADC1 Channel4 Data Offset High Register	Address	0xF0A9
Register	ADO5H0	ADC0 Channel5 Data Offset High Register	Address	0xF02B
Register	ADO5H1	ADC1 Channel5 Data Offset High Register	Address	0xF0AB
Register	ADO6H0	ADC0 Channel6 Data Offset High Register	Address	0xF02D
Register	ADO6H1	ADC1 Channel6 Data Offset High Register	Address	0xF0AD
Register	ADO7H0	ADC0 Channel7 Data Offset High Register	Address	0xF02F
Register	ADO7H1	ADC1 Channel7 Data Offset High Register	Address	0xF0AF
Register	ADO8H0	ADC0 Channel8 Data Offset High Register	Address	0xF031
Register	ADO8H1	ADC1 Channel8 Data Offset High Register	Address	0xF0B1
Register	ADO9H0	ADC0 Channel9 Data Offset High Register	Address	0xF033
Register	ADO9H1	ADC1 Channel9 Data Offset High Register	Address	0xF0B3
Register	ADOAH0	ADC0 Channel10 Data Offset High Register	Address	0xF035
Register	ADOAH1	ADC1 Channel10 Data Offset High Register	Address	0xF0B5
Register	ADOBH0	ADC0 Channel11 Data Offset High Register	Address	0xF037
Register	ADOBH1	ADC1 Channel11 Data Offset High Register	Address	0xF0B7

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	Sign extension bit Writing 1 to the OFFSTSIGN bit sets the bit to 1.	
6	Reserved	R	0		
5	Reserved	R	0		
4	OFFSTSIGN	R/W	0	Sign bit	
3	OFFSET	R/W	0	The higher bits of offset An offset value used for each channel can be set by signed 13-bit values (−4096 to 4096). The OFFSTSIGN bit defines which sign is to be set.	
2		R/W	0		
1		R/W	0		
0		R/W	0		

22.2.14. ADACCLRn (ADCn Data Read Access Counter Clear Register) (n = 0 to 1)

Register	ADACCLR0	ADC0 Data Read Access Counter Clear Register	Address	0xF002
Register	ADACCLR1	ADC1 Data Read Access Counter Clear Register	Address	0xF082

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	ADACCLR	R/C	0	Clearing the data read access counters of the ADmn and ADOmn registers Read 0: Accessible to the lower 8 bits Read 1: Accessible to the higher 8 bits Write 0: No change Write 1: Clears the access counter, then restarts the first-time data reading of the lower bits	

22.2.15. ADSmLn (ADCn Group m Channel Sequence Low Register) (n = 0 to 1) (m = 0 to 7)

Register	ADS0L0	ADC0 Group0 Channel Sequence Low Register	Address	0xF040
Register	ADS0L1	ADC1 Group0 Channel Sequence Low Register	Address	0xF0C0
Register	ADS1L0	ADC0 Group1 Channel Sequence Low Register	Address	0xF048
Register	ADS1L1	ADC1 Group1 Channel Sequence Low Register	Address	0xF0C8
Register	ADS2L0	ADC0 Group2 Channel Sequence Low Register	Address	0xF050
Register	ADS2L1	ADC1 Group2 Channel Sequence Low Register	Address	0xF0D0
Register	ADS3L0	ADC0 Group3 Channel Sequence Low Register	Address	0xF058
Register	ADS3L1	ADC1 Group3 Channel Sequence Low Register	Address	0xF0D8
Register	ADS4L0	ADC0 Group4 Channel Sequence Low Register	Address	0xF060
Register	ADS4L1	ADC1 Group4 Channel Sequence Low Register	Address	0xF0E0
Register	ADS5L0	ADC0 Group5 Channel Sequence Low Register	Address	0xF068
Register	ADS5L1	ADC1 Group5 Channel Sequence Low Register	Address	0xF0E8
Register	ADS6L0	ADC0 Group6 Channel Sequence Low Register	Address	0xF070
Register	ADS6L1	ADC1 Group6 Channel Sequence Low Register	Address	0xF0F0
Register	ADS7L0	ADC0 Group7 Channel Sequence Low Register	Address	0xF078
Register	ADS7L1	ADC1 Group7 Channel Sequence Low Register	Address	0xF0F8

Bit	Bit Name	R/W	Initial	Description	Remarks
7	ADSCH7	R/W	0	Conversion of Channel7 0: Channel7 is not converted 1: Channel7 is converted	
6	ADSCH6	R/W	0	Conversion of Channel6 0: Channel6 is not converted 1: Channel6 is converted	
5	ADSCH5	R/W	0	Conversion of Channel5 0: Channel5 is not converted 1: Channel5 is converted	
4	ADSCH4	R/W	0	Conversion of Channel4 0: Channel4 is not converted 1: Channel4 is converted	
3	ADSCH3	R/W	0	Conversion of Channel3 0: Channel3 is not converted 1: Channel3 is converted	
2	ADSCH2	R/W	0	Conversion of Channel2 0: Channel2 is not converted 1: Channel2 is converted	
1	ADSCH1	R/W	0	Conversion of Channel1 0: Channel1 is not converted 1: Channel1 is converted	
0	ADSCH0	R/W	0	Conversion of Channel0 0: Channel0 is not converted 1: Channel0 is converted	

22.2.16. ADSmHn (ADCn Group m Channel Sequence High Register) (n = 0 to 1) (m = 0 to 7)

Register	ADS0H0	ADC0 Group0 Channel Sequence High Register	Address	0xF041
Register	ADS0H1	ADC1 Group0 Channel Sequence High Register	Address	0xF0C1
Register	ADS1H0	ADC0 Group1 Channel Sequence High Register	Address	0xF049
Register	ADS1H1	ADC1 Group1 Channel Sequence High Register	Address	0xF0C9
Register	ADS2H0	ADC0 Group2 Channel Sequence High Register	Address	0xF051
Register	ADS2H1	ADC1 Group2 Channel Sequence High Register	Address	0xF0D1
Register	ADS3H0	ADC0 Group3 Channel Sequence High Register	Address	0xF059
Register	ADS3H1	ADC1 Group3 Channel Sequence High Register	Address	0xF0D9
Register	ADS4H0	ADC0 Group4 Channel Sequence High Register	Address	0xF061
Register	ADS4H1	ADC1 Group4 Channel Sequence High Register	Address	0xF0E1
Register	ADS5H0	ADC0 Group5 Channel Sequence High Register	Address	0xF069
Register	ADS5H1	ADC1 Group5 Channel Sequence High Register	Address	0xF0E9
Register	ADS6H0	ADC0 Group6 Channel Sequence High Register	Address	0xF071
Register	ADS6H1	ADC1 Group6 Channel Sequence High Register	Address	0xF0F1
Register	ADS7H0	ADC0 Group7 Channel Sequence High Register	Address	0xF079
Register	ADS7H1	ADC1 Group7 Channel Sequence High Register	Address	0xF0F9

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	ADSCH11	R/W	0	Conversion of Channel11 0: Channel11 is not converted 1: Channel11 is converted	
2	ADSCH10	R/W	0	Conversion of Channel10 0: Channel10 is not converted 1: Channel10 is converted	
1	ADSCH9	R/W	0	Conversion of Channel9 0: Channel9 is not converted 1: Channel9 is converted	
0	ADSCH8	R/W	0	Conversion of Channel8 0: Channel8 is not converted 1: Channel8 is converted	

22.2.17. ADSTSELmn (ADCn Group m Start Trigger Select Register) (n = 0 to 1) (m = 0 to 7)

Register	ADSTSEL00	ADC0 Group0 Start Trigger Select Register	Address	0xF042
Register	ADSTSEL01	ADC1 Group0 Start Trigger Select Register	Address	0xF0C2
Register	ADSTSEL10	ADC0 Group1 Start Trigger Select Register	Address	0xF04A
Register	ADSTSEL11	ADC1 Group1 Start Trigger Select Register	Address	0xF0CA
Register	ADSTSEL20	ADC0 Group2 Start Trigger Select Register	Address	0xF052
Register	ADSTSEL21	ADC1 Group2 Start Trigger Select Register	Address	0xF0D2
Register	ADSTSEL30	ADC0 Group3 Start Trigger Select Register	Address	0xF05A
Register	ADSTSEL31	ADC1 Group3 Start Trigger Select Register	Address	0xF0DA
Register	ADSTSEL40	ADC0 Group4 Start Trigger Select Register	Address	0xF062
Register	ADSTSEL41	ADC1 Group4 Start Trigger Select Register	Address	0xF0E2
Register	ADSTSEL50	ADC0 Group5 Start Trigger Select Register	Address	0xF06A
Register	ADSTSEL51	ADC1 Group5 Start Trigger Select Register	Address	0xF0EA
Register	ADSTSEL60	ADC0 Group6 Start Trigger Select Register	Address	0xF072
Register	ADSTSEL61	ADC1 Group6 Start Trigger Select Register	Address	0xF0F2
Register	ADSTSEL70	ADC0 Group7 Start Trigger Select Register	Address	0xF07A
Register	ADSTSEL71	ADC1 Group7 Start Trigger Select Register	Address	0xF0FA

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	TRIGGER	R/W	0	Activation trigger number Select a trigger number used as an activation trigger from Table 22-7, then set it in 6 bits.	
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

22.2.18. ADEVTmLn (ADCn Group m Event Output Channel Low Register) (n = 0 to 1) (m = 0 to 7)

Register	ADEVT0L0	ADC0 Group0 Event Output Channel Low Register	Address	0xF044
Register	ADEVT0L1	ADC1 Group0 Event Output Channel Low Register	Address	0xF0C4
Register	ADEVT1L0	ADC0 Group1 Event Output Channel Low Register	Address	0xF04C
Register	ADEVT1L1	ADC1 Group1 Event Output Channel Low Register	Address	0xF0CC
Register	ADEVT2L0	ADC0 Group2 Event Output Channel Low Register	Address	0xF054
Register	ADEVT2L1	ADC1 Group2 Event Output Channel Low Register	Address	0xF0D4
Register	ADEVT3L0	ADC0 Group3 Event Output Channel Low Register	Address	0xF05C
Register	ADEVT3L1	ADC1 Group3 Event Output Channel Low Register	Address	0xF0DC
Register	ADEVT4L0	ADC0 Group4 Event Output Channel Low Register	Address	0xF064
Register	ADEVT4L1	ADC1 Group4 Event Output Channel Low Register	Address	0xF0E4
Register	ADEVT5L0	ADC0 Group5 Event Output Channel Low Register	Address	0xF06C
Register	ADEVT5L1	ADC1 Group5 Event Output Channel Low Register	Address	0xF0EC
Register	ADEVT6L0	ADC0 Group6 Event Output Channel Low Register	Address	0xF074
Register	ADEVT6L1	ADC1 Group6 Event Output Channel Low Register	Address	0xF0F4
Register	ADEVT7L0	ADC0 Group7 Event Output Channel Low Register	Address	0xF07C
Register	ADEVT7L1	ADC1 Group7 Event Output Channel Low Register	Address	0xF0FC

Bit	Bit Name	R/W	Initial	Description	Remarks
7	EVENTOUTCH7	R/W	0	Issuing an event at the completion of Channel7 conversion 0: Event is not issued 1: Event is issued	
6	EVENTOUTCH6	R/W	0	Issuing an event at Channel6 conversion completion 0: Event is not issued 1: Event is issued	
5	EVENTOUTCH5	R/W	0	Issuing an event at Channel5 conversion completion 0: Event is not issued 1: Event is issued	
4	EVENTOUTCH4	R/W	0	Issuing an event at Channel4 conversion completion 0: Event is not issued 1: Event is issued	
3	EVENTOUTCH3	R/W	0	Issuing an event at Channel3 conversion completion 0: Event is not issued 1: Event is issued	
2	EVENTOUTCH2	R/W	0	Issuing an event at Channel2 conversion completion 0: Event is not issued 1: Event is issued	
1	EVENTOUTCH1	R/W	0	Issuing an event at Channel1 conversion completion 0: Event is not issued 1: Event is issued	
0	EVENTOUTCH0	R/W	0	Issuing an event at Channel0 conversion completion 0: Event is not issued 1: Event is issued	

22.2.19. ADEVTmHn (ADCn Group m Event Output Channel High Register) (n = 0 to 1) (m = 0 to 7)

Register	ADEVT0H0	ADC0 Group0 Event Output Channel High Register	Address	0xF045
Register	ADEVT0H1	ADC1 Group0 Event Output Channel High Register	Address	0xF0C5
Register	ADEVT1H0	ADC0 Group1 Event Output Channel High Register	Address	0xF04D
Register	ADEVT1H1	ADC1 Group1 Event Output Channel High Register	Address	0xF0CD
Register	ADEVT2H0	ADC0 Group2 Event Output Channel High Register	Address	0xF055
Register	ADEVT2H1	ADC1 Group2 Event Output Channel High Register	Address	0xF0D5
Register	ADEVT3H0	ADC0 Group3 Event Output Channel High Register	Address	0xF05D
Register	ADEVT3H1	ADC1 Group3 Event Output Channel High Register	Address	0xF0DD
Register	ADEVT4H0	ADC0 Group4 Event Output Channel High Register	Address	0xF065
Register	ADEVT4H1	ADC1 Group4 Event Output Channel High Register	Address	0xF0E5
Register	ADEVT5H0	ADC0 Group5 Event Output Channel High Register	Address	0xF06D
Register	ADEVT5H1	ADC1 Group5 Event Output Channel High Register	Address	0xF0ED
Register	ADEVT6H0	ADC0 Group6 Event Output Channel High Register	Address	0xF075
Register	ADEVT6H1	ADC1 Group6 Event Output Channel High Register	Address	0xF0F5
Register	ADEVT7H0	ADC0 Group7 Event Output Channel High Register	Address	0xF07D
Register	ADEVT7H1	ADC1 Group7 Event Output Channel High Register	Address	0xF0FD

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	EVENTOUTCH11	R/W	0	Issuing an event at Channel11 conversion completion 0: Event is not issued 1: Event is issued	
2	EVENTOUTCH10	R/W	0	Issuing an event at Channel10 conversion completion 0: Event is not issued 1: Event is issued	
1	EVENTOUTCH9	R/W	0	Issuing an event at Channel9 conversion completion 0: Event is not issued 1: Event is issued	
0	EVENTOUTCH8	R/W	0	Issuing an event at Channel8 conversion completion 0: Event is not issued 1: Event is issued	

22.2.20. ADOmn (ADCn Channel m Data Offset Register) (n = 0 to 1) (m = 0 to 11)

Register	ADO00	ADC0 Channel0 Data Offset Register	Address 0x00
Register	ADO01	ADC1 Channel0 Data Offset Register	Address 0x01
Register	ADO10	ADC0 Channel1 Data Offset Register	Address 0x08
Register	ADO11	ADC1 Channel1 Data Offset Register	Address 0x09
Register	ADO20	ADC0 Channel2 Data Offset Register	Address 0x10
Register	ADO21	ADC1 Channel2 Data Offset Register	Address 0x11
Register	ADO30	ADC0 Channel3 Data Offset Register	Address 0x18
Register	ADO31	ADC1 Channel3 Data Offset Register	Address 0x19
Register	ADO40	ADC0 Channel4 Data Offset Register	Address 0x20
Register	ADO41	ADC1 Channel4 Data Offset Register	Address 0x21
Register	ADO50	ADC0 Channel5 Data Offset Register	Address 0x28
Register	ADO51	ADC1 Channel5 Data Offset Register	Address 0x29
Register	ADO60	ADC0 Channel6 Data Offset Register	Address 0x30
Register	ADO61	ADC1 Channel6 Data Offset Register	Address 0x31
Register	ADO70	ADC0 Channel7 Data Offset Register	Address 0x38
Register	ADO71	ADC1 Channel7 Data Offset Register	Address 0x39
Register	ADO80	ADC0 Channel8 Data Offset Register	Address 0x40
Register	ADO81	ADC1 Channel8 Data Offset Register	Address 0x41
Register	ADO90	ADC0 Channel9 Data Offset Register	Address 0x48
Register	ADO91	ADC1 Channel9 Data Offset Register	Address 0x49
Register	ADOA0	ADC0 Channel10 Data Offset Register	Address 0x50
Register	ADOA1	ADC1 Channel10 Data Offset Register	Address 0x51
Register	ADOB0	ADC0 Channel11 Data Offset Register	Address 0x58
Register	ADOB1	ADC1 Channel11 Data Offset Register	Address 0x59

Bit	Bit Name	R/W	Initial	Description	Remarks
15	Reserved	R	0	Sign extension bit Writing 1 to the SIGN bit sets the bit to 1.	
14	Reserved	R	0		
13	Reserved	R	0		
12	SIGN	R/W	0	Sign bit	
11	OFFSET	R/W	0	Offset An offset value used for each channel can be set by signed 13-bit values (−4096 to 4096). The SIGN bit defines which sign is to be set.	
10		R/W	0		
9		R/W	0		
8		R/W	0		
7		R/W	0		
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

22.2.21. ADmn (ADCn Channel m Data Register) (n = 0 to 1) (m = 0 to 11)

For details, see Section 22.6.1.

Register	AD00	ADC0 Channel0 Data Register	Address	0x99
Register	AD01	ADC1 Channel0 Data Register	Address	0x9A
Register	AD10	ADC0 Channel1 Data Register	Address	0xA1
Register	AD11	ADC1 Channel1 Data Register	Address	0xA2
Register	AD20	ADC0 Channel2 Data Register	Address	0xA9
Register	AD21	ADC1 Channel2 Data Register	Address	0xAA
Register	AD30	ADC0 Channel3 Data Register	Address	0xB1
Register	AD31	ADC1 Channel3 Data Register	Address	0xB2
Register	AD40	ADC0 Channel4 Data Register	Address	0xB9
Register	AD41	ADC1 Channel4 Data Register	Address	0xBA
Register	AD50	ADC0 Channel5 Data Register	Address	0xC1
Register	AD51	ADC1 Channel5 Data Register	Address	0xC2
Register	AD60	ADC0 Channel6 Data Register	Address	0xC9
Register	AD61	ADC1 Channel6 Data Register	Address	0xCA
Register	AD70	ADC0 Channel7 Data Register	Address	0xD1
Register	AD71	ADC1 Channel7 Data Register	Address	0xD2
Register	AD80	ADC0 Channel8 Data Register	Address	0xD9
Register	AD81	ADC1 Channel8 Data Register	Address	0xDA
Register	AD90	ADC0 Channel9 Data Register	Address	0xE1
Register	AD91	ADC1 Channel9 Data Register	Address	0xE2
Register	ADA0	ADC0 Channel10 Data Register	Address	0xE9
Register	ADA1	ADC1 Channel10 Data Register	Address	0xEA
Register	ADB0	ADC0 Channel11 Data Register	Address	0xF1
Register	ADB1	ADC1 Channel11 Data Register	Address	0xF2

Bit	Bit Name	R/W	Initial	Description	Remarks
15	Reserved	R	0	Sign extension bit	
14	Reserved	R	0	Writing 1 to the DATASIGN bit sets the bit to 1.	
13	DATASIGN	R	0	Sign bit	
12	DATA	R	0	Conversion result	
11		R	0		
10		R	0		
9		R	0		
8		R	0		
7		R	0		
6		R	0		
5		R	0		
4		R	0		
3		R	0		
2		R	0		
1		R	0		
0		R	0		

22.2.22. ADIFn (ADCn Interrupt Flag Register) (n = 0 to 1)

Register	ADIF0	ADC0 Interrupt Flag Register	Address	0x89
Register	ADIF1	ADC1 Interrupt Flag Register	Address	0x8A

Bit	Bit Name	R/W	Initial	Description	Remarks
15	Reserved	R	0	The read value is 0. The write value must always be 0.	
14	Reserved	R	0	The read value is 0. The write value must always be 0.	
13	Reserved	R	0	The read value is 0. The write value must always be 0.	
12	Reserved	R	0	The read value is 0. The write value must always be 0.	
11	Reserved	R	0	The read value is 0. The write value must always be 0.	
10	Reserved	R	0	The read value is 0. The write value must always be 0.	
9	Reserved	R	0	The read value is 0. The write value must always be 0.	
8	Reserved	R	0	The read value is 0. The write value must always be 0.	
7	IFLG7	R/C	0	Interrupt flag of Group7 Read 0: Interrupt is not detected Read 1: Interrupt is detected Write 0: No change Write 1: The bit is cleared	
6	IFLG6	R/C	0	Interrupt flag of Group6 Read 0: Interrupt is not detected Read 1: Interrupt is detected Write 0: No change Write 1: The bit is cleared	
5	IFLG5	R/C	0	Interrupt flag of Group5 Read 0: Interrupt is not detected Read 1: Interrupt is detected Write 0: No change Write 1: The bit is cleared	
4	IFLG4	R/C	0	Interrupt flag of Group4 Read 0: Interrupt is not detected Read 1: Interrupt is detected Write 0: No change Write 1: The bit is cleared	
3	IFLG3	R/C	0	Interrupt flag of Group3 Read 0: Interrupt is not detected Read 1: Interrupt is detected Write 0: No change Write 1: The bit is cleared	
2	IFLG2	R/C	0	Interrupt flag of Group2 Read 0: Interrupt is not detected Read 1: Interrupt is detected Write 0: No change Write 1: The bit is cleared	
1	IFLG1	R/C	0	Interrupt flag of Group1 Read 0: Interrupt is not detected Read 1: Interrupt is detected Write 0: No change Write 1: The bit is cleared	
0	IFLG0	R/C	0	Interrupt flag of Group0 Read 0: Interrupt is not detected Read 1: Interrupt is detected Write 0: No change Write 1: The bit is cleared	

22.3. Operation

22.3.1. Basic Operation

Before a register is operated, enable the ADC module clock (set the MCLKE2.ME_ADC0 and MCLKE2.ME_ADC1 bits to 1). After that, the ADC is enabled and operated by setting the ADENn.ADENABLE bit to 1. If the ADC is not used, power consumption can be reduced by setting the ADENn.ADENABLE bit to 0. Figure 22-2 shows the flowchart of the ADC initialization process.

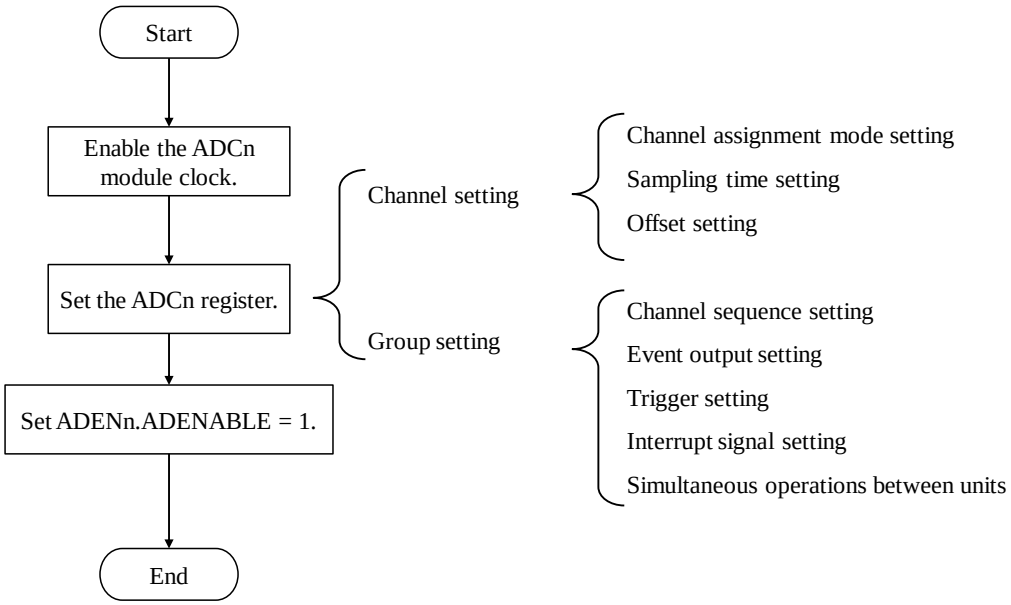


Figure 22-2. Flowchart of Initialization Process

If the group to be activated by the trigger exists when the trigger from the GPIO, comparator, PWM, timer, DSAC, EPU, or CPU is received during the ADC operation, the conversion process is started. This trigger is called an activation trigger. As an exception, the trigger from the CPU starts the conversion process regardless of the ADENn register value. Care must be taken when the trigger is received from the CPU while the ADENn.ADENABLE bit is 0. At this time, since the power supply voltage is not applied to the comparator, the ADC cannot generate the correct conversion result. When the activation trigger is received while the ADC is in an idle state (i.e., a state where the ADC is not converting), the conversion process is started after the group to be processed is selected. For this group selection, one cycle is consumed, i.e., the ADLOOP trigger, which is the CPU trigger, operates one cycle behind compared to other triggers.

The internal capacitor is charged (sample-and-hold) before the ADC converts the input to the digital value. The sampling period is in accordance with the cycles (1 cycle to 256 cycles) set by each channel. After the sample-and-hold, the capacitance is converted to the digital. The conversion algorithm takes 12 cycles with successive approximation register (SAR) by a binary search.

As shown in Figure 22-3, the time taking to convert the first channel after receiving the trigger is the sampling period (t) + 13 cycles.

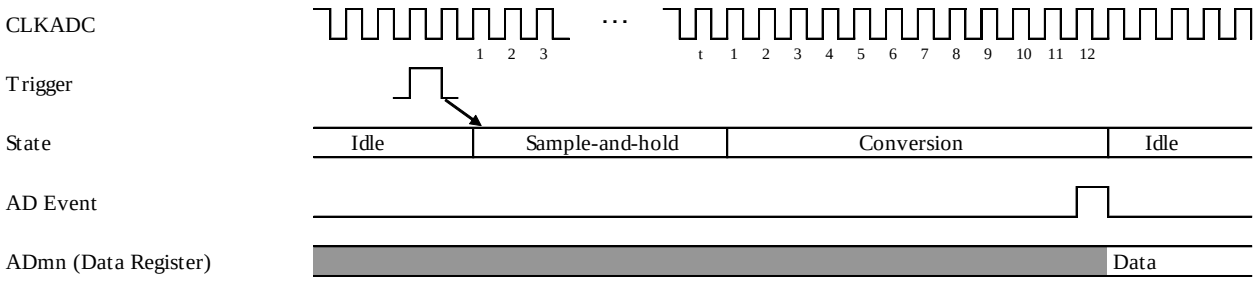


Figure 22-3. Conversion Timing of 1 Channel by Binary Search

Since the second or subsequent channels and the groups are processed continuously, the conversion process requires the sampling period (t) + 12 cycles. The result converted to the digital value is stored in the ADmn register for each channel after adding the offset. When the conversion process of each channel is completed, the ADC generates the ADC event for each group. Also, when the processing of each group is completed, the ADC generates an interrupt signal. Since the interrupt signal is not automatically cleared, user must clear the interrupt signal.

Figure 22-4, Figure 22-5, and Figure 22-6 show the basic operation timings. Figure 22-4 shows the basic operation when issuing the trigger to activate the Group x that converts the Channel a. Figure 22-5 shows the basic operation when issuing the trigger to activate the Group x that converts the Channel a and Channel b. Figure 22-6 shows the basic operation when issuing one trigger to activate the multiple groups (Group x and Group y), and then issuing a trigger to activate another group, Group z, during the execution (it is assumed that the Group z has higher priority than the Group y).

