Dear customer

ROHM Co., Ltd. ("ROHM"), on the 1st day of April, 2024,
has absorbed into merger with 100%-owned subsidiary of LAPIS Technology Co., Ltd.

Therefore, all references to "LAPIS Technology Co., Ltd.", "LAPIS Technology"
and/or "LAPIS" in this document shall be replaced with "ROHM Co., Ltd."
Furthermore, there are no changes to the documents relating to our products other than
the company name, the company trademark, logo, etc.

Thank you for your understanding.

ROHM Co., Ltd.
April 1, 2024

ML7416N-060

Sub-GHz (900 MHz Band) Broadband RF Transceiver IC with Built-in MCU

■ 1. General Description

ML7416 is a low power consumption LSI for sub-GHz broadband radio communication, which integrates the MCU block and RF block in a single chip.

●Product Name ML7416N-660ALA

●Application Remote control
Home, Building Security
Sensor Network
Smart Meters

The features are shown below. *** For details of the RF block, refer to ML7396B Data Sheet (FJDL7396A_B_E).**

- Industry-standard ARM® Cortex®-M0+ CPU core * Maximum operating frequency: 40 MHz
- 512 KB flash memory (for program [256 KB x 2 bank configuration is possible]. Can be used as data flash)
- 64 KB RAM (for data. Whether to hold data in low power consumption mode can be selected)
- Start-stop synchronous serial communication interface (UART) * Max. 3 ch
- Synchronous serial communication interface (SPI) * Max. 2 ch (additionally, 1 ch for SPI dedicated to RF control, which does not exist on the external port)
- WDT
- General-purpose IO
- Timer * 32-bit timer x 10 ch. Among them, 4 ch (2 sets) can be used as 64-bit timer by the cascade connection
- RTC
- Flexible timer (FTM) (PWM mode, etc.)
- I2C * Master/slave supported
- RF control transmit/receive data interface (DIO) * As this is dedicated to RF control, it does not have an external port
- Random number generation circuit (RAND)
- Clock correction counter (CLK_Timer)
- SWD (two-wire serial wire debug port)
- XTAL OSC * 32.768 kHz
- PLL * Multiplying/dividing 32 kH
- ADC * 10 bits, Max. 3 ch (Max. 2 ch when using the temperature sensor function)

ROHM SEMICONDUCTOR

- Voltage drop detection (LVD)
- Temperature sensor (TEMP)
- CR OSC * High-speed, low-speed
- AES * ECB, CBC, CTR, CCM, GCM, CFB and OFB supported
- DMA controller * Transfer between SPI and RAM and between AES and RAM, 4 ch
- Flash DMA controller * Write to flash, Verify

- Supply voltage 1.8 to 3.6 V (when the transmission power is set to 1 mW mode)
  2.3 to 3.6 V (when the transmission power is set to 10 mW mode)
  2.6 to 3.6 V (when the transmission power is set to 20 mW mode)
- Operating temperature -40 to 85 °C

- Current consumption

At DeepSleep	2uA(Typ.)
At idle	11 mA(Typ.)
At reception	24 mA (Typ.)
At transmission 1 mW	22 mA (Typ.)
10 mW	33 mA(Typ.)
20 mW	41 mA(Typ.)

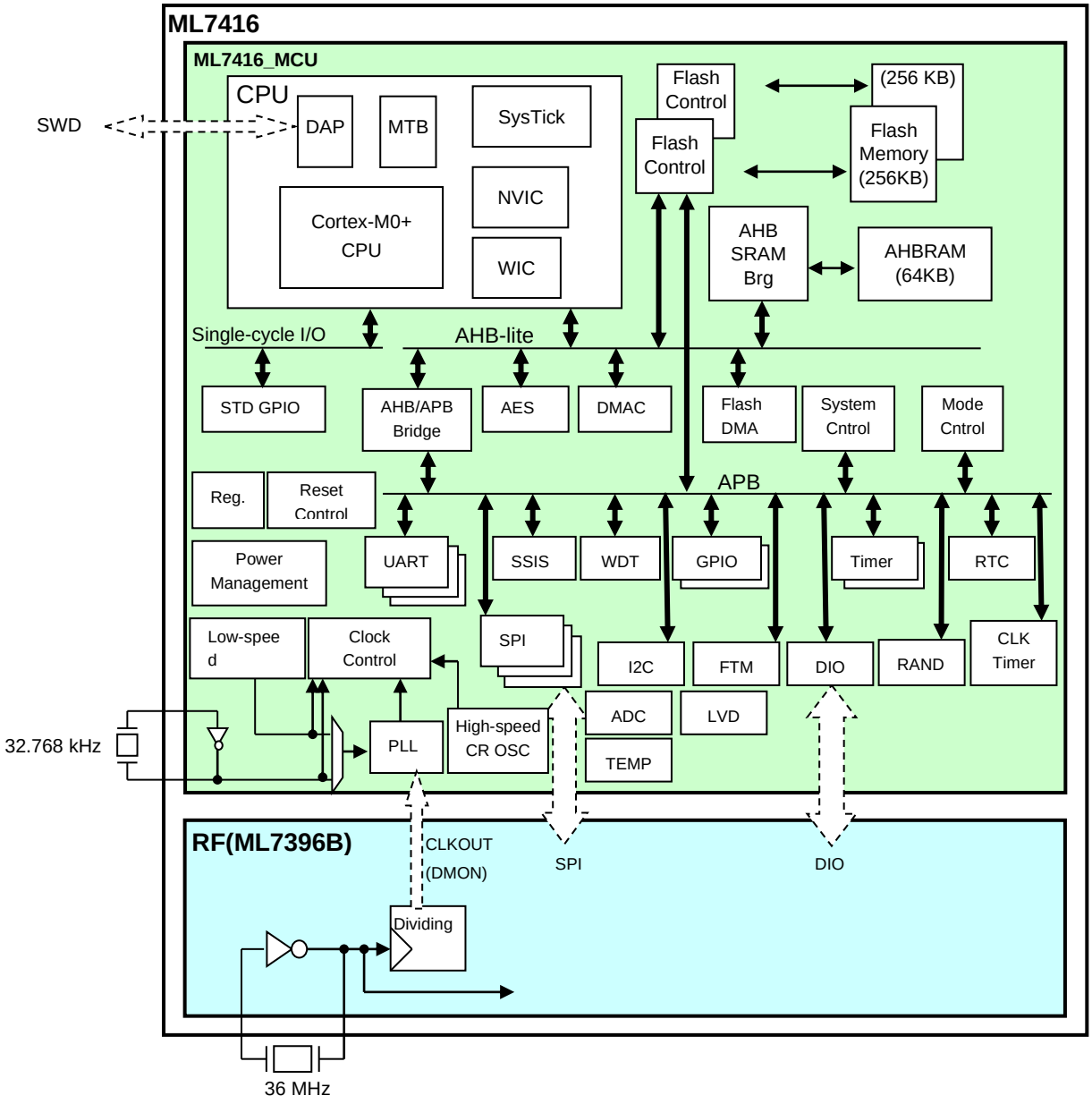
- Package

  81-pin BGA (MCP product/P-LFBGA81-1010-1.00-1-MC)
  10 mm x 10 mm (pin pitch: 1 mm)
  Lead-free package conforming to RoHS

****

****

■ 2. Block Diagram



■ 3. Pin Layout

*** Please contact us before board design.**

○ 81BGA (10 mm x 10 mm, pin pitch: 1 mm)

1	2	3	4	5	6	7	8	9	
VDDIO_RF	GPIOA0	GPIOA1	GPIOA2	GPIOA4	GPIOA5	GPIOA6	GPIOA7	GND_CPU	A
DCNT	VDD_PA	GPIOA3	MODE0	MODE1	GPIOA12	GPIOA8	GND_CPU	GPIOA9	B
TRX_SW	ANT_SW	VDD_REG	REG_CORE_CPU	TEST_CPU	RESETN	GND_CPU	ADC0	GPIOA10	C
REG_PA	TEST	A_MON	GND_RF	GND_RF	REGPDIN	GND_CPU	ADC1	GPIOA11	D
GND_RF	GND_RF	GND_RF	GND_RF	GND_RF	GND_RF	VDDIO_CPURF	ADC2	CXOUT	E
GND_RF	GND_RF	GND_RF	GND_RF	GND_RF	GND_RF	VPP	VDD_REG_CPU	CXIN	F
PA_OUT	GND_RF	VDD_CP	GND_RF	GND_RF	GND_RF	TCXO	VDDIO_CPU	SWCK	G
LNA_P	GND_RF	VDD_RF	VDD_IF	VDD_VCO	REG_OUT	REG_CORE	VDDIO_CPU	SWD	H
GND_RF	LP1	IND1	IND2	VB_EXT	VBG	XIN	XOUT	VDDIO_CPU	J

TOP View

■ 4. Pin Description

*** The pin names and LSI numbers may be changed in the future.**

Input/output definition

I_{RF}	:RF input pin
O_{RF}	:RF output pin
I_A	:Analog input pin
O_A	:Analog output pin
I_{OS}	:36 MHz oscillation circuit input pin
O_{OS}	:36 MHz oscillation circuit output pin
I_{OSL}	:32.768 kHz oscillation circuit input pin
O_{OSL}	:32.768 kHz oscillation circuit output pin
I	:Digital input pin
O	:Digital output pin
Is	:Schmitt trigger input pin
O_D	:Open drain pin
oZ	:High-impedance output pin

4-1. Power Supply

Pin name	LSI No.	I/O	Active level	Attribute/Value at reset	Functional description
VDDIO_CPU	G8 H8 J9	Power supply	-	- / -	Power supply pin for the digital I/O (Typ. 3.3 V)
VDDIO_RF	A1	Power supply	-	- / -	Power supply pin for RF (Typ. 3.3 V)
VDD_REG	C3	Power supply	-	- / -	Regulator voltage supply pin (RF side/Typ. 3.3 V)
VDD_REG_CPU	F8	Power supply	-	- / -	Regulator voltage supply pin (CPU side/Typ. 3.3 V)
VDD_PA	B2	Power supply	-	- / -	Power supply pin for PA (Typ. 3.3 V)
VDDIO_CPURF	E7	Power supply	-	- / -	Power supply pin for supply from CPU to RF (Typ. 3.3 V)
VDD_RF	H3	Power supply	-	- / -	Power supply pin for LNA/MIX (Typ. 1.5 V)
VDD_IF	H4	Power supply	-	- / -	Power supply pin for IF (Typ. 1.5 V)
VDD_VCO	H5	Power supply	-	- / -	Power supply pin for VCO (Typ. 1.5 V)
VDD_CP	G3	Power supply	-	- / -	Power supply pin for CP (Typ. 1.5 V)
GND_RF	D4 to D5 E1 to E6 F1 to F6 G2,G4 to G6 H2 J1	Ground	-	- / -	Ground pin (for RF)
GND_CPU	A9 B8 C7 D7	Ground	-	- / -	Ground pin (for CPU)

4-2. Regulator Interface

Pin name	LSI No.	I/O	Active level	Attribute/Value at reset	Functional description
REG_OUT	H6	-	-	- / -	Regulator output (Typ. 1.5V) C connection (10 uF) * Outputs 0 V at sleep state.
REG_PA	D1	-	-	- / -	PA regulator output pin * Outputs 0 V at sleep state.
VBG	J6	-	-	- / -	C connection pin (RF side/0.1 uF)
REGPDIN	D6	I	H	I / -	Regulator power down pin * Fixed to "L" input in normal operation
REG_CORE	H7	-	-	- / -	Monitor pin for power supply for digital core (RF side /Typ. 1.5 V)/C connection(10 uF))
REG_CORE_CPU	C4	-	-	- / -	Monitor pin for power supply for digital core (CPU side/Typ. 1.5 V/C connection (0.22 uF))

4-3. RF Interface

Pin name	LSI No.	I/O	Active level	Attribute/Value at reset	Functional description
LNA_P	H1	I_{RF}	-	I	RF antenna input pin
PA_OUT	G1	O_{RF}	-	O	RF antenna output pin
IND1	J3	-	-	- / -	External inductor connection pin
IND2	J4	-	-	- / -	External inductor connection pin
LP1	J2	-	-	- / -	Loop filter connection pin
VB_EXT	J5	-	-	- / -	Internal bias averaging capacitor connection pin
A_MON	D3	O_{RF}	-	Hi-Z	Test pin for analog monitor, IF block, and analog circuit

4-4. ADC Interface

Pin name	LSI No.	I/O	Active level	Attribute/Value at reset	Functional description
ADC0	C8	I_A	-	I / -	ADC input pin 0
ADC1	D8	I_A	-	I / -	ADC input pin 1
ADC2	E8	I_A	-	I / -	ADC input pin 2 * Input from this pin is disabled when using the temperature sensor

4-5. CPU Interface

Pin name	LSI No.	I/O	Active level	Attribute/Value at reset	Functional description
GPIOA0	A2	I / O	-	oZ / -	Primary function: General-purpose pin
		I	-		Secondary function: UART data input pin (RXD)
		I / O	P or N		Tertiary function: SPI clock pin (SCK)
		Is / O_D	P or N		Quartic function: I2C clock pin (SCL)
		I / O	-		Quintic function: Single cycle IO pin
GPIOA1	A3	I / O	-	oZ / -	Primary function: General-purpose pin
		O	-		Secondary function: UART data output pin (TXD)
		I / O	L		Tertiary function: SPI enable pin (SSN)
		Is / O_D	-		Quartic function: I2C data I/O pin (SDA)
		I / O	-		Quintic function: Single cycle IO pin
GPIOA2	A4	I / O	-	oZ / -	Primary function: General-purpose pin
		I	L		Secondary function: UART CTS (Clear To Send) pin
		I / O	-		Tertiary function: SPI data I/O pin 1 (MISO)
		I / O	-		Quartic function: FTM I/O pin
		I / O	-		Quintic function: Single cycle IO pin
GPIOA3	B3	I / O	-	oZ / -	Primary function: General-purpose pin
		O	L		Secondary function: UART RTS (Ready To Send) pin
		I / O	-		Tertiary function: SPI data I/O pin 2 (MOSI)
		I / O	-		Quartic function: Reserved
		I / O	-		Quintic function: Single cycle IO pin

Pin name	LSI No.	I/O	Active level	Attribute/Value at reset	Functional description
GPIOA4	A5	I / O	-	oZ / -	Primary function: General-purpose pin
		I	-		Secondary function: UART data input pin (RXD)
		I / O	P or N		Tertiary function: SSI slave clock pin (SSICK)
		Is / O_D	P or N		Quartic function: I2C clock pin (SCL)
		I / O	-		Quintic function: Single cycle IO pin
GPIOA5	A6	I / O	-	oZ / -	Primary function: General-purpose pin
		I	L		Secondary function: UART data output pin (TXD)
		I	L		Tertiary function: SSI slave enable pin (SSIN)
		Is / O_D	-		Quartic function: I2C data I/O pin (SDA)
		I / O	-		Quintic function: Single cycle IO pin
GPIOA6	A7	I / O	-	oZ / -	Primary function: General-purpose pin
		I	L		Secondary function: UART CTS (Clear To Send) pin
		O	-		Tertiary function: SSI Slave data output pin (TXD)
		I / O_D	-		Quartic function: FTM I/O pin
		I / O	-		Quintic function: Single cycle IO pin
GPIOA7	A8	I / O	-	oZ / -	Primary function: General-purpose pin
		O	L		Secondary function: UART RTS (Ready To Send) pin
		I	-		Tertiary function: SSI Slave data input pin (RXD)
		I / O	-		Quartic function: Reserved
		I / O	-		Quintic function: Single cycle IO pin
GPIOA8	B7	I / O	-	oZ / -	Primary function: General-purpose pin
		I	-		Secondary function: UART data input pin (RXD)
		I / O	P or N		Tertiary function: SPI clock pin (SCK)
		Is / O_D	P or N		Quartic function: I2C clock pin (SCL)
		I / O	-		Quintic function: Single cycle IO pin
GPIOA9	B9	I / O	-	oZ / -	Primary function: General-purpose pin
		O	-		Secondary function: UART data output pin (TXD)
		I / O	L		Tertiary function: SPI enable pin (SSN)
		Is / O_D	-		Quartic function: I2C data I/O pin (SDA)
		I / O	-		Quintic function: Single cycle IO pin

Pin name	LSI No.	I/O	Active level	Attribute/Value at reset	Functional description
GPIOA10	C9	I / O	-	oZ / -	Primary function: General-purpose pin
		I	L		Secondary function: UART CTS (Clear To Send) pin
		I / O	-		Tertiary function: SPI data I/O pin 1 (MISO)
		I / O	-		Quartic function: FTM I/O pin
		I / O	-		Quintic function: Single cycle IO pin
GPIOA11	D9	I / O	-	oZ / -	Primary function: General-purpose pin
		O	L		Secondary function: UART RTS (Ready To Send) pin
		I / O	-		Tertiary function: SPI data I/O pin 2 (MOSI)
		I / O	-		Quartic function: Reserved
		I / O	-		Quintic function: Single cycle IO pin
GPIOA12	B6	I / O	-	oZ / -	In 1 bank mode: General-purpose pin In 2 bank mode: System mode input pin (for software) 0: User application mode 1: ISP mode

4-6. Debugger Interface

Pin name	LSI No.	I/O	Active level	Attribute/Value at reset	Functional description
SWCK	G9	I	-P or N	I / -	SWD clock input pin
SWD	H9	I / O	H or L	I / -	SWD data I/O pin

4-7. Others

Pin name	LSI No.	I/O	Active level	Attribute/Value at reset	Functional description
RESETN	C6	Is	L	I / -	Hardware reset pin
XIN	J7	Ios	P or N	I	36 MHz crystal oscillator connection pin 1 * Connect this pin to GND when using an external clock.
XOUT	J8	Oos	P or N	O	36 MHz crystal oscillator connection pin 2 * Connect this pin to GND when using an external clock.
TCXO	G7	I_A	-	I	External clock (TCXO) input pin * Connect this pin to GND when using an oscillator.
CXIN	F9	I_{OSL}	P or N	I	32.768 kHz crystal oscillator connection pin 1
CXOUT	E9	O_{OSL}	P or N	O	32.768 kHz crystal oscillator connection pin 2
TRX_SW	C1	O	H or L or OD	O / L	Transmit/receive switch pin
ANT_SW	C2	O	H or L or OD	O / L	Diversity switch pin
TEST	D2	I	H	I / -	Test mode pin * Always fix this pin to "L" input in normal operation
TEST_CPU	C5	I	H	I / -	Test mode pin * Always fix this pin to "L" input in normal operation
MODE0	B4	I	H or L	I / -	Remapping pin 0: The program executes from the address 0 in the internal Flash ROM 1: The program executes from the boot area in the internal Flash ROM
MODE1	B5	I	H or L	I / -	Test mode pin * Always fix this pin to "L" input in normal operation
DCNT	B1	O	H or L or OD	oZ / L	External PA control pin
VPP	F7	-	H	- / -	High voltage application pin for flash core test * Normally, leave this pin open.

4-8. Handling of Unused Pins

See below for handling of unused pins.

Pin name	Recommended treatment
XOUT	GND (when TCXO is used)
XIN	GND (when TCXO is used)
TCXO	GND (when an oscillator is used)
A_MON	Open
ANT_SW	Open
DCNT	Open
VPP	Open
ADC0	Open
ADC1	Open
ADC2	Open
SWCK	Connect this pin to a pull-up or pull-down resistor.
SWD	Connect this pin to a pull-up or pull-down resistor.

■ 5. Electrical Characteristics

*** The electrical characteristics may be changed as a result of evaluation or any other reason.**

5-1.Absolute Maximum Ratings

Item	Symbol	Conditions	Rating	Unit
Power supply voltage (I/O) (*1)	VDDIO	Ta=-40 to 85 °C GND=0 V	-0.3 to +4.6	V
Power supply voltage (RF) (*2)	VDDRF		-0.3 to +2.0	V
Digital input voltage	VDIN		-0.3 to VDDIO+0.3	V
RF input voltage	VRFIN		-1.0 to +2.0	V
Analog input voltage	VAIN		-0.3 to VDDIO+0.3	V
Analog input voltage 2 (*3)	VAIN2		-0.3 to VDDRF+0.3	V
TCXO input voltage	VTCXO		-0.3 to +1.75	V
Digital output voltage	VDO		-0.3 to VDDIO+0.3	V
RF output voltage	VRFO		-0.3 to VDDRF+1.9	V
Analog output voltage	VAO		-0.3 to VDDIO+0.3	V
Analog output voltage 2 (*4)	VAO2		-0.3 to VDDRF+0.3	V
Digital input current	IDI		-10 to +10	mA
RF input current	IRF		-2 to +2	mA
Analog input current	IAI		-2 to +2	mA
Analog input current 2 (*3)	IAI2		-2 to +2	mA
TCXO input current	ITCXO		-2 to +2	mA
Digital output current	IDO		-8 to +8	mA
RF output current	IRFO		-2 to +60	mA
Analog output current	IAO		-2 to +2	mA
Analog output current 2 (*4)	IAO2		-2 to +2	mA
Power dissipation	PD	Ta=+25 °C	300	mW
Storage temperature	Tstg	-	-55 to +150	℃

(*1) VDDIO_CPU, VDDIO_RF, VDD_REG, VDD_REG_CPU, VDD_PA, and VDDIO_CPURF pins

(*2) VDD_RF, VDD_IF, VDD_VCO, and VDD_CP pins

(*3) XIN, TCXO, and CXIN pins

(*4) XOUT and CXOUT pins

5-2. Recommended Operating Conditions

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Power supply voltage (I/O)	VDDIO	VDD_IO* pin and VDD_REG* pin	1.8	3.3	3.6	V
Power supply voltage (PA)	VDD_PA	VDD_PA pin Transmission power +1 mW mode	1.8	3.3	3.6	V
		VDD_PA pin 10 mW mode	2.3	3.3	3.6	V
		VDD_PA pin 20 mW mode	2.6	3.3	3.6	V
Power supply voltage (RF) (*2)	VDDRF	VDD_RF pin, VDD_IF pin, VDD_VCO pin and VDD_CP pin	1.4	1.5	1.6	V
Operating temperature	Ta	-	-40	+25	+85	°C
Digital input rise time	tIR1	Digital input pins (*1)	-	-	20	ns
Digital input fall time	tIF1	Digital input pins (*1)	-	-	20	ns
Digital output load	CDL	All digital output pins	-	-	20	pF
Master clock 1 36 MHz crystal oscillator frequency	FMCK1	XIN pin, XOUT pin	-20 ppm (*3)	36	+20 ppm (*3)	MHz
Master clock 2 36 MHz TCXO frequency	FMCK2	TCXO pin	-20 ppm (*3)	36	+20 ppm (*3)	MHz
TCXO	VTCXO	DC cut	0.8	-	1.5	Vpp
Slow clock 32.768 kHz crystal oscillator frequency	FSCK1	CXIN pin, CXOUT pin	-20ppm	32.768	+20ppm	kHz
RC clock 1 High-speed CR clock frequency	FRCK1		-15 %	34	+15 %	MHz
RC clock 2 Low-speed CR clock frequency	FRCK21	Other than DeepSleep	-5 %	32	+5 %	kHz
	FRCK22	DeepSleep	-15 %	32	+15 %	kHz
SPI clock input frequency	FSCLK	SCLK pin	-	-	CPUCLK/4	MHz
SPI clock input duty ratio	DSCLK	SCLK pin	45	50	55	%
RF channel frequency	FRF	LNA_P and PA_OUT pins	896	-	960	MHz

(*1) Pins described as I or Is in the Input/output column in "Pin Description".

(*2) Use the REG_OUT output of this LSI.

(*3) If set to 10 kbps, the maximum is +10 ppm, and the minimum is -10 ppm.

(*4) These values are provided under the condition of 25 °C. Under the condition of -40 to 85 °C, the maximum value is +150 ppm, and the minimum value -150 ppm.

[Notices]

The electrical characteristics are measured under the recommended operating conditions above, unless otherwise specially noted.

The timings are measured at the 20 % and 80 % levels of VDDIO, unless otherwise specially noted.

5-3. Common Characteristics

Item	Symbol	Conditions	Min.	Typ. (*2)	Max.	Unit
Supply current (*1)	IDD1	Sleep state (*3)	-	2	5.6	µA
	IDD2	Idle state	-	11	14	mA
	IDD3	RF receiving state (*4)	-	24	32	mA
	IDD4	RF transmitting state (1 mW) (*4)	-	22	32	mA
	IDD5	RF transmitting state (10 mW) (*4)	-	33	47	mA
	IDD6	RF transmitting state (20 mW) (*4)	-	41	55	mA
High level input voltage	VIH1	Digital input pin	VDDIOx0.75	-	VDDIO	V
	VIH2	XIN pin	VDDRFx0.9	-	VDDRF	V
Low level input voltage	VIL1	Digital input pin	0	-	VDDIOx0.18	V
	VIL2	XIN pin	0	-	VDDRFx0.1	V
Schmitt trigger high level decision threshold value	VT+	Digital pins with schmitt trigger	-	1.2	VDDIOx0.75	V
Schmitt trigger low level decision threshold value	VT-	Digital pins with schmitt trigger	VDDIOx0.18	0.8	-	V
Input leakage current	IIH1	Digital input pin	-1	-	3.6	µA
	IIH2	XIN pin	-0.3	-	0.3	µA
	IIH3	CXIN pin	-0.3	-	0.3	µA
	IIL1	Digital input pin	-1	-	1	µA
	IIL2	XIN pin	-0.3	-	0.3	µA
	IIL3	CXIN pin	-0.3	-	0.3	µA
Tri-state Output leakage current	IOZH1	Digital I/O pin	-1	-	3.6	µA
	IOZL1	Digital I/O pin	-1	-	1	µA
High level output voltage	VOH	IOH=-4 mA /-2 mA (*5)	VDDIOx0.8	-	VDDIO	V
Low level output voltage	VOL	IOL=4 mA /2 mA (*5)	0	-	0.3	V
Pin capacitance	CIN	input pin	-	6	-	pF
	COUT	output pin	-	9	-	pF
	CRFIO	RF I/O pin	-	9	-	pF
	CAI	Analog input pin	-	20	-	pF

(*1) The power supply current is the total current of all power supply pins.

(*2) The "Typ." value is the center value under the condition of VDDIO = 3.3 V and 25 °C.

(*3) The "Typ." and "Max." values are under the condition of 25 °C.

(*4) Values when the data transfer speed is 100 kpbs and the frequency is 920 MHz.

(*5) This condition applies to TRX_SW, ANT_SW, DCNT, GPIOA0-12, and SWD. For TRX_SW, ANT_SW, and DCNT, only 4 mA mode is used.

(*6) REG_CORE pin and REG_OUT pin. REG_OUT outputs 0 V at sleep state.

5-4. RF Characteristics

Modulated data rate	:	10 kbps/ 20 kbps/ 40 kbps/ 50 kbps / 100 kbps / 150 kbps/ 200 kbps/ 400 kbps
Modulation method	:	Binary GFSK
Channel spacing	:	200 kHz / 400 kHz / 600 kHz
Frequency range	:	A frequency from 750 MHz to 1 GHz can be set by changing external circuit constants.

The measurement point is at antenna end specified in the recommended circuits.

Characteristics not described here and ones of 400 kbps (optional) will be provided separately as reference data.

5.4.1 [TX Characteristics]

Item	Conditions	Min.	Typ.	Max.	Unit
Transmitter power output	When set to 20 mW (13 dBm) mode	9	13	15	dBm
	When set to 10 mW (10 dBm) mode	6	10	12	dBm
	When set to 1 mW (0 dBm) mode	-4	0	2	dBm
Adjustment range of frequency shift [Fdev] (*1)		-	-	2,250	kHz
920 MHz band (920.5 MHz to 928.1 MHz)					
Occupied bandwidth	n: Number of unit channels (n = 1, 2, 3, 4 or 5)	-	-	200 *n	kHz
Power at the edges of radio channel	20 mW mode (920.5 MHz to 922.3 MHz)	-	-	-7	dBm
	10 mW mode	-	-	-10	dBm
	1 mW mode	-	-	-20	dBm
Adjacent channel leakage [ACP]	20 mW mode ± 1 ch, bandwidth 200 kHz	-	-33	-15	dBm
	10 mW mode ± 1 ch, bandwidth 200 kHz	-	-39	-18	dBm
	1 mW mode ± 1 ch, bandwidth 200 kHz	-	-47	-26	dBm
Spurious emission level (20 mW mode)	710 MHz or lower, 100 kHz band	-	-65	-36	dBm
	Higher than 710 MHz to 900 MHz, 1 MHz band	-	-70	-55	dBm
	Higher than 900 MHz to 915 MHz, 100 kHz band	-	-72	-55	dBm
	Higher than 915 MHz to 930 MHz, 100 kHz band (Excluding within 200 + 100xnkHz above and below the channel frequency, however, within 100 + 100xn kHz above and below for 920.5 MHz to 922.3 MHz. n is the number of concurrently used channels)	-	-51	-36	dBm
	Higher than 930 MHz to 1000 MHz, 100 kHz band	-	-70	-55	dBm
	Higher than 1000 MHz to 1215 MHz, 1 MHz band	-	-75	-45	dBm
	Higher than 1215 MHz, 1 MHz band (Equal to or higher than the 2nd harmonics)	-	-40	-30	dBm
915 MHz band (902 MHz to 928 MHz)					
6dB bandwidth	Frequency shift = 171 kHz	500	-	-	kHz
Power spectrum density	20 mW mode, frequency shift = 171 kHz, 3 kHz band	-	-	8	dBm
Spurious emission level (20 mW mode)	900 MHz or below	-	-65	-56	dBm
	Higher than 960 MHz (2nd harmonics or higher)	-	-50	-41	dBm
868 MHz band (863 MHz to 870 MHz) (*2)					
Spurious emission level (10 mW mode)	Higher than 1000 MHz (2nd harmonics or higher)	-	-35	-30	dBm

*1 While the adjustment range is described as above, the possible maximum value depends on the RF channel frequency to be used.

RF channel frequency ± frequency shift should not include a multiple of 36 MHz (864 MHz, 900 MHz, 936 MHz, and so on).

Example) For 902 MHz, 2,000 kHz can be set at a maximum.

*2 863.5 MHz to 866.2 MHz cannot be used. For details, refer to "Setting channel frequency" in ML7396B Data Sheet.

5.4.2 [RX Characteristics]

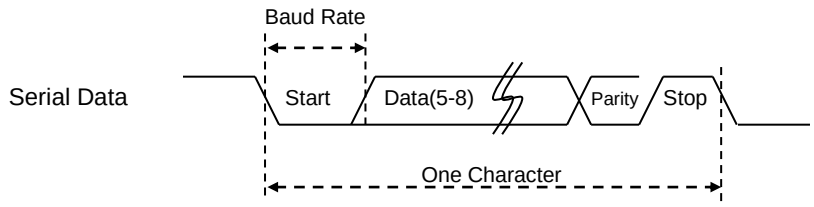
Item	Conditions	Min.	Typ.	Max.	Unit
920 MHz band (920.5 MHz to 928.1 MHz)					
Minimum receiver sensitivity BER < 0.1 %	50 kbps mode (*1)	-	-107	-102	dBm
	100 kbps mode (*1)	-	-105	-100	dBm
	200 kbps mode (*1)	-	-102	-97	dBm
Maximum receiver input level	50 kbps mode, 100 kbps mode, 200 kbps mode	0	-	-	dBm
Receiver C/I adjacent interference	50 kbps mode	20	35	-	dB
	100 kbps mode	20	35	-	dB
	200 kbps mode	20	35	-	dB
Receiver C/I second adjacent interference	50 kbps mode	30	45	-	dB
	100 kbps mode	30	45	-	dB
	200 kbps mode	30	45	-	dB
Minimum energy detection level (ED value)		-	-	-100	dBm
Energy detection range	Dynamic range	60	70	-	dB
Energy detection accuracy		-6	-	+6	dB
Collateral emission level ARIB T108 measurement condition 915.9MHz to 916.9MHz 920.5MHz to 929.7MHz	710 MHz or lower, 100 kHz band	-	<-93	-54	dBm
	Higher than 710 MHz to 900 MHz, 1 MHz band	-	<-83	-55	dBm
	Higher than 900 MHz to 915 MHz, 100 kHz band	-	<-93	-55	dBm
	Higher than 915 MHz to 930 MHz, 100 kHz band	-	-63	-54	dBm
	Higher than 930 MHz to 1000 MHz, 100 kHz band	-	<-93	-55	dBm
	Higher than 1000 MHz, 1 MHz band	-	-57	-47	dBm
915 MHz band (902 MHz to 928 MHz)					
Minimum receiver sensitivity BER < 0.1 %	100 kbps mode (modulation index = 1) (*1)	-	-106	-99	dBm
	150 kbps mode (modulation index = 0.5) (*1)	-	-102	-96	dBm
	200 kbps mode (modulation index = 1) (*1)	-	-102	-96	dBm
	100 kbps mode (frequency shift: 171 kHz)	-	-100	-87	dBm
	150 kbps mode (frequency shift: 171 kHz)	-	-97.5	-84	dBm
	200 kbps mode (frequency shift: 171 kHz)	-	-96.5	-83	dBm
868 MHz band (863 MHz to 870 MHz) (*2)					
Minimum receiver sensitivity BER < 0.1 %	50 kbps mode (*1)	-	-108	-102	dBm
	100 kbps mode (*1)	-	-106	-100	dBm
	200 kbps mode (*1)	-	-102	-97	dBm
Collateral emission level	1000 MHz or below (local frequency)	-	-63	-57	dBm
	Frequency over 1000 MHz	-	-57	-47	dBm

*1 When normal bandwidth mode (NBO_SEL = 0) is set. See the [DATA_SET] register (B0 0x47).

*2 863.5 MHz to 866.2 MHz cannot be used. For details, refer to "Setting channel frequency" in ML7396B Data Sheet.

5-5. UART Interface Characteristics

Item	Symbol	Conditions	Min	Typ	Max	Unit
Baud Rate	F_{BAUD}	Load capacitance CL=20pF	-	115200	-	bps

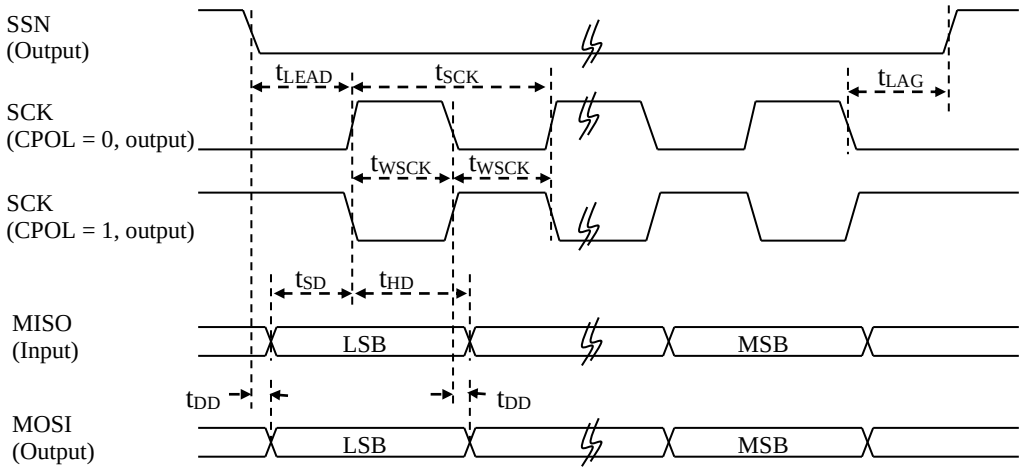


5-6. SPI Interface Characteristics

5-6-1.Master

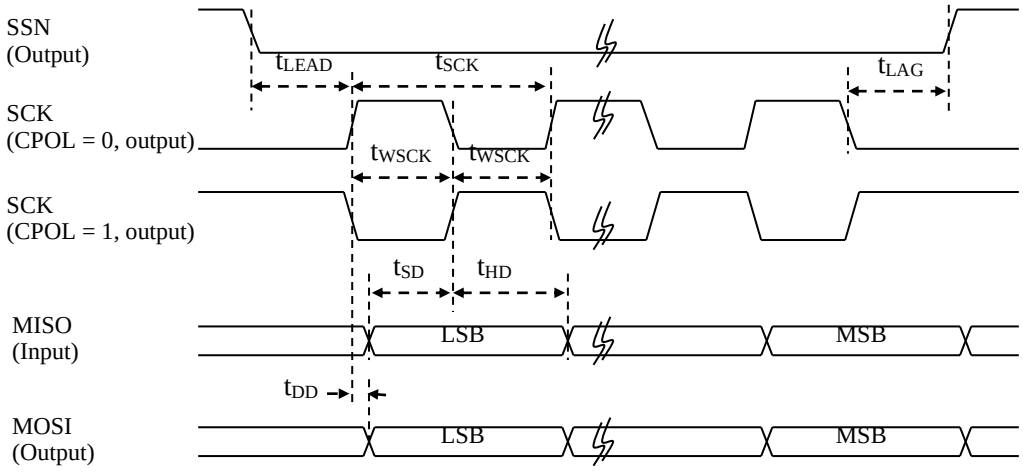
Item	Symbol	Conditions	Min	Typ	Max	Unit
Serial clock cycle time	t_{SCK}	-	100	-	-	ns
Serial clock High/Low time	t_{WSCK}	-	48	-	-	ns
Data delay time (output)	t_{DD}	-	-	-	50	ns
Data setup time (input)	t_{SD}	CL=20pF	-	-	48	ns
Data hold time (input)	t_{HD}	-	0	-	-	ns
SSN-SCK lead time	t_{LEAD}	-	0.5* t_{SCK}	-	1.6* t_{SCK}	ns
SCK-SSN lag time	t_{LAG}	-	0.5* t_{SCK}	-	1.6* t_{SCK}	ns
SSN H minimum guaranteed time	t_{WSSH}	-	1* t_{SCK}	-	511* t_{SCK}	ns

◎ SPI master mode timing (CPHA = 0)



CPHA=0

◎ SPI master mode timing (CPHA = 1)



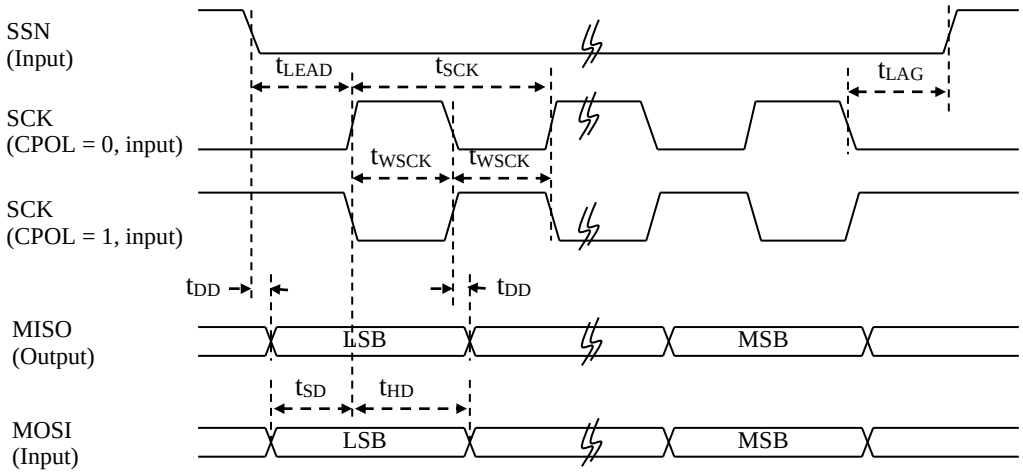
CPHA=1

*Note: For CPHA and CPOL, refer to SPI register.

5-6-2.Slave

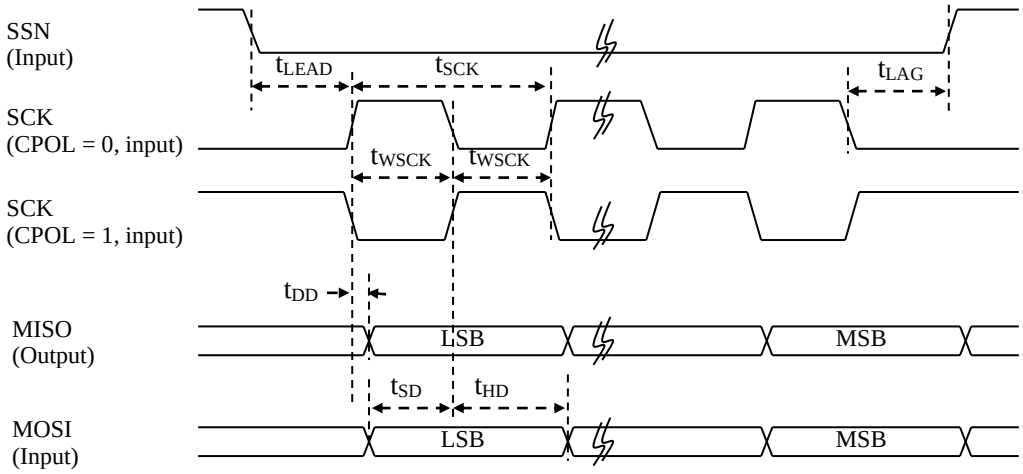
Item	Symbol	Conditions	Min	Typ	Max	Unit
Serial clock cycle time	t_{SCK}	-	100	-	-	ns
Serial clock High/Low time	t_{WSCK}	-	50	-	-	ns
Data delay time (output)	t_{DD}	-	-	-	50	ns
Data setup time (input)	t_{SD}	CL=20pF	-	-	50	ns
Data hold time (input)	t_{HD}	-	10	-	-	ns
SSN-SCK lead time	t_{LEAD}	-	0.5* t_{SCK}	-	-	ns
SCK-SSN lag time	t_{LAG}	-	0.5* t_{SCK}	-	-	ns

◎ SPI slave mode timing (CPHA = 0)



CPHA=0

◎ SPI slave mode timing (CPHA = 1)



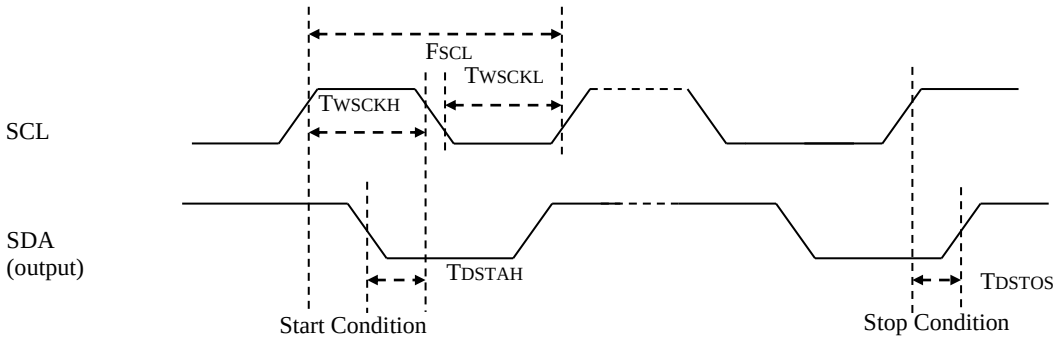
CPHA=1

*Note: For CPHA and CPOL, refer to SPI register.

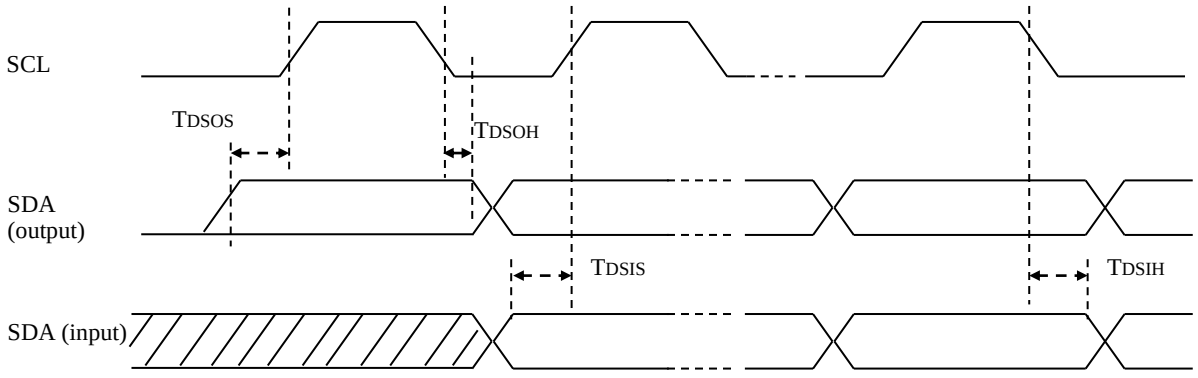
5-7. I2C Interface Characteristics

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCL clock frequency	Fscl	Conditions Load capacitance CL= 20pF	-	-	400	kHz
SCL H pulse width	TWSCKH		600	-	-	ns
SCL L pulse width	TWSCKL		1300	-	-	ns
Start condition hold time	TDSTAH		450	-	-	ns
Stop condition setup time	TDSTOS		575	-	-	ns
SDA output hold time	TDSOH		0	-	-	ns
SCL output delay time	TDSOS		600	-	-	ns
SDA input setup time	TDSIS		100	-	-	ns
SDA input hold time	TDSIH		0	-	-	ns

◎ Stop condition (SDA fall at SCL = 1), Start condition (SDA rise at SCL = 1)



◎ Transmission/reception



[Notices]

The SCL H pulse width (TWSCKH) should be set to 750 ns or longer when Fast (400 kHz) mode is used.

Otherwise, the start condition hold time (TDSTAH = 600 ns) and stop condition setup time (TDSTOS = 600 ns) in the I2C standard would not be met.

5-8. A/D Conversion Characteristics

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Number of bits	NSAR	Number of SAR register bits	-	10	-	bits
Resolution	RES	VIN=0 to VDDIO	1.8	-	3.6	mV/LSB
Input voltage range	VIN		0	-	VDDIO	V
Zero-scale error	EZS	•10-bit accuracy •Input signal source impedance ≤ 1 KΩ	-2.0	-	2.0	LSB
Full-scale error	EFS		-2.0	-	2.0	LSB
Differential non-linearity	DNL		-2.0	-	2.0	LSB
Integral non-linearity	INL		-2.0	-	2.0	LSB
Conversion time	TL		10	-	20	us

[Note] The ADC output is based on power/GND (tracked).
Connect a sufficient bypass capacitor between each power and GND to suppress the power fluctuation.

5-9. Temperature Sensor Characteristics

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Accuracy	E_{TS1}	VDDIO=3.3 V IDLE state 0 °C or more and 85 °C or less	-	±5	-	°C
	E_{TS2}	VDDIO=3.3 V IDLE state -40 °C or more and less than 0 °C	-	±10	-	°C
Conversion time	T_L		10	-	20	us

5-10. Low Voltage Detection Characteristics (LVD)

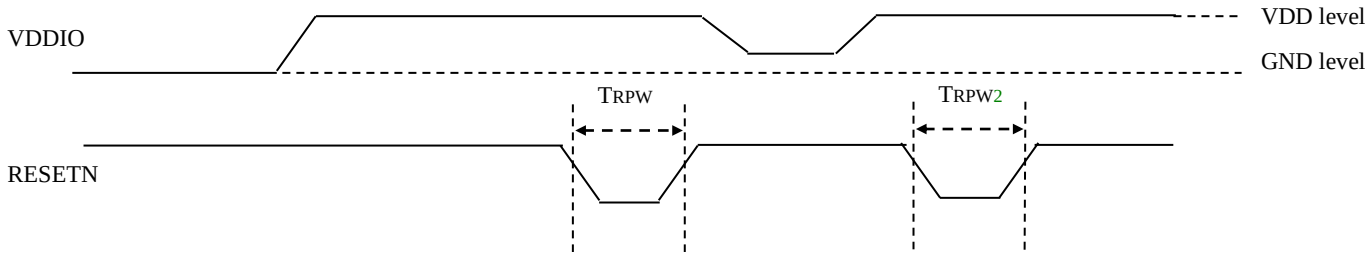
Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Detection error		VDDIO=1.8 to 3.0 V	-6.0	—	+6.0	%
Minimum response pulse width			200	—	—	us

5-11. Low Voltage Detection Characteristics (LLD)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Detecting voltage			—	1.8	—	V
Detection error			-0.2	—	+0.2	V
Minimum response pulse width			200	—	—	us

5-12. Reset Characteristics

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
RESETN activation time (pulse width) (When starting from VDDIO=0V)	T_{RPW}	-	200	-	-	ns
RESETN pulse time 2 (*1) (When starting from VDDIO□0 V)	T_{RPW2}	VDD>1.8 V	500	-	-	us

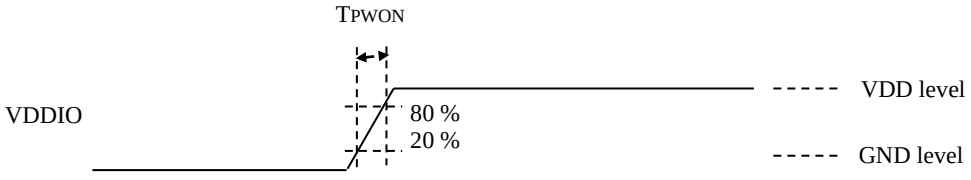


(*1) When starting from VDDIO≠0V, input a pulse to the RESETN signal after VDDIO exceeds 1.8 V.

(*2) This is reset by the power-on reset circuit built in the LSI at power-on.

5-13. Power-On Characteristics

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Power-on time difference	T_{PWON}	At power on (All power supply pins)	-	-	5	ms



5-14. Flash ROM Characteristics

Item	Symbol	Conditions	Spec	Unit
Erase cycles endurance (Data)	C_{EP1}	8KB	10000	cycles
Erase cycles endurance (Program)	C_{EP2}	504KB	100	

■ 6. Functional Description

*** For the RF block, refer to ML7396B Data Sheet.**

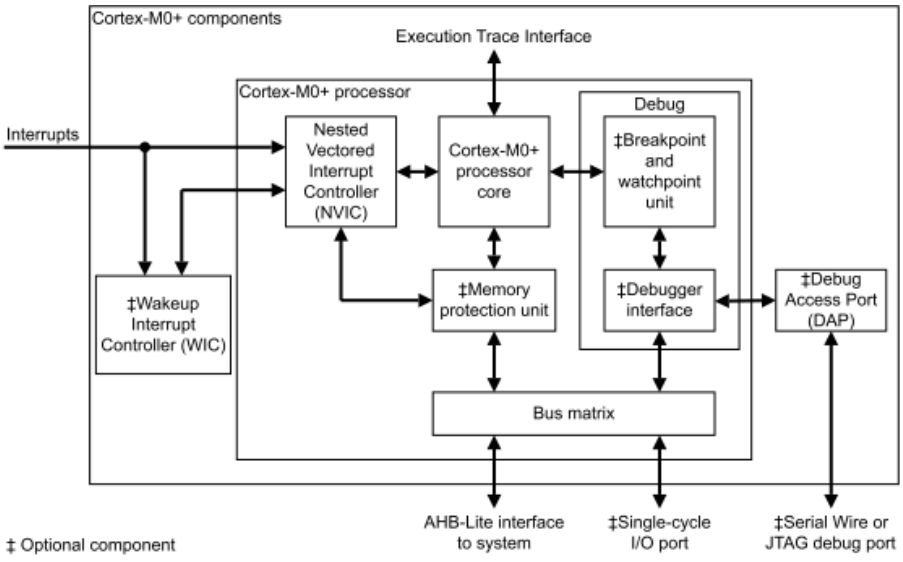
6-1.CPU (Cortex®-M0+)

A RISC processor manufactured by ARM®.

It is a 32-bit processor for small size and low power consumption applications and has a 2-stage pipeline configuration. It implements the ARMv6-M architecture, and operates with 16-bit Thumb® instructions and Thumb®-2 instructions.

The configuration is as follows:

- Little-Endian
- Number of break points: 4
- Number of watch points: 2
- SysTick timer, a 24-bit system timer, is included
- NVIC (Nested Vectored Interrupt Controller) is included
- Multiplier: High-speed (1-cycle) hardware multiplier is provided
- SLEEP/DEEPSLEEP supported
- WFI (Wait for Interrupts)/WFE (Wait for Events) supported
- Relocatable vector table
- MTB-M0+ supported

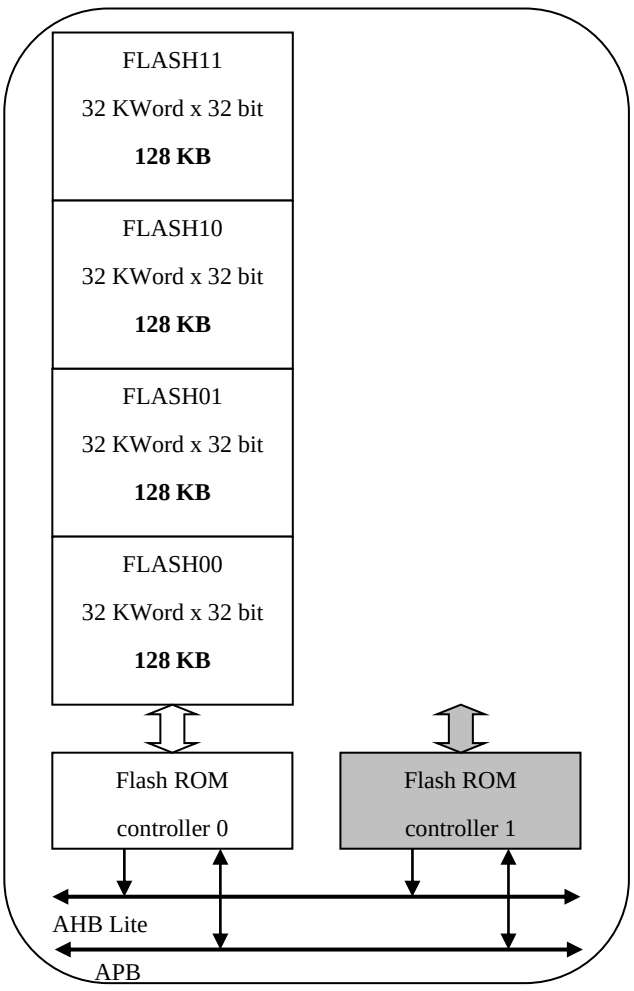


Cortex®-M0+ Schematic Diagram

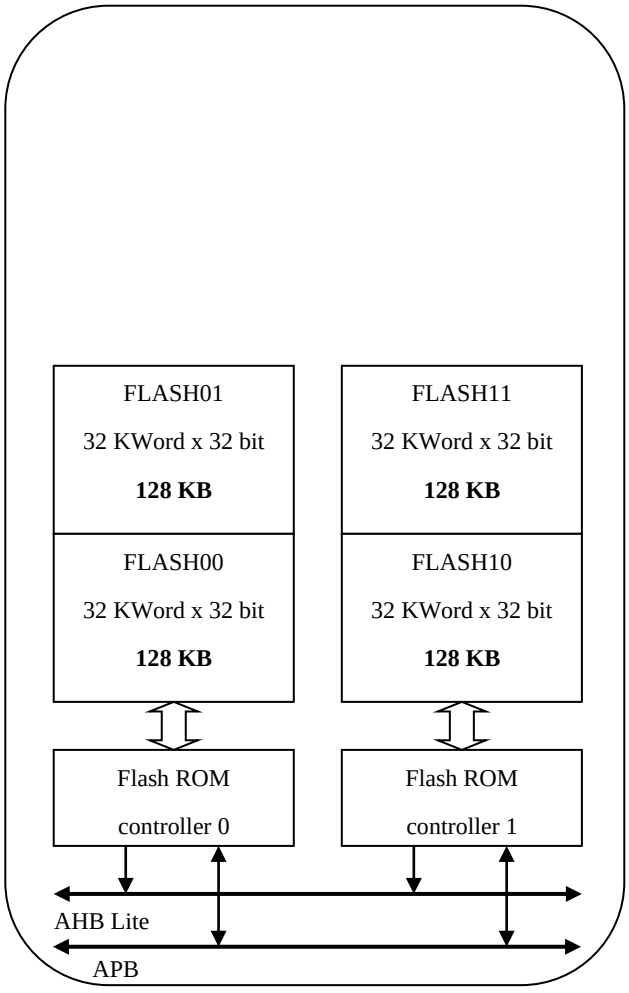
6-2.Flash ROM Controller

Flash ROM controllers are included. The boot from the Flash ROM is possible.