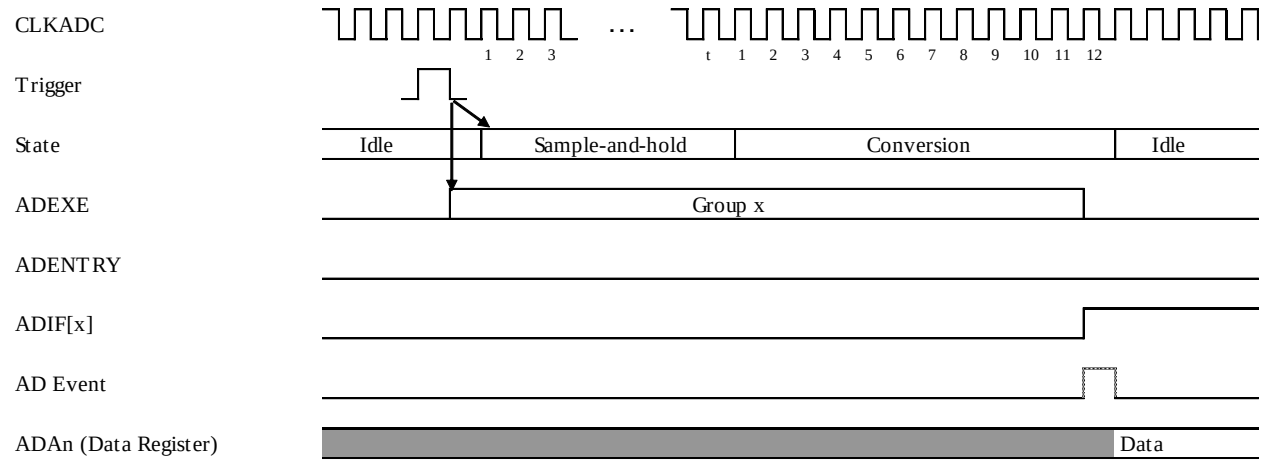


Figure 22-4. Basic Operation Timing in Conversion Group of 1 Channel

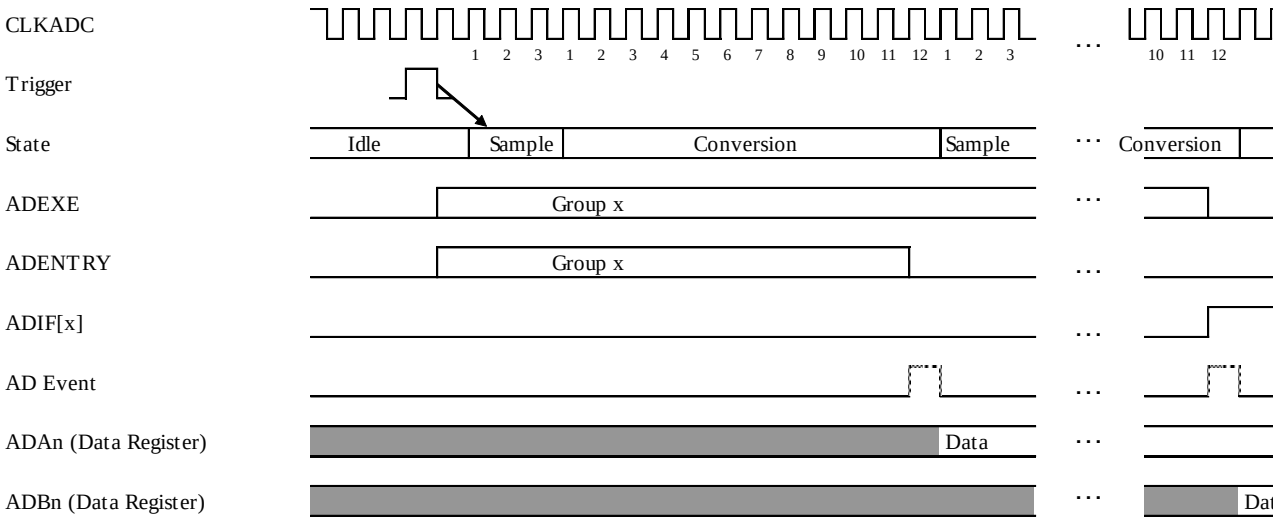


Figure 22-5. Basic Operation Timing in Conversion Group of 2 Channels

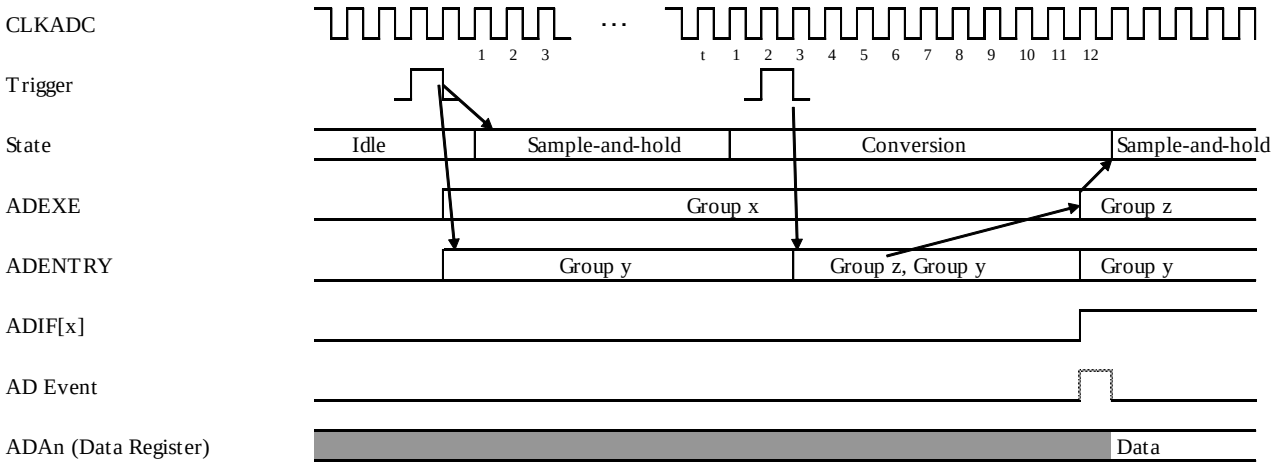


Figure 22-6. Basic Operation Timing in Operating 2 or More Groups

22.3.2. Register Access

The following cares should be taken when accessing the registers.

- If the ADOmn registers of the XDATA BUS and the SFR BUS are written simultaneously, the XDATA BUS register has the highest priority.
- If the setting operation and the resetting operation of the ADIFn registers occur simultaneously, the setting operation has the highest priority.
- When the ADmn register is read by 8-bit access from the CPU, the lower 8-bit data is read in the first access, and the high 8-bit data is read in the second access. Be sure to perform the read operation continuously to the lower and higher bits.

22.4. Analog Inputs and Channels

Each ADC has 12 analog inputs. The particular channel numbers (0 to 11) are assigned to the analog input. Table 22-5 describes the channel number assignment. The ADCn.CHCONMODE bit determines the channel numbers to assign to the analog inputs.

Each analog input has a switch. While this switch is turned off, the analog input becomes high impedance. During the sampling, only one input switch is turned on, and the analog input pin is connected to the ADC internal sampling capacitor. If the ADENn.ADENABLE bit is set to 0, all analog input switches are turned off even while the conversion process is being executed.

The following sections describe the sampling period and the offset, which are the channel setting items.

Table 22-5. Channel Number Assignment to Analog Input

ADC0			ADC1		
Analog Input	Channel Number		Analog Input	Channel Number	
	Mode 0	Mode 1		Mode 0	Mode 1
ANEX0	0	11	ANEX0	0	11
ANEX1	1	10	ANEX1	1	10
ANEX2	2	9	ANEX2	2	9
ANEX3	3	8	ANEX3	3	8
ANEX4	4	7	ANEX4	4	7
ANEX5	5	6	ANEX5	5	6
ANEX6	6	5	ANEX6	6	5
ANEX7	7	4	ANEX7	7	4
ANEX8	8	3	ANEX12	8	3
ANEX9	9	2	ANEX13	9	2
ANEX10	10	1	TEMP	10	1
ANEX11	11	0	VREF	11	0

22.4.1. Sample

A sample operates to inject the charge into the ADC internal sampling capacitor and to make the analog input voltage equal to the ADC internal sampling capacitor voltage. The sampling cycle is determined by the ADNSMPmn register for each channel.

The ADC analog input pin is connected to the ADC internal sampling capacitor by turning on the input switch. The voltage of the sampling capacitor is undefined. During the sampling cycle, the current flows so that the voltages of the sampling capacitor and the analog input are the same.

The ADNSMPmn register determines the sampling period of each channel from 1 cycle to 256 cycles. This setting is common to all groups. If the ADNSMPmn register is set to 0x00, the sampling period is regarded as 256 cycles.

The ADC is designed within the error range described in the electrical characteristics (Section 29) when the impedance of the part where the voltage is measured is 0 and the sampling period is 3 cycles outside the LSI. When the sampling period is shortened, the charge of the internal sampling capacitor is not changed sufficiently, and the correct conversion result may not be obtained in some cases. Conversely, the longer the sampling period, the smaller the error can be, but the longer the conversion time takes. The appropriate sampling period must be set according to the impedance size of the part where the voltage is measured.

22.4.2. Offset

Before the result converted to the digital value is output to each channel, the offset to be added can be set. To set the offset of the Channel m, write the signed 13-bit values (−4096 to 4095) to the ADOmn register. This setting is common to all groups.

The ADOmn register is accessed from the XDATA BUS and the SFR BUS.

The number of cycles required to store the result in the register is constant regardless of the offset setting.

22.5. Group

The ADC manages the followings in a group: when to activate, and which channel to be processed in which order. Up to 8 groups can be used in one ADC. The numbers (0 to 7) are assigned to each group.

For each group, it is necessary to set the information such as an activation trigger, a channel sequence, an event generation channel, and the enable or disable of interrupt signal.

The following sections describe the group processing method and the group setting items executed by the ADC.

22.5.1. Group Status

In each group, an execution status and an entry status are managed separately.

The execution status indicates whether the group is currently being processed or not. The execution status has 2 states, “executing” and “stopping”. The execution status of the Group m can be acquired from the ADEXEn.EXECUTIONm bit. When ADEXEn.EXECUTIONm = 1, the Group m is in the executing status.

The entry status indicates whether or not a channel conversion process has been entered to the ADC. The entry status has 2 states, “entered” and “unentered”. If the group no longer needs to make a channel conversion request to the ADC (or, starts processing the last channel of the channel sequence), the entry status turns into the unentered state. The entry status of the Group m can be acquired from the ADENTRYn.ENTRYm bit. When ADENTRYn.ENTRYm = 1, the Group m is being entered. In this way, each group makes transitions among 4 statuses (2 execution statuses × 2 entry statuses). These 4 statuses are classified as shown in Table 22-6.

Table 22-6. Group States

Execution Status	Entry Status	Group Status
Stopping	Unentered	Stopped
	Entered	Idling
Executing	Unentered	Processing the final channel
	Entered	Processing the channels other than the final or Processing the final channel and waiting for the next processing

22.5.2. Processes from Group Activation to Completion

This section describes a process flow from when a group receives an activation trigger to when the group is activated, the ADC performs the conversion process, and the group is completed.

To activate the group, the ADC needs to receive the activation trigger of the group. Section 22.5.6 describes the details of the activation trigger. When the group receives the activation trigger, the entry status is changed to the entered state. The execution status is not changed.

When the ADC receives the activation trigger, or completes the execution group processing, the ADC selects the next execution group. The ADC selects the group with the smallest number in the groups of the entered state, and turns into the execution status. Where, the receiving order of the activation trigger is not taken into account. Even if the ADC receives the last activation trigger, the group is selected when the group number is the smallest.

When the group enters the activation state, the ADC processes all channels of the channel sequence in the order of the smaller channel number. The channel to be converted by the group is set in the channel sequence. Section 22.5.4

describes the details of the channel sequence. If the event generation is specified, the ADC event is issued every time each channel processing completes.

While the group is in the entered state, the ADC is ignored even if the activation trigger is received. Note that, when the ADC starts processing the last channel, the entry status of the group turns into the unentered state. Therefore, while the group processes the last channel, this group being executed can also receive the activation trigger.

When the channel sequence of the group has one channel, the operation is as follows:

- If the group becomes the execution status, the entry status of the group turns into unentered state simultaneously.
- If the group is activated after receiving the activation trigger, the entry status maintains the last unentered state.

When the processing of all channels in the channel sequence is completed, the group completes the processing by setting 1 to the interrupt signal flag of the ADIFn register.

22.5.3. Processing of Execution Group Selection

When the ADC receives the activation trigger, or completes the execution group processing, the ADC selects the next execution group. The ADC selects the group with the smallest number in the groups of the entered state, and turns into the execution status. Where, the receiving order of the activation trigger is not taken into account. Even if the ADC receives the last activation trigger, the group is selected when the group number is the smallest.

Even if the ADC receives the activation trigger multiple times while the group is being entered, the entry status turns into the unentered state whenever the group is executed, regardless of its number of times. To entry the group again, issue the activation trigger in the unentered state. To keep the group always entered state, use the ADLOOP trigger, which is the CPU trigger. The activation trigger continues to be issued automatically for each cycle by setting ADLOOPn.LOOPm = 1. For details of the ADLOOPn register, see Section 22.5.6.3.

22.5.4. Channel Sequence

Each group has a channel sequence that sets which channel to be processed. When the Channel m is processed in the Group m, set the ADSCHm bit of the ADS mLn or ADSmHn register to 1. Where, converting one channel by one group processing multiple times cannot be set. For example, to process 3 channels (Channel1, Channel5, and Channel11) in the Group1, write 0b0000_1000_0010_0010, which is binary number, to the ADS1Ln and ADS1Hn registers. Where, the lower 8 bits (0b0010_0010) are written to the ADS1Ln register, and the higher 8 bits (0b0000_1000) are written to the ADS1Hn register.

When the ADC processes the group, all groups registered in the channel sequence are processed in the order of the channel number. Do not rewrite the channel sequence of the group (i.e., the setting values of the ADSmHn and ADSmLn registers) while the ADC processes the group.

22.5.5. Event Generation Channel

The ADEVTmn register determines whether the event is generated or not for each channel. In the same way as the ADSmHn and ADSmLn registers, to process the Channel m in the Group m, set the ADEVTm bit of the ADEVTmLn and ADEVTmHn registers to 1. It is no problem even if this value is different from the channel sequence values (i.e., the values of the ADSmHn and ADSmLn registers).

For example, to generate the events of the Channel1, Channel5, and Channel11, in the Group0, write 0b0000_1000_0010_0010, which is binary number, to the ADEVT0Ln and ADEVT0Hn registers. Where, the lower 8 bits (0b0010_0010) are written to the ADEVT0Ln register, and the higher 8 bits (0b0000_1000) are written to the ADEVT0Hn register.

For details of the ADC event, see Section 22.6.2.

22.5.6. Activation Trigger

The following 3 activation triggers are used in each group. No distinction exists among these 3 triggers; these are used as the same activation triggers.

- Trigger by the event (select one trigger from GPIO trigger, comparator trigger, PWM trigger, timer trigger, DSAC trigger, and EPU trigger)
- ADC trigger from the CPU
- ADLOOP trigger from the CPU

22.5.6.1. Activation Trigger by Event

To specify the activation trigger of the Group m, select the activation trigger from Table 22-7, and set the corresponding trigger numbers to the ADSTSELmn register. To not specify the activation trigger, set the ADSTSELmn register to 0. Also, one trigger can be specified as the activation trigger for multiple groups.

To use the trigger numbers of 1 to 9, enable the EVC module clock (i.e., set MCLKE1.ME_EVC = 1). To use the trigger numbers of 34 to 57, set the event input to the ADC by the EVC module register (any of EVSEL5, EVSEL6, EVSEL7, or EVSEL8).

Table 22-7. Trigger Numbers

Number	Trigger Type	Number	Trigger Type	Number	Trigger Type
0	—	20	TMR1_CMA	40	EPU0_8/EPU1_8
1	GPIO0 rise	21	TMR1_CMB	41	EPU0_9/EPU1_9
2	GPIO0 fall	22	TMR2_CMA	42	EPU2_2/EPU3_2
3	GPIO0 both	23	TMR2_CMB	43	EPU2_3/EPU3_3
4	GPIO1 rise	24	TMR3_CMA	44	EPU2_4/EPU3_4
5	GPIO1 fall	25	TMR3_CMB	45	EPU2_5/EPU3_5
6	GPIO1 both	26	PWM0_0	46	EPU2_6/EPU3_6
7	GPIO2 rise	27	PWM0_1	47	EPU2_7/EPU3_7
8	GPIO2 fall	28	PWM1_0	48	EPU2_8/EPU3_8
9	GPIO2 both	29	PWM1_1	49	EPU2_9/EPU3_9
10	CMP0	30	PWM2_0	50	EPU4_2/EPU5_2
11	CMP1	31	PWM2_1	51	EPU4_3/EPU5_3
12	CMP2	32	PWM3_0	52	EPU4_4/EPU5_4
13	CMP3	33	PWM3_1	53	EPU4_5/EPU5_5
14	CMP4	34	EPU0_2/EPU1_2	54	EPU4_6/EPU5_6
15	CMP5	35	EPU0_3/EPU1_3	55	EPU4_7/EPU5_7
16	—	36	EPU0_4/EPU1_4	56	EPU4_8/EPU5_8
17	—	37	EPU0_5/EPU1_5	57	EPU4_9/EPU5_9
18	TMR0_CMA	38	EPU0_6/EPU1_6		
19	TMR0_CMB	39	EPU0_7/EPU1_7		

22.5.6.2. ADT Trigger

The ADTn register is to issue the activation trigger from the CPU to the ADC. Since the CPU trigger is set in advance to the group activation trigger, the setting cannot be changed.

One trigger is issued by writing to the ADTn register once. To issue the activation trigger from the CPU to the Group m, set ADTn.TRIGGERm = 1. This register is dedicated to the writing.

22.5.6.3. ADLOOP Trigger

The ADLOOPn register is to issue the activation trigger from the CPU to the ADC. Since the CPU trigger is set in advance to the group activation trigger, the setting cannot be changed.

To issue the activation trigger from the CPU to the Group m, set ADLOOPn.LOOPm = 1. During ADLOOPn.LOOPm = 1, the activation trigger of the Group m is issued continuously to the ADC for each cycle. To stop issuing this activation trigger, set ADLOOPn.LOOPm = 0.

The ADLOOP trigger can be used for continuing to operate the ADC without using the CPU resources for the channel that the value to be always monitored.

Since the ADLOOP trigger issues the activation trigger after the value is stored in the register, the time from the idle state to the ADC activation state is delayed by one cycle from the ADT trigger.

22.5.7. Enable/Disable Setting of Interrupt Signal

To enable the interrupt signal of the Group m, set ADIENn.IENm = 1.

When processing of the Group m is completed, the ADC sets the ADIFn.IFLGm bit to 1, and set the interrupt flag of the Group m to 1. If the interrupt signal of the group is not enabled here, the interrupt signal from the ADC is not generated.

For details of the interrupt signal, see Section 22.6.3.

22.6. Output

22.6.1. Acquisition of Conversion Result

The ADC stores the signed 16-bit value in the ADmn register. The signed 16-bit value is result of adding the unsigned 12-bit value (0 to 4095), which is acquired by digitally converting the input of Channel m, and the signed 13-bit channel offset (−4096 to 4095), which is defined by the ADOmn register.

When reading the ADmn register from the CPU, the amount of data that can be read out at once is 8 bit. Thus, it is necessary to read it twice in order to acquire 16-bit data. When reading twice sequentially from the ADmn register, the lower 8 bits are acquired in the first reading and the higher 8 bits are acquired in the second reading. The ADmn register is basically read twice sequentially. If reading the first lower bits again after reading the first lower bits, set the ADACCLRn register to 1.

When reading from the DSAC or the EPU, all data (16 bits) can be read at once.

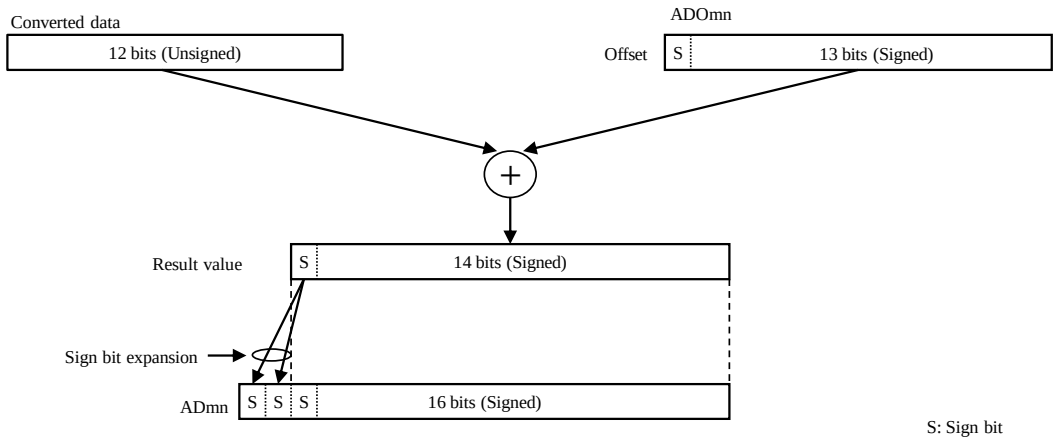


Figure 22-7. Method of Offset Addition

22.6.2. ADC Event

When completing the conversion of each channel, the ADC issues the ADC event if specified. The ADC event issues the different signals for each group. At this time, the same ADC event is issued in the same groups even if the channels are different.

The ADEVTmHn and ADEVTmLn registers determine whether the event that is issued in each channel of each group exists or not. While one group is executed, multiple ADC events can be issued.

22.6.3. Interrupt Signal

When existing the group that the interrupt flag is 1 and the interrupt signal is enabled, the ADC interrupt signal is issued. The interrupt flag of the group that the interrupt signal is not enabled is ignored. To enable the interrupt signal of the Group m, set ADIENn.IENm = 1.

The interrupt flag of each group becomes 1 when each group processing is completed regardless of the interrupt enable or disable. Also, the interrupt flag that becomes 1 once does not become 0 unless user clears it. The interrupt flag of the Group m is set to 1 by setting ADIFn.IFLGm = 1. To clear all interrupt flags, write 0xFF to the ADIFn register. If the clearing operation of the interrupt flag and the setting operation by the ADC occur simultaneously, the setting operation by the ADC has the highest priority.

The ADIFn register can also be used to check which group processing is completed.

22.7. Synchronous Operations between Units

Although the conversions of the ADC0 and the ADC1 are normally processed independently, setting the ADSYNCn.SYNCHRONOUS bits of ADC0 and ADC1 to 1 allows the conversion timing of the Group0 (the timing to go into sample-and-hold operation) to be synchronized (see Figure 22-8).

This mode is called "synchronous operations between units," i.e., synchronous operation mode. Be sure to set the ADSYNCn.SYNCHRONOUS bits of both ADC0 and ADC1 to the same value to make this mode work properly. In the synchronous operation mode, the period necessary for synchronizing the ADC0 and the ADC1 is 3 cycles after the ADEXEn register changes.

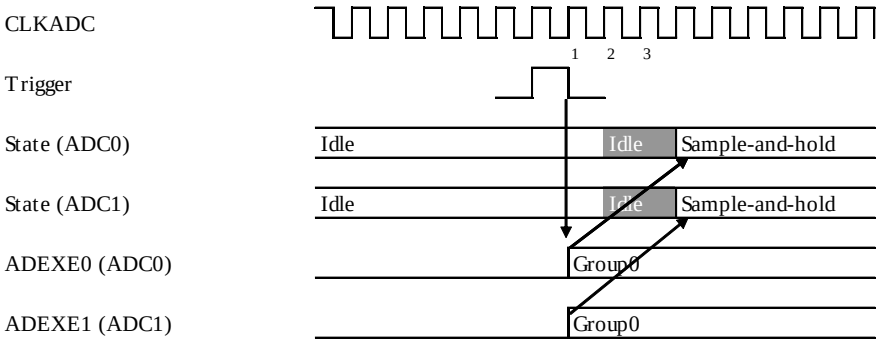


Figure 22-8. Timing to Activate Simultaneously with Same Triggers

During the synchronous operation mode, when either ADC0 or ADC1 attempts to execute the Group0, it enters the idle state until the other executes the Group0. When the ADEXEn.EXECUTION0 bits of both ADC0 and ADC1 become 1, the ADC0 and ADC1 start the conversion process of the Group0 simultaneously.

Figure 22-9 and Figure 22-10 show the basic timings of synchronous operations between units.

When only the ADSYNCn.SYNCHRONOUS bit of either ADC0 or ADC1 is set to 1, the state continues after it enters the idle state, because the ADEXEn.EXECUTION0 bit of the ADC that has ADSYNCn.SYNCHRONOUS = 0 does not become 1. Therefore, be sure to set the ADSYNCn.SYNCHRONOUS bits of both ADC0 and ADC1 to the same values.

When setting ADSYNCn.SYNCHRONOUS = 0, the ADC returns from the idle state and starts the Group0 processing (see Figure 22-11).

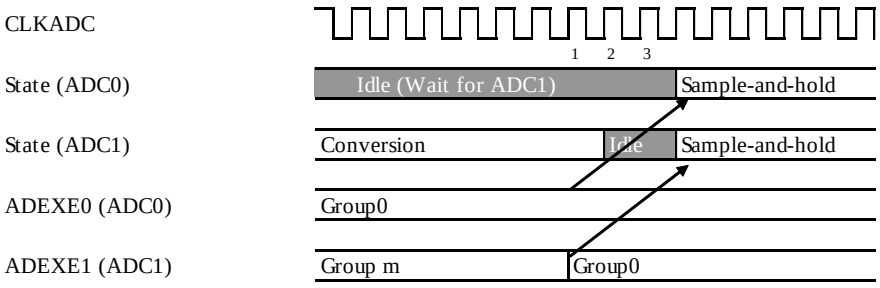


Figure 22-9. Timing to Start Group0 Processing Simultaneously with ADC0 after Group m Processing Completion by ADC1

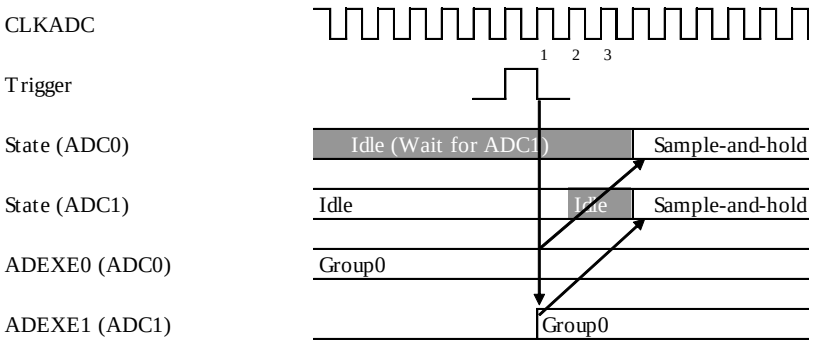


Figure 22-10. Timing to Activate Group0 by ADC1 after Receiving Trigger

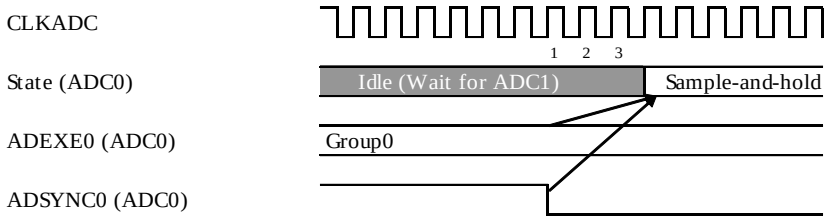


Figure 22-11. Timing to Cancel ADC0 Idling with Setting ADSYNC0.SYNCHRONOUS = 0

22.8. Usage Notes and Restrictions

Do not rewrite the ADCn register during conversion operations.

23. Op Amp (OPAMP)

23.1 Overview

The LSI has a general-purpose op amp that is selected from a standalone or a unity-gain (also called a voltage follower) type.

The input and output pins of the op amp (OPAMP) can be connected to an external pins or an internal resource. These connections are set by the corresponding register.

Table 23-1. OPAMP Functional Descriptions

Item	Description	Remarks
Number of Units	2 units	
Selectable Mode	- Standalone mode - Unity-gain mode (×1 or ×4) - Low power consumption mode - Enable or disable control by an event - Positive input bypass function	The setting bit of the low power consumption mode of IBIAS is in SYSC.

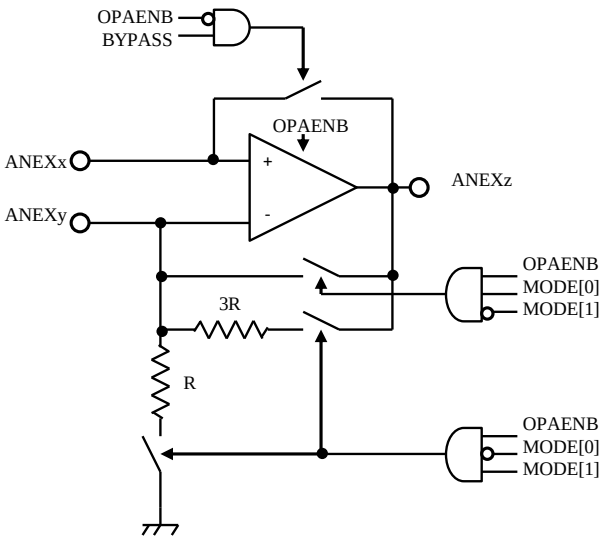


Figure 23-1. OPAMP Block Diagram

23.2 Register Descriptions

Table 23-2. List of Registers

Symbol	Name	Address	Initial Value
MIXOPA0	Mix OPAMP0 Configuration Register	0xF600	0x00
MIXPGA0	Mix OPAMP0 PGA Configuration Register	0xF601	0x00
MIXEEVCR0	Mix OPAMP0 Enable Event Control Register	0xF602	0x00
MIXDEVCR0	Mix OPAMP0 Disable Event Control Register	0xF603	0x00
MIXOPA1	Mix OPAMP1 Configuration Register	0xF680	0x00
MIXPGA1	Mix OPAMP1 PGA Configuration Register	0xF681	0x00
MIXEEVCR1	Mix OPAMP1 Enable Event Control Register	0xF682	0x00
MIXDEVCR1	Mix OPAMP1 Disable Event Control Register	0xF683	0x00

23.2.1 MIXOPAn (Mix OPAMP n Configuration Register) (n = 0 to 1)

Register	MIXOPA0	Mix OPAMP0 Configuration Register	Address	0xF600
Register	MIXOPA1	Mix OPAMP1 Configuration Register	Address	0xF680

Bit	Bit Name	R/W	Initial	Description	Remarks
7	OPAENB	R/W	0	OPAMP enable 0: OPAMP is disabled 1: OPAMP is enabled When a disable event is detected, the OPAENB bit is set to 0. When an enable event is detected, the OPAENB bit is set to 1. If a write operation to the bit and either of these event-driven rewrite operations occurred simultaneously, the write operation to the OPAENB bit has the highest priority.	
6	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
5	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
4	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
3	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
2	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
1	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
0	Reserved	R/W	0	The read value is 0. The write value must always be 0.	