- For the Flash ROM area of 512 KB, the 256 KB x 2 bank configuration or the 512 KB x 1 bank configuration can be selected.
- For each bank, a Flash ROM controller is connected.
- Rewriting from the debugger through the debug port (SWD) is possible.
  In the 2-bank configuration, the program of one bank can be rewritten dynamically while the other CPU is running on another bank.
  In the 1-bank configuration, a rewriting program can be started from the ISP area or SRAM area.
- It is possible to perform the 1-word rewrite, sector erase (in 512 Bytes), block erase (in 4 Kbytes), or chip erase.
- The bank can be assigned to the remapping area (4 Kbytes) starting at 0x00000000 by the remapping function.
- The ROM area is divided into the trimming area to store the trimming value of analog macro, boot program area to store the starting program such as ISP, security area to store the security setting, and user area, and the trimming area, boot program area, and security area are protected.



1-bank configuration



2-bank configuration

* At power-on, the system always starts in 1 bank mode.

* In 2 bank mode, only the bank 0 side can be rewritten from the debugger.

* Only when both of the two Flash ROM controllers are not executing the processing, the transition to Sleep or DeepSleep is possible.

6-3.Interrupt Controller

The NVIC (Nested Vector Interrupt Controller) is included in Cortex®-M0+. The following interrupt sources are provided:

- 30 IRQs

6-4.SRAM IF

The SRAM IF of 64 KB is provided.

6-5.AHB-lite Bus

An AMBA 3 AHB-lite bus is included, to which an AHB slave can be connected.

6-6.APB Bus

An AMBA 2.0 APB bus is included, to which an APB slave can be connected.

6-7.Single-cycle I/O Port

A single-cycle I/O port which can be accessed in one cycle is included.

6-8.Debug Port

The SWD (2-wire serial wire debug port) is included as an interface for communicating with the debugger.

The internal resources can be accessed through this interface.

It also can be used for writing to on-chip memory such as a Flash ROM from the debugger.

6-9.MTB

The MTB (Micro Trace Buffer) which enables the execution trace of Cortex®-M0+ is included. This MTB shares the data RAM as the memory for trace.

6-10. Clock

This section describes the following three clocks:

- System clock system
- Debug port clock system
- Peripheral clock system

6.10.1 System Clock System

Clocks supplied to the CPU core and bus.

The maximum frequency is 40 MHz.

There are three derived clocks that are gated depending on the processor operation state.

Clock	Name	Description
CLK	Source Clock	The source clock. This clock can operate at up to 40 MHz with the internal PLL (multiplying/dividing of N/M ratio) or up to 34 MHz at the internal RC OSC.
FCLK	Free Running Clock	This clock can be stopped during the DEEPSLEEP state. It is supplied during the SLEEP state. This is used by the Wakeup Interrupt Controller and Power Management Unit in the Cortex®-M0+.
SCLK	Processor System Clock	This clock is stopped during the DEEPSLEEP state. It is supplied during the SLEEP state. This clock (or its derived clock) should be connected to any device that needs to operate during the SLEEP state.
HCLK	AHB Clock	This clock is stopped during the DEEPSLEEP or SLEEP state. This clock (or its derived clock) should be connected to any device that does not operate during the SLEEP state.
DCLK	Debug domain clock	This clock is always supplied when the Debugger is connected.

CPU Clock System

6.10.2 Debug Port Clock System

This clock system is supplied to JTAG I/F.

It is supplied from the debugger outside the chip.

The maximum frequency of the debug port clock system is 10 MHz.

6.10.3 Peripheral Clock System

This clock system is supplied to peripherals.

6.10.3.1 UART Reference Clock

This is the reference clock for baud rate generation. To reduce the baud rate error, this clock frequency needs to be adjusted. Example) When the reference clock frequency = 40.57 MHz (32.768 kHz XTAL OSC multiplied by 1238 by PLL), the baud rate error is about 0.1 % at 115.2 kbps.

6.10.3.2 SPI Reference Clock

This clock is the source for generating the SPI(Master/Slave) serial clock. The frequency of the clock for serial communication is lower than 1/2 of the frequency of this reference clock.

6.10.3.3 SSIS Sampling Clock

This clock is used for the data transmission/reception in SSIS (SSI Slave). The frequency of the clock for serial communication is lower than 1/10 of the frequency of this reference clock.

6.10.3.4 WDT Clock

This clock drives the watchdog timer.

It can measure the time of about hundreds of μsec to dozens of msec.

32.768 kHz input, low-speed CR input or source clock divide can be selected.

6.10.3.5 GPIO Debounce Clock

The debounce circuit that performs sampling at a fixed interval is implemented to eliminate the effect of noise and chattering when using the input signal from GPIO as an external interrupt source. This clock is used by the debounce circuit.

6.10.3.6 Timer Clock

This clock is used by the standard 32-bit timer.

XTAL 32.768 kHz input, low-speed CR input or source clock divide can be selected.

6.10.3.7 RTC Clock

This clock is used by the real time clock (RTC).

XTAL 32.768 kHz input and low-speed CR input, or their dividing can be selected.

6.10.3.8 Flexible Timer Clock

This clock is used by the flexible timer (FTM).

XTAL 32.768 kHz input or low-speed CR input can be selected.

6.10.3.9 I2C Reference Clock

The reference clock for generating a serial clock for I2C communication. The I2C clock depends on the serial communication mode (Standard mode, Fast mode) that it supports.

The frequency of I2C reference clock must be higher than that of system clock.

Minimum Frequency of I2C Reference Clock

Mode	Minimum frequency	I2C bus ratio
Standard mode	2.7 MHz	100 kbps
Fast mode	12 MHz	400 kbps

6.10.3.10 Flash ROM Controller Clock

This clock is used by the flash ROM controller.

6.10.3.11 DIO Clock

This clock is used by DIO.

6.10.3.12 RAND_GEN Clock

This clock is used by RAND_GEN (pseudo-random number generation circuit).

6.10.3.13 CLK_Timer Reference Clock

This clock counts the low-speed clock with CLK_Timer.

High-speed CR clock output, RF clock output or PLL clock output can be selected.

6.10.3.14 SysTick Timer Clock

This clock drives the SysTick timer in Cortex®-M0+.

It is usually 1 MHz. It is 32 kHz at the CPU low speed (32 kHz) mode.

6.10.3.15 AES Clock

This clock is used by AES.

6.10.3.16 ADC Clock

This clock is used by the ADC controller. It is the ADC sampling frequency (Max = 2.5 MHz) or less.

6.10.3.17 TEMP Clock

This clock is used by the temperature sensor controller. It is the ADC sampling frequency (Max = 2.5 MHz) or less.

6.10.3.18 LVD Clock

This clock is used by the low voltage detection controller.

6.10.3.19 DMAC Clock

This clock is used by the DMA controller.

6.10.3.20 Flash DMA Clock

This clock is used by the Flash DMA controller.

6-11. Reset

The relationship between the reset system and the reset target is shown in the table "Reset Causes and Reset Targets" below.

The reset causes include hardware reset, SYSRESETREQ reset of Cortex®-M0+, peripheral reset, debugger reset without the debugger connected, WDT reset, voltage detection reset and reset at CPU LOCKUP.

The hardware reset occurs during the initial operation of hardware including reset by an external pin or reset at power-on triggered by power-on detection. When a hardware reset occurs, the reset circuit asserts the reset systems connected to all initializable circuits and initializes those circuits.

The SYSRESETREQ reset of Cortex®-M0+ occurs when the SYSRESETREQ bit of Application Interrupt and Reset Control Register (AIRCR) within Cortex®-M0+ is set from the CPU or debugger. The CPU, peripherals, bus, and memory IF are initialized, while the program fetches the reset exception vector.

The peripheral reset resets only the target peripheral by setting the bit assigned to each peripheral in the peripheral reset registers.

When the debugger is not connected, the debug circuit is always in reset state to prevent unnecessary switching or malfunctions caused by the debug circuit.

The WDT reset, voltage detection reset and the automatic reset at CPU LOCKUP initialize all initializable circuits except the CPU status register and the debug circuit.

Reset Causes and Reset Targets

Reset target	Reset causes					
	Hardware Reset Power-on reset	LVD reset	WDT reset/voltage detection reset/automatic reset at CPU LOCKUP	SYSRESETREQ bit set of Cortex®-M0+ AIRCR register	Peripheral Reset	When the debugger is not connected
CPU_ST register	○	-	-	-	-	-
Control registers with the Sticky attribute, excluding the CPU_ST register	○	○	○	-	-	-
System hardware such as clock control circuit and power management circuit	○	○	○	-	-	-
CPU AHB bus/AHB peripheral	○	○	○	○	-	-
APB bus/peripheral other than above	○	○	○	○	○	-
SingleCycleIO bus/peripheral	○	○	○	○	○	-
Flash ROM	○	○	○	-	-	-
RF chip	○	○	-	-	○	-
Debug circuit	○	-	-	-	-	○

6-12. Power Management

Low power consumption is realized by clock control and power control.

6.12.1 Operation Mode

The power management function of this LSI has the following features:

- The low power consumption states, SLEEP and DEEPSLEEP, are supported.
- The clocks are stopped depending on each low power consumption state.

The following basic power states are defined in this LSI:

Operation Mode	Operation state		Current consumption	Return to Active mode	
	Power mode of Cortex®-M0+	LSI	Current consumption	Method	Time
Active	RUN	All clocks are supplied. However, the clock delivery to peripherals (including RF) can be set to on/off by the register.	11 mA	-	-
SLEEP	Sleep	AHB bus clock (FlashROM/RAM) is stopped. The clock delivery to peripherals (including RF) can be set to on/off by the register.	5 mA	Interrupt from a peripheral. Start the debugger	75 nsec
DEEPSLEEP	DeepSleep	The main clock is stopped. The clock delivery to sub clock system peripherals (RTC, TIMER, etc.) can be set to on/off by the register. The power supply to FlashROM/SRAM/non-sub clock system peripherals/RF can be set to on/off with the register setting.	2 uA (All internal power supply are OFF)	Interrupt from a sub clock system peripheral. Start the debugger	150 usec

6.12.2 Power Separation

This LSI can operate at low power consumption by turning off some power supplies in the LSI during DEEPSLEEP state.

The following function blocks can be set to power on/off.

Function block	Target range	Setting register	Remarks
SRAM	Unit of 32 KB (up to 64 KB)	Deep sleep control register (0x40050040) bits 8-9	
FLASH	Whole 512 KB area	Deep sleep control register (0x40050040) bit 10	
Logic	CPU, FlashCnt, STD GPIO, AES, DMAC, FlashDMA, UART, SPI, I2C, RAND, SSIS, FTM, CLK Timer, DIO, ADC, TEMP	Deep sleep control register (0x40050040) bit 11	
RF	Whole RF chip area	Deep sleep control register (0x40050040) bit 12	

6-13. System Control

System Control is a block which controls the whole system (including the control of clocks, reset, remapping, interrupt, and SysTick timer) and displays the ID information specific to each chip and the CPU status information.

6.13.1 Clock Control

- Selects the clock source (high-speed CR, RF clock, XTAL32kHz, low-speed CR, or PLL).
- Stops/resumes the clock of each peripheral.
- Sets the operation (automatically stops the clock) when the CPU goes to the low power consumption state (SLEEP or DEEPSLEEP) for each peripheral.
- Changes the frequency of system clock or clock supplied to each peripheral.

6.13.2 Reset Control

- Able to reset individual peripherals.
- Sets the operation when the CPU goes to the LOCKUP state.
- Indicates that the CPU is initialized by the reset due to LOCKUP or the reset caused by low voltage detection or watchdog timer.

6.13.3 Information Display

Displays the ID information specific to each chip to distinguish individual chips.

6.13.4 Remapping Control

Selects a device to be placed in the remapping area on the address space.

6.13.5 Cortex®-M0+ Control

Changes the settings of Nested Vectored Interrupt Controller (NVIC) and SysTick timer mounted in Cortex®-M0+.

6.13.6 Boot Program Area

The boot program is written to the boot program area.

For details, see "ML7416 Boot Program Functional Descriptions".

6-14. Peripheral

This LSI implements the following peripherals: UART, SPI, SSIS, WDT, GPIO (APB GPIO), SingleCycleIO (STD GPIO), timer, RTC, flexible timer (FTM), I2C, Flash controller, DIO, RAND_GEN, CLK_Timer, AES, ADC, thermometer (TEMP), low voltage detection (LVD), DMAC, and Flash DMA.

6.14.1 UART

A start-stop synchronous serial communication interface which has functions equivalent to the industry standard 16550. The features are shown below.

- Includes a 16-byte FIFO for each of transmission and reception.
- Full-duplex communication is possible.
- Includes a programmable baud rate generator. Note that the baud rate is the same for transmission and reception.
- The character size of 5- to 8-bit is supported.
- 1 or 2 (1.5 for the 5-bit character size) stop bit can be selected.
- For parity generation/check, supports even/odd/none/stick.
- Supports the auto-flow control function.

6.14.2 SPI

A synchronous serial communication interface (master/slave). The features are shown below.

- Performs the full-duplex data transfer.
- Master or Slave mode can be selected.
- Includes a 16-byte or 16-word (16-bit) FIFO for each of the transmission and reception sides.
- For the transfer size, 8 bits (bytes) or 16 bits (words) can be selected.
- The interrupt caused by the number of received bytes (words) and the number of untransmitted bytes (words) can be set in the range 1 to 16.
- Either LSB first or MSB first can be selected.
- The polarity and phase of the serial clock can be selected.
- Able to control the interval before/after transfer in Master mode.
- Uses the status bit to indicate the completion of transmission/reception and the FIFO status.
- Able to detect a mode fault error to avoid multi-master bus contention.
- Able to detect a write overflow error if any further writing is attempted when the transmit FIFO is in the full state.
- Generates an interrupt when the transmit/receive FIFO is in a specific state or when a cause such as mode fault error occurs.

6.14.3 SSIS(SSI Slave)

A synchronous serial communication interface (for slave only). The features are shown below.

- Supports the Motorola SPI.
- Data length: 4 to 16 bits
- 8-stage FIFO for each of transmit and receive

6.14.4 WDT

Programmable 16-bit watchdog timer. The features are shown below.

- When the counter reaches its timeout, this timer asserts an interrupt at the first timeout, and performs the system reset operation at the second timeout. The mode of asserting only an interrupt without reset operation can be set.
- If the CPU is stopped by the SLEEP mode or debugger, the counter operation of watchdog timer is stopped.
- The source clock of the watchdog timer can be selected using the control register.

6.14.5 GPIO(APB-GPIO)

General-purpose port with interrupt function. The features are shown below.

- 13-bit (GPIOA) general-purpose port.
- Can be used as external interrupt input.* Either level/edge or Low/High can be selected.
- Includes the debounce circuit in the input side, which performs sampling at a fixed interval to eliminate the effect of noise and chattering.
- Can be used as return cause from SLEEP/DEEPSLEEP.

6.14.6 Standard GPIO (Single-cycle I/O)

General-purpose port connected to a single-cycle I/O which can be accessed in one cycle. The features are shown below.

- 4-bit (GPIOA) general-purpose port (assigned to the quintic function).

6.14.7 Timer

32-bit x 10ch general-purpose timer. The features are shown below.

- If the CPU is stopped by the debugger, the counter operation of timer is stopped.
- Can be used as 64-bit timer by the cascade connection (TimerB, TimerC, TimerD, and TimerE only).

6.14.8 RTC

Real time clock with perpetual calendar which can be read/written from a second unit. The features are shown below.

- Operates at 32.768 kHz input from the external or at internal low-speed RC.
- It is possible to set, correct, and read the time.
- Can be used as interrupt source.
- The time-designated interrupt can be generated.

6.14.9 Flexible Timer (FTM)

16-bit multifunction timer. The features (operation mode) are shown below.

- Auto-reload timer (ART)
- Compare out (CMO)
- Pulse width modulation (PWM)
- Capture (CAP)

6.14.10 I2C

2-wire (SCL, SDA) serial interface. The features are shown below.

- Supports the standard mode (up to 100 kbps) and the fast mode (up to 400 kbps).
- Supports the 7- or 10-bit addressing.
- Supports the 7- or 10-bit composite format transfer.
- Supports the bulk transfer mode.

6.14.11 Flash ROM Controller

Memory controller that controls the internal Flash ROM.
It operates as AHB slave when reading the Flash ROM.
Erase/write/register access of the Flash ROM operates as an APB slave.

6.14.12 DIO

Data transmission/reception interface dedicated to the RF block (ML7396B) control.

6.14.13 RAND_GEN

Pseudo-random number generation circuit. The features are shown below.

- The RAND length can be selected from RAND9, RAND15, and RAND23.
- Random number generation result can be output with the logical complement of 2.

6.14.14 CLK_Timer

This function uses the high-speed clock to count a certain time period of the low-speed clock and stores the count result in a register.

6.14.15 AES

This function performs the encryption and decryption of transmit/receive data by using Advanced Encryption Standard (AES).

The features are shown below.

- Encryption and decryption of 128-bit data (ECB, CBC, CTR, CCM, GCM, CFB, and OFB supported)
- Generation and decryption of authentication TAG (CCM128/64/32/16/8)

6.14.16 ADC

This function controls the 10-bit successive approximation type A/D converter.

The features are shown below.

- Programmable scan of up to three channels (CH0 to CH2) (The scan time and scan order can be set.)
  * Two channels when using the temperature sensor (CH2).
- Scan result notification (The scan completion is notified by an interrupt.)
- Averaging of A/D conversion data (The average value of A/D conversion results is displayed.)
- Calculation of CH0 to CH2 input voltage (It is assumed that the reference voltage output from the regulator at CH3 is monitored.)

6.14.17 Thermometer (TEMP)

1-channel temperature sensor to measure the temperature in the chip.

The features are shown below.

- Accuracy: ± 5 °C
- Converts the temperature to voltage and digitizes the converted voltage by using the 10-bit A/D converter.

6.14.18 Low Voltage Detection (LVD)

Low voltage detection function.

The features are shown below.

- The voltage detection level can be set.
- After detection, interrupt notification or reset can be selected.
- Starts the reference voltage (VBG) periodically by using the dedicated low speed timer to compare and determine the voltage detection level.

6.14.19 DMAC

Direct memory access controller. Among peripherals, SPI2 and AES support DMA transfer.

The features are shown below.

- Four-channel DMA controller.
- Each channel includes a 16-stage FIFO (8-stage for channels 0 and 1) for source transfer and destination transfer.
- Supports the peripheral-to-memory transfer.
- Includes the hardware interface to handshake with SPI and AES.
- Supports up to 2048-byte block transfer.
- Supports the channel priority setting.
- Has one AHB master port.
- Supports increment/decrement of the transfer address and transfer to a single address.
- Supports multiple block transfer using a linked list.

6.14.20 Flash DMA

Controls the data write to the flash area and the data compare of the flash area at high speed, instead of CPU.

The features are shown below.

- Batch writes the data in the RAM area to the flash area.
- Compares the RAM area data and the Flash area data and notifies the result.

6.14.21 Other

Mode control (MODE_CNT): A set of registers for clock dividing setting, power separation control, and adjustment of analog circuits (regulator, RCOSC, RF, etc.).

■ 7. Programming Model

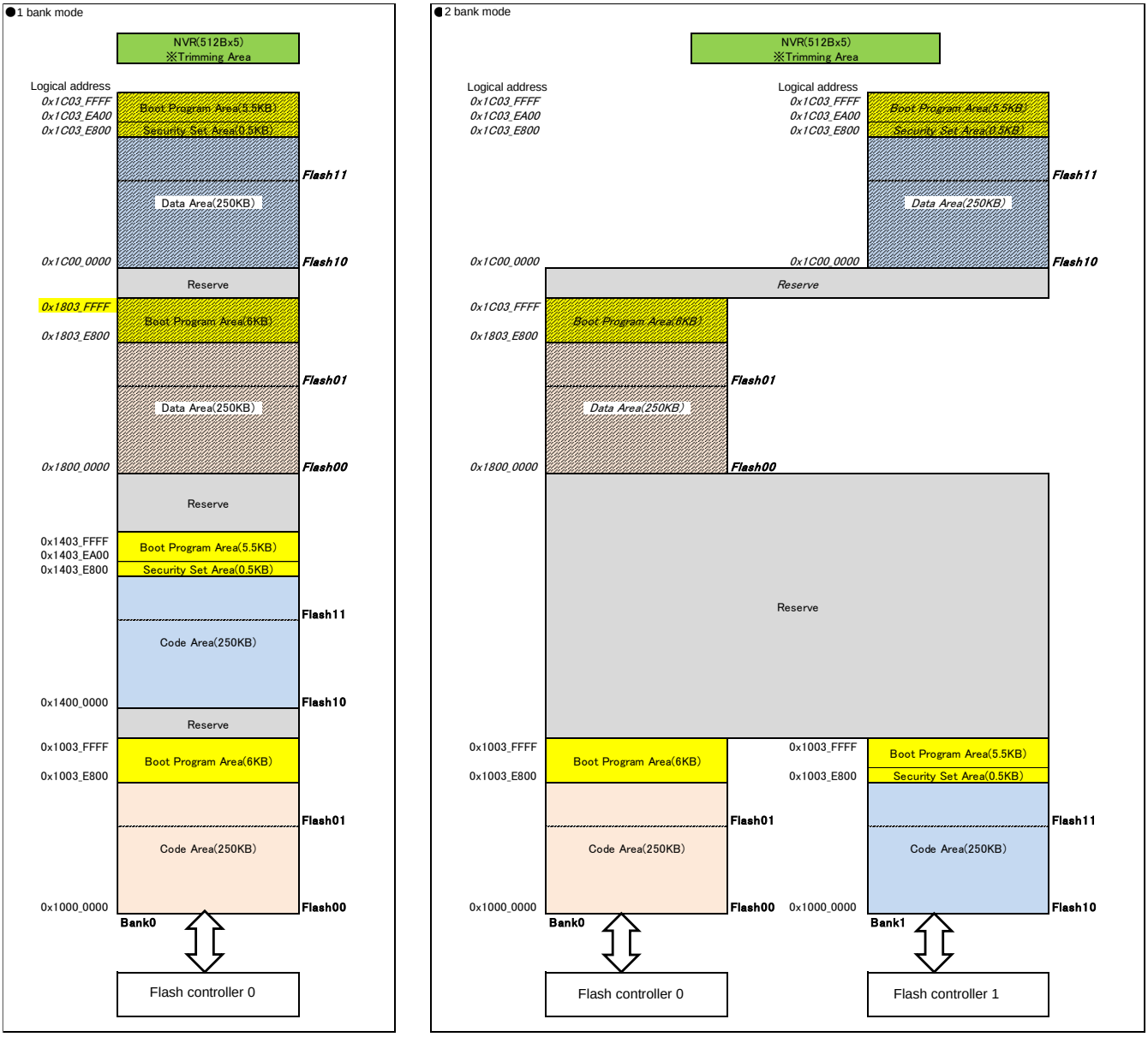
7-1.Address Map

Table 7-1 Address Map and Table 7-2 Address Map (Details of AHB/APB/IO) show address maps.

Table 7-1 Address Map

Address range	Response device		Description
	Normal	REMAP	
0xF000_3000-0xFFFF_FFFF	Reserved		Reserved area.
0xF000_2000-0xF000_2FFF	MTB		Area assigned to the Micro Trace Buffer (SFR). This area is responded by the default slave when the security function of the flash ROM controller is enabled.
0xF000_1000-0xF000_1FFF	Reserved		Reserved area.
0xF000_0000-0xF000_0FFF	ROM Table		Area assigned to the system ROM table. This area is responded by the default slave when the security function of the flash ROM controller is enabled.
0x6000_0000-0xEFFF_FFFF	Reserved		Reserved area.
0x5C00_0404-0x5FFF_FFFF	Reserved		Reserved area.
0x5C00_0000-0x5C00_0403	STD GPIO		Area assigned to the Single-cycle I/O peripheral.
0x5800_0000-0x5BFF_FFFF	Reserved		Reserved area.
0x5000_0000-0x57FF_FFFF	Reserved		Reserved area.
0x4000_0000-0x4FFF_FFFF	AHB/APB		Area assigned to the AHB/APB device. For details, please refer to "Table 7-2 Address Map (Details of AHB/APB/IO)".
0x2001_0000-0x3FFF_FFFF	Reserved		Reserved area.
0x2000_0000-0x2000_FFFF	SRAM		SRAM area (64 KB).
0x1C04_0000-0x1FFF_FFFF	Reserved		Reserved area.

Address range	Response device		Description
	Normal	REMAP	
0x1C03_EA00-0x1C03_FFFF	Flash ROM		Boot area of FLASH11 (5.5 KB).* Only for data reference
0x1C03_E800-0x1C03_E9FF	Flash ROM		Security area of FLASH11 (0.5 KB).* Only for data reference
0x1C02_0000-0x1C03_E7FF	Flash ROM		User area of FLASH11 (122 KB).* Only for data reference
0x1C00_0000-0x1C01_FFFF	Flash ROM		User area of FLASH10 (128 KB).* Only for data reference
0x1804_0000-0x1BFF_FFFF	Reserved		Reserved area.
0x1803_E800-0x1803_FFFF	Flash ROM		Boot area of FLASH01 (6 KB).* Only for data reference
0x1802_0000-0x1803_E7FF	Flash ROM		User area of FLASH01 (122 KB).* Only for data reference
0x1800_0000-0x1801_FFFF	Flash ROM		User area of FLASH00 (128 KB).* Only for data reference
0x1404_0000-0x17FF_FFFF	Reserved		Reserved area.
0x1403_EA00-0x1403_FFFF	Flash ROM		Boot area of FLASH11 (5.5 KB).
0x1403_E800-0x1403_E9FF	Flash ROM		Security area of FLASH11 (0.5 KB).
0x1402_0000-0x1403_E7FF	Flash ROM		User area of FLASH11 (122 KB).
0x1400_0000-0x1401_FFFF	Flash ROM		User area of FLASH10 (128 KB).
0x1004_0000-0x13FF_FFFF	Reserved		Reserved area.
0x1003_E800-0x1003_FFFF	Flash ROM		Boot area of FLASH01 (6 KB).
0x1002_0000-0x1003_E7FF	Flash ROM		User area of FLASH01 (122 KB).
0x1000_0000-0x1001_FFFF	Flash ROM		User area of FLASH00 (128 KB).
0x0008_0000-0x0FFF_FFFF	Reserved		Reserved area.
0x0000_0000-0x0007_FFFF	Flash ROM	SRAM	Remap area. Flash ROM is assigned as a mirror by default. When remapping, Flash ROM and SRAM areas can be assigned by using a value of MODE0 pin or control register.


1 bank mode
NVR(512Bx5)
※Trimming Area
Logical address
0x1C03_FFFF
0x1C03_EA00
0x1C03_E800
Boot Program Area(5.5KB)
Security Set Area(0.5KB)
Data Area(250KB)
Flash11
0x1C00_0000
Flash10
Reserve
0x1803_FFFF
Boot Program Area(6KB)
0x1803_E800
Flash01
Data Area(250KB)
0x1800_0000
Flash00
Reserve
0x1403_FFFF
0x1403_EA00
0x1403_E800
Boot Program Area(5.5KB)
Security Set Area(0.5KB)
Flash11
Code Area(250KB)
0x1400_0000
Flash10
Reserve
0x1003_FFFF
Boot Program Area(6KB)
0x1003_E800
Flash01
Code Area(250KB)
0x1000_0000
Flash00
Bank0
Flash controller 0
2 bank mode
NVR(512Bx5)
※Trimming Area
Logical address
0x1C03_FFFF
0x1C03_EA00
0x1C03_E800
Boot Program Area(5.5KB)
Security Set Area(0.5KB)
Data Area(250KB)
Flash11
0x1C00_0000
Flash10
Reserve
0x1C03_FFFF
Boot Program Area(6KB)
0x1803_E800
Flash01
Data Area(250KB)
0x1800_0000
Flash00
Reserve
0x1003_FFFF
Boot Program Area(6KB)
0x1003_E800
Flash01
Code Area(250KB)
0x1000_0000
Flash00
Bank0
Flash controller 0
0x1003_FFFF
Boot Program Area(5.5KB)
0x1003_E800
Security Set Area(0.5KB)
Flash11
Code Area(250KB)
0x1000_0000
Flash10
Bank1
Flash controller 1

- The Flash ROM area consists of total 512 KB (128 KB x 4), including FLASH00/FLASH01/FLASH10/FLASH11.
- In 2 bank mode, it consists of FLASH00/FLASH01 (128 KB x 2) and FLASH10/FLASH11 (128 KB x 2).
- At power-on, the system always starts in 1 bank mode.
- The selection between 1 bank mode and 2 bank mode is made by the remapping control register SYSCON_REMAP_CON.
- The Boot Program Area is 11.5 KB in total, consisting of 0x1003_E800 to 0x1003_FFFF (6 KB) and 0x1403_EA00 to 0x1403_FFFF (5.5 KB).
- In the boot program start mode where the power is turned on by setting the mode0 pin to "H", 4 KB from 0x1003_E800 are mapped to 0x0000_0000.
- The Security Set Area is 512 bytes from 0x1403_E800 to 0x1403_E9FF.
- In 2 bank mode, the logical addresses of both the Flash controller 0 and Flash controller 1 start from 0x1000_0000.
- In 2 bank mode, the update program can be written on the other side of the overlapping address from 0x1000_0000 to 0x1003_E800 by specifying the Flash controller in the software.
- For the area from 0x1800_0000 to 0x1C03_0000, 0x1800_0000 to 0x1803_FFFF and 0x1C00_0000 to 0x1C03_FFFF can always be referred by Flash00/Flash01 and Flash10/Flash11 as mirror areas respectively, regardless of the bank mode and bank selection. However, these mirror areas can be used only for data reference, but not for program execution (program code cannot be fetched).

Table 7-2 Address Map (Details of AHB/APB/IO)

Address range	Response device	Description
0x5C000404-0x5FFFFFFF	Reserved	Reserved area.
0x5C000000-0x5C000403	STD GPIO	Single-cycle I/O area.
0x58000000-0x5BFFFFFF	Reserved	Reserved area.
0x4017009C-0x57FFFFFF	Reserved	Reserved area.
0x40170000-0x4017009B	AES	AES area.
0x40160020-0x4016FFFF	Reserved	Reserved area.
0x40160000-0x4016001F	Flash DMA	Flash DMA area.
0x40150400-0x4015FFFF	Reserved	Reserved area.
0x40150000-0x401503FF	DMAC	DMAC area.
0x40140820-0x4014FFFF	Reserved	Reserved area.
0x40140800-0x4014081F	Reserved	Reserved area.
0x40090030-0x401407FF	Reserved	Reserved area.
0x40090000-0x4009002F	LVD	Low voltage detection area.
0x40080040-0x4008FFFF	Reserved	Reserved area.
0x40080000-0x4008003F	Reserved	Reserved area.
0x4007002C-0x4007FFFF	Reserved	Reserved area.
0x40070000-0x4007002B	ADC	ADC area.
0x40060040-0x4006FFFF	Reserved	Reserved area.
0x40060000-0x4006003F	Reserved	Reserved area.
0x40050260-0x4005FFFF	Reserved	Reserved area.
0x40050000-0x4005025F	MODE _CNT	Mode control area.
0x40045014-0x40045FFF	Reserved	Reserved area.
0x40045000-0x40045013	CLK_Timer	CLK_Timer area.
0x4004400C-0x40044FFF	Reserved	Reserved area.
0x40044000-0x4004400B	RAND_GEN	RAND_GEN area.
0x4004303C-0x40043FFF	Reserved	Reserved area.
0x40043000-0x4004303B	DIO	DIO area.* For ML7396B control
0x40042114-0x40042FFF	Reserved	Reserved area.
0x40042000-0x40042113	Ext_Timer	TimerF area. * 6ch
0x40041CB0-0x40041FFF	Reserved	Reserved area.

0x40041C00-0x40041CAF	TimerE	TimerE area.* Making 64-bit can be enabled by TimerD and the cascade connection
0x400418B0-0x40041BFF	Reserved	Reserved area.
0x40041800-0x400418AF	TimerD	TimerD area.* Making 64-bit can be enabled by TimerE and the cascade connection
0x400414B0-0x400417FF	Reserved	Reserved area.
0x40041400-0x400414AF	TimerC	TimerC area.* Making 64-bit can be enabled by TimerB and the cascade connection
0x400410B0-0x400413FF	Reserved	Reserved area.
0x40041000-0x400410AF	TimerB	TimerB area.* Making 64-bit can be enabled by TimerC and the cascade connection
0x40040820-0x40040FFF	Reserved	Reserved area.
0x40040800-0x4004081F	SPI2	SPI2 area.* For ML7396B control (dedicated to master)
0x40040420-0x400407FF	Reserved	Reserved area.
0x40040400-0x4004041F	SPI1	SPI1 area.
0x40040020-0x400403FF	Reserved	Reserved area.
0x40040000-0x4004001F	SPI0	SPI0 area.
0x40018210-0x4003FFFF	Reserved	Reserved area.
0x40018000-0x4001820F	Port	Port configuration area.
0x40010100-0x40017FFF	Reserved	Reserved area.
0x40010000-0x400100FF	WDT	WDT area.
0x4000D0F4-0x4000FFFF	Reserved	Reserved area.
0x4000D000-0x4000D0F3	SSIS0	SSI (Slave) area.
0x4000A378-0x4000CFFF	Reserved	Reserved area.
0x4000A300-0x4000A377	GPIOD	GPIOD area.* For ML7396B control. The interrupt source is [29].
0x4000A278-0x4000A2FF	Reserved	Reserved area.
0x4000A200-0x4000A277	GPIOC	GPIOC area.
0x4000A178-0x4000A1FF	Reserved	Reserved area.
0x4000A100-0x4000A177	GPIOB	GPIOB area.
0x4000A078-0x4000A0FF	Reserved	Reserved area.
0x4000A000-0x4000A077	GPIOA	GPIOA area.
0x40008100-0x40009FFF	Reserved	Reserved area.
0x40008000-0x400080FF	I2C0	I2C area.
0x40004900-0x40007FFF	Reserved	Reserved area.
0x40004800-0x400048FF	UART2	UART2 area.
0x40004500-0x400047FF	Reserved	Reserved area.

0x40004400-0x400044FF	UART1	UART1 area.
0x40004100-0x400043FF	Reserved	Reserved area.
0x40004000-0x400040FF	UART0	UART0 area.* It is used in the ISP function
0x40003070-0x40003FFF	Reserved	Reserved area.
0x40003000-0x4000306F	RTC	RTC area.
0x40002208-0x40002FFF	Reserved	Reserved area.
0x40002000-0x40002207	Flexible Timer	Flexible timer area.
0x400010B0-0x40001FFF	Reserved	Reserved area.
0x40001000-0x400010AF	TimerA	TimerA area.
0x40000760-0x40000FFF	Reserved	Reserved area.
0x40000600-0x4000075F	Flash Control1	Flash ROM controller 1 area.
0x40000560-0x400005FF	Reserved	Reserved area.
0x40000400-0x4000055F	Flash Control0	Flash ROM controller 0 area.
0x40000200-0x400003FF	Reserved	Reserved area.
0x40000000-0x400001FF	System Control	System control area.

0x58000000-0x5FFFFFFF is responded by STD GPIO. For a reserved area, writing is ignored and 0 is read at reading.

0x40150000-0x57FFFFFF is responded by various peripherals in AHB. For a reserved area, an AHB error is returned.

0x40000000-0x4014FFFF is responded by various peripherals in APB. For a reserved area, writing is ignored and 0 is read at reading.

7-2.Remapping Control

A device to be assigned to the remapping area starting at the address 0x00000000 is selected by the external pin or control register. When power-on reset is generated, a memory device to be placed in the remapping area is determined according to the external pin state. After the power-on reset, a memory device to be placed in the remapping area is determined by the remapping control register.

For details of the remapping control register, refer to SYSCON_REMAP_CON (0x40000010).

Example of implementing the remapping control is shown below.

When REMAP_EN of the remapping control register = 0 (initial value)

When the external pin MODE0 = L, the program executes from the address 0 in the internal Flash ROM.

When the external pin MODE0 = H, the program executes from the boot area in the internal Flash ROM.

When REMAP_EN of the remapping control register = 1 (at software reset after rewriting the remapping control register by software)

When REMAP[3:0] = 0b0000, internal Flash ROM is started first.

When REMAP[3:0] = 0bxxx1, internal SRAM is started first.

When REMAP[3:0] = 0bxx10, reserved

When REMAP[3:0] = 0bx100, the address (boot program area) set at the remapping base address is started first.

When REMAP[3:0] = 0b1000, reserved

[About Vector Table Relocation]

Cortex®-M0+ makes the vector table relocatable by using the vector table offset register (VTOR). Please note that VTOR is reset by a software reset (AIRCR.SYSRESETREQ).

Also, note that the relocation of the vector table is performed immediately after rewriting VTOR. The access to the SCS area involves the DSB instruction in Cortex®-M0+. It is not necessary to insert the DSB instruction. For details, refer to the following:

DAI0321A_programming_guide_memory_barriers_for_m_profile.pdf

4.11 Vector table configuration - Vector Table Offset Register (VTOR)

7-3.Internal Flash ROM Space

Area to which the internal Flash ROM is assigned (512 KB). It is normally used as the program ROM space.

At reading, internal Flash ROM is read via the Flash ROM controller. Rewriting the Flash ROM (erasing and programming) is performed by the rewrite sequence via the Flash ROM control register.

For the assignment of Flash ROM space, refer to the address map.

7-4.Internal SRAM Space

Area where internal SRAM is assigned (64 KB). It is normally used as a data RAM space.

It can be assigned as a remapping area by the remapping control register. It can be used as a program storage area when rewriting the Flash ROM.

7-5.Interrupt Source

Assignment of interrupt sources for this LSI is shown in Table List of Interrupt Sources below.

List of Interrupt Sources

Interrupt number	Interrupt source
NMI	Reserved
IRQ[0]	WDT
IRQ[1]	Reserved
IRQ[2]	GPIOA
IRQ[3]	TimerA
IRQ[4]	GPIOB
IRQ[5]	RTC
IRQ[6]	TimerB
IRQ[7]	TimerC
IRQ[8]	Flexible Timer
IRQ[9]	GPIOC
IRQ[10]	UART0
IRQ[11]	SSIS
IRQ[12]	ADC
IRQ[13]	AES
IRQ[14]	UART1
IRQ[15]	UART2
IRQ[16]	TimerD
IRQ[17]	Flash Control0
IRQ[18]	TimerE
IRQ[19]	Ext_Timer
IRQ[20]	I2C
IRQ[21]	DMAC
IRQ[22]	SPI0
IRQ[23]	SPI1
IRQ[24]	Flash DMA
IRQ[25]	Flash Control1
IRQ[26]	SPI2

IRQ[27]	DIO
IRQ[28]	LVD
IRQ[29]	RF (GPIOD)
IRQ[30]	CLK_Timer
IRQ[31]	MODE_CNT

7-6.System Control

7.6.1 General Description

ID display, remapping control, CPU control, status display, IRQ/SysTick control, clock control, and power management control are performed.

7.6.2 List of Registers

Address	Name	Symbol	R/W	Initial value	Description
0x40000000	ID register 0	SYSCON_ID0	R	0x11800000	Indicates model information.
0x40000004	ID register 1	SYSCON_ID1	R	0x00000000	Indicates model information.
0x40000008	ID register 2	SYSCON_ID2	R	0x00000000	Indicates model information.
0x4000000C	ID register 3	SYSCON_ID3	R	0x00000000	Indicates model information.
0x40000010	Remapping control	SYSCON_REMAP_CON	R/W	0x00000000	Controls REMAP operation.
0x40000014	Remapping base address	SYSCON_REMAP_BASE	R/W	0x1003E800	Controls REMAP operation.
0x40000020	CPU control	SYSCON_CPU_CON	R	0x00000000	Sets the operation when the CPU is locked up.
0x40000024	CPU status	SYSCON_CPU_ST	R	0x00000000	Indicates CPU state.
0x40000030	IRQ control	SYSCON_IRQ_CON	R/W	0x0000000D	Sets IRQ latency.
0x40000034	SysTick timer clock control	SYSCON_STCALIB	R/W	STCALIBINIT	Corrects SysTick timer clock.
0x40000040	Peripheral clock enable register	SYSCON_PCLK_EN	R/W	0xFFFFFCFB	Stops a clock supplied to each peripheral.
0x40000044	Peripheral clock disable register	SYSCON_PCLK_DIS	R/W	0x00000000	Stops a clock supplied to each peripheral.

0x40000048	Peripheral power management register 1	SYSCON_PPM1	R/W	0x00000000	Automatically stops a peripheral clock at DEEPSLEEP.
0x4000004C	Peripheral power management register 2	SYSCON_PPM2	R/W	0x00000000	Automatically stops a peripheral clock at SLEEP.
0x40000050	Peripheral reset register	SYSCON_PRST_CON	R/W	0x00000000	Resets each peripheral.
0x40000060	Peripheral clock control register	SYSCON_PERI_ CKCON	R/W	0x00000000	Clock control register.
0x40000064	System clock control register	SYSCON_SYS_CKCON	R/W	0x00000000	System clock control register.
0x40000100	ISP system clock register	SYSCON_ISP_SYSCLK	R	-	Indicates a system clock frequency when started from the boot program area.
0x40000104	ISP UART source clock register	SYSCON_ISP_UART_ SCLK	R	-	Indicates a UART baud rate reference frequency when started from the boot program area.
0x40000108	ISP SSI slave source clock register	SYSCON_ISP_SSIS_ SCLK	R	-	Indicates a SSI slave sampling clock frequency when started from the boot program area.
0x4000010C	ISP timer clock source register	SYSCON_ISP_TMR_ SCLK	R	-	Indicates a timer clock frequency when started from the boot program area.

7.6.3 Description of Registers

7.6.3.1 ID Register 0 (0x40000000) (SYSCON_ID0)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PLATFORM_ID[7:0]								CHIP_ID[15:0]																CONFIGURATION[3:0]				REVISION[3:0]			
Initial value	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Configuration of chip and system is shown to software, debugger, etc.

PLATFORM_ID[7:0]	Indicates that this is Ux (Cortex®-M0+) Platform. PLATFORM_ID[7:4]: CoreID: 0x1 Cortex®-M0+ (0x0 Cortex®-M0)
	PlatformID[3:0]: Version: 0x0 MTB not included 0x1 MTB included
CHIP_ID[15:0]	This field is used to distinguish among chips. The upper 4 bits and the lower 12 bits are used for the field category and the product category respectively.
CONFIGURATION[3:0]	This field is used to manage derived products. Normally used to distinguish ROM code derived products, memory size derived products, etc.
REVISION[3:0]	This field is used for version management. Normally used to distinguish versions based on the circuit update caused by a bug fix or any other reason.

7.6.3.3 Remapping Control Register (0x40000010) (SYSCON_REMAP_CON)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																				BANK_SEL_MON	BANK_MODE_MON	BANK_SEL	BANK_MODE	Reserved			REMAP_EN	REMAP[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RWS	RWS	R	R	R	RWS	RWS	RWS	RWS	RWS

S: Sticky:= Not initialized at software reset.

BANK_SEL_MON	Displays the bank in 2 bank mode. 0: FLASH0 bank (Flash00/Flash01) 1: FLASH1 bank (Flash10/Flash11)
BANK_MODE_MON	Displays the bank mode. 0: 1 bank mode 1: 2 bank mode
BANK_SEL	Selects the bank in 2 bank mode. 0: FLASH0 bank (Flash00/Flash01) 1: FLASH1 bank (Flash10/Flash11) This register value is reflected by SYSRESETREQ reset.
BANK_MODE	Selects the bank mode. 0: 1 bank mode (initial value) 1: 2 bank mode This register value is reflected by SYSRESETREQ reset.
REMAP_EN	0: Selects a boot device depending on the state of external pin MODE0. 1: Selects a boot device depending on the state of REMAP[3:0] bits of this register.
REMAP[3:0]	Selects a boot device. When a value of REMAP[3:0] bits is as follows: 4'b0000: Flash ROM responds. 4'bxxx1: Internal SRAM responds. 4'bxx10: Internal ROM responds. (Not implemented) 4'bx100: The device placed at the address set by the remapping base address responds. 4'b1000: External DRAM responds. (Not implemented)

7.6.3.4 Remapping Base Address Register (0x40000014) (SYSCON_REMAP_BASE)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved		REMAP_BASE[29:9]																					Reserved									
Initial value	0	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	0	0	0	0	0	0	0	0
R/W	R	R	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	R	R	R	R	R	R	R	R	R

S: Sticky:= Not initialized at software reset.

When a device placed at the address set by this register is selected as the boot device by the REMAP register, assign the 6 KB area starting at the address set by this register sequentially to the area starting at the address 0. Normal operation cannot be guaranteed if any device other than the bootable device is selected.

* The initial value of this register is 0x1003_E800. (6 KB of the boot program area)
* Area assigned to the remapping area is only 6 KB at the remapping by the remapping base address. If the space exceeding 6 KB is needed for ISP, for example, the program needs to jump to the area where the entity such as Flash ROM is placed (0x10000000 and after for Flash ROM) to be executed, except for the minimum necessary codes such as exception vectors.

7.6.3.5 CPU Control Register (0x40000020) (SYSCON_CPU_CON)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															RESET_LU
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RWS	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RWS

S: Sticky:= Not initialized at software reset.

Set the operation to be executed when the CPU falls in the Lockup state due to the unrecoverable exception.

RESET_LU	Sets the operation when the CPU goes to the LOCKUP state. 0: No action is taken. At this time, no program is executed by CPU, and only access to internal resource from a debugger is possible. 1: Resets the whole system. This bit is initialized only by the power-on reset.

7.6.3.6 CPU Status Register (0x40000024) (SYSCON_CPU_ST)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved														LVD1RST	LVD0RST	Reserved							WDTRST	Reserved							LOCKUP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RWS	RWS	R	R	R	R	R	R	R	R	R	R	R	R	R	RWS	RWS	R	R	R	R	R	R	R	RWS	R	R	R	R	R	R	R	RWS

S: Sticky:= Not initialized at software reset.

Indicates CPU state.

LVD0RST	Indicates that CPU is reset by the low voltage detection (LVD0). This bit is not initialized by means other than the power-on reset. To clear this bit, write 1.
LVD1RST	Indicates that CPU is reset by the low voltage detection (LVD1). This bit is not initialized by means other than the power-on reset. To clear this bit, write 1.
WDTRST	Indicates that CPU is reset by the watchdog timer. This bit is not initialized by means other than the power-on reset. Information in this bit is reserved even when a reset occurs due to expiration of the watchdog timer. To clear this bit, write 1.
LOCKUP	Indicates that CPU falls in the LOCKUP state. (This bit is set when the LOCKUP pin of CPU is asserted.) This bit is not initialized by means other than the power-on reset. Information in this bit is reserved even when the system is configured to be reset at LOCKUP by the CPU control register. To clear this bit, write 1.

7.6.3.7 IRQ Control Register (0x40000030) (SYSCON_IRQ_CON)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								IRQLATENCY[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
RW	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

IRQLATENCY[7:0]	Sets the minimum number of cycles from the interrupt pending to the vector fetch in NVIC. Normally the initial value should be used.

7.6.3.8 SysTick Timer Clock Control Register (0x40000034) (SYSCON_STCALIB)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NOREF	SKEW	Reserved						TENMS																							
Initial value	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0
R/W	RW	RW	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Corrects SysTick timer clock source.

SYSCON_STCALIB[31]	NOREF	Indicates that there is no clock source other than the system clock. If the system clock frequency is variable, it is recommended to use a fixed frequency clock which is independent of the system clock frequency as the SysTick timer. In this case, set 0 to this bit.
SYSCON_STCALIB[30]	SKEW	If a clock source frequency is not an integer multiple of 100 Hz, set this bit to 1.
SYSCON_STCALIB[23:0]	TENMS	Set 10 ms (100 Hz) divided by the cycle of clock source as the correction value. The initial value can be set by a parameter.

Setting example of the correction value is shown below.

The value of SYST_CALIB register (read only) within Cortex®-M0+ can be rewritten by rewriting this register. The value of this register does not need to be rewritten because it is normally determined to be a unique value by the system configuration (SysTick clock configuration).

STCSEL	System clock Frequency (MHz)	Source clock frequency (MHz)	10ms/(System clock cycle)	Correction value (HEX)
0	5	5	50000	C350
0	33.3	33.3	333333.3	51615
0	40	40.0	400000.0	61A80
0	50	50.0	500000.0	7A120

7.6.3.9 Peripheral Clock Enable Register (0x40000040)[1] (SYSCON_PCLK_EN)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	FDMA_EN	DMA_EN	AES_EN	SPI2_EN	SPI1_EN	SPI0_EN	ADC_EN	FTM0_EN	GPIO2_EN	GPIO1_EN	GPIO0_EN	TMRA_EN	TMRB_EN	TMRC_EN	TMRD_EN	TMRE_EN	EXT_TMR_EN	STDGPIO_EN	CLKTMR0_EN	I2C0_EN	RAND0_EN	SSIS0_EN	Reserved		DIO0_EN	UART2_EN	UART1_EN	UART0_EN	LVD_EN	Reserved	RTC_EN	WDT_EN
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	0	1	1
R / W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	R	RW	RW	RW	RW	RW	R	RW	RW

Read

0: Indicates that the clock supply to the corresponding peripheral is stopped.

1: A clock is supplied to the corresponding peripheral.

Write

0: Writing is ignored.

1: Starts clock supply to the corresponding peripheral.

* Each bit may be changed depending on the presence or absence of peripherals.

After making the peripheral clock enabled (after writing 1 to the corresponding bit of the target peripheral), be sure to start the operation of peripheral after the target bit is set to 1.

[1]It is assumed to be controlled by the system software such as OS, boot loader, or power manager instead of an individual driver software.

7.6.3.10 Peripheral Clock Disable Register (0x40000044) (SYSCON_PCLK_DIS)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	FDMA_DIS	DMA_DIS	AES_DIS	SPI2_DIS	SPI1_DIS	SPI0_DIS	ADC_DIS	FTM0_DIS	GPIO2_DIS	GPIO1_DIS	GPIO0_DIS	TMRA_DIS	TMRB_DIS	TMRC_DIS	TMRD_DIS	TMRE_DIS	EXT_TMR_DIS	STDGPIO_DIS	CLKTMR0_DIS	I2C0_DIS	RAND0_DIS	SSIS0_DIS	Reserved		DIO0_DIS	UART2_DIS	UART1_DIS	UART0_DIS	LVD_DIS	reserved	RTC_DIS	WDT_DIS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	R	RW	RW	RW	RW	RW	R	RW	RW

Read

0: A clock is supplied to the corresponding peripheral.

1: Indicates that the clock supply to the corresponding peripheral is stopped.

Write

0: Writing is ignored.

1: Stops clock supply to the corresponding peripheral.

* Each bit may be changed depending on the presence or absence of peripherals.

After making the peripheral clock disabled (after writing 1 to the corresponding bit of the target peripheral), and after the clock of the corresponding bit is stopped, 1 can be read from the corresponding bit. If you need to wait for the completion of clock stop, be sure to wait for the target bit to be set to 1 after reading this register.

7.6.3.11 Peripheral Power Management Register 1 (0x40000048) (SYSCON_PPM1)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	FDMA_PPM1	DMA_PPM1	AES_PPM1	SPI2_PPM1	SPI1_PPM1	SPI0_PPM1	ADC_PPM1	FTM0_PPM1	GPIO2_PPM1	GPIO1_PPM1	GPIO0_PPM1	TMRA_PPM1	TMRB_PPM1	TMRC_PPM1	TMRD_PPM1	TMRE_PPM1	EXT_TMR_PPM1	STDGPIO_PPM1	CLKTMR0_PPM1	I2C0_PPM1	RAND0_PPM1	SSIS0_PPM1	Reserved		DIO0_PPM1	UART2_PPM1	UART1_PPM1	UART0_PPM1	LVD_PPM1	FCNT_PPM1	RTC_PPM1	WDT_PPM1
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS	R	R	RWS	RWS	RWS	RWS	RWS	RWS	RWS	RWS

S: Sticky:= Not initialized at software reset.

A clock for each peripheral is automatically stopped at the DEEPSLEEP state.

0: The clock is not stopped at DEEPSLEEP.

(It is stopped if attempted by the peripheral clock disable register.)

1: The clock is stopped at DEEPSLEEP.

[Notices]

The clocks corresponding to the following bits are always stopped at DEEPSLEEP regardless of these register setting values:

02,04-07,10-14,21-31

7.6.3.12 Peripheral Power Management Register 2 (0x4000004C) (SYSCON_PPM2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	FDMA_PPM2	DMA_PPM2	AES_PPM2	SPI2_PPM2	SPI1_PPM2	SPI0_PPM2	ADC_PPM2	FTM0_PPM2	GPIO2_PPM2	GPIO1_PPM2	GPIO0_PPM2	TMRA_PPM2	TMRB_PPM2	TMRC_PPM2	TMRD_PPM2	TMRE_PPM2	EXT_TMR_PPM2	STDGPIO_PPM2	CLKTMR0_PPM2	I2C0_PPM2	RAND0_PPM2	SSIS0_PPM2	Reserved		DIO0_PPM2	UART2_PPM2	UART1_PPM2	UART0_PPM2	LVD_PPM2	FCNT_PPM2	RTC_PPM2	WDT_PPM2
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RWS

A clock for each peripheral is automatically stopped at the SLEEP state.

0: The clock is not stopped at SLEEP.

(It is stopped if attempted by the peripheral clock disable register.)

1: The clock is stopped at SLEEP.

Setting by the peripheral clock enable/disable register, setting this register, power state, and state of clock supplied to peripheral are shown in Table Combination of Peripheral Clock Control Related Registers and Clock State.

RUN state is where all functions except for bus access are supplying operable clocks as with the normal time. If you use the peripheral interrupt as a return cause from SLEEP/DEEPSLEEP, the target peripheral must be in RUN state. STOP state is where all stoppable clocks are stopped. Low power consumption can be realized by stopping clocks of peripherals unnecessary for the operation.

If the target peripheral clock is made disabled by the peripheral clock enable register/disable register, the peripheral always goes to the stoppable state regardless of the peripheral power management register 1, 2 and power state. Hardware can stop all clocks connected to the

target peripheral. No operation/response for peripherals can be guaranteed[2] .

If the target peripheral clock is made enabled by the peripheral clock enable register/disable register, the clock is controlled according to the corresponding bits of the peripheral power management register 1, 2 and power state. If the corresponding bit of the peripheral power management register 2 is 1, the clock of the target peripheral is automatically stopped when CPU goes to SLEEP or DEEPSLEEP state.

If the peripheral power management register 2 is 0, the clock is controlled by the corresponding bit of the peripheral power management register 1 and the power state. If the corresponding bit of the peripheral power management register 1 is 1, the clock of the target peripheral is automatically stopped when CPU goes to DEEPSLEEP state[3] .

Table Combination of Peripheral Clock Control Related Registers and Clock State

Peripheral clock enable/ disable register	SYSCON_PPM1[n]	SYSCON_PPM2[n]	Power state	Peripheral clock supply status
Enable	1	0	DEEPSLEEP	STOP
	0			RUN
	*		SLEEP	RUN
	*		ACTIVE	RUN
	*	1	DEEPSLEEP	STOP
			SLEEP	STOP
			ACTIVE	RUN
Disable	*	*	*	STOP

[2] It is recommended that clocks of peripherals not used in the system are stopped at the system start using the peripheral disable register.
[3] When using the target peripheral interrupt as a return event from DEEPSLEEP, be sure to clear the corresponding bit of the peripheral power management register 1 to 0.

7.6.3.13 Peripheral Reset Register (0x40000050) (SYSCON_PRST_CON)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	FDMA_PRST	DMA_PRST	AES_PRST	SPI2_PRST	SPI1_PRST	SPI0_PRST	ADC_PRST	FTM0_PRST	GPIO2_PRST	GPIO1_PRST	GPIO0_PRST	TMRA_PRST	TMRB_PRST	TMRC_PRST	TMRD_PRST	TMRE_PRST	EXT_TMR_PRST	STDGPIO_PRST	CLKTMR0_PRST	I2C0_PRST	RAND0_PRST	SSI0_PRST	MODE_CNT_PRST	RF_PRST	DIO0_PRST	UART2_PRST	UART1_PRST	UART0_PRST	LVD_PRST	Reserved	RTC_PRST[4]	WDT_PRST
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	RW	RW

When writing 1, the corresponding peripheral is reset.

This bit is automatically cleared to 0 after the reset is completed.

To start the operation of the target peripheral after reset, you must wait for the reset to be completed. After the reset, be sure to wait for the corresponding bit of the target peripheral to be cleared to 0.

[4] The initial value of the RTC control register is undefined and not initialized by this bit.

7.6.3.14 Peripheral Clock Control Register (0x40000060) (SYSCON_PCLK_CON)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

7.6.3.15 System Clock Control Register (0x40000064) (SYSCON_SYS_CKCON)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

7.6.3.16 ISP System Clock Register: 0x40000100 (SYSCON_ISP_SYSCLK)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ISP_SYSCLK																															
Initial value	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Indicates the system clock frequency when being started from the remapping base address by the external pin (normally ISP program is executed). (Unit: Hz)

7.6.3.17 ISP UART Source Clock Register: 0x40000104 (SYSCON_ISP_UART_SCLK)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ISP_UART_SCLK																															
Initial value	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Indicates the source clock frequency used for the UART baud rate generation when being started from the remapping base address by the external pin (normally ISP program is executed). (Unit: Hz)

7.6.3.18 ISP SSI Slave Source Clock Register: 0x40000108 (SYSCON_ISP_SSIS_SCLK)

	3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
	ISP_SSIS_SCLK																															
Initial value	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Indicates the sampling source clock frequency of SSI Slave when being started from the remapping base address by the external pin (normally ISP program is executed). (Unit: Hz)

The serial clock is supplied from the master device. The sampling clock is the same frequency as the system clock.

7.6.3.19 ISP Timer Source Clock Register: 0x4000010C (SYSCON_ISP_TMR_SCLK)

	3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
	ISP_TMR_SCLK																															
Initial value	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Indicates 32-bit timer frequency when being started from the remapping base address by the external pin (normally ISP program is executed). (Unit: Hz)

7.6.3.20 ISP Control Register: 0x40000110 to 0x4000017F

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Register area reserved for future expansion of the ISP features.

Writing to registers is ignored. 0 is read when reading.

7-7.UART

7.7.1 General Description

A start-stop synchronous serial communication interface which has functions equivalent to the industry standard 16550. The features are shown below.

- Includes a 16-byte FIFO for each of transmission and reception.
- Full-duplex communication is possible.
- Includes a programmable baud rate generator. Note that the baud rate is the same for transmission and reception.
- The character size of 5- to 8-bit is supported.
- 1 or 2 (1.5 for the 5-bit character size) stop bit can be selected.
- For parity generation/check, supports even/odd/none/stick.
- Supports the auto-flow control function.

7.7.2 List of Registers

The programming model of this UART consists of registers shown in Table List of Registers Compatible with 16550 and Table List of Registers Not Compatible with 16550.

Table List of Registers Compatible with 16550

Address	Name	Symbol	Size	R/W	Initial value
BASE+0x00	Receive Buffer Register	UARTn_RBR	32	R	0x00000000
	Transmit Holding Register	UARTn_THR	32	W	0x00000000
	Divisor Latch Low	UARTn_DLL	32	R/W	0x00000000
BASE+0x04	Interrupt Enable Register	UARTn_IER	32	R/W	0x00000000
	Divisor Latch High	UARTn_DLH	32	R/W	0x00000000
BASE+0x08	Interrupt Identity Register	UARTn_IIR	32	R	0x00000001
	FIFO Control Register	UARTn_FCR	32	W	0x00000000
BASE+0x0C	Line Control Register	UARTn_LCR	32	R/W	0x00000000
BASE+0x10	Modem Control Register	UARTn_MCR	32	R/W	0x00000000
BASE+0x14	Line Status Register	UARTn_LSR	32	R	0x00000060
BASE+0x18	Modem Status Register	UARTn_MSR	32	R	0x00000000
BASE+0x1C	Scratchpad Register	UARTn_SCR	32	R/W	0x00000000

Table List of Registers Not Compatible with 16550

Address	Name	Symbol	Size	R/W	Initial value
BASE+0x20	Low Power Divisor Latch (Low) Register	UARTn_LPDLL	32	R/W	0x00000000
BASE+0x24	Low Power Divisor Latch (High) Register	UARTn_LPDLH	32	R/W	0x00000000
BASE+0x30	Shadow Receive Buffer Register	UARTn_SRBR	32	R	0x00000000
| BASE+0x6C	Shadow Transmit Holding Register	UARTn_STHR	32	W	0x00000000
BASE+0x70	FIFO Access Register	UARTn_FAR	32	R/W	0x00000000
BASE+0x74	Transmit FIFO Read	UARTn_TFR	32	R	0x00000000
BASE+0x78	Receive FIFO Write	UARTn_RFW	32	W	0x00000000
BASE+0x7C	UART Status Register	UARTn_USR	32	R	0x00000006
BASE+0x80	Transmit FIFO Level	UARTn_TFL	32	R	0x00000000
BASE+0x84	Receive FIFO Level	UARTn_RFL	32	R	0x00000000
BASE+0x88	Software Reset Register	UARTn_SRR	32	W	0x00000000
BASE+0x8C	Shadow Request to Send	UARTn_SRTS	32	R/W	0x00000000
BASE+0x90	Shadow Break Control Register	UARTn_SBCR	32	R/W	0x00000000
BASE+0x94	Shadow DMA Mode	UARTn_SDMAM	32	R/W	0x00000000
BASE+0x98	Shadow FIFO Enable	UARTn_SFE	32	R/W	0x00000000
BASE+0x9C	Shadow RCVR Trigger	UARTn_SRT	32	R/W	0x00000000
BASE+0xA0	Shadow TX Empty Trigger	UARTn_STET	32	R/W	0x00000000
BASE+0xA4	Halt TX	UARTn_HTX	32	R/W	0x00000000
BASE+0xA8	DMA Software Acknowledge	UARTn_DMASA	32	W	0x00000000
BASE+0xF4	Component Parameter Register	UARTn_CPR	32	R	0x00011F72
BASE+0xF8	UART Component Version	UARTn_UCV	32	R	0x3331342A
BASE+0xFC	Component Type Register	UARTn_CTR	32	R	0x44570110

* n indicates a module number (n = 0, 1, 2).

* Base address of UART0 (n = 0) is 0x40004000.

* Base address of UART1 (n = 1) is 0x40004400.

* Base address of UART2 (n = 2) is 0x40004800.

* When using In-System Programming (ISP) that rewrites the on-chip Flash-ROM included in a system, apply UART0 of BASE = 0x40004000 (n=0).

7.7.3 Description of Registers

7.7.3.1 Receive Buffer Register (UARTn_RBR): BASE+0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								RBR							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
7:0	RBR	R	Received data. This data is valid when the Data Ready bit of the Line Status Register is 1.

7.7.3.2 Transmit Holding Register (UARTn_THR): BASE+0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								THR							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Name	R/W	Description
7:0	THR	W	Transmitted data. Write to these bits only when the THR Empty bit of the Line Status Register is 1.

7.7.3.3 Divisor Latch Low (DLL): UARTn_BASE+0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																								DLL							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Description
7:0	DLL	R/W	Low-order 8 bits of a frequency divider for 16-bit baud rate.

7.7.3.4 Interrupt Enable Register (UARTn_IER): BASE+0x04

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																								PTIME	Reserved			EDSSI	ELSI	ETBEI	ERBFI
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	R	R	R	RW	RW	RW	RW

Bit	Name	R/W	Description
7	PTIME	R/W	Programmable THRE Interrupt Mode Enable. Controls generation of the THRE interrupt. 0: Disabled 1: Enabled
3	EDSSI	R/W	Enable MODEM Status Interrupt Controls a mask of the MODEM Status interrupt. 0: Mask 1: No mask
2	ELSI	R/W	Enable Receiver Line Status Interrupt Controls a mask of the Receiver Line Status interrupt. 0: Mask 1: No mask
1	ETBEI	R/W	Enable Transmit Holding Register Empty Interrupt Controls a mask of the THRE interrupt. 0: Mask 1: No mask
0	ERBFI	R/W	Enable Received Data Available Interrupt Controls a mask of the Received Data Ready interrupt and the character timeout interrupt in the FIFO mode. 0: Mask 1: No mask

7.7.3.5 Divisor Latch High (UARTn_DLH): BASE+0x04

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																								DLH							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Description
7:0	DLH	R/W	High-order 8 bits of a frequency divider for 16-bit baud rate.

7.7.3.6 Interrupt Identity Register (UARTn_IIR): BASE+0x08

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								FIFOSE		Reserved		IID			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
7:6	FIFOSE	R	FIFOs Enabled. When the FIFO Enable bit of FCR is set and both FIFOs for the transmit and receive are enabled, both bits in this field are set. 00: Disabled 11: Enabled
3:0	IID	R/W	Interrupt ID. Indicates an interrupt with the highest priority among those occurring. For priority of interrupts, refer to "Table 7-7 Interrupt List".

Table 7-7 Interrupt List

Interrupt ID			Interrupt Pending	Priority Level	source	How to reset
FIFO mode Only	Common to all modes					
bit3	bit2	bit1	bit0			
0	0	0	1	-	State in which no interrupt has been generated	-
0	1	1	0	Highest	Overrun Error, Parity Error, Framing Error, or Break Interrupt has been detected	Read LSR
0	1	0	0	2nd	Data has been received. Or the number of characters in FIFO has reached a trigger level in the FIFO mode.	Read the received characters (In the FIFO mode, read characters until the number of remaining characters becomes fewer than a trigger level).
1	1	0	0	2nd	In the state where at least one character exists in the receive FIFO, no incoming or outgoing characters has been detected for the receive FIFO during the time duration of 4 characters or more.	Read RBR (the receive FIFO is visible in the FIFO mode).
0	0	1	0	3rd	No character exists in THR.	Read IIR or write characters to THR when the interrupt ID indicates this cause.
0	0	0	0	4th	When any of the events below occurs in MSR. For details of conditions, refer to "2.3.9. MSR". •DCTS has been set •DDSR has been set •TERI has been set •DDCD has been set	Read MSR

7.7.3.7 FIFO Control Register (UARTn_FCR): BASE+0x08

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																								RT		TET		DMAM	XFIFOR	RFIFOR	FIFOE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Name	R/W	Description
7:6	RT	W	RCVR Trigger. Field to specify a trigger level of the receive FIFO interrupt. The meaning of values is as shown in the table below: 00: One character exists in FIFO 01: At least one character exists in a quarter of the FIFO 10: At least one character exists in a half of the FIFO 11: Less than two characters until FIFO becomes FULL
5:4	TET	W	TX Empty Trigger. Sets a threshold level of the FIFO empty for THRE interrupt when the THRE interrupt mode is enabled. 00: FIFO is empty 01: Two characters exist in the FIFO 10: At least one character exists in a quarter of the FIFO 11: At least one character exists in a half of the FIFO

3	DMAM	W	DMA Mode. When this bit is set, the output signal mode of the ports dma_tx_req_n and dma_rx_req_n changes from 0 to 1. The difference between the mode 0 and mode 1 is as follows: 0: mode 0 - Drops the DMA transfer request each time one character is transferred. 1: mode 1 - Transfers characters as much as possible by one DMA request. Therefore, the request is not dropped at receive until the receive FIFO becomes empty. The request is not dropped at transmit until the transmit FIFO becomes full.
2	XFIFOR	W	XMIT FIFO Reset. Setting this bit clears the transmit FIFO (however, no transmit shift register is affected). A value of this bit is automatically cleared after the transmit FIFO is cleared.
1	RFIFOR	W	RCVR FIFO Reset. Setting this bit clears the receive FIFO (however, no receive shift register is affected). A value of this bit is automatically cleared after the receive FIFO is cleared.
0	FIFOE	W	FIFO Enable. Setting this bit causes this UART to enter the FIFO mode. Both the transmit and receive FIFOs are enabled in the FIFO mode. If a value of this bit changes, a part of XMIT and RCVR FIFO is reset.

7.7.3.8 Line Control Register (UARTn_LCR): BASE+0x0C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																								DLAB	BC	Stick Parity	EPS	PEN	STOP	DLS	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Description
7	DLAB	R/W	Divisor Latch Access Bit. Field to control access to the dividing ratio setting registers (DLH, DLL, LPDLL, LPDLH) for the baud rate generator. When this is set to 1, reading/writing from/to the dividing ratio setting registers is enabled.
6	BC	R/W	Break Control Bit. Field to control transmission of break condition. When this is set to 1, the serial data output pin TXD is set to the space state (logical "0") (Only a value of pin is set to the space state and no other logic of the transmitter section is affected).

5	Stick Parity	R/W	Stick Parity. Field to instruct to use a fixed value as the value of parity bit to be generated/checked when performing communication. This bit is enabled when the generation/check of parity is specified by PEN. 0: No Stick parity is performed (generation/check of parity is performed normally). 1: When odd parity is specified (PEN = 1, EPS = 0), a value of parity bit to be generated/checked is fixed to 1. When even parity is specified (PEN = 1, EPS = 1), a value of parity bit to be generated/checked is fixed to 0.
4	EPS	R/W	Even Parity Select. Field to specify a type of parity to be generated/checked when the generation/check of parities is specified by PEN. 0: Generates/checks the odd parity (summation of the number of 1 bits in the data of character and parity bits should be odd). 1: Generates/checks the even parity (summation of the number of 1 bits in the data of character and parity bits should be even).
3	PEN	R/W	Parity Enable. Field to specify that a parity is generated/checked when transmitting/receiving a character. 0: No parity bit is generated/checked. 1: The parity bit is inserted between the last bit configuring a character and the stop bit at the transmit.
2	STOP	R/W	Number of stop bits. Flag to specify the length of stop bit(s) added when transmitting/receiving characters. Stop bit(s) with a length corresponding to the value of STOP is added at the transmit. However, the stop bit checked at the receive is only 1 bit regardless of the value of STOP. 0: The stop bit is 1 bit. 1: If the character length is 5 bits, stop bits of 1.5 bits are added; if it is 6, 7, or 8 bits, 2 bits are added.
1:0	DLS	R/W	Data Length Select. Field to specify the character length (the number of bits per character) at the serial transfer. 00: 5 bits 01: 6 bits 10: 7 bits 11: 8 bits

7.7.3.9 Modem Control Register (UARTn_MCR): BASE+0x10

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																									SIRE	AFCE	LB	OUT2	OUT1	RTS	DTR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Description
6	SIRE	R/W	SIR Mode Enable Controls the enabled/disabled of IrDA SIR Mode. 0: IrDA SIR Mode is disabled 1: IrDA SIR Mode is enabled * Because the IrDA mode is not supported by this LSI, this bit should be fixed to 0.
5	AFCE	R/W	Auto Flow Control Enable Controls the enabled/disabled of Auto Flow Control. Auto Flow Control is enabled when FIFOs are enabled. 0: Auto Flow Control is disabled 1: Auto Flow Control is enabled

4	LB	R/W	LoopBack Bit Setting this bit causes this UART to enter the diagnosis local loopback mode. The state of UART in the local loop back mode is as follows: ■ The module serial data output pin becomes mark state (H level). ■ The module serial data input pin is separated from the internal circuit. The serial data output inside the module is connected to the serial data input inside the UART consisting the loop back. ■ Four modem control input pins (DSR, CTS, RI, DCD) are also separated from the internal circuit. Instead for them, four modem control output pins (DTR, RTS, OUT1, OUT2) are looped back inside the module. ■ Four modem control output pins (DTR, RTS, OUT1, OUT2) become negating state (H level). ■ Operation of the transmission interrupt and reception interrupt is performed normally during the local loopback mode. However, the MODEM status interrupt (due to change of the MSR bit) is generated according to the change of the looped back value of the modem control output pin rather than the modem control input. ■ In the infrared mode, the serial data output is reversed and looped back to sir_in during which sir_out_n is set to L level.
3	OUT2	R/W	OUT2 A value of this bit is reversed and output to the module output pin OUT2.
2	OUT1	R/W	OUT1 A value of this bit is reversed and output to the module output pin OUT1.
1	RTS	R/W	Request to Send A value of this bit is reversed and output to the module output pin RTS.
0	DTR	R/W	Data Terminal Ready A value of this bit is reversed and output to the module output pin DTR.

7.7.3.10 Line Status Register (UARTn_LSR): BASE+0x14

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																								RFE	TEMT	THRE	BI	FE	PE	OE	DR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
7	RFE	R	Receiver FIFO Error bit. Valid bit during the FIFO mode. This bit is set when at least one character that has been received with any of framing error, parity error, or break detection exists in the receive FIFO. If no such characters exists in the receive FIFO, this bit is cleared by reading the LSR. This bit is always 0 while not in the FIFO mode.
6	TEMT	R	Transmitter Empty bit. Indicates that both the transmit shift register and FIFO are empty while in the FIFO mode. Indicates that both Transmitter Holding Register and the transmit shift register are empty while in the non-FIFO mode.

5	THRE	R	Transmit Holding Register Empty bit. Indicates that THR or TX FIFO is empty regardless of FIFO being enabled or not (FCR[0]) when the THRE mode is disabled (IER[7] is 0). The THRE interrupt is generated when the THRE interrupt is enabled. When the THRE mode is enabled (ER[7] is 1), the transmit FIFO indicates FULL if FIFOs are enabled (FCR[0] is 1). The THRE interrupt is controlled by FCR[5:4]. Condition for setting: Data is transferred to the transmit shift register from THR or TX FIFO and THR or TX FIFO is empty.
4	BI	R	Break Interrupt bit. Indicates that a break is detected. Condition for setting: In the UART mode, a break is assumed to be detected if the serial input stays 0 for longer period than the total transfer time of one character (transfer time of start bit + data + parity bit + stop bit). In the infrared mode, a break is assumed to be detected if the serial input stays 0 pulse for longer period than the total transfer time of one character (transfer time of start bit + data + parity bit + stop bit). For the non-FIFO mode, this bit is set when a break is detected. For the FIFO mode, one character with all bits being set to zero is transferred to the receive FIFO when a break is detected. This bit is set when a character with all bits being set to zero arrives at the head of FIFO. Condition for clearing: Cleared when reading LSR.

3	FE	R	Framing Error bit. Indicates that a value of the stop bit, added to the received character, is invalid. Condition for setting: If the value of stop bit is invalid when receiving a character, this UART estimates that the cause of the error is "due to the next start bit" and performs the recovery operation from the error. In other words, it samples this start bit again and receives the data. In the non-FIFO mode, this bit is set when a character received with the invalid stop bit is transferred to RBR. In the FIFO mode, this bit is set when a character received with the invalid stop bit arrives at the beginning of FIFO. Condition for clearing: Cleared when reading LSR.
2	PE	R	Parity Error bit. Indicates that a value of the parity bit, added to the received character, is invalid. This bit is valid when PEN(LCR[3]) is set. Condition for setting: In the non-FIFO mode, when a character received with the invalid parity bit is transferred to RBR. In the FIFO mode, if a character appearing at the beginning of the receive FIFO has been detected to contain an invalid parity bit. Condition for clearing: Cleared when reading LSR.
1	OE	R	Overrun error bit. Indicates that one or more characters are lost because reading the received characters is too late. In the non-FIFO mode, characters in RBR are overwritten. In the FIFO mode, data of the receive shift register is lost although the FIFO data is saved. Condition for setting: In the non-FIFO mode, when a new character arrives at the receiver section before reading from RBR. In the FIFO mode, when the receive FIFO is full and a new character arrives at the receiver section. Condition for clearing: Cleared when reading LSR.

0	DR	R	Data Ready bit. Indicates that received characters exist in RBR or FIFO. Condition for clearing: Characters are read from RBR when being in the non-FIFO mode. The receive FIFO becomes empty when being in the FIFO mode.

7.7.3.11 Modem Status Register (UARTn_MSR): BASE+0x18

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
7	DCD	R	Data Carrier Detect. A value of this bit changes as shown in the table below depending on a value of the LOOP bit of MCR. LOOP=0: The value of module input pin dcd_n is reversed and then read. LOOP=1: The same value as that of the OUT2 bit of MCR is read.
6	RI	R	Ring Indicator. A value of this bit changes as shown in the table below depending on a value of the LOOP bit of MCR. LOOP=0: The value of module input pin ri_n is reversed and then read. LOOP=1: The same value as that of the OUT1 bit of MCR is read.
5	DSR	R	Data Set Ready. A value of this bit changes as shown in the table below depending on a value of the LOOP bit of MCR. LOOP=0: The value of module input pin dsr_n is reversed and then read. LOOP=1: The same value as that of the DTR bit of MCR is read.

4	CTS	R	Clear to Send. A value of this bit changes as shown in the table below depending on a value of the LOOP bit of MCR. LOOP=0: The value of module input pin cts_n is reversed and then read. LOOP=1: The same value as that of the RTS bit of MCR is read.
3	DDCD	R	Delta Data Carrier Detect. This bit is set when a value of the module pin dcd_n changes, and cleared when a value of MSR is read. When the EDSSI bit of IER is set, the MODEM status interrupt is generated if this bit is set. LOOP = 0: Indicates that no value of the module pin dcd_n changes since the latest value of MSR was read. LOOP = 1: Indicates that a value of the module pin dcd_n changed at least one time since the latest value of MSR was read.
2	TERI	R	Trailing Edge of Ring Indicator. This bit is set when a value of the module pin ri_n changes from the L level to the H level, and cleared when a value of MSR is read. When the EDSSI bit of IER is set, the MODEM status interrupt is generated if this bit is set. LOOP = 0: Indicates that no value of the module pin ri_n changes from the L level to the H level since the latest value of MSR was read. LOOP = 1: Indicates that a value of the module pin ri_n changed from the L level to the H level at least one time since the latest value of MSR was read.
1	DDSR	R	Delta Data Set Ready. This bit is set when a value of the module pin dsr_n changes, and cleared when a value of MSR is read. When the EDSSI bit of IER is set, the MODEM status interrupt is generated if this bit is set. LOOP = 0: Indicates that no value of the module pin dsr_n changed since the latest value of MSR was read. LOOP = 1: Indicates that a value of the module pin DSR changed at least one time since the latest value of MSR was read.
0	DCTS	R	Delta Clear to Send. This bit is set when a value of the module pin cts_n changes, and cleared when a value of MSR is read. When the EDSSI bit of IER is set, the MODEM status interrupt is generated if this bit is set. LOOP = 0: Indicates that no value of the module pin cst_n changed since the latest value of MSR was read. LOOP = 1: Indicates that a value of the module pin cts_n changed at least one time since the latest value of MSR was read.

7.7.3.12 Scratchpad Register (UARTn_SCR): BASE+0x1C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																								SCR							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	Rw	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Description
7:0	SCR	R/W	Temporary data retention area. Programmers can use it freely.

7.7.3.13 Low Power Divisor Latch (Low) Register (UARTn_LPDLL): BASE+0x20

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																								LPDLL							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	Rw	RW	RW	RW	RW	RW	RW

* This register is not supported for use in the IrDA SIR mode. Do not change its initial value.

Bit	Name	R/W	Description
7:0	LPDLL	R/W	This register can be accessed when LCR[7] (DLAB bit) is set. This register gives the lower byte of the setting value of the Low Power Divisor Latch register that provides 115.2 kbps. If a value of the Low Power Divisor Latch register is 0, the Low Power baud rate clock is disabled and no pulse detection is performed including the Low-Power pulse detection.

7.7.3.14 Low Power Divisor Latch (High) Register (UARTn_LPDLH): BASE+0x24

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																								LPDLH							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	Rw	RW	RW	RW	RW	RW	RW

* This register is not supported for use in the IrDA SIR mode. Do not change its initial value.

Bit	Name	R/W	Description
7:0	LPDLH	R/W	This register can be accessed when LCR[7] (DLAB bit) is set. This register gives the higher byte of the setting value of the Low Power Divisor Latch register that provides 115.2 kbps. If a value of the Low Power Divisor Latch register is 0, the Low Power baud rate clock is disabled and no pulse detection is performed including the Low-Power pulse detection.

7.7.3.15 Shadow Receive Buffer Register (UARTn_SRBR): BASE+0x30-6C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																								SRBR							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
7:0	SRBR	R	This register is a shadow register for the RBR register. This register can be used to read RBR by using the burst transfer.

7.7.3.16 Shadow Transmit Holding Register (UARTn_STHR): BASE+0x30-6C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																								STHR							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Name	R/W	Description
7:0	STHR	W	This register is a shadow register for the THR register. This register can be used to write to THR by using the burst transfer.

7.7.3.17 FIFO Access Register (UARTn_FAR): BASE+0x70

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															FAR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Bit	Name	R/W	Description
0	FAR	R/W	Enables the access to FIFO aimed at testing. Writing to the receive FIFO and reading from the transmit FIFO is made enabled. 0 – FIFO access mode is disabled 1 – FIFO access mode is enabled

7.7.3.18 TFR Register (UARTn_TFR): BASE+0x74

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																								TFR							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
7:0	TFR	R	Reads the transmit FIFO when FIFO Access Mode is enabled.

7.7.3.19 Receive FIFO Write (UARTn_RFW): BASE+0x78

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																						RFFE	RFPE	RFWD							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W	W	W

Bit	Name	R/W	Description
9	RFFE	W	Writes the receive FIFO framing error information when FIFO Access Mode aimed at testing is enabled.
8	RFPE	W	Writes the receive FIFO parity error information when FIFO Access Mode aimed at testing is enabled.
7:0	RFWD	W	Writes the receive FIFO data when FIFO Access Mode aimed at testing is enabled.

7.7.3.20 UART Status Register (UARTn_USR): BASE+0x7C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																											RFF	RFNE	TFE	TFNF	Reserved
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
4	RFF	R	Indicates that the receive FIFO is full. 0: Indicates that the receive FIFO is not full. 1: Indicates that the receive FIFO is full. When the receive FIFO becomes non-full, this bit is cleared.
3	RFNE	R	Indicates that the receive FIFO is not empty. 0: Indicates that the receive FIFO is empty. 1: Indicates that the receive FIFO is not empty. When FIFO becomes empty, this bit is cleared.
2	TFE	R	Indicates that the transmit FIFO is empty. 0: Indicates that the transmit FIFO is not empty. 1: Indicates that the transmit FIFO is empty. When the transmit FIFO becomes non-empty, this bit is cleared.

1	TFNF	R	Indicates that the transmit FIFO is not full. 0: Indicates that the transmit FIFO is full. 1: Indicates that the transmit FIFO is not full. When the transmit FIFO becomes full, this bit is cleared.

7.7.3.21 Transmit FIFO Level (UARTn_TFL): BASE+080

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												TFL			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
3:0	TFL	R	Indicates the transmit FIFO level. Indicates the number of data entered in the transmit FIFO.

7.7.3.22 Receive FIFO Level Register (UARTn_RFL): BASE+0x84

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												RFL			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
3:0	RFL	R	Indicates the receive FIFO level. Indicates the number of data entered in the receive FIFO.

7.7.3.23 Software Reset Register (UARTn_SRR): BASE+0x88

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																													XFR	RFR	UR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W

Bit	Name	R/W	Description
2	XFR	W	Shadow register for the transmit FIFO reset bits (FCR[2]).
1	RFR	W	Shadow register for the receive FIFO reset bits (FCR[1]).
0	UR	W	Resets UART. It takes up to 2 cycles of UART reference clock + 3 cycles of system clock to complete the reset.

7.7.3.24 Shadow Request to Send (UARTn_SRTS): BASE+0x8C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															SRTS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Bit	Name	R/W	Description
0	SRTS	R/W	Shadow register for the RTS bit (MCR[1]).

7.7.3.25 Shadow Break Control Register (UARTn_SBCR): BASE+0x90

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															SBCR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Bit	Name	R/W	Description
0	SBCR	R/W	Shadow register for the break bit (LCR[6]).

7.7.3.26 Shadow DMA Mode (UARTn_SDMAM): BASE+0x94

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															SDMAM
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Bit	Name	R/W	Description
0	SDMAM	R/W	Shadow register for the DMA mode bit (FCR[3]).

7.7.3.27 Shadow FIFO Enable (UARTn_SFE): BASE+0x98

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															SFE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Bit	Name	R/W	Description
0	SFE	R/W	Shadow register for the FIFO enable bit (FCR[0]).

7.7.3.28 Shadow RCVR Trigger (UARTn_SRT): BASE+0x9C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																														SRT	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Bit	Name	R/W	Description
1:0	SRT	R/W	Shadow register for the receive trigger bits (FCR[7:6]).

7.7.3.29 Shadow TX Empty Trigger (UARTn_STET): BASE+0xA0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																														STET	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Bit	Name	R/W	Description
1:0	STET	R/W	Shadow register for the transmit empty trigger bits (FCR[5:4]).

7.7.3.30 Halt TX (UARTn_HTX): BASE+0xA4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															HTX
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Bit	Name	R/W	Description
0	HTX	R/W	Register for stop of transmission aiming for testing. The transmit FIFO can be intentionally set to full by stopping the transmit FIFO.

7.7.3.31 DMA Software Acknowledge (UARTn_DMASA): BASE+0xA8

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															DMASA
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Bit	Name	R/W	Description
0	DMASA	W	The DMA acknowledge is generated according to a software request.

7.7.3.32 Component Parameter Register (UARTn_CPR): BASE+0xF4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved								FIFO_MODE								Reserved		DMA_EXTRA	UART_ADD_ENCODED_PARAMS	SHADOW	FIFO_STAT	FIFO_ACCESS	ADDITIONAL_FEAT	SIR_LP_MODE	SIR_MODE	THRE_MODE	AFCE_MODE	Reserved		APB_DATA_WIDTH	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	1	1	1	0	0	1	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The configuration of the UART module is shown below.

Bit	Name	R/W	Description
23:16	FIFO_MODE	R	Indicates the number of FIFO stages. 0x00 = 0 0x01 = 16 0x02 = 32 to 0x80 = 2048 0x81- 0xff = reserved
13	DMA_EXTRA	R	Indicates whether DMA is supported. 0 - Unsupported 1 - Supported
12	UART_ADD_ENCODED_PARAMS	R	Indicates that this register (Component Parameter Register) is implemented. 0 - Not implemented 1 - Implemented
11	SHADOW	R	Indicates that Shadow Register is implemented. 0 - Not implemented 1 - Implemented
10	FIFO_STAT	R	Indicates that the bit representing FIFO status information is implemented. 0 - Not implemented 1 - Implemented
9	FIFO_ACCESS	R	Indicates that the FIFO access mode is implemented. 0 - Not implemented 1 - Implemented
8	ADDITIONAL_FEAT	R	Indicates that the UARTn_UCV register and UARTn_CRT register are implemented. 0 - Not implemented 1 - Implemented
7	SIR_LP_MODE	R	Indicates whether the Low Power IrDASIR mode is supported. 0 - Unsupported 1 - Supported (*1)
6	SIR_MODE	R	Indicates whether the IrDA SIR mode is supported. 0 - Unsupported 1 - Supported

5	THRE_MODE	R	Indicates that the THRE interrupt is supported. 0 - Unsupported 1 - Supported
4	AFCE_MODE	R	Indicates that the auto-flow control function is supported. 0 - Unsupported 1 - Supported
1:0	APB_DATA_WIDTH	R	Indicates the APB bus width in the LSI. 00 – 8 bits 01 – 16 bits 10 – 32 bits 11 – reserved

(*1) This mode is supported for the UART module but not for ML7416.

7.7.3.33 UART Component Version (UARTn_UCV): BASE+0xF8

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	UCV																															
Initial value	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
31:0	UCV	R	Indicates the component version of UART IP.

7.7.3.34 Component Type Register (UARTn_CTR): BASE+0xFC

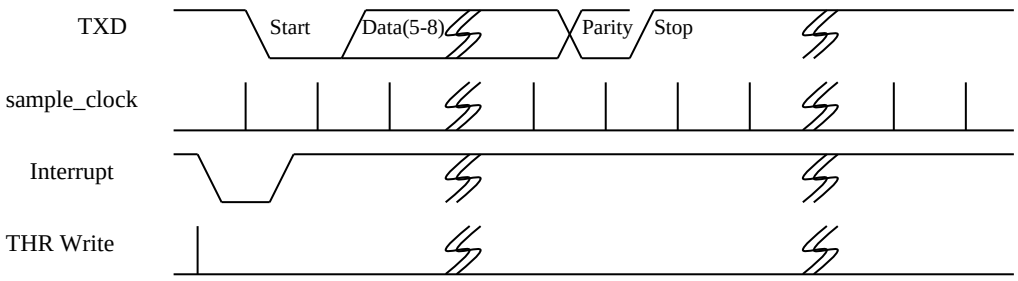
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	CTR																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
31:0	CTR	R	Indicates the peripheral ID code of UART IP.

7.7.4 Description of Operation

7.7.4.1 Data Transmission

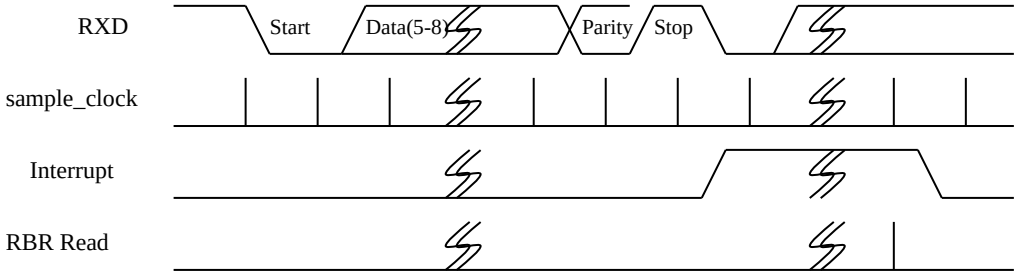
Figure below shows the transmit timing.



Transmit timing

7.7.4.2 Data Reception

Figure below shows the receive timing.



Receive timing

7.7.5 UART Pin Assignment

The UART function (UART0) is used at execution of ISP. This function is assigned to GPIOA[11:8] as the secondary function.

When selecting the primary function of GPIOA[11:8]	When selecting the secondary function of GPIOA[11:8]
GPIOA[8]	RXD
GPIOA[9]	TXD
GPIOA[10]	CTS
GPIOA[11]	RTS

7-8.SPI

7.8.1 General Description

A synchronous serial communication interface. The features are shown below.

- Performs the full-duplex data transfer.
- Master or Slave mode can be selected.
- Includes a 16-byte or 16-word (16-bit) FIFO for each of the transmission and reception sides.
- For the transfer size, 8 bits (bytes) or 16 bits (words) can be selected.
- The interrupt caused by the number of received bytes (words) and the number of untransmitted bytes (words) can be set in the range 1 to 16.
- Either LSI first or MSB first can be selected.
- The polarity and phase of the serial clock can be selected.
- Able to control the interval before/after transfer in Master mode.
- Uses the status bit to indicate the completion of transmission/reception and the FIFO status.
- Able to detect a mode fault error to avoid multi-master bus contention.
- Able to detect a write overflow error if any further writing is attempted when the transmit FIFO is in the full state.
- Generates an interrupt when the transmit/receive FIFO is in a specific state or when a cause such as mode fault error occurs.
- Able to perform the burst transfer with SSn fixed to L during the master operation.
- Has a DMA interface.

7.8.2 List of Registers

Address [H]	Name	Symbol	R/W	Initial value [H]	Description
BASE+0x00	SPI control register	SPIn_CR	R/W	0x00000000 (SPI2 only 0x00000002)	Controls the operation mode
BASE+0x04	SPI baud rate register	SPIn_BRR	R/W	0x00025002	Controls the operation mode
BASE+0x08	SPI status register	SPIn_SR	R	0x00140000	Indicates the data transfer status and the error status.
BASE+0x0C	SPI write data register	SPIn_DWR	R/W	0xXXXXXXXX	8 (16)-bit register to hold the transmit data.
BASE+0x10	SPI read data register	SPIn_DRR	R	0xXXXXXXXX	8 (16)-bit register to hold the received data.
BASE+0x18	SPI DMA register	SPIn_DMA	R/W	0x00000000 (Only SPI2 are valid)	Makes the settings for DMA.
BASE+0x1C	SPI ID register	SPIn_ID	R	0x00001010 (Only SPI2 are valid)	Reads the ID value.

* n indicates a module number (n = 0, 1, 2).

* Base address of SPI0 (n = 0) is 0x40040000.

* Base address of SPI1 (n = 1) is 0x40040400.

* Base address of SPI2 (n = 2) is 0x40040800. SPI2 is for controlling ML7396B (dedicated to master).

7.8.3 Description of Registers

7.8.3.1 SPIn_CR Register: BASE + 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	SSNL	Reserved			MOZ	SOZ	SSZ	FICLR	RFIC				TFIC				Reserved			MDFIE	ORIE	FIE	RFIE	TFIE	Reserved	CPOL	CPHA	LSBF	MODFEN	Reserved	MSTR	SPE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	RW	RW	RW	RW	RW	R	RW	RW	RW	RW	R	RW	RW

SSNL	SSN output control (Enabled only in Master mode and when SPE = 1 is set. Use this during the burst access in SPI2) 0: SSN free (normal) 1: SSN = Low fixed output (SDO pin is in Hi-Z state)
MOZ	MOSI output control 0: 0/1 output 1:HiZ
SOZ	MISO output control when SSn = 0 0: 0/1 output 1:HiZ
SSZ	SSn output control 0: 0/1 output 1:HiZ
FICLR	FIFO Clear 0: NONE 1: The number of receive/transmit bytes (words) is cleared. After clearing, set it back to "0".

RFIC	Receive FIFO interrupt control. Initial value: 0. Generates a receive DMA request when DMA is enabled. 0000: An interrupt/receive DMA request occurs when 1 byte (1 word) has been received 0001: An interrupt/receive DMA request occurs when 2 bytes (2 words) have been received : 1111: An interrupt/receive DMA request occurs when 16 bytes (8 words) have been received
TFIC	Transmit FIFO remaining byte count interrupt control. Initial value: 0. Generates a transmit DMA request when DMA is enabled. 0000: An interrupt/transmit DMA request occurs when the number of remaining byte becomes 0 byte (1 word) or less 0001: An interrupt/transmit DMA request occurs when the number of remaining byte becomes 1 byte (2 words) or less : 1111: An interrupt/transmit DMA request occurs when the number of remaining byte becomes 15 bytes (8 words) or less
MDFIE	SPI mode fault interrupt enable. Initial value: 0 0:Disable interrupt 1:Enable interrupt
ORIE	SPI overrun error interrupt enable. Initial value: 0 0:Disable interrupt 1:Enable interrupt
FIE	Transfer completion interrupt enable. Initial value: 0 0:Disable interrupt 1:Enable interrupt
RFIE	SPI reception interrupt enable. Initial value: 0 0:Disable interrupt 1:Enable interrupt
TFIE	SPI transmission interrupt enable. Initial value: 0 0:Disable interrupt 1:Enable interrupt
CPOL	Serial clock polarity. Initial value: 0 0:Serial clock default is "0" ("0" during transmission/reception) 1:Serial clock default is "1" ("1" during transmission/reception)
CPHA	Serial clock phase. Initial value: 0 0:The data is sampled at the first edge and shifted at the second edge. 1:The data is shifted at the first edge and sampled at the second edge.
LSBF	Data transfer order. Initial value: 0 0: LSB first 1: MSB first

MODFEN	Mode fault control signal The mode fault can be executed when MSTR = 1, MODEFEN = 1, and not transferring. 1: The mode fault is executed when not transferring. 0: The mode fault is not executed.
MSTR	Master/slave selection. Initial value 0 (1 only for SPI2) 0:Slave 1:Master
SPE	SPI enable. Initial value: 0 0: Disables SPI transfer 1: Enables SPI transfer

7.8.3.2 SPIn_BRR Register: BASE + 0x04

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved							DTL									LAG		LEAD		Reserved	SIZE	SPBR										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

DTL	Minimum data transfer interval (setting enabled only in Master mode). Initial value: 2
LAG	SSn-SCK delay interval (setting enabled only in Master mode). Initial value: 01 0x : 0.5 X SCK 10 : 1.0 X SCK 11 : 1.5 X SCK
LEAD	SCK-SSn delay interval (setting enabled only in Master mode). Initial value: 01 0x : 0.5 X SCK 10 : 1.0 X SCK 11 : 1.5 X SCK
SIZE	Transfer size setting 0: 8 bits 1: 16 bits

SPBR	Baud rate setting (setting enabled in Master mode). Initial value: 2 Baud Rate=fPCLK/(2* SPBR) 00_0000_000x: 2 dividing 00_0000_0010: 4 dividing 00_0000_0011: 6 dividing : 11_1111_1111: 2046 dividing

7.8.3.3 SPIn_SR Register: BASE + 0x08

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved											RFE	RFF	TFE	TFF	WDF	RFD					TFO					SPIF	MDF	ORF	FI	RFI	TFI
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

RFE	Receive FIFO Empty. Initial value: 1 0: Not empty 1: Empty (No interrupt is generated)
RFF	Receive FIFO Full. Initial value: 0 0: Not Full 1: Full (No interrupt is generated)
TFE	Transmit FIFO Empty. Initial value: 1 0: Not Empty 1: Empty (No interrupt is generated)
TFF	Transmit FIFO Full. Initial value: 0 0: Not Full 1: Full (No interrupt is generated)
WDF	Write overflow. Initial value: 0 0: Normal 1: A write overflow occurred (No interrupt is generated) This is cleared when this bit is read.
RFD	Number of bytes (words) received in the receive FIFO. Initial value: 00 00000: Empty 00001: 1Byte/1Word : 01111: 15Byte/15Word 1xxxx: 16Byte/16Word (Full)

TFO	Number of untransmitted bytes (words) in the transmit FIFO. Initial value: 00 00000: Empty 00001: 1Byte/1Word : 01111: 15Byte/15Word 1xxxx: 16Byte/16Word (Full)
SPIF	SPI1 byte (word) transfer completion display. Initial value: 0 0:Transferring 1:Transfer completed This is cleared when this bit is read.
MDF	Mode fault. Initial value: 0 0:Normal 1:A mode fault occurred. SSn changes to 0 in Master mode. (An interrupt is generated) The interrupt request is cleared by writing "1".
ORF	Overrun error flag. Initial value: 0 0: Normal 1: An overrun error occurred (an interrupt is generated) The interrupt request is cleared by writing "1".
FI	Transfer completion (transmit FIFO is empty, the last 8 bit transfer is completed). Initial value: 0. 0: No interrupt request 1: Interrupt request The interrupt request is cleared by writing "1".
RFI	Reception interrupt. Initial value: 0 0: No interrupt request 1: Interrupt request The interrupt request is cleared by writing "1".
TFI	Transmission interrupt. Initial value: 0 0: No interrupt request 1: Interrupt request The interrupt request is cleared by writing "1".

7.8.3.4 SPIn_DW Register: BASE + 0x0C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																DW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

DW	8 (16)-bit register to hold the transmit data.

7.8.3.5 SPIn_DR Register: BASE + 0x10

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																DR															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

DR	8 (16)-bit register to hold the received data.

7.8.3.6 SPI DMA Register: BASE + 0x18 (SPIn_DMA) (n = 0..3)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														RX_DMA_EN	TX_DMA_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

* This register is effective only in SPI2.

When the SPE bit of the SPI control register is set to 1, this register sets the DMA transfer request according to the values of TFIC and RFIC of the SPI control register.

Field	Symbol	Initial value	Description
1	RX_DMA_EN	1	0: Does not generate a receive DMA transfer request. 1: Generates a receive DMA transfer request.
0	TX_DMA_EN	0	0: Does not generate a transmit DMA transfer request. 1: Generates a transmit DMA transfer request.

7.8.3.7 SPI ID Register: BASE + 0x1C (SPIn_ID) (n = 0..3)

32-bit register to hold the received data.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VER																FIFO								DMA				REV			
Initial value	0x0000																0x10								0x1				0x0			
R / W	R																															

* This register is effective only in SPI2.

Indicates the ID number of SPI module.

Used for the management of version numbers and branched revisions of SPI module and the identification of SPI module by software.

Field	Symbol	Initial value	Description
31-16	ID	0	0: UxPlatform DMA standard SPI
15-8	FIFO	0x10	Indicates the number of FIFO stages.
7-4	DMA	0x1	Indicates the existence of DMA/revision. 0000: No DMA 0001: DMA (transmit/receive req/ack Handshake)
3-0	REV	0	Indicates the version.

7.8.4 Functional Description

7.8.4.1 Master Mode and Slave Mode

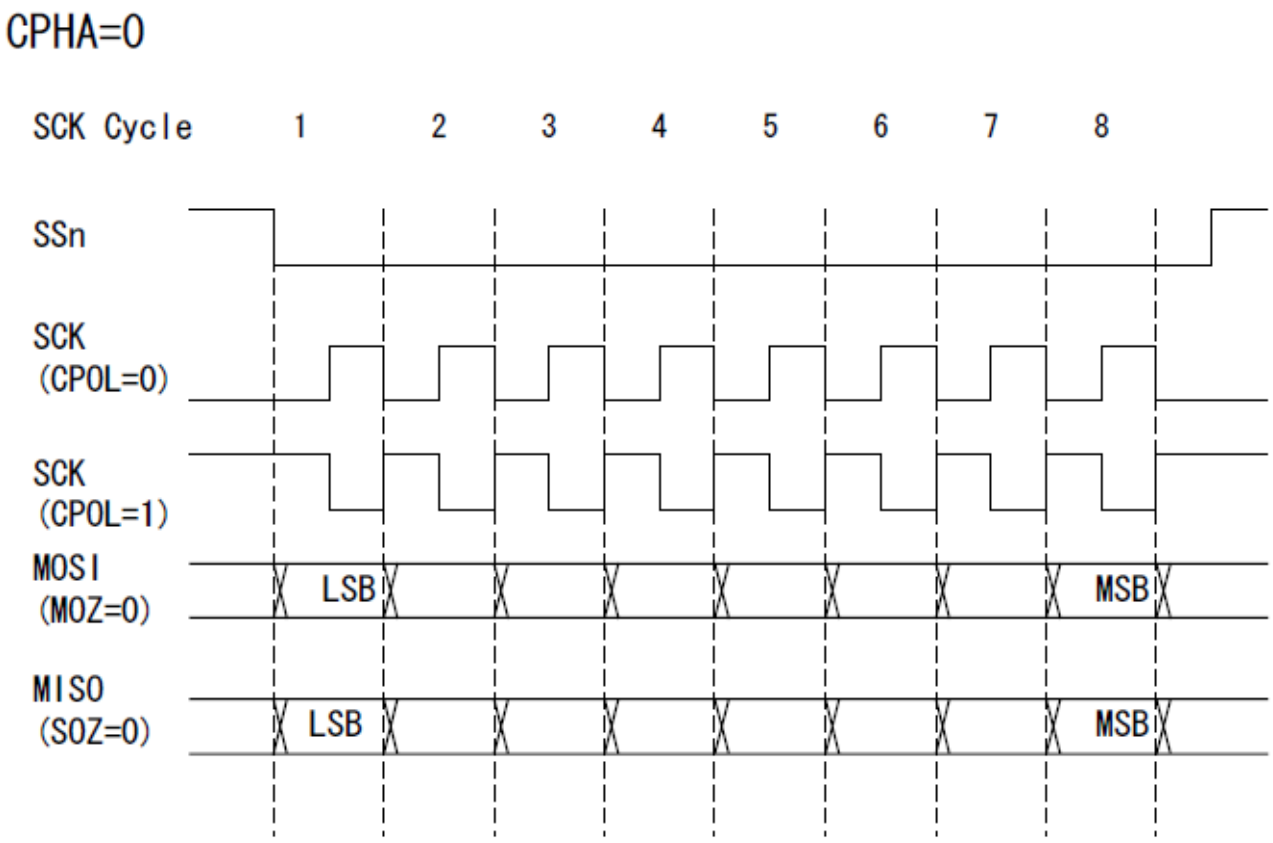
Two modes, Master mode and Slave mode, are provided as the transmit/receive mode. The mode can be selected by the MSTR bit of SPCR.
As the SPBR, LEAD, LAG, DSCK, and SSn signals are determined by the master, only the setting values of master are effective.
CPOL, CPHA, LSBF, and SIZE have the same value for both master and slave.

7.8.4.2 Control of Polarity and Phase of Serial Clock

CPOL of SPCR controls the clock polarity. CPHA controls the clock phase and determines the shift timing of transmit data and the sampling timing of received data. The master and slaves that communicate with each other must have the same setting values for CPOL and CPHA.

7.8.4.2.1 CPHA=0

The figure below shows the data transfer timing when CPHA = 0. For SCK, the case of CPOL = 0 and the case of CPOL = 1 are shown. MOSI outputs the transmit data in Master mode, and performs the sampling of the received data in Slave mode. MISO performs the sampling of the received data in Master mode, and outputs the transmit data in Slave mode. SSn is input as slave selection in Slave mode.
In Master mode, the transfer is started when data is written to SPDR. In Slave mode, the transfer is started at the SSn falling edge. The received data is sampled at the first clock edge of SCK and at the succeeding odd-numbered edges. The transmit data is shifted at the second clock edge and at the succeeding even-numbered edges.



Example for MOZ=SOZ=SSZ=SIZE=LSBF=LEAD=LAG=0.

Clock Waveform When CPHA = 0

7.8.4.2.2 CPHA=1

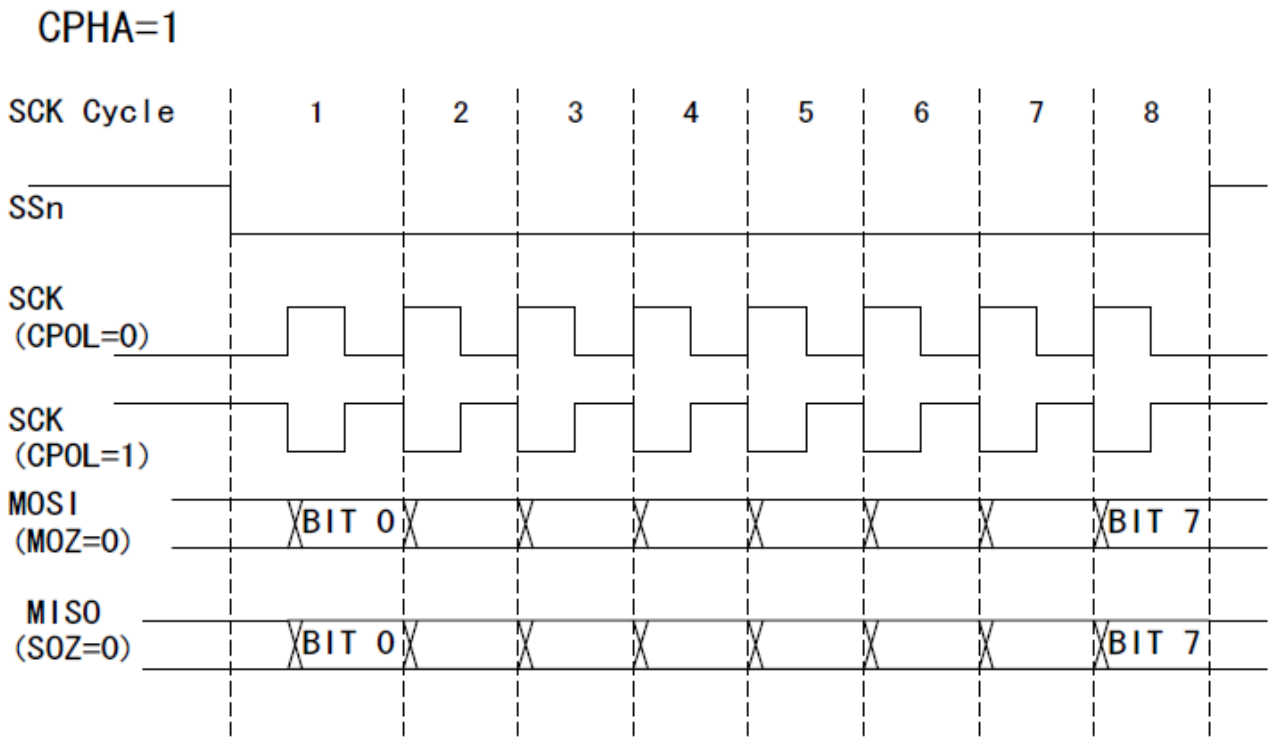
The figure below shows the data transfer timing when CPHA = 1. For SCK, the case of CPOL = 0 and the case of CPOL = 1 are shown.

MOSI outputs the transmit data in Master mode, and inputs the received data in Slave mode.

MISO inputs the received data in Master mode, and outputs the transmit data in Slave mode.

SSn is input as slave selection in Slave mode.

In Master mode, the transfer is started when data is written to SPDR. In Slave mode, the transfer is started at the first edge of SCK. The received data is sampled at the second clock edge and at the succeeding even-numbered edges. The transmit data is shifted at the first clock edge and at the succeeding odd-numbered edges.



Example for MOZ=SOZ=SSZ=SIZE=LSBF=LEAD=LAG=0.

Clock Waveform When CPHA = 1

7.8.4.3 Serial Clock Baud Rate

The baud rate can be selected by the SPBR bit of SPBRR. Only the setting of Master mode is effective. The baud-rate clock SCK is generated by dividing the system clock (SYSCLK).

The method of calculating the baud rate (fSCK) is as follows.

$$f_{SCK} = f_{SYSCLK} / (2 \times SPBR)$$

f_{SCK}: Frequency of baud-rate clock

f_{SYSCLK}: Frequency of system clock

SPBR: Value set to SPBR of the SPBRR register (1 to 1023). If set to 0, it is treated as 1.

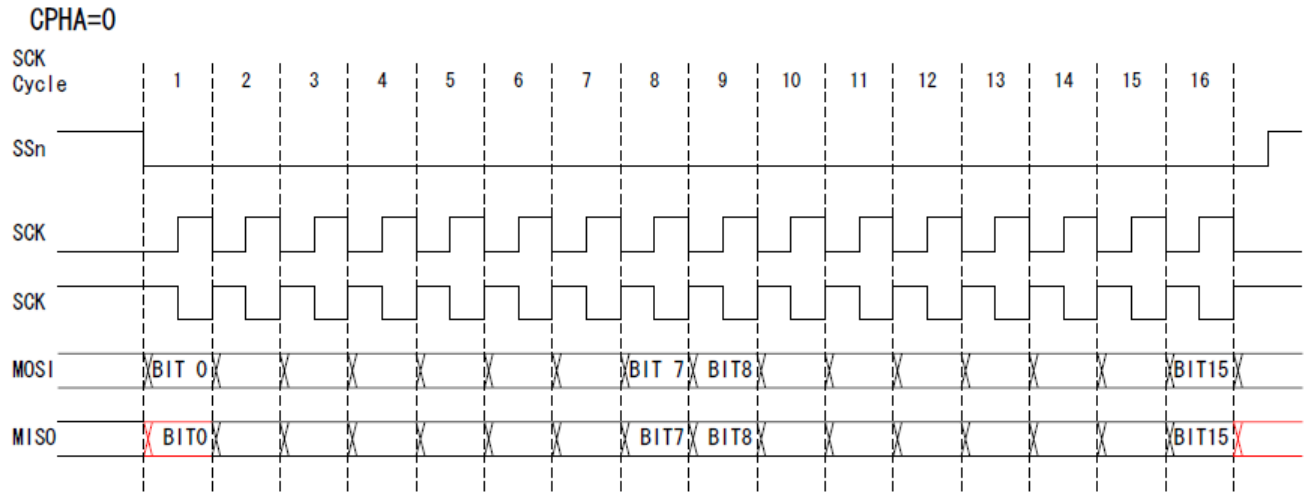
For SPBR, 1023 types of dividing (2 to 2046) can be selected.

7.8.4.4 Transfer Size

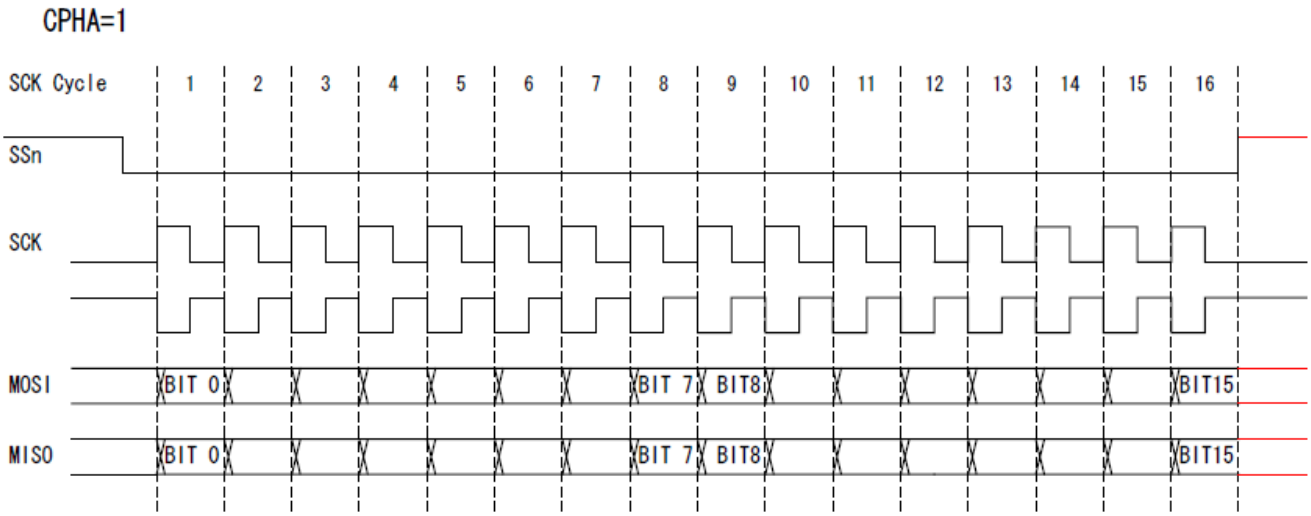
The transfer size can be selected between 8 bits (byte) or 16 bits (word).