23.2.2 MIXPGAn (Mix OPAMP n PGA Configuration Register) (n = 0 to 1)

Register	MIXPGA0	Mix OPAMP0 PGA Configuration Register	Address	0xF601
Register	MIXPGA1	Mix OPAMP1 PGA Configuration Register	Address	0xF681

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BYPASS	R/W	0	Positive input and output connection setting 0: No connection 1: Positive input and output are connected The bit is valid only when the OPAENB bit is 0. When the OPAENB bit is 1, the positive input and output are not connected, regardless of the BYPASS bit setting.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
2	Reserved	R/W	0	The read value is 0. The write value must always be 0.	
1	MODE	R/W	0	OPAMP operating mode selection 00: Op amp 01: Unity-gain amplifier (×1) 10: Unity-gain amplifier (×4) 11: Setting prohibited	
0		R/W	0		

23.2.3 MIXEEVCRn (Mix OPAMP n Enable Event Control Register) (n = 0 to 1)

Register	MIXEEVCR0	Mix OPAMP0 Enable Event Control Register	Address	0xF602
Register	MIXEEVCR1	Mix OPAMP1 Enable Event Control Register	Address	0xF682

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	EVON	R/W	0	OPAMP enable event selection 00000: No selection made. Only writing to the OPAENB bit enables the OPAMP. 00001: TMR0_CMA 00010: TMR0_CMB 00011: TMR1_CMA 00100: TMR1_CMB 00101: TMR2_CMA 00110: TMR2_CMB 00111: TMR3_CMA 01000: TMR3_CMB 01001: PWM0_0 01010: PWM0_1 01011: PWM1_0 01100: PWM1_1 01101: PWM2_0 01110: PWM2_1 01111: PWM3_0 10000: PWM3_1 10001: EPU0 10010: EPU1 10011: EPU2 10100: EPU3 10101: EPU4 10110: EPU5 Other than above: Setting prohibited	
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

23.2.4 MIXDEVCRn (Mix OPAMP n Disable Event Control Register) (n = 0 to 1)

Register	MIXDEVCR0	Mix OPAMP0 Disable Event Control Register	Address	0xF603
Register	MIXDEVCR1	Mix OPAMP1 Disable Event Control Register	Address	0xF683

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	EVOFF	R/W	0	OPAMP disable event selection 00000: No selection made. Only writing to the OPAENB bit disables the OPAMP. 00001: TMR0_CMA 00010: TMR0_CMB 00011: TMR1_CMA 00100: TMR1_CMB 00101: TMR2_CMA 00110: TMR2_CMB 00111: TMR3_CMA 01000: TMR3_CMB 01001: ADC0_0 01010: ADC0_1 01011: ADC0_2 01100: ADC0_3 01101: ADC0_4 01110: ADC0_5 01111: ADC0_6 10000: ADC0_7 10001: ADC1_0 10010: ADC1_1 10011: ADC1_2 10100: ADC1_3 10101: ADC1_4 10110: ADC1_5 10111: ADC1_6 11000: ADC1_7 11001: EPU0 11010: EPU1 11011: EPU2 11100: EPU3 11101: EPU4 11110: EPU5 Other than above: Setting prohibited	
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

The enabled/disabled defined by the OPAMP.OPAENB bit can be set by an event.

An event is defined by the MIXEEVCRn.EVON bit or the MIXDEVCRn.EVOFF bit (other than zero). When the setting event of the MIXEEVCRn.EVON bit or the MIXDEVCRn.EVOFF bit is generated, the OPAMP can be enabled or disabled, respectively. In addition, the OPAENB can be set 1 or 0 directly to enable or disable the OPAMP, respectively. If an enable event defined by the MIXEEVCRn.EVON bit and a disable event defined by the MIXDEVCRn.EVOFF are generated at the same time, the disable event is ignored because the enable event has the highest priority.

Figure 23-2 shows the OPAMP operation example when a PWM and an AD conversion are set to the enable and disable events, respectively. When the PWM Event is generated, the OPAMP is enabled, and the TMR counter is concurrently cleared. Then, the AD conversion is started by the compare match of the TMR. When the AD conversion complete event is generated, the OPAMP is disabled. Since the OPAMP only operates during the AD conversion, the power consumption is reduced. In the operation, the TMR is used to generate the settling time of the OPAMP.

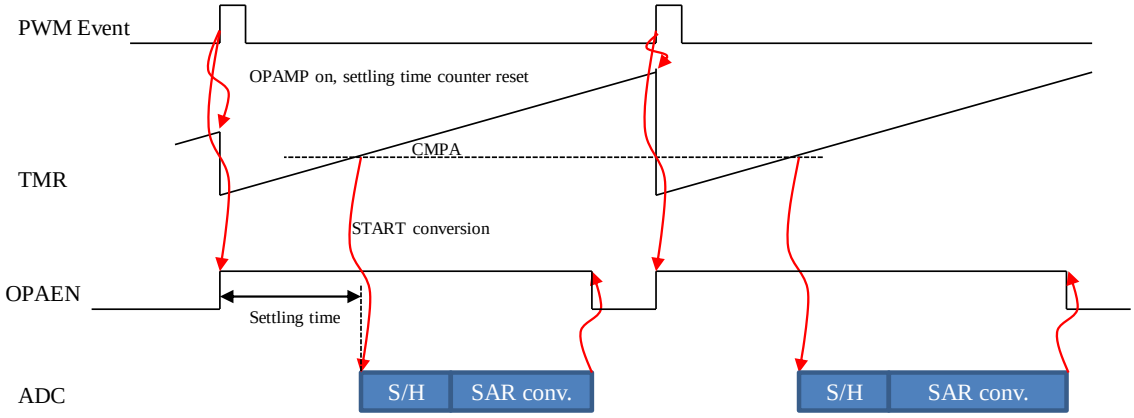


Figure 23-2. OPAMP Operational Example

24. Comparator

24.1. Overview

The LSI has 6 units of high-speed comparators with the DAC to generate reference voltage.

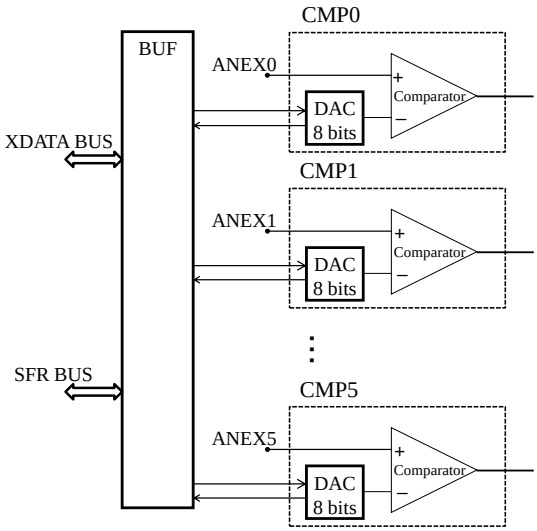


Figure 24-1. Comparator Block Diagram

Table 24-1. Comparator Functional Descriptions

Item	Description
Number of Units	6 units
Reference DAC	- R-2R structure - Resolution: 8 bits - Conversion speed: See Section 29. - Controls updating of DAC outputs by events: CPU writing, DSAC writing, comparator event, timer event, and PWM event - Generates masking signal so that comparator does not generate event before and after updating DAC
Comparator	- Response speed: See Section 29. - Enable/Disable control for hysteresis - Low power consumption mode - Event/Interrupt generation by edge detection - Event/Interrupt generation by level detection - Input sampling function and noise filtering function - Sampling interval setting function - Comparator output masking function by PWM signals - Event detection stopping function when DAC is updated - Return to standby mode by level detection - Output function to LUT

Table 24-2. Input Pin of Each Comparator

Unit No.	External Pin (−)	External Pin (+)
0	DAC0	ANEX0
1	DAC1	ANEX1
2	DAC2	ANEX2
3	DAC3	ANEX3
4	DAC4	ANEX4
5	DAC5	ANEX5

24.1.1. Comparator Control

The input signals to the high-speed comparator can be set, respectively. The output of the high-speed comparator can be used for the interrupt request and the trigger events for other modules. Figure 24-1 shows the comparator block diagram.

The comparators have the following functions.

- Comparator outputs have a glitch filter. The filter period can be selected from 1 cycle to 4 cycles. The period of 1 cycle can be extended by up to 128 times by the prescaler setting.
- Comparator outputs can be masked for a period of 32 cycles from the timing that the DAC setting is updated.
- Comparator outputs can be masked using a PWM signal. The signal can be selected from 8 PWM signals.
- Whether to enable or disable comparator's hysteresis can be selected.
- Using the slow mode, which the response speed of comparators is decreased, the current consumption can be reduced.

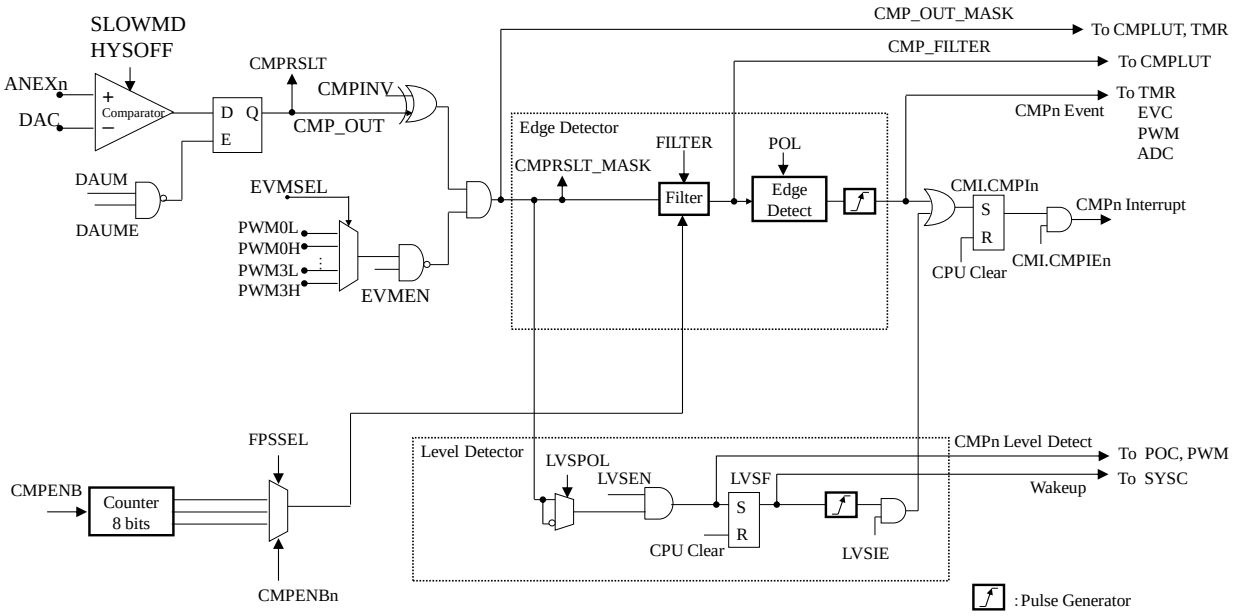


Figure 24-2. Block Diagram of Comparator

24.1.2. DAC Control

The output level of the 8-bit DAC to generate reference voltage is updated by the CPU, DSAC, comparator events, timer events, or PWM events.

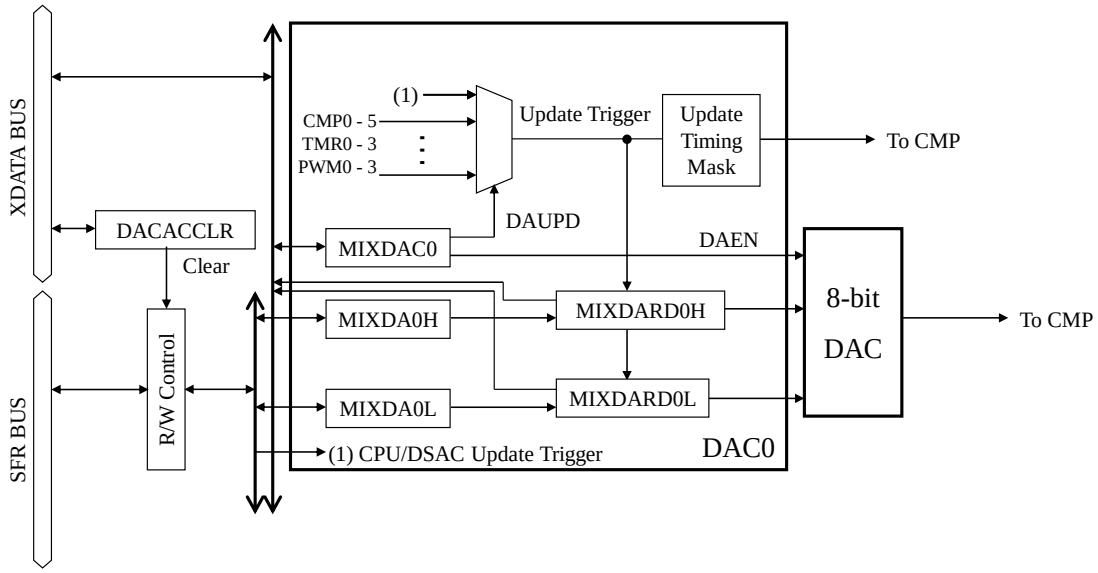


Figure 24-3. Block Diagram of 8-bit DAC

24.2. Register Descriptions

Table 24-3. List of XDATA BUS Registers

Symbol	Name	Address	Initial Value
MIXCMP0	Mix Comparator0 Configuration	0xF380	0x00
MIXCMS0	Mix Comparator0 Functional Select	0xF381	0x00
MIXCMR0	Mix Comparator0 Result	0xF382	0x00
MIXCMF0	Mix Comparator0 Function	0xF383	0x00
MIXCMEM0	Mix Comparator0 Event Mask	0xF384	0x00
MIXDAC0	Mix DAC0 Configuration	0xF3C0	0x00
MIXDARD0L	Mix DAC0 Read Data Low	0xF3C1	0x00
MIXDARD0H	Mix DAC0 Read Data High	0xF3C2	0x00
MIXDAFUNC0	Mix DAC0 Function	0xF3C3	0x00
DACACCLR0	Mix DAC0 Access Counter Clear Register	0xF3C4	0x00
MIXCMP1	Mix Comparator1 Configuration	0xF400	0x00
MIXCMS1	Mix Comparator1 Functional Select	0xF401	0x00
MIXCMR1	Mix Comparator1 Result	0xF402	0x0X
MIXCMF1	Mix Comparator1 Function	0xF403	0x00
MIXCMEM1	Mix Comparator1 Event Mask	0xF404	0x00
MIXDAC1	Mix DAC1 Configuration	0xF440	0x00
MIXDARD1L	Mix DAC1 Read Data Low	0xF441	0x00
MIXDARD1H	Mix DAC1 Read Data High	0xF442	0x00
MIXDAFUNC1	Mix DAC1 Function	0xF443	0x00
DACACCLR1	Mix DAC1 Access Counter Clear Register	0xF444	0x00
MIXCMP2	Mix Comparator2 Configuration	0xF480	0x00
MIXCMS2	Mix Comparator2 Functional Select	0xF481	0x00
MIXCMR2	Mix Comparator2 Result	0xF482	0x0X
MIXCMF2	Mix Comparator2 Function	0xF483	0x00
MIXCMEM2	Mix Comparator2 Event Mask	0xF484	0x00
MIXDAC2	Mix DAC2 Configuration	0xF4C0	0x00
MIXDARD2L	Mix DAC2 Read Data Low	0xF4C1	0x00
MIXDARD2H	Mix DAC2 Read Data High	0xF4C2	0x00
MIXDAFUNC2	Mix DAC2 Function	0xF4C3	0x00
DACACCLR2	Mix DAC2 Access Counter Clear Register	0xF4C4	0x00
MIXCMP3	Mix Comparator3 Configuration	0xF500	0x00
MIXCMS3	Mix Comparator3 Functional Select	0xF501	0x00
MIXCMR3	Mix Comparator3 Result	0xF502	0x0X
MIXCMF3	Mix Comparator3 Function	0xF503	0x00

Symbol	Name	Address	Initial Value
MIXCMEM3	Mix Comparator3 Event Mask	0xF504	0x00
MIXDAC3	Mix DAC3 Configuration	0xF540	0x00
MIXDARD3L	Mix DAC3 Read Data Low	0xF541	0x00
MIXDARD3H	Mix DAC3 Read Data High	0xF542	0x00
MIXDAFUNC3	Mix DAC3 Function	0xF543	0x00
DACACCLR3	Mix DAC3 Access Counter Clear Register	0xF544	0x00
MIXCMP4	Mix Comparator4 Configuration	0xED80	0x00
MIXCMS4	Mix Comparator4 Functional Select	0xED81	0x00
MIXCMR4	Mix Comparator4 Result	0xED82	0x0X
MIXCMF4	Mix Comparator4 Function	0xED83	0x00
MIXCMEM4	Mix Comparator4 Event Mask	0xED84	0x00
MIXDAC4	Mix DAC4 Configuration	0xEDC0	0x00
MIXDARD4L	Mix DAC4 Read Data Low	0xEDC1	0x00
MIXDARD4H	Mix DAC4 Read Data High	0xEDC2	0x00
MIXDAFUNC4	Mix DAC4 Function	0xEDC3	0x00
DACACCLR4	Mix DAC4 Access Counter Clear Register	0xEDC4	0x00
MIXCMP5	Mix Comparator5 Configuration	0xEE00	0x00
MIXCMS5	Mix Comparator5 Functional Select	0xEE01	0x00
MIXCMR5	Mix Comparator5 Result	0xEE02	0x0X
MIXCMF5	Mix Comparator5 Function	0xEE03	0x00
MIXCMEM5	Mix Comparator5 Event Mask	0xEE04	0x00
MIXDAC5	Mix DAC5 Configuration	0xEE40	0x00
MIXDARD5L	Mix DAC5 Read Data Low	0xEE41	0x00
MIXDARD5H	Mix DAC5 Read Data High	0xEE42	0x00
MIXDAFUNC5	Mix DAC5 Function	0xEE43	0x00
DACACCLR5	Mix DAC5 Access Counter Clear Register	0xEE44	0x00

Table 24-4. List of SFR BUS Registers

Symbol	Name	Address	Initial Value
CMI0	Mix Comparator Interrupt0	0xF3	0x00
CMI1	Mix Comparator Interrupt1	0x9D	0x00
MIXDA0L	Mix DAC0 Data Low	0x96	0x00
MIXDA0H	Mix DAC0 Data High	0x96	0x00
MIXDA1L	Mix DAC1 Data Low	0x95	0x00
MIXDA1H	Mix DAC1 Data High	0x95	0x00
MIXDA2L	Mix DAC2 Data Low	0x92	0x00
MIXDA2H	Mix DAC2 Data High	0x92	0x00
MIXDA3L	Mix DAC3 Data Low	0x93	0x00
MIXDA3H	Mix DAC3 Data High	0x93	0x00
MIXDA4L	Mix DAC4 Data Low	0x8D	0x00
MIXDA4H	Mix DAC4 Data High	0x8D	0x00
MIXDA5L	Mix DAC5 Data Low	0x8E	0x00
MIXDA5H	Mix DAC5 Data High	0x8E	0x00

24.2.1. MIXCMPn (Mix Comparator n Configuration) (n = 0 to 5)

Register	MIXCMP0	Mix Comparator0 Configuration	Address	0xF380
Register	MIXCMP1	Mix Comparator1 Configuration	Address	0xF400
Register	MIXCMP2	Mix Comparator2 Configuration	Address	0xF480
Register	MIXCMP3	Mix Comparator3 Configuration	Address	0xF500
Register	MIXCMP4	Mix Comparator4 Configuration	Address	0xED80
Register	MIXCMP5	Mix Comparator5 Configuration	Address	0xEE00

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMPENB	R/W	0	Comparator enable 0: Comparator function is disabled 1: Comparator function is enabled	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

24.2.2. MIXCMSn (Mix Comparator n Functional Select) (n = 0 to 5)

When MIXCMPn.CMPENB = 0, set the MIXCMSn register.

Register	MIXCMS0	Mix Comparator0 Functional Select	Address	0xF381
Register	MIXCMS1	Mix Comparator1 Functional Select	Address	0xF401
Register	MIXCMS2	Mix Comparator2 Functional Select	Address	0xF481
Register	MIXCMS3	Mix Comparator3 Functional Select	Address	0xF501
Register	MIXCMS4	Mix Comparator4 Functional Select	Address	0xED81
Register	MIXCMS5	Mix Comparator5 Functional Select	Address	0xEE01

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	LVSEN	R/W	0	Level detection enable 0: Level detection function is disabled 1: Level detection function is enabled	
5	LVSIE	R/W	0	Level detection interrupt enable 0: Level detection interrupt function is disabled 1: Level detection interrupt function is enabled	
4	LVSPOL	R/W	0	Level detection polarity 0: Low level 1: High level	
3	POL	R/W	0	Edge polarity 00: Not detected (same operation as MIXCMPn.CMPENB = 0) 01: Detected by falling edge 10: Detected by rising edge 11: Detected by both rising and falling edges	
2		R/W	0		
1	FILTER	R/W	0	Glitch filter 00: Glitch filter is not used 01: 1 cycle 10: 2 cycles 11: 4 cycles One cycle of the glitch filter is defined by the MIXCMFn.FPSSEL bits.	
0		R/W	0		

24.2.3. MIXCMRn (Mix Comparator n Result) (n = 0 to 5)

Register	MIXCMR0	Mix Comparator0 Result	Address 0xF382
Register	MIXCMR1	Mix Comparator1 Result	Address 0xF402
Register	MIXCMR2	Mix Comparator2 Result	Address 0xF482
Register	MIXCMR3	Mix Comparator3 Result	Address 0xF502
Register	MIXCMR4	Mix Comparator4 Result	Address 0xED82
Register	MIXCMR5	Mix Comparator5 Result	Address 0xEE02

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CRSLTM	R	x	CMP_OUT_MASK signal monitor MASKED_CMP_OUT is monitored.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	LVSF	R/C	0	Level detection flag Read 0: Level is not detected Read 1: Level is detected Write 0: No change Write 1: The bit is cleared The bit can be cleared by writing 1 to the bit when the comparator is not detecting the level.	
0	CMPRSLT	R	x	CMP_OUT signal monitor CMP_OUT is monitored.	

24.2.4. MIXCMFn (Mix Comparator n Function) (n = 0 to 5)

Register	MIXCMF0	Mix Comparator0 Function	Address	0xF383
Register	MIXCMF1	Mix Comparator1 Function	Address	0xF403
Register	MIXCMF2	Mix Comparator2 Function	Address	0xF483
Register	MIXCMF3	Mix Comparator3 Function	Address	0xF503
Register	MIXCMF4	Mix Comparator4 Function	Address	0xED83
Register	MIXCMF5	Mix Comparator5 Function	Address	0xEE03

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	FPSSEL	R/W	0	Filter prescaler (Set frequency of signal for the glitch filter) 000: 1/1 001: 1/8 010: 1/16 011: 1/32 100: 1/64 101: 1/128 Other than above: Setting prohibited	
5		R/W	0		
4		R/W	0		
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	CMPINV	R/W	0	CMPRSLT signal (see Figure 24-2) inversion 0: CMPRSLT signal is not inverted 1: CMPRSLT signal is inverted	
1	HYSOFF	R/W	0	Comparator hysteresis 0: Hysteresis is set 1: Hysteresis is not set	
0	SLOWMD	R/W	0	Comparator slow mode enable 0: Slow mode is disabled 1: Slow mode is enabled	

24.2.5. MIXCMEMn (Mix Comparator n Event Mask) (n = 0 to 5)

Register	MIXCMEM0	Mix Comparator0 Event Mask	Address	0xF384
Register	MIXCMEM1	Mix Comparator1 Event Mask	Address	0xF404
Register	MIXCMEM2	Mix Comparator2 Event Mask	Address	0xF484
Register	MIXCMEM3	Mix Comparator3 Event Mask	Address	0xF504
Register	MIXCMEM4	Mix Comparator4 Event Mask	Address	0xED84
Register	MIXCMEM5	Mix Comparator5 Event Mask	Address	0xFE04

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DAUME	R/W	0	Mask enable for DAC updating 0: Masking function for DAC updating is disabled 1: Masking function for DAC updating is enabled	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	EVMEN	R/W	0	Event mask enable 0: Event mask is disabled 1: Event mask is enabled	
2	EVMSEL	R/W	0	Event mask (Selection of the PWM signal used to mask the comparator output) 000: PWM0L 001: PWM0H 010: PWM1L 011: PWM1H 100: PWM2L 101: PWM2H 110: PWM3L 111: PWM3H	
1		R/W	0		
0		R/W	0		

24.2.6. CMI0 (Mix Comparator Interrupt0)

Register	CMI0	Mix Comparator Interrupt0	Address	0xF3

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CMPIE3	R/W	0	Comparator3 interrupt enable 0: Interrupt is disabled 1: Interrupt is enabled	
6	CMPIE2	R/W	0	Comparator2 interrupt enable 0: Interrupt is disabled 1: Interrupt is enabled	
5	CMPIE1	R/W	0	Comparator1 interrupt enable 0: Interrupt is disabled 1: Interrupt is enabled	
4	CMPIE0	R/W	0	Comparator0 interrupt enable 0: Interrupt is disabled 1: Interrupt is enabled	
3	CMPI3	R/C	0	Comparator3 interrupt flag (The flag is generated regardless of the setting of the CMI0.CMPIE3 bit) Read 0: Interrupt signal is not detected Read 1: Interrupt signal is detected Write 0: No change Write 1: The bit is cleared	
2	CMPI2	R/C	0	Comparator2 interrupt flag (The flag is generated regardless of the setting of the CMI0.CMPIE2 bit) Read 0: Interrupt signal is not detected Read 1: Interrupt signal is detected Write 0: No change Write 1: The bit is cleared	
1	CMPI1	R/C	0	Comparator1 interrupt flag (The flag is generated regardless of the setting of the CMI0.CMPIE1 bit) Read 0: Interrupt signal is not detected Read 1: Interrupt signal is detected Write 0: No change Write 1: The bit is cleared	
0	CMPI0	R/C	0	Comparator0 interrupt flag (The flag is generated regardless of the setting of the CMI0.CMPIE0 bit) Read 0: Interrupt signal is not detected Read 1: Interrupt signal is detected Write 0: No change Write 1: The bit is cleared	

24.2.7. CMI1 (Mix Comparator Interrupt1)

Register	CMI1		Mix Comparator Interrupt1		Address	0x9D
Bit	Bit Name	R/W	Initial	Description		Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.		
6	Reserved	R	0	The read value is 0. The write value must always be 0.		
5	CMPIE5	R/W	0	Comparator5 interrupt enable 0: Interrupt is disabled 1: Interrupt is enabled		
4	CMPIE4	R/W	0	Comparator4 interrupt enable 0: Interrupt is disabled 1: Interrupt is enabled		
3	Reserved	R	0	The read value is 0. The write value must always be 0.		
2	Reserved	R	0	The read value is 0. The write value must always be 0.		
1	CMPI5	R/C	0	Comparator5 interrupt flag (The flag is generated regardless of the setting of the CMI1.CMPIE5 bit) Read 0: Interrupt signal is not detected Read 1: Interrupt signal is detected Write 0: No change Write 1: The bit is cleared		
0	CMPI4	R/C	0	Comparator4 interrupt flag (The flag is generated regardless of the setting of the CMI1.CMPIE4 bit) Read 0: Interrupt signal is not detected Read 1: Interrupt signal is detected Write 0: No change Write 1: The bit is cleared		

24.2.8. MIXDACn (Mix DAC n Configuration) (n = 0 to 5)

Register	MIXDAC0	Mix DAC0 Configuration	Address	0xF3C0
Register	MIXDAC1	Mix DAC1 Configuration	Address	0xF440
Register	MIXDAC2	Mix DAC2 Configuration	Address	0xF4C0
Register	MIXDAC3	Mix DAC3 Configuration	Address	0xF540
Register	MIXDAC4	Mix DAC4 Configuration	Address	0xFDC0
Register	MIXDAC5	Mix DAC5 Configuration	Address	0xFE40

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DAEN	R/W	0	DAC enable 0: DAC function is disabled 1: DAC function is enabled	
6	DFORM	R/W	0	Data register format 0: MSB side is used (dddd_dddd_0000_0000) 1: LSB side is used (0000_0000_dddd_dddd) To read from/write to the MIXDAnH register, set DFORM = 0. To read from/write to the MIXDAnL register, set DFORM = 1.	
5	DAUPD	R/W	0	DAC update timing (The trigger event to update DAC is selected.) 000000: The MIXDAnH register is updated by CPU or DSAC 001000: CMP0 (pulse) 001001: CMP1 (pulse) 001010: CMP2 (pulse) 001011: CMP3 (pulse) 001100: Timer0_CMA (pulse) 001101: Timer0_CMB (pulse) 001110: Timer1_CMA (pulse) 001111: Timer1_CMB (pulse) 010000: PWM0_Event0 (pulse) 010001: PWM0_Event1 (pulse) 010010: PWM1_Event0 (pulse) 010011: PWM1_Event1 (pulse) 010100: PWM2_Event0 (pulse) 010101: PWM2_Event1 (pulse) 010110: PWM3_Event0 (pulse) 010111: PWM3_Event1 (pulse) 011000: CMP4 (pulse) 011001: CMP5 (pulse) 011010: Timer2_CMA (pulse) 011011: Timer2_CMB (pulse) 011100: Timer3_CMA (pulse) 011101: Timer3_CMB (pulse) Other than above: Setting prohibited	
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

24.2.9. MIXDAnL (Mix DAC n Data Low) (n = 0 to 5)

Register	MIXDA0L	Mix DAC0 Data Low	Address	0x96
Register	MIXDA1L	Mix DAC1 Data Low	Address	0x95
Register	MIXDA2L	Mix DAC2 Data Low	Address	0x92
Register	MIXDA3L	Mix DAC3 Data Low	Address	0x93
Register	MIXDA4L	Mix DAC4 Data Low	Address	0x8D
Register	MIXDA5L	Mix DAC5 Data Low	Address	0x8E

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DADATA	R/W	0	Lower side of the input data of DAC See description on the MIXDAnH register.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

24.2.10. MIXDAnH (Mix DAC n Data High) (n = 0 to 5)

Register	MIXDA0H	Mix DAC0 Data High	Address	0x96
Register	MIXDA1H	Mix DAC1 Data High	Address	0x95
Register	MIXDA2H	Mix DAC2 Data High	Address	0x92
Register	MIXDA3H	Mix DAC3 Data High	Address	0x93
Register	MIXDA4H	Mix DAC4 Data High	Address	0x88
Register	MIXDA5H	Mix DAC5 Data High	Address	0xB8

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DADATA	R/W	0	Higher side of the input data of DAC ● When DFORM = 0: The data is stored in bits 15 to 8. 0x00 is stored in bits 7 to 0. ● When DFORM = 1: The data is stored in bits 7 to 0. 0x00 is stored in bits 15 to 8. ● When MIXDACn.DAUPD = 0b000000: The DA output is updated when the MIXDAnH register is set by the CPU or the DSAC. (For the CPU, this is set by the second access. For the DSAC, this is set by the second access in the byte access mode or by word access.) ● When MIXDACn.DAUPD bits are set other than 0b000000: The DA output is updated by the issue of the specified updating trigger. Before the updating trigger is issued, the MIXDAnL and MIXDAnH registers must be set.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

24.2.11. MIXDARDnL (Mix DAC n Read Data Low) (n = 0 to 5)

After data is written to both the MIXDAnH and MIXDAnL registers, when the trigger determined by the MIXDACn.DAUPD bits is detected, the MIXDARDnL and MIXDARDnH registers are updated.