Transfer data read/write must be adjusted to the transfer size. As the number of FIFO stages is the same for both byte and word, the number of transfers is the same.

The master and slaves that communicate with each other must have the same value for SIZE.



SPI Bus Waveform When Transfer Size (SIZE) = 1 (16 Bits)



SPI Bus Waveform When Transfer Size (SIZE) = 1 (16 Bits)

7.8.4.5 Transfer Interval Setting

LEAD (SSn-SCK time), LAG (SCK_Lag time), and TDTL (SSn(H)-SSn(H)) can be set to align the speed with the slave. Only the setting of Master mode is effective. The setting value of slave is ignored.

Setting during transferring is invalid.

(1) LEAD

A value from 0.5 to 1.5 Tsck can be set.

(2)LAG

A value from 0.5 to 1.5 Tsck can be set.

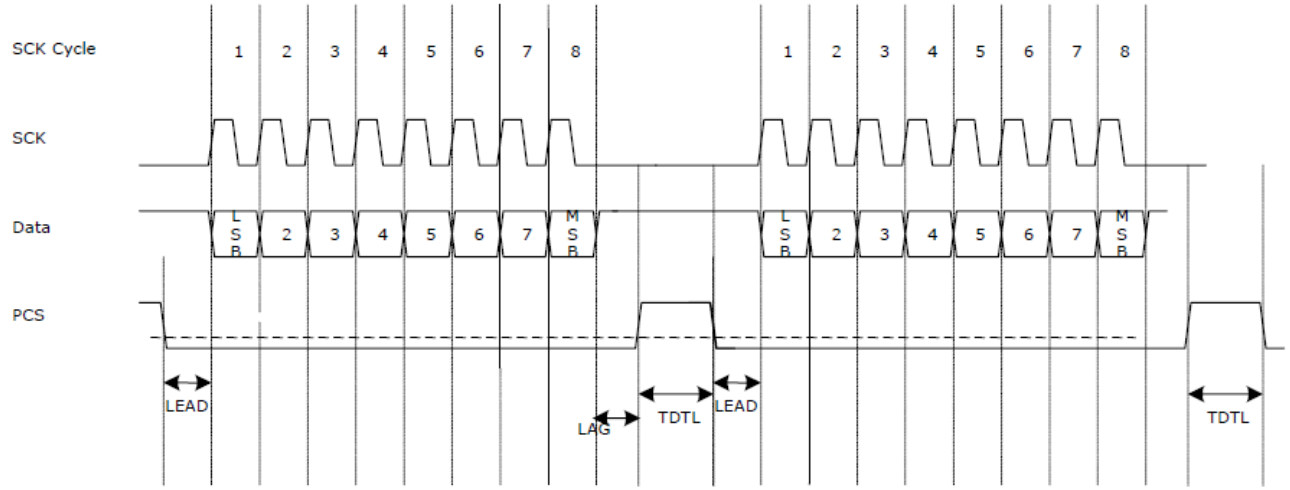
(3)TDTL

The minimum transfer interval can be controlled in SCK clocks depending on the DTL setting of SPBRR.

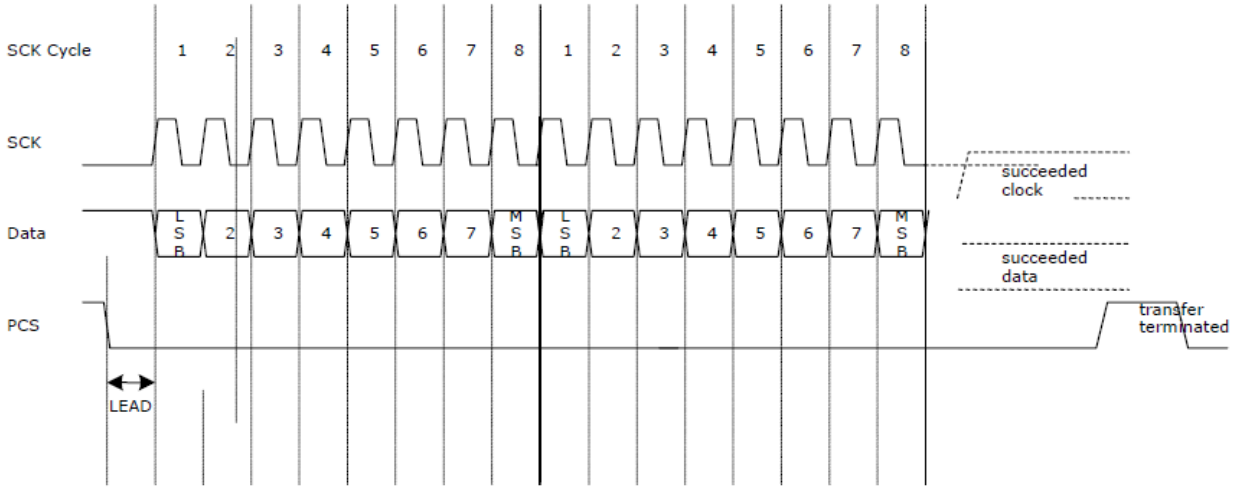
If there is any transfer data in FIFO, the time set by this setting (SSn) changes to H during byte/word transfers.

If there is no transfer data in FIFO, this is H until any transmit data is written.

If DTL is set to 0, the interval after transfer (TDTL) disappears and a continuous transfer is performed. In this case, SSn is held at L and returns to H after the transfer is completed.



Case of DTL disable (continuous transfer)



Note: disable or enable DSCKL do not generate the interval between 8bit data in continuous transfer.

SSn timing

7.8.5 Description of Operation

7.8.5.1 Transmit Operation (Master Mode)

(1) Write the necessary values to SPCR and SPBRR, set the MSTR bit to Master mode, and set the SPE bit to enable the SPI transfer.

(2) When the transmit data is written to SPDWR, the transmit FIFO Empty flag changes to 0 (TFE = 0). SPI starts the automatic transmission and outputs the transmit data from LSB or MSB on the MOSI pin according to the LSBF setting.

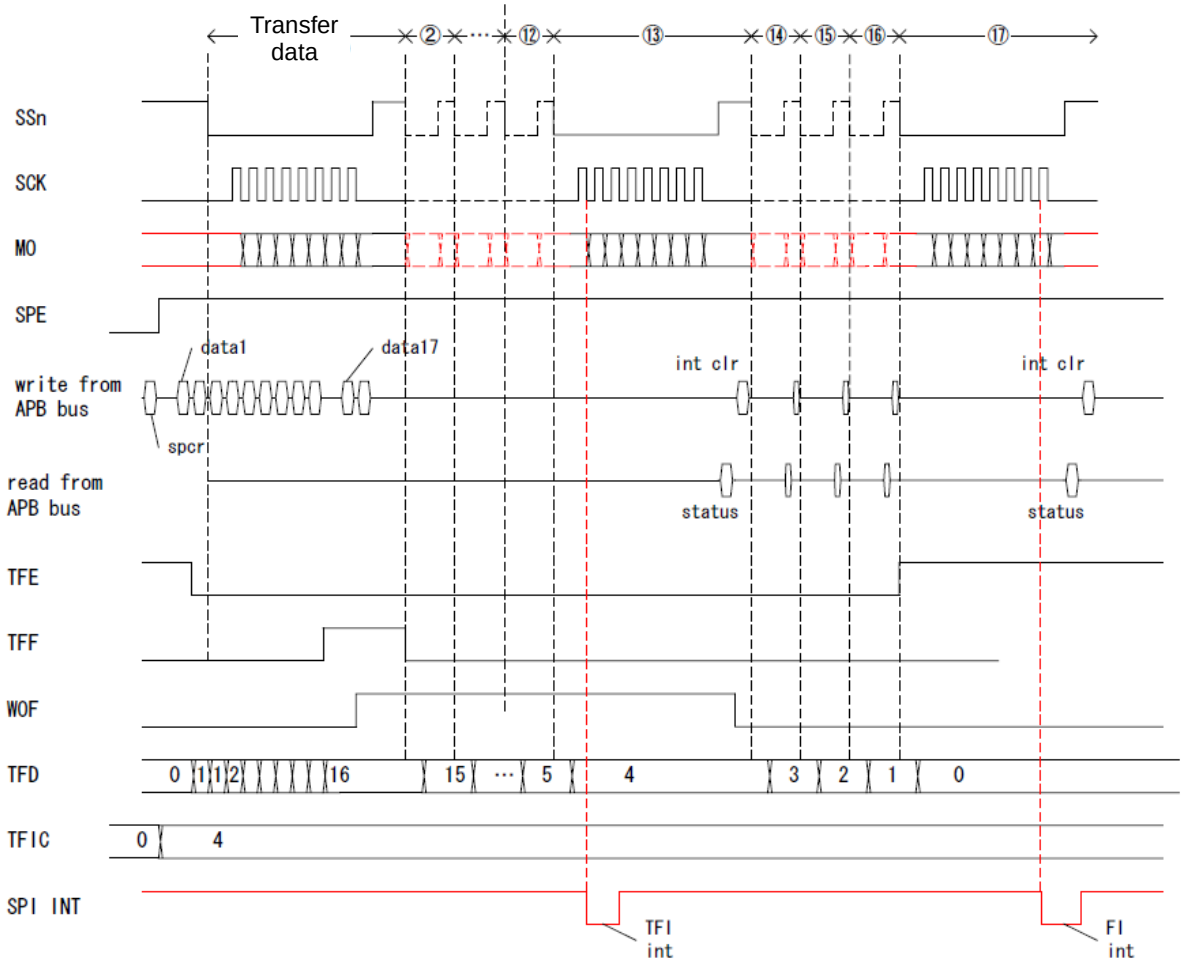
(3) Output the synchronous clock set by using the LSBF, CPOL, CPHA, and SPBRR registers from the SCK pin.

(4) Transmit data can be written to SPDWR successively. However, if further writing is performed when the transmit FIFO is in Full status (TFF = 1), a write overflow occurs. (WOF = 1, No interrupt is generated.)

(5) SPIF bit is set each time the transfer of 8 bits is completed. (SPIF=1)

(6) If the amount of remaining data in the transmit FIFO matches the number of bytes selected with TFIC, a transmission interrupt is generated. (TFI=1)

(7) If the transmit FIFO becomes empty and the transfer of the last byte is completed, a transfer completion interrupt is generated. (FI=1)



Master Mode (Transmit Operation)

7.8.5.2 Receive Operation (Master Mode)

(1) Write the necessary values to SPCR and SPBRR, set the MSTR bit to Master mode, and set the SPE bit to enable the SPI transfer.

(2) When the data is written to SPDWR, the SPI transfer is started.

(3) Output the synchronous clock set by using the LSBF, CPOL, CPHA, and SPBRR registers from the SCK pin.

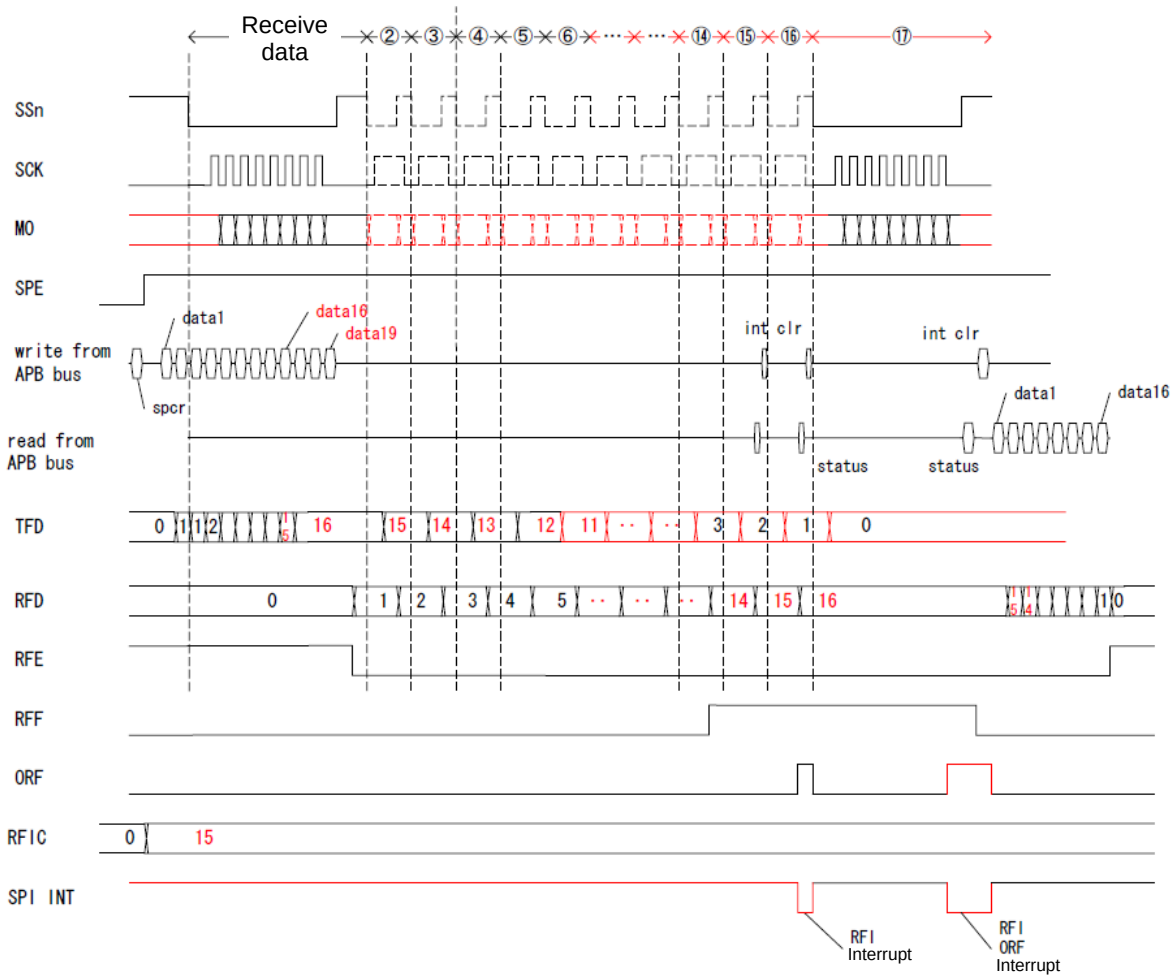
(4) On the MISO pin, the received data is sampled from LSB or MSB according to the LSBF setting and stored in the receive FIFO. The receive FIFO Empty flag changes to 0 (RFE = 0).

(5) SPIF bit is set each time the transfer of 8 bits is completed. (SPIF=1)

(6) If the amount of data received in the receive FIFO exceeds the number of bytes selected with RFIC, a reception interrupt is generated. (RFI=1)

(7) When the receive FIFO becomes Full, the subsequent reception is disabled. If the reception is performed in this state, an overrun error interrupt is generated. (ORF=1)

(8) If the temporary data of transmit FIFO becomes empty and the transfer of the last byte is completed, a transfer completion interrupt is generated. (FI=1)



Master Mode (Receive Operation)

7.8.5.3 FIFO Operation

SPI includes the receive FIFO of 16 bytes (words) and the transmit FIFO of 16 bytes (words). The status of FIFO is indicated by the TFF, TFE, TFD, RFF, RFE, and RFD bits of SPSR.

FIFO has the following three statuses: Full (TFF, RFF), Empty (TFE, RFE), and Depth (TFD, RFD).

7.8.5.4 Write Overflow

If further writing is performed when the transmit FIFO is in Full status (TFF = 1), a write overflow is set. (WOF=1)

However, no interrupt is generated even when a write overflow occurs.

WOF is cleared when SPSR is read.

7.8.5.5 Overrun Error

If further reception is performed when the receive FIFO is in Full status (RFF = 1), an overrun error occurs. (ORF=1)

If an overrun error occurs, the ORF bit of SPSR is set, and an overrun error interrupt is generated.

The newly received data is not held.

Read the content of the receive FIFO and clear the RFF bit, and then write "1" to the ORF bit to clear the ORF bit.

7.8.5.6 FIFOCLR

If this bit is set to 1, the transmit/receive counter control of FIFO returns to the initial setting state.

In the SPSR register, TFF = 0, TFE = 1, TFD = 00000, RFF = 0, RFE = 1, RFD = 00000 are set.

This bit is effective only when SPE = 0 is set.

Set FIFOCLR back to 0 before performing a transfer.

Even when this bit is set to 1, RFIC, TFIC, ORIE, FIE, RFIE, and TFIE of the SPCR register and the ORF, FI, RFI, and TFI interrupts of the SPSR register are not changed.

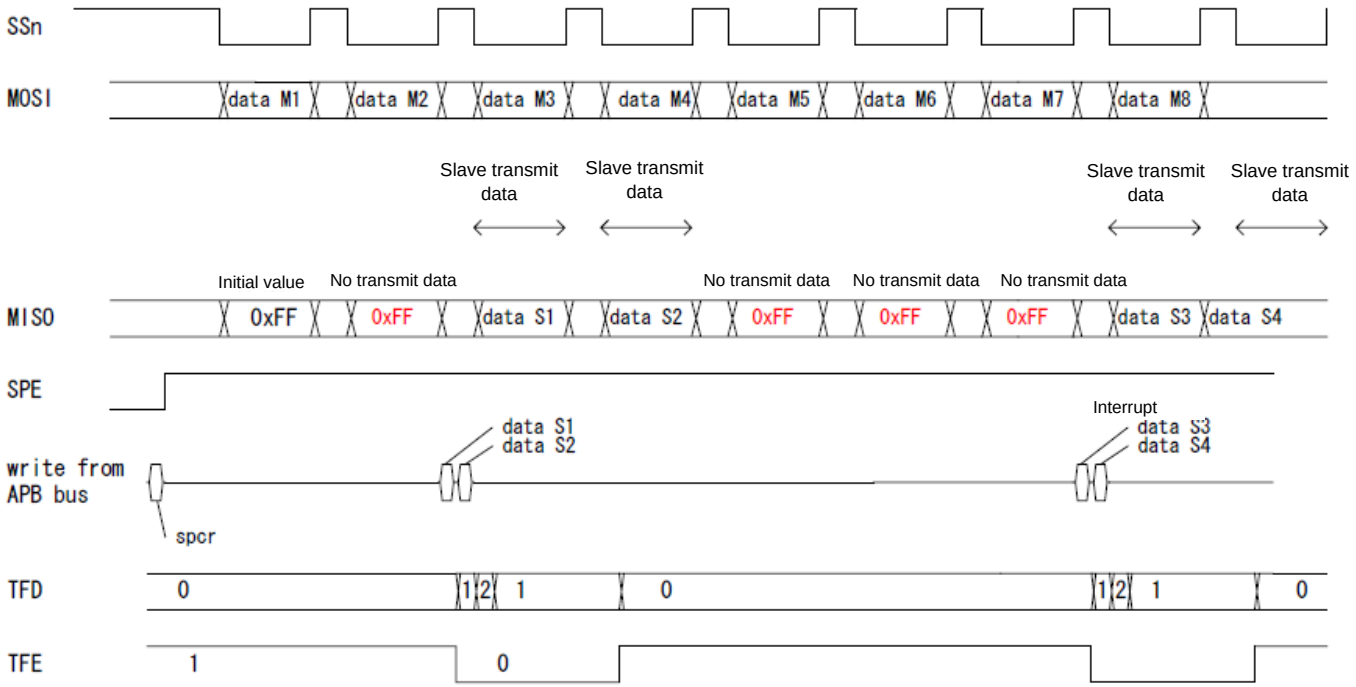
This bit can be used to discard the data of FIFO when the communication is stopped.

7.8.5.7 Transfer When Slave Has Different Number of FIFO Transfer Bytes/Words

(1) The master sends data only when the transmit data is already written in FIFO.

(2) As the transmission of slave is determined by the master, data is transferred as follows if the number of FIFO transfer bytes/words of slave is different from that of the master.

•If no transmit data is written in the slave's FIFO, 0xFF (0xFFFF in the case of words) is sent, including the state after a reset.



Transfer When Slave Has Different Number of FIFO Transfer Bytes/Words

7.8.5.8 Mode Fault (MDF)

A mode fault error occurs if the SSn signal goes into the low level in Master mode (MDF of SPSR is set). If this bit is 1, that means the risk of two or more masters competing for the bus.

When a mode fault error occurs, SPI performs the following operations since there is a risk of bus latch-up:

1. Automatically sets the MSTR bit of SPCR to 0 (slave).
2. Automatically sets the SPE bit of SPCR to 0 (disabled) to make the SPI unable to transfer.
3. Sets MDF of SPSR, and also generates an interrupt if the MDFIE bit of SPCR1 is 1 (interrupt enabled).

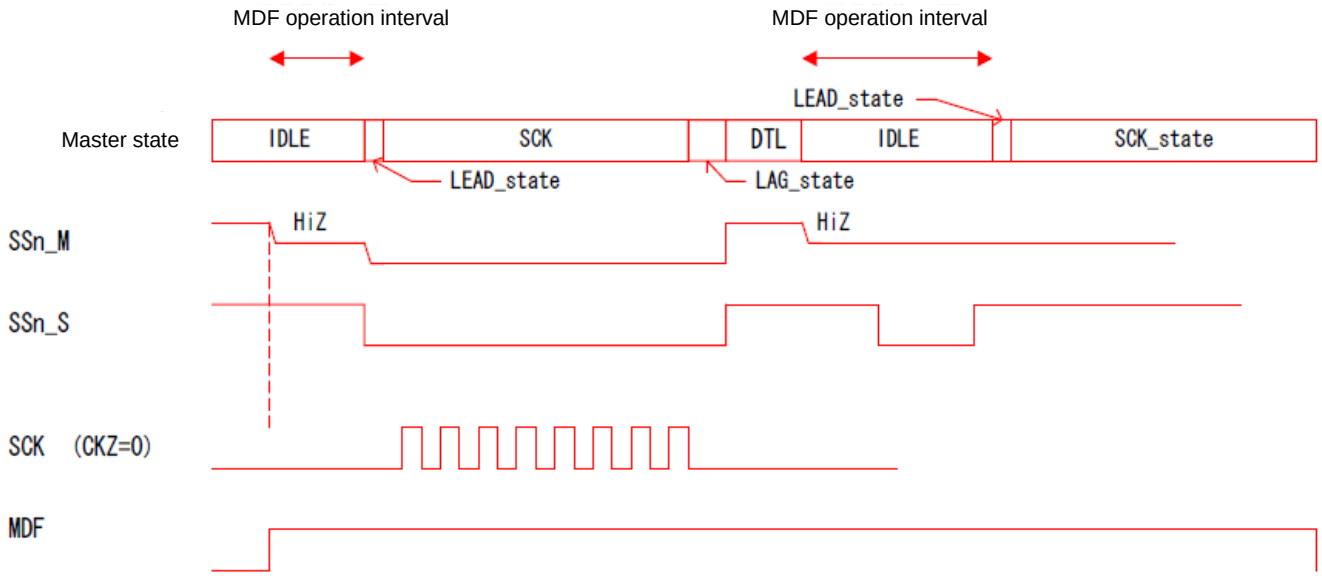
The system should resolve the causes of the mode fault, and then clear MDF according to the following steps:

1. Write 1 to MDF to clear it.

2. Set the correct values in SPCR.

At a mode fault, all outputs become Hi-Z.

The figure below shows the timing that allows a mode fault operation.



Timing That Allows Mode Fault Operation

7.8.5.9 Interrupt Source

7.8.5.9.1 Interrupt Sources of SPI

There are the following five types of interrupt sources:

- Mode fault

If a mode fault (multi-master bus contention) occurs, MDF of SPSR is set and a mode fault interrupt is generated.

- Overrun

If an overrun occurs, ORF of SPSR is set, and an overrun error interrupt is generated.

- Transmit FIFO threshold

If the amount of remaining data in the transmit FIFO matches the number of bytes selected with TFIC, TFI of SPSR is set, and a transmission interrupt is generated.

- Receive FIFO threshold

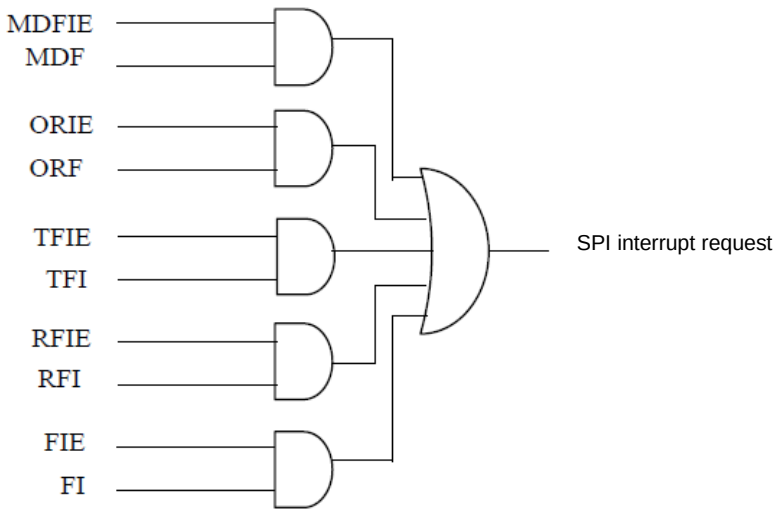
If the amount of data received in the receive FIFO exceeds the number of bytes selected with RFIC of SPCR, RFI of SPSR is set, and a reception interrupt is generated.

- Transfer completed

If the transmit FIFO becomes empty and the transfer of the last byte is completed, FI of SPSR is set, and a transfer completion interrupt is generated.

7.8.5.9.2 Interrupt Clear of SPI

An interrupt request is cleared by writing 1 to the interrupt bits of SPSR (TFI, RFI, MDF, ORF, FI).



SPI Interrupt Signal Logic

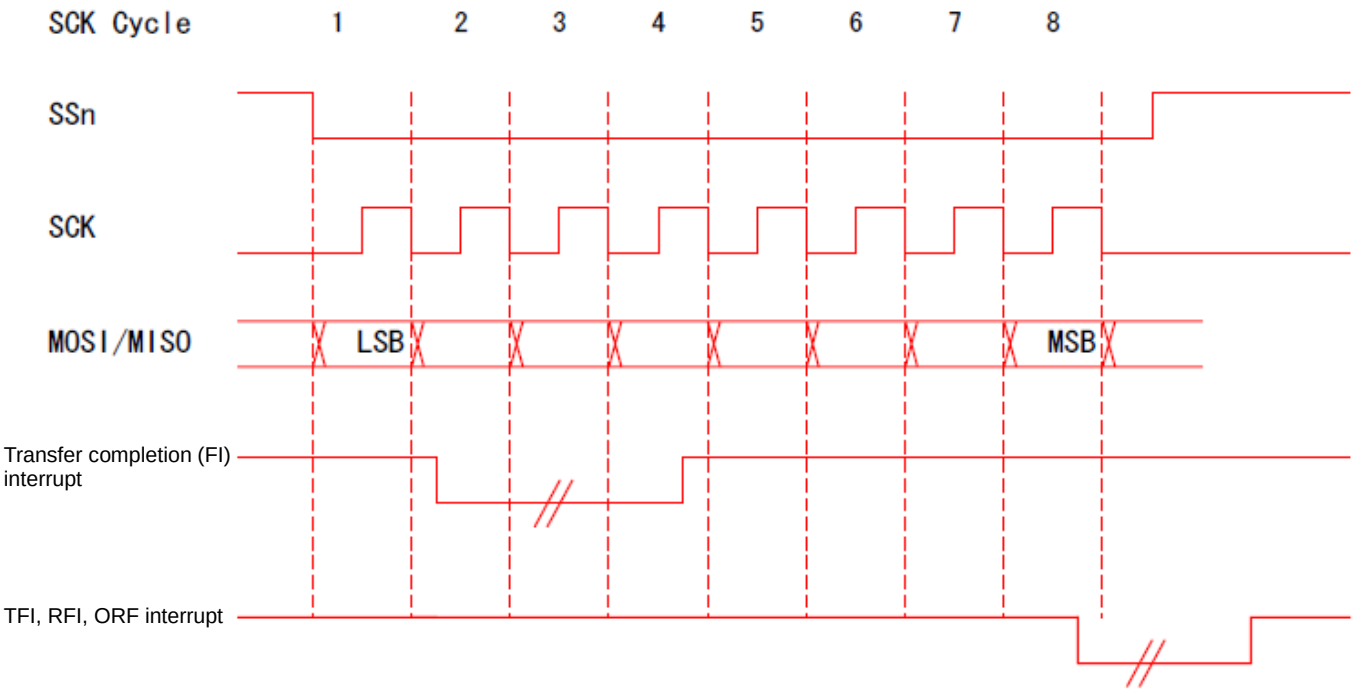
7.8.5.9.3 Interrupt Timing of SPI

Figure below shows the interrupt timing.

The remaining transmit byte count interrupt (TFI) generates an interrupt 1 to 2 SYSCLK after the shift clock of the second bit.

Any receive byte count interrupt (RFI), transfer completion interrupt (FI), or overrun (ORF) generates an interrupt 1 to 2 SYSCLK after the sampling clock of the MSB bit.

MDF generates an interrupt at a mode fault occurrence.



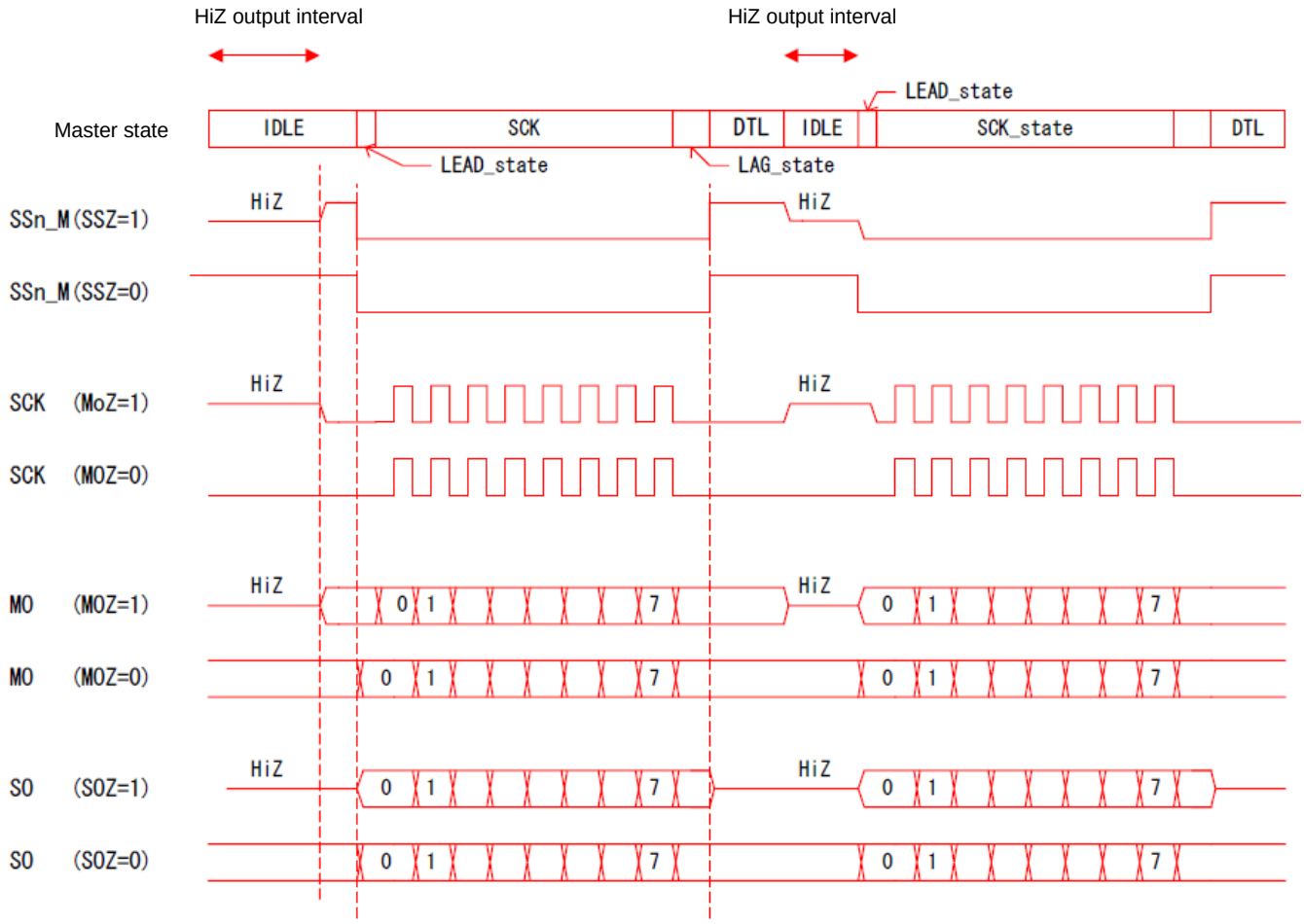
Timing That Allows Mode Fault Operation

7.8.5.10 Hi-Z Operation

The figure below shows an example of using Hi-Z (MOZ, SOZ, SSZ).

The Hi-Z transmit interval of the master is limited to the IDLE time shown below.

To reduce the effect of noise in the Hi-Z state, "1"/"0" is fixed 1SCK before the transmission starts, and "1"/"0" is fixed during the DTL time of the transfer interval.

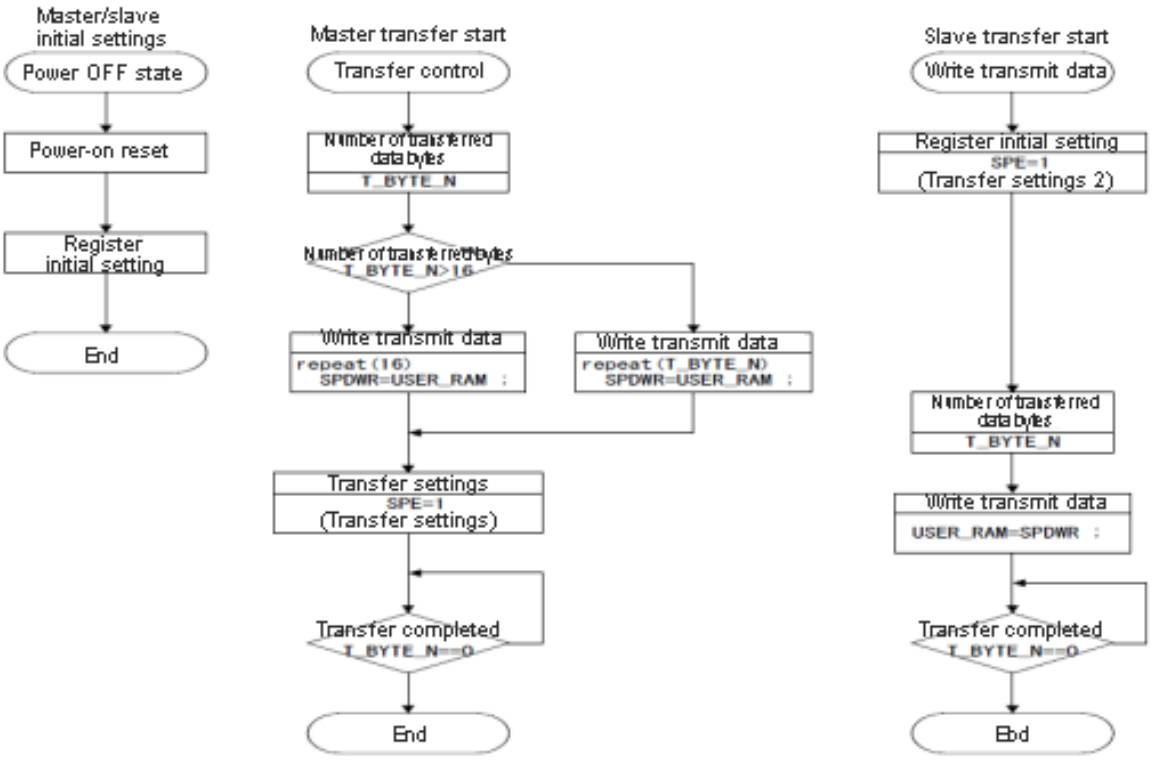


Hi-Z Operation

7.8.5.11 Interval from MSTR Setting to Transfer Start

The SPI bus (MISO, MOSI, SCK, SSn) is in a high impedance state until Master mode is set.

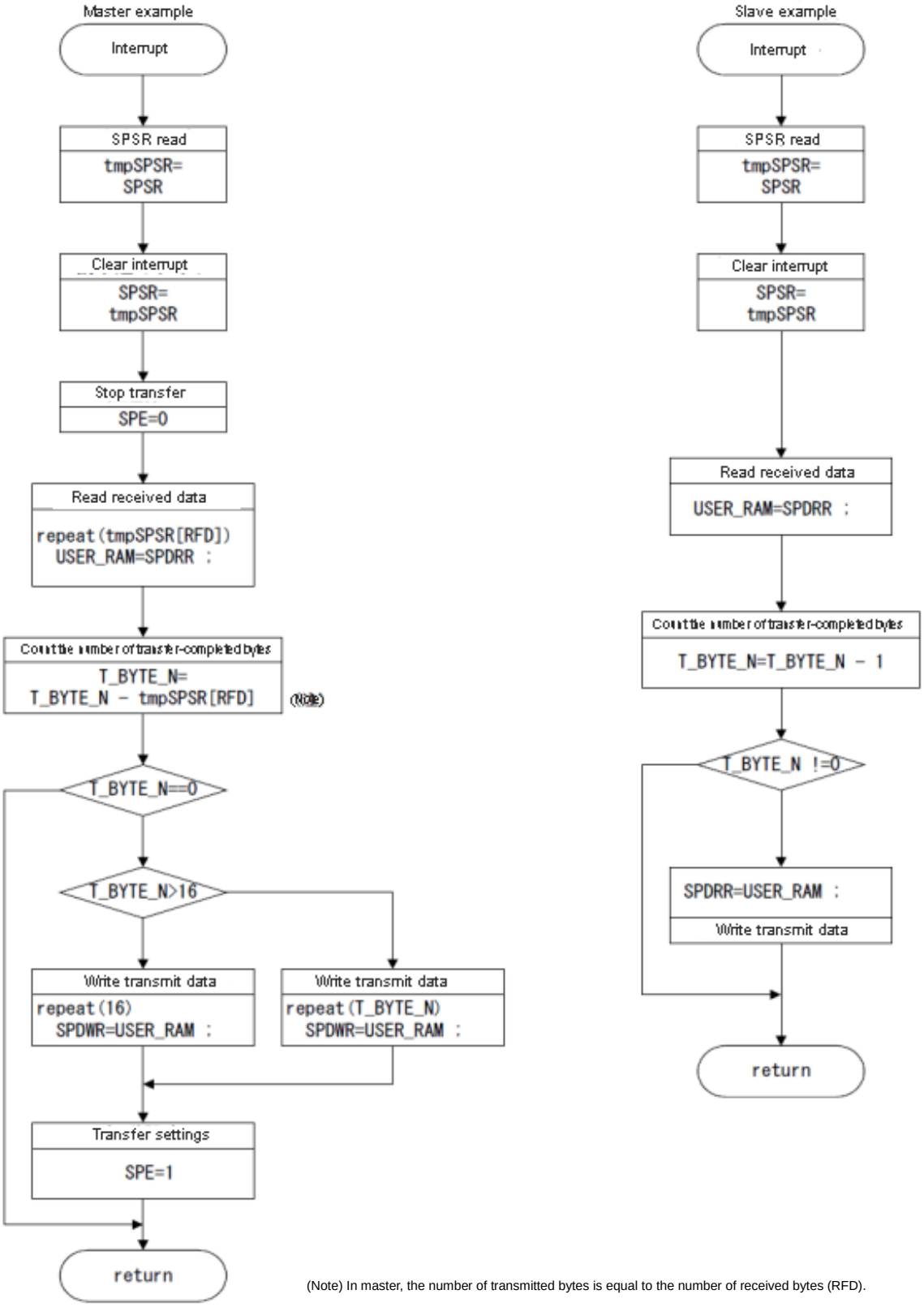
After setting MSTR, wait for at least 100ns before starting the transmission (SPE = 1, or transfer started by data write).



Example of Initial Settings

		Master						Slave	
Register name	Control bit	Initial setting		Transfer settings		Stop Transfer		Initial setting	
SPCP	MOZ	0	Initial value	0	Initial value	0	Initial value	0	Initial value
	SOZ	0	Initial value	0	Initial value	0	Initial value	0	Initial value
	SSZ	0	Initial value	0	Initial value	0	Initial value	0	Initial value
	FICLR	0	Initial value	0	Initial value	0	Initial value	0	Initial value
	RFIC	0	Initial value	0	Initial value	0	Initial value	0	Initial value
	TFIC	0	Initial value	0	Initial value	0	Initial value	0	Initial value
	MDFIE	0	Initial value	0	Initial value	0	Initial value	0	Initial value
	ORIE	0	Initial value	0	Initial value	0	Initial value	0	Initial value
	FIE	0	Initial value	1	Transfer completed	1	Transfer completed	0	Initial value
	RFIE	0	Initial value	0	Initial value	0	Initial value	1	Per 1 byte
	TFIE	0	Initial value	0	Initial value	0	Initial value	0	Initial value
	CPOL	0	Initial value	0	Initial value	0	Initial value	0	Same as master
	CPHA	0	Initial value	0	Initial value	0	Initial value	0	Same as master
	LSBF	0	Initial value	0	Initial value	0	Initial value	0	Same as master
	MODEFEN	0	Initial value	0	Initial value	0	Initial value	0	Initial value
	-	0	Initial value	0	Initial value	0	Initial value	0	Initial value
	MSTR	1	MASTER	1	MASTER	1	MASTER	0	SLAVE
	SPE	0	Initial value	1	Start transmission	0	Stop transmission	1	Communication enabled
SPBRR	DTL	16	Transfer interval 16	Not set when transferring				2	Initial value
	LAG	1	Initial value					1	Initial value
	LEAD	1	Initial value					1	Initial value
	SIZE	0	8bit					0	Same as master
	SPBR	2	8Mbit/s()					2	Initial value

Example of Register Initial



(Note) In master, the number of transmitted bytes is equal to the number of received bytes (RFD).

Example of Interrupt Control Flow

7-9.SSIS(SSI Slave)

7.9.1 General Description

A synchronous serial communication interface (slave). The features are shown below.

- Supports the Motorola SPI.
- Data length: 4 to 16 bits
- 8-stage FIFO for each of transmit and receive

7.9.2 List of Registers

Table List of SSI Registers

Address	Name	Symbol	Size	R/W	Initial value
BASE+0x00	control register 0	SSISn_CTRLR0	32	R/W	0x00000007
BASE+0x04	control register 1	SSISn_CTRLR1	32	R/W	0x00000000
BASE+0x08	SSI enable register	SSISn_SSIENR	32	R/W	0x00000000
BASE+0x0C	Microwire control register	SSISn_MWCR	32	R/W	0x00000000
BASE+0x10	Slave enable register	SSISn_SER	32	R/W	0x00000000
BASE+0x14	Baud rate select register	SSISn_BAUDR	32	R/W	0x00000000
BASE+0x18	Transmit FIFO threshold level register	SSISn_TXFTLR	32	R/W	0x00000000
BASE+0x1C	Receive FIFO threshold level register	SSISn_RXFTLR	32	R/W	0x00000000
BASE+0x20	Transmit FIFO level register	SSISn_TXFLR	32	R	0x00000000
BASE+0x24	Receive FIFO level register	SSISn_RXFLR	32	R	0x00000000
BASE+0x28	status register	SSISn_SR	32	R	0x00000006
BASE+0x2C	Interrupt mask register	SSISn_IMR	32	R/W	0x0000001F
BASE+0x30	Interrupt status register	SSISn_ISR	32	R	0x00000000
BASE+0x34	RAW interrupt status register	SSISn_RISR	32	R	0x00000000
BASE+0x38	Transmit FIFO overflow interrupt clear register	SSISn_TXOICR	32	R	0x00000000
BASE+0x3C	Receive FIFO overflow interrupt clear register	SSISn_RXOICR	32	R	0x00000000
BASE+0x40	Receive FIFO underflow interrupt clear register	SSISn_RXUICR	32	R	0x00000000
BASE+0x44	Multi-master interrupt clear register	SSISn_MSTICR	32	R	0x00000000
BASE+0x48	Interrupt clear register	SSISn_ICR	32	R	0x00000000
BASE+0x58	ID register	SSISn_IDR	32	R	Master 0x00000707 Slave 0x80000707
BASE+0x5C	Version ID register	SSISn_SSI_COMP_VER	32	R	0x3332322A
BASE+0x60-BASE+0xEC	Data register	SSISn_DR	32	R/W	0x00000000
BASE+0xF0	RXD sample delay register	SSISn_RX_SAMPLE_DLY	32	R/W	0x00000000

* n indicates a module number (n = 0). ※ Base address of SSI Slave (n = 0) is 0x4000D000.

7.9.3 Description of Registers

7.9.3.1 Control Register 0 (SSIxn_CTRLR0): BASE + 0x00

This register controls the operation mode of the SSI module.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																CTRLR0															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Description
CTRLR0[15:12]	CFS	R/W	Specifies the control word length in the Microwire frame format. "The setting value of this field + 1" bits is the actual length of the control word.
CTRLR0[11]	SRL	R/W	Register for test. Normally use the initial value as it is. 0: Normal mode 1: Test mode
CTRLR0[10]	SLV_OE	R/W	This bit is effective only in SSI Slave. It controls the data output of the slave. 0: The slave transmit data is enabled. 1: The slave transmit data is disabled.
CTRLR0[9:8]	TMOD	R/W	Sets the transfer mode. 00: Transmit/receive mode 01: Transmission only 10: Reception only 11: Read EEPROM
CTRLR0[7]	SCPOL	R/W	Sets the polarity of the serial clock. 0: The inactivated state is L. 1: The inactivated state is H.
CTRLR0[6]	SCPH	R/W	Sets the phase of the serial clock. 0: The phase is toggled in the middle of the first data transfer. 1: The phase is toggled when starting the first data transfer.

CTRLR0[5:4]	FRF	R/W	Sets the frame format. 00: Motorola SPI 01: TI SSP 10: National Semiconductor Microwire 11: Reserved
CTRLR0[3:0]	DFS	R/W	Sets the data frame length. "The setting value of this field + 1" is the data length. The minimum settable value is 0011 (data length 4 bits), and the maximum settable value is 1111 (data length 16 bits).

7.9.3.2 Control Register 1 (SSIxn_CTRLR1): BASE + 0x04

This register controls the operation mode of the SSI module.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																CTRLR1															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Description
CTRLR1[15:0]	NDF	R/W	Sets the number of data frames to be continuously received when TMOD = 10 or TMOD = 11."The setting value of this field + 1" is the number of frames.

7.9.3.3 SSI Enable Register (SSIxn_SSIENR): BASE + 0x08

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															SSI_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Bit	Name	R/W	Description
SSIENR[0]	SSI_EN	R/W	Enable the SSI module. When SSI is disabled, the transfer is stopped immediately and the transmit FIFO and receive FIFO are cleared. The control register cannot be rewritten when SSI is enabled. 0: Disabled 1: Enabled

7.9.3.4 Microwire Control Registers (SSIxn_MWCR): BASE + 0x0C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													MWCR [2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

Bit	Name	R/W	Description
MWCR[2]	MHS	RW	* This bit is effective only in SSI Master. It enables the handshake that is based on the busy/ready status of Microwire. 0: The handshake is disabled. 1: The handshake is enabled.
MWCR[1]	MDD	RW	Sets the data transfer direction when Microwire is set. 0: Reception 1: Transmission
MWCR[0]	MWMOD	RW	Sets the Microwire transfer mode. 0: Non-sequential transfer 1: Sequential transfer

7.9.3.5 Slave Enable Register (SSIxn_SER): BASE + 0x10

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															SER
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

Bit	Name	R/W	Description
SER[0]	SER	RW	* This bit is effective only in SSI Master. Asserts the corresponding slave select signal. 1: Selected 0: Not selected

* Ensure that this bit is set when the master operation is started.

7.9.3.6 Baud Rate Select Register (SSIxn_BAUDR): BASE + 0x14

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																SCKDV															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R

Bit	Name	R/W	Description
BAUDR	SCKDV	RW	* This bit is effective only in SSI Master. The value of the frequency divider of SSI clock. LSB is always 0, which means writing is invalid. The communication clock cycle can be calculated by the following equation: SSI communication clock cycle = Input communication clock frequency/SCKDV SCKDV can be set to an even number from 2 to 65534.

7.9.3.7 Transmit FIFO Threshold Level Register (SSIxn_TXFTLR): BASE + 0x18

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													TXFTLR		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

Bit	Name	R/W	Description
TXFTLR[2:0]	TFT	RW	Sets the threshold of transmit FIFO. If the amount of data in the transmit FIFO is smaller than or equal to the setting value of this field, a transmit FIFO Empty interrupt occurs. 000: An interrupt occurs if the amount is smaller than or equal to 0 data. 001: An interrupt occurs if the amount is smaller than or equal to 1 data. : 111: An interrupt occurs if the amount is smaller than or equal to 7 data.

7.9.3.8 Receive FIFO Threshold Level Register (SSIxn_RXFTLR): BASE + 0x1C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													RXFTLR		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Bit	Name	R/W	Description
RXFTLR[2:0]	RFT	RW	Sets the threshold of the receive FIFO. If the amount of data in the receive FIFO is greater than or equal to the setting value of this field, a receive FIFO Full interrupt occurs. 000: An interrupt occurs if the amount is greater than or equal to 1 data. 001: An interrupt occurs if the amount is greater than or equal to 2 data. : 111: An interrupt occurs if the amount is greater than or equal to 8 data.

7.9.3.9 Transmit FIFO Level Register (SSIxn_TXFLR): BASE + 0x20

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													TXFLR		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
TXFLR[2:0]	TXTFL	R	Indicates the number of valid data within the transmit FIFO.

7.9.3.10 Receive FIFO Level Register (SSIxn_RXFLR): BASE + 0x24

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													RXTFL		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
RXFLR[2:0]	RXTFL	R	Indicates the number of valid data within the receive FIFO.

7.9.3.11 Status Register (SSIxn_SR): BASE + 0x28

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																									SR[6:0]						
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
SR[6]	DCOL	R	Data collision error. This bit is effective only in Master mode. This bit is set if this device is selected as a slave by another master device while this device is operating as the master device. This bit indicates that the last data transfer is stopped before it is completed. This bit is cleared when it is read. 0: No error 1: A data collision error occurred
SR[5]	TXE	R	Transmit error. This bit is set if the transmit FIFO attempts transmission when it is empty. This bit is effective only in Slave mode. This bit is cleared when it is read. 0: No error 1: A transmit error occurred.

SR[4]	RFF	R	Receive FIFO Full. This bit is set when the receive FIFO is full. When the receive FIFO becomes non-full, this bit is cleared. 0: not Full 1: Full
SR[3]	RFNE	R	Receive FIFO not empty. This bit is set when there is one or more data in the receive FIFO, and cleared when the receive FIFO becomes empty. 0: Receive FIFO empty 1: Receive FIFO not empty
SR[2]	TFE	R	Transmit FIFO Empty. This bit is set when the transmit FIFO is empty. This bit is cleared when there is one or more valid data. This bit does not cause an interrupt. 0: Transmit FIFO not empty 1: Transmit FIFO empty
SR[1]	TFNF	R	Transmit FIFO not full. This bit is set when there is one or more data in the transmit FIFO, and cleared when the transmit FIFO becomes empty. 0: Transmit FIFO full 1: Transmit FIFO not full
SR[0]	BUSY	R	SSI busy. This bit is set during the serial transfer. This bit is cleared at IDLE or SSI Disable. 0: SSI module is in IDLE or Disable status 1: Serial transferring

7.9.3.12 Interrupt Mask Register (SSIxn_IMR): BASE + 0x2C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																										IMR[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Description
IMR[5]	MSTIM	RW	* This bit is effective only in SSI Master. Always 0 is read in SSI Slave. Multi-master contention interrupt mask 0: Mask/1: Non-mask
IMR[4]	RXFIM	RW	Receive FIFO Full interrupt mask 0: Mask/1: Non-mask
IMR[3]	RXOIM	RW	Receive FIFO overflow interrupt mask 0: Mask/1: Non-mask
IMR[2]	RXUIM	RW	Receive FIFO underflow interrupt mask 0: Mask/1: Non-mask
IMR[1]	TXOIM	RW	Transmit FIFO overflow interrupt mask 0: Mask/1: Non-mask
IMR[0]	TXEIM	RW	Transmit FIFO empty interrupt mask 0: Mask/1: Non-mask

This bit is masked when 0 is written to it.

7.9.3.13 Interrupt Status Register (SSIxn_ISR): BASE + 0x30

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										ISR[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW

Indicates the interrupt status after mask.

Bit	Name	R/W	Description
ISR[5]	MSTIS	RW	* This bit is effective only in SSI Master. Multi-master contention interrupt
ISR[4]	RXFIS	RW	Receive FIFO full interrupt
ISR[3]	RXOIS	RW	Receive FIFO overflow interrupt
ISR[2]	RXUIS	RW	Receive FIFO underflow interrupt
ISR[1]	TXOIS	RW	Transmit FIFO overflow interrupt
ISR[0]	TXEIS	RW	Transmit FIFO empty interrupt

7.9.3.14 RAW Interrupt Status Register (SSIxn_RISR): BASE + 0x34

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										RISR[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Indicates the interrupt status before mask.

Bit	Name	R/W	Description
RISR[5]	MSTIR	R	* This bit is effective only in SSI Master. Multi-master contention RAW interrupt
RISR[4]	TXFIR	R	Receive FIFO full RAW interrupt
RISR[3]	TXOIR	R	Receive FIFO overflow RAW interrupt
RISR[2]	RXUIR	R	Receive FIFO underflow RAW interrupt
RISR[1]	TXOIR	R	Transmit FIFO overflow RAW interrupt
RISR[0]	TXEIR	R	Transmit FIFO empty RAW interrupt

7.9.3.15 Transmit FIFO Overflow Interrupt Clear Register (SSIxn_TXOICR): BASE + 0x38

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															TXOICR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
TXOICR[0]	TXOICR	R	When this register is read, the transmit FIFO overflow interrupt is cleared. Writing is ignored.

7.9.3.16 Receive FIFO Overflow Interrupt Clear Register (SSIxn_RXOICR): BASE + 0x3C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															RXOICR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
RXOICR[0]	RXOICR	R	When this register is read, the receive FIFO overflow interrupt is cleared. Writing is ignored..

7.9.3.17 Receive FIFO Underflow Interrupt Clear Register (SSIxn_RXUICR): BASE + 0x40

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															RXUICR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
RXUICR [0]	RXUICR	R	When this register is read, the receive FIFO underflow interrupt is cleared. Writing is ignored.

7.9.3.18 Multi-master Interrupt Clear Register (SSIxn_MSTICR): BASE + 0x44

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															MSTICR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
MSTICR [0]	MSTICR	R	When this register is read, the multi-master contention interrupt is cleared. Writing is ignored.

7.9.3.19 Interrupt Clear Register (SSIxn_ICR): BASE + 0x48

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															ICR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
ICR[0]	ICR	R	When this register is read, all active interrupts are cleared. Writing is ignored.

7.9.3.20 ID Register (SSIxn_IDR): BASE + 0x58

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IDCODE																															
Initial value	Refer to the description below.																															
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
IDR[31:0]	IDCODE	R	User-defined ID codes. This register indicates the configuration of SSI. Bit[31]=0:Master, 1:Slave. Bit:[30:25] = Reserved. The value is 0. Bit:[24:20] = Peripheral numbers. 0-31. Bit:[19:16] = Number of SSI signals (SSI_NUM_SLAVES). The values 0 to 15 correspond to numbers of signals 1 to 16. Bit:[15:8] = Depth of transmit FIFO. The values 1 to 255 correspond to depths of FIFO 2 to 256. Bit:[7:0] = Depth of receive FIFO. The values 1 to 255 correspond to depths of FIFO 2 to 256. Initial value Master:0x00000707 Slave:0x80000707

7.9.3.21 Version ID Register (SSIxn_SSI_CMP_VERSION): BASE + 0x5C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SS_CMP_VERSION																															
Initial value	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	1	0	1	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description
SSI_CMP_VERSION [31:0]	SSI_CMP_VERSION	R	Component version

7.9.3.22 Data Register (SSIxn_DR): BASE + 0x60 to 0xEC

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																DR															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Description
DR [15:0]	DR	R/W	Read: Receive FIFO buffer Write: Transmit FIFO buffer

7.9.3.23 RX Sample Delay Register (SSIxn_RX_SAMPLE_DLY): BASE + 0xF0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								RSD							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

※This register is effective only in SSI Master. In SSI Slave, write is ignored and read returns 0x00000000.

Bit	Name	R/W	Description
RX_SAMPLE_DLY [7:0]	RSD	R/W	Received data sample delay If set to 5 or larger, the delay is 0.

7-10.WDT

7.10.1 General Description

Programmable 16-bit watchdog timer. The features are shown below.

- When the counter reaches its timeout, this timer asserts an interrupt at the first timeout, and performs the system reset operation at the second timeout. The mode of asserting only an interrupt without reset operation can be set. For mode setting, refer to "* 7.29.3.1 WDT Setting Register".
- The SYSCON_PPM1 and SYSCON_PPM2 registers can be used to select whether the counter operates or stops in SLEEP mode.
- The source clock of the watchdog timer can be selected using the control register.

7.10.2 List of Registers

Address [H]	Name	Symbol	R/W	Initial value [H]	Description
0x40010000	WDT control register	WDT_CR	RW	0x00000002	
0x40010004	WDT timer timeout range register	WDT_TORR	RW	0x0000000F	
0x40010008	WDT current count value register	WDT_CCVR	R	0x007FFFFF	
0x4001000C	WDT counter resume register	WDT_CRR	W	0x00000000	
0x40010010	WDT interrupt status register	WDT_STAT	R	0x00000000	
0x40010014	WDT interrupt clear register	WDT_EOI	R	0x00000000	
0x400100E4	WDT Configuration Parameter Register 5	WDT_COMP_PARAMS_5	R	0x007FFFFF	
0x400100E8	WDT Configuration Parameter Register 4	WDT_COMP_PARAMS_4	R	0x00000000	
0x400100EC	WDT Configuration Parameter Register 3	WDT_COMP_PARAMS_3	R	0x0000000F	
0x400100F0	WDT Configuration Parameter Register 2	WDT_COMP_PARAMS_2	R	0x007FFFFF	
0x400100F4	WDT Configuration Parameter Register 1	WDT_COMP_PARAMS_1	R	0x100F0212	
0x400100F8	Component version register	WDT_CMP_VERSION	R	0x3130372A	
0x400100FC	Component type register	WDT_COMP_TYPE	R	0x44570120	

7.10.3 Description of Registers

7.10.3.1 WDT control register: 0x40010000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																											RPL			RMOD	WDT_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW

RPL	Sets the reset pulse width. Normally the initial value should be used.
RMOD	Sets the operation when a counter timeout occurs. 0: Resets the system. 1: An interrupt is asserted at the first timeout. If the interrupt source is not cleared until the second timeout, the system is reset.
WDT_EN	Enable the watchdog timer. 0: The watchdog timer is disabled. 1: The watchdog timer is enabled.

* Bit 5 is a reserved R/W bit. When bit 6 to 31 are read, always 0 is read.

7.10.3.2 WDT Timeout Range Register: 0x40010004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												TOP			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

TOP	Sets the counter value when the counter operation is resumed. The relationship between the setting value and the counter initial value is as follows: TOP: Counter initial value 0x0: 0x0000_00ff 0x1: 0x0000_01ff 0x2: 0x0000_03ff 0x3: 0x0000_07ff 0x4: 0x0000_0fff 0x5: 0x0000_1fff 0x6: 0x0000_3fff 0x7: 0x0000_7fff 0x8: 0x0000_ffff 0x9: 0x0001_ffff 0xa: 0x0003_ffff 0xb: 0x0007_ffff 0xc: 0x000f_ffff 0xd: 0x001f_ffff 0xe: 0x003f_ffff 0xf: 0x007f_ffff

7.10.3.3 WDT Current Counter Value Register: 0x40010008

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDT_CCVR																															
Initial value	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

WDT_CCVR	Indicates the value of internal counter.

7.10.3.4 WDT counter resume register: 0x4001000C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								WDT_CCR							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

WDT_CCR	Used to resume WDT. Write 0x76 to this register to resume WDT. When WDT is resumed, the interrupt is cleared. Always 0 is read when reading.

7.10.3.5 WDT interrupt status register: 0x40010010

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															WDT_STAT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

WDT_STAT	Indicates the interrupt status. 1: Indicates that an interrupt has been generated. 0: Indicates that no interrupt has been generated.

7.10.3.6 WDT interrupt clear register: 0x40010014

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															WDT_EOI
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

WDT_EOI	Used to clear an interrupt without resuming the watchdog timer. This can be used for 1-shot timer.

7-11.Port Configuration

7.11.1 General Description

- Selects the function of each general-purpose port (Pmn: m = A, B, n = 0, 1, 2, .., 15).

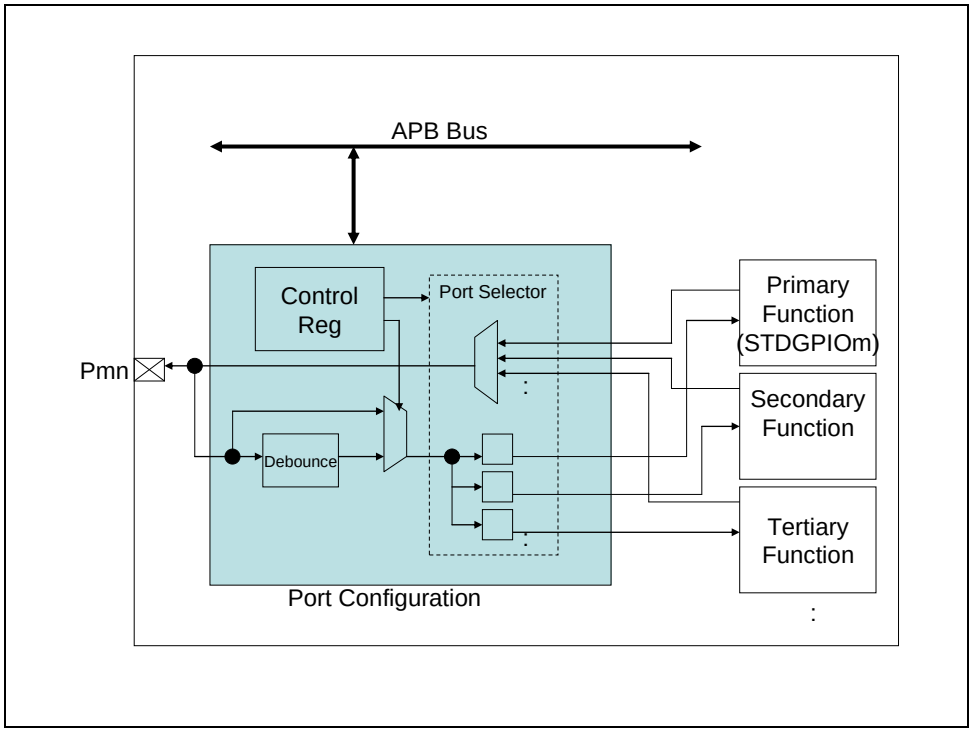


Figure 7-11 Port Configuration Schematic Diagram

7.11.2 List of Registers

Address	Name	Symbol	R/W	Description
BASE+0x00	Port m secondary function select register	STDPORTm_SEL2	R/W	Enables the secondary function of GPIOm (general-purpose port m).
BASE+0x04	Port m tertiary function select register	STDPORTm_SEL3	R/W	Enables the tertiary function of GPIOm (general-purpose port m).
BASE+0x08	Port m quartic function select register	STDPORTm_SEL4	R/W	Enables the quartic function of GPIOm (general-purpose port m).
BASE+0x0C	Port m quintic function select register	STDPORTm_SEL5	R/W	Enables the quintic function of GPIOm (general-purpose port m).
0x40018010	Test control register	TESTCON	R/W	Enables the test mode such as BER measurement.
0x40018014	RF chip wiring function selection register	INTRPORT_SEL	R/W	Enables the RF interface function.

* m indicates a module number (m = A, B).

  m=A: BASE=0x40018000; m=B: BASE=0x40018100

* 0x40018014 (RF chip wiring function selection register) is a reserved register and should be used with the initial value.

7.11.3 Description of Registers

7.11.3.1 Port m Secondary Function Select Register (STDPORTm_SEL2): BASE + 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																PORTm_SEL2															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PORTm_SEL2[n]: 0: Does not set the secondary function to the source of Pmn (general-purpose port m, bit n).

1: Sets the secondary function to the source of Pmn (general-purpose port m, bit n).

* When m = A, bits 00 to 12 are enabled.

7.11.3.2 Port m Tertiary Function Select Register (STDPORTm_SEL3): BASE + 0x04

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																PORTm_SEL3															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PORTm_SEL3[n]: 0: Does not set the tertiary function to the source of Pmn (general-purpose port m, bit n).

1: Sets the tertiary function to the source of Pmn (general-purpose port m, bit n).

* When m = A, bits 00 to 12 are enabled.

7.11.3.3 Port m Quartic Function Select Register (STDPORTm_SEL4): BASE + 0x08

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																PORTm_SEL4															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PORTm_SEL4[n]: 0: Does not set the quartic function to the source of Pmn (general-purpose port m, bit n).

1: Sets the quartic function to the source of Pmn (general-purpose port m, bit n).

* When m = A, bits 00 to 12 are enabled.

7.11.3.4 Port m Quintic Function Select Register (STDPORTm_SEL5): BASE + 0x0C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																PORTm_SEL5															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PORTm_SEL5[n]: 0: Does not set the quintic function to the source of Pmn (general-purpose port m, bit n).

1: Sets the quintic function to the source of Pmn (general-purpose port m, bit n).

* When m = A, bits 00 to 12 are enabled.

7.11.3.5 Test Control Register (TESTCON) : 0x40018010

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																Reserved											RFMON_EN	Reserved				BER_MODE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

RFMON_EN	DMON pin input is written to GPIOA3 0: Normal mode 1: DMON pin input is written to GPIOA3
BER_MODE	BER measurement mode enable 0: Normal mode 1: Switches GPIOA1 and GPIOA2 to BER measurement mode When this bit is set to 1, GPIOA1 and GPIOA2 are assigned to DCLK and DIO outputs respectively.

7.11.4 Description of Operation

7-11.4.1 Function Selection

The functions assigned to the pins of general-purpose ports, GPIOAn, GPIOBn (n = 0, 1, .., 15) are selected.

The function of STDGPIOm[n] (m = A, B/n = 0 .. 15) is determined by the following conditions:

Table 7-11-1 General-purpose Port (GPIOmn) Function Selection Conditions

Conditions	Feature	Remarks
PORTm_SEL2[n]=0 and PORTm_SEL3[n]=0 and PORTm_SEL4[n]=0 and PORTm_SEL5[n]=0	Primary function	Initial value
PORTm_SEL2[n]=1	Secondary function	
PORTm_SEL2[n]=0 and PORTm_SEL3[n]=1	Tertiary function	
PORTm_SEL2[n]=0 and PORTm_SEL3[n]=0 and PORTm_SEL4[n]=1	Quartic function	
PORTm_SEL2[n]=0 and PORTm_SEL3[n]=0 and PORTm_SEL4[n]=0 and PORTm_SEL5[n]=1	Quintic function	

7.11.5 General-purpose Port Setting

Table 7-11-2 List of GPIO Functions

Port	Primary function	Secondary function	Tertiary function	Quartic function	Quintic function
GPIOA[0]	General-purpose port	UART RXD	SPI SCK	I2C SCL	STD GPIO
GPIOA[1]	General-purpose port	UART TXD	SPI SSN	I2C SDA	STD GPIO
GPIOA[2]	General-purpose port	UART CTS	SPI MISO	FTM	STD GPIO
GPIOA[3]	General-purpose port	UART RTS	SPI MOSI	Reserved	STD GPIO
GPIOA[4]	General-purpose port	UART RXD	SSI Slave SCK	I2C SCL	STD GPIO
GPIOA[5]	General-purpose port	UART TXD	SSI Slave SSn	I2C SDA	STD GPIO
GPIOA[6]	General-purpose port	UART CTS	SSI Slave TXD	FTM	STD GPIO
GPIOA[7]	General-purpose port	UART RTS	SSI Slave RXD	Reserved	STD GPIO
GPIOA[8]	General-purpose port	UART RXD	SPI SCK	I2C SCL	STD GPIO
GPIOA[9]	General-purpose port	UART TXD	SPI SSN	I2C SDA	STD GPIO
GPIOA[10]	General-purpose port	UART CTS	SPI MISO	FTM	STD GPIO
GPIOA[11]	General-purpose port	UART RTS	SPI MOSI	Reserved	STD GPIO
GPIOA[12]	General-purpose port	Reserved	Reserved	Reserved	Reserved
GPIOD[0]	General-purpose port				

* The yellow cells indicate the functions (port) used by ISP(In-System Programming).

* The purple cells indicate functions used when an RF interrupt is detected (they do not exist as LSI ports).

In ML7416, several functions have output ports more than the number of blocks of the internal peripherals.

Peripheral name	Number of blocks	Number of GPIO output ports
UART	3	7
SPI	2(*1)	6
I2C	1	7
FTM	1	7
STD GPIO	4	28

(*1) SPI (SPI2) connected to ML7396B is not assigned to GPIO, but output from the dedicated port.

For these functions, the assignment is automatically changed depending on whether GPIO is enabled or not.

The assignment method is as follows:

1. If a pin has any standard application block, that block is always applied when GPIO is enabled.
2. If a pin does not have any standard application block, the remaining block is applied depending on the enable status of other pins.
   If there are multiple remaining blocks, a block with the smallest number is applied first.
   If there is no remaining block, nothing is assigned even if the GPIO function is enabled.

For details of combinations, refer to the table below. In the table, the blocks indicated with (*2) are assigned to the general-purpose port (APB GPIO).

STD GPIO is assigned as shown below. * indicates A or B.

Only one of GPIO*[0], [4], and [8] can be selected.

Only one of GPIO*[1], [5], and [9] can be selected.

Only one of GPIO*[2], [5], and [10] can be selected.

Only one of GPIO*[3], [6], and [11] can be selected.

○UART

Enable conditions							GPIOB secondary (HW primary)				GPIOA secondary (HW primary)		
B[15:12]	B[11: 8]	B[7: 4]	B[3: 0]	A[11: 8]	A[7: 4]	A[3: 0]	[15:12]	[11: 8]	[7: 4]	[3: 0]	[11: 8]	[7: 4]	[3: 0]
0	0	0	0	0	0	0							
0	0	0	0	0	0	1							UART0
0	0	0	0	0	1	0						UART0	
0	0	0	0	0	1	1						UART1	UART0
0	0	0	0	1	0	0					UART0		
0	0	0	0	1	0	1					UART0		UART1
0	0	0	0	1	1	0					UART0	UART1	
0	0	0	0	1	1	1					UART0	UART2	UART1
0	0	0	1	0	0	0				UART0			
0	0	0	1	0	0	1				UART1			UART0
0	0	0	1	0	1	0				UART1		UART0	
0	0	0	1	0	1	1				UART2		UART1	UART0
0	0	0	1	1	0	0				UART1	UART0		
0	0	0	1	1	0	1				UART2	UART0		UART1
0	0	0	1	1	1	0				UART2	UART0	UART1	
0	0	0	1	1	1	1				-(*2)	UART0	UART2	UART1
0	0	1	0	0	0	0			UART0				
0	0	1	0	0	0	1			UART1				UART0
0	0	1	0	0	1	0			UART1			UART0	
0	0	1	0	0	1	1			UART2			UART1	UART0
0	0	1	0	1	0	0			UART1		UART0		
0	0	1	0	1	0	1			UART2		UART0		UART1
0	0	1	0	1	1	0			UART2		UART0	UART1	
0	0	1	0	1	1	1			-(*2)		UART0	UART2	UART1
0	0	1	1	0	0	0			UART1	UART0			
0	0	1	1	0	0	1			UART2	UART1			UART0
0	0	1	1	0	1	0			UART2	UART1		UART0	
0	0	1	1	0	1	1			-(*2)	UART2		UART1	UART0
0	0	1	1	1	0	0			UART2	UART1	UART0		
0	0	1	1	1	0	1			-(*2)	UART2	UART0		UART1
0	0	1	1	1	1	0			-(*2)	UART2	UART0	UART1	
0	0	1	1	1	1	1			-(*2)	-(*2)	UART0	UART2	UART1
0	1	0	0	0	0	0		UART0					
0	1	0	0	0	0	1		UART1					UART0
0	1	0	0	0	1	0		UART1				UART0	
0	1	0	0	0	1	1		UART2				UART1	UART0
0	1	0	0	1	0	0		UART1			UART0		
0	1	0	0	1	0	1		UART2			UART0		UART1
0	1	0	0	1	1	0		UART2			UART0	UART1	
0	1	0	0	1	1	1		-(*2)			UART0	UART2	UART1
0	1	0	1	0	0	0		UART1		UART0			
0	1	0	1	0	0	1		UART2		UART0			UART0
0	1	0	1	0	1	0		UART2		UART0		UART1	
0	1	0	1	0	1	1		-(*2)		UART0		UART1	UART0
0	1	0	1	1	0	0		UART2		UART0	UART1		
0	1	0	1	1	0	1		-(*2)		UART0	UART1		UART1
0	1	0	1	1	1	0		-(*2)		UART0	UART1	UART2	
0	1	0	1	1	1	1		-(*2)		-(*2)	UART0	UART2	UART1
0	1	1	0	0	0	0		UART1	UART0				
0	1	1	0	0	0	1		UART2	UART0				UART0
0	1	1	0	0	1	0		UART2	UART1			UART0	
0	1	1	0	0	1	1		-(*2)	UART2			UART1	UART0
0	1	1	0	1	0	0		UART2	UART1		UART0		
0	1	1	0	1	0	1		-(*2)	UART2		UART0		UART1
0	1	1	0	1	1	0		-(*2)	UART2		UART0	UART1	
0	1	1	0	1	1	1		-(*2)	-(*2)		UART0	UART2	UART1
0	1	1	1	0	0	0		UART2	UART1	UART0			
0	1	1	1	0	0	1		-(*2)	UART2	UART1			UART0
0	1	1	1	0	1	0		-(*2)	UART2	UART1		UART0	
0	1	1	1	0	1	1		-(*2)	-(*2)	UART2		UART1	UART0
0	1	1	1	1	0	0		-(*2)	UART2	UART1	UART0		
0	1	1	1	1	0	1		-(*2)	-(*2)	UART2	UART0		UART1
0	1	1	1	1	1	0		-(*2)	-(*2)	UART2	UART0	UART1	
0	1	1	1	1	1	1		-(*2)	-(*2)	-(*2)	UART0	UART2	UART1
1	0	0	0	0	0	0	UART0						
1	0	0	0	0	0	1	UART1						UART0
1	0	0	0	0	1	0	UART1					UART0	
1	0	0	0	0	1	1	UART2					UART1	UART0
1	0	0	0	1	0	0	UART1				UART0		
1	0	0	0	1	0	1	UART2				UART0		UART1
1	0	0	0	1	1	0	UART2				UART0	UART1	
1	0	0	0	1	1	1	-(*2)				UART0	UART2	UART1
1	0	0	1	0	0	0	UART1			UART0			
1	0	0	1	0	0	1	UART2			UART0			UART0
1	0	0	1	0	1	0	UART2			UART1		UART0	
1	0	0	1	0	1	1	-(*2)			UART2		UART1	UART0
1	0	0	1	1	0	0	UART2			UART1	UART0		
1	0	0	1	1	0	1	-(*2)			UART2	UART0		UART1
1	0	0	1	1	1	0	-(*2)			UART2	UART0	UART1	
1	0	0	1	1	1	1	-(*2)			-(*2)	UART0	UART2	UART1
1	0	1	0	0	0	0	UART1		UART0				
1	0	1	0	0	0	1	UART2		UART1				UART0
1	0	1	0	0	1	0	UART2		UART1			UART0	
1	0	1	0	0	1	1	-(*2)		UART2			UART1	UART0
1	0	1	0	1	0	0	UART2		UART1		UART0		
1	0	1	0	1	0	1	-(*2)		UART2		UART0		UART1
1	0	1	0	1	1	0	-(*2)		UART2		UART0	UART1	
1	0	1	0	1	1	1	-(*2)		-(*2)		UART0	UART2	UART1
1	0	1	1	0	0	0	UART2		UART1	UART0			
1	0	1	1	0	0	1	-(*2)		UART2	UART1			UART0
1	0	1	1	0	1	0	-(*2)		UART2	UART1		UART0	
1	0	1	1	0	1	1	-(*2)		-(*2)	UART2		UART1	UART0
1	0	1	1	1	0	0	-(*2)		UART2	UART1	UART0		
1	0	1	1	1	0	1	-(*2)		-(*2)	UART2	UART0		UART1
1	0	1	1	1	1	0	-(*2)		-(*2)	UART2	UART0	UART1	
1	0	1	1	1	1	1	-(*2)		-(*2)	-(*2)	UART0	UART2	UART1
1	1	0	0	0	0	0	UART1	UART0					
1	1	0	0	0	0	1	UART2	UART1					UART0
1	1	0	0	0	1	0	UART2	UART1				UART0	
1	1	0	0	0	1	1	-(*2)	UART2				UART1	UART0
1	1	0	0	1	0	0	UART2	UART1			UART0		
1	1	0	0	1	0	1	-(*2)	UART2			UART0		UART1
1	1	0	0	1	1	0	-(*2)	UART2			UART0	UART1	
1	1	0	0	1	1	1	-(*2)	-(*2)			UART0	UART2	UART1
1	1	0	1	0	0	0	UART2	UART1	UART0				
1	1	0	1	0	0	1	-(*2)	UART2	UART1				UART0
1	1	0	1	0	1	0	-(*2)	UART2	UART1			UART0	
1	1	0	1	0	1	1	-(*2)	-(*2)	UART2			UART1	UART0
1	1	0	1	1	0	0	-(*2)	UART2	UART1		UART0		
1	1	0	1	1	0	1	-(*2)	-(*2)	UART2		UART0		UART1
1	1	0	1	1	1	0	-(*2)	-(*2)	UART2		UART0	UART1	
1	1	0	1	1	1	1	-(*2)	-(*2)	-(*2)		UART0	UART2	UART1
1	1	1	0	0	0	0	UART2	UART1	UART0				
1	1	1	0	0	0	1	-(*2)	UART2	UART1				UART0
1	1	1	0	0	1	0	-(*2)	UART2	UART1			UART0	
1	1	1	0	0	1	1	-(*2)	-(*2)	UART2			UART1	UART0
1	1	1	0	1	0	0	-(*2)	UART2	UART1		UART0		
1	1	1	0	1	0	1	-(*2)	-(*2)	UART2		UART0		UART1
1	1	1	0	1	1	0	-(*2)	-(*2)	UART2		UART0	UART1	
1	1	1	0	1	1	1	-(*2)	-(*2)	-(*2)		UART0	UART2	UART1
1	1	1	1	0	0	0	UART2	UART1	UART0				
1	1	1	1	0	0	1	-(*2)	-(*2)	UART2	UART1			UART0
1	1	1	1	0	1	0	-(*2)	-(*2)	UART2	UART1		UART0	
1	1	1	1	0	1	1	-(*2)	-(*2)	-(*2)	UART2		UART1	UART0
1	1	1	1	1	0	0	-(*2)	-(*2)	UART2	UART1	UART0		
1	1	1	1	1	0	1	-(*2)	-(*2)	-(*2)	UART2	UART0		UART1
1	1	1	1	1	1	0	-(*2)	-(*2)	-(*2)	UART2	UART0	UART1	
1	1	1	1	1	1	1	-(*2)	-(*2)	-(*2)	-(*2)	UART0	UART2	UART1

○SPI, SSIS

Enable conditions							GPIOB tertiary (HW secondary)				GPIOB tertiary (HW secondary)		
B[15:12]	B[11: 8]	B[7: 4]	B[3: 0]	A[11: 8]	A[7: 4]	A[3: 0]	[15:12]	[11: 8]	[7: 4]	[3: 0]	[11: 8]	[7: 4]	[3: 0]
0	0	0	0	0	0	0							
0	0	0	0	0	0	1							SPI0
0	0	0	0	0	1	0						SSIS	
0	0	0	0	0	1	1						SSIS	SPI0
0	0	0	0	1	0	0					SPI0		
0	0	0	0	1	0	1					SPI1		SPI0
0	0	0	0	1	1	0					SPI0	SSIS	
0	0	0	0	1	1	1					SPI1	SSIS	SPI0
0	0	0	1	0	0	0				SPI0			
0	0	0	1	0	0	1				SPI1			SPI0
0	0	0	1	0	1	0				-(*2)		SSIS	
0	0	0	1	0	1	1				-(*2)		SSIS	SPI0
0	0	0	1	1	0	0				SPI1	SPI0		
0	0	0	1	1	0	1				-(*2)	SPI1		SPI0
0	0	0	1	1	1	0				-(*2)	SPI0	SSIS	
0	0	0	1	1	1	1				-(*2)	SPI1	SSIS	SPI0
0	0	1	0	0	0	0			SPI0				
0	0	1	0	0	0	1			SPI1				SPI0
0	0	1	0	0	1	0			SPI1			SSIS	
0	0	1	0	0	1	1			-(*2)			SSIS	SPI0
0	0	1	0	1	0	0			SPI1		SPI0		
0	0	1	0	1	0	1			-(*2)		SPI1		SPI0
0	0	1	0	1	1	0			-(*2)		SPI0	SSIS	
0	0	1	0	1	1	1			-(*2)		SPI1	SSIS	SPI0
0	0	1	1	0	0	0			SPI1	SPI0			
0	0	1	1	0	0	1			-(*2)	SPI1			SPI0
0	0	1	1	0	1	0			-(*2)	SPI1		SSIS	
0	0	1	1	0	1	1			-(*2)	-(*2)		SSIS	SPI0
0	0	1	1	1	0	0			-(*2)	SPI1	SPI0		
0	0	1	1	1	0	1			-(*2)	-(*2)	SPI1		SPI0
0	0	1	1	1	1	0			-(*2)	-(*2)	SPI0	SSIS	
0	0	1	1	1	1	1			-(*2)	-(*2)	SPI1	SSIS	SPI0
0	1	0	0	0	0	0		SPI0					
0	1	0	0	0	0	1		SPI1					SPI0
0	1	0	0	0	1	0		SPI1				SSIS	
0	1	0	0	0	1	1		-(*2)				SSIS	SPI0
0	1	0	0	1	0	0		SPI1			SPI0		
0	1	0	0	1	0	1		-(*2)			SPI1		SPI0
0	1	0	0	1	1	0		-(*2)			SPI0	SSIS	
0	1	0	0	1	1	1		-(*2)			SPI1	SSIS	SPI0
0	1	0	1	0	0	0		SPI1		SPI0			
0	1	0	1	0	0	1		-(*2)		SPI1			SPI0
0	1	0	1	0	1	0		-(*2)		SPI1		SSIS	
0	1	0	1	0	1	1		-(*2)		-(*2)		SSIS	SPI0
0	1	0	1	1	0	0		SPI1		SPI0	SPI0		
0	1	0	1	1	0	1		-(*2)		SPI1	SPI0		SPI0
0	1	0	1	1	1	0		-(*2)		SPI0	SPI0	SSIS	
0	1	0	1	1	1	1		-(*2)		SPI1	SPI0	SSIS	SPI0
0	1	1	0	0	0	0		SPI1	SPI0				
0	1	1	0	0	0	1		-(*2)	SPI1				SPI0
0	1	1	0	0	1	0		-(*2)	SPI1			SSIS	
0	1	1	0	0	1	1		-(*2)	-(*2)			SSIS	SPI0
0	1	1	0	1	0	0		SPI1	SPI0		SPI0		
0	1	1	0	1	0	1		-(*2)	SPI1		SPI0		SPI0
0	1	1	0	1	1	0		-(*2)	SPI1		SPI0	SSIS	
0	1	1	0	1	1	1		-(*2)	-(*2)		SPI1	SSIS	SPI0
0	1	1	1	0	0	0		SPI1	SPI0	SPI0			
0	1	1	1	0	0	1		-(*2)	SPI1	SPI0			SPI0
0	1	1	1	0	1	0		-(*2)	SPI1	SPI0		SSIS	
0	1	1	1	0	1	1		-(*2)	-(*2)	SPI0		SSIS	SPI0
0	1	1	1	1	0	0		SPI1	SPI0	SPI0	SPI0		
0	1	1	1	1	0	1		-(*2)	SPI1	SPI0	SPI0		SPI0
0	1	1	1	1	1	0		-(*2)	SPI1	SPI0	SPI0	SSIS	
0	1	1	1	1	1	1		-(*2)	-(*2)	SPI0	SPI0	SSIS	SPI0
1	0	0	0	0	0	0	SPI0						
1	0	0	0	0	0	1	SPI1						SPI0
1	0	0	0	0	1	0	SPI1					SSIS	
1	0	0	0	0	1	1	-(*2)					SSIS	SPI0
1	0	0	0	1	0	0	SPI1				SPI0		
1	0	0	0	1	0	1	-(*2)				SPI1		SPI0
1	0	0	0	1	1	0	-(*2)				SPI0	SSIS	
1	0	0	0	1	1	1	-(*2)				SPI1	SSIS	SPI0
1	0	0	1	0	0	0	SPI1			SPI0			
1	0	0	1	0	0	1	-(*2)			SPI1			SPI0
1	0	0	1	0	1	0	-(*2)			SPI1		SSIS	
1	0	0	1	0	1	1	-(*2)			-(*2)		SSIS	SPI0
1	0	0	1	1	0	0	-(*2)			SPI1	SPI0		
1	0	0	1	1	0	1	-(*2)			-(*2)	SPI1		SPI0
1	0	0	1	1	1	0	-(*2)			-(*2)	SPI0	SSIS	
1	0	0	1	1	1	1	-(*2)			-(*2)	SPI1	SSIS	SPI0
1	0	1	0	0	0	0	SPI1		SPI0				
1	0	1	0	0	0	1	-(*2)		SPI1				SPI0
1	0	1	0	0	1	0	-(*2)		SPI1			SSIS	
1	0	1	0	0	1	1	-(*2)		-(*2)			SSIS	SPI0
1	0	1	0	1	0	0	-(*2)		SPI1		SPI0		
1	0	1	0	1	0	1	-(*2)		-(*2)		SPI1		SPI0
1	0	1	0	1	1	0	-(*2)		-(*2)		SPI0	SSIS	
1	0	1	0	1	1	1	-(*2)		-(*2)		SPI1	SSIS	SPI0
1	0	1	1	0	0	0	-(*2)		SPI1	SPI0			
1	0	1	1	0	0	1	-(*2)		-(*2)	SPI0			SPI0
1	0	1	1	0	1	0	-(*2)		-(*2)	SPI0		SSIS	
1	0	1	1	0	1	1	-(*2)		-(*2)	SPI0		SSIS	SPI0
1	0	1	1	1	0	0	-(*2)		-(*2)	SPI0	SPI0		
1	0	1	1	1	0	1	-(*2)		-(*2)	SPI0	SPI0		SPI0
1	0	1	1	1	1	0	-(*2)		-(*2)	SPI0	SPI0	SSIS	
1	0	1	1	1	1	1	-(*2)		-(*2)	SPI0	SPI0	SSIS	SPI0
1	1	0	0	0	0	0	SPI1	SPI0					
1	1	0	0	0	0	1	-(*2)	SPI1					SPI0
1	1	0	0	0	1	0	-(*2)	SPI1				SSIS	
1	1	0	0	0	1	1	-(*2)	-(*2)				SSIS	SPI0
1	1	0	0	1	0	0	-(*2)	SPI1			SPI0		
1	1	0	0	1	0	1	-(*2)	-(*2)			SPI1		SPI0
1	1	0	0	1	1	0	-(*2)	-(*2)			SPI0	SSIS	
1	1	0	0	1	1	1	-(*2)	-(*2)			SPI1	SSIS	SPI0
1	1	0	1	0	0	0	-(*2)	SPI1		SPI0			
1	1	0	1	0	0	1	-(*2)	-(*2)		SPI0			SPI0
1	1	0	1	0	1	0	-(*2)	-(*2)		SPI0		SSIS	
1	1	0	1	0	1	1	-(*2)	-(*2)		SPI0		SSIS	SPI0
1	1	0	1	1	0	0	-(*2)	-(*2)		SPI0	SPI0		
1	1	0	1	1	0	1	-(*2)	-(*2)		SPI0	SPI0		SPI0
1	1	0	1	1	1	0	-(*2)	-(*2)		SPI0	SPI0	SSIS	
1	1	0	1	1	1	1	-(*2)	-(*2)		SPI0	SPI0	SSIS	SPI0
1	1	1	0	0	0	0	-(*2)	SPI1	SPI0				
1	1	1	0	0	0	1	-(*2)	-(*2)	SPI1				SPI0
1	1	1	0	0	1	0	-(*2)	-(*2)	SPI1			SSIS	
1	1	1	0	0	1	1	-(*2)	-(*2)	-(*2)			SSIS	SPI0
1	1	1	0	1	0	0	-(*2)	-(*2)	SPI1		SPI0		
1	1	1	0	1	0	1	-(*2)	-(*2)	-(*2)		SPI1		SPI0
1	1	1	0	1	1	0	-(*2)	-(*2)	-(*2)		SPI0	SSIS	
1	1	1	0	1	1	1	-(*2)	-(*2)	-(*2)		SPI1	SSIS	SPI0
1	1	1	1	0	0	0	-(*2)	-(*2)	SPI1	SPI0			
1	1	1	1	0	0	1	-(*2)	-(*2)	-(*2)	SPI0			SPI0
1	1	1	1	0	1	0	-(*2)	-(*2)	-(*2)	SPI0		SSIS	
1	1	1	1	0	1	1	-(*2)	-(*2)	-(*2)	-(*2)		SSIS	SPI0
1	1	1	1	1	0	0	-(*2)	-(*2)	-(*2)	-(*2)	SPI0		
1	1	1	1	1	0	1	-(*2)	-(*2)	-(*2)	-(*2)	SPI1		SPI0
1	1	1	1	1	1	0	-(*2)	-(*2)	-(*2)	-(*2)	SPI0	SSIS	
1	1	1	1	1	1	1	-(*2)	-(*2)	-(*2)	-(*2)	SPI1	SSIS	SPI0

○I2C

Enable conditions							GPIOB quartic (HW tertiary)				GPIOB quartic (HW tertiary)		
B[15:12]	B[11: 8]	B[7: 4]	B[3: 0]	A[11: 8]	A[7: 4]	A[3: 0]	[15:12]	[11: 8]	[7: 4]	[3: 0]	[11: 8]	[7: 4]	[3: 0]
0	0	0	0	0	0	0							
0	0	0	0	0	0	1							I2C
0	0	0	0	0	1	0						I2C	
0	0	0	0	0	1	1						-(*2)	I2C
0	0	0	0	1	0	0					I2C		
0	0	0	0	1	0	1					-(*2)		I2C
0	0	0	0	1	1	0					-(*2)	I2C	
0	0	0	0	1	1	1					-(*2)	-(*2)	I2C
0	0	0	1	0	0	0				I2C			
0	0	0	1	0	0	1				-(*2)			I2C
0	0	0	1	0	1	0				-(*2)		I2C	
0	0	0	1	0	1	1				-(*2)		-(*2)	I2C
0	0	0	1	1	0	0				-(*2)	I2C		
0	0	0	1	1	0	1				-(*2)	-(*2)		I2C
0	0	0	1	1	1	0				-(*2)	-(*2)	I2C	
0	0	0	1	1	1	1				-(*2)	-(*2)	-(*2)	I2C
0	0	1	0	0	0	0			I2C				
0	0	1	0	0	0	1			-(*2)				I2C
0	0	1	0	0	1	0			-(*2)			I2C	
0	0	1	0	0	1	1			-(*2)			-(*2)	I2C
0	0	1	0	1	0	0			-(*2)		I2C		
0	0	1	0	1	0	1			-(*2)		-(*2)		I2C
0	0	1	0	1	1	0			-(*2)		-(*2)	I2C	
0	0	1	0	1	1	1			-(*2)		-(*2)	-(*2)	I2C
0	0	1	1	0	0	0			-(*2)	I2C			
0	0	1	1	0	0	1			-(*2)	-(*2)			I2C
0	0	1	1	0	1	0			-(*2)	-(*2)		I2C	
0	0	1	1	0	1	1			-(*2)	-(*2)		-(*2)	I2C
0	0	1	1	1	0	0			-(*2)	-(*2)	I2C		
0	0	1	1	1	0	1			-(*2)	-(*2)	-(*2)		I2C
0	0	1	1	1	1	0			-(*2)	-(*2)	-(*2)	I2C	
0	0	1	1	1	1	1			-(*2)	-(*2)	-(*2)	-(*2)	I2C
0	1	0	0	0	0	0		I2C					
0	1	0	0	0	0	1		-(*2)					I2C
0	1	0	0	0	1	0		-(*2)				I2C	
0	1	0	0	0	1	1		-(*2)				-(*2)	I2C
0	1	0	0	1	0	0		-(*2)			I2C		
0	1	0	0	1	0	1		-(*2)			-(*2)		I2C
0	1	0	0	1	1	0		-(*2)			-(*2)	I2C	
0	1	0	0	1	1	1		-(*2)			-(*2)	-(*2)	I2C
0	1	0	1	0	0	0		-(*2)		I2C			
0	1	0	1	0	0	1		-(*2)		-(*2)			I2C
0	1	0	1	0	1	0		-(*2)		-(*2)		I2C	
0	1	0	1	0	1	1		-(*2)		-(*2)		-(*2)	I2C
0	1	0	1	1	0	0		-(*2)		-(*2)	I2C		
0	1	0	1	1	0	1		-(*2)		-(*2)	-(*2)		I2C
0	1	0	1	1	1	0		-(*2)		-(*2)	-(*2)	I2C	
0	1	0	1	1	1	1		-(*2)		-(*2)	-(*2)	-(*2)	I2C
0	1	1	0	0	0	0		-(*2)	I2C				
0	1	1	0	0	0	1		-(*2)	-(*2)				I2C
0	1	1	0	0	1	0		-(*2)	-(*2)			I2C	
0	1	1	0	0	1	1		-(*2)	-(*2)			-(*2)	I2C
0	1	1	0	1	0	0		-(*2)	-(*2)		I2C		
0	1	1	0	1	0	1		-(*2)	-(*2)		-(*2)		I2C
0	1	1	0	1	1	0		-(*2)	-(*2)		-(*2)	I2C	
0	1	1	0	1	1	1		-(*2)	-(*2)		-(*2)	-(*2)	I2C
0	1	1	1	0	0	0		-(*2)	-(*2)	-(*2)	I2C		
0	1	1	1	0	0	1		-(*2)	-(*2)	-(*2)	-(*2)		I2C
0	1	1	1	0	1	0		-(*2)	-(*2)	-(*2)	-(*2)	I2C	
0	1	1	1	0	1	1		-(*2)	-(*2)	-(*2)	-(*2)	-(*2)	I2C
0	1	1	1	1	0	0		-(*2)	-(*2)	-(*2)	-(*2)	I2C	
0	1	1	1	1	0	1		-(*2)	-(*2)	-(*2)	-(*2)	-(*2)	I2C
0	1	1	1	1	1	0		-(*2)	-(*2)	-(*2)	-(*2)	I2C	
0	1	1	1	1	1	1		-(*2)	-(*2)	-(*2)	-(*2)	-(*2)	I2C
1	0	0	0	0	0	0	I2C						
1	0	0	0	0	0	1	-(*2)						I2C
1	0	0	0	0	1	0	-(*2)					I2C	
1	0	0	0	0	1	1	-(*2)					-(*2)	I2C
1	0	0	0	1	0	0	-(*2)				I2C		
1	0	0	0	1	0	1	-(*2)				-(*2)		I2C
1	0	0	0	1	1	0	-(*2)				-(*2)	I2C	
1	0	0	0	1	1	1	-(*2)				-(*2)	-(*2)	I2C
1	0	0	1	0	0	0	-(*2)			I2C			
1	0	0	1	0	0	1	-(*2)			-(*2)			I2C
1	0	0	1	0	1	0	-(*2)			-(*2)		I2C	
1	0	0	1	0	1	1	-(*2)			-(*2)		-(*2)	I2C
1	0	0	1	1	0	0	-(*2)			-(*2)	I2C		
1	0	0	1	1	0	1	-(*2)			-(*2)	-(*2)		I2C
1	0	0	1	1	1	0	-(*2)			-(*2)	-(*2)	I2C	
1	0	0	1	1	1	1	-(*2)			-(*2)	-(*2)	-(*2)	I2C
1	0	1	0	0	0	0	-(*2)		I2C				
1	0	1	0	0	0	1	-(*2)		-(*2)				I2C
1	0	1	0	0	1	0	-(*2)		-(*2)			I2C	
1	0	1	0	0	1	1	-(*2)		-(*2)			-(*2)	I2C
1	0	1	0	1	0	0	-(*2)		-(*2)		I2C		
1	0	1	0	1	0	1	-(*2)		-(*2)		-(*2)		I2C
1	0	1	0	1	1	0	-(*2)		-(*2)		-(*2)	I2C	
1	0	1	0	1	1	1	-(*2)		-(*2)		-(*2)	-(*2)	I2C
1	0	1	1	0	0	0	-(*2)		-(*2)	-(*2)	I2C		
1	0	1	1	0	0	1	-(*2)		-(*2)	-(*2)	-(*2)		I2C
1	0	1	1	0	1	0	-(*2)		-(*2)	-(*2)	-(*2)	I2C	
1	0	1	1	0	1	1	-(*2)		-(*2)	-(*2)	-(*2)	-(*2)	I2C
1	0	1	1	1	0	0	-(*2)		-(*2)	-(*2)	I2C		
1	0	1	1	1	0	1	-(*2)		-(*2)	-(*2)	-(*2)		I2C
1	0	1	1	1	1	0	-(*2)		-(*2)	-(*2)	-(*2)	I2C	
1	0	1	1	1	1	1	-(*2)		-(*2)	-(*2)	-(*2)	-(*2)	I2C
1	1	0	0	0	0	0	-(*2)	I2C					
1	1	0	0	0	0	1	-(*2)	-(*2)					I2C
1	1	0	0	0	1	0	-(*2)	-(*2)				I2C	
1	1	0	0	0	1	1	-(*2)	-(*2)				-(*2)	I2C
1	1	0	0	1	0	0	-(*2)	-(*2)			I2C		
1	1	0	0	1	0	1	-(*2)	-(*2)			-(*2)		I2C
1	1	0	0	1	1	0	-(*2)	-(*2)			-(*2)	I2C	
1	1	0	0	1	1	1	-(*2)	-(*2)			-(*2)	-(*2)	I2C
1	1	0	1	0	0	0	-(*2)	-(*2)		I2C			
1	1	0	1	0	0	1	-(*2)	-(*2)		-(*2)			I2C
1	1	0	1	0	1	0	-(*2)	-(*2)		-(*2)		I2C	
1	1	0	1	0	1	1	-(*2)	-(*2)		-(*2)		-(*2)	I2C
1	1	0	1	1	0	0	-(*2)	-(*2)		-(*2)	I2C		
1	1	0	1	1	0	1	-(*2)	-(*2)		-(*2)	-(*2)		I2C
1	1	0	1	1	1	0	-(*2)	-(*2)		-(*2)	-(*2)	I2C	
1	1	0	1	1	1	1	-(*2)	-(*2)		-(*2)	-(*2)	-(*2)	I2C
1	1	1	0	0	0	0	-(*2)	-(*2)	-(*2)				
1	1	1	0	0	0	1	-(*2)	-(*2)	-(*2)				I2C
1	1	1	0	0	1	0	-(*2)	-(*2)	-(*2)			I2C	
1	1	1	0	0	1	1	-(*2)	-(*2)	-(*2)			-(*2)	I2C
1	1	1	0	1	0	0	-(*2)	-(*2)	-(*2)		I2C		
1	1	1	0	1	0	1	-(*2)	-(*2)	-(*2)		-(*2)		I2C
1	1	1	0	1	1	0	-(*2)	-(*2)	-(*2)		-(*2)	I2C	
1	1	1	0	1	1	1	-(*2)	-(*2)	-(*2)		-(*2)	-(*2)	I2C
1	1	1	1	0	0	0	-(*2)	-(*2)	-(*2)	-(*2)			
1	1	1	1	0	0	1	-(*2)	-(*2)	-(*2)	-(*2)			I2C
1	1	1	1	0	1	0	-(*2)	-(*2)	-(*2)	-(*2)		I2C	
1	1	1	1	0	1	1	-(*2)	-(*2)	-(*2)	-(*2)		-(*2)	I2C
1	1	1	1	1	0	0	-(*2)	-(*2)	-(*2)	-(*2)	I2C		
1	1	1	1	1	0	1	-(*2)	-(*2)	-(*2)	-(*2)	-(*2)		I2C
1	1	1	1	1	1	0	-(*2)	-(*2)	-(*2)	-(*2)	-(*2)	I2C	
1	1	1	1	1	1	1	-(*2)	-(*2)	-(*2)	-(*2)	-(*2)	-(*2)	I2C

○FTM

Enable conditions							GPIOB quartic (HW tertiary)				GPIOB quartic (HW tertiary)		
B[15:12]	B[11: 8]	B[7: 4]	B[3: 0]	A[11: 8]	A[7: 4]	A[3: 0]	[15:12]	[11: 8]	[7: 4]	[3: 0]	[11: 8]	[7: 4]	[3: 0]
0	0	0	0	0	0	0							
0	0	0	0	0	0	1							FTMO
0	0	0	0	0	1	0						FTMO	
0	0	0	0	0	1	1						-(*2)	FTMO
0	0	0	0	1	0	0					FTMO		
0	0	0	0	1	0	1					-(*2)		FTMO
0	0	0	0	1	1	0					-(*2)	FTMO	
0	0	0	0	1	1	1					-(*2)	-(*2)	FTMO
0	0	0	1	0	0	0				FTMO			
0	0	0	1	0	0	1				-(*2)			FTMO
0	0	0	1	0	1	0				-(*2)		FTMO	
0	0	0	1	0	1	1				-(*2)		-(*2)	FTMO
0	0	0	1	1	0	0				-(*2)	FTMO		
0	0	0	1	1	0	1				-(*2)	-(*2)		FTMO
0	0	0	1	1	1	0				-(*2)	-(*2)	FTMO	
0	0	0	1	1	1	1				-(*2)	-(*2)	-(*2)	FTMO
0	0	1	0	0	0	0			FTMO				
0	0	1	0	0	0	1			-(*2)				FTMO
0	0	1	0	0	1	0			-(*2)			FTMO	
0	0	1	0	0	1	1			-(*2)			-(*2)	FTMO
0	0	1	0	1	0	0			-(*2)		FTMO		
0	0	1	0	1	0	1			-(*2)		-(*2)		FTMO
0	0	1	0	1	1	0			-(*2)		-(*2)	FTMO	
0	0	1	0	1	1	1			-(*2)		-(*2)	-(*2)	FTMO
0	0	1	1	0	0	0			-(*2)	FTMO			
0	0	1	1	0	0	1			-(*2)	-(*2)			FTMO
0	0	1	1	0	1	0			-(*2)	-(*2)		FTMO	
0	0	1	1	0	1	1			-(*2)	-(*2)		-(*2)	FTMO
0	0	1	1	1	0	0			-(*2)	-(*2)	FTMO		
0	0	1	1	1	0	1			-(*2)	-(*2)	-(*2)		FTMO
0	0	1	1	1	1	0			-(*2)	-(*2)	-(*2)	FTMO	
0	0	1	1	1	1	1			-(*2)	-(*2)	-(*2)	-(*2)	FTMO
0	1	0	0	0	0	0		FTMO					
0	1	0	0	0	0	1		-(*2)					FTMO
0	1	0	0	0	1	0		-(*2)				FTMO	
0	1	0	0	0	1	1		-(*2)				-(*2)	FTMO
0	1	0	0	1	0	0		-(*2)			FTMO		
0	1	0	0	1	0	1		-(*2)			-(*2)		FTMO
0	1	0	0	1	1	0		-(*2)			-(*2)	FTMO	
0	1	0	0	1	1	1		-(*2)			-(*2)	-(*2)	FTMO
0	1	0	1	0	0	0		-(*2)		FTMO			
0	1	0	1	0	0	1		-(*2)		-(*2)			FTMO
0	1	0	1	0	1	0		-(*2)		-(*2)		FTMO	
0	1	0	1	0	1	1		-(*2)		-(*2)		-(*2)	FTMO
0	1	0	1	1	0	0		-(*2)		-(*2)	FTMO		
0	1	0	1	1	0	1		-(*2)		-(*2)	-(*2)		FTMO
0	1	0	1	1	1	0		-(*2)		-(*2)	-(*2)	FTMO	
0	1	0	1	1	1	1		-(*2)		-(*2)	-(*2)	-(*2)	FTMO
0	1	1	0	0	0	0		-(*2)	FTMO				
0	1	1	0	0	0	1		-(*2)	-(*2)				FTMO
0	1	1	0	0	1	0		-(*2)	-(*2)			FTMO	
0	1	1	0	0	1	1		-(*2)	-(*2)			-(*2)	FTMO
0	1	1	0	1	0	0		-(*2)	-(*2)		FTMO		
0	1	1	0	1	0	1		-(*2)	-(*2)		-(*2)		FTMO
0	1	1	0	1	1	0		-(*2)	-(*2)		-(*2)	FTMO	
0	1	1	0	1	1	1		-(*2)	-(*2)		-(*2)	-(*2)	FTMO
0	1	1	1	0	0	0		-(*2)	-(*2)	FTMO			
0	1	1	1	0	0	1		-(*2)	-(*2)	-(*2)			FTMO
0	1	1	1	0	1	0		-(*2)	-(*2)	-(*2)		FTMO	
0	1	1	1	0	1	1		-(*2)	-(*2)	-(*2)		-(*2)	FTMO
0	1	1	1	1	0	0		-(*2)	-(*2)	-(*2)	FTMO		
0	1	1	1	1	0	1		-(*2)	-(*2)	-(*2)	-(*2)		FTMO
0	1	1	1	1	1	0		-(*2)	-(*2)	-(*2)	-(*2)	FTMO	
0	1	1	1	1	1	1		-(*2)	-(*2)	-(*2)	-(*2)	-(*2)	FTMO
1	0	0	0	0	0	0	FTMO						
1	0	0	0	0	0	1	-(*2)						FTMO
1	0	0	0	0	1	0	-(*2)					FTMO	
1	0	0	0	0	1	1	-(*2)					-(*2)	FTMO
1	0	0	0	1	0	0	-(*2)				FTMO		
1	0	0	0	1	0	1	-(*2)				-(*2)		FTMO
1	0	0	0	1	1	0	-(*2)				-(*2)	FTMO	
1	0	0	0	1	1	1	-(*2)				-(*2)	-(*2)	FTMO
1	0	0	1	0	0	0	-(*2)			FTMO			
1	0	0	1	0	0	1	-(*2)			-(*2)			FTMO
1	0	0	1	0	1	0	-(*2)			-(*2)		FTMO	
1	0	0	1	0	1	1	-(*2)			-(*2)		-(*2)	FTMO
1	0	0	1	1	0	0	-(*2)			-(*2)	FTMO		
1	0	0	1	1	0	1	-(*2)			-(*2)	-(*2)		FTMO
1	0	0	1	1	1	0	-(*2)			-(*2)	-(*2)	FTMO	
1	0	0	1	1	1	1	-(*2)			-(*2)	-(*2)	-(*2)	FTMO
1	0	1	0	0	0	0	-(*2)		FTMO				
1	0	1	0	0	0	1	-(*2)		-(*2)				FTMO
1	0	1	0	0	1	0	-(*2)		-(*2)			FTMO	
1	0	1	0	0	1	1	-(*2)		-(*2)			-(*2)	FTMO
1	0	1	0	1	0	0	-(*2)		-(*2)		FTMO		
1	0	1	0	1	0	1	-(*2)		-(*2)		-(*2)		FTMO
1	0	1	0	1	1	0	-(*2)		-(*2)		-(*2)	FTMO	
1	0	1	0	1	1	1	-(*2)		-(*2)		-(*2)	-(*2)	FTMO
1	0	1	1	0	0	0	-(*2)		-(*2)	FTMO			
1	0	1	1	0	0	1	-(*2)		-(*2)	-(*2)			FTMO
1	0	1	1	0	1	0	-(*2)		-(*2)	-(*2)		FTMO	
1	0	1	1	0	1	1	-(*2)		-(*2)	-(*2)		-(*2)	FTMO
1	0	1	1	1	0	0	-(*2)		-(*2)	-(*2)	FTMO		
1	0	1	1	1	0	1	-(*2)		-(*2)	-(*2)	-(*2)		FTMO
1	0	1	1	1	1	0	-(*2)		-(*2)	-(*2)	-(*2)	FTMO	
1	0	1	1	1	1	1	-(*2)		-(*2)	-(*2)	-(*2)	-(*2)	FTMO
1	1	0	0	0	0	0	-(*2)	FTMO					
1	1	0	0	0	0	1	-(*2)	-(*2)					FTMO
1	1	0	0	0	1	0	-(*2)	-(*2)				FTMO	
1	1	0	0	0	1	1	-(*2)	-(*2)				-(*2)	FTMO
1	1	0	0	1	0	0	-(*2)	-(*2)			FTMO		
1	1	0	0	1	0	1	-(*2)	-(*2)			-(*2)		FTMO
1	1	0	0	1	1	0	-(*2)	-(*2)			-(*2)	FTMO	
1	1	0	0	1	1	1	-(*2)	-(*2)			-(*2)	-(*2)	FTMO
1	1	0	1	0	0	0	-(*2)	-(*2)		FTMO			
1	1	0	1	0	0	1	-(*2)	-(*2)		-(*2)			FTMO
1	1	0	1	0	1	0	-(*2)	-(*2)		-(*2)		FTMO	
1	1	0	1	0	1	1	-(*2)	-(*2)		-(*2)		-(*2)	FTMO
1	1	0	1	1	0	0	-(*2)	-(*2)		-(*2)	FTMO		
1	1	0	1	1	0	1	-(*2)	-(*2)		-(*2)	-(*2)		FTMO
1	1	0	1	1	1	0	-(*2)	-(*2)		-(*2)	-(*2)	FTMO	
1	1	0	1	1	1	1	-(*2)	-(*2)		-(*2)	-(*2)	-(*2)	FTMO
1	1	1	0	0	0	0	-(*2)	-(*2)	FTMO				
1	1	1	0	0	0	1	-(*2)	-(*2)	-(*2)				FTMO
1	1	1	0	0	1	0	-(*2)	-(*2)	-(*2)			FTMO	
1	1	1	0	0	1	1	-(*2)	-(*2)	-(*2)			-(*2)	FTMO
1	1	1	0	1	0	0	-(*2)	-(*2)	-(*2)		FTMO		
1	1	1	0	1	0	1	-(*2)	-(*2)	-(*2)		-(*2)		FTMO
1	1	1	0	1	1	0	-(*2)	-(*2)	-(*2)		-(*2)	FTMO	
1	1	1	0	1	1	1	-(*2)	-(*2)	-(*2)		-(*2)	-(*2)	FTMO
1	1	1	1	0	0	0	-(*2)	-(*2)	-(*2)	-(*2)			
1	1	1	1	0	0	1	-(*2)	-(*2)	-(*2)	-(*2)			FTMO
1	1	1	1	0	1	0	-(*2)	-(*2)	-(*2)	-(*2)		FTMO	
1	1	1	1	0	1	1	-(*2)	-(*2)	-(*2)	-(*2)		-(*2)	FTMO
1	1	1	1	1	0	0	-(*2)	-(*2)	-(*2)	-(*2)	FTMO		
1	1	1	1	1	0	1	-(*2)	-(*2)	-(*2)	-(*2)	-(*2)		FTMO
1	1	1	1	1	1	0	-(*2)	-(*2)	-(*2)	-(*2)	-(*2)	FTMO	
1	1	1	1	1	1	1	-(*2)	-(*2)	-(*2)	-(*2)	-(*2)	-(*2)	FTMO