Register	MIXDARD0L	Mix DAC0 Read Data Low	Address	0xF3C1
Register	MIXDARD1L	Mix DAC1 Read Data Low	Address	0xF441
Register	MIXDARD2L	Mix DAC2 Read Data Low	Address	0xF4C1
Register	MIXDARD3L	Mix DAC3 Read Data Low	Address	0xF541
Register	MIXDARD4L	Mix DAC4 Read Data Low	Address	0xFDC1
Register	MIXDARD5L	Mix DAC5 Read Data Low	Address	0xFE41

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DARD	R	0	Lower side of the input read data of DAC See description on the MIXDARDnH register.	
6		R	0		
5		R	0		
4		R	0		
3		R	0		
2		R	0		
1		R	0		
0		R	0		

24.2.12. MIXDARDnH (Mix DAC n Read Data High) (n = 0 to 5)

After data is written to both the MIXDAnH and MIXDAnL registers, when the trigger determined by the MIXDACn.DAUPD bits is detected, the MIXDARDnL and MIXDARDnH registers are updated.

Register	MIXDARD0H	Mix DAC0 Read Data High	Address	0xF3C2
Register	MIXDARD1H	Mix DAC1 Read Data High	Address	0xF442
Register	MIXDARD2H	Mix DAC2 Read Data High	Address	0xF4C2
Register	MIXDARD3H	Mix DAC3 Read Data High	Address	0xF542
Register	MIXDARD4H	Mix DAC4 Read Data High	Address	0xFDC2
Register	MIXDARD5H	Mix DAC5 Read Data High	Address	0xFE42

Bit	Bit Name	R/W	Initial	Description	Remarks
7	DARD	R	0	Higher side of the input data of DAC ● When MIXDACn.DFORM = 0: The data is read from bits 15 to 8. 0x00 is read from bits 7 to 0. ● When MIXDACn.DFORM = 1: The data is read from bits 7 to 0. 0x00 is read from bits 15 to 8.	
6		R	0		
5		R	0		
4		R	0		
3		R	0		
2		R	0		
1		R	0		
0		R	0		

24.2.13. DACACCLRn (Mix DAC n Access Counter Clear Register) (n = 0 to 5)

Register	DACACCLR0	Mix DAC0 Access Counter Clear Register	Address	0xF3C4
Register	DACACCLR1	Mix DAC1 Access Counter Clear Register	Address	0xF444
Register	DACACCLR2	Mix DAC2 Access Counter Clear Register	Address	0xF4C4
Register	DACACCLR3	Mix DAC3 Access Counter Clear Register	Address	0xF544
Register	DACACCLR4	Mix DAC4 Access Counter Clear Register	Address	0xFDC4
Register	DACACCLR5	Mix DAC5 Access Counter Clear Register	Address	0xFE44

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CPUCLR	W	0	Clears the CPU access counter for the DAC conversion data register Write 0: No change Write 1: CPU access counter register is cleared (CPU SFR access counter is cleared)	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

24.2.14. MIXDAFUNCn (Mix DAC n Function) (n = 0 to 5)

Register	MIXDAFUNC0	Mix DAC0 Function	Address	0xF3C3
Register	MIXDAFUNC1	Mix DAC1 Function	Address	0xF443
Register	MIXDAFUNC2	Mix DAC2 Function	Address	0xF4C3
Register	MIXDAFUNC3	Mix DAC3 Function	Address	0xF543
Register	MIXDAFUNC4	Mix DAC4 Function	Address	0xFDC3
Register	MIXDAFUNC5	Mix DAC5 Function	Address	0xFE43

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	SELS	R/W	0	Selection of signed/unsigned DAC data 0: Unsigned DAC data 1: Signed DAC data	

24.3. Operation

In order to prevent an interrupt from being generated by mistake, clear the interrupt flag before enabling the interrupt by the comparator.

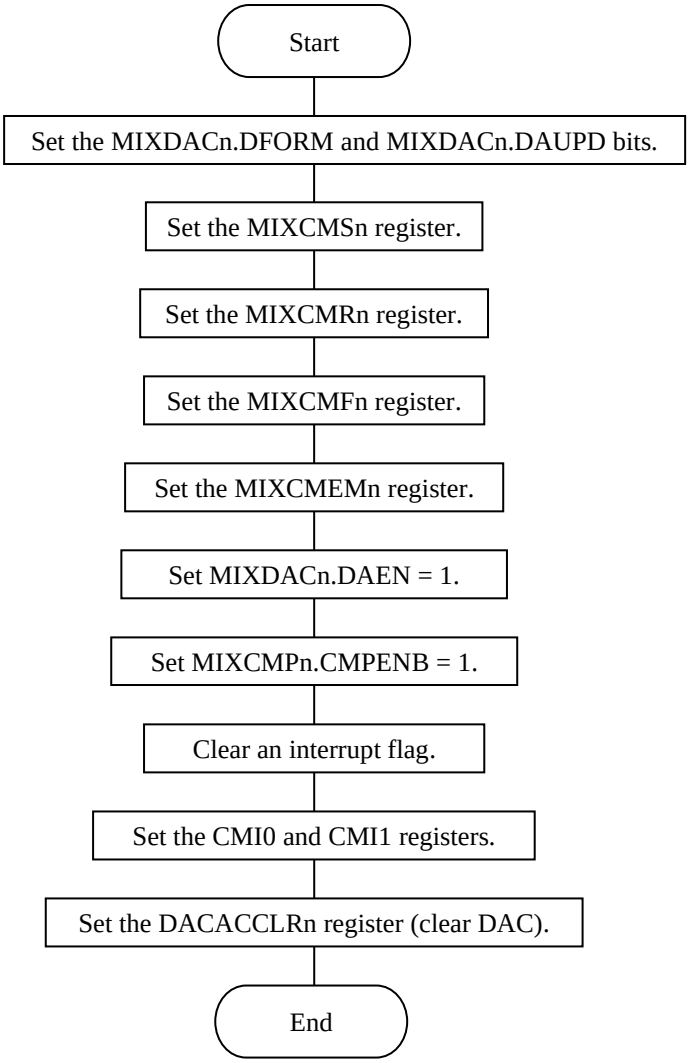
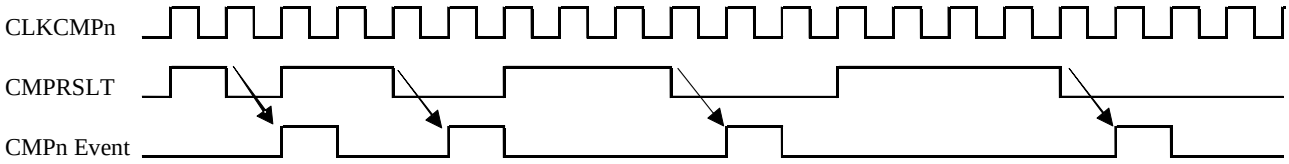
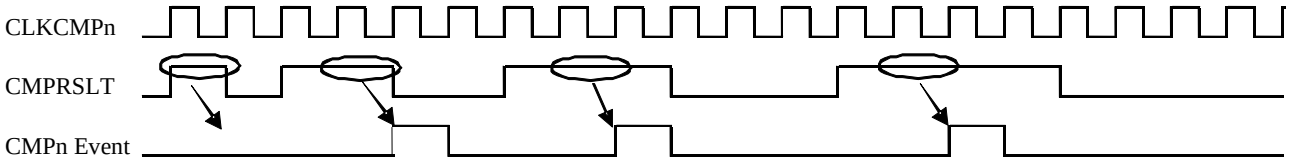


Figure 24-4. Operation Flowchart

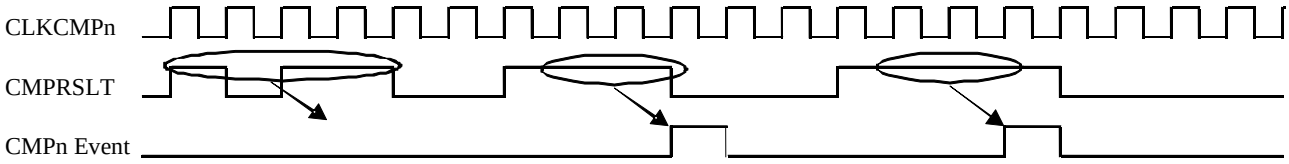


(a) Negative Edge Detection without Glitch Filter



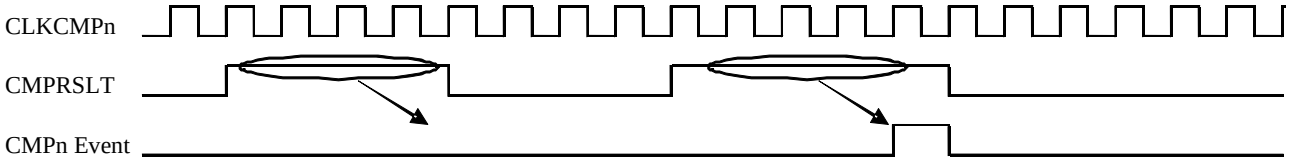
If the pulse length of the CMPRSLT is 1 cycle of the CLKCMPn, the CMPn event is not generated.

(b) Positive Edge Detection with Glitch Filter (1 Cycle)



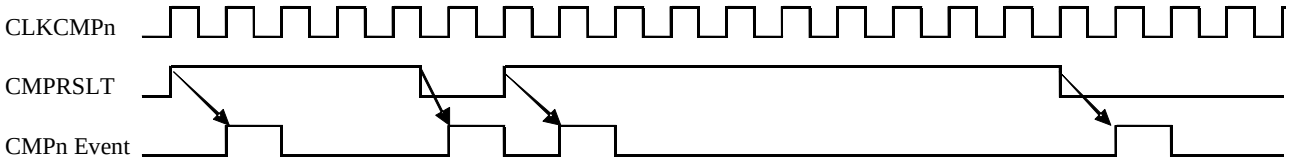
If the pulse length of the CMPRSLT is 1 cycle or 2 cycles of the CLKCMPn, the CMPn event is not generated.

(c) Positive Edge Detection with Glitch Filter (2 Cycles)



If the pulse length of the CMPRSLT is <4 cycles of the CLKCMPn, the CMPn event is not generated.

(d) Positive Edge Detection with Glitch Filter (4 Cycles)



(e) Both-edge Detection without Glitch Filter

Figure 24-5. Examples of CMPn Level Detection Generation

24.3.1. Activation and Stop

When MIXDACn.DAEN = 1, the DAC function is enabled. When MIXDACn.DAEN = 0, the DAC function is disabled.

When MIXCMPn.CMPENB = 1, the comparator is enabled. When the MIXCMPn.CMPENB = 0, the comparator is disabled.

24.3.2. Setting and Updating of DAC

Write the DA conversion value of 8-bit digital value to the MIXDAnL.DADATA and MIXDAnH.DADATA bits (total 16 bits). To write 8-bit digital value to the MIXDAnH register, set MIXDACn.DFORM = 0. 0x00 is automatically written to the MIXDAnL register. On the other hand, to write 8-bit digital value to the MIXDAnL register, set MIXDACn.DFORM = 1. 0x00 is automatically written to the MIXDAnH register.

To read out the input value of the DAC, read the MIXDARDnL.DARDx bit or the MIXDARDnH.DARDx bit. 0x00 is read from the register not selected by the MIXDACn.DFORM bit.

The system is initially set up so that the output value of the DAC is updated when the value is overwritten by the CPU or the DSAC. The trigger to update the DAC output value can be changed by the MIXDACn.DAUPD bits.

To clear the CPU access counter of the MIXDAnL/H register, set DACACCLR.CPUCLR = 1 after writing to the MIXDAnL register. After the access counter is cleared, the MIXDAnL register can be accessed again.

The format of the DAC data to be written to/read from can be selected from either signed or unsigned. To set the format of unsigned or signed, set the MIXDAFUNCn.SELS bit to 0 or 1, respectively. However, when the MIXDACn.DFORM bit is 0 (MSB side), the lower 8 bits are discarded in the case of writing, and 0x00 is read from the lower 8 bits in the case of reading.

The Comparator output can be masked for a fixed period from the timing to update the DAC output value. When the MIXCMEMn.DAUME bit is set to 1, masking function is enabled. The masking period is 32 cycles from the clock cycle next to the timing at which the update trigger becomes 1. If an update trigger is generated during the masking period, counting is started again from the next cycle, and the masking period is extended.

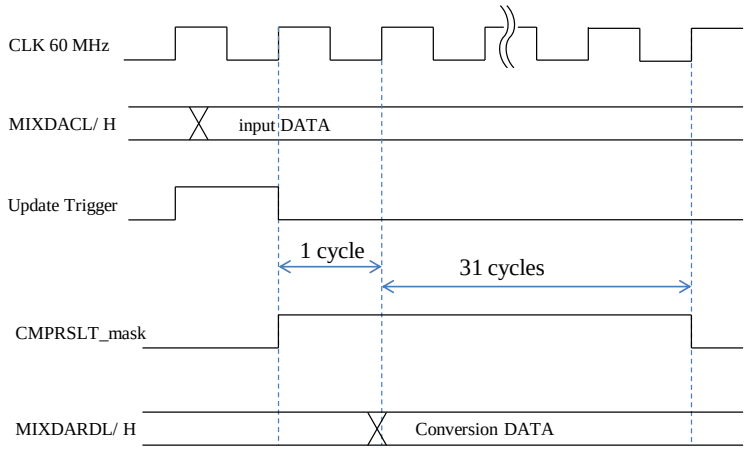


Figure 24-6. Comparator Output Masking Operation for Updating DAC Output

24.3.3. Comparator Operation Mode

Using the slow mode decreases the response speed of comparators, and reduces the current consumption. To enable the slow mode, set MIXCMFn.SLOWMD = 1.

In addition, whether to enable or disable comparator's hysteresis can be selected. When the MIXCMFn.HYSOFF bit is set to 1, the hysteresis is disabled.

24.3.4. Comparator Output Control

When the MIXCMFn.CMPINV bit is set to 1, the comparator output (the CMPRSLT signal shown in Figure 24-2) is inverted.

The comparator output can be masked using a PWM signal. When the MIXCMEMn.EVMEN bit is set to 1, this function is enabled. There are 8 usable PWM signals that are defined by the MIXCMEMn.EVMSEL bits. The comparator output masked by an event can be monitored by reading the MIXCMRn.CRSLTM bit.

Moreover, the comparator output masked by the PWM signal can be masked for a fixed period of 32 cycles when the output value of the DAC is updated. This function is enabled by setting the MIXCMEMn.DAUME bit to 1.

24.3.5. Interrupt and Event Generation

When the MIXCMSn.LVSEN bit is set to 1, the level detection function can be used. In addition, the detection level is defined by the MIXCMSn.LVSPOL bit. When the bit is set to1 or 0, the detection level is high or low, respectively.. When the level detection is occurred, the MIXCMRn.LVSF bit is set to 1. To clear this value, write 1 to the MIXCMRn.LVSF bit.

The level detection signal that is output to the POC, PWM, and SYSC can be used as a signal for the PWM output control or the returning from the standby state.

The polarity of the edge detection is determined by the MIXCMSn.POL bits. The edge polarity can be selected from rising (0b01), falling (0b10), and both rising and falling (0b11).

When detecting the edge, the glitch filter can be used. The filtering period is defined by the MIXCMSn.FILTER bits. The filtering period can be selected from filter disabled (0b00), 1 cycle (0b01), 2 cycles (0b10), and 4 cycles (0b11). In addition, the filter is equipped with the prescaler. Therefore, the width of one cycle of the filter period can be changed by the MIXCMFn.FPSSEL bits. The edge detection signal can be output to the DSAC, PWM, and TMR.

The edge detection signal and the level detection signal of the comparator can be used as an interrupt signal. To use the level detection signal of comparator for the interrupt signal, set the MIXCMSn.LVSIE bit to 1. The comparator to generate the interrupt signal is defined by the CMPIEn bit of the CMI0 or CMI1 register. When the interrupt signal is output, the CMPIn bit of the CMI0 or CMI1 register is set to 1 regardless of the setting of the CMPIEn bit. To clear this value, write 1 to the same bit.

24.3.6. Output to LUT

The monitoring signal of CMP_OUT (the MIXCMRn.CMPRSLT bit) or the signal after passing the noise filter (CMP_FILLTER) can be output to LUT.

24.4. Usage Notes and Restrictions

To read the MIXDAnL/H register by 8-bit access, make sure to set MIXDACn.DFORM = 1 (LSB side). The register cannot be read by setting MIXDACn.DFORM = 0 (MSB side).

25. Temperature Sensor (TEMP)

25.1. Overview

The LSI has the temperature sensor (TEMP) that generates a voltage according to the junction temperature. The voltage is measured by the ADC. For the output voltage of the TEMP, see Section 29.

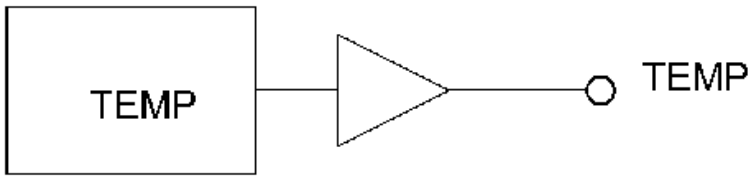


Figure 25-1. TEMP Block Diagram

25.2. Register Descriptions

25.2.1. TEMP (Temperature Sensor Control)

Register	TEMP		Temperature Sensor Control	Address	0xFFC1
Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	TEMPE	R/W	0	TEMP enable 0: TEMP is disabled 1: TEMP is enabled	

26. PWM Output Controller (POC)

26.1. Overview

The PWM output controller (POC) fixes the PWM output pin to the preset pin state by detecting the event from the CMP or the CMPLUT.

Figure 26-1 shows the POC block diagram.

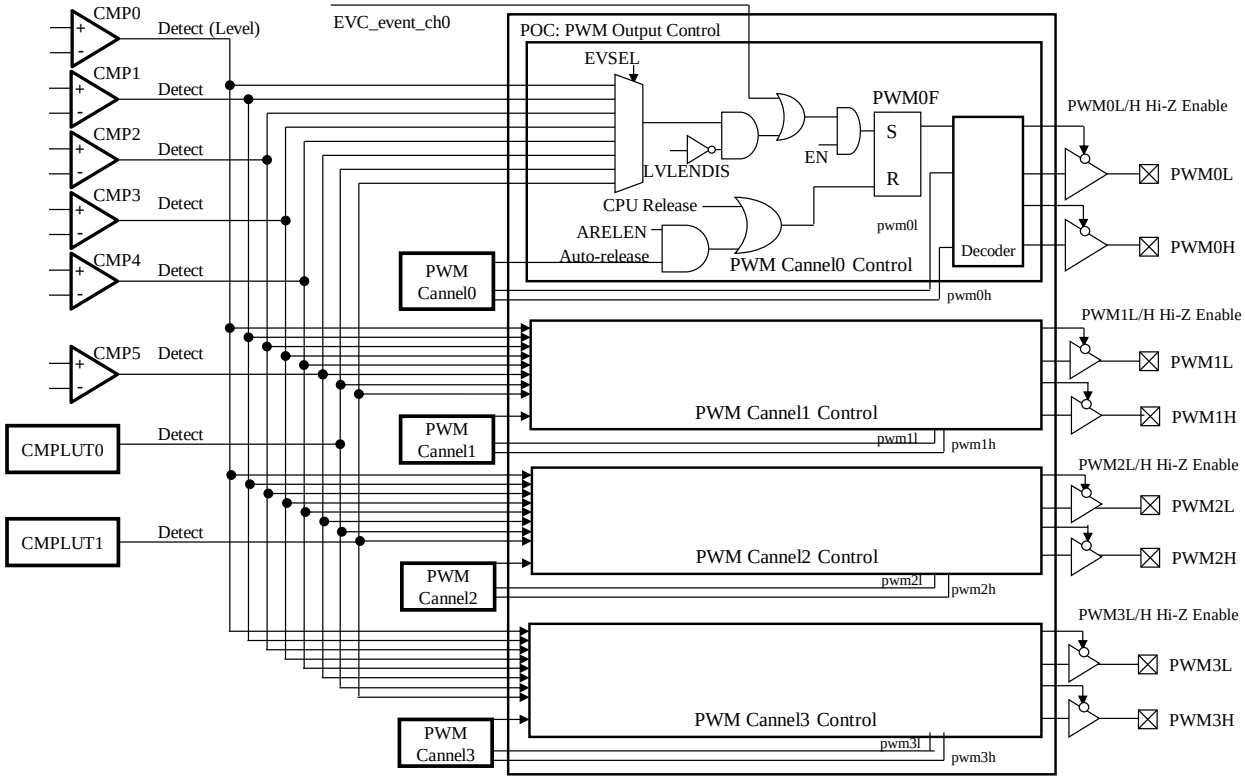


Figure 26-1. POC Block Diagram

26.2. Register Descriptions

Table 26-1. List of Registers

Symbol	Name	Address	Initial Value
POCCR0	POC Control Register0	0xFD80	0x00
POCCR1	POC Control Register1	0xFD81	0x00
POCCR2	POC Control Register2	0xFD82	0x00
POCCR3	POC Control Register3	0xFD83	0x00
POCSTS	POC Status Register	0xFD88	0x00
POCBAS	POC BUS I/F Access Status Register	0xFD8C	0x00
POCOCR0	POC Output Control Register0	0xFD90	0x00
POCOCR1	POC Output Control Register1	0xFD91	0x00
POCOCR2	POC Output Control Register2	0xFD92	0x00
POCOCR3	POC Output Control Register3	0xFD93	0x00
POCTRG	POC CPU Trigger Register	0xFD98	0x00
POCDTC0	POC Dead Time Control Register0	0xFDA0	0x00
POCDTC1	POC Dead Time Control Register1	0xFDA1	0x00
POCDTC2	POC Dead Time Control Register2	0xFDA2	0x00
POCDTC3	POC Dead Time Control Register3	0xFDA3	0x00
POCDTP0	POC Dead Time Period Register0	0xFDA8	0x00
POCDTP1	POC Dead Time Period Register1	0xFDA9	0x00
POCDTP2	POC Dead Time Period Register2	0xFDAA	0x00
POCDTP3	POC Dead Time Period Register3	0xFDAB	0x00

26.2.1. POCCRn (POC Control Register n) (n = 0 to 3)

Register	POCCR0	POC Control Register0	Address	0xFD80
Register	POCCR1	POC Control Register1	Address	0xFD81
Register	POCCR2	POC Control Register2	Address	0xFD82
Register	POCCR3	POC Control Register3	Address	0xFD83

Bit	Bit Name	R/W	Initial	Description	Remarks
7	EN	R/W	0	POC control enable 0: PWMnL/H control by POC is disabled 1: PWMnL/H control by POC is enabled If the POCSTS.PWMnF bit is set when POCCRn.EN = 1, the PWMnL/H pin is put into the preset pin state.	
6	ARELEN	R/W	0	Automatic release of high impedance of PWMn output pin 0: Not automatically released (CPU/EPU release); can only be released by CPU or EPU 1: Automatically released in either of the cases where: − The counter matches with the CMP_MAX value of the PWM in up mode − The counter matches with the CMP_MIN value of the PWM in up-down mode The automatic release function is not available for the control delay addition function.	
5	CONFHn0	R/W	0	PWMnH output control The PWMnH pin control is determined by this bit and the POCOCRn.CONFLn1 bit (see also Section 26.2.4). If CONFHn = {POCOCRn.CONFHn1, POCCRn.CONFHn0} CONFHn = 00: Sets PWMnH to a high impedance state after receiving an event CONFHn = 01: Does not control PWMnH even after receiving an event CONFHn = 10: Sets PWMnH output to low level after receiving an event CONFHn = 11: Sets PWMnH output to high level after receiving an event	
4	CONFLn0	R/W	0	PWMnL output control The PWMnL pin control is determined by this bit and the POCOCRn.CONFLn1 bit (see also Section 26.2.4). If CONFLn = {POCOCRn.CONFLn1, POCCRn.CONFLn0} CONFLn = 00: Sets PWMnL to a high impedance state after receiving an event CONFLn = 01: Does not control PWMnL even after receiving an event CONFLn = 10: Sets PWMnL output to low level after receiving an event CONFLn = 11: Sets PWMnL output to high level after receiving an event	

Register	POCCR0	POC Control Register0	Address	0xFD80
Register	POCCR1	POC Control Register1	Address	0xFD81
Register	POCCR2	POC Control Register2	Address	0xFD82
Register	POCCR3	POC Control Register3	Address	0xFD83

Bit	Bit Name	R/W	Initial	Description	Remarks
3	LVLEVDIS	R/W	0	Determining whether or not to accept the level event selected by the EVSEL bit 0: Level event acceptance is enabled 1: Level event acceptance is disabled	
2	EVSEL	R/W	0	Event of PWMn output pin (high impedance) 000: CMP0 is selected 001: CMP1 is selected 010: CMP2 is selected 011: CMP3 is selected 100: CMP4 is selected 101: CMP5 is selected 110: CMPLUT0 is selected 111: CMPLUT1 is selected	
1		R/W	0		
0		R/W	0		

26.2.2. POCSTS (POC Status Register)

Register	POCSTS	POC Status Register	Address	0xFD88

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	PWM3F	R/C	0	Control status of PWM3 output pin Read 0: PWM3L/H is controlled by PWM3 Read 1: PWM3L/H is controlled by POC Write 0: No change Write 1: PWM3L/H control by POC is cleared	
2	PWM2F	R/C	0	Control status of PWM2 output pin Read 0: PWM2L/H is controlled by PWM2 Read 1: PWM2L/H is controlled by POC Write 0: No change Write 1: PWM2L/H control by POC is cleared	
1	PWM1F	R/C	0	Control status of PWM1 output pin Read 0: PWM1L/H is controlled by PWM1 Read 1: PWM1L/H is controlled by POC Write 0: No change Write 1: PWM1L/H control by POC is cleared	
0	PWM0F	R/C	0	Control status of PWM0 output pin Read 0: PWM0L/H is controlled by PWM0 Read 1: PWM0L/H is controlled by POC Write 0: No change Write 1: PWM0L/H control by POC is cleared	

26.2.3. POCBAS (POC BUS I/F Access Status Register)

Register	POCBAS	POC BUS I/F Access Status Register	Address	0xFD8C

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	POCXACS	R	0	XDATA BUS access status 0: Waiting Writing to the POCSTS and POCTRG registers is allowed 1: Accessing Writing to the POCSTS or POCTRG register is prohibited	

26.2.4. POCOCRn (POC Output Control Register n) (n = 0 to 3)

Register	POCOCR0	POC Output Control Register0	Address	0xFD90
Register	POCOCR1	POC Output Control Register1	Address	0xFD91
Register	POCOCR2	POC Output Control Register2	Address	0xFD92
Register	POCOCR3	POC Output Control Register3	Address	0xFD93

Bit	Bit Name	R/W	Initial	Description	Remarks
7	PLINV	R/W	0	Event level inversion 0: Pin control by POC starts at positive level 1: Pin control by POC starts at negative level Events which are output from the event controller (EVC) are independent of the setting of this register.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	CONFHn1	R/W	0	PWMnH output control The PWMnH pin control is determined by this bit and the POCCRn.CONFHn0 bit. For more details, see Section 26.2.1.	
0	CONFLn1	R/W	0	PWMnL output control The PWMnL pin control is determined by this bit and the POCCRn.CONFLn0 bit. For more details, see Section 26.2.1.	