○STD GPIO

Enable conditions							GPIOB quintic (HW quartic)				GPIOB quartic (HW tertiary)		
B[15:12]	B[11: 8]	B[7: 4]	B[3: 0]	A[11: 8]	A[7: 4]	A[3: 0]	[15:12]	[11: 8]	[7: 4]	[3: 0]	[11: 8]	[7: 4]	[3: 0]
0	0	0	0	0	0	0							
0	0	0	0	0	0	1							STD_GPIO
0	0	0	0	0	1	0						STD_GPIO	
0	0	0	0	0	1	1						-(*2)	STD_GPIO
0	0	0	0	1	0	0					STD_GPIO		
0	0	0	0	1	0	1					-(*2)		STD_GPIO
0	0	0	0	1	1	0					-(*2)	STD_GPIO	
0	0	0	0	1	1	1					-(*2)	-(*2)	STD_GPIO
0	0	0	1	0	0	0				STD_GPIO			
0	0	0	1	0	0	1				-(*2)			STD_GPIO
0	0	0	1	0	1	0				-(*2)		STD_GPIO	
0	0	0	1	0	1	1				-(*2)		-(*2)	STD_GPIO
0	0	0	1	1	0	0				-(*2)	STD_GPIO		
0	0	0	1	1	0	1				-(*2)	-(*2)		STD_GPIO
0	0	0	1	1	1	0				-(*2)	-(*2)	STD_GPIO	
0	0	0	1	1	1	1				-(*2)	-(*2)	-(*2)	STD_GPIO
0	0	1	0	0	0	0			STD_GPIO				
0	0	1	0	0	0	1			-(*2)				STD_GPIO
0	0	1	0	0	1	0			-(*2)			STD_GPIO	
0	0	1	0	0	1	1			-(*2)			-(*2)	STD_GPIO
0	0	1	0	1	0	0			-(*2)		STD_GPIO		
0	0	1	0	1	0	1			-(*2)		-(*2)		STD_GPIO
0	0	1	0	1	1	0			-(*2)		-(*2)	STD_GPIO	
0	0	1	0	1	1	1			-(*2)		-(*2)	-(*2)	STD_GPIO
0	0	1	1	0	0	0			-(*2)	STD_GPIO			
0	0	1	1	0	0	1			-(*2)	-(*2)			STD_GPIO
0	0	1	1	0	1	0			-(*2)	-(*2)		STD_GPIO	
0	0	1	1	0	1	1			-(*2)	-(*2)		-(*2)	STD_GPIO
0	0	1	1	1	0	0			-(*2)	-(*2)	STD_GPIO		
0	0	1	1	1	0	1			-(*2)	-(*2)	-(*2)		STD_GPIO
0	0	1	1	1	1	0			-(*2)	-(*2)	-(*2)	STD_GPIO	
0	0	1	1	1	1	1			-(*2)	-(*2)	-(*2)	-(*2)	STD_GPIO
0	1	0	0	0	0	0		STD_GPIO					
0	1	0	0	0	0	1		-(*2)					STD_GPIO
0	1	0	0	0	1	0		-(*2)				STD_GPIO	
0	1	0	0	0	1	1		-(*2)				-(*2)	STD_GPIO
0	1	0	0	1	0	0		-(*2)			STD_GPIO		
0	1	0	0	1	0	1		-(*2)			-(*2)		STD_GPIO
0	1	0	0	1	1	0		-(*2)			-(*2)	STD_GPIO	
0	1	0	0	1	1	1		-(*2)			-(*2)	-(*2)	STD_GPIO
0	1	0	1	0	0	0		-(*2)		STD_GPIO			
0	1	0	1	0	0	1		-(*2)		-(*2)			STD_GPIO
0	1	0	1	0	1	0		-(*2)		-(*2)		STD_GPIO	
0	1	0	1	0	1	1		-(*2)		-(*2)		-(*2)	STD_GPIO
0	1	0	1	1	0	0		-(*2)		-(*2)	STD_GPIO		
0	1	0	1	1	0	1		-(*2)		-(*2)	-(*2)		STD_GPIO
0	1	0	1	1	1	0		-(*2)		-(*2)	-(*2)	STD_GPIO	
0	1	0	1	1	1	1		-(*2)		-(*2)	-(*2)	-(*2)	STD_GPIO
0	1	1	0	0	0	0		-(*2)	STD_GPIO				
0	1	1	0	0	0	1		-(*2)	-(*2)				STD_GPIO
0	1	1	0	0	1	0		-(*2)	-(*2)			STD_GPIO	
0	1	1	0	0	1	1		-(*2)	-(*2)			-(*2)	STD_GPIO
0	1	1	0	1	0	0		-(*2)	-(*2)		STD_GPIO		
0	1	1	0	1	0	1		-(*2)	-(*2)		-(*2)		STD_GPIO
0	1	1	0	1	1	0		-(*2)	-(*2)		-(*2)	STD_GPIO	
0	1	1	0	1	1	1		-(*2)	-(*2)		-(*2)	-(*2)	STD_GPIO
0	1	1	1	0	0	0		-(*2)	-(*2)	STD_GPIO			
0	1	1	1	0	0	1		-(*2)	-(*2)	-(*2)			STD_GPIO
0	1	1	1	0	1	0		-(*2)	-(*2)	-(*2)		STD_GPIO	
0	1	1	1	0	1	1		-(*2)	-(*2)	-(*2)		-(*2)	STD_GPIO
0	1	1	1	1	0	0		-(*2)	-(*2)	-(*2)	STD_GPIO		
0	1	1	1	1	0	1		-(*2)	-(*2)	-(*2)	-(*2)		STD_GPIO
0	1	1	1	1	1	0		-(*2)	-(*2)	-(*2)	-(*2)	STD_GPIO	
0	1	1	1	1	1	1		-(*2)	-(*2)	-(*2)	-(*2)	-(*2)	STD_GPIO
1	0	0	0	0	0	0	STD_GPIO						
1	0	0	0	0	0	1	-(*2)						STD_GPIO
1	0	0	0	0	1	0	-(*2)					STD_GPIO	
1	0	0	0	0	1	1	-(*2)					-(*2)	STD_GPIO
1	0	0	0	1	0	0	-(*2)				STD_GPIO		
1	0	0	0	1	0	1	-(*2)				-(*2)		STD_GPIO
1	0	0	0	1	1	0	-(*2)				-(*2)	STD_GPIO	
1	0	0	0	1	1	1	-(*2)				-(*2)	-(*2)	STD_GPIO
1	0	0	1	0	0	0	-(*2)			STD_GPIO			
1	0	0	1	0	0	1	-(*2)			-(*2)			STD_GPIO
1	0	0	1	0	1	0	-(*2)			-(*2)		STD_GPIO	
1	0	0	1	0	1	1	-(*2)			-(*2)		-(*2)	STD_GPIO
1	0	0	1	1	0	0	-(*2)			-(*2)	STD_GPIO		
1	0	0	1	1	0	1	-(*2)			-(*2)	-(*2)		STD_GPIO
1	0	0	1	1	1	0	-(*2)			-(*2)	-(*2)	STD_GPIO	
1	0	0	1	1	1	1	-(*2)			-(*2)	-(*2)	-(*2)	STD_GPIO
1	0	1	0	0	0	0	-(*2)		STD_GPIO				
1	0	1	0	0	0	1	-(*2)		-(*2)				STD_GPIO
1	0	1	0	0	1	0	-(*2)		-(*2)			STD_GPIO	
1	0	1	0	0	1	1	-(*2)		-(*2)			-(*2)	STD_GPIO
1	0	1	0	1	0	0	-(*2)		-(*2)		STD_GPIO		
1	0	1	0	1	0	1	-(*2)		-(*2)		-(*2)		STD_GPIO
1	0	1	0	1	1	0	-(*2)		-(*2)		-(*2)	STD_GPIO	
1	0	1	0	1	1	1	-(*2)		-(*2)		-(*2)	-(*2)	STD_GPIO
1	0	1	1	0	0	0	-(*2)		-(*2)	STD_GPIO			
1	0	1	1	0	0	1	-(*2)		-(*2)	-(*2)			STD_GPIO
1	0	1	1	0	1	0	-(*2)		-(*2)	-(*2)		STD_GPIO	
1	0	1	1	0	1	1	-(*2)		-(*2)	-(*2)		-(*2)	STD_GPIO
1	0	1	1	1	0	0	-(*2)		-(*2)	-(*2)	STD_GPIO		
1	0	1	1	1	0	1	-(*2)		-(*2)	-(*2)	-(*2)		STD_GPIO
1	0	1	1	1	1	0	-(*2)		-(*2)	-(*2)	-(*2)	STD_GPIO	
1	0	1	1	1	1	1	-(*2)		-(*2)	-(*2)	-(*2)	-(*2)	STD_GPIO
1	1	0	0	0	0	0	-(*2)	-(*2)					
1	1	0	0	0	0	1	-(*2)	-(*2)					STD_GPIO
1	1	0	0	0	1	0	-(*2)	-(*2)				STD_GPIO	
1	1	0	0	0	1	1	-(*2)	-(*2)				-(*2)	STD_GPIO
1	1	0	0	1	0	0	-(*2)	-(*2)			STD_GPIO		
1	1	0	0	1	0	1	-(*2)	-(*2)			-(*2)		STD_GPIO
1	1	0	0	1	1	0	-(*2)	-(*2)			-(*2)	STD_GPIO	
1	1	0	0	1	1	1	-(*2)	-(*2)			-(*2)	-(*2)	STD_GPIO
1	1	0	1	0	0	0	-(*2)	-(*2)		STD_GPIO			
1	1	0	1	0	0	1	-(*2)	-(*2)		-(*2)			STD_GPIO
1	1	0	1	0	1	0	-(*2)	-(*2)		-(*2)		STD_GPIO	
1	1	0	1	0	1	1	-(*2)	-(*2)		-(*2)		-(*2)	STD_GPIO
1	1	0	1	1	0	0	-(*2)	-(*2)		-(*2)	STD_GPIO		
1	1	0	1	1	0	1	-(*2)	-(*2)		-(*2)	-(*2)		STD_GPIO
1	1	0	1	1	1	0	-(*2)	-(*2)		-(*2)	-(*2)	STD_GPIO	
1	1	0	1	1	1	1	-(*2)	-(*2)		-(*2)	-(*2)	-(*2)	STD_GPIO
1	1	1	0	0	0	0	-(*2)	-(*2)	STD_GPIO				
1	1	1	0	0	0	1	-(*2)	-(*2)	-(*2)				STD_GPIO
1	1	1	0	0	1	0	-(*2)	-(*2)	-(*2)			STD_GPIO	
1	1	1	0	0	1	1	-(*2)	-(*2)	-(*2)			-(*2)	STD_GPIO
1	1	1	0	1	0	0	-(*2)	-(*2)	-(*2)		STD_GPIO		
1	1	1	0	1	0	1	-(*2)	-(*2)	-(*2)		-(*2)		STD_GPIO
1	1	1	0	1	1	0	-(*2)	-(*2)	-(*2)		-(*2)	STD_GPIO	
1	1	1	0	1	1	1	-(*2)	-(*2)	-(*2)		-(*2)	-(*2)	STD_GPIO
1	1	1	1	0	0	0	-(*2)	-(*2)	-(*2)	STD_GPIO			
1	1	1	1	0	0	1	-(*2)	-(*2)	-(*2)	-(*2)			STD_GPIO
1	1	1	1	0	1	0	-(*2)	-(*2)	-(*2)	-(*2)		STD_GPIO	
1	1	1	1	0	1	1	-(*2)	-(*2)	-(*2)	-(*2)		-(*2)	STD_GPIO
1	1	1	1	1	0	0	-(*2)	-(*2)	-(*2)	-(*2)	STD_GPIO		
1	1	1	1	1	0	1	-(*2)	-(*2)	-(*2)	-(*2)	-(*2)		STD_GPIO
1	1	1	1	1	1	0	-(*2)	-(*2)	-(*2)	-(*2)	-(*2)	STD_GPIO	
1	1	1	1	1	1	1	-(*2)	-(*2)	-(*2)	-(*2)	-(*2)	-(*2)	STD_GPIO

7-12.GPIO(APB-GPIO)

7.12.1 General Description

General-purpose port with interrupt function. The features are shown below.

- 13-bit (PortA 13 bit) general-purpose port.
- Can be used as external interrupt input.* Either level/edge or Low/High can be selected.
- Includes the debounce circuit in the input side.
- Can be used as return cause from SLEEP/DEEPSLEEP.

7.12.2 List of Registers

Address	Name	Symbol	R/W	Description
BASE+0x00	GPIO port data register	GPIOn_SWPORTA_DR	R/W	Determines the output value of the port when a general-purpose port is selected.
BASE+0x04	GPIO port data direction register	GPIOn_SWPORTA_DDR	R/W	Reads the input level of the port when a general-purpose port is selected.
BASE+0x30	GPIO interrupt enable register	GPIOn_INTEN	R/W	Enables the interrupt from the port.
BASE+0x34	GPIO interrupt mask register	GPIOn_INTMASK	R/W	Masks the interrupt from the port.
BASE+0x38	GPIO interrupt level register	GPIOn_INTTYPE_LEVEL	R/W	Specifies the type of the interrupt from the port.
BASE+0x3C	GPIO interrupt polarity register	GPIOn_INT_POLARITY	R/W	Specifies the polarity of the interrupt from the port.
BASE+0x40	GPIO interrupt status register	GPIOn_INTSTATUS	R	Indicates the interrupt status from the port.
BASE+0x44	GPIO RAW interrupt status register	GPIOn_RAW_INTSTATUS	R	Indicates the interrupt status before mask from the port.
BASE+0x48	GPIO debounce enable register	GPIOn_DEBOUNCE	R/W	Enables the debounce circuit of the port.
BASE+0x4C	GPIO interrupt clear register	GPIOn_PORTA_EOI	W	Clears the interrupt from the port.
BASE+0x50	GPIO external port register	GPIOn_EXT_PORT	R	Indicates the port input level.
BASE+0x60	GPIO level-sensitive synchronization enable	GPIOn_LS_SYNC	R/W	Sets the synchronization of the level interrupt from the port.

* n indicates a module number (n = 0, 1, 2, …).

* Base address of GPIOA (n = 0) is 0x4000A000.

* Base address of GPIOD (n = 3) is 0x4000A300. For ML7396B control. This is not a LSI port.

7.12.3 Description of Registers

7.12.3.1 Port Data Register (GPIOn_SWPORTA_DR): BASE + 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																SWPORTA_DR[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

SWPORTA_DR[n]: 0: Sets the output to the L level when GPIO[n] is a general-purpose port (primary function mode) and is in the output mode.

1: Sets the output to the H level when GPIO[n] is a general-purpose port (primary function mode) and is in the output mode.

* In case of GPIOA, bits 00 to 12 are enabled.

7.12.3.2 Port Data Direction Register (GPIOn_SWPORTA_DDR): BASE + 0x04

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																SWPORTA_DDR[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

SWPORTA_DDR[n]: 0: Sets GPIO[n] as input.

1: Sets GPIO[n] as output.

* In case of GPIOA, bits 00 to 12 are enabled.

7.12.3.3 GPIO Interrupt Enable Register (GPIOn_INTEN): BASE + 0x30

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																INT_EN[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Uses each bit of the port as an external interrupt source.

INT_EN[n]: 0: The interrupt from GPIO[n] is disabled.

1: The interrupt from GPIO[n] is enabled.

* In case of GPIOA, bits 00 to 12 are enabled.

7.12.3.4 GPIO Interrupt Mask Register (GPIOn_INTMASK): BASE + 0x34

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																INTMASK[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Masks the interrupt from each bit of the port.

INTMASK[n]: 0: The interrupt from GPIO[n] is not masked.

1: The interrupt from GPIO[n] is masked.

* In case of GPIOA, bits 00 to 12 are enabled.

7.12.3.5 GPIO Interrupt Level Register (GPIOn_INTTYPE_LEVEL): BASE + 0x38

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																INTTYPE_LEVEL[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Sets the type of the interrupt from each bit of the port.

INTTYPE_LEVEL[n]: 0: GPIOA[n] is used as a level interrupt.

1: GPIOA[n] is used as an edge interrupt.

* When the GPIO clock is stopped just like in DeepSleep mode, only the level interrupt is enabled.

* In case of GPIOD, only the level interrupt can be used during DeepSleep.

* In case of GPIOA, bits 00 to 12 are enabled.

7.12.3.6 GPIO Interrupt Polarity Register (GPIOn_INT_POLARITY): BASE + 0x3C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																INT_POLARITY [15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Sets the polarity of the interrupt from each bit of the port.

INT_POLARITY [n]: 0: L level or falling edge

1: H level or rising edge

* In case of GPIOA, bits 00 to 12 are enabled.

7.12.3.7 GPIO Interrupt Status Register (GPIOn_INTSTATUS): BASE + 0x40

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																INTSTATUS [15:0]																
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Indicates the status of the interrupt from each bit of the port.

* In case of GPIOA, bits 00 to 12 are enabled.

7.12.3.8 GPIO RAW Interrupt Status Register (GPIOn_RAW_INTSTATUS): BASE + 0x44

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																RAW_INTSTATUS [15:0]																
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Indicates the status (before mask) of the interrupt from each bit of the port.

* In case of GPIOA, bits 00 to 12 are enabled.

7.12.3.9 GPIO Debounce Enable Register (GPIOn_DEBOUNCE): BASE + 0x48

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																DEBOUNCE [15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Enables the debounce circuit of the port.

DEBOUNCE[n]: 0: The debounce circuit is disabled. (bypassed)

1: The debounce circuit is enabled.

* When the CPU returns by a level interrupt with the GPIO clock stopped like in DeepSleep mode, the debounce circuit should be disabled.

* In case of GPIOA, bits 00 to 12 are enabled.

7.12.3.10 GPIO Interrupt Clear Register (GPIOn_PORTA_EOI): BASE + 0x4C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																EOI [15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Clears the interrupt from the port.

EOI [n]: 0: The interrupt is not cleared.

1: The interrupt is cleared.

* In case of GPIOA, bits 00 to 12 are enabled.

7.12.3.11 GPIO Port External Input Register (GPIOn_EXT_PORTA): BASE + 0x50

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																EXT_PORTA [7:0]																
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Reads the state of the GPIO port.

EXT_PORT [n]: 0: Indicates that this is L level.

1: Indicates that this is H level.

* In case of GPIOA, bits 00 to 12 are enabled.

7.12.3.12 GPIO Level-Sensitive Synchronization Enable Register (GPIOn_LS_SYNC): BASE + 0x60

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															SL
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

SL(Synchronization Level): 0: Outputs a level-sensitive interrupt without synchronizing it.

1: Outputs a level-sensitive interrupt by synchronizing it.

- To use the level interrupt from GPIO Portm at return from DEEPSLEEP, be sure to set this to 0.
- To use the level interrupt from GPIO Portm at return from SLEEP or during normal operation, set this to 1.

7-13.Standard GPIO (Single-cycle I/O)

7.13.1 General Description

General-purpose port function module connected to a single-cycle I/O which can be accessed in one cycle.

* The interrupt function is same as that of APB-GPIO (refer to the chapter about APB-GPIO).

7.13.2 List of Registers

Table 7-13-1 List of STDGPIO Registers

Address	Name	Symbol	R/W	Description
BASE + 0x000- BASE + 0x3FF	Standard GPIOn data register	STDGPIOm_DATA	R/W	When reading, the status of general-purpose ports is read. In output mode, the set values are output to the general-purpose ports.
BASE + 0x400	Standard GPIOn input/output control register	STDGPIOm_DIR	R/W	Controls the direction of input/output for each general-purpose port.

(m=A)

m=A: BASE=0x5C000000

7.13.3 Description of Registers

7.13.3.1 Standard GPIOm Data Register (STDGPIOm_DATA)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPIOmDATA [31:0]																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

* In ML7416, only the lower four bits are assigned as ports.
* GPIOmDATA[n] is assigned to the external pin Pmn (m=A,B,.., n=0,1,…) of LSI as the quintic function.

7.13.3.1.1 Word/Half-Word Access

GPIOmDATA[n]: When GPIOmDIR[n] is 0 (input), the value of the external pin Pmn is read.
When GPIOmDIR[n] is 1 (output), the value set to GPIOmDATA[n] is output from the external pin Pmn.
(m=A,B,C,..,nx=0..3)

7.13.3.1.2 Byte Write

Performs a bit set/bit clear operation for the byte lane selected in HADDR[1:0] by using HADDR[9:2] as mask. Other byte lanes are not changed.

(Example) When writing the byte 0x02 to 0x5C000009 (HADDR[9:2] = 0x02, HADDR[1:0] = 0x01, HWDATA = 0x200, HSIZE = 0b000)

GPIOADATA[15:8] = HADDR[15:8] & ~(HADDR[9:2]) | HWDATA[15:8] & HADDR[9:2]
(GPIOADATA[9] bit is set)

(Example) When writing the byte 0x00 to 0x5C000009 (HADDR[9:2] = 0x02, HADDR[1:0] = 0x01, HWDATA = 0x0, HSIZE = 0b000)

GPIOADATA[15:8] = HADDR[15:8] & ~(HADDR[9:2]) | HWDATA[15:8] & HADDR[9:2]
(GPIOADATA[9] bit is cleared)

7.13.3.1.3 Byte Read

Performs a bit read operation for the byte lane selected in HADDR[1:0] by using HADDR[9:2] as mask. For the other byte lanes and the masked bits, 0 is read.

(Example) When reading the byte from 0x5C000009 (HADDR[9:2] = 0x02, HADDR[1:0] = 0x01, HSIZE = 0b000)

HRDATA[15:8] = GPIOADATA[15:8] & HADDR[9:2]

7.13.3.2 Standard GPIOm Input/Output Control Register (STDGPIOm_DIR: m = A)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPIOmDIR[31:0]																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

GPIOmDIR[n]: 0: Set input mode.
1: Set output mode.
(m=A, n=0..3)

* In ML7416, only the lower four bits are assigned as ports.

7-14.Timer

7.14.1 General Description

32-bit general-purpose timer. The features are shown below.

- If the CPU is stopped by the debugger, the counter operation of timer is stopped.
- Can be used as 64-bit timer by the cascade connection. The combination of TimerB and TimerC or TimerD and TimerE is available.

7.14.2 List of Registers

Address [H]	Name	Symbol	R/W	Initial value [H]	Description
BASE+0x00	Timer 1 load count register	TMRm_TIMER1LOADCOUNT	R/W	0x00000000	Sets the timer load value.
BASE+0x04	Timer 1 current value register	TMRm_TIMER1CURRENTVALUE	R	0xFFFFFFFF	Indicates the timer count value.
BASE+0x08	Timer 1 control register	TMRm_TIMER1CONTROLREG	R/W	0x00000000	Controls the operation of timer.
BASE+0x0C	Timer 1 interrupt clear register	TMRm_TIMER1EOI	R	0x00000000	Clears the interrupt from the timer.
BASE+0x10	Timer 1 interrupt status register	TMRm_TIMER1INTSTATUS	R	0x00000000	Indicates the interrupt status of the timer (after mask).
BASE+0xA8	Timer RAW interrupt status register	TMRm_TIMERSRAWINTSTATUS	R	0x00000000	Indicates the interrupt status of the timer (before mask).
BASE+0xAC	Timer component version register	TMRm_TIMERS_COMP_VERSION	R	0x3230382A	Timer component version

* m indicates a module number (m = A, B, C, D, E, F).

* Base address of TimerA (m = A) is 0x40001000.

* Base address of TimerB (m = B) is 0x40041000.

* Base address of TimerC (m = C) is 0x40041400.

* Base address of TimerD (m = D) is 0x40041800.

* Base address of TimerE (m = E) is 0x40041C00.

7.14.3 Description of Registers

7.14.3.1 Timer 1 Load Count Register: BASE + 0x00 (TMRm_TIMER1LOADCOUNT)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIMER1LOADCOUNT																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Set the value to be loaded to the timer 1. The countdown operation starts at the value set to this register.

7.14.3.2 Timer 1 Current Value Register: BASE + 0x04 (TMRm_TIMER1CURRENTVALUE)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIMER1CURRENTVALUE																															
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The current value of Timer 1.

7.14.3.3 Timer 1 Control Register: BASE + 0x08 (TMRm_TIMER1CONTROLREG)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																													INT_MASK	MODE	TIMER_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

INT_MASK	Masks the interrupt. 0: No mask 1: Mask
MODE	Sets the operation mode of timer. When the timer value reaches 0, the following operations are performed according to the value of this bit: 0: Free running mode The countdown operation starts at the maximum value of Timer 1. 1: User-defined count mode The countdown operation starts at the value set to the timer 1 load count register.
TIMER_EN	Enable the timer. 0: disable 1: enable

Writing 1 to bit3 is prohibited.

7.14.3.4 Timer 1 Interrupt Clear Register: BASE + 0x0C (TMRm_TIMER1EOI)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															EOI
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When this register is read, the interrupt from Timer 1 is cleared.

Always 0 is read when reading.

7.14.3.5 Timer 1 Interrupt Status Register: BASE + 0x10 (TMRm_TIMER1INTSTATUS)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															STATUS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Displays the status of the interrupt from Timer 1 (after mask).

7.14.3.6 Timer RAW Interrupt Status Register: BASE + 0xA8 (TMRm_TIMERSRAWINTSTATUS)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NUM_TIMERS-1:0																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	Description
n (n=0, … , NUM_TIMERS-1)	TIMERn_RAWSTATUS	Indicates the status of the interrupt from Timer n (before mask).

Displays the status (before mask) of the interrupt from a channel of the timer corresponding to each bit.

7.14.3.7 Timer Component Version Register: BASE + 0xAC (TMRm_TIMERS_COMP_VERSION)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIMERS_COMPONET _VERSION																															
Initial value	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0	1	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Component version ID.

7.14.4 Description of Operation

The starting procedure is as follows:

(1) Set the appropriate timer interrupt of the interrupt controller (NVIC) to enabled.
(2) Set the desired value to the Timer1 load count register.
(3) Set the MODE bit of the Timer1 control register to a desired value, the mask bit to 0, and the TIMER_EN bit to 1 to start the countdown operation.

Example) When the following setting is used, a timer interrupt is generated 1 second after the timer start.

SUB_CLK_SEL=0(XTAL32kHz(32.768 kHz))
TIMERm_CLK_SEL=1 (sub clock)
TMRm_DIV=0x0 (no division)
TMRm_TIMER1CONTROLREG.MODE=1
TMRm_TIMER1LOADCOUNT=0x7D00

The stopping procedure is as follows:

(1) Read the Timer1 interrupt clear register.
(2) Set the appropriate timer interrupt of the interrupt controller (NVIC) to disabled.
(3) Set the appropriate timer bit of the peripheral reset register (SYSCON_PRST_CON) to 1 to perform initialization.
(4) Confirm that the appropriate timer bit of the peripheral reset register is set to 0.

7-15.EXT_Timer

7.15.1 General Description

Six channels of 32-bit general-purpose timers are included. The interrupt number is IRQ[19], which is common to these six channels.

7.15.2 List of Registers

Address [H]	Name	Symbol	R/W	Initial value [H]	Description
BASE+0x00	Timer n load count register	EXTTMRm_TIMERnLOADCOUNT	R/W	0x00000000	Sets the timer load value.
BASE+0x04	Timer n current value register	EXTTMRm_TIMERnCURRENTVALUE	R	0xFFFFFFFF	Indicates the timer count value.
BASE+0x08	Timer n control register	EXTTMRm_TIMERnCONTROLREG	R/W	0x00000000	Controls the operation of timer.
BASE+0x0C	Timer n interrupt clear register	EXTTMRm_TIMERnEOI	R	0x00000000	Clears the interrupt from the timer.
BASE+0x10	Timer n interrupt status register	EXTTMRm_TIMERnINTSTATUS	R	0x00000000	Indicates the interrupt status of the timer (after mask).
0x400420A0	EXT timer interrupt status register	EXTTMRm_TIMERSINTSTATUS	R	0x00000000	Indicates the interrupt status of the EXT timer (after mask).
0x400420A4	EXT timer interrupt clear register	EXTTMRm_TIMERSEOI	R	0x00000000	Clears the interrupt from the EXT timer.
0x400420A8	EXT timer RAW interrupt status register	EXTTMRm_TIMERSRAWINTSTATUS	R	0x00000000	Indicates the interrupt status of the EXT timer (before mask).
0x400420AC	EXT timer component version register	EXTTMRm_TIMERS_COMP_VERSION	R	0x3230382A	EXT timer component version

* n indicates a channel number (n = 1, 2, 3, 4, 5, 6).

* Base address of EXT_Timer1 (n = 1) is 0x40042000.

* Base address of EXT_Timer2 (n = 2) is 0x40042014.

* Base address of EXT_Timer3 (n = 3) is 0x40042028.

* Base address of EXT_Timer4 (n = 4) is 0x4004203C.

* Base address of EXT_Timer5 (n = 5) is 0x40042050.

* Base address of EXT_Timer6 (n = 6) is 0x40042064.

7.15.3 Description of Registers

7.15.3.1 Timer n Load Count Register: BASE + 0x00 (EXTTMRm_TIMERnLOADCOUNT)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIMERnLOADCOUNT																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Set the value to be loaded to the timer n. The countdown operation starts at the value set to this register.

7.15.3.2 Timer n Current Value Register: BASE + 0x04 (EXTTMRm_TIMERnCURRENTVALUE)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIMERnCURRENTVALUE																															
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The current value of Timer n.

7.15.3.3 Timer n Control Register: BASE + 0x08 (EXTTMRm_TIMERnCONTROLREG)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																													INT_MASK	MODE	TIMER_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

INT_MASK	Masks the interrupt. 0: No mask 1: Mask
MODE	Sets the operation mode of timer. When the timer value reaches 0, the following operations are performed according to the value of this bit: 0: Free running mode The countdown operation starts at the maximum value of Timer 1. 1: User-defined count mode The countdown operation starts at the value set to the timer 1 load count register.
TIMER_EN	Enable the timer. 0: disable 1: enable

Writing 1 to bit3 is prohibited.

7.15.3.4 Timer n Interrupt Clear Register: BASE + 0x0C (EXTTMRm_TIMERnEOI)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															EOI
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When this register is read, the interrupt from Timer n is cleared.

Always 0 is read when reading.

7.15.3.5 Timer n Interrupt Status Register: BASE + 0x10 (EXTTMRm_TIMERnINTSTATUS)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															STATUS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Displays the status of the interrupt from Timer n (after mask).

7.15.3.6 EXT Timer Interrupt Status Register: 0x400420A0 (EXTTMRm_TIMERINTSTATUS)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										TIMERn_STATUS					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Displays the status of the interrupt from EXT Timer (after mask).

7.15.3.7 EXT Timer Interrupt Clear Register: 0x400420A4 (EXTTMRm_TIMEREOI)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										TIMERn_EOI					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When this register is read, the interrupt from EXT Timer n is cleared.

Always 0 is read when reading.

7.15.3.8 EXT Timer RAW Interrupt Status Register: 0x400420A8 (EXTTMRm_TIMERSRAWINTSTATUS)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NUM_TIMERS-1:0																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	Description
n (n=0, … , NUM_TIMERS-1)	TIMERn_RAWSTATUS	Indicates the status of the interrupt from EXT Timer n (before mask).

Displays the status (before mask) of the interrupt from a channel of the timer corresponding to each bit.

7.15.3.9 EXT Timer Component Version Register: 0x400420AC (EXTTMRm_TIMERS_COMP_VERSION)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIMERS_COMPONET _VERSION																															
Initial value	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0	1	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Component version ID.

7.15.4 Description of Operation

The operation procedure is as follows:

(1) Clear the TIMER_EN bit of the EXT_Timer1 control register to 0.

(2) Set the MODE bit of the EXT_Timer1 control register to the desired value.

(3) Set the desired value to the EXT_Timer1 load count register.

(4) Set the TIMER_EN bit of the EXT_Timer1 control register to 1 and start the countdown operation.

7-16.RTC

7.16.1 General Description

Real time clock with perpetual calendar which can be read/written from a second unit. The features are shown below.

- It is possible to set, correct, and read the time.
- Can be used as interrupt source.

7.16.2 List of Registers

Address [H]	Name	Symbol	R/W	Initial value [H]	Description
0x40003000	S1 register	RTC_S1	RW	0xXXXXXXXX	One-second digit register
0x40003004	S10 register	RTC_S10	RW	0xXXXXXXXX	10-second digit register
0x40003008	MI1 register	RTC_MI1	RW	0xXXXXXXXX	One-minute digit register
0x4000300C	MI10 register	RTC_MI10	RW	0xXXXXXXXX	10-minute digit register
0x40003010	H1 register	RTC_H1	RW	0xXXXXXXXX	One-hour digit register
0x40003014	H10 register	RTC_H10	RW	0xXXXXXXXX	PM/AM 10-hour digit register
0x40003018	D1 register	RTC_D1	RW	0xXXXXXXXX	One-day digit register
0x4000301C	D10 register	RTC_D10	RW	0xXXXXXXXX	10-day digit register
0x40003020	MO1 register	RTC_MO1	RW	0xXXXXXXXX	One-month digit register
0x40003024	MO10 register	RTC_MO10	RW	0xXXXXXXXX	10-month digit register
0x40003028	Y1 register	RTC_Y1	RW	0xXXXXXXXX	One-year digit register
0x4000302C	Y10 register	RTC_Y10	RW	0xXXXXXXXX	10-year digit register
0x40003030	W register	RTC_W	RW	0xXXXXXXXX	Week register
0x40003034	Control register D	RTC_CD	RW	0xXXXXXXXX	RTC control
0x40003038	Control register E	RTC_CE	RW	0xXXXXXXXX	RTC control
0x4000303C	Control register F	RTC_CF	RW	0xXXXXXXXX	RTC control
0x40003040	FT_S1 register	RTC_FT_S1	RW	0xXXXXXXXX	One-second digit register for time-designated interrupt
0x40003044	FT_S10 register	RTC_FT_S10	RW	0xXXXXXXXX	Ten-second digit register for time-designated interrupt

0x40003048	FT_MI1 register	RTC_FT_MI1	RW	0xXXXXXXXX	One-minute digit register for time-designated interrupt
0x4000304C	FT_MI10 register	RTC_FT_MI10	RW	0xXXXXXXXX	Ten-minute digit register for time-designated interrupt
0x40003050	FT_H1 register	RTC_FT_H1	RW	0xXXXXXXXX	One-hour digit register for time-designated interrupt
0x40003054	FT_H10 register	RTC_FT_H10	RW	0xXXXXXXXX	PM/AM 10-hour digit register for time-designated interrupt
0x40003058	FT_D1 register	RTC_FT_D1	RW	0xXXXXXXXX	One-day digit register for time-designated interrupt
0x4000305C	FT_D10 register	RTC_FT_D10	RW	0xXXXXXXXX	Ten-day digit register for time-designated interrupt
0x40003060	FT_MO1 register	RTC_FT_MO1	RW	0xXXXXXXXX	One-month digit register for time-designated interrupt
0x40003064	FT_MO10 register	RTC_FT_MO10	RW	0xXXXXXXXX	10-month digit register for time-designated interrupt
0x40003068	FT_Y1 register	RTC_FT_Y1	RW	0xXXXXXXXX	One-year digit register for time-designated interrupt
0x4000306C	FT_Y10 register	RTC_FT_Y10	RW	0xXXXXXXXX	Ten-year digit register for time-designated interrupt

7.16.3 Description of Registers

7.16.3.1 S1 register: 0x40003000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												S1			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Indicates the first digit of second.

7.16.3.2 S10 register: 0x40003004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													S10		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

Indicates the second digit of second.

7.16.3.3 MI1 register: 0x40003008

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												M1			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Indicates the first digit of minute.

7.16.3.4 MI10 register: 0x4000300C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													M10		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

Indicates the second digit of minute.

7.16.3.5 H1 register: 0x40003010

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													H1		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Indicates the first digit of hour.

7.16.3.6 H10 register: 0x40003014

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													AM PM	H10	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

H10: Indicates the second digit of hour.

AM/PM: This bit is ignored when 24-hour time is used.

Indicates AM or PM when 12-hour time is used. (0: AM, 1: PM)

7.16.3.7 D1 register: 0x40003018

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												D1			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Indicates the first digit of day.

7.16.3.8 D10 register: 0x4000301C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														D10	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Indicates the second digit of day.

7.16.3.9 M1 Register: 0x40003020

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												MO1			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Indicates the first digit of month.

7.16.3.10 M10 Register: 0x40003024

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															MO10
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Indicates the second digit of month.

7.16.3.11 Y1 register: 0x40003028

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												Y1			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Indicates the first digit of year.

7.16.3.12 Y10 register: 0x4000302C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												Y10			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Indicates the second digit of year.

7.16.3.13 W register: 0x40003030

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																													W		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

Displays the day of the week. The days of the week are assigned as shown in "Table Correspondence between W Register and Days of the Week".

Table Correspondence between W Register and Days of the Week

W	Day
000	Sunday
001	Monday
010	Tuesday
011	Wednesday
100	Thursday
101	Friday
110	Saturday

7.16.3.14 Control Register D: 0x40003034

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																											FT_IRQ	ADJ30S	IRQ	BUSY	HOLD
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

HOLD: This is used to inhibit a carry of the counter for digits larger than or equal to the one-second digit when reading or writing the S1 to W registers.

BUSY: Indicates whether the S1 to W registers can be rewritten or not. The register can be rewritten when the HOLD bit is 1 and the BUSY bit is 0.

If the HOLD bit is used for reading, reading is possible when the BUSY bit is 0.

IRQ: Indicates presence or absence of interrupt request.

0: No interrupt exists

1: An interrupt exists

ADJ30S: When 1 is written to this bit, 30-second correction is performed. If the indicated second by S1 and S10 is in the range of 0 to 29, the S1 and S10 registers are reset to 0; if it is in the range of 30 to 59, the S1 and S10 registers are reset to 0 and +1 is added to the MI1 digit.

The S1 to W registers (addresses 0x40003000 to 0x40003030) cannot be read and write for a period of 190 μs since "1" is written to this bit. The value "1" is kept for a period of 190 μs since it is written to this bit and then automatically returns to "0".

FT_IRQ: Indicates whether a time-designated interrupt exists.

0: No interrupt exists

1: An interrupt exists

7.16.3.15 Control Register E: 0x40003038

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																											FT_MASK	T1	T0	ITRPT/STND	MASK
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

MASK: Masks an interrupt request. (1: Mask, 0: No mask)

ITRPT/STND: Sets the operation of interrupt signal.

1: Outputs the interrupt request.

The interrupt request is asserted when it is not masked by the interrupt timing.

It is deasserted by writing 0 to the IRQ FLAG bit, or setting 1 to the interrupt mask request.

The interrupt timing (cycle) is determined by the values of T1 and T0 bits.

0: Outputs a fixed-cycle waveform.

The interrupt request is asserted when it is not masked by the interrupt timing.

It is deasserted by automatic return, writing 0 to the IRQ FLAG bit, or setting 1 to the interrupt mask request.

The interrupt timing (cycle) and the time to the automatic return are determined by the values of T1 and T0 bits.

T1, T0: Sets the interrupt cycle and the cycle of fixed-cycle waveform.

T1	T0	Period	L level DUTY when the ITRPT/STND bit is 0
0	0	1/64 second	1/2
0	1	1 second	1/128
1	0	1 minute	1/7680
1	1	1 hour	1/460800

FT_MASK: Masks a time-designated interrupt.

0: Mask

1: No mask

7.16.3.16 Control Register F: 0x4000303C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												TEST	24/12	STOP	REST
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

REST: Resets the internal counter value less than a second. If 1 is written to this bit, the counter is reset and maintains that state until 0 is written.

STOP: If 1 is written to this bit, all timings are stopped. If 0 is written, timings are resumed.

24/12: Selects 24-hour time or 12-hour time.

1: 24-hour time

0: 12-hour time

TEST: Bit for testing. If this bit is set to 1, the carry of S1 register that is usually 1 Hz is changed to 4.096 kHz. The REST operation to reset the value less than a second and the STOP operation are disabled.

7.16.3.17 FT_S1 register: 0x40003040

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												FT_S1			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Indicates the first digit of second of time-designated interrupt.

Disable the time-designated interrupt before setting the value of this register.

7.16.3.18 FT_S10 register: 0x40003044

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													FT_S10		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

Indicates the second digit of second of time-designated interrupt.

Disable the time-designated interrupt before setting the value of this register.

7.16.3.19 FT_MI1 register: 0x40003048

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												FT_M1			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Indicates the first digit of minute of time-designated interrupt.

Disable the time-designated interrupt before setting the value of this register.

7.16.3.20 FT_MI10 register: 0x4000304C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													FT_M10		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

Indicates the second digit of minute of time-designated interrupt.

Disable the time-designated interrupt before setting the value of this register.

7.16.3.21 FT_H1 register: 0x40003050

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												FT_H1			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Indicates the first digit of hour of time-designated interrupt.

Disable the time-designated interrupt before setting the value of this register.

7.16.3.22 FT_H10 register: 0x40003054

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												FT_AMPM		FT_H10	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

H10: Indicates the second digit of hour of time-designated interrupt.

AM/PM: This bit is ignored when 24-hour time is used.

Indicates AM or PM when 12-hour time is used. (0: AM, 1: PM)

Disable the time-designated interrupt before setting the value of this register.

7.16.3.23 FT_D1 register: 0x40003058

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												FT_D1			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Indicates the first digit of day of time-designated interrupt.

Disable the time-designated interrupt before setting the value of this register.

7.16.3.24 FT_D10 register: 0x4000305C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														FT_D10	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Indicates the second digit of day of time-designated interrupt.

Disable the time-designated interrupt before setting the value of this register.

7.16.3.25 FT_M1 Register: 0x40003060

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												FT_MO1			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Indicates the first digit of month of time-designated interrupt.

Disable the time-designated interrupt before setting the value of this register.

7.16.3.26 FT_M10 Register: 0x40003064

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															FT_MO10
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Indicates the second digit of month of time-designated interrupt.

Disable the time-designated interrupt before setting the value of this register.

7.16.3.27 FT_Y1 register: 0x40003068

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												FT_Y1			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Indicates the first digit of year of time-designated interrupt.

Disable the time-designated interrupt before setting the value of this register.

7.16.3.28 FT_Y10 register: 0x4000306C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												FT_Y10			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Indicates the second digit of year of time-designated interrupt.

Disable the time-designated interrupt before setting the value of this register.

7.16.4 Description of Operation

7.16.4.1 Procedure of Initial Operation

Figure 7--16-1 Procedure of RTC Initial Operation shows the initial operation. As the register values are undefined at power-on, be sure to follow the procedures described below to initialize the registers.

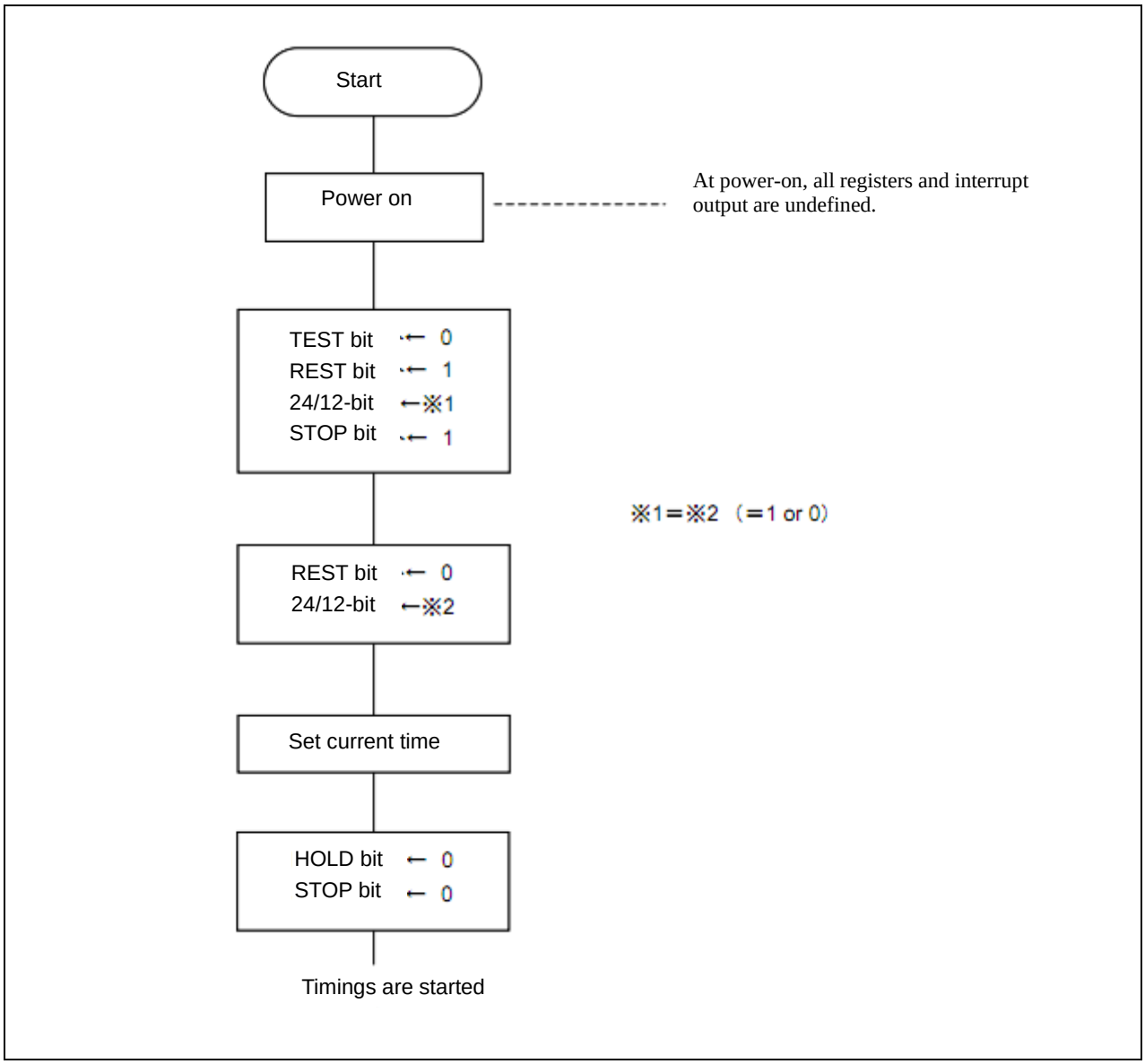


Figure 7--16-1 Procedure of RTC Initial Operation

7.16.4.2 Procedure of Reading/Writing S1 to W Registers

Figure 7--16-2 Procedure of Reading/Writing RTC S1 to W Registers shows the procedure of reading and writing the S1 to W registers when using the HOLD bit. (* For the idling time in the figure, 62 usec or longer should be secured.)

Figure 7-16-3 Procedure of Reading S1 to W Registers (When Not Using HOLD Bit) shows the procedure of reading the S1 to W registers when not using the HOLD bit.

Figure 7-16-4 Procedure of Reading S1 to W Registers (Through Reference to IRQ) shows the method of reading the S1 to W registers through reference to the IRQ bit.

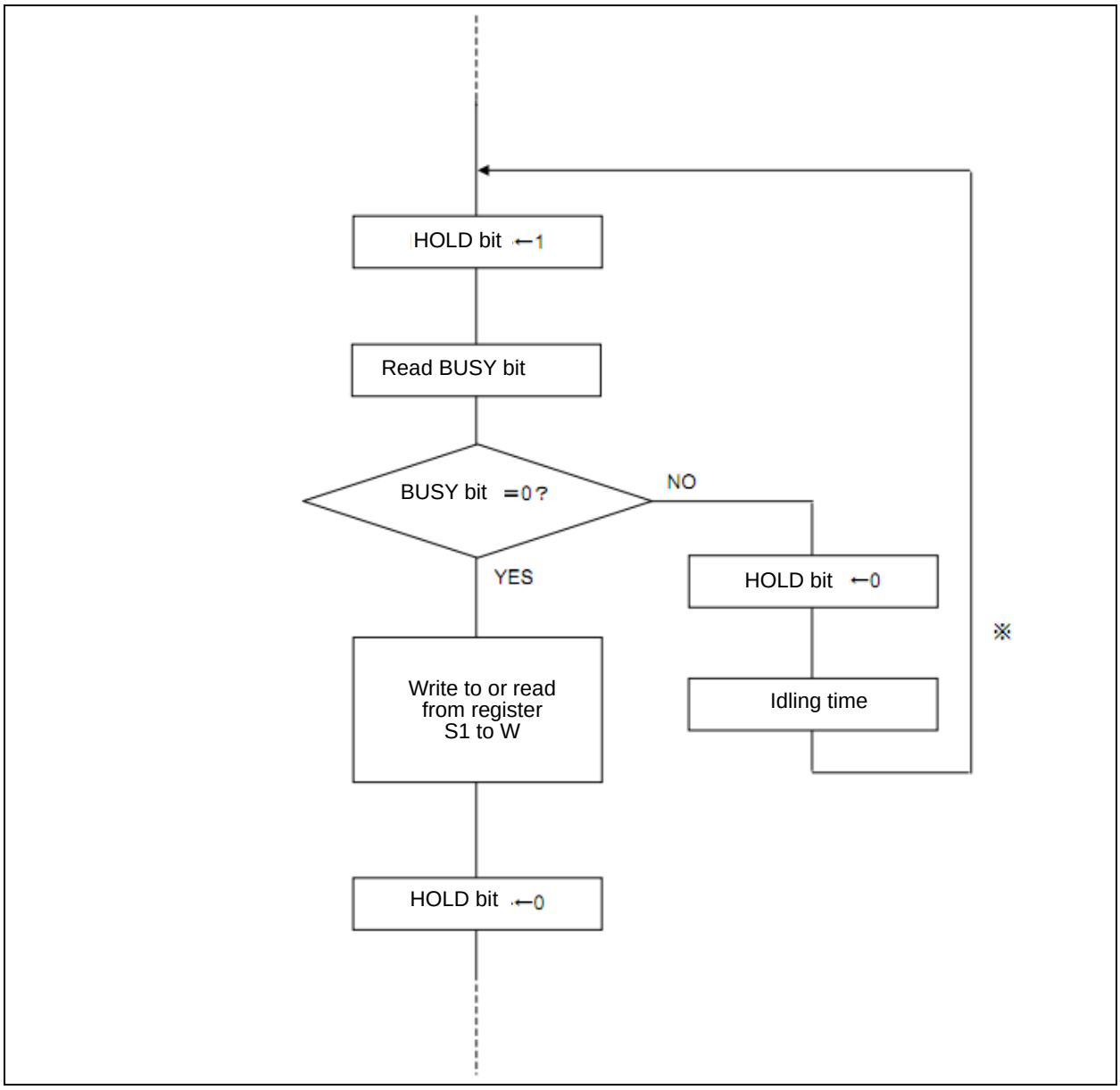


Figure 7--16-2 Procedure of Reading/Writing RTC S1 to W Registers (When Using HOLD Bit)

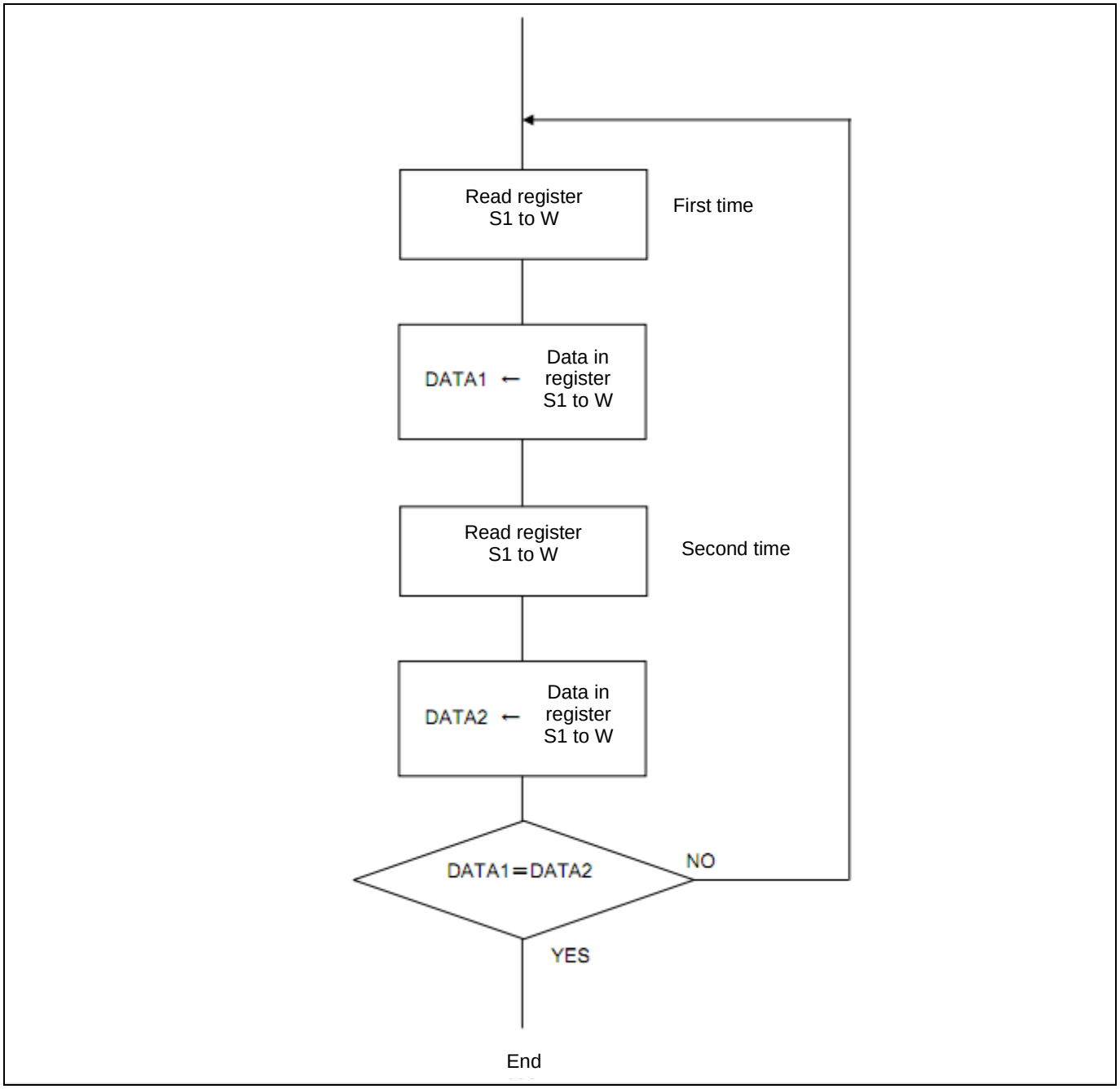


Figure 7-16-3 Procedure of Reading S1 to W Registers (When Not Using HOLD Bit)

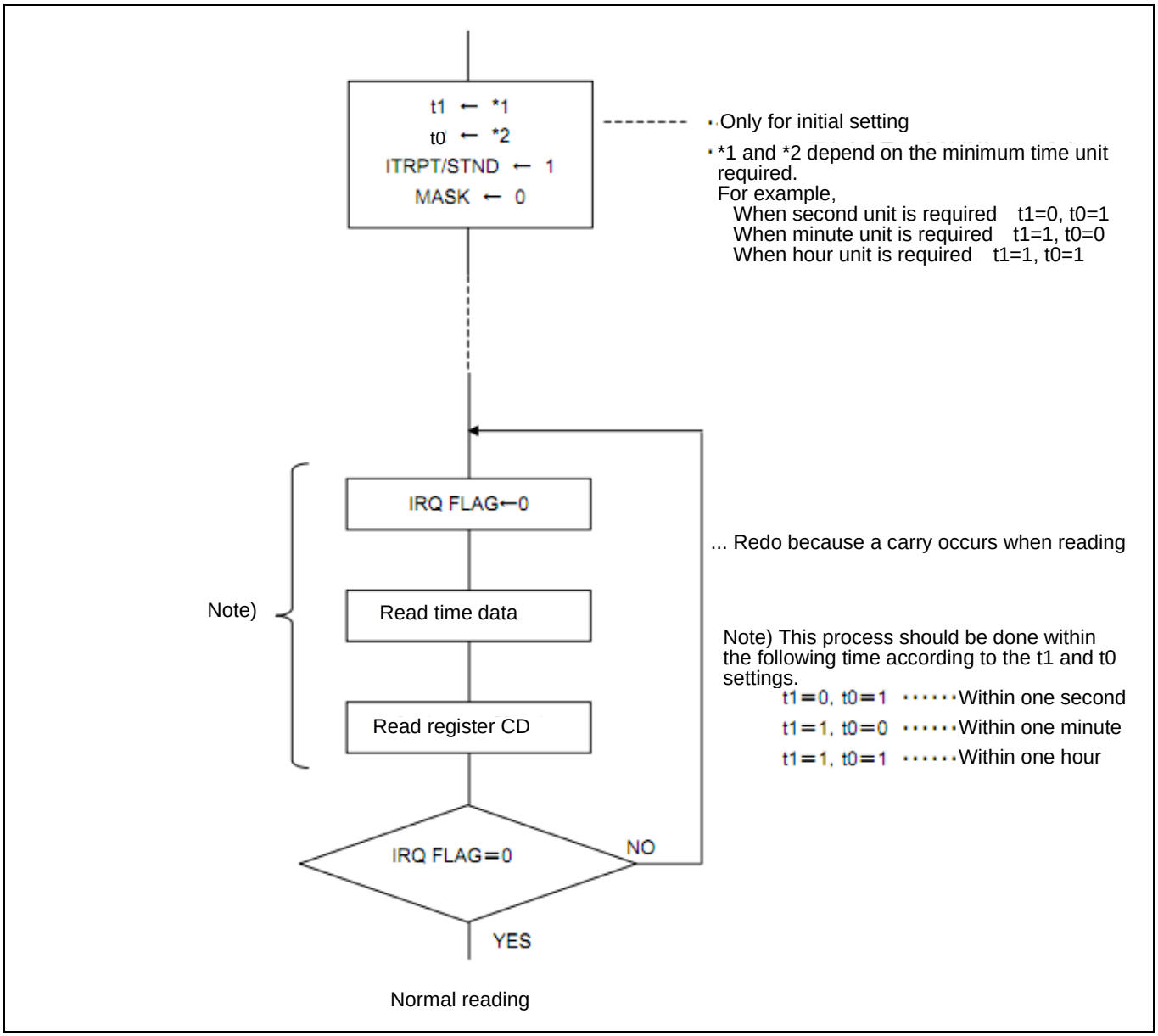


Figure 7-16-4 Procedure of Reading S1 to W Registers (Through Reference to IRQ)

7.16.4.3 Procedure of Writing ADJ30S Register

Figure 7-16-5 Procedure of Writing ADJ30S shows the procedure of writing the ADJ30S bit (method 1 and method 2).

In the method 1, set 1 to the ADJ30S bit, and then poll the register value until the ADJ30S bit is cleared.

In the method 2, wait for 190 µsec which guarantees that the ADJ30S bit is cleared, instead of polling the register value.

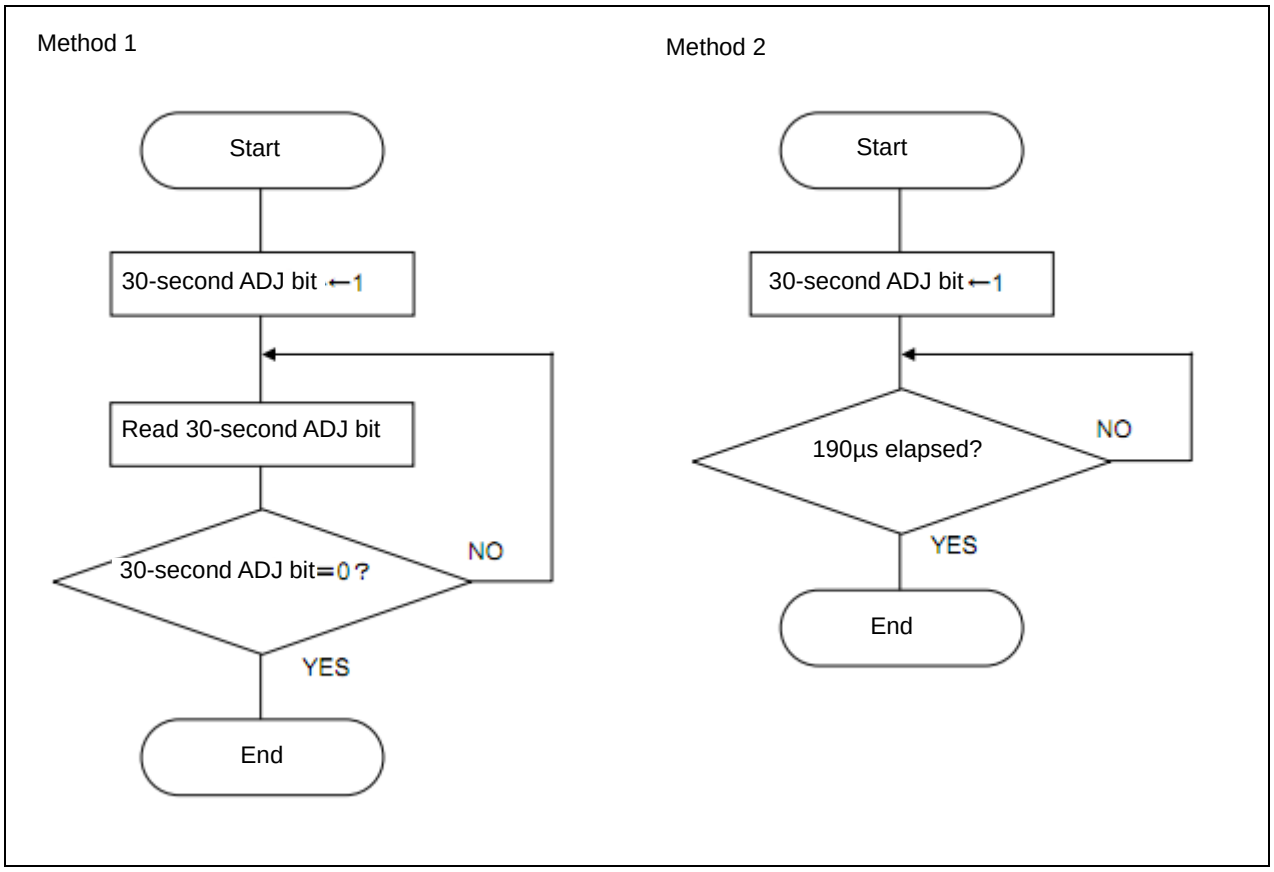


Figure 7-16-5 Procedure of Writing ADJ30S

7-17. Flexible Timer (FTM)

7.17.1 General Description

16-bit multifunction timer. The features are shown below.

- Auto-reload (ART)
- Compare out (CMO)
- Pulse width modulation (PWM)
- Capture (CAP)

7.17.2 List of Registers

Address [H]	Name	Symbol	R/W	Initial value [H]	Description
0x40002000	Timer control register 0	FTMm_FTM0CON	R/W	0x00000000	
0x40002004	Timer status register 0	FTMm_FTM0ST	R/W	0x00000000	
0x40002008	Timer counter 0	FTMm_FTM0C	R/W	0x00000000	
0x4000200C	Timer register 0	FTMm_FTM0R	R/W	0x00000000	
0x40002010	General-purpose timer register 0	FTMm_FTM0GR	R/W	0x00000000	
0x40002014	Timer input/output level register 0	FTMm_FTM0IOLV	R/W	0x00000000	
0x40002018	Timer output register 0	FTMm_FTM0OUT	R/W	0x00000000	
0x4000201C	Timer interrupt enable register 0	FTMm_FTM0IER	R/W	0x00000000	
0x40002020	Timer clock control 0	FTMm_FTM0CKCON	R/W	0x00000000	
0x40002200	Timer enable register	FTMm_FTMEN	R/W	0x00000000	
0x40002204	Timer disable register	FTMm_FTMDIS	W	0x00000000	

* n indicates a module number (n = A).

7.17.3 Description of Registers

7.17.3.1 Timer Control Register 0 (FTMm_FTMnCON: n = 0~~ to 7~~)

Address: BASE + 0x0040*n

Access: R/W

Access size: 32 bits

Initial value: 0x00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*
Access	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	MOD[1:0]		FTMCLK[2:0]		
Access	-	-	-	-	-	-	-	-	-	-	-	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

*: Reserved bit for future extension. "0" is read when reading. Write "0" when writing.

[Description of Register]

FTMnCON is a read/write register that sets the operation mode and the counter clock of Timer n. The configuration of FTMnCON is shown below. (n=0~~-7~~)

[Description of bits]

- **FTMCLK[2:0]** (bits 0 to 2)

  Selects the counter clock.

  • FTMnCON (n=0~~,1,2~~):

FTMCLK[2:0]	Description
000	Operates with system clock.
001	Operates with 2 dividing of system clock.
010	Operates with 4 dividing of system clock.
011	Operates with 8 dividing of system clock.
100	Operates with 16 dividing of system clock.
101	Operates with 32 dividing of system clock.

110	Rising edge of FTMCLK[n]
111	Falling edge of FTMCLK[n]

* The source clock of FTMCLK[n] is determined by the timer clock control register. For details, please refer to "07.17.3.9 Timer Clock Control Register 0 (FTMm_FTMnCKCON: **n=0 to 7)**".

- **MOD[1:0]** (bits 3 to 4)

MOD[1:0]	Description
00	Auto-reload (ART) mode
01	Compare out (CMO) mode
10	Pulse width modulation (PWM) mode
11	Capture (CAP) mode

7.17.3.2 Timer Status Register 0 (FTMm_FTMnST: n = 0~~ to 7~~)

Address: BASE + 0x040*n + 0x004

Access: R/W

Access size: 32 bits

Initial value: 0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*
Access	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	OVF	CM_CAPEV
Access	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

*: Reserved bit for future extension. "0" is read when reading. Write "0" when writing.

Since the OVF and CM/CAPEV bits are not automatically reset once they are set, they must be reset by a program. Each bit is reset when "1" is written, and the state of each bit does not change when "0" is written. If a flag setting by the interrupt request conflicts with a flag reset by the write, the flag setting has priority.

[Description of Register]

FTMnST is a read/write register that indicates the status for each channel. The configuration of FTMnST is shown below. (n=0-7)

[Description of bits]

- **CM_CAPEV** (bit 0)

  • In CMO mode:

  This bit is set when the value of the timer counter FTMnC coincides with the value of the general-purpose timer register FTMnGR. At this time, an interrupt request is generated. This bit is cleared when "1" is written to it.

CM_CAPEV	Description
0	The value of the timer counter FTMnC differs from the value of the general-purpose timer register
1	The value of the timer counter FTMnC coincides with the value of the general-purpose timer register

  • In CAP mode:

  This bit is set when input from the timer input (FTMINn) changes (capture trigger). At this time, an interrupt request is generated. This bit is cleared when "1" is written to it.

CM_CAPEV	Description
0	No capture trigger is generated on the timer input pin
1	A capture trigger is generated on the timer input pin

  • In ART/PWM mode:

  No value of this bit changes.

- **OVF** (bit 1)

  This bit is set when the timer counter overflows. At this time, an interrupt request is generated. This bit is cleared when "1" is written to it.

OVF	Description
0	No overflow generation
1	Overflow generation

7.17.3.3 Timer Counter 0 (FTMm_FTMnC: n = 0~~ to 7~~)

Address: BASE + 0x040*n + 0x008

Access: R/W

Access size: 32 bits

Initial value: 0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*
Access	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Symbol name	FTMnC [15:0]															
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

*: Reserved bit for future extension. "0" is read when reading. Write "0" when writing.

[Description of Register]

FTMnC is a 16-bit read/write counter that is counted up by the counter clock selected by the timer control register FTMnCON. The configuration of FTMnC is shown below. (n=0-7)

[Description of bits]

- **FTMnC[15:0]** (bits 0 to 15)

  FTMnC is started by writing "1" to the corresponding channel for the timer enable register FTMnEN and stopped by writing "1" to the corresponding channel for the timer disable register FTMnDIS.

  When overflowed, an interrupt request is generated and the value of the timer register FTMnR is loaded at the same time.

  When writing a value to the timer counter (FTMm_FTMnC), the same value is written to the timer counter (FTMm_FTMnC) and the timer register (FTMm_FTMnR).

7.17.3.4 Timer Register 0 (FTMm_FTMnR: n = 0~~ to 7~~)

Address: BASE + 0x040*n + 0x00C

Access: R/W

Access size: 32 bits

Initial value: 0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*
Access	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Symbol name	FTMnR[15:0]															
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

*: Reserved bit for future extension. "0" is read when reading. Write "0" when writing.

[Description of Register]

FTMnR is a 16-bit read/write register that sets a reload value of the timer counter FTMnC. The configuration of FTMnR is shown below. (n=0-7)

[Description of bits]

- **FTMnR[15:0]** (bits 0 to 15)

  When writing a value to the timer counter (FTMm_FTMnC), the same value is written to the timer counter (FTMm_FTMnC) and the timer register (FTMm_FTMnR).

  When writing a value to the timer register (FTMm_FTMnR), the value is written to the timer register (FTMm_FTMnR) while not written to the timer counter (FTMm_FTMnC).

7.17.3.5 General-purpose Timer Register 0 (FTMm_FTMnGR: n = 0 ~~to 7~~)

Address: BASE + 0x040*n + 0x010

Access: R/W

Access size: 32 bits

Initial value: 0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*
Access	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Symbol name	FTMnGR[15:0]															
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

*: Reserved bit for future extension. "0" is read when reading. Write "0" when writing.

[Description of Register]

FTMnGR is a 16-bit read/write register that is used when a value is set in the CMO, PWM, or CAP mode. The configuration of FTMnGR is shown below. (n=0~~-7~~)

[Description of bits]

- **FTMnGR[15:0]** (bits 0 to 15)

  • In CMO/PWM mode:

  Retains the value to compare with the timer counter FTMnC.

  • In CAP mode:

  Retains the value of the timer counter FTMnC when input from the timer input (FTMINn) changes.

  However, if the timing when a value of the timer counter FTMnC is retained conflicts with the write to this register from CPU, the write from CPU has priority.

7.17.3.6 Timer Input/output Level Register 0 (FTMm_FTMnIOLV: n = 0~~ to 7~~)

Address: BASE + 0x040*n + 0x014

Access: R/W

Access size: 32 bits

Initial value: 0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*
Access	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	IOLV[1:0]	
Access	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

*: Reserved bit for future extension. "0" is read when reading. Write "0" when writing.

[Description of Register]

FTMnIOLV is a read/write register that specifies settings about the timer output TIMERn (FTMOUT[n]) in the CMO and PWM modes, or settings about the timer input (FTMIN[n]) in the CAP mode. (n=0~~-7~~)

[Description of bits]

- **IOLV[1:0]** (bits 0 to 1)

  • In CMO mode:

  These bits make settings for TIMERn (FTMOUT[n]).

IOLV[1:0]	Description
00	Outputs "0" when the timer counter FTMnC coincides with the general-purpose timer register FTMnGR
01	Outputs "1" when the timer counter FTMnC coincides with the general-purpose timer register FTMnGR
10	Output is reversed when the timer counter FTMnC coincides with the general-purpose timer register FTMnGR
11	Do not use

  • In PWM mode:

  These bits make settings for TIMERn (FTMOUT[n]).

IOLV[1:0]	Description
00	Outputs "0" when the timer counter FTMnC ≦ the general-purpose timer register FTMnGR Outputs "1" when the timer counter FTMnC > the general-purpose timer register FTMnGR
01	Outputs "1" when the timer counter FTMnC ≦ the general-purpose timer register FTMnGR Outputs "0" when the timer counter FTMnC > the general-purpose timer register FTMnGR
1X	Do not use

• In CAP mode:

Sets a capture trigger of FTMIN[n].

IOLV[1:0]	Description
00	Does not set a capture trigger.
01	Detects a rising edge of the timer input FTMIN[n] as capture trigger.
10	Detects a falling edge of the timer input FTMIN[n] as capture trigger.
11	Detects both edges of the timer input FTMIN[n] as capture trigger.

7.17.3.7 Timer Output Level Register 0 (FTMm_FTMnOUT: n = 0~~to 7~~)

Address: BASE + 0x040*n + 0x018

Access: R/W

Access size: 32 bits

Initial value: 0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*
Access	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	FTMOUT
Access	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

*: Reserved bit for future extension. "0" is read when reading. Write "0" when writing.

In the CMO or PWM mode, if the timing when a timer output value TIMERn (FTMOUT[n]) is determined by the coincidence of the timer counter value (FTMm_FTMnC) and the general-purpose timer register value (FTMm_FTMnGR) conflicts with the write to this register from CPU, the write to this register has priority.

[Description of Register]

FTMnOUT is a read/write register that retains a value output from the timer output TIMERn (FTMOUT[n]). The configuration of FTMnOUT is shown below. (n=0~~-7~~)

[Description of bits]

- **FTMOUT** (bit 0)

  The bit to set TIMERn (FTMOUT[n]).

FTMOUT	Description
0	Sets FTMOUT[n] to "0".
1	Sets FTMOUT[n] to "1".

7.17.3.8 Timer Interrupt Enable Register 0 (FTMm_FTMnIER: n = 0~~ to 7~~)

Address: BASE + 0x040*n + 0x01C

Access: R/W

Access size: 32 bits

Initial value: 0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*
Access	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	OVFIE	CMOIE_CAPIE
Access	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

*: Reserved bit for future extension. "0" is read when reading. Write "0" when writing.

[Description of Register]

FTMnIER is a read/write register that set an interrupt for each channel to enabled. The configuration of FTMnIER is shown below. (n=0~~-7~~)

[Description of bits]

- **CMOIE_CAPIE** (bit 0)

• In CMO mode:

Sets an interrupt when the value of the timer counter FTMnC coincides with the value of the general-purpose timer register FTMnGR to enabled/disabled. An interrupt to CPU is not output at the time of disabled.

CMOIE/CAPIE	Description
0	Sets the interrupt when the value of the timer counter FTMnC coincides with the value of the general-purpose timer register to disabled
1	Sets the interrupt when the value of the timer counter FTMnC coincides with the value of the general-purpose timer register to enabled

• In CAP mode:

Sets an interrupt when input from the timer input (FTMIN[n]) changes (capture trigger) to enabled/disabled. An interrupt to CPU is not output at the time of disabled.

CMOIE/CAPIE	Description
0	Sets the timer input capture interrupt to disabled
1	Sets the timer input capture interrupt to enabled

- **OVFIE** (bit 1)

Sets an interrupt when the timer counter overflows to enabled/disabled.

An interrupt to CPU is not output at the time of disabled.

OVFIE	Description
0	Sets the overflow interrupt to disabled
1	Sets the overflow interrupt to enabled

7.17.3.9 Timer Clock Control Register 0 (FTMm_FTMnCKCON: n=0~~ to 7~~)

Address: BASE + 0x040*n + 0x020

Access: R/W

Access size: 32 bits

Initial value: 0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*
Access	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	CKCON[3:0]			
Access	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

*: Reserved bit for future extension. "0" is read when reading. Write "0" when writing.

[Description of Register]

FTMnCKCON controls FTMCLK[n] supplied from the clock generation section for each channel[5].

Table Relationship between CKCON[3:0] and FTMCLK[n]

CKCON[3:0]	Frequency of FTMCLK[n]	Remarks
0000	64 dividing of system clock	
0001	128 dividing of system clock	
0010	256 dividing of system clock	
0011	512 dividing of system clock	
01xx	Uses a 1 MHz clock that is independent from the system clock as the clock.	
1xxx	Uses the ftmn_io input as the clock. (n is a corresponding channel number from 0 to 7)	

7.17.3.10 Timer Enable Register (FTMm_FTMEN)

Address: BASE + 0x200

Access: R/W

Access size: 32 bits

Initial value: 0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*
Access	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	FTMEN[7:0]							
Access	-	-	-	-	-	-	-	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

*: Reserved bit for future extension. "0" is read when reading. Write "0" when writing.

[Description of Register]

FTMEN is a read/write register that permits each timer operation. The configuration of FTMEN is shown below.

[Description of bits]

- **FTMEN[7:0]** (bits 0 to 7)

  When writing "1", these bits are set to "1", and the timer operation is started. Writing "0" does not change these bits.

  FTMEN[n] : Controls the operation start of TimerN

  * Writing to bits that correspond to the disabled channels is not allowed (Only bit 0 is enabled for this LSI).

7.17.3.11 Timer Disable Register (FTMDIS)

Address: BASE + 0x204

Access: W

Access size: 32 bits

Initial value: 0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*	-*
Access	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Symbol name	-*	-*	-*	-*	-*	-*	-*	-*	FTMDIS[7:0]							
Access	-	-	-	-	-	-	-	-	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

*: Reserved bit for future extension. Write "0" when writing.

[Description of Register]

FTMDIS is a write only register that disables each timer operation. The configuration of FTMDIS is shown below.

[Description of bits]

- **FTMDIS[2:0]** (bits 0 to 2)

  When writing "1", the corresponding bit for the timer enable register FTMEN is reset to "0", and the timer operation is stopped.

  Writing "0" does not change FTMEN.

  FTMDIS[N] : Controls the operation stop of TimerN

  * Writing to bits that correspond to the disabled channels is not allowed (Only bit 0 is enabled for this LSI).

7.17.4 Operation Sequence (TimerN)

Each channel of the flexible timer has the following operation modes:

- Auto-reload timer mode (ART)
- Compare out mode (CMO)
- PWM mode
- Capture mode (CAP)

Each operation mode can be selected by the MOD bits of the timer control register (FTMm_FTMnCON) that correspond to each channel.

Operation for each mode is described below.

7.17.4.1 Auto-reload Timer Mode (ART)

The timer can operate in the auto-reload timer mode by setting the MOD bits of the timer control register (FTMm_FTMnCON) that corresponds to each channel to 00 (ART).

If the timer counter (FTMm_FTMnC) overflows, a value of the timer register (FTMm_FTMnR) is loaded into the timer counter. At the same time, an interrupt request occurs.

For the auto-reload timer mode, the state of timer output pin TIMERn (FTMOUT[n]) does not change.

The operation in the auto-reload timer mode is shown in Figure 0-1 Operation in the Auto-reload Timer Mode.

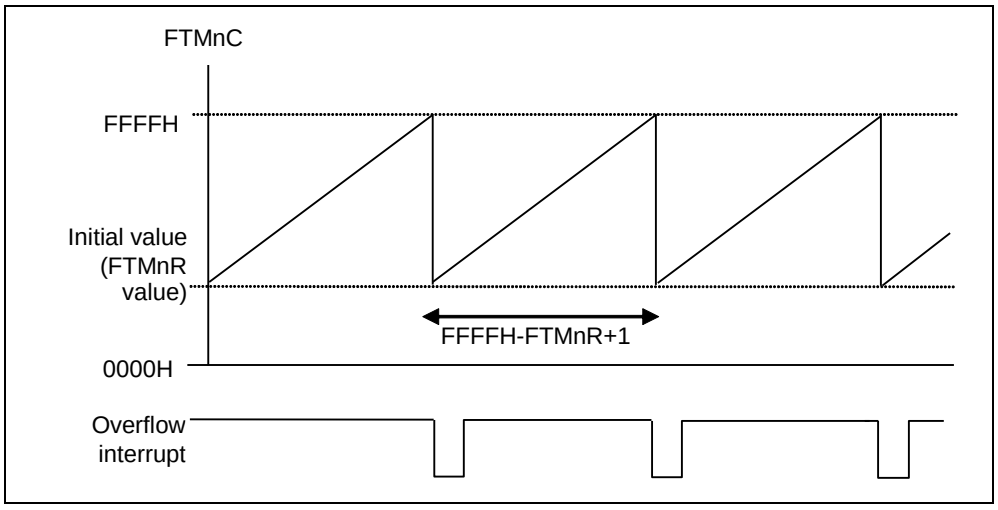


Figure 0-1 Operation in the Auto-reload Timer Mode

7.17.4.2 Compare Out Mode (CMO)

The timer can operate in the compare out mode by setting the MOD bits of the timer control register (FTMm_FTMnCON) that corresponds to each channel to 01 (CMO).

The value output from the timer input/output ftm_io[n] (FTMOUT[n]) by the coincidence of a value of the timer counter (FTMm_FTMnC) and a value of the general-purpose timer register (FTMm_FTMnGR) is determined by the IOLV bit setting of the timer input/output level register (FTMm_FTMnIOLV).

An interrupt request is generated when an overflow occurs and a value of the timer counter coincides with a value of the general-purpose timer register.

The operation in the compare out mode is shown in Figure 0-2 Operation in the Compare Out Mode.

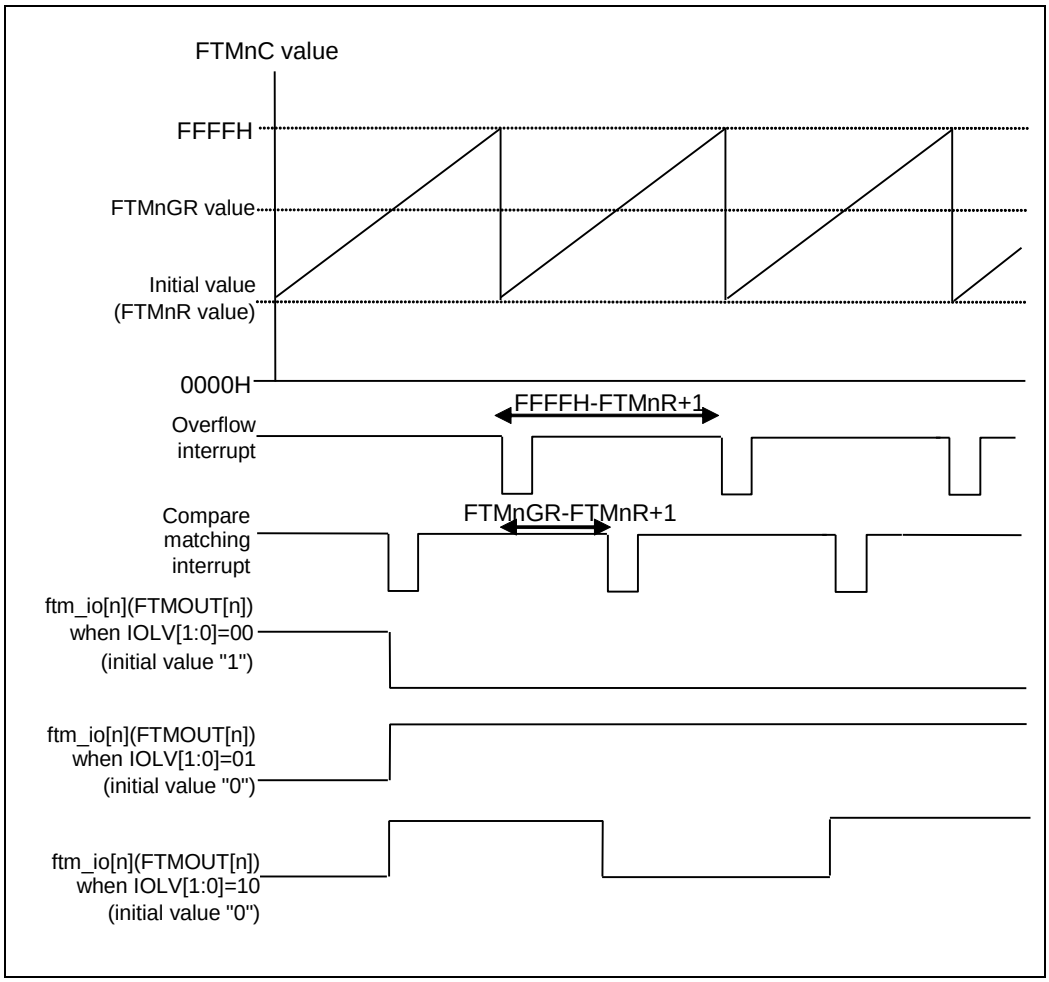


Figure 0-2 Operation in the Compare Out Mode

7.17.4.3 PWM Mode

The timer can operate in the PWM mode by setting the MOD bits of the timer control register (FTMm_FTMnCON) that corresponds to each channel to 10 (PWM).

Sets the cycle in the timer register (FTMm_FTMnR) and sets the length of the first half of phase of PWM in the general-purpose timer register (FTMm_FTMnGR). A value output from the timer input/output ftm_io[n] (FTMOUT[n]) is determined by the IOLV field setting of the timer input/output level register (FTMm_FTMnIOLV).

An interrupt request occurs when the timer counter (FTMm_FTMnC) overflows.

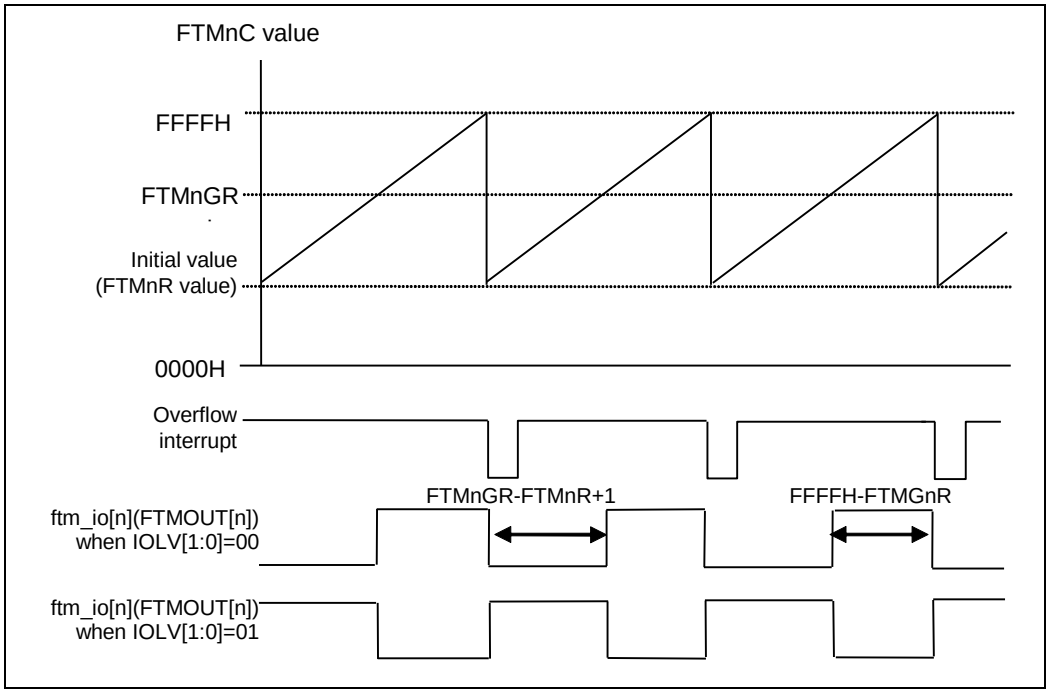


Figure 0-3 Operation in PWM Mode

7.17.4.4 Capture Mode (CAP)

The timer can operate in the capture mode by setting the MOD bits of the timer control register (FTMm_FTMnCON) that corresponds to each channel to 11 (CAP).

When a capture trigger specified in the timer input/output level register (FTMm_FTMnIOLV) is input (generation of the capture event) in the timer input/output ftm_io[n] (FTMIN[n]), a value of the timer counter (FTMm_FTMnC) is stored in the general-purpose timer register (FTMm_FTMnGR).

An interrupt request is generated when an overflow occurs and the capture event occurs.

The operation in the CAP mode is shown in Figure 0-4 Operation in CAP Mode.

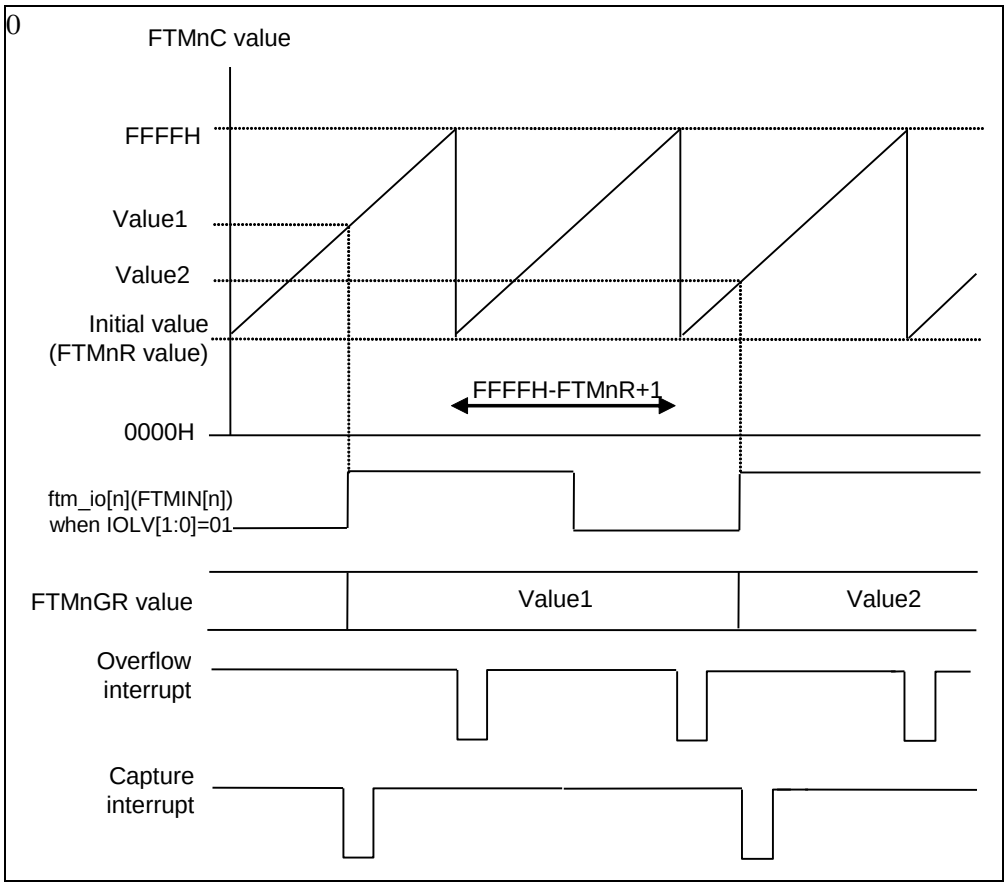


Figure 0-4 Operation in CAP Mode

7.17.4.5 Timer Interrupt

Each channel has two types of interrupts, overflow and capture/compare matching. A signal reversed after executing a logical addition (OR) between the logical product (AND) of the overflow status register (FTMm_FTMnST[1]) and the overflow interrupt enable register (FTMm_FTMnIER[1]) and the logical product (AND) of the capture/compare matching status register (FTMm_FTMnST[0]) and the capture/compare matching interrupt enable register (FTMm_FTMnIER[0]) is output to CPU for each channel. (Total of three Low level outputs Figure 0-5 Interrupt Signal Output to CPU) An interrupt is cleared by writing "1" to the status register (FTMm_FTMnST) from CPU.

However, if the interrupt generation conflicts with clearing the register from CPU, setting by the interrupt generation has priority.

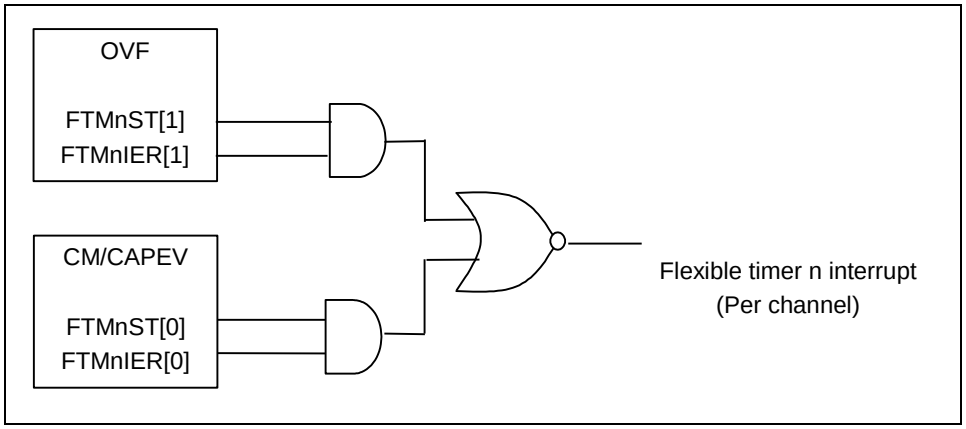


Figure 0-5 Interrupt Signal Output to CPU

7.17.4.6 Switching Input/Output

Since pins of the timer input/output (FTMIN[n]/FTMOUT[n]) are shared in this LSI, input/output is switched by the operation mode. Direction of input/output in each mode is shown below.

ART mode	:	Input (default)
CMO mode	:	Output
PWM mode	:	Output
CAP mode	:	Input

The direction of input/output for each pin can be set for each channel. (The operation mode can be set for each channel)

7.17.4.7 Sampling Timing of Input Signal and Timing of Output Signal

7.17.4.7.1. Sampling of Timer Clock Input

External clock input from the timer clock input pin (FTMCLK[n], n = 0, 1) can be counted.

Figure 0-6 Sampling of External Clock and the Count Timing of Timer Counter (FTMm_FTMnC) shows the sampling of external clock and the count timing of timer counter (FTMm_FTMnC).

FTMCLK[n] is sampled at the rising edge of the bus clock (SYSCLK). When a rising edge of the clock that is input by the timer control register (FTMm_FTMnCON) is selected, if a signal transitions from the L level to the H level at the time of sampling, FTMnC is counted up at the timing of a rising edge of the bus clock (SYSCLK) after 2 clocks.

When a falling edge is selected, if a signal transitions from the H level to the L level at the time of sampling, FTMnC is counted up at the timing of a rising edge of the bus clock (SYSCLK) after 2 clocks.

The duration of H level and L level of the clock input to the FTMCLK[n] pin must be 2 or more cycles of the bus clock (SYSCLK).

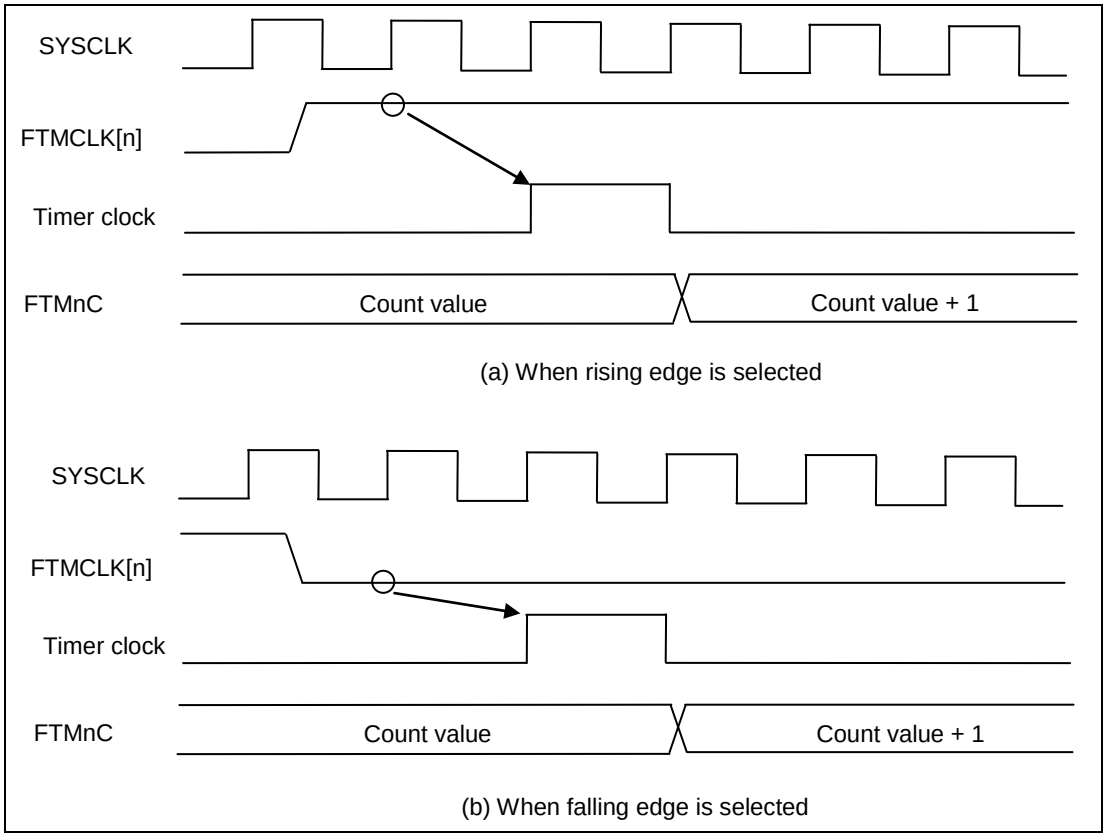


Figure 0-6 Sampling of External Clock and the Count Timing of Timer Counter (FTMm_FTMnC)

7.17.4.7.2. Sampling of Capture Trigger Input

When the flexible timer is set as the capture mode, the timer I/O pin becomes input and is used as the input FTMIN[n] of the capture trigger.

The sampling timing of capture trigger and the timing when the count value of the timer counter (FTMm_FTMnC) is stored in the general-purpose timer register (FTMm_FTMnGR) are shown in Figure 0-7 Timing When Count Value of Capture Trigger Is Stored in the General Purpose Timer Register.

FTMIN[n] is sampled at the rising edge of the bus clock (SYSCLK). When a rising edge is selected as the capture trigger by the timer input/output level register (FTMm_FTMnIOLV), if a signal transitions from the L level to the H level at the time of sampling, the count value of FTMnC is stored in FTMnGR at the timing of a rising edge of the bus clock (SYSCLK) after 2 clocks.

When a falling edge is selected, if a signal transitions from the H level to the L level at the time of sampling, the count value of FTMnC is stored in FTMnGR at the timing of a rising edge of the bus clock (SYSCLK) after 2 clocks.

The duration of H level and L level of the clock input to FTMIN[n] must be 2 or more cycles of the bus clock (SYSCLK).

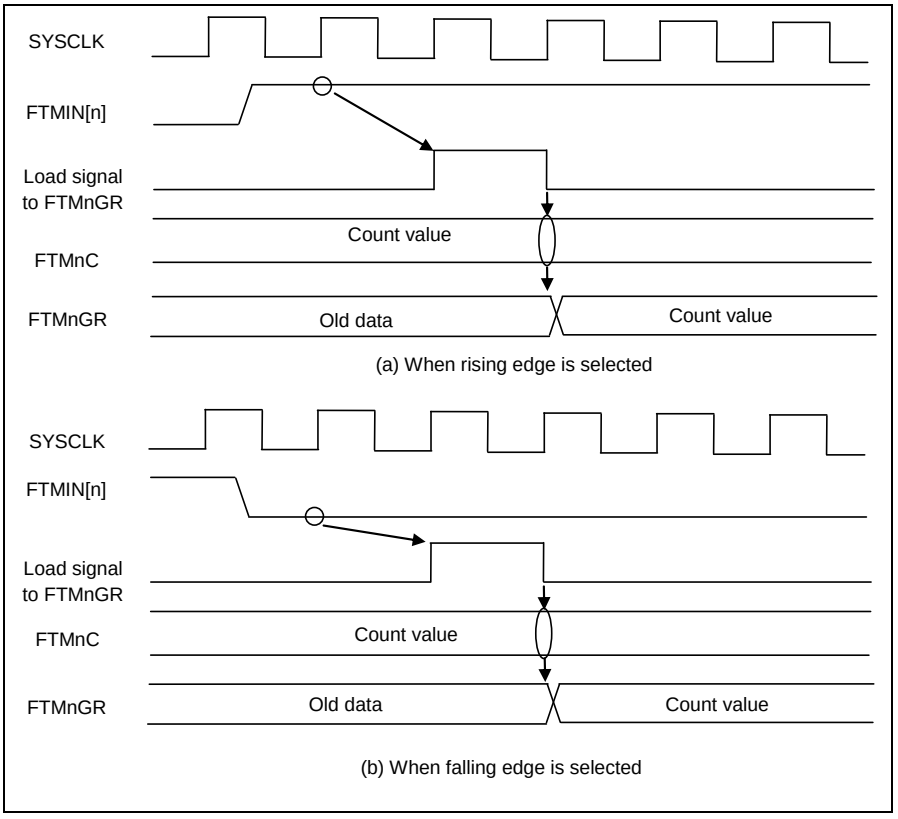


Figure 0-7 Timing When Count Value of Capture Trigger Is Stored in the General Purpose Timer Register

7.17.4.8 Timing of Timer Output

When the flexible timer is set as the CMO mode or PWM mode, the timer input/output is used as the timer output pin TIMERn (FTMOUT[n]).

The timing when the timer output transitions in the CMO mode is shown in Figure 0-8 Timing When the Timer Output Transitions in the CMO Mode. When the flexible timer is set as the CMO mode, the state of timer output transitions at the timing of the count clock after the count value of the timer counter (FTMm_FTMnC) coincides with the value stored in the general-purpose timer register (FTMm_FTMnGR).

The timing when the timer output transitions in the PWM mode is shown in Figure 0-9 Timing When the Timer Output Transitions in the PWM Mode. When the flexible timer is set as the PWM mode, the state of timer output transitions at the timing of the count clock after the count value of FTMnC and the value stored in FTMnGR meet the specified conditions.

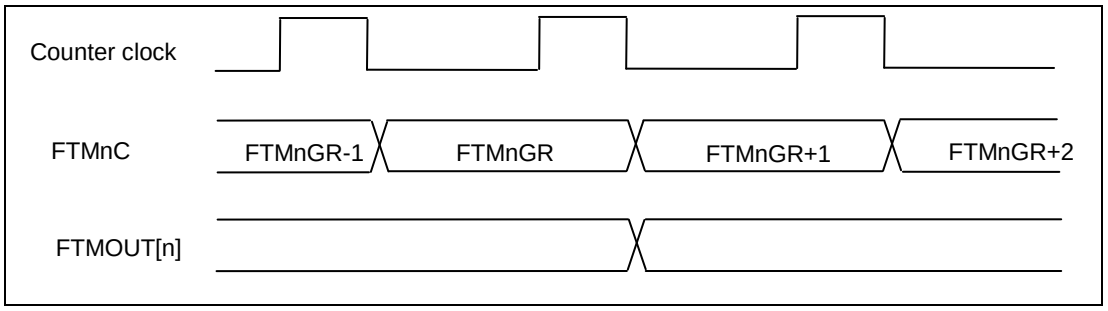


Figure 0-8 Timing When the Timer Output Transitions in the CMO Mode

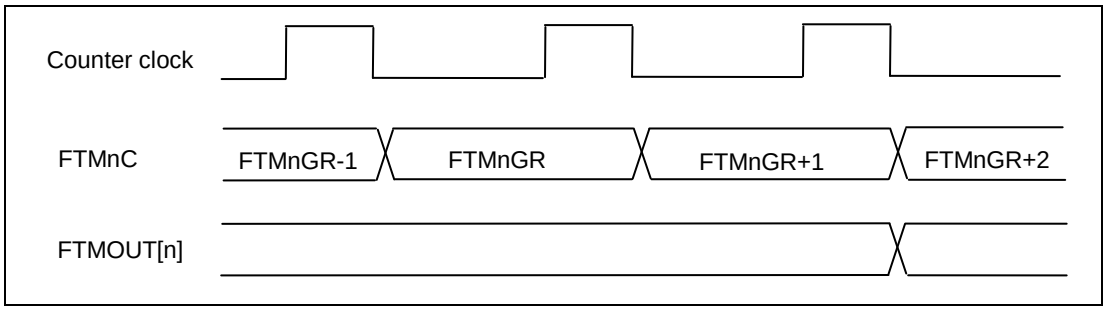


Figure 0-9 Timing When the Timer Output Transitions in the PWM Mode

7-18.I2C

7.18.1 General Description

2-wire (SCL, SDA) serial interface. The features are shown below.

- Supports the standard mode (up to 100 kbps) and the fast mode (up to 400 kbps).
- Supports the 7- or 10-bit addressing.
- Supports the 7- or 10-bit composite format transfer.
- Supports the bulk transfer mode.

7.18.2 List of Registers

Address [H]	Name	Symbol	R/W	Initial value [H]	Description
0x40008000	I2C control register	I2Cn_CON	R/W	0x0000007D	
0x40008004	I2C target address register	I2Cn_TAR	R/W	0x00000055	
0x40008008	I2C slave address register	I2Cn_SAR	R/W	0x00000055	
0x40008010	I2C Rx/Tx data buffer command register	I2Cn_DATA_CMD	R/W	0x00000000	
0x40008014	Standard I2C clock SCL high count register	I2Cn_SS_SCL_HCNT	R/W	0x0000XXXX	
0x40008018	Standard I2C clock SCL low count register	I2Cn_SS_SCL_LCNT	R/W	0x0000XXXX	
0x4000801C	Fast I2C clock SCL high count register	I2Cn_FS_SCL_HCNT	R/W	0x0000XXXX	
0x40008020	Fast I2C clock SCL low count register	I2Cn_FS_SCL_LCNT	R/W	0x0000XXXX	
0x4000802C	I2C interrupt status register	I2Cn_INTR_STAT	R	0x00000000	
0x40008030	I2C interrupt mask register	I2Cn_INTR_MASK	R/W	0x000008FF	
0x40008034	I2C RAW interrupt status register	I2Cn_RAW_INTR_STAT	R	0x00000000	
0x40008038	I2C receive FIFO threshold register	I2Cn_RX_TL	R/W	0x00000000	
0x4000803C	I2C transmit FIFO threshold register	I2Cn_TX_TL	R/W	0x00000000	
0x40008040	I2C interrupt clear register	I2Cn_CLR_INTR	R	0x00000000	
0x40008044	RX_UNDER interrupt clear register	I2Cn_CLR_RX_UNDER	R	0x00000000	
0x40008048	RX_OVER interrupt clear register	I2Cn_CLR_RX_OVER	R	0x00000000	
0x4000804C	TX_OVER interrupt clear register	I2Cn_CLR_TX_OVER	R	0x00000000	
0x40008050	RD_REQ interrupt clear register	I2Cn_CLR_RD_REQ	R	0x00000000	
0x40008054	TX_ABRT interrupt clear register	I2Cn_CLR_TX_ABRT	R	0x00000000	
0x40008058	RX_DONE interrupt clear register	I2Cn_CLR_RX_DONE	R	0x00000000	
0x4000805C	ACTIVITY interrupt clear register	I2Cn_CLR_ACTIVITY	R	0x00000000	
0x40008060	STOP_DET interrupt clear register	I2Cn_CLR_STOP_DET	R	0x00000000	
0x40008064	START_DET interrupt clear register	I2Cn_CLR_START_DET	R	0x00000000	

0x40008068	GEN_CALL interrupt clear register	I2Cn_CLR_GEN_CALL	R	0x00000000	
0x4000806C	I2C enable register	I2Cn_ENABLE	R/W	0x00000000	
0x40008070	I2C status register	I2Cn_STATUS	R	0x00000060	
0x40008074	Transmit FIFO level register	I2Cn_TXFLR	R	0x00000000	
0x40008078	Receive FIFO level register	I2Cn_RXFLR	R	0x00000000	
0x4000807C	SDA hold time register	I2Cn_SDA_HOLD	R/W	0x00000001	
0x40008080	I2C transmit abort status register	I2Cn_TX_ABRT_SOURCE	R	0x00000000	
0x40008084	I2C SLV_DATA_NACK generation register	I2Cn_SLV_DATA_NACK_ONLY	R/W	0x00000000	
0x40008094	I2C SDA setup register	I2Cn_SDA_SETUP	R/W	0x00000064	
0x40008098	I2C ACK GENERAL CALL register	I2Cn_ACK_GENERAL_CALL	R/W	0x00000001	
0x4000809C	I2C enable status register	I2Cn_ENABLE_SATUS	R	0x00000000	
0x400080A0	I2C spike inhibiting limit register	I2Cn_FS_SPKLEN	R/W	0x00000002	
0x400080F4	Configuration parameter register 1 register	I2Cn_COMP_PARAM_1	R	0x000007AA	
0x400080F8	I2C component version register	I2Cn_COMP_VERSION	R	0x3132302A	
0x400080FC	I2C component type register	I2Cn_COMP_TYPE	R	0x44570140	

* n indicates a module number (n = 0).

7.18.3 Description of Registers

7.18.3.1 I2C Control Register: BASE + 0x00 (I2Cn_CON)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																									IC_CON[6:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1
R / W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Description	Remarks
IC_CON[6]	IC_SLAVE_DISABLE	R/W	Disables the slave function. 0: Slave is enabled. 1: Slave is disabled.	
IC_CON[5]	IC_RESTART_EN	R/W	Determines whether or not the RESTART state is transmitted in the master mode. 0: RESTART transmission is disabled. 1: RESTART transmission is enabled.	
IC_CON[4]	IC_10BITADDR_MASTER	R/W	Sets the addressing in the master mode. 0: 7-bit addressing 1: 10-bit addressing	
IC_CON[3]	IC_10BITADDR_SLAVE	R/W	Sets the addressing in the slave mode. 0: 7-bit addressing 1: 10-bit addressing	
IC_CON[2:1]	SPEED	R/W	Sets the transfer speed. 1: Standard mode 2: Fast mode	
IC_CON[0]	MASTER_MODE	R/W	Enables the master function. 0: Master function is disabled. 1: Master function is enabled.	

* Different values cannot be set in IC_CON[6] and IC_CON[0] simultaneously.

  To operate as slave, set IC_CON[6] = 0 && IC_CON[0] = 0.

  To operate as master, set IC_CON[6] = 1 && IC_CON[0] = 1.

* To switch between the master operation and slave operation, reset I2C module through the peripheral reset register.

7.18.3.2 I2C Target Address Register: BASE + 0x04 (I2Cn_TAR)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																				IC_TAR[12:0]												
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Description	Remarks
IC_TAR[11]	SPECIAL	R/W	Sets the use of General Call or START BYTE. 0: Uses IC_TART[9:0], ignoring GC_OR_START of IC_STAR[10]. 1: Uses the I2C command set in GC_OR_START.	
IC_TAR[10]	GC_OR_START	R/W	Selects the General Call or START BYTE command when the SPECIAL bit of IC_TAR[11] is set. 0: General call address 1: START byte	
IC_TAR[9:0]	IC_TAR	R/W	Sets the target address when the master operates. This field is ignored during the transmission of General Call. To generate a START BYTE, a value must be written to this field only once.	

7.18.3.3 I2C Slave Address Register: BASE + 0x08 (I2Cn_SAR)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																						IC_SAR[9:0]									
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Description	Remarks
IC_SAR[9:0]	IC_SAR	R/W	Sets the slave address when the slave operates. Only IC_SAR[6:0] is used at the time of 7-bit addressing. Write to this field is enabled when the I2C interface is disabled, which means only when IC_ENABLE[0] is 0.	

7.18.3.4 I2C Rx/Tx Buffer Data Command: BASE + 0x10 (I2Cn_DATA_CMD)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																							IC_DATA_CMD[10:0]								
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Description	Remarks
IC_DATA_CMD[8]	CMD	W	Selects a read or write operation when the master operates. 0: Write 1: Read	
IC_DATA_CMD[7:0]	DAT	R/W	Stores the transmit or received data.	

7.18.3.5 Standard I2C Clock SCL High Count Register: BASE + 0x14 (I2Cn_SS_SCL_HCNT)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																IC_SS_SCL_HCNT[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Sets the number of cycles in a H interval of the SCL clock in the standard mode.

This register should be set when the I2C interface is disabled (IC_ENABLE[0] is 0).

A value equal to or greater than 6 can be set. When writing a value less than 6, 6 is written.

7.18.3.6 Standard I2C Clock SCL Low Count Register: BASE + 0x18 (I2Cn_SS_SCL_LCNT)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																IC_SS_SCL_LCNT[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Sets the number of cycles in a L interval of the SCL clock in the standard mode.

This register should be set when the I2C interface is disabled (IC_ENABLE[0] is 0).

A value equal to or greater than 8 can be set. When writing a value less than 8, 8 is written.

7.18.3.7 Fast I2C Clock SCL High Count Register: BASE + 0x1C (I2Cn_FS_SCL_HCNT)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																IC_FS_SCL_HCNT[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Sets the number of cycles in a H interval of the SCL clock in the fast mode.

This register should be set when the I2C interface is disabled (IC_ENABLE[0] is 0).

A value equal to or greater than 6 can be set. When writing a value less than 6, 6 is written.

7.18.3.8 Fast I2C Clock SCL Low Count Register: BASE + 0x20 (I2Cn_FS_SCL_LCNT)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																IC_FS_SCL_LCNT[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Sets the number of cycles in a L interval of the SCL clock in the fast mode.

This register should be set when the I2C interface is disabled (IC_ENABLE[0] is 0).

A value equal to or greater than 8 can be set. When writing a value less than 8, 8 is written.

○Pulse width

SCL H pulse width: SCL_H_time

SCL L pulse width: SCL_L_time

These can be represented by the following expressions.

* SCL_Fall_time and SCL_Rise_time depend on the system configuration.

Standard mode (100 kHz)

SCL_H_time = (I2C_SS_SCL_HCNT[15:0] + I2C_SS_SPKLEN[7:0] + 6) × I2C reference clock cycle + SCL_Fall_time

SCL_L_time = (I2C_SS_SCL_LCNT[15:0] + 1) × I2C reference clock cycle + SCL_Rise_time

Fast mode (400 kHz)

SCL_H_time = (I2C_FS_SCL_HCNT[15:0] + I2C_FS_SPKLEN[7:0] + 6) × I2C reference clock cycle + SCL_Fall_time

SCL_L_time = (I2C_FS_SCL_LCNT[15:0] + 1) × I2C reference clock cycle + SCL_Rise_time

When I2C clock is set, each pulse width should be adjusted to meet the following conditions by reference to the above expressions.