26.2.5. POCTRG (POC CPU Trigger Register)

Register	POCTRG	POC CPU Trigger Register	Address	0xFD98

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	TRG3	W	0	CPU trigger for POC operation of PWM3L/H Write 0: No change Write 1: POC operation of PWM3 is executed The read value is always 0.	
2	TRG2	W	0	CPU trigger for POC operation of PWM3L/H Write 0: No change Write 1: POC operation of PWM2 is executed The read value is always 0.	
1	TRG1	W	0	CPU trigger for POC operation of PWM1L/H Write 0: No change Write 1: POC operation of PWM1 is executed The read value is always 0.	
0	TRG0	W	0	CPU trigger for POC operation of PWM0L/H Write 0: No change Write 1: POC operation of PWM0 is executed The read value is always 0.	

26.2.6. POCDTCn (POC Dead Time Control Register n) (n = 0 to 3)

Register	POCDTC0	POC Dead Time Control Register0	Address	0xFDA0
Register	POCDTC1	POC Dead Time Control Register1	Address	0xFDA1
Register	POCDTC2	POC Dead Time Control Register2	Address	0xFDA2
Register	POCDTC3	POC Dead Time Control Register3	Address	0xFDA3

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CLRWAIT	R/W	0	Delay function of POC control clear 0: Output control by POC is finished at the reception of a release signal 1: Output control by POC is finished after the completion of dead time counting, if a release signal is received during POC dead time count The bit defines the setting for POC operation at the reception of a release signal during a dead time count.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	HDEN	R/W	0	PWMnH dead time count enable 0: No delay time is added 1: A delay time is added When PWMnH is set to high level by the POC, the bit determines whether or not to add a delay to the control timing at the reception of an event.	
0	LDEN	R/W	0	PWMnL dead time count enable 0: No delay time is added 1: A delay time is added When PWMnL is set to high level by the POC, the bit determines whether or not to add a delay to the control timing at the reception of an event.	

26.2.7. POCDTPn (POC Dead Time Period Register n) (n = 0 to 3)

Register	POCDTP0	POC Dead Time Period Register0	Address	0xFDA8
Register	POCDTP1	POC Dead Time Period Register1	Address	0xFDA9
Register	POCDTP2	POC Dead Time Period Register2	Address	0xFDAA
Register	POCDTP3	POC Dead Time Period Register3	Address	0xFDAB

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	DTP	R/W	0	POC dead time period The bit determines the amount of delay time to be added. The delay time is defined as follows: Delay time = (DTP + 2) × (8 × CLKPWM period)	
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

26.3. Operation

26.3.1. Basic Operation

When a selected event is generated, the POC fixes the PWMnL/H (n = 0 to 3) output pin to the preset pin state.

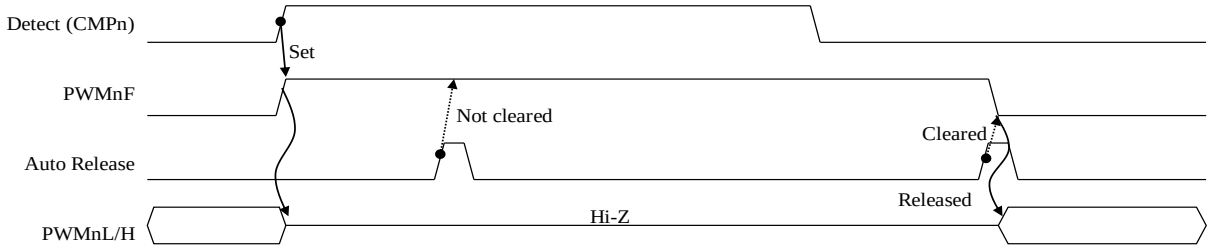
The POCCRn.EVSEL bit selects the trigger event from the level detection events of the CMP0 to the CMP5 and CMPLUT0/1. The POCSTS.PWMnF bit is set to 1 by the selected event, the event from the EPU, or the writing to the POCTRG register from the CPU or the EPU. To disable the level event selected by the POCCRn.EVSEL bit, write 1 to the POCCRn.LVLEVDIS bit.

When POCSTS.PWMnF = 1, PWMnL/H of the PWM output becomes the pin state controlled by the POC. The PWMnH pin state controlled by the POC is determined by the POCCRn.CONFHn0 bit and the POCOCRn.CONFHn1 bit. The PWMnL pin state controlled by the POC is determined by the POCCRn.CONFLn0 bit and the POCOCRn.CONFLn1 bit. The following explains the example of the PWM0H pin setting. The setting of the PWMnL pin is the same.

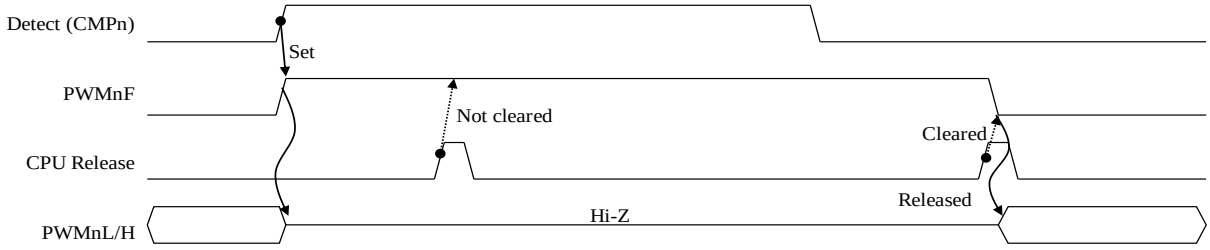
- In the settings of POCCR0.CONFH00 = 0 and POCOCR0.CONFH01 = 0
  When POCSTS.PWM0F = 1, the PWM0H pin becomes the high impedance.
- In the settings of POCCR0.CONFH00 = 1 and POCOCR0.CONFH01 = 0
  Even if POCSTS.PWM0F = 1, the POC does not control the PWM0H pin.
- In the settings of POCCR0.CONFH00 = 0 and POCOCR0.CONFH01 = 1
  When POCSTS.PWM0F = 1, the PWM0H pin becomes the low level.
- In the settings of POCCR0.CONFH00 = 1 and POCOCR0.CONFH01 = 1
  When POCSTS.PWM0F = 1, the PWM0H pin becomes the high level.

There are 2 methods to release the control by the POC: the CPU/EPU release and the automatic release. The CPU/EPU release is issued by writing 1 to the POCSTS.PWMnF bit. The automatic release is issued by the PWM channel n when POCCRn.ARELEN = 1. The PWM channel n issues the automatic release trigger, when either of the following compare match event is generated: the compare match event between the PWMn counter and the CMP_MIN value, which is in the up-down mode, or between the counter and the CMP_MAX value, which is in the up mode, (i.e., the CMP_MIN value is loaded when the PWMnCNT register's value matches the CMP_MAX value in the up mode). The POCSTS.PWMnF bit is cleared when the CPU/EPU release or the automatic release is issued while the selected event is not detected. For the PWMnL/H output control timing, see Figure 26-2.

First, make sure that the bit corresponding to the channel of the POCBAS register is 0, and then write to the POCTRG and POCSTS registers. Even if the writing is performed when the bit corresponding to the channel of the POCBAS register is 1, the writing is not reflected in the POC operation.



(a) Auto-release (POCCRn.EN = 1, POCCRn.ARELEN = 1)



(b) CPU Release (POCCRn.EN = 1)

Figure 26-2. Operation Timing

When the POC controls the PWMnL/H pin to high impedance and the corresponding pull-down of the PWMnL/H pin is enabled (when the corresponding bit of the PPD1 register is set to 1), the corresponding PWMnL/H pin is pulled down while the POC controls the pin output.

The POCCRn.EN bit enables the POC function of the PWM channel n. When POCCRn.EN = 1, the POCSTS.PWMnF bit can be set by the selected control events. When POCCRn.EN = 0, the POCSTS.PWMnF bit cannot be set. Before the register that sets the POC control is set again, set POCCRn.EN = 0.

26.3.2. Control Delay Addition

Only when controlling the pin level to high, the POC can add a delay to the time from receiving the event until starting to control the pin. The setting whether to add a delay or not can be selected for each pin. To add a delay to the PWMnH pin control, set POCDTCn.HDEN = 1 in addition to the normal setting. To add a delay to the PWMnL pin control, set POCDTCn.LDEN = 1 in addition to the normal setting.

The delay time is determined by the POCDTPn register. For the setting value, see Section 26.2.7. While the control delay is added, the POC response when the release signal is issued in the control delay can be selected from the following operations:

(1) Release the POC control immediately
(2) Release the control after waiting for the set delay time and the pin is controlled

When selecting the operation (2) above, set POCDTCn.CLRWAIT = 1. At this time, the automatic clear function by the PWM (POCCRn.ARELEN = 1) cannot be used. Also, be sure to check that the POC is controlling the pin, and then clear the POC control from the CPU or the EPU.

26.4. Usage Notes and Restrictions

26.4.1. Clock Settings

To use the POC appropriately, be sure to set the following settings: enable the POC clock by the MCLKE5 register of the system controller; enable the PWM clock with setting proper values to the PWMCSC0 and PWMENBL registers.

26.4.2. Usage Notes on POCCRn.EN Bit

Before changing the POC setting value, be sure to set the POCCRn.EN bit to 0. To avoid malfunctions, do not change another field value in the same address of the POCCRn.EN bit when the POCCRn.EN bit is changed to 0.

26.4.3. Operational Timing at Control Delay Addition

When the control delay is added to the POC, the POC channel with control delay synchronizes the following signals with its own clock.

- Operation starting event signal
- Operation starting signal by writing to the POCTRG register
- Operation release signal by writing to the POCSTS register

For the PWM channels with the pins where the control delay, the time after a signal detection until a reflection of the signals to the pin control is about 16 cycles longer in the PWM cycle than that of the PWM channels without the pin added the control delay.

26.4.4. Operational Restrictions on Using Clear-wait Function

The following restrictions are implemented when clearing the POC control in the case where the control delay addition function is used and the clear-wait function is enabled (i.e., setting the POCDTCn register to either of 0x81, 0x82, or 0x83):

(1) Auto-release by PWM is not available. Be sure to set POCCRn.ARELEN = 0.
(2) Before clearing the POC control by writing 1 to the POCSTS.PWMnF bit, read the POCSTS register to make sure that the bit to be cleared is set to 1 (or, currently being controlled). Otherwise, the LSI may result in unexpected behaviors thus malfunctions.

27. Comparator Lookup Table (CMPLUT)

27.1. Overview

The comparator lookup table (CMPLUT) generates the arbitrary event signals from 6 comparator outputs and the GPIO level events. The CMPLUT outputs are connected to the event inputs of POC, PWM, EPU, and the LUT0/1 output pin.

Table 27-1 shows the CMPLUT functional descriptions. Figure 27-1 shows the CMPLUT block diagram.

Table 27-1. CMPLUT Functional Descriptions

Item	Description
Number of Units	2 units
Input Signal	Inputs the output signals of comparators (CMP0 to CMP5) (The asynchronous output or the noise filter output is selectable per unit)
Output Signal	Outputs the logic calculation results of the lookup table and each GPIO level event (The asynchronous output or the noise filter output is selectable per unit)

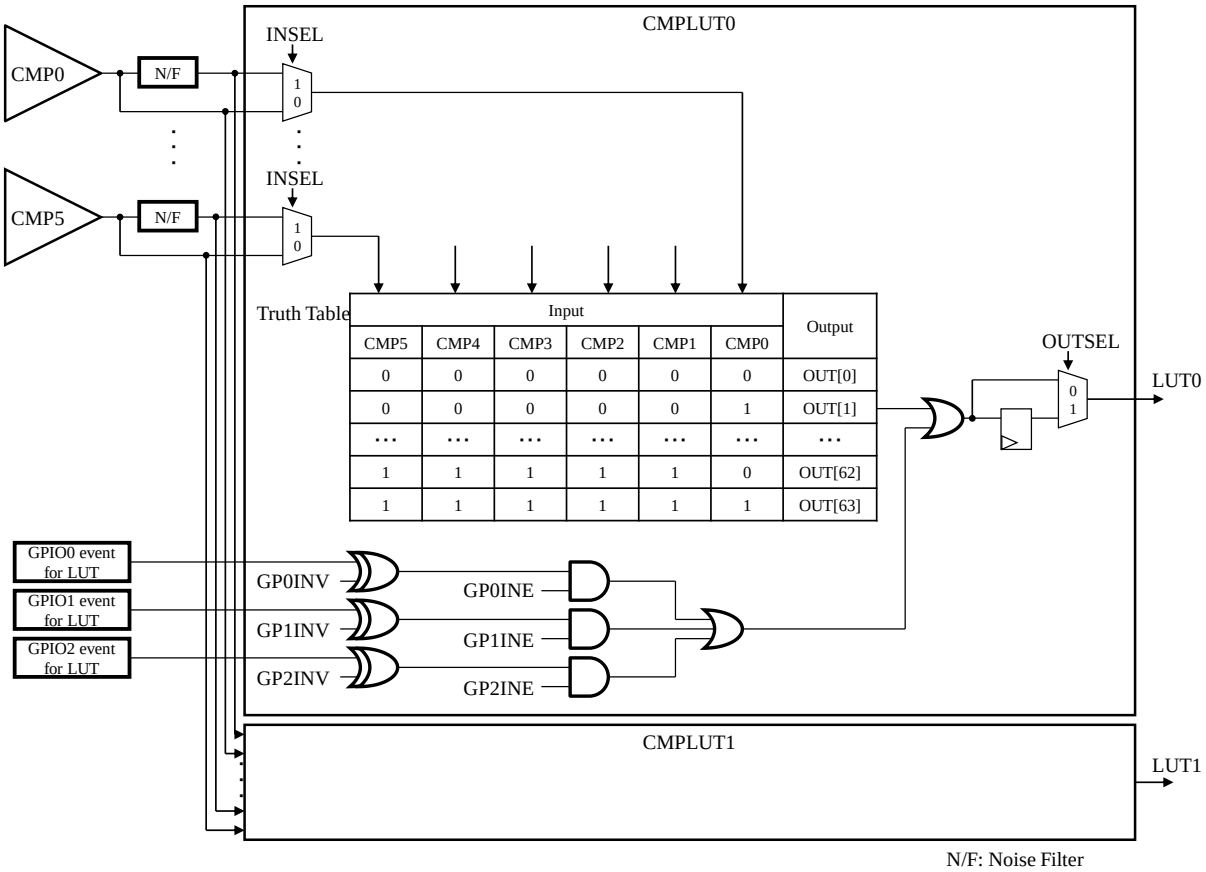


Figure 27-1. CMPLUT Block Diagram

27.2. Register Descriptions

Table 27-2. List of XDATA BUS Registers

Symbol	Name	Address	Initial Value
LUT0CR	LUT0 Control Register	0xEC80	0x00
LUT0GPCR	LUT0 GPIO Event Control Register	0xEC81	0x00
LUT0OUT0	LUT0 Output Register0	0xEC88	0x00
LUT0OUT1	LUT0 Output Register1	0xEC89	0x00
LUT0OUT2	LUT0 Output Register2	0xEC8A	0x00
LUT0OUT3	LUT0 Output Register3	0xEC8B	0x00
LUT0OUT4	LUT0 Output Register4	0xEC8C	0x00
LUT0OUT5	LUT0 Output Register5	0xEC8D	0x00
LUT0OUT6	LUT0 Output Register6	0xEC8E	0x00
LUT0OUT7	LUT0 Output Register7	0xEC8F	0x00
LUT1CR	LUT1 Control Register	0xEC90	0x00
LUT1GPCR	LUT1 GPIO Event Control Register	0xEC91	0x00
LUT1OUT0	LUT1 Output Register0	0xEC98	0x00
LUT1OUT1	LUT1 Output Register1	0xEC99	0x00
LUT1OUT2	LUT1 Output Register2	0xEC9A	0x00
LUT1OUT3	LUT1 Output Register3	0xEC9B	0x00
LUT1OUT4	LUT1 Output Register4	0xEC9C	0x00
LUT1OUT5	LUT1 Output Register5	0xEC9D	0x00
LUT1OUT6	LUT1 Output Register6	0xEC9E	0x00
LUT1OUT7	LUT1 Output Register7	0xEC9F	0x00

27.2.1. LUTnCR (LUTn Control Register) (n = 0 to 1)

Register	LUT0CR	LUT0 Control Register	Address	0xEC80
Register	LUT1CR	LUT1 Control Register	Address	0xEC90

Bit	Bit Name	R/W	Initial	Description	Remarks
7	OUTSEL	R/W	0	LUT output selection 0: LUT output not synchronized with CLKFAST (combinational logic output) 1: LUT output synchronized with CLKFAST (flip-flop output)	
6	INSEL	R/W	0	LUT input selection 0: CMP asynchronous output is input to LUT 1: CMP noise filter output is input to LUT For more details on the asynchronous output, see Section 27.3.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	Reserved	R	0	The read value is 0. The write value must always be 0.	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

27.2.2. LUTnGPCR (LUTn GPIO Event Control Register) (n = 0 to 1)

Register	LUT0GPCR	LUT0 GPIO Event Control Register	Address	0xEC81
Register	LUT1GPCR	LUT1 GPIO Event Control Register	Address	0xEC91

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	GP2INE	R/W	0	GPIO2 event input enable 0: The logical OR is not taken 1: The logical OR is taken This bit determines whether or not to take the logical OR of the following: the LUT output and an inversion-controlled event signal.	
5	GP1INE	R/W	0	GPIO1 event input enable 0: The logical OR is not taken 1: The logical OR is taken This bit determines whether or not to take the logical OR of the following: the LUT output and an inversion-controlled event signal.	
4	GP0INE	R/W	0	GPIO0 event input enable 0: The logical OR is not taken 1: The logical OR is taken This bit determines whether or not to take the logical OR of the following: the LUT output and an inversion-controlled event signal.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	GP2INV	R/W	0	Inversion of GPIO2 event input 0: GPIO2 event signal is not inverted 1: GPIO2 event signal is inverted	
1	GP1INV	R/W	0	Inversion of GPIO1 event input 0: GPIO1 event signal is not inverted 1: GPIO1 event signal is inverted	
0	GP0INV	R/W	0	Inversion of GPIO0 event input 0: GPIO0 event signal is not inverted 1: GPIO0 event signal is inverted	

27.2.3. LUTnOUTm (LUTn Output Register m) (n = 0 to 1, m = 0 to 7)

Register	LUT0OUT0	LUT0 Output Register0	Address	0xEC88
Register	LUT0OUT1	LUT0 Output Register1	Address	0xEC89
Register	LUT0OUT2	LUT0 Output Register2	Address	0xEC8A
Register	LUT0OUT3	LUT0 Output Register3	Address	0xEC8B
Register	LUT0OUT4	LUT0 Output Register4	Address	0xEC8C
Register	LUT0OUT5	LUT0 Output Register5	Address	0xEC8D
Register	LUT0OUT6	LUT0 Output Register6	Address	0xEC8E
Register	LUT0OUT7	LUT0 Output Register7	Address	0xEC8F
Register	LUT1OUT0	LUT1 Output Register0	Address	0xEC98
Register	LUT1OUT1	LUT1 Output Register1	Address	0xEC99
Register	LUT1OUT2	LUT1 Output Register2	Address	0xEC9A
Register	LUT1OUT3	LUT1 Output Register3	Address	0xEC9B
Register	LUT1OUT4	LUT1 Output Register4	Address	0xEC9C
Register	LUT1OUT5	LUT1 Output Register5	Address	0xEC9D
Register	LUT1OUT6	LUT1 Output Register6	Address	0xEC9E
Register	LUT1OUT7	LUT1 Output Register7	Address	0xEC9F

Bit	Bit Name	R/W	Initial	Description	Remarks
7	OUT[8×m+7]	R/W	0	LUT output when $CMP[5:0] = 8 \times m + 7$	
6	OUT[8×m+6]	R/W	0	LUT output when $CMP[5:0] = 8 \times m + 6$	
5	OUT[8×m+5]	R/W	0	LUT output when $CMP[5:0] = 8 \times m + 5$	
4	OUT[8×m+4]	R/W	0	LUT output when $CMP[5:0] = 8 \times m + 4$	
3	OUT[8×m+3]	R/W	0	LUT output when $CMP[5:0] = 8 \times m + 3$	
2	OUT[8×m+2]	R/W	0	LUT output when $CMP[5:0] = 8 \times m + 2$	
1	OUT[8×m+1]	R/W	0	LUT output when $CMP[5:0] = 8 \times m + 1$	
0	OUT[8×m]	R/W	0	LUT output when $CMP[5:0] = 8 \times m$	

27.3. Operation

The comparator (CMP) output is input to the CMPLUT. Each CMP has 2 outputs: the asynchronous output (output signal is not synchronized with the CLKFAST) and the noise filter output. The asynchronous output is an event masked comparator output processed by the output latch and the PWM signal in updating the DAC. The input to the CMPLUT can be determined by the LUTnCR.INSEL bit.

The LUT outputs in the truth table of 6 inputs and 1 output (see Figure 27-1) are defined by the LUTnOUTm register. The OUT[x] bit of LUTnOUTm register indicates the output when inputting from CMP[CMP5, CMP4, ⋯, CMP1, CMP0] = x. From the truth table, select the LUT output that is set to 1, and then set only the corresponding OUT[x] bit to 1.

For example, to output 1 when all comparator inputs (CMP0 to CMP5) are 1, set only the OUT[63] bit (i.e., bit 7 of the LUTnOUT7 register) to 1. Set the remained OUT[x] bits to 0.

In addition, the logical OR of the LUT output and the GPIO event signal is output to the CMPLUT. The event signals that the GPIO outputs are level types. To invert the event signal of GPIO0/1/2, set the LUTnGPCR.GPxINV bit (x = 0/1/2) to 1.

The LUTn pin operates as the CMPLUTn output. The LUTn pin signal can output to the LSI output pin, or can be used for the input events of the PWM and the POC. There are 2 outputs determined by the LUTnCR.OUTSEL bit: the LUT output that is not synchronized with the CLKFAST and the flip-flop output that is synchronized with the CLKFAST. When LUTnCR.INSEL = 1 and LUTnCR.OUTSEL = 1, a glitch pulse of the LUTn pin can be suppressed, but the output latency is increased. Although the glitch may occur in other settings, the output latency is decreased.

28. Serial Communication Interface with Debugger (SCID)

The LSI has a serial communication interface with debugger (SCID) functional module. This is a bidirectional single-wire interface, which can communicate with an external host device by the half-duplex universal asynchronous receiver transmitter (UART) method.

The SCID interfaces between the integrated development environment and the on chip debugger (OCD) incorporated in the LSI, used for program debugging by users and programming on the flash memory.

The communication functions for users and the communication functions for debugger are commonly exclusive because they use a single-wire interface. By externally preparing a debugging interface board (separately supplied), the LSI can simultaneously process communications for users and the debugger.

28.1. Overview

The purpose of the SCID is communications with an external host device (see Figure 28-1), debugging of user software, and software programming on the flash memory (erasing and writing).

The communication protocol with an external host device is based on the half-duplex UART method, which employs bi-directional single-wire signals. The single-wire signals must be operated by the open drain method so that these signals do not electrically conflict.

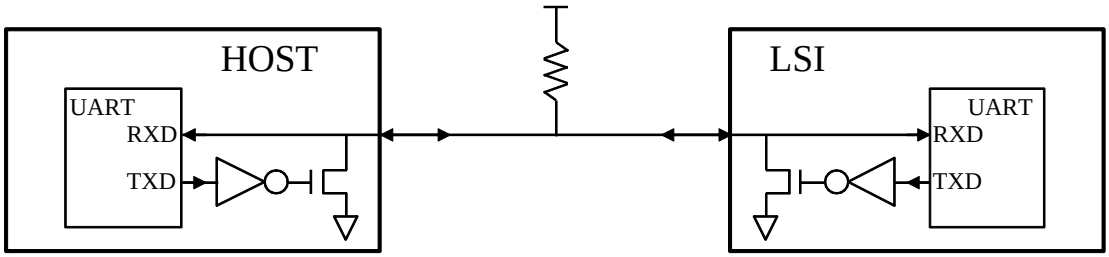


Figure 28-1. Single-wire Communications with External Host Device

Debugging of user software and software programming on the flash memory are conducted by connecting the OCD implemented on the CPU core with the external debugging interface board. Communications are performed by the half-duplex UART method, which employs bi-directional single-wire signals.

When constructing a user communication system with the SCID function as shown in Figure 28-1, the software must also be debugged. When debugging the software, it is ideal that the 2 types of communication interface for users and the debugger are independent on each other as shown in Figure 28-2. However, the LSI has only one signal interface for communications. The SCID provides a function to use the system shown in Figure 28-2 by employing one single-wire communication interface as a multiplex interface for users and the debugger. The next section describes the basic policy of the SCID.

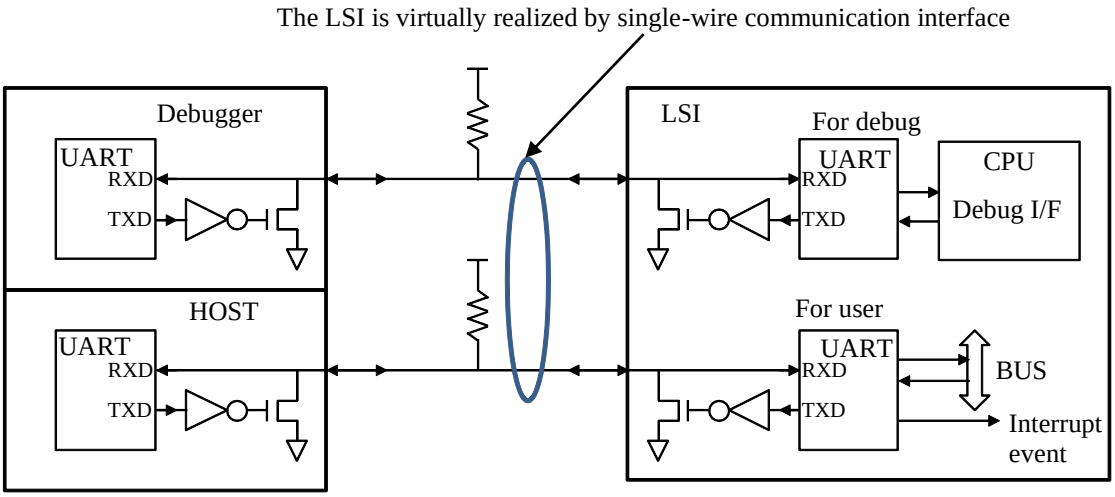


Figure 28-2. Simultaneous Use of Single-wire Communications for Users and Debugger

28.2. Basic Operation

The SCID has 2 operation modes, the UART and OCD modes.

28.2.1. Operation Mode Setting

Immediately after the LSI is reset, the SCID waits for receiving data 0x55 that is the 8N1 (8 bits, non-parity, and one stop bit length) format from outside. While the data is being received, baud rate is measured from the pulse width of the received signal waveform. Once baud rate is measured, the SCID communicates using the measured baud rate afterward.

The measured baud rate is a 16-bit value and is reflected in the UART_BAUD_H (higher 8 bits) and UART_BAUD_L (lower 8 bits) registers. The operation mode of the SCID is determined according to this value as follows.

- UART mode: When {UART_BAUD_H, UART_BAUD_L} $>$ 48 (i.e., when baud rate is smaller than about 250 Kbps)
- OCD mode: When {UART_BAUD_H, UART_BAUD_L} $\leq$ 48 (i.e., when baud rate is about 250 Kbps or more)

28.2.2. UART Mode

As shown in Figure 28-3, in the UART mode, the SCID communicates with a single external host device by the single-wire half-duplex UART method. This is the mode to perform the UART communications as a normal user application program with the debugging function disabled. Users can set the UART communication format and baud rate again.

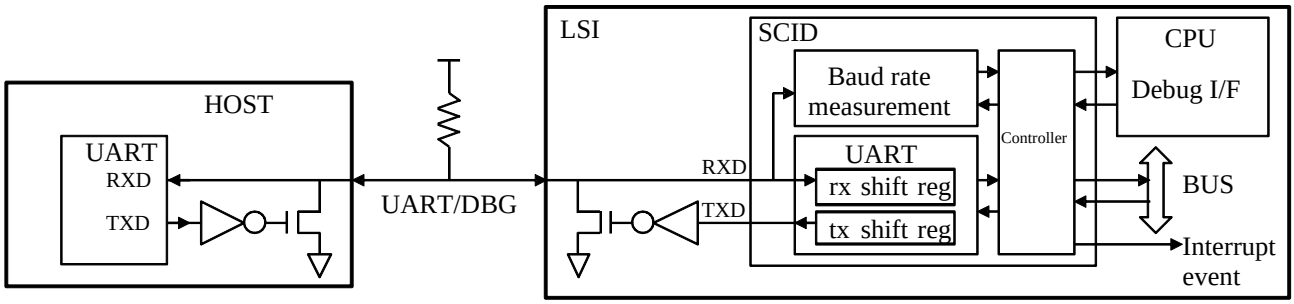


Figure 28-3. Single-wire Communications with External Host Device

28.2.3. OCD Mode

The OCD mode is used to directly connect to the OCD interface board. This board is prepared for debugging used in the integrated development environment, which is separately supplied by us. In the OCD mode, debug a normal program and erasing/programming to the flash memory.

In addition, Figure 28-4 shows a configuration to use debugger function while user communications equivalent to the UART mode. To configure the system of Figure 28-4, use the SCID dedicated interface board supplied by us. The SCID dedicated interface board has the following 3 communication ports:

- **Communications with External Host Device**

An external host device is connected with the SCID dedicated interface with single-wire communication signals. The protocol is the same as the communications between the external host device and the SCID in the UART mode (see Figure 28-3). When the program on the LSI changes the SCID setting registers to set the UART communication format and baud rate again, those settings are reflected in the communications between the SCID dedicated interface board and the external host device.

- **Communications with Personal Computer**

This is the USB interface to connect with a personal computer. The debugging and erasing/programming on the flash memory are controlled by the integrated development environment operating on the personal computer through the USB interface.

- **Communications with SCID**

The SCID of the LSI and the SCID dedicated interface board are connected to each other by single-wire communication signals. This baud rate is a fixed value (not disclosed). After starting up of the LSI, the SCID dedicated interface board transmits the 0x55, which is the baud rate that can select the OCD mode, to the SCID. For the baud rate to communicate with the SCID, the baud rate used when the 0x55 was transmitted is used as it is. As the format of the UART communications, 8N1 is used when the SCID receives data and 8N2 is used when the SCID transmits data. The communications used in the port is the non-disclosure dedicated protocol This is realized by the SCID and the SCID dedicated interface board which collaborate with each other.

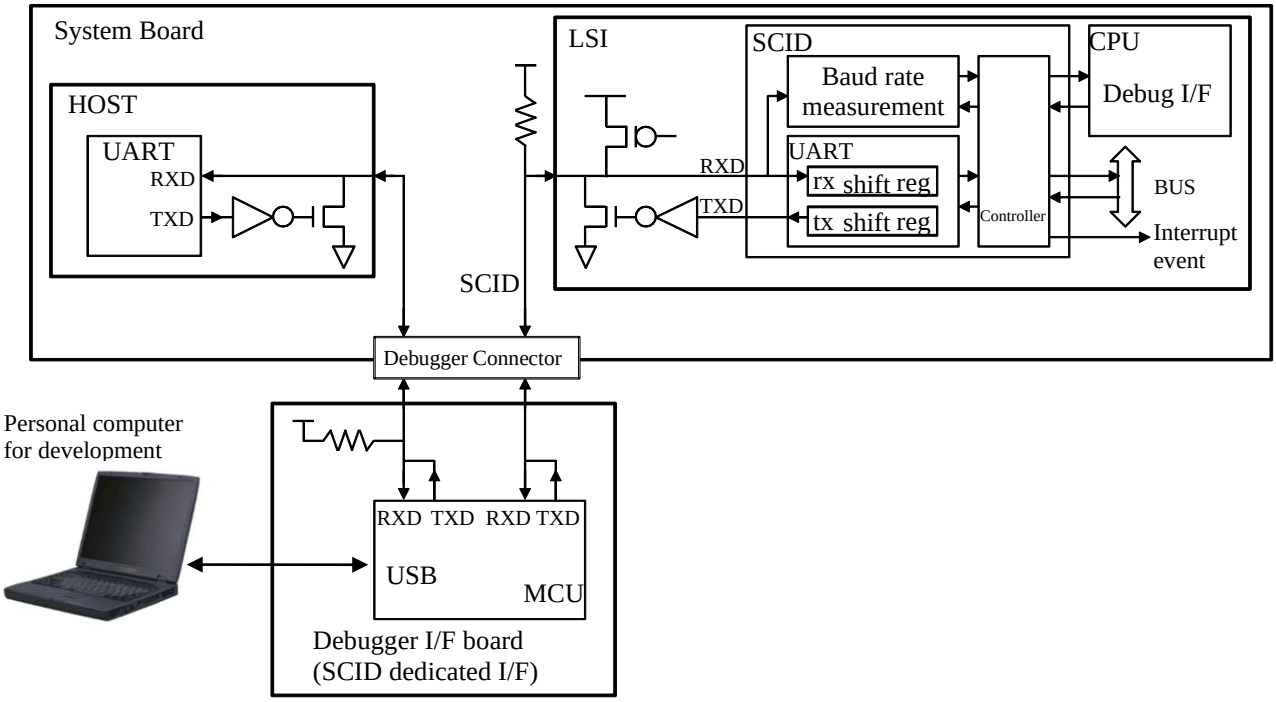


Figure 28-4. When Superimposing Debugging Communications from Integrated Development Environment on User Communications with External Host Device over Single-wire Communication Line

28.2.4. Notes for UART Operation

Regarding the communications between the external host device and the SCID, the UART mode is almost equal to the OCD mode. However, note that the communications timing and the communications latency may differ because communications are multiplexed on a single line in the OCD mode.

28.2.5. Debugging Operation and Flash Memory Programming without Communicating External Host Device

When communicating as follows, directly connect Sanken Electric's OCD interface board to the SCID: When communications are not necessary between the external host device and the SCID during debugging operation from the integrated development environment, or when only programming to the flash memory.

For using the SCID dedicated interface board, the port to communicate with the external host device can be opened (see Figure 28-5). It is not necessary to disconnect the signals between the external host device and the SCID. However, do not have the SCID communicate with the external host device.

Notes:

Section 28 describes the UART operation. The details of the operation of the OCD mode and the communication protocol that the SCID dedicated interface board uses between the personal computer and the SCID are not disclosed.

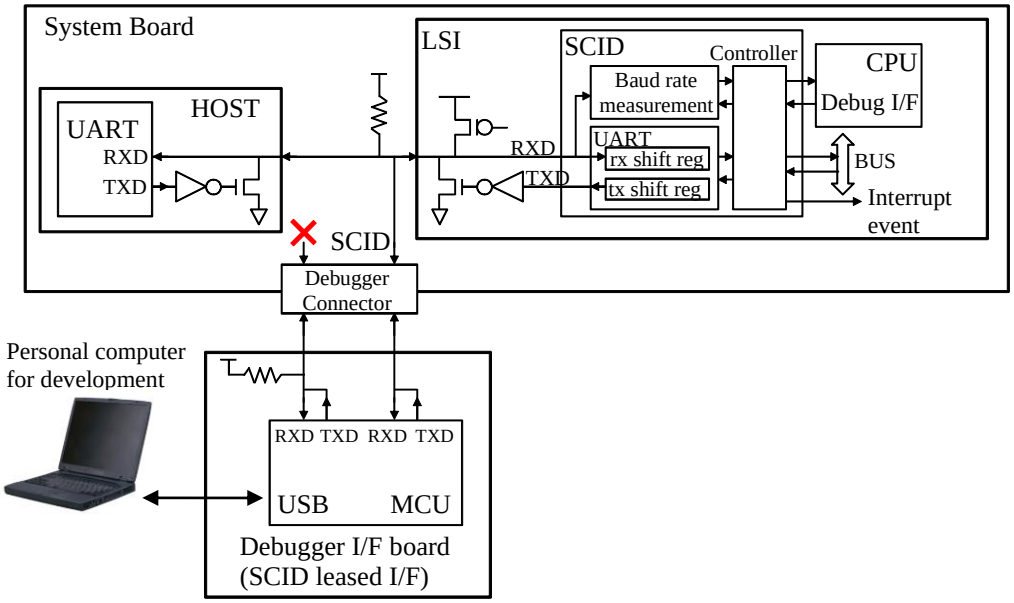


Figure 28-5. When not Communicating with External Host Device

28.3. UART Functional Descriptions in SCID

Table 28-1. UART Functional Descriptions in SCID

Item	Description
Character Format	Data length: 5 bits, 6 bits, 7 bits, or 8 bits Start bit: 1 bit Parity bit: None, even, odd, or sticky Stop bit: 1 bit or 2 bits
UART Communication Line	Bi-directional single-wire method Half-duplex Receiving operation can be masked during transmission
Baud Rate	Baud rate is automatically measured and set Baud rate can also be specified by users $\text{Baud rate (bps)} = \frac{\text{CLKIRC}}{\text{N}}$ Where: N is setting value of the baud rate register (N > 3)
FIFO	RXFIFO: 8 bits × 16 stages TXFIFO: 8 bits × 16 stages
Reception Ready Notification	When the number of bytes of the stacked received data in the RXFIFO reaches the specified number or more When the number of bytes of the stacked received data in the RXFIFO is smaller than the specified number and timeout occurs
Transmission Ready Notification	When the free space of the TXFIFO reaches the specified number or more
Event Flag Setting Condition	Framing error, parity error, overrun error, transmission completion, or reception break detection
Interrupt Request	Reception ready, transmission ready, or event flag Enabling can be set for each interrupt
Break Detection	Reception break can be detected
Modem Control	Not supported

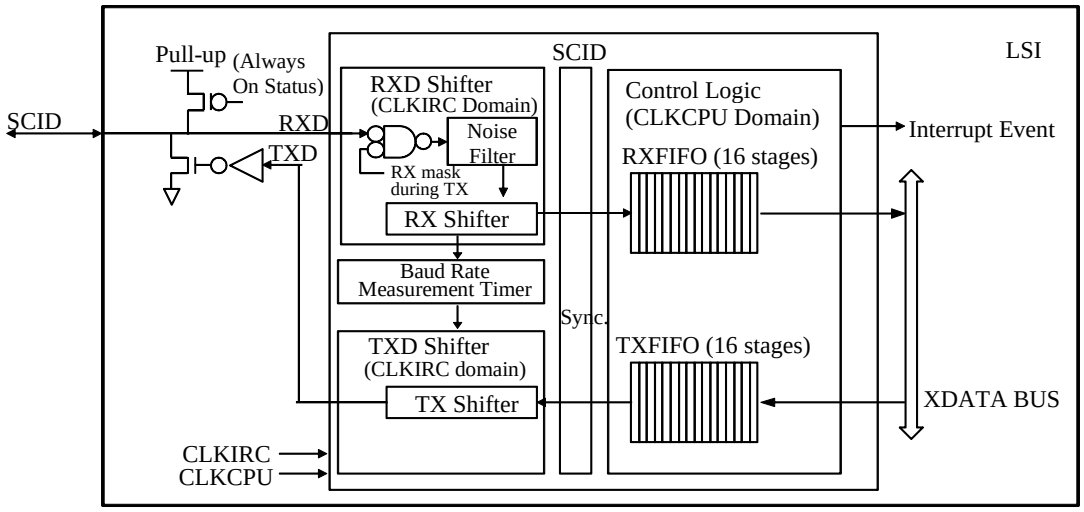


Figure 28-6. UART Functional Block Diagram in SCID

28.4. Register Descriptions

Symbol	Name	Address	Initial Value
UART_TXD	UART TX Data Register	0xE200	0x00
UART_RXD	UART RX Data Register	0xE201	0x00
UART_CR	UART Control Register	0xE202	0x03
UART_BRK	UART Break Control Register	0xE203	0x00
UART_SR	UART Status Register	0xE204	0x01
UART_IE	UART Interrupt Enable Register	0xE205	0x00
UART_TXFIFO_SR	UART TXFIFO Status Register	0xE206	0x80
UART_TXFIFO_CR	UART TXFIFO Control Register	0xE207	0x00
UART_RXFIFO_SR	UART RXFIFO Status Register	0xE208	0x80
UART_RXFIFO_CR	UART RXFIFO Control Register	0xE209	0x00
UART_RXFIFO_TO_L	UART RXFIFO Timeout Register Low	0xE20A	0x00
UART_RXFIFO_TO_H	UART RXFIFO Timeout Register High	0xE20B	0x00
UART_BAUD_L	UART Baud Rate Register Low	0xE20C	0x00
UART_BAUD_H	UART Baud Rate Register High	0xE20D	0x00
SCID_SYSR_CR	SCID System Control Register	0xE20E	0x00
SCID_STATE	SCID Internal State Register	0xE20F	0x00

28.4.1. UART_TXD (UART TX Data Register)

Register	UART_TXD		UART TX Data Register	Address	0xE200
Bit	Bit Name	R/W	Initial	Description	Remarks
7	TXD	R/W	0	Transmission data stored to TXFIFO Write: Written transmission data is stored to TXFIFO Read: Data written previously are read	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

28.4.2. UART_RXD (UART RX Data Register)

Register	UART_RXD		UART RX Data Register	Address	0xE201
Bit	Bit Name	R/W	Initial	Description	Remarks
7	RXD	R	0	Reception data read from RXFIFO Read: Reception data is read from RXFIFO Write: No effect	
6		R	0		
5		R	0		
4		R	0		
3		R	0		
2		R	0		
1		R	0		
0		R	0		

28.4.3. UART_CR (UART Control Register)

When enabling UART by the ENBL bit, the other bits can be simultaneously set.

Register	UART_CR	UART Control Register	Address	0xE202

Bit	Bit Name	R/W	Initial	Description	Remarks
7	ENBL	R/W	0	UART function enable 0: UART function is disabled 1: UART function is enabled	
6	RXMASK	R/W	0	Reception mask setting during transmission 0: Transmission data is received during transmission operation 1: Transmission data is not received during transmission operation (masked)	
5	PARITY	R/W	0	Parity bit 001: Odd parity 011: Even parity 101: High-level parity (sticky) 111: Low-level parity (sticky) Other than above: No parity	
4		R/W	0		
3		R/W	0		
2	STOP	R/W	0	Stop bit length of transmission characters 0: 1 stop bit 1: 2 stop bits	
1	DATALEN	R/W	1	Data length of transmission/reception characters 00: 5 bits 01: 6 bits 10: 7 bits 11: 8 bits	
0		R/W	1		

28.4.4. UART_BRK (UART Break Control Register)

Register	UART_BRK	UART Break Control Register	Address	0xE203

Bit	Bit Name	R/W	Initial	Description	Remarks
7	Reserved	R	0	The read value is 0. The write value must always be 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	RXBRK_E	R/W	0	Reception break detection enable 0: Reception break detection is disabled 1: Reception break detection is enabled	
0	Reserved	R	0	The read value is 0. The write value must always be 0.	

28.4.5. UART_SR (UART Status Register)

Register	UART_SR	UART Status Register	Address	0xE204

Bit	Bit Name	R/W	Initial	Description	Remarks
7	RX_BRK	R/C	0	Reception break detection flag Read 0: Reception break is not detected Read 1: Reception break is detected Write 0: No change Write 1: The bit is cleared	
6	RX_PER	R/C	0	Reception parity error detection flag Read 0: Reception parity error is not detected Read 1: Reception parity error is detected Write 0: No change Write 1: The bit is cleared	
5	RX_FER	R/C	0	Reception framing error detection flag Read 0: Reception framing error is not detected Read 1: Reception framing error is detected Write 0: No change Write 1: The bit is cleared	
4	RX_OVR	R/C	0	Reception overrun error detection flag Read 0: Reception overrun error is not detected Read 1: Reception overrun error is detected Write 0: No change Write 1: The bit is cleared	
3	TX_DONE	R/C	0	Transmission character sending completion detection flag Read 0: Sending completion is not detected Read 1: Sending completion is detected Write 0: No change Write 1: The bit is cleared The bit is cleared even when transmission data is written to the UART_TXD register.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	RX_RDY	R	0	Reception (RXFIFO) ready state detection flag Read 0: RXFIFO is not ready Read 1: RXFIFO is ready The flag just shows the state of the RXFIFO based on the setting of the UART_RXFIFO_CR register. It is not able to directly clear the flag only.	
0	TX_RDY	R	1	Transmission (TXFIFO) ready state detection flag Read 0: TXFIFO is not ready Read 1: TXFIFO is ready The flag just shows the state of the TXFIFO based on the setting of the UART_TXFIFO_CR register. It is not able to directly clear the flag only.	

28.4.6. UART_IE (UART Interrupt Enable Register)

Register	UART_IE	UART Interrupt Enable Register	Address	0xE205

Bit	Bit Name	R/W	Initial	Description	Remarks
7	IE_RX_BRK	R/W	0	RX_BRK interrupt request enable 0: RX_BRK interrupt request is disabled 1: RX_BRK interrupt request is enabled	
6	IE_RX_PER	R/W	0	RX_PER interrupt request enable 0: RX_PER interrupt request is disabled 1: RX_PER interrupt request is enabled	
5	IE_RX_FER	R/W	0	RX_FER interrupt request enable 0: RX_FER interrupt request is disabled 1: RX_FER interrupt request is enabled	
4	IE_RX_OVR	R/W	0	RX_OVR interrupt request enable 0: RX_OVR interrupt request is disabled 1: RX_OVR interrupt request is enabled	
3	IE_TX_DONE	R/W	0	TX_DONE interrupt request enable 0: TX_DONE interrupt request is disabled 1: TX_DONE interrupt request is enabled	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	IE_RX_RDY	R/W	0	RX_RDY interrupt request enable 0: RX_RDY interrupt request is disabled 1: RX_RDY interrupt request is enabled	
0	IE_TX_RDY	R/W	0	TX_RDY interrupt request enable 0: TX_RDY interrupt request is disabled 1: TX_RDY interrupt request is enabled	

28.4.7. UART_TXFIFO_SR (UART TXFIFO Status Register)

Register	UART_TXFIFO_SR	UART TXFIFO Status Register	Address	0xE206

Bit	Bit Name	R/W	Initial	Description	Remarks
7	NOT_FULL	R	1	TXFIFO Non-full state flag Read 0: TXFIFO is full Read 1: TXFIFO is not full	
6	NOT_EPTY	R	0	TXFIFO Non-empty state flag Read 0: TXFIFO is empty Read 1: TXFIFO is not empty	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	OCCUPATION	R	0	Data bytes in TXFIFO 00000: 0 byte 00001: 1 byte 00010: 2 bytes … 01111: 15 bytes 10000: 16 bytes Other than above: None	
3		R	0		
2		R	0		
1		R	0		
0		R	0		

28.4.8. UART_TXFIFO_CR (UART TXFIFO Control Register)

Register	UART_TXFIFO_CR	UART TXFIFO Control Register	Address	0xE207

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CLR	W	0	TXFIFO clearing Write 0: No change Write 1: TXFIFO is cleared The read value is always 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	RDY_LVL	R/W	0	TXFIFO ready level 0000: 1 byte 0001: 2 bytes 0010: 3 bytes … 1110: 15 bytes 1111: 16 bytes When the TXFIFO has free space equal to or larger than the value set by the bit, set the UART_SR.TX_RDY bit to 1.	
2		R/W	0		
1		R/W	0		
0		R/W	0		

28.4.9. UART_RXFIFO_SR (UART RXFIFO Status Register)

Register	UART_RXFIFO_SR	UART RXFIFO Status Register	Address	0xE208

Bit	Bit Name	R/W	Initial	Description	Remarks
7	NOT_FULL	R	1	RXFIFO Non-full state flag Read 0: RXFIFO is full Read 1: RXFIFO is not full	
6	NOT_EPTY	R	0	RXFIFO Non-empty state flag Read 0: RXFIFO is empty Read 1: RXFIFO is not empty	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	OCCUPATION	R	0	Data bytes in RXFIFO 00000: 0 byte 00001: 1 byte 00010: 2 bytes … 01111: 15 bytes 10000: 16 bytes Other than above: None	
3		R	0		
2		R	0		
1		R	0		
0		R	0		

28.4.10. UART_RXFIFO_CR (UART RXFIFO Control Register)

Register	UART_RXFIFO_CR	UART RXFIFO Control Register	Address	0xE209

Bit	Bit Name	R/W	Initial	Description	Remarks
7	CLR	W	0	RXFIFO clearing Write 0: No change Write 1: RXFIFO is cleared The read value is always 0.	
6	Reserved	R	0	The read value is 0. The write value must always be 0.	
5	Reserved	R	0	The read value is 0. The write value must always be 0.	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	RDY_LVL	R/W	0	RXFIFO ready level setting 0000: 1 byte 0001: 2 bytes 0010: 3 bytes … 1110: 15 bytes 1111: 16 bytes When the RXFIFO has data equal to or longer than the value set by the bit, set the UART_SR.RX_RDY bit to 1.	
2		R/W	0		
1		R/W	0		
0		R/W	0		

28.4.11. UART_RXFIFO_TO_L/H (UART RXFIFO Timeout Register Low/High)

When setting a value, make sure to write to UART_RXFIFO_TO_L and UART_RXFIFO_TO_H registers in that order.

Register	UART_RXFIFO_TO_L	UART RXFIFO Timeout Register Low	Address	0xE20A

Bit	Bit Name	R/W	Initial	Description	Remarks
7	TOUT_L	R/W	0	RXFIFO timeout (LSB side) Set the lower 8 bits of RXFIFO_TOUT that is used in the following equation. When the state that UART_SR.RX_RDY = 0 and there is one byte or more in the RXFIFO continues for the timeout period, the UART_SR.RX_RDY bit is set even when the condition is not reached to the value determined by the UART_RXFIFO_CR.RDY_LVL bits. $\text{Timeout period (s)} = \frac{\text{RXFIFO\_TOUT} + 1}{\text{Baud rate (bps)}}$ Where the RXFIFO_TOUT is a 16-bit value that the TOUT_L and TOUT_H bits are connected.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

Register	UART_RXFIFO_TO_H	UART RXFIFO Timeout Register High	Address	0xE20B

Bit	Bit Name	R/W	Initial	Description	Remarks
7	TOUT_H	R/W	0	RXFIFO timeout (MSB side) Set the higher 8 bits of RXFIFO_TOUT that is used in the following equation. When the state that UART_SR.RX_RDY = 0 and there is one byte or more in the RXFIFO continues for the timeout period, the UART_SR.RX_RDY bit is set even when the condition is not reached to the value determined by the UART_RXFIFO_CR.RDY_LVL bits. $\text{Timeout period (s)} = \frac{\text{RXFIFO\_TOUT} + 1}{\text{Baud rate (bps)}}$ Where the RXFIFO_TOUT is a 16-bit value that the TOUT_L and TOUT_H bits are connected.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

28.4.12. UART_BAUD_L/H (UART Baud Rate Register Low/High)

When setting a value, make sure to write to UART_BAUD_L and UART_BAUD_H registers in that order.

Register	UART_BAUD_L	UART Baud Rate Register Low	Address	0xE20C

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BAUD_L	R/W	0	Baud rate (LSB side) Set the lower 8 bits of the dividing ratio of the baud rate. $\text{Baud rate (bps)} = \frac{\text{CLKIRC}}{\text{N}}$ Where N is a 16-bit value that the BAUD_H and BAUD_L bits are connected. In addition, N must be set to a value larger than 3.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

Register	UART_BAUD_H	UART Baud Rate Register High	Address	0xE20D

Bit	Bit Name	R/W	Initial	Description	Remarks
7	BAUD_H	R/W	0	Baud rate (MSB side) Set the higher 8 bits of the dividing ratio of the baud rate. $\text{Baud rate (bps)} = \frac{\text{CLKIRC}}{\text{N}}$ Where N is a 16-bit value that is the BAUD_H and BAUD_L bits are connected. In addition, N must be set to a value larger than 3.	
6		R/W	0		
5		R/W	0		
4		R/W	0		
3		R/W	0		
2		R/W	0		
1		R/W	0		
0		R/W	0		

28.4.13. SCID_SYSR_CR (SCID System Control Register)

Even when the internal state of the SCID is reset by this register, the TXFIFO and RXFIFO are not cleared.
Accessing to the SCID_SYSR_CR register is not recommended. The detailed specification is not disclosed.

Register	SCID_SYSR_CR	SCID System Control Register	Address	0xE20E

Bit	Bit Name	R/W	Initial	Description	Remarks
7	MODE	R/W	1	Forced setting of SCID operation mode Read 0: SCID operates with UART mode Read 1: SCID operates with OCD mode Write from 0 to 0: No change Write from 1 to 1: No change Write from 0 to 1: UART mode is forcibly changed to OCD mode* Write 1 to 0: OCD mode is forcibly changed to UART mode* * Mode is forcibly changed regardless of the baud rate that is set again and the mode setting by the baud rate of the reception data 0x55 during the start-up.	
6	OCD_RX_PER	R/C	0	Detection flag of reception parity error when the debugger is in communication Read 0: Reception parity error is not detected Read 1: Reception parity error is detected Write 0: No change Write 1: The bit is cleared The OCD communication side does not add a parity bit, so the bit is not set to 1.	
5	OCD_RX_FER	R/C	0	Detection flag of reception framing error when the debugger is in communication Read 0: Reception framing error is not detected Read 1: Reception framing error is detected Write 0: No change Write 1: The bit is cleared	
4	Reserved	R	0	The read value is 0. The write value must always be 0.	
3	Reserved	R	0	The read value is 0. The write value must always be 0.	
2	Reserved	R	0	The read value is 0. The write value must always be 0.	
1	RESET_MODE	W	0	Internal state resetting of operation mode that is currently set Write 0: No change Write 1: The internal state of the operation mode that is currently set is reset (the operation mode itself is not changed) The read value is always 0.	
0	RESET_ROOT	W	0	Internal state resetting of SCID Write 0: No change Write 1: The internal state of the SCID is reset to the initial state (In the initial state, the operation mode enters the state of writing for setting, and is set with the baud rate of the reception data 0x55) The read value is always 0.	

28.4.14. SCID_STATE (SCID Internal State Register)

Accessing to the SCID_STATE register is not recommended. The detailed specification is not disclosed.

Register	SCID_STATE	SCID Internal State Register	Address	0xE20F

Bit	Bit Name	R/W	Initial	Description	Remarks
7	STATE	R	0	Internal state The internal state is expressed by 16 bits. The lower 8 bits and higher 8 bits are read by the first and second reading, respectively.	
6		R	0		
5		R	0		
4		R	0		
3		R	0		
2		R	0		
1		R	0		
0		R	0		

28.5. SCID Operation in UART Mode

28.5.1. UART Character Format

The UART character format is configured with the following elements. This format is defined by the UART_CR register. Figure 28-7 shows an example of the UART character format.

● **Start Bit**

One character string is transmitted from the start bit to indicate the start of transfer. The start bit becomes the low level for a period of one bit (t_{BIT}). t_{BIT} can be calculated as the inverse of baud rate (bps).

● **Data Bits**

Data bits are transmitted after the start bit. The data bit length defined by the UART_CR.DATALEN bits can be selected from one of 5 to 8 bits. Data bits are transferred by the LSB first.

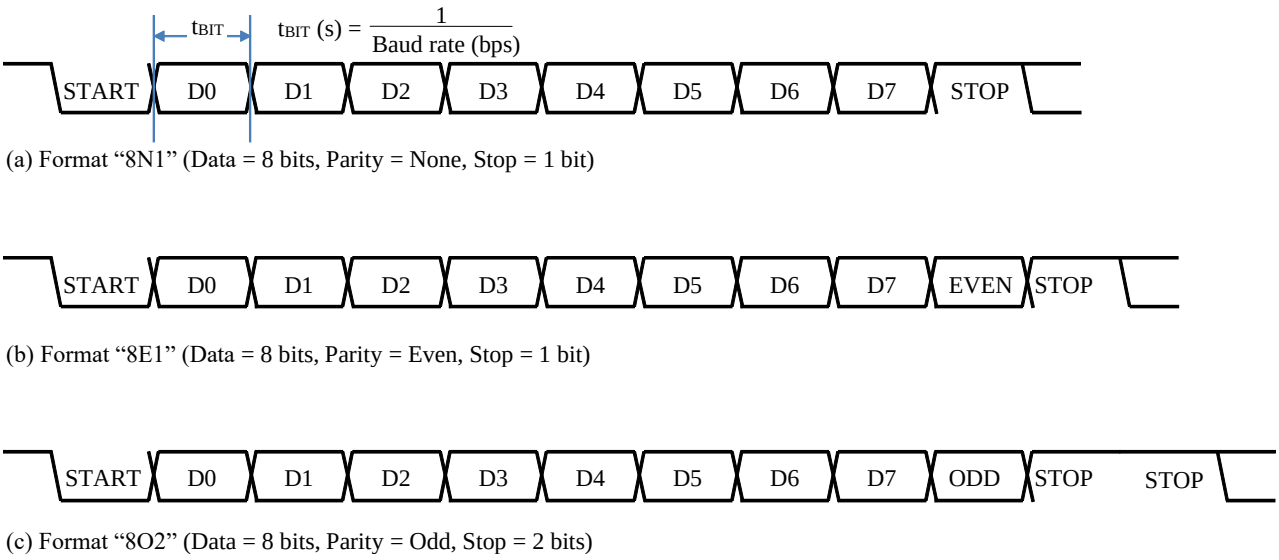
● **Parity Bit**

The parity bit can be determined by the UART_CR.PARITY bits. The setting is as follows:

- No Parity
  The parity bit is not added to the character format when data is transmitted. The parity is not checked when data is received. In addition, the parity error does not occur.
- Odd Parity
  When data is transmitted, the parity bit is set so that the number of 1 bits is odd number among the data and parity bits of each character string. Therefore, when the number of 1 bits in the data bits is even, the parity bit is set to1. When data is received, correctness of the parity is checked.
- Even Parity
  When data is transmitted, the parity bit is set so that the number of 1 bits is even number among the data and parity bits of each character string. Therefore, when the number of 1 bits in the data bits is odd, the parity bit set to 1. When data is received, correctness of the parity is checked.
- High-level Parity (Sticky)
  The parity bit is always set to 1 when data is transmitted, and it is checked when the data is received.
- Low-level Parity (Sticky)
  The parity bit is always set to 0 when data is transmitted, and it is checked when the data is received.

● **Stop Bit**

One or 2 stop bits are added to the end of the character according to the setting of the UART_CR.STOP bit. Regardless of the setting of the number of stop bits, the reception side checks the first stop bit only.



(a) Format "8N1" (Data = 8 bits, Parity = None, Stop = 1 bit)

(b) Format "8E1" (Data = 8 bits, Parity = Even, Stop = 1 bit)

(c) Format "8O2" (Data = 8 bits, Parity = Odd, Stop = 2 bits)

Figure 28-7. UART Character Formats Example

28.5.2. Reception Method

The UART character is received by the following sequence.

(1) The SCID waits for starting of the start bit.
(2) The falling edge of the start bit is detected.
(3) When half a bit period ($t_{BIT}/2$) has elapsed after the falling edge of the start bit is detected, the center of the start bit is sampled. When the start bit is not the low level, it is not recognized as occurrence of the start bit, and processing returns to step (1). When it is confirmed that the start bit is the low level, processing goes to step (4).
(4) Sampling of data bits starts at the center point of the first data bit (D0), which is one bit period (t_{BIT}) after the center point of the start bit. All data bits lasting after that are also sampled at each center point. The number of sampling times of data bits varies depending on the character format setting.
(5) When the parity bit is added to the character format, the center point of the parity bit is sampled to check if the parity is correct. When an error is detected, a parity error is indicated.
(6) Finally, the center point of the stop bit directly after the data bits or the parity bit is sampled. When the stop bit is not the high level, a framing error is indicated. Regardless of the number of the stop bits, processing returns to step (1) after the center of the first stop bit is sampled.

28.5.3. Reception Filter

As shown in Figure 28-6, there is a digital filter to remove short noise pulses before the register to shift the received data in the SCID. The filtering method employs simple majority decision.

28.5.4. Reception Masking

The UART employs single-wire bi-directional communications, so the data transmitted by the UART are looped back as is, and are received. The UART_CR.RXMASK bit determines whether or not to mask the receiving of the transmitted data.

28.5.5. Baud Rate

When the LSI is reset, the SCID becomes active with the initial state, and waits for receiving of the baud rate adjustment character shown in Figure 28-8 (data 0x55 that is 8N1 format). While this character is received, the interval between the rising and falling edges of the waveform is measured to calculate the corresponding baud rate. The baud rate value is automatically set to the UART_BAUD_H and UART_BAUD_L registers.

In addition, baud rate can be set at any time by rewriting of the UART_BAUD_H and UART_BAUD_L registers.

$$\text{Baud rate (bps)} = \frac{\text{CLKIRC}}{\text{N}}$$

Where, N is the 16-bit value that the values of the UART_BAUD_H.BAUD_H (8 bits) and UART_BAUD_L.BAUD_L (8 bits) bits are connected. N must be set >3.

Example 1
Target baud rate = 115200 bps:
When CLKIRC = 12 MHz and N = 104 (UART_BAUD_H = 0x00 and UART_BAUD_L = 0x68),
the actual baud rate = 115384.6 bps (error is +0.16%)

Example 2
Target baud rate = 9600 bps:
When CLKIRC = 12 MHz and N = 1250 (UART_BAUD_H = 0x04 and UART_BAUD_L = 0xE2),
the actual baud rate = 9600.0 bps (error is ±0.00)

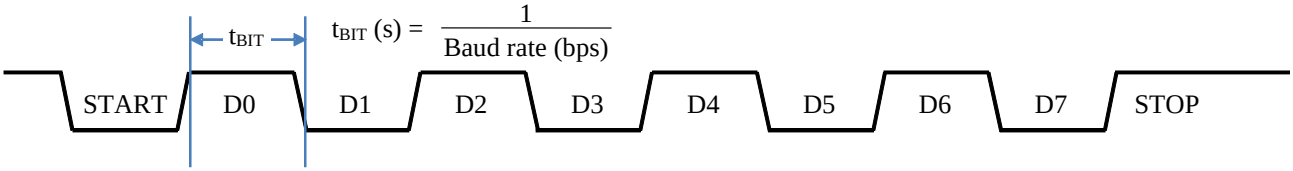


Figure 28-8. Reception Character for Baud Rate Adjustment

28.5.6. Baud Rate Setting and Operation Mode

When the SCID in the initial state receives the baud rate adjustment character, or a set value of baud rate is forcibly written to the UART_BAUD_H and UART_BAUD_L registers, the internal control logic of the SCID is initialized, and the operation mode is set according to the setting value of baud rate (see Table 28-2).

Table 28-2. Baud Rate Setting and Operation Mode

Value of Baud Rate Setting Register (16 bits) {UART_BAUD_H, UART_BAUD_L}	Baud Rate Value	Operation Mode
>48	Smaller than approx. 250 Kbps	UART mode
≤48	Approx. 250 Kbps or more	OCD mode

28.5.7. Transmission Data (TXD) and Transmission FIFO (TXFIFO)

To transmit data from the TXD signal line, write the transmission data to the UART_TXD register. The written transmission data is immediately buffered in the transmission FIFO (TXFIFO) of 16 stages. When the transmission shifter (TX shifter) is empty, the transmission data is transmitted from the TXFIFO to the TX shifter. The TX shifter transmits the data according to the setting value of the baud rate and the character format.

The ready state of the TXFIFO is indicated on the UART_SR.TX_RDY bit. When the free space of the TXFIFO reaches the value defined by the UART_TXFIFO_CR.RDY_LVL bits or more, the UART_SR.TX_RDY bit is set to 1.

The loading state of data into the TXFIFO is indicated on the UART_TXFIFO_SR register. If the transmission data is written to the UART_TXD register when the TXFIFO is not full, the data is transmitted to the TXFIFO as usual. Then, the transmission data can be confirmed again as needed by reading the UART_TXD register because the value transmitted to TXFIFO is left as is.

On the other hand, even if the transmission data is written to the UART_TXD register when the TXFIFO is full, the data is not transmitted to the TXFIFO. In this case, the written transmission data is left in the UART_TXD register. However, note that the data is not transmitted from the UART_TXD register to the TXFIFO even if the TXFIFO becomes not full.

The UART_TXD register operates as a window to write data to the TXFIFO (not a buffer). Therefore, while the TXFIFO is ready (i.e., UART_SR.TX_RDY = 1), write the transmission data to the UART_TXD register.

To clear the TXFIFO, write 1 to the UART_TXFIFO_CR.CLR bit.

28.5.8. Reception Data (RXD) and Reception FIFO (RXFIFO)

When the SCID receives data from the RXD signal line, the received data is input to the reception shifter (RX shifter) according to the setting value of the baud rate and the character format. The reception data is buffered in the reception FIFO (RXFIFO) of 16 stages. When RXFIFO has no free space, an overrun error (see Section 28.5.11) occurs. To read the reception data buffered in the RXFIFO, read the UART_RXD register.

The ready state of the RXFIFO is indicated on the UART_SR.RX_RDY bit. When the number of data bytes in the RXFIFO reaches the value defined by the UART_RXFIFO_CR.RDY_LVL bits or more, the UART_SR.RX_RDY bit is set to 1.

The loading state of data into the RXFIFO is indicated on the UART_RXFIFO_SR register. If the UART_RXD register is read when the RXFIFO is not empty, the reception data can be read from the RXFIFO.

If the UART_RXD register is read when the RXFIFO is empty, the RXFIFO is not operated. Therefore, the read value from this register has no meaning. Thus, while the RXFIFO is ready (i.e., UART_SR.RX_RDY = 1), read the reception data from the UART_RXD register.

To clear the RXFIFO, write 1 to the UART_RXFIFO_CR.CLR bit.

The UART_SR.RX_RDY bit remains 0 if the data satisfies following condition when data of one byte or more is received: The data in the RXFIFO is not reached to the condition defined by the UART_RXFIFO_CR.RDY_LVL bits, and the data is no longer received. If an application program is monitoring the UART_SR.RX_RDY bit only, the data in

the RXFIFO is continuously ignored. To avoid the state, the UART_SR.RX_RDY bit is kept set to 1 until the RXFIFO becomes empty when both following conditions continue for the timeout period: UART_SR.RX_RDY = 0 and the data of one byte or more is contained in the RXFIFO. Where, the UART_SR.RX_RDY bit is kept set to 1 even if the condition specified by the UART_RXFIFO_CR.RDY_LVL bits is not reached. The timeout period can be calculated using the following equation.

$$\text{Timeout period} = \frac{\text{RXFIFO\_TOUT} + 1}{\text{Baud rate (bps)}}$$

Where, RXFIFO_TOUT is a 16-bit value that the UART_RXFIFO_TO_H.TOUT_H (8 bits) and UART_RXFIFO_TO_L.TOUT_L (8 bits) bits are connected.

28.5.9. Detection of Reception (RX) Parity Error

When a parity error is detected in the received data, the UART_SR.RX_PER bit is set to 1. The following reception data is not transmitted to the RXFIFO: the data detected a parity error, and the data received during UART_SR.RX_PER = 1. Table 28-3 shows the setting and operation of the parity bit.

Table 28-3. Setting and Operation of Parity Bit

Setting of UART_CR.PARITY Bit	Description	How to Add Parity Bit for Transmission	Parity Bit Checking on Reception Side
001	Odd parity	Add so that the number of 1 bits in the data and the parity bits of the transmission characters are odd.	Check if the parity bit is added as defined in the addition rule in the left-hand column.
011	Even parity	Add so that the number of 1 bits in the data and the parity bits of the transmission characters are even.	
101	High-level parity (sticky)	Add the parity bit of 1 to the transmission characters.	
111	Low-level parity (sticky)	Add the parity bit of 0 to the transmission characters.	
Other than above	No parity	The parity bit is not added.	The parity bit is not checked.

28.5.10. Detection of Reception (RX) Framing Error

When the stop bit of a received character string is at the low level, a framing error is detected and the UART_SR.RX_FER bit is set to 1. The following reception data is not transmitted to RXFIFO: the data detected a framing error, and the data received during UART_SR.RX_FER bit = 1. In addition, the system does not perform receiving operation (waiting for the start bit) after a framing error is detected until the RXD signal becomes high level.

28.5.11. Detection of Reception (RX) Overrun Error

When the received characters that are constructed in the RX shifter are transmitted to the RXFIFO in the full state, an overrun error is detected and the UART_SR.RX_OVR bit is set to 1. In this case, the received data is discarded.

28.5.12. Detection of Data Transmission Completion (TX Done)

In the data transmission, if the TXFIFO is empty when data transmission of the TX shifter is completed (from the start bit to the final stop bit), it is allowed to stop the transmission and to conclude that the data transmission is completed (TX done). When this state is detected, the UART_SR.TX_DONE bit is set to 1. After that, when the next transmission data written to the UART_TXD register are transmitted to the TXFIFO, the UART_SR.TX_DONE bit is cleared.

28.5.13. Detection of Reception (RX) Break

To enable the reception (RX) break detection function, set UART_BRK.RXBRK_E =1. When the character waveform of the RXD signal satisfies the following condition, the SCID detects that the RX break is requested.

- Start bit = 0
- Data bits = 0x00
- Parity bit = 0 (if using parity bit)
- Stop bit = 0 (framing error state)

When the RX break is detected, the UART_SR.RX_BRK bit is set to 1. A framing error is also detected when the RX break is detected, so the UART_SR.RX_FER bit is set to 1. Moreover, a parity error is also detected depending on the case, and the UART_SR.RX_PER bit may be set to 1. The following reception data is not transmitted to the RXFIFO: the data detected the RX break, and the received data during UART_SR.RX_BRK = 1. Since a framing error occurs triggered by the detection of the RX break, the system does not perform receiving operation (waiting for the start bit) after the RX break is detected until the RXD signal becomes high level.

The SCID does not have a function to transmit break.

28.5.14. Interrupt Request

The interrupt request from the SCID is output according to the logical equation below. The bits used in this equation are in the UART_SR and UART_IE registers.

- Interrupt request = RX_BRK & IE_RX_BRK
  | RX_PER & IE_RX_PER
  | RX_FER & IE_RX_FER
  | RX_OVR & IE_RX_OVR
  | TX_DONE & IE_TX_DONE
  | RX_RDY & IE_RX_RDY
  | TX_RDY & IE_TX_RDY;

The corresponding bits (flags) in the UART_SR register must be cleared in the interrupt routine.

28.5.15. Setting Procedure of UART Function

Set the UART function of the SCID as follows:

(1) Reset the LSI.
(2) Receive the character string with a baud rate value in the range of the UART mode (8N1, 0x55), or set a baud rate with the range of the UART mode to the UART_BAUD_H and UART_BAUD_L registers.
(3) Set the UART_TXFIFO_CR and UART_RXFIFO_CR registers.
(4) Set the UART_RXFIFO_TO_L and UART_RXFIFO_TO_H registers.
(5) Set the UART_IE register.
(6) Set the UART_CR register.

When the operation mode of the SCID is determined by the setting value of baud rate, the value in each register other than the UART_BAUD_H and UART_BAUD_L registers is not changed. Therefore, step (2) can be always set after step (1).

28.6. Precautions for Using SCID

- If the SCID signals (single-wire signals of the TXD/RXD) are forcibly set to the low level from outside while the LSI is reset, the CPU keeps in the stopping state, which is the state before executing the instruction at address zero, and the SCID enters the OCD mode. This is the debugging function provided by the SCID.
- In the above state (the CPU is not running, and the SCID is in the OCD mode), or the SCID is set in the OCD mode, the built-in oscillator, IRC, does not stop even by setting the LSI to the low power consumption mode.
- When the MCLKE0.ME_SCID bit of the system controller is cleared, not only the UART function of the SCID is stopped but also the debugging function of the LSI (OCD) is stopped. Therefore, the LSI cannot be accessed from the integrated development environment.

29. Specifications

29.1. Absolute Maximum Ratings

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Storage Temperature	T_{STG}		−40	—	125	°C
Digital Power Supply	$V_{DVCCAMR}$		−0.3	—	4.0	V
Analog Power Supply	$V_{AVCCAMR}$		−0.3	—	4.0	V
Digital Input Voltage of 5 V Tolerant Pin	$V_{DIN5AMR}$		−0.3	—	5.5	V
Digital Input Voltage except 5 V Tolerant Pin	$V_{DIN3AMR}$		−0.3	—	V_{DVCC} + 0.3(1)	V
Analog Input Voltage	$V_{AIN3AMR}$		−0.3	—	V_{AVCC} + 0.3(2)	V
Total Output Current of Digital Pin	$\Sigma I_{DOUTAMR}$		—	—	58	mA
Total Output Current of Analog Pin	$\Sigma I_{AOUTAMR}$		—	—	32	mA
Voltage Difference between Digital Power Supply and Analog Power Supply(3)	$\lvert V_{DVCC} - V_{AVCC} \rvert$		—	—	0.3	V

(1) V_{DVCC} + 0.3 V < 4.0 V
(2) V_{AVCC} + 0.3 V < 4.0 V
(3) When the time occurring voltage difference is longer than 1 ms (allowable range is within 1 ms).

29.2. Recommended Operating Range

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Ambient Temperature	T_A		−40	—	110	°C
Ambient Temperature at Flash Memory Program/Erasing	T_{A_FLASH}		0	—	55	°C
Digital Power Supply*	V_{DVCC}		3.0	3.3	3.6	V
Analog Power Supply*	V_{AVCC}		3.0	3.3	3.6	V

* $\lvert V_{DVCC} - V_{AVCC} \rvert \leq 0.3$ V

29.3. Electrical Characteristics

29.3.1. Package Thermal Characteristics

The junction temperature of the LSI, T_J, must be used at 125 °C or less.

T_J is calculated by the following equation based on power loss, P_C, package thermal resistance, $R_{\theta(J-A)}$, ambient temperature, T_A.

$$T_J = T_A + P_C \times R_{\theta(J-A)}$$

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
QFN40 Thermal Resistance	$R_{\theta(J-A)}$	Wind speed:0 m/s	—	40	—	°C/W
	$R_{\theta(J-C)}$	Wind speed:0 m/s	—	20	—	°C/W

29.3.2. Consumption Current

An external load is not included in all parameters.

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
DVCC Current (Active)	I_{DVCC_ACTIVE}		—	50	90	mA
DVCC Current (Sleep)	I_{DVCC_SLEEP}	CPU stop	—	45	80	mA
DVCC Current (Standby)(1)	I_{DVCC_STBY}		—	1.5	4.5	mA
DVCC Current at Flash Memory Program/Erasing	I_{DVCC_FLASH}		—	55	—	mA
DVCC Current (ADC12)(2)	I_{DVCC_ADC12}		—	2	3.5	mA
AVCC Current (ADC12)(2)	I_{AVCC_ADC12}		—	30	40	µA
AVCC Current (DAC8)(2)	I_{AVCC_DAC8}		—	40	55	µA
DVCC Current (COMP)(2)	$I_{DVCC\_COMP\_H}$		—	100	150	µA
DVCC Current (COMP Low Power Consumption Mode)(2)	$I_{DVCC\_COMP\_L}$	When low power consumption mode is selected	—	30	60	µA
DVCC Current (OPAMP)(2)	I_{DVCC_OPAMP}		—	1.6	4.0	mA
DVCC Current (OPAMP Low Power Consumption Mode)(2)	$I_{DVCC\_OPAMP\_L}$	When low power consumption mode is selected	—	0.8	2.0	mA
DVCC Current (TEMP)(2)	I_{DVCC_TEMP}		—	0.3	0.5	mA
AVCC Current (Standby)(3)	I_{AVCC_STBY}		—	—	3.0	µA

(1) The built-in regulator, VREF, and POR consume the power even in standby mode.
(2) The consumption current of each analog module one unit that is enabled.
(3) The ADCC consumption current when the ADC and DAC (0x00 is set) are disabled.

29.3.3. Low Voltage Detection (LVD)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Voltage Detection Level	V_{DET}		2.7	2.9	3.0	V

29.3.4. Reset Operation

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
External Reset Width	t_{RES}	Cold start	10	—	—	ms
		Hot start	1	—	—	µs
POR Detection Voltage	V_{PORH}		—	2.6	—	V
	V_{PORL}		—	2.5	—	V
POR Detection Hysteresis Voltage	V_{POR_HYS}		—	100	—	mV

29.3.5. Clock Operation

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
IRC Oscillation Stable Time	t_{IRC}		—	—	100	µs
IRC Oscillation Frequency	f_{IRC_1}	T_A = −10 °C to 65 °C	11.76	—	12	MHz
	f_{IRC_2}	T_A = −20 °C to 85 °C	11.64	—	12	MHz
	f_{IRC_3}	T_A = −40 °C to 110 °C	11.52	—	12	MHz
PLL Oscillation Stable Time	t_{PLL_OSC}		—	—	100	µs

29.3.6. 12-bit SAR ADC

The condition of source impedance (R_{OUT_ADC12}) is 200 Ω or less.

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	BIT_ADC12		—	12	—	bits
Input Voltage Range*	V_{IN_ADC12}		—	V_{AVSS} to V_{AVCC}	—	V
Conversion Speed (Sampling Time + Conversion Time)	t_{CONV_ADC12}	Clock: 60 MHz, sampling time: 3 cycles	250	—	—	ns
Absolute Accuracy	ABS_ADC12	$\lvert V_{DVCC} - V_{AVCC} \rvert \leq 0.1$ V	−22	−4	14	LSB

* V_{AVSS} and V_{AVCC} are the voltage of the AVSS power supply pin and the voltage of the AVCC power supply pin, respectively.

29.3.7. 8-bit DAC

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	BIT_DAC8		—	8	—	bits
Output Voltage Range*	V_{OUT_DAC8}		V_{AVSS}	—	V_{AVCC} − 1 LSB	V
Output Settling Time	t_{CONV_DAC8}		—	120	500	ns
Absolute Accuracy	ABS_DAC8		—	±1	—	LSB

* V_{AVSS} and V_{AVCC} are the voltage of the AVSS power supply pin and the voltage of the AVCC power supply pin, respectively.

29.3.8. Op Amp (OPAMP)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input Voltage Range*	V_{IN_OPAMP}		V_{DVSS} + 0.05	—	V_{DVCC} − 0.5	V
Output Voltage Range*	V_{OUT_OPAMP}		V_{DVSS} + 0.05	—	V_{DVCC} − 0.5	V
Input Voltage Offset	V_{OFFSET_OPAMP}		—	±3	—	mV
Output Current	I_{OUT_OPAMP}		—	±1	—	mA
Common Mode Rejection Ratio	CMRR_OPAMP		—	70	—	dB
Power Supply Rejection Ratio	PSRR_OPAMP		—	50	—	dB
Output Noise	V_{ON_OPAMP}	1 kHz to 1 GHz	—	45	—	μVrms
Open Loop Gain	GAIN_OPAMP		—	80	—	dB
Gain Band Width Product	f_{GBW_OPAMP}		—	20	—	MHz
Voltage Gain	VGAIN_OPAMP1	×1	—	1	—	—
	VGAIN_OPAMP2	×4	See Figure 29-1			—
Slew Rate	$V_{SR\_OPAMP(R)}$	Rising	—	15	—	V/μs
	$V_{SR\_OPAMP(F)}$	Falling	—	15	—	V/μs

* V_{DVSS} and V_{DVCC} are the voltage of the DVSS power supply pin and the voltage of the DVCC power supply pin, respectively.

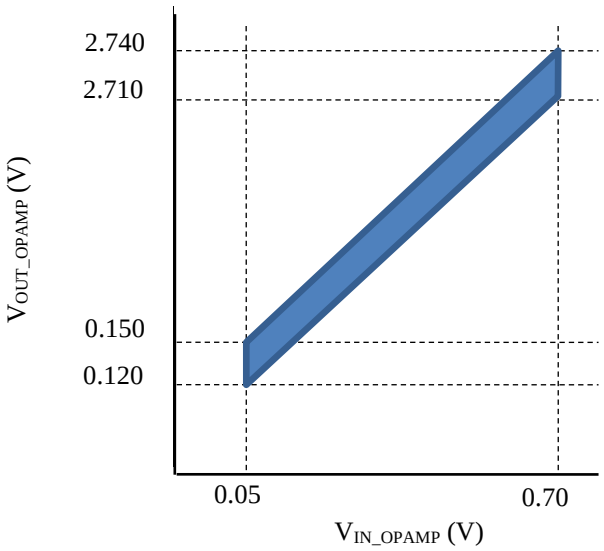


Figure 29-1. AMP Output Characteristics (setting: voltage gain ×4)

29.3.9. Comparator

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input Voltage Range(1)	V_{IN_COMP}		V_{DVSS} + 0.1	—	V_{DVCC} − 0.1	V
Comparison Voltage Range(1)	V_{IN_REF}		V_{DVSS} + 0.1	—	V_{DVCC} − 0.1	V
Hysteresis(2)	V_{IN_HYS}		—	56	120	mV
Response Time(3)	$t_{RESP\_COMP(HR)}$	When output signal is increased in high-speed mode	—	8	20	ns
	$t_{RESP\_COMP(HF)}$	When output signal is decreased in high-speed mode	—	24	40	ns
	$t_{RESP\_COMP(LR)}$	When output signal is increased in low-speed mode	—	18	40	ns
	$t_{RESP\_COMP(LF)}$	When output signal is decreased in low-speed mode	—	50	80	ns

(1) V_{DVSS} and V_{DVCC} are the voltage of the DVSS power supply pin and the voltage of the DVCC power supply pin, respectively.
(2) For the hysteresis characteristics, see Figure 29-2.
(3) For the conditions of measurement timing, see Figure 29-3.

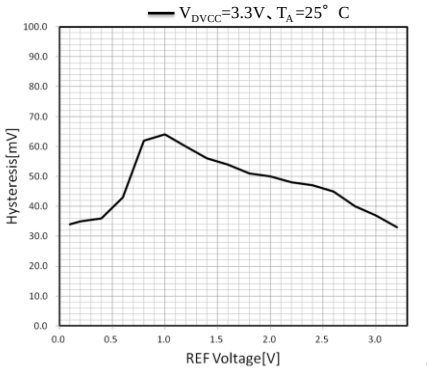


Figure 29-2. Comparator Hysteresis

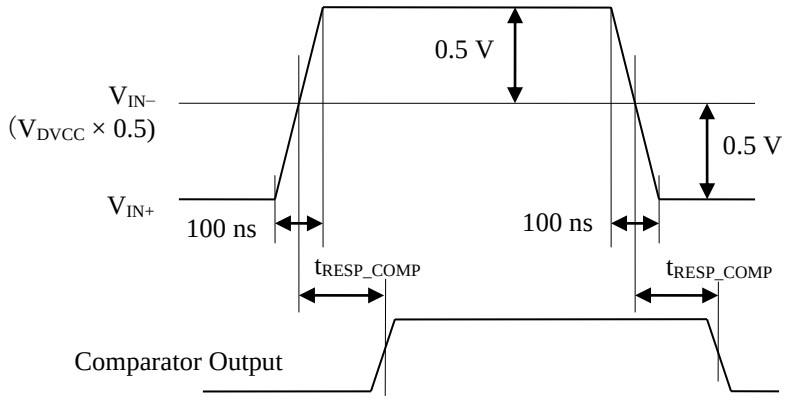


Figure 29-3. Comparator Timing

29.3.10. Reference Voltage (VREF)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Output Voltage	V_{REF}		—	1.2	—	V

29.3.11. Temperature Sensor (TEMP)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Output Voltage (T_J = 25 °C)	V_{TEMP}		—	1.52	—	V
Temperature Characteristic Slope	V_{DTEMP}		—	4.8	—	mV/°C
Settling Time	t_{TEMP}	After enable	—	—	2	ms

29.3.12. DC Specifications of Digital Input/Output

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input Voltage High Level	V_{IH}		2	—	—	V
Input Voltage Low Level	V_{IL}		—	—	0.8	V
Input Voltage High Level (Schmitt)	V_{IH_S}		2	—	—	V
Input Voltage Low Level (Schmitt)	V_{IL_S}		—	—	0.8	V
Schmitt Hysteresis	V_{HYS_S}		—	0.05	—	V
Pull-up Resistor (Excluding GPIO20 and GPIO21)	R_{PUP}		20	60	100	kΩ
Pull-up Resistor (GPIO20 and GPIO21)	R_{PUP2}		7.9	10	12.4	kΩ
Pull-down Resistor	R_{PDN}		20	90	200	kΩ
Input Leak Current	I_L		−2	±1	2	μA
Input Capacitance (Excluding ANEX0 to ANEX13)	C_{IN}		—	—	20	pF
Input Capacitance (ANEX0 to ANEX13)	C_{IN2}		—	—	30	pF
Output Voltage High Level (4 mA)	V_{OH4}	I_{OH} = −4 mA	2.4	—	—	V
Output Voltage Low Level (4 mA) (Excluding GPIO20 and GPIO21)	V_{OL4}	I_{OL} = 4 mA	—	—	0.4	V
Output Voltage Low Level (4 mA) (GPIO20 and GPIO21)	V_{OL42}	I_{OL} = 4 mA	—	—	0.5	V
Output Voltage High Level (16 mA)	V_{OH16}	I_{OH} = −16 mA	V_{DVCC} − 0.7	—	—	V
Output Voltage Low Level (16 mA)	V_{OL16}	I_{OL}= 16 mA	—	—	0.5	V

29.3.13. AC Specifications of Digital Input/Output

29.3.13.1. PWM Timing

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Rising Time of PWM Pin (GPIO10 to GPIO17)	t_r	C = 30 pF $V_{OH} = V_{DVCC} \times 0.7$ $V_{OL} = V_{DVCC} \times 0.3$	—	2.0	—	ns
Falling Time of PWM Pin (GPIO10 to GPIO17)	t_f	C = 30 pF $V_{OH} = V_{DVCC} \times 0.7$ $V_{OL} = V_{DVCC} \times 0.3$	—	2.0	—	ns

29.3.13.2. Serial Peripheral Interface (SPI) Timing

(1) Master Mode

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCK Cycle	t_{SCK}	C = 50 pF	80	—	—	ns
SO Output Delay Time	t_{DSPI}	C = 50 pF	0	—	10	ns
SI Hold Time	t_{HLSPI}		−3	—	—	ns
SI Setup Time	t_{SUSPI}		13	—	—	ns

(2) Subordinate Mode

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCK Cycle	t_{SCK}	30 MHz	33.33	—	—	ns
SO Output Delay Time	t_{DSPI}	C = 50 pF	5	—	15	ns
SI Hold Time	t_{HLSPI}		5	—	—	ns
SI Setup Time	t_{SUSPI}		5	—	—	ns

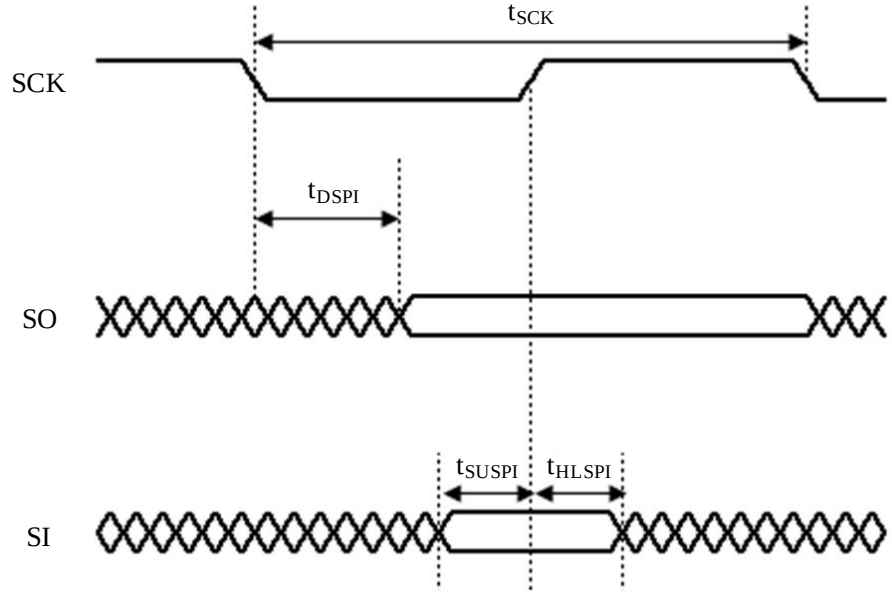


Figure 29-4. SPI Timing (Mode 0 and Mode 3)

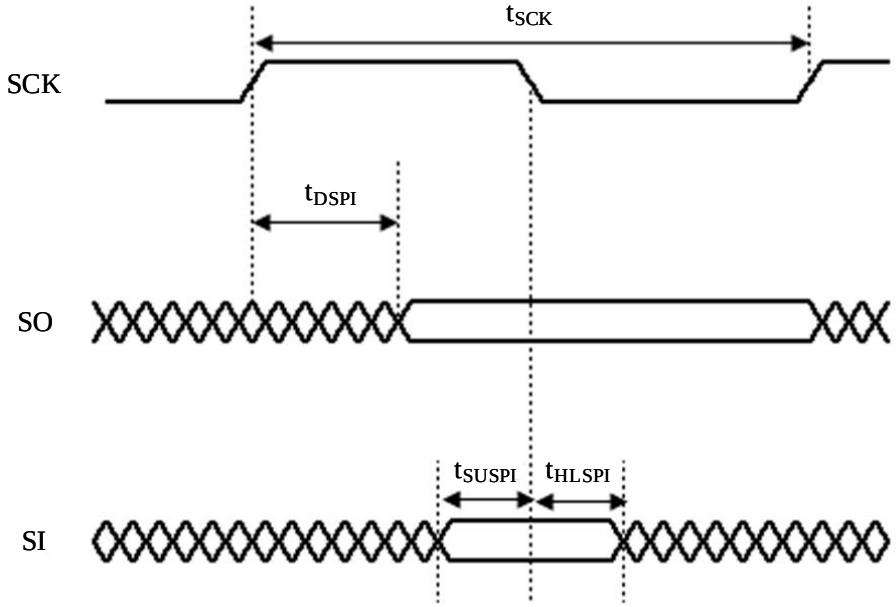


Figure 29-5. SPI Timing (Mode 1 and Mode 2)

29.3.13.3. I²C Timing

(1) Normal Mode

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCL Clock Frequency	f_{SCL}		0	—	100	kHz
Hold Time of Start Condition	$t_{HD:STA}$		4.0	—	—	µs
Low Level Period of SCL Clock	t_{LOWI2C}		4.7	—	—	µs
High Level Period of SCL Clock	$t_{HIGHI2C}$		4.0	—	—	µs
Setup Time of Start Condition	t_{SU_STA}		4.7	—	—	µs
Data Hold Time (I²C Bus Device)	t_{HD_DAT}		0	—	—	µs
Data Setup Time	t_{SU_DAT}		250	—	—	ns
Rising Time of SDA and SCL Signals	t_{rI2C}		—	—	1000	ns
Falling Time of SDA and SCL Signals	t_{fI2C}		—	—	300	ns
Setup Time of Stop Condition	t_{SU_STO}		4.0	—	—	µs
Bus Free Time between Stop Condition and Start Condition	t_{BUFI2C}		4.7	—	—	µs
Capacitive Load of Each Bus Line	C_b		—	—	400	pF
Noise Margin at Low Level in Each Connection Device (Including Hysteresis)	V_{nL}		0.1 ×V_{DVCC}	—	—	V
Noise Margin at High Level in Each Connection Device (Including Hysteresis)	V_{nH}		0.2 ×V_{DVCC}	—	—	V
Spike Pulse Width Suppressed by Input Filter	t_{SPI2C}		—	—	—	ns

(2) High-speed Mode

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCL Clock Frequency	f_{SCL}		0	—	400	kHz
Hold Time of Start Condition	t_{HD_STA}		0.6	—	—	µs
Low Level Period of SCL Clock	t_{LOWI2C}		1.3	—	—	µs
High Level Period of SCL Clock	$t_{HIGHI2C}$		0.6	—	—	µs
Setup Time of Start Condition	t_{SU_STA}		0.6	—	—	µs
Data Hold Time (I^2C Bus Device)	t_{HD_DAT}		0	—	0.9	µs
Data Setup Time	t_{SU_DAT}		100	—	—	ns
Rising Time of SDA and SCL Signals	t_{rI2C}		20 + 0.1C_b	—	300	ns
Falling Time of SDA and SCL Signals	t_{fI2C}		20 + 0.1C_b	—	300	ns
Setup Time of Stop Condition	$t_{SU:STO}$		0.6	—	—	µs
Bus Free Time between Stop Condition and Start Condition	t_{BUFI2C}		1.3	—	—	µs
Capacitive Load of Each Bus Line	C_b		—	—	400	pF
Noise Margin at Low Level in Each Connection Device (Including Hysteresis)	V_{nL}		0.1 ×V_{DVCC}	—	—	V
Noise Margin at High Level in Each Connection Device (Including Hysteresis)	V_{nH}		0.2 ×V_{DVCC}	—	—	V
Spike Pulse Width Suppressed by Input Filter	t_{SPI2C}		0	—	50	ns

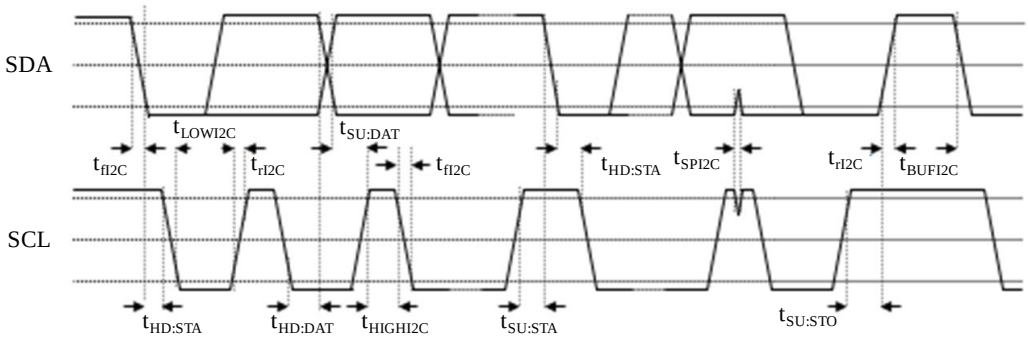
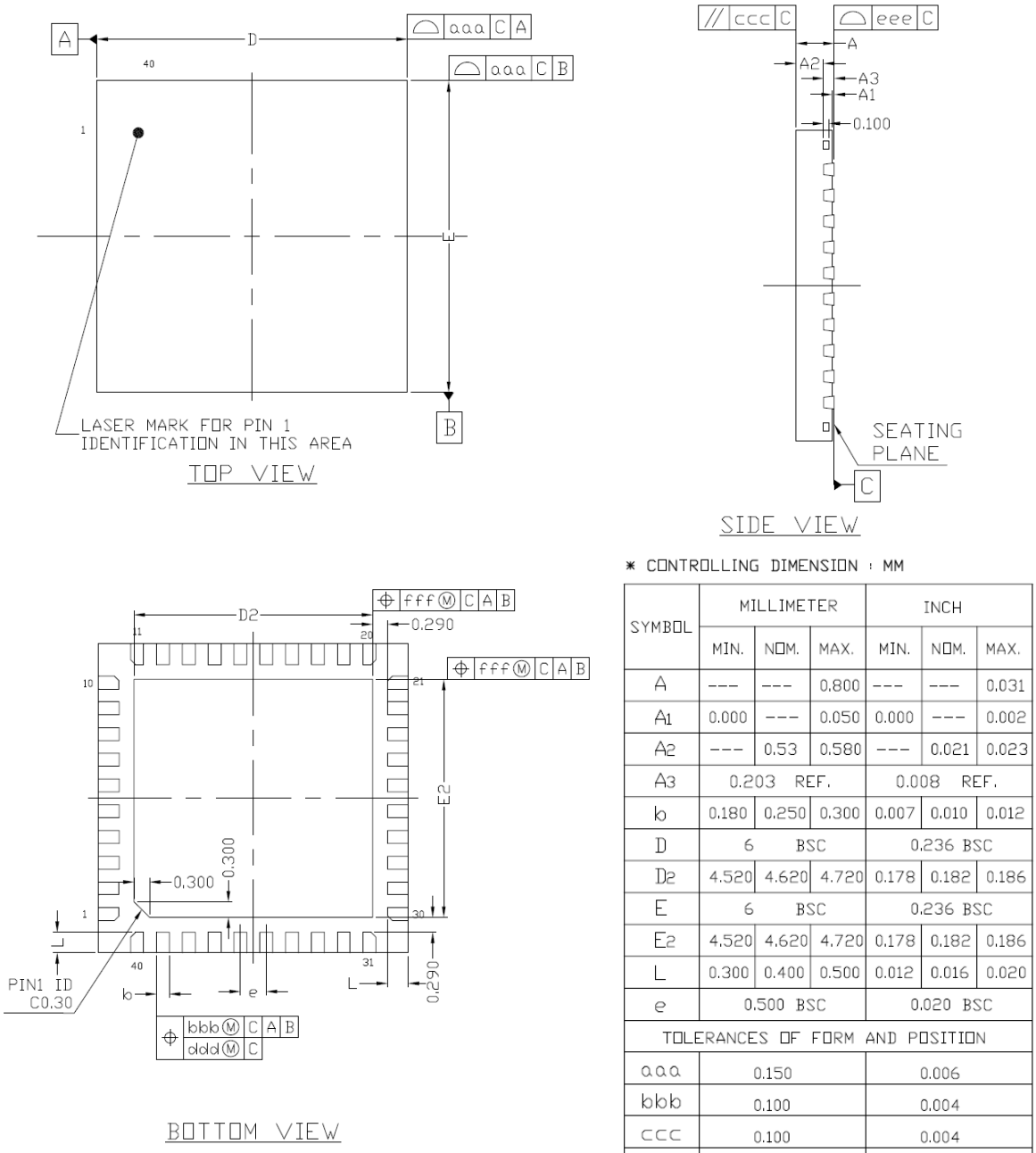


Figure 29-6. I^2C Timing

30. Package



✱ CONTROLLING DIMENSION : MM

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	---	---	0.800	---	---	0.031
A1	0.000	---	0.050	0.000	---	0.002
A2	---	0.53	0.580	---	0.021	0.023
A3	0.203 REF.			0.008 REF.		
b	0.180	0.250	0.300	0.007	0.010	0.012
D	6 BSC			0.236 BSC		
D2	4.520	4.620	4.720	0.178	0.182	0.186
E	6 BSC			0.236 BSC		
E2	4.520	4.620	4.720	0.178	0.182	0.186
L	0.300	0.400	0.500	0.012	0.016	0.020
e	0.500 BSC			0.020 BSC		

TOLERANCES OF FORM AND POSITION		
aaa	0.150	0.006
bbb	0.100	0.004
ccc	0.100	0.004
ddd	0.050	0.002
eee	0.080	0.003
fff	0.100	0.004

Figure 30-1. QFN40 Physical Dimensions

Note: Do not add excessive stress to the device, after the device is mounted on a PCB, to prevent a change of characteristics.

Important Notes

- All data, illustrations, graphs, tables and any other information included in this document (the "Information") as to Sanken's products listed herein (the "Sanken Products") are current as of the date this document is issued. The Information is subject to any change without notice due to improvement of the Sanken Products, etc. Please make sure to confirm with a Sanken sales representative that the contents set forth in this document reflect the latest revisions before use.
- The Sanken Products are intended for use as components of general purpose electronic equipment or apparatus (such as home appliances, office equipment, telecommunication equipment, measuring equipment, etc.). Prior to use of the Sanken Products, please put your signature, or affix your name and seal, on the specification documents of the Sanken Products and return them to Sanken. When considering use of the Sanken Products for any applications that require higher reliability (such as transportation equipment and its control systems, traffic signal control systems or equipment, disaster/crime alarm systems, various safety devices, etc.), you must contact a Sanken sales representative to discuss the suitability of such use and put your signature, or affix your name and seal, on the specification documents of the Sanken Products and return them to Sanken, prior to the use of the Sanken Products. The Sanken Products are not intended for use in any applications that require extremely high reliability such as: aerospace equipment; nuclear power control systems; and medical equipment or systems, whose failure or malfunction may result in death or serious injury to people, i.e., medical devices in Class III or a higher class as defined by relevant laws of Japan (collectively, the "Specific Applications"). Sanken assumes no liability or responsibility whatsoever for any and all damages and losses that may be suffered by you, users or any third party, resulting from the use of the Sanken Products in the Specific Applications or in manner not in compliance with the instructions set forth herein.
- In the event of using the Sanken Products by either (i) combining other products or materials or both therewith or (ii) physically, chemically or otherwise processing or treating or both the same, you must duly consider all possible risks that may result from all such uses in advance and proceed therewith at your own responsibility.
- Although Sanken is making efforts to enhance the quality and reliability of its products, it is impossible to completely avoid the occurrence of any failure or defect or both in semiconductor products at a certain rate. You must take, at your own responsibility, preventative measures including using a sufficient safety design and confirming safety of any equipment or systems in/for which the Sanken Products are used, upon due consideration of a failure occurrence rate and derating, etc., in order not to cause any human injury or death, fire accident or social harm which may result from any failure or malfunction of the Sanken Products. Please refer to the relevant specification documents and Sanken's official website in relation to derating.
- No anti-radioactive ray design has been adopted for the Sanken Products.
- The circuit constant, operation examples, circuit examples, pattern layout examples, design examples, recommended examples, all information and evaluation results based thereon, etc., described in this document are presented for the sole purpose of reference of use of the Sanken Products.
- Sanken assumes no responsibility whatsoever for any and all damages and losses that may be suffered by you, users or any third party, or any possible infringement of any and all property rights including intellectual property rights and any other rights of you, users or any third party, resulting from the Information.
- No information in this document can be transcribed or copied or both without Sanken's prior written consent.
- Regarding the Information, no license, express, implied or otherwise, is granted hereby under any intellectual property rights and any other rights of Sanken.
- Unless otherwise agreed in writing between Sanken and you, Sanken makes no warranty of any kind, whether express or implied, including, without limitation, any warranty (i) as to the quality or performance of the Sanken Products (such as implied warranty of merchantability, and implied warranty of fitness for a particular purpose or special environment), (ii) that any Sanken Product is delivered free of claims of third parties by way of infringement or the like, (iii) that may arise from course of performance, course of dealing or usage of trade, and (iv) as to the Information (including its accuracy, usefulness, and reliability).
- In the event of using the Sanken Products, you must use the same after carefully examining all applicable environmental laws and regulations that regulate the inclusion or use or both of any particular controlled substances, including, but not limited to, the EU RoHS Directive, so as to be in strict compliance with such applicable laws and regulations.
- You must not use the Sanken Products or the Information for the purpose of any military applications or use, including but not limited to the development of weapons of mass destruction. In the event of exporting the Sanken Products or the Information, or providing them for non-residents, you must comply with all applicable export control laws and regulations in each country including the U.S. Export Administration Regulations (EAR) and the Foreign Exchange and Foreign Trade Act of Japan, and follow the procedures required by such applicable laws and regulations.
- Sanken assumes no responsibility for any troubles, which may occur during the transportation of the Sanken Products including the falling thereof, out of Sanken's distribution network.
- Although Sanken has prepared this document with its due care to pursue the accuracy thereof, Sanken does not warrant that it is error free and Sanken assumes no liability whatsoever for any and all damages and losses which may be suffered by you resulting from any possible errors or omissions in connection with the Information.
- Please refer to our official website in relation to general instructions and directions for using the Sanken Products, and refer to the relevant specification documents in relation to particular precautions when using the Sanken Products.
- All rights and title in and to any specific trademark or tradename belong to Sanken and such original right holder(s).

DSGN-CEZ-16003

Revision History

Numbers of sections, figures, tables, and so on may vary depending on revisions.

Revision	Date of Issue	No.	Title	Section	Description
1.1	Apr. 6, 2018	—	—	—	Initial release
1.2	May 25, 2021	13	SYSC	4.5	Corrected the following description: Incorrect: It is required to wait at least 2 cycles before reading the LVDCTRL.LVDIF bit immediately after the LVDCTRL.LVDIF bit is cleared. Correct: It is required to wait at least 3 cycles before reading the LVDCTRL.LVDIF bit immediately after the LVDCTRL.LVDIF bit is cleared.
		14	8051 CPU	5.7.7	Newly added the notes of the CPU and EPU simultaneous access to XDATA space.
		15	GPIO	7.2.44	Corrected the R/W column of the TMRIS[1] and TMRIS[3] bits. Incorrect: R/W Correct: R
		16	DSAC	12.3	Corrected the following register addresses in Table 12-3: Symbol / Incorrect / Correct DSACNTA12 / 0xF888 / 0xF8B0 DSACNTB12 / 0xF889 / 0xF8B1 DSASRC12 / 0xF88A / 0xF8B2 DSADST12 / 0xF88B / 0xF8B3 DSACNTA13 / 0xF88C / 0xF8B4 DSACNTB13 / 0xF88D / 0xF8B5 DSASRC13 / 0xF88E / 0xF8B6 DSADST13 / 0xF88F / 0xF8B7 DSACNTA14 / 0xF890 / 0xF8B8 DSACNTB14 / 0xF891 / 0xF8B9 DSASRC14 / 0xF892 / 0xF8BA DSADST14 / 0xF893 / 0xF8BB DSACNTA15 / 0xF894 / 0xF8BC DSACNTB15 / 0xF895 / 0xF8BD DSASRC15 / 0xF896 / 0xF8BE DSADST15 / 0xF897 / 0xF8BF
		17	TMR	17.2	Added the TPSNF0 to TPSNF3 registers in Table 17-2.
		18	TMR	17.2	Corrected the following register addresses in Table 17-2: Symbol / Incorrect / Correct TBUFAH0 / 0xFA22 / 0xFA21 TBUFAL1 / 0xFA21 / 0xFA22 TBUFBH0 / 0xFA26 / 0xFA25 TBUFBL1 / 0xFA25 / 0xFA26 TBUFAH2 / 0xFA62 / 0xFA61 TBUFAL3 / 0xFA61 / 0xFA62 TBUFBH2 / 0xFA66 / 0xFA65 TBUFBL3 / 0xFA65 / 0xFA66

Revision	Date of Issue	No.	Title	Section	Description
		19	TMR	17.2	Corrected the following register names in Table 17-3: TBUFBL2 Incorrect: Timer2 Buffer A Low Correct:Timer2 Buffer B Low TBUFBL3 Incorrect: Timer3 Buffer A Low Correct: Timer3 Buffer B Low TBUFBH2 Incorrect: Timer2 Buffer A High Correct: Timer2 Buffer B High TBUFBH3 Incorrect: Timer3 Buffer A High Correct: Timer3 Buffer B High
		20	TMR	17.3.4	Corrected the bit setting in Figure 17-2 a): Incorrect: TOACRn.TOCMPA = 0b01, OACRn.TOCLR = 0b10 Correct: TOACRn.TOCMPA =0b10, OACRn.TOCLR = 0b01
		21	SPI	18.2.1.2	Added the following description: In the subordinate mode (SPICR.MSTR = 0), the SPI modes 0 and 2 cannot be used (i.e., only the SPI modes 1 and 3 can be used).
		22	SPI	18.2.1.2	Corrected Table 18-3.
		23	I^2C/SMBUS	19.1	Added “Clock stretching” to Table 19-1.
		24	I^2C/SMBUS	19.2.3	Corrected the following description: Incorrect: Whether the data reception is completed can be confirmed by the ICSR.IRIC0 to ICSR.IRIC5 bit. Correct: Whether the data reception is completed can be confirmed by the ICSR.IRIC3 bit.
		25	I^2C/SMBUS	19.2.6	Added the following descriptions: When the ICSAR.SVA bits = 0, the IC does not judge whether or not the subordinate address matches the ICSAR.SVA bits. The ICSAR.CMD bit of the reception command is updated when the ICSAR.CMD bit matches with either of the following addresses: with the GCA when GCA = enabled; with the SAA when SAA = enabled.
		26	I^2C/SMBUS	19.2.7	Corrected the ICCLK[1:0] bit description: Incorrect: 00: t_{SCLH}/t_{SCLL} is set to 8 cycles of CLKFAST Correct: 00: t_{SCLH}/t_{SCLL} is set to 9 cycles of CLKFAST
		27	I^2C/SMBUS	19.2.7	Added “filter time”; corrected the t_{SCLH} and t_{SCL} definitions in Figure 19-3.
		28	I^2C/SMBUS	19.2.8	Corrected the ICCMD.ACK bit description: Incorrect: When the ACK response is completed, the bit is cleared. When NACK = 1, the bit is not set. Correct: The bit is not set when: ● NACK = 1 ● In the master mode

Revision	Date of Issue	No.	Title	Section	Description
		29	I^2C/SMBUS	19.2.10	Corrected the ICSHTR.ICSHEXP bit description: Incorrect: When not using the LSI with CLKFAST ≤ 12.5 MHz or for SMBUS, set the bit to 0. When using the LSI with CLKFAST > 12.5 MHz or for SMBUS, set the bit to 1. Correct: When using the LSI with CLKFAST ≤ 12.5 MHz or when not using the I^2C as the SMBUS, set the bit to 0. When using the LSI with CLKFAST > 12.5 MHz and the I^2C as the SMBUS, set the bit to 1.
		30	I^2C/SMBUS	19.2.13	Added the description of the cycle setting of the reference timing signal for the violation detection defined by the SMBUS standards.
		31	I^2C/SMBUS	19.2.15	Added the following descriptions: When the subordinate alert address defined by the ICSAIR.SAAEN bit is enabled and the received subordinate address matches the ICSAA.SAA bits, the LSI operates as the subordinate device specified by the master device. When ICSAA.SAA bits = 0, the IC does not judge whether or not the subordinate address matches the ICSAA.SAA bits. The ICSAA.CMD bit of the reception command is updated when the ICSAR.CMD bit matches with either of the following addresses regardless of enabling/disabling the SAA: with the SVA; with the GCA when GCA = enabled; with the SSA when SSA = enabled.
		32	I^2C/SMBUS	19.2.15	Corrected the ICSAA[0] bit name and description: Incorrect: Reserved Correct: CMD
		33	I^2C/SMBUS	19.4 to 19.7	Modified the following descriptions: - Corrected the steps. - Added the interrupt enable bits. - Added the operations when the GCA and SAA are enabled.
		34	I^2C/SMBUS	19.4	Corrected the generation timing of ACK/NACK signal of SDA in Figure 19-8 b).
		35	I^2C/SMBUS	19.5	Modified Figure 19-9 b) as follows: - Corrected the SCL (Subordinate output) to show its output per data transmission. - Corrected the SDA (Subordinate output) to show its becoming high level after data transmission. - Added the broken line to NA on the SDA (Master output).
		36	I^2C/SMBUS	19.8	Modified the following descriptions: - Added the description of the GCA. - Deleted "In the slave mode, the ICSAR register can be read and be written, and the SAA register can be read only." - Corrected the priority when the GCA, SAA, and ICSAR match with subordinate address. - Added the description when ICSAR = 0 and SAA = 0. - Deleted "Note that the address written to the slave device does not necessarily match the address indicated in the ICSAIR register." - Added the way of the judgement whether the command is writing or reading.

Revision	Date of Issue	No.	Title	Section	Description
		37	POC	26.1	Corrected the LVLEVEN signal to the inverted signal of the LVLENDIS signal.
		38	POC	26.4.2	Newly added the notes of the POCCRn.EN bit.
		39	POC	26.4.3	Newly added the notes of the control delay addition.
		40	SCID	28.4	Corrected the initial value of the following registers: Symbol / Incorrect / Correct UART_CR 0x00 0x03 UART_TXFIFO_SR 0x00 0x80 UART_RXFIFO_SR 0x00 0x80 UART_BAUD_L/H 0xFF 0x00
		41	SCID	28.4.3	Corrected the initial value of the UART_CR[1:0] bits: Incorrect: 0 Correct: 1
		42	SCID	28.4.7	Corrected the initial value of the UART_TXFIFO_SR[7] bit: Incorrect: 0 Correct: 1
		43	SCID	28.4.9	Corrected the initial value of the UART_RXFIFO_SR[7] bit: Incorrect: 0 Correct: 1
		44	SCID	28.4.12	Corrected the initial value of the BAUD_L[7:0] and BAUD_H[7:0] bits: Incorrect: 1 Correct: 0
		45	SCID	28.5.5	Corrected the following description: Incorrect: Where, N is the 16-bit value that the values of the UART_BAUD_H.TOUT_H (8 bits) and UART_BAUD_L.TOUT_L (8 bits) bits are connected. N must be set >3. Correct: Where, N is the 16-bit value that the values of the UART_BAUD_H.BAUD_H (8 bits) and UART_BAUD_L.BAUD_L (8 bits) bits are connected. N must be set >3.
		46	Electrical Characteristics	29.3.13.2	Modified the following specifications: - Added the conditions of t_{SCK}. - Corrected the t_{SCK} specification: Incorrect: 80 ns Correct: 33.33 ns - Corrected the t_{DSPI} specification: Incorrect: 4 ns Correct: 5 ns
1.3	Jul. 14, 2021	47	ADC	22.2.1	- Corrected the ADCn[2] bit and modified the Description column: Incorrect: Reserved Correct: MODE3 - Modified the Description column of the ADCn.MODE1 bit.

Revision	Date of Issue	No.	Title	Section	Description
					- Corrected the ADCn[0] bit and modified the Description column: Incorrect: Reserved Correct: MODE3 - Added Table 22-4. Restricted Setting Conditions for ADCn.MODE1 and ADCn.MODE3