Standard mode (100 kHz): SCL_H_time + SCL_L_time = 10 μs

Fast mode (400 kHz): SCL_H_time + SCL_L_time = 2.5 μs

7.18.3.9 I2C Interrupt Status Register: BASE + 0x2C (I2Cn_INTR_STAT)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																				IC_INTR_STAT											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description	Remarks
IC_INTR_STAT[11]	R_GEN_CALL	R	Indicates the interrupt status.	
IC_INTR_STAT[10]	R_START_DET	R	For details of each bit, refer to the interrupt RAW status register. The masked	
IC_INTR_STAT[9]	R_STOP_DET	R	interrupt status is read from this register.	
IC_INTR_STAT[8]	R_ACTIVITY	R	To clear the interrupt source, read the interrupt clear register corresponding to	
IC_INTR_STAT[7]	R_RX_DONE	R	the appropriate interrupt source.	
IC_INTR_STAT[6]	R_TX_ABRT	R		
IC_INTR_STAT[5]	R_RD_REQ	R		
IC_INTR_STAT[4]	R_TX_EMPTY	R		
IC_INTR_STAT[3]	R_TX_OVER	R		
IC_INTR_STAT[2]	R_RX_FULL	R		
IC_INTR_STAT[1]	R_RX_OVER	R		
IC_INTR_STAT[0]	R_RX_UNDER	R		

7.18.3.10 I2C Interrupt Mask Register: BASE + 0x30 (I2Cn_INTR_MASK)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																				IC_INTR_MASK											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Description	Remarks
IC_INTR_MASK[11]	M_GEN_CALL	R/W	Masks the corresponding interrupt source. 0: Mask 1: No mask	
IC_INTR_MASK[10]	M_START_DET	R/W		
IC_INTR_MASK[9]	M_STOP_DET	R/W		
IC_INTR_MASK[8]	M_ACTIVITY	R/W		
IC_INTR_MASK[7]	M_RX_DONE	R/W		
IC_INTR_MASK[6]	M_TX_ABRT	R/W		
IC_INTR_MASK[5]	M_RD_REQ	R/W		
IC_INTR_MASK[4]	M_TX_EMPTY	R/W		
IC_INTR_MASK[3]	M_TX_OVER	R/W		
IC_INTR_MASK[2]	M_RX_FULL	R/W		
IC_INTR_MASK[1]	M_RX_OVER	R/W		
IC_INTR_MASK[0]	M_RX_UNDER	R/W		

7.18.3.11 I2C Interrupt RAW Status Register: BASE + 0x34 (I2Cn_RAW_INTR_STAT)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																				IC_RAW_INTR_STAT											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Description	Remarks
IC_RAW_INTR_STAT[11]	GEN_CALL	R	Set when receiving the General Call address or the response to it.	
IC_RAW_INTR_STAT[10]	START_DET	R	Set when the START or RESTART state occurs.	
IC_RAW_INTR_STAT[9]	STOP_DET	R	Set when the STOP state occurs.	
IC_RAW_INTR_STAT[8]	ACTIVITY	R	Set when this I2C circuit becomes active. There are following four conditions to clear: •Disable this I2C •Read IC_CLR_ACTIVITY register •Read IC_CLR_INTR register •Reset	
IC_RAW_INTR_STAT[7]	RX_DONE	R	When operating as a slave transmitter, this bit is set to 1 if there is no response from the master.	
IC_RAW_INTR_STAT[6]	TX_ABRT	R	When operating as a transmitter, this bit is set to 1 if data transmission within the transmit FIFO is not completed. Detailed abort cause is stored in the I2C transmit abort status register.	
IC_RAW_INTR_STAT[5]	RD_REQ	R	When operating as a slave, this bit is set to 1 if the data read occurs from the master. The I2C bus continues the waiting state until an interrupt processing is started. When this interrupt source is asserted, the host CPU must write data to IC_DATA_CMD.	
IC_RAW_INTR_STAT[4]	TX_EMPTY	R	This bit is set to 1 when the transmit buffer is less than or equal to the value set in the I2C transmit FIFO threshold register. When the transmit buffer exceeds the setting value of the transmit FIFO threshold register, it is automatically cleared.	

IC_RAW_INTR_STAT[3]	TX_OVER	R	This bit is set to 1 when writing data to IC_DATA_CMD while the transmit buffer is full.	
IC_RAW_INTR_STAT[2]	RX_FULL	R	This bit is set to 1 when the number of data within the receive buffer is more than or equal to the value set in the I2C receive FIFO threshold register. When it is less than the setting value of the receive FIFO threshold register, it is automatically cleared.	
IC_RAW_INTR_STAT[1]	RX_OVER	R	This bit is set to 1 when receiving data from outside while the receive buffer is full. Responds to the transmitting side on the I2C bus though the received data will be discarded.	
IC_RAW_INTR_STAT[0]	RX_UNDER	R	This bit is set to 1 when reading the IC_DATA_CMD register while the receive buffer is empty.	

7.18.3.12 I2C Receive FIFO Threshold Register: BASE + 0x38 (I2Cn_RX_TL)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								RX_TL[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Sets the threshold level of the receive FIFO.

7.18.3.13 I2C Transmit FIFO Threshold Register: BASE + 0x3C (I2Cn_TX_TL)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								TX_TL[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Sets the threshold level of the transmit FIFO.

7.18.3.14 I2C Interrupt Clear Register: BASE + 0x40 (I2Cn_CLR_INTR)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															CLR_INTR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When reading this register, the interrupt source and the transmit abort status register are cleared. The source to be cleared by hardware is not cleared.

7.18.3.15 RX_UNDER Interrupt Clear Register: BASE + 0x44 (I2Cn_CLR_RX_UNDER)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															CLR_RX_UNDER
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When this register is read, the RX_UNDER interrupt is cleared.

7.18.3.16 RX_OVER Interrupt Clear Register: BASE + 0x48 (I2Cn_CLR_RX_OVER)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															CLR_RX_OVER
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When this register is read, the RX_OVER interrupt is cleared.

7.18.3.17 TX_OVER Interrupt Clear Register: BASE + 0x4C(I2Cn_CLR_TX_OVER)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															CLR_TX_OVER
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When this register is read, the TX_OVER interrupt is cleared.

7.18.3.18 RD_REQ Interrupt Clear Register: BASE + 0x50(I2Cn_CLR_RD_REQ)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															CLR_RD_REQ
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When this register is read, the RD_REQ interrupt is cleared.

7.18.3.19 TX_ABRT Interrupt Clear Register: BASE + 0x54 (I2Cn_CLR_TX_ABRT)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															CLR_TX_ABRT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When reading this register, the TX_ABRT interrupt and the transmit abort status register are cleared.

7.18.3.20 RX_DONE Interrupt Clear Register: BASE + 0x58 (I2Cn_CLR_RX_DONE)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															CLR_RX_DONE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When this register is read, the RX_DONE interrupt is cleared.

7.18.3.21 ACTIVITY Interrupt Clear Register: BASE + 0x5C (I2Cn_CLR_ACTIVITY)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															CLR_ACTIVITY
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When reading this register, the ACTIVITY interrupt is cleared if the I2C module is not active. Maintains the state where 1 is set if it is Active.

7.18.3.22 STOP_DET Interrupt Clear Register: BASE + 0x60 (I2Cn_CLR_STOP_DET)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															CLR_STOP_DET
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When this register is read, the STOP_DET interrupt is cleared.

7.18.3.23 START_DET Interrupt Clear Register: BASE + 0x64 (I2Cn_CLR_START_DET)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															CLR_START_DET
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When this register is read, the START_DET interrupt is cleared.

7.18.3.24 GEN_CALL Interrupt Clear Register: BASE + 0x68 (I2Cn_CLR_GEN_CALL)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															CLR_GEN_CALL
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When this register is read, the GEN_CALL interrupt is cleared.

7.18.3.25 IC_ENABLE Interrupt Clear Register: BASE + 0x6C (I2Cn_ENABLE)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														ABORT	ENABLE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Bit	Name	R/W	Description	Remarks
IC_ENABLE[1]	ABORT	R/W	This bit is set to 1 when starting ABORT. 0: Indicates that ABORT is not started or ABORT is completed. 1: Indicates that the ABORT operation is running.	
IC_ENABLE[0]	ENABLE	R/W	Enable the I2C module. 0: The I2C module is disabled. 1: The I2C module is enabled.	

* ABORT can be used only when the transmit FIFO (I2Cn_STATUS[2] = 1) is used and the Master state machine is Active (not IDLE, I2Cn_STATUS[5] = 1).

7.18.3.26 I2C Status Register: BASE + 0x70 (I2Cn_STATUS)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																									IC_STATUS						
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	Description	Remarks
IC_STATUS[6]	SLV_ACTIVITY	Indicates that the Slave state machine is Active (not IDLE). 0: IDLE 1: not IDLE	
IC_STATUS[5]	MST_ACTIVITY	Indicates that the Master state machine is Active (not IDLE). 0: IDLE 1: not IDLE	
IC_STATUS[4]	RFF	0: Indicates that the receive FIFO is not full. 1: Indicates that the receive FIFO is full.	
IC_STATUS[3]	RFNE	0: Indicates that the receive FIFO is empty. 1: Indicates that the receive FIFO is not empty.	
IC_STATUS[2]	TFE	0: Indicates that the transmit FIFO is not empty. 1: Indicates that the transmit FIFO is empty.	
IC_STATUS[1]	TFNF	0: Indicates that the transmit FIFO is full. 1: Indicates that the transmit FIFO is not full.	
IC_STATUS[0]	ACTIVITY	Indicates the I2C Activity status.	

7.18.3.27 I2C Transmit FIFO Level Register: BASE + 0x74 (I2Cn_TXFLR)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												TXFLR			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Indicates the number of valid data included in the transmit FIFO.

7.18.3.28 I2C Receive FIFO Level Register: BASE + 0x78 (I2Cn_RXFLR)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												RXFLR			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Indicates the number of valid data included in the receive FIFO.

7.18.3.29 I2C SDA Hold Time Register: BASE + 0x7C (I2Cn_SDA_HOLD)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																IC_SDA_HOLD[15:0]																
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Sets the SDA hold time. A value from 1 to 4 can be set. (Cycle unit of the source clock supplied to the I2C module)

7.18.3.30 I2C Transmit Abort Status Register: BASE + 0x80 (I2Cn_TX_ABRT_SOURCE)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TX_ABRT_SOURCE																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	Description	Remarks
TX_ABORT_SOURCE[31:24]	TX_FLUSH_CNT	Holds the transmit FIFO level when TX_ABRT occurs.	Master transmission
TX_ABORT_SOURCE[23:17]	Reserved		
TX_ABORT_SOURCE[16]	ABRT_USER_ABRT	This bit is set to 1 when the master detects the transmit abort.	Master transmission
TX_ABORT_SOURCE[15]	ABRT_SLVRD_INTX	This bit is set to 1 when setting (Read setting) the CMD bit of the IC_DATA_CMD register responding to the data request from the master during the slave mode.	Slave transmission
TX_ABORT_SOURCE[14]	ABRT_SLV_ARBLOST	This bit is set to 1 when the slave loses the bus privilege during the data transmission to master.	Slave transmission
TX_ABORT_SOURCE[13]	ABRT_SLVFLUSH_TXFIFO	If any data remains in the transmit FIFO when receiving the read command from the master in the slave mode, generate the TX_ABRT interrupt by setting this bit in order to delete the old data in the transmit FIFO.	Slave transmission
TX_ABORT_SOURCE[12]	ARB_LOST	This bit is set to 1 when the master loses the arbitration or TX_ABORT_SOURCE[14] is set.	Master transmission Slave transmission
TX_ABORT_SOURCE[11]	ABRT_MSTER_DIS	This bit is set to 1 when attempting to start the master operation while the master mode is disabled.	Master transmission Master reception

TX_ABORT_SOURCE[10]	ABRT_10B_RD_NORSTRT	This bit is set to 1 when the master issues a read command by the 10-bit addressing with RESTART set to disabled.	Master reception
TX_ABORT_SOURCE[9]	ABRT_SBYTE_NORSTRT	This bit is set to 1 when attempting to transmit the START byte while RESTART is set to disabled.	Master
TX_ABORT_SOURCE[8]	Reserved		HS Only
TX_ABORT_SOURCE[7]	ABRT_SBYTE_ACKDET	This bit is set to 1 when receiving Ack after transmitting the START byte.	Master
TX_ABORT_SOURCE[6]	Reserved		HS Only
TX_ABORT_SOURCE[5]	ABRT_GCALL_READ	This bit is set to 1 when attempting to transmit a read command following a General Call in the master mode.	Master transmission
TX_ABORT_SOURCE[4]	ABRT_GCALL_NOACK	This bit is set to 1 when there is no response from any slave for the General Call transmission in the master mode.	Master transmission
TX_ABORT_SOURCE[3]	ABRT_TXDATA_NOACK	This bit is set to 1 when there is no response from any slave for the data transmission in the master mode.	Master transmission
TX_ABORT_SOURCE[2]	ABRT_10ADDR2_NOACK	This bit is set to 1 when there is no response from any slave for the second address byte transmission based on the 10-bit addressing in the master mode.	Master transmission Master reception
TX_ABORT_SOURCE[1]	ABRT_10ADDR1_NOACK	This bit is set to 1 when there is no response from any slave for the first address byte transmission based on the 10-bit addressing in the master mode.	Master transmission Master reception
TX_ABORT_SOURCE[0]	ABRT_7B_ADDR_NOACK	This bit is set to 1 when there is no response from any slave for the address transmission based on the 7-bit addressing in the master mode.	Master transmission Master reception

7.18.3.31 I2C SLV_DATA_NACK Generation Register: BASE + 0x84 (I2Cn_SLV_DATA_NACK_ONLY)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															NACK
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Generates NACK. Enabled only when operating as a slave receiver.

When setting this register to 1, NACK is generated after receiving the data byte. Data transfer is aborted and no received data is stored in the receive buffer.

When this register is 0, NACK/ACK is generated according to the normal reference.

7.18.3.32 I2C SDA Setup Time Register: BASE + 0x94 (I2Cn_SDA_SETUP)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																								SDA_SETUP[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

(SDA_SETUP[7:0]-1) * (ic_clk period) is used as setup time.

Minimum settable value is 2.

7.18.3.33 I2C ACK GENERAL CALL Register: BASE + 0x98 (I2Cn_ACK_GENERAL_CALL)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															ACK_GENN_CALL
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Sets the response (NACK or ACK) when receiving GENERAL CALL.

0: Response is not made and no General Call interrupt is generated when receiving GENERAL CALL.

1: Responds ACK when receiving GENERAL CALL.

7.18.3.34 I2C Enable Status Register: BASE + 0x9C (I2C_ENABLE_STATUS)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													IC_ENABLE_STATUS		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	Description	Remarks
IC_ENABLE_STATUS[2]	SLV_RX_DATA_LOST	Indicates that at least 1 data byte receive has been aborted because IC_ENABLE[0] switches from 1 to 0 while operating as a slave receiver.	
IC_ENABLE_STATUS[1]	SLV_DISABLED_WHILE_BUSY	Indicates that the slave operation has been aborted because IC_ENABLE[0] switches from 1 to 0 while operating as a slave transmitter or receiver.	
IC_ENABLE_STATUS[0]	IC_EN	1: This module is enabled. 0: This module is disabled.	

7.18.3.35 I2C Spike Inhibiting Limit Register: BASE + 0xA0 (I2Cn_FS_SPKLEN)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								IC_FS_SPKLEN							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Sets the maximum length of spike occurring in SCL and SDA. (unit: ic_clk)

The spike less than or equal to the value set in this register is removed by the spike inhibiting circuit.

Minimum settable value is 1.

7.18.3.36 Component Register 1: BASE + 0xf4 (I2Cn_COMP_PARAM_1)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								IC_CMP_PARAM_1																								
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	1	0	1	0	1	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	Description	Remarks
IC_CMP_PARAM_1[23:16]	TX_BUFFER_DEPTH	Associated with the transmit buffer and receive buffer sizes. 0x00=Reserved 0x01=2 0x02=3 : 0xFF=256	
IC_CMP_PARAM_1[15:8]	RX_BUFFER_DEPTH		
IC_CMP_PARAM_1[7]	ADD_ENCODE_PARAMS	Indicates the same parameter value as the name.	
IC_CMP_PARAM_1[6]	HAS_DMA		
IC_CMP_PARAM_1[5]	INTR_IO		
IC_CMP_PARAM_1[4]	HC_COUNT_VALIES		
IC_CMP_PARAM_1[3:2]	MAX_SPEED_MODE		
IC_CMP_PARAM_1[1:0]	APB_DATA_WIDTH		

7.18.3.37 I2C Component Register Version Register: BASE + 0xf8 (I2Cn_COMP_VERSION)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IC_CMP_VERSION																															
Initial value	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	1	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Component version ID.

7.18.3.38 I2C Component Type Register: BASE + 0xfc (I2Cn_COMP_TYPE)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IC_CMP_TYPE																															
Initial value	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Hardware component type number.

7-19. Flash ROM Controller

7.19.1 General Description

Memory controller that controls the internal Flash ROM.

It operates as AHB slave when reading the Flash ROM.

Erase/write/register access of the Flash ROM operates as an APB slave.

* In 1 bank mode, the register access of the controller on the SUB side is performed as follows:

•Writing is ignored.

•0 is read when reading.

7.19.2 List of Registers

Address [H]	Name	Symbol[6]	R/W	Initial value [H]	Description
BASE+0x00	Flash-ROM status	FLCnSTA	R	0000_0000	
BASE+0x04	Flash-ROM acceptor	FLCnACP	W	0000_0000	
BASE+0x08	Flash-ROM address	FLCnADR	R/W	0000_0000	
BASE+0x0C	Flash-ROM write data	FLCnWDA	W	0000_0000	
BASE+0x10	Flash-ROM erase	FLCnERA	R/W	0000_0000	
BASE+0x14	Flash-ROM control	FLCnCTR	R/W	0000_0001	
BASE+0x1C	Flash-ROM protection status	FLCn_PSTA	R	FFFF_FFFF	
BASE+0x20	Flash-ROM size	FLCnRSIZ	R	0008_0000	
BASE+0x24	Boot program address	FLCnBADR	R	0003_E800	
BASE+0x28	Interrupt mask	FLCnINTMSK	R/W	0000_001F	
BASE+0x2C	Interrupt status	FLCnINTSTA	R/W	0000_0000	
BASE+0x30	RAW interrupt status	FLCnRINTSTA	R/W	0000_0000	
BASE+0xC4	Flash-ROM error status	FLCn_ERRSTA	R/W	0000_0000	
BASE+0xD0	Protection lock key unlock	FLCn_PROTUNLOCK	W	0000_0000	
BASE+0xD4	Protection lock key status	FLCn_PROTLOCKSTA	RW	0000_0002	
BASE+0xD8	Flash-ROM protection set write data	FLCn_PROTWDA	W	0000_0000	
BASE+0xDC	Flash-ROM protection set erase	FLCn_PROTERA	R/W	0000_0000	

[6] n indicates the identification number when multiple Flash-ROM controllers are mounted. n starts at 0. When only one Flash-ROM controller is mounted, n = 0.

BASE+0x100	Flash-ROM configuration information	FLCn_CONFIG	R	0000_0002	
BASE+0x110	Flash-ROM code user application area #0 address	FLCn_CU0ADR	R	1000_0000	
BASE+0x114	Flash-ROM code protection set area #0 address	FLCn_CP0ADR	R	0000_0000	
BASE+0x118	Flash-ROM code boot program area #0 address	FLCn_CB0ADR	R	1003_E800	
BASE+0x11C	Flash-ROM code area #0 size	FLCn_C0SIZ	R	0004_0000	
BASE+0x120	Flash-ROM data user application area #0 address	FLCn_DU0ADR	R	1800_0000	
BASE+0x124	Flash-ROM data protection set area #0 address	FLCn_DP0ADR	R	0000_0000	
BASE+0x128	Flash-ROM data boot program area #0 address	FLCn_DB0ADR	R	1803_E800	
BASE+0x12C	Flash-ROM data area #0 size	FLCn_D0SIZ	R	0004_0000	
BASE+0x140	Flash-ROM code user application area #1 address	FLCn_CU1ADR	R	1400_0000/1000_0000	
BASE+0x144	Flash-ROM code protection set area #1 address	FLCn_CP1ADR	R	1403_E800/1003_E800	
BASE+0x148	Flash-ROM code boot program area #1 address	FLCn_CB1ADR	R	1403_EA00/1003_EA00	
BASE+0x14C	Flash-ROM code area #1 size	FLCn_C1SIZ	R	0004_0000	
BASE+0x150	Flash-ROM data user application area #1 address	FLCn_DU1ADR	R	1C00_0000	
BASE+0x154	Flash-ROM data protection set area #1 address	FLCn_DP1ADR	R	1C03_E800	
BASE+0x158	Flash-ROM data boot program area #1 address	FLCn_DB1ADR	R	1C03_EA00	
BASE+0x15C	Flash-ROM data area #1 size	FLCn_D1SIZ	R	0004_0000	

* n indicates a module number (n = 0, 1).

* Base address of FLC0 (n = 0) is 0x40000400. The control target Flash ROM area is 0x1800_0000-0x1803_FFFF.

* Base address of FLC1 (n = 1) is 0x40000600. The control target Flash ROM area is 0x1C00_0000-0x1C03_FFFF.

7.19.3 Description of Registers

7.19.3.1 Flash-ROM Status Register: BASE + 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															Reserved													BUSY11	BUSY10	BUSY01	BUSY00
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

BUSYxx	Reads the state of the Flash ROM. This bit is "1" during chip erase/block erase/sector erase/1-word write of Flash xx. It automatically changes to "0" when chip erase/block erase/sector erase/1-word write is completed. This bit is also "1" while performing sector erase in the protection set area when the protection lock key is unlocked. (When Flash xx has the protection set area) In 2 bank mode, this bit is enabled only in the MAIN side controller. 1: Indicates that the state is during chip erase, block erase, sector erase, or 1-word program. 0: Indicates that the state is not during (has completed) chip erase, block erase, sector erase, or 1-word program.

7.19.3.2 Flash-ROM Acceptor Register: BASE + 0x04

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								FAC[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

FAC[7:0]	FAC07 to FAC00 registers are used to restrict chip erase, sector erase, block erase, and 1-word write operations in order to prevent an unintended operation. When writing "0000_00FAH" and "0000_00F5H" to FLCn_ACP in this order, one of the erase operations or 1-word write operation is enabled only once. To perform sector/block erases or 1-word writes continuously, you must write "0000_00FAH" and "0000_00F5H" to FLCn_ACP each time. Even if another instruction is inserted between "0000_00FAH" and "0000_00F5H" written to FLCn_ACP, the erase or 1-word write operation is enabled. However, if you write data other than "0000_00F5H" in FLCn_ACP after writing "0000_00FAH", it is disabled. Therefore, you must write from "0000_00FAH" again to enable it. In addition, if you write to FLCn_ACP without executing erase or 1-word write after writing "0000_00FAH" and "0000_00F5H", it is disabled regardless of the value. Therefore, you must write "0000_00FAH" and "0000_00F5H" in this order again to enable it.

state	Description
Disabled	Write/erase execution disabled
Request	Write/erase execution disabled
Enabled	Write/erase execution enabled

state	Read value of this register	Transition conditions
Disabled	0H	•Write data other than FAH to FLCn_ACP in the prohibited state •Write data other than F5H to FLCn_ACP in the request state •Write any data to FLCn_ACP in the permitted state •Write/erase is completed in the permitted state •Write to FLCn_PROTUNLOCK •Protection lock key unlock operation (PROTUNLOCKWE = 1)

Request	1H	When all the following three conditions are satisfied: •Write/erase is not being executed (BUSY00 to BUSY11 = 0) •Protection lock key unlock is not being executed (PROTUNLOCKWE = 1) •Write FAH to FLCn_ACP in the prohibited state
Enabled	2H	Write F5H to FLCn_ACP in the request state

[Note]

If you write "00" in FLE field with the FLCn_ACP register enabled, it is still maintained as enabled.

During the protection lock key unlock operation, the write/erase operation for Flash-ROM is prohibited.

If all erase of the user application area is caused by the protection lock key unlock operation at the same time as a normal write/erase operation for Flash-ROM, the former has a priority.

7.19.3.3 Flash-ROM Address Register: BASE + 0x08

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FA[31:2]																														Reserved	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	R

FA[31:2]	The FA31 to FA02 bits are used to set the address for partial erase or 1-word write. When writing a byte address of the flash memory area to this register, the address of Flash-ROM is set. An absolute address of Flash-ROM to be written should be written to this register. The upper bits exceeding the address width of Flash-ROM are used to determine whether the specified area is an assigned Flash-ROM area. If it is not a Flash-ROM area, write/erase is ignored.

7.19.3.4 Flash-ROM Write Data Register: BASE + 0x0C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FD[31:0]																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

FD[31:0]	The FD31 to FD00 bits are used to set the write data when performing the 1-word write to the user application area and the boot program area. Writing to the FD31 to FD00 bits starts the 1-word write.

[Note]

Clear the contents of the target addresses in advance. The content of an overwritten address is not guaranteed.

Write to the protection set area cannot be performed by using this register.

7.19.3.5 Flash-ROM Erase Register: BASE + 0x10

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																														FLE [1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

FLE[1:0]	The FLE bit is used to specify the erase type and erase start for the user application area and the boot program area. Writing to FLE starts erase according to the data. It automatically changes to "00" when the erase is completed. Writing "00" is prohibited. This bit remains as "00" when FLCn_ACP is not in the permitted state except during erase/write operation, or when a write to this field is performed. It remains the same as before write if a write to this field is performed during erase/write operation. Write: 00: Disabled 01: Start chip erase 10: Start block erase 11: Start sector erase Read: 00: Erase completed 01: During chip erase 10: During block erase 11: During sector erase

[Note]

Erase of the protection set area cannot be performed by using this register.

7.19.3.6 Flash-ROM Control: BASE + 0x14

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															FLWA
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Sets the maximum AHB latency (presence or absence of prefetch function) when reading a Flash ROM.

FLWA	**Description**
0	0 wait/prefetch function disable
1	Maximum 1 wait/prefetch function enable

* Use the initial value (FLWA = fixed to 1) for this product.

[Note] When writing to FLWA, the program should be started from other than the Flash ROM space[7] .

[7] If the wait/prefetch setting switches during access to the Flash ROM, the operation cannot be guaranteed.

7.19.3.7 Flash-ROM Protection Status: BASE + 0x1C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	ERAALL	LOCKUSR	Reserved												DISEPISP	DISRISP	Reserved													Reserved	DISSWD	LOCKBOOT
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

FLCn_PSTA indicates the protection mode setting state that was read at the automatic load.

- **LOCKBOOT** (bit 0)

  Indicates whether or not to enable the write/erase operation for the boot program area.

  0: Disable. 1: Enable.

- **DISSWD** (bit 1)

  Indicates whether or not to enable the connection to SWD.

  0: Disable. 1: Enable.

- **DISRISP** (bit 16)

  Indicates whether or not to disable the FLASH read & verify command in ISP.

  0: Disable. 1: Enable.

- **DISEPISP** (bit 17)

  Indicates whether or not to disable the FLASH erase & rewrite command in ISP.

  0: Disable. 1: Enable.

- **LOCKUSR** (bit 30)

  Indicates whether or not to enable the write/erase operation for the user program area.

  0: Disable. 1: Enable.

- **ERAALL** (bit 31)

  Indicates whether or not to execute the automatic erase of the whole user application FLASH area when the protection lock key is unlocked.

  0: Execute. 1: Do not execute.

7.19.3.8 Flash-ROM Size: BASE + 0x20

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FSI[31:0]																															
Initial value	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Indicates the Flash ROM capacity (byte).

[Note]

In 1 bank mode, the initial value of this register is 0x00080000 in the MAIN controller.

In 2 bank mode, the initial value of this register is 0x00040000 in the MAIN/SUB controller.

7.19.3.9 Boot Program Address: BASE + 0x24

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BPA[31:0]																															
Initial value	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Indicates the offset address from the beginning of the Flash ROM to the beginning of the boot program area.

The initial value of this register is automatically calculated from the parameter indicating the Flash ROM capacity.

[Note]

This register always indicates the address (relative address) of the boot program area #1.

7.19.3.10 Interrupt Mask Register: BASE + 0x28

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																											IMERR	IMERAC	IMERAB	IMERAS	IMPROW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

IMPROW	Word program complete interrupt mask. 1: Mask. 0: No mask.
IMERAS	Sector erase complete interrupt mask. 1: Mask. 0: No mask.
IMERAB	Block erase complete interrupt mask. 1: Mask. 0: No mask.
IMERAC	Chip erase complete interrupt mask. 1: Mask. 0: No mask.
IMERR	Error interrupt mask. 1: Mask. 0: No mask.

7.19.3.11 Interrupt Status Register: BASE + 0x2C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																											ISERR	ISERAC	ISERAB	ISERAS	ISPROW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

ISPROW	Indicates the word program complete interrupt status. 1: Indicates that the word program interrupt occurs. 0: Indicates that the word program interrupt does not occur.
ISERAS	Indicates the sector erase complete interrupt status. 1: Indicates that the sector erase interrupt occurs. 0: Indicates that the sector erase interrupt does not occur.
ISERAB	Indicates the block erase complete interrupt status. 1: Indicates that the block erase interrupt occurs. 0: Indicates that the block erase interrupt does not occur.
ISERAC	Indicates the chip erase complete interrupt status. 1: Indicates that the chip erase interrupt occurs. 0: Indicates that the chip erase interrupt does not occur.
ISERR	Indicates the error interrupt status. 1: Indicates that an error interrupt occurs. 0: Indicates that an error interrupt does not occur.

[Note]

Clearing the interrupt status register clears the RAW interrupt status register as well.

7.19.3.12 RAW Interrupt Status Register: BASE + 0x30

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																											IRERR	IRERAC	IRERAB	IRERAS	IRPROW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

IRPROW	Indicates the word program complete interrupt status (before mask). 1: Indicates that the word program interrupt (before mask) occurs. 0: Indicates that the word program interrupt (before mask) does not occur.
IRERAS	Indicates the sector erase complete interrupt status (before mask). 1: Indicates that the sector erase interrupt (before mask) occurs. 0: Indicates that the sector erase interrupt (before mask) does not occur.
IRERAB	Indicates the block erase complete interrupt status (before mask). 1: Indicates that the block erase interrupt (before mask) occurs. 0: Indicates that the block erase interrupt (before mask) does not occur.
IRERAC	Indicates the chip erase complete interrupt status (before mask). 1: Indicates that the chip erase interrupt (before mask) occurs. 0: Indicates that the chip erase interrupt (before mask) does not occur.
IRERR	Indicates the error interrupt status (before mask). 1: Indicates that an error interrupt (before mask) occurs. 0: Indicates that an error interrupt (before mask) does not occur.

[Note]

Clearing the RAW interrupt status register clears the interrupt status register as well.

7.19.3.13 Flash-ROM Error Status Register: BASE + 0xC4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved															Reserved	Reserved			ERAVERR3	ERAVERR2	Reserved				COMERR	Reserved		NONIMP	USERNOE	BOOTNOE	PROTNOE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	R	R	R	RW	RW	R	R	R	W	RW	W	W	RW	RW	RW	RW

This register indicates an error in a write/erase operation.

All bits of this register are enabled when the BUSY00 to BUSY11 bits of the Flash-ROM status register are set to 0H after a write/erase operation is performed.

Description of bits

- **PROTNOE** (bit 0)

  This bit is set when one of the write/partial erase operations described below is attempted in the protection set area.

  This bit is not cleared until it is cleared by software.

Condition for setting	Set by any one of the following conditions: ・ A sector erase in the protection set area is performed when the protection lock key is set ・ A write to the protection set area is performed when the protection lock key is set ・ A sector erase in the protection set area is performed by using FLCn_ERA ・ A write to the protection set area is performed by using FLCn_WDA
Condition for clearing	Can be cleared by any one of the following conditions: ・ Write 1 to the PROTNOE bit ・ Write 1 to the ISERR bit of FLCn_INTSTA ・ Write 1 to the IRERR bit of FLCn_RINTSTA

- **BOOTNOE** (bit 1)

  This bit is set when the area is protected by the protection function and a write/sector erase/block erase is attempted on the boot program area.

  This bit is not cleared until it is cleared by software.

Condition for setting	Set by any one of the following conditions: ・ A write/sector erase is performed on the protected boot program area or an address in it ・ A block erase is attempted by specifying the boot program area ・ A sector erase is performed on the boot program area by using FLCn_PROTERA ・ A write is performed on the boot program area by using FLCn_PROTWDA
Condition for clearing	Can be cleared by any one of the following conditions: ・ Write 1 to the BOOTNOE bit ・ Write 1 to the ISERR bit of FLCn_INTSTA ・ Write 1 to the IRERR bit of FLCn_RINTSTA

- **USERNOE** (bit 2)

  This bit is set when a write/erase (including chip erase) is attempted on the user application area protected by the protection function.

  This bit is not cleared until it is cleared by software.

Condition for setting	Set by any one of the following conditions: ・ A write/erase is performed on the protected user application area or an address in it ・ A sector erase is performed on the user application area by using FLCn_PROTERA ・ A write is performed on the user application area by using FLCn_PROTWDA
Condition for clearing	Can be cleared by any one of the following conditions: ・ Write 1 to the USERNOE bit ・ Write 1 to the ISERR bit of FLCn_INTSTA Write 1 to the IRERR bit of FLCn_RINTSTA

- **NONIMP** (bit 3)

  This bit is set when a write/partial erase is attempted on an unimplemented area.

  This bit is not cleared until it is cleared by software.

Condition for setting	A write/partial erase is performed on an unimplemented area
Condition for clearing	Can be cleared by any one of the following conditions: ・ Write 1 to the NONIMP bit ・ Write 1 to the ISERR bit of FLCn_INTSTA ・ Write 1 to the IRERR bit of FLCn_RINTSTA

- **COMERR** (bit 6)

  This bit is set when FLCn_ACP is not in the permitted state and a write access is made to FLCn_WDA or FLCn_WDA.

  This bit is not cleared until it is cleared by software.

Condition for setting	A write access is made to FLCn_WDA, FLCn_ERA, FLCn_PROTWDA, or FLCn_PROTERA when FLCn_ACP is not in the permitted state * Note that this bit is not set when **エラー! 参照元が見つかりません。** is in the permitted state (during debugging).
Condition for clearing	Can be cleared by any one of the following conditions: ・ Write 1 to the COMERR bit ・ Write 1 to the ISERR bit of FLCn_INTSTA

	・ Write 1 to the IRERR bit of FLCn_RINTSTA

- **ERAVERR2** (bit 11)

  This bit is set when FLCn_ACP is in the permitted state and a write access of 0H is made to the FLE field of FLCn_ERA.

  This bit is not cleared until it is cleared by software.

Condition for setting	A write access of 0H is made to the FLE field of FLCn_ERA when FLCn_ACP is in the permitted state
Condition for clearing	Can be cleared by any one of the following conditions: ・ Write 1 to the ERAVERR2 bit ・ Write 1 to the ISERR bit of FLCn_INTSTA ・ Write 1 to the IRERR bit of FLCn_RINTSTA

- **ERAVERR3** (bit 12)

  This bit is set when FLCn_ACP is in the permitted state and a write access of 0H, 1H or 2H is made to the FLPE field of FLCn_PROTERA.

  This bit is not cleared until it is cleared by software.

Condition for setting	A write access of invalid values (0H, 1H, 2H) is made to the FLPE field of FLCn_PROTERA when FLCn_ACP is in the permitted state
Condition for clearing	Can be cleared by any one of the following conditions: ・ Write 1 to the ERAVERR3 bit ・ Write 1 to the ISERR bit of FLCn_INTSTA ・ Write 1 to the IRERR bit of FLCn_RINTSTA

7.19.3.14 Protection Lock Key Unlock Register: BASE + 0xD0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROTUNLOCK[31:0]																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W	R W

This register is used to unlock the protection lock key that was set. When this register is written four times continuously, the protection lock key is unlocked if the written 4-word data matched the protection lock key.

When the protection lock key is unlocked, all contents in the protection set area are erased.

This register can be used only after the protection lock key is set (after the key is set and then enabled by the power-on reset).

The unlocking of the protection lock key is enabled by the power-on reset after the operation of this register.

Write to this register is enabled only for the MAIN side controller in 1 bank mode or for the SUB side controller in 2 bank mode.

Description of bits

- **PROTUNLOCK00-PROTUNLOCK31**

  By being written four times continuously, the written 4-word data are compared with the protection lock key.

7.19.3.15 Protection Lock Key Status Register: BASE + 0xD4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved															PROTLOCKKEYSTA	Reserved														PROTUNLOCKWE	PROTUNLOCKSTA
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

This register indicates the setting state of the protection lock key and whether the protection lock key is successfully unlocked.

Description of bits

- **PROTUNLOCKSTA**

  This bit indicates whether the protection lock key is successfully unlocked. It is set to 1 under the conditions described below.

  This bit is updated when PROTUNLOCKWE changes to 1.

  This bit is cleared when 1 is written to it.

  0: Successful. 1: Failed

  [Conditions for 1]

- Another register is written to before writing to FLCn_PROTUNLOCK four times continuously (unless the controller is executing or attempting to execute the unlock operation of the protection lock key after the protection lock key matches successfully.)
- The data written to FLCn_PROTUNLOCK does not match the protection lock key (unless the controller is executing or attempting to execute the unlock operation of the protection lock key after the protection lock key matches successfully.)
- FLCn_PROTUNLOCK is written to when the protection lock key is not set
- FLCn_PROTUNLOCK of a controller that does not control the protection set area is written to
- The lower 1 byte is not 0H when verifying the protection lock setting information data
- FLCn_PROTUNLOCK is written to when PROTUNLOCKWE = 0 (unless the controller is executing or attempting to execute the unlock operation of the protection lock key after the protection lock key matches successfully.)

- **PROTUNLOCKWE**

  This bit indicates whether or not to enable write to FLCn_PROTUNLOCK.

  Write to FLCn_PROTUNLOCK is enabled only when this bit is 1.

  0: Write disabled. 1: Write enabled

  This bit is set to 0 under the following conditions.

  [Conditions for 0]

  - During write/erase operation for Flash-ROM (any bit of BUSY00-BUSY11 is set to 1)
  - During comparison of the protection lock key or erase operation related to the unlocking of the protection lock key
  - VOLTAGE bit = 1

- **PROTLOCKKEYSTA**

  This bit indicates the setting state of the protection lock key.

  0: Set. 1: Not set

7.19.3.16 Flash-ROM Protection Set Write Data Register: BASE + 0xD8

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FPD[31:0]																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

The FLCn_PROTWDA register sets the Flash-ROM rewrite data.

- **FPD31 to FPD00** (bits 31-0)
  The FPD31 to FPD00 bits set data to be written when 1-word write to the protection set area is performed.
  Writing to the FPD31 to FPD00 bits starts the 1-word write.

[Note]

Clear the contents of the target addresses in advance. The content of an overwritten address is not guaranteed.

Write to the user application area and the boot program area cannot be performed by using this register.

7.19.3.17 Flash-ROM Protection Set Erase Register: BASE + 0xDC

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														FLPE	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

This register starts the sector erase of Flash-ROM.

- **FLPE[1:0]** (bits 1-0)

  The FLPE bit specifies the sector erase start for the protection set area.

  Writing to FLPE starts erase according to the data. It automatically changes to "00" when the erase is completed. Writing "00" is prohibited. This bit remains as "00" when FLCn_ACP is not in the permitted state except during erase/write operation, or when a write to this field is performed. It remains the same as before write if a write to this field is performed during erase/write operation.

FLPE[1:0]	Description
00	Erase completed (initial value)
01	Undefined
10	Undefined
11	Start sector erase

[Note]

Erase of the user application area and the boot program area cannot be performed by using this register.

7.19.3.18 Flash-ROM Configuration Information Register: BASE + 0x100

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved															BANK_MODE	Reserved								BANK_SEL	Reserved							NCON	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register indicates the controller configuration and mode.

Description of bits

- **NCON** (bits 1-0)

  Indicates the maximum number of controllers in the configuration. This field is fixed to 2H.

- **BANK_SEL** (bit 8)

  Indicates the currently selected bank in 2 bank mode.

  0: Bank 0 (code area 0), 1: Bank 1 (code area 1)

- **BANK_MODE** (bit 16)

  Indicates the currently selected bank mode.

  0: 1 bank mode, 1: 2 bank mode.

7.19.3.19 Flash-ROM Code User Application Area #0 Address Register: BASE + 0x110

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FCU0ADR[31:0]																															
Initial value	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register indicates the beginning address (absolute address) of the user application area #0 in the code area.

If the read value of this register is 0H, it means that the corresponding area does not exist.

Description of bits

- **FCU0ADR31 to FCU0ADR00** (bits 31-0)

  FCU0ADR31 to FCU0ADR00 indicate the beginning address of the user application area #0 in the code area.

7.19.3.20 Flash-ROM Code Protection Set Area #0 Address Register: BASE + 0x114

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FCP0ADR[31:0]																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register indicates the beginning address (absolute address) of the protection set area #0 in the code area.

If the read value of this register is 0H, it means that the corresponding area does not exist.

Description of bits

- **FCP0ADR31 to FCP0ADR00** (bits 31-0)

  FCP0ADR31 to FCP0ADR00 indicate the beginning address of the protection set area #0 in the code area.

  The initial value is 0H because the protection set area #0 does not exist.

7.19.3.21 Flash-ROM Code Boot Program Area #0 Address Register: BASE + 0x118

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FCB0ADR[31:0]																															
Initial value	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register indicates the beginning address (absolute address) of the boot program area #0 in the code area.

If the read value of this register is 0H, it means that the corresponding area does not exist.

Description of bits

- **FCB0ADR31 to FCB0ADR00** (bits 31-0)

  FCB0ADR31 to FCB0ADR00 indicate the beginning address of the boot program area #0 in the code area.

7.19.3.22 Flash-ROM Code Area #0 Size Register: BASE + 0x11C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FC0SIZ[31:0]																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

This register indicates the size of the code area #0.

Description of bits

- **FC0SIZ31 to FC0SIZ00** (bits 31-0)

  FC0SIZ31 to FC0SIZ00 indicate the size of the code area #0.

7.19.3.23 Flash-ROM Data User Application Area #0 Address Register: BASE + 0x120

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FDU0ADR[31:0]																															
Initial value	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register indicates the beginning address (absolute address) of the user application area #0 in the data area.

If the read value of this register is 0H, it means that the corresponding area does not exist.

Description of bits

- **FDU0ADR31 to FDU0ADR00** (bits 31-0)

  FDU0ADR31 to FDU0ADR00 indicate the beginning address of the user application area #0 in the data area.

7.19.3.24 Flash-ROM Data Protection Set Area #0 Address Register: BASE + 0x124

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FDP0ADR[31:0]																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register indicates the beginning address (absolute address) of the protection set area #0 in the data area.

If the read value of this register is 0H, it means that the corresponding area does not exist.

Description of bits

- **FDP0ADR31 to FDP0ADR00** (bits 31-0)

  FDP0ADR31 to FDP0ADR00 indicate the beginning address of the protection set area #0 in the data area.

  The initial value is 0H because the protection set area #0 does not exist.

7.19.3.25 Flash-ROM Data Boot Program Area #0 Address Register: BASE + 0x128

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FDB0ADR[31:0]																															
Initial value	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register indicates the beginning address (absolute address) of the boot program area #0 in the data area.

If the read value of this register is 0H, it means that the corresponding area does not exist.

Description of bits

- **FDB0ADR31 to FDB0ADR00** (bits 31-0)

  FDB0ADR31 to FDB0ADR00 indicate the beginning address of the boot program area #0 in the data area.

7.19.3.26 Flash-ROM Data Area #0 Size Register: BASE + 0x12C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FD0SIZ[31:0]																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

This register indicates the size of the code area #0.

Description of bits

- **FD0SIZ31 to FD0SIZ00** (bits 31-0)

  FD0SIZ31 to FD0SIZ00 indicate the size of the data area #0.

7.19.3.27 Flash-ROM Code User Application Area #1 Address Register: BASE + 0x140

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FCU1ADR[31:0]																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register indicates the beginning address (absolute address) of the user application area #1 in the code area.

If the read value of this register is 0H, it means that the corresponding area does not exist.

Description of bits

- **FCU1ADR31 to FCU1ADR00** (bits 31-0)

  FCU1ADR31 to FCU1ADR00 indicate the beginning address of the user application area #1 in the code area.

7.19.3.28 Flash-ROM Code Protection Set Area #1 Address Register: BASE + 0x144

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FCP1ADR[31:0]																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register indicates the beginning address (absolute address) of the protection set area #1 in the code area.

If the read value of this register is 0H, it means that the corresponding area does not exist.

Description of bits

- **FCP1ADR31 to FCP1ADR00** (bits 31-0)

  FCP1ADR31 to FCP1ADR00 indicate the beginning address of the protection set area #1 in the code area.

  The initial value is 0H because the protection set area #0 does not exist.

7.19.3.29 Flash-ROM Code Boot Program Area #1 Address Register: BASE + 0x148

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FCB1ADR[31:0]																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register indicates the beginning address (absolute address) of the boot program area #1 in the code area.

If the read value of this register is 0H, it means that the corresponding area does not exist.

Description of bits

- **FCB1ADR31 to FCB1ADR00** (bits 31-0)

  FCB1ADR31 to FCB1ADR00 indicate the beginning address of the boot program area #0 in the code area.

7.19.3.30 Flash-ROM Code Area #1 Size Register: BASE + 0x14C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FC1SIZ[31:0]																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

This register indicates the size of the code area #0.

Description of bits

- **FC1SIZ31 to FC1SIZ00** (bits 31-0)

  FC1SIZ31 to FC1SIZ00 indicate the size of the code area #0.

7.19.3.31 Flash-ROM Data User Application Area #1 Address Register: BASE + 0x150

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FDU1ADR[31:0]																															
Initial value	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register indicates the beginning address (absolute address) of the user application area #1 in the data area.

If the read value of this register is 0H, it means that the corresponding area does not exist.

Description of bits

- **FDU1ADR31 to FDU1ADR00** (bits 31-0)

  FDU1ADR31 to FDU1ADR00 indicate the beginning address of the user application area #1 in the data area.

7.19.3.32 Flash-ROM Data Protection Set Area #1 Address Register: BASE + 0x154

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FDP1ADR[31:0]																															
Initial value	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register indicates the beginning address (absolute address) of the protection set area #1 in the data area.

If the read value of this register is 0H, it means that the corresponding area does not exist.

Description of bits

- **FDP1ADR31 to FDP1ADR00** (bits 31-0)

  FDP1ADR31 to FDP1ADR00 indicate the beginning address of the protection set area #1 in the data area.

  The initial value is 0H because the protection set area #0 does not exist.

7.19.3.33 Flash-ROM Data Boot Program Area #1 Address Register: BASE + 0x158

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FDB1ADR[31:0]																															
Initial value	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register indicates the beginning address (absolute address) of the boot program area #1 in the data area.

If the read value of this register is 0H, it means that the corresponding area does not exist.

Description of bits

- **FDB1ADR31 to FDB1ADR00** (bits 31-0)

  FDB1ADR31 to FDB1ADR00 indicate the beginning address of the boot program area #1 in the data area.

7.19.3.34 Flash-ROM Data Area #1 Size Register: BASE + 0x15C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FD1SIZ[31:0]																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

This register indicates the size of the code area #1.

Description of bits

- **FD1SIZ31 to FD1SIZ00** (bits 31-0)

  FD1SIZ31 to FD1SIZ00 indicate the size of the data area #1.

7.19.4 Functional Description

7.19.4.1 Wait/Prefetch Control

Presence or absence of the AHB bus maximum wait and prefetch function can be selected by the FLWA bit of the Flash-ROM control. The maximum wait cycle and prefetch function cannot be controlled independently. The prefetch function is disabled at the time of 0 wait setting and enabled at the time of 1 wait setting. When the prefetch function is enabled, CPU can be operated at about 1/2 cycles of the memory access time.

The prefetch function is a function to pre-read the data stored at the address following the previously read one while no valid access to the Flash ROM controller exists. It can improve the performance by reducing the system clock cycle as compared to the program memory read latency.

7.19.4.2 Erase/Program Function

The erase and program is executed by starting the Flash ROM control register placed within the APB space. The following shows the erase and program procedures.

7.19.4.2.1. Chip Erase

The following shows the chip erase procedure.

(1) Write 0x000000FA to FLCACP
(2) Write 0x000000F5 to FLCACP
(3) Write 0x00000001 to FLCERA
(4) Wait for FLCSTA to be 0x00000000.

7.19.4.2.2. Block Erase

The following shows the block erase procedure.

(1) Write 0x000000FA to FLCACP
(2) Write 0x000000F5 to FLCACP
(3) Write the beginning address of the erase target block to FLCADR
(4) Write 0x00000002 to FLCERA
(5) Wait for FLCSTA to be 0x00000000.

7.19.4.2.3. Sector Erase

The following shows the sector erase procedure.

(1) Write 0x000000FA to FLCACP

(2) Write 0x000000F5 to FLCACP

(3) Write the beginning address of the erase target sector to FLCADR

(4) Write 0x00000003 to FLCERA

(5) Wait for FLCSTA to be 0x00000000.

7.19.4.2.4 Word Program

The following shows the 1-word program procedure.

(1) Write 0x000000FA to FLCACP

(2) Write 0x000000F5 to FLCACP

(3) Write the address of program target word to FLCADR

(4) Write the data to be written to FLCWDA

(5) Wait for FLCSTA to be 0x00000000.

7-20.DIO

7.20.1 General Description

Data transmission/reception interface dedicated to the RF block (ML7396B) control.

[Notices]

When this DIO function is used, release the pull-down setting of DIO_CPU(LSI internal pin).

For release method, refer to ", 7.29.3.28 IO Setting Register 0-25".

7.20.2 List of Registers

Address [H]	Name	Symbol	R/W	Initial value [H]	Description
0x40043000	DIO enable	DIO_EN	R/W	0x00000000	Enables the DIO function.
0x40043004	DIO control	DIO_CTRL	R/W	0x00000008	Controls the DIO operation mode.
0x40043008	Receive data matching pattern length setting	DIO_MATCH_LEN	R/W	0x00000000	Sets the receive data matching pattern length.
0x4004300C	Receive data matching pattern setting	DIO_MATCH_PAT	R/W	0x00000000	Sets the receive data matching pattern.
0x40043010	FIFO-FULL threshold setting	DIO_F_FULL_TRG	R/W	0x0000000F	Sets the threshold of FIFO-FULL.
0x40043014	FIFO-EMPTY threshold setting	DIO_F_EMPTY_TRG	R/W	0x00000000	Sets the threshold of FIFO-EMPTY.
0x40043018	FIFO valid number of data display	DIO_F_LEV	R	0x00000000	Displays the number of valid data within the FIFO.
0x4004301C	Interrupt mask	DIO_IMSK	RW	0x0000001F	Masks the interrupt status.
0x40043020	Interrupt status	DIO_IST	R	0x00000000	Displays the interrupt status (after mask).
0x40043024	Interrupt RAW status	DIO_RIST	R	0x00000008	Displays the interrupt status (before mask).
0x40043028	FIFO overflow interrupt clear	DIO_F_O_CLR	R	0x00000000	Clears FIFO overflow interrupt.

0x4004302C	FIFO underflow interrupt clear	DIO_F_U_CLR	R	0x00000000	Clears FIFO underflow interrupt.
0x40043030	Received data coincident interrupt clear register	DIO_MATCH_CLR	R	0x00000000	Clears the received data coincident interrupt.
0x40043034	Interrupt clear	DIO_ICLR	R	0x00000000	Clears all interrupts within the DIO function.
0x40043038	Transmit/receive data	DIO_FIFO	R/W	0x00000000	Stores the transmit data during the transmission. Stores the receive data during the reception.

7.20.3 Description of Registers

7.20.3.1 DIO_EN Register: 0x40043000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																														DIO_TX_EN	DIO_RX_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

DIO_TX_ON	Enables the DIO transmit mode. 0: disable 1: enable
DIO_RX_EN	Enables the DIO receive mode. 0: disable 1: enable

7.20.3.2 DIO_CTRL Register: 0x40043004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																										DOUT_MOD		PRST_EN	MATCH_EN	DIO_FAM	DIO_EDGE_SEL
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW

DOUT_MOD	TX DIO pin output mode setting 00: OD output mode (default) 01: CMOS output mode 10: OD/CMOS automatic switching mode 11: Reserved (OD output mode) * It is recommended that 00 (default) setting is used.
PRST_EN	1st data preset setting 0: disable 1: enable * It is recommended that 0 (default) setting is used.
MATCH_EN	Received data matching function enable 0: disable 1: enable

DIO_FAM	FIFO access mode setting (1 byte/4 bytes) 0: Accesses to FIFO in 1 byte. (High-order 3 bytes of write and read data are disabled) 1: Accesses to FIFO in 4 byte.
DCLK_EDGE_SEL	DCLK polarity selection 0: pos edge 1: neg edge

7.20.3.3 DIO_MATCH_LEN Register: 0x40043008

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										MATCH_LEN					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW

MATCH_LEN	Sets the receive data matching pattern length. A value from 0 to 32 can be set. 32 or more setting is assumed to be 32.

7.20.3.4 DIO_MATCH_PAT Register: 0x4004300C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MATCH_PAT																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

MATCH_PAT	Sets the receive data matching pattern. Bits less than the setting value of MATCH_LEN are enabled. Example: When MATCH_LEN = 0x10, bits 0 to 15 of MATCH_PAT are enabled.

7.20.3.5 DIO_F_FULL_TRG Register: 0x40043010

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												F_FULL_TRG			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

F_FULL_TRG	FIFO-FULL threshold setting

7.20.3.6 DIO_F_EMPTY_TRG Register: 0x40043014

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												F_EMPTY_TRG			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

F_EMPTY_TRG	FIFO-EMPTY threshold setting

7.20.3.7 DIO_F_LEV Register: 0x40043018

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																											F_LEV				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

F_LEV	FIFO valid number of data display

7.20.3.8 DIO_IMSK Register: 0x4004301C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																											IMSK				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

IMSK[4]	Masks the MATCH interrupt. 1: Mask 0: No mask
IMSK[3]	Masks the EMPTY interrupt. 1: Mask 0: No mask
IMSK[2]	Masks the UNDER interrupt. 1: Mask 0: No mask
IMSK[1]	Masks the OVER interrupt. 1: Mask 0: No mask
IMSK[0]	Masks the FULL interrupt. 1: Mask 0: No mask

7.20.3.9 DIO_IST Register: 0x40043020

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																											IST				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

IST[4]	Indicates the interrupt status.
IST[3]	For details of each bit, refer to the interrupt RAW status register. The masked interrupt status is read from this register.
IST[2]	To clear the interrupt source, read the interrupt clear register corresponding to the appropriate interrupt source.
IST[1]	
IST[0]	

7.20.3.10 DIO_RIST Register: 0x40043024

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																											RIST				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

RIST[4]	Indicates the MATCH interrupt status. This bit is set when the received data matches with MATCH_PAT.
RIST[3]	Indicates the EMPTY interrupt status. This bit is set when the valid data within FIFO is less than or equal to F_EMPTY_TRG. This bit is automatically cleared when the valid data exceeds F_EMPTY_TRG.
RIST[2]	Indicates the UNDER interrupt status. This bit is set when transmitting data while the valid data within FIFO is 0 or when reading data from HOST. In addition, if the FIFO access unit is 4 bytes (DIO_FAM = 1), this bit is set when reading data from HOST while the valid data within FIFO is 3 or less.
RIST[1]	Indicates the OVER interrupt status. This bit is set when receiving data while the valid data within FIFO is 16 or when setting data from HOST. In addition, if the FIFO access unit is 4 bytes (DIO_FAM = 1), this bit is set when setting data from HOST while the valid data within FIFO is 13 or more.
RIST[0]	Indicates the FULL interrupt status. This bit is set when the valid data within FIFO exceeds F_FULL_TRG. This bit is automatically cleared when the valid data is less than or equal to F_FULL_TRG.

7.20.3.11 DIO_F_O_CLR Register: 0x40043028

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															F_O_CLR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

F_O_CLR	When this register is read, the FIFO overflow interrupt is cleared. Writing is ignored.

7.20.3.12 DIO_F_U_CLR Register: 0x4004302C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															F_U_CLR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

F_U_CLR	When this register is read, the FIFO underflow interrupt is cleared. Writing is ignored.

7.20.3.13 DIO_MATCH_CLR Register: 0x40043030

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															MATCH_CLR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

MATCH_CLR	When this register is read, the received data coincident interrupt is cleared. Writing is ignored. [NOTICE] Before clearing the received data coincident interrupt, be sure to make the DIO receive mode disabled (DIO_RX_EN = 0).

7.20.3.14 DIO_ICLR Register: 0x40043034

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															ICLR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

ICLR	When this register is read, all active interrupts are cleared. Writing is ignored.

7.20.3.15 DIO_FIFO Register: 0x40043038

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	FIFO																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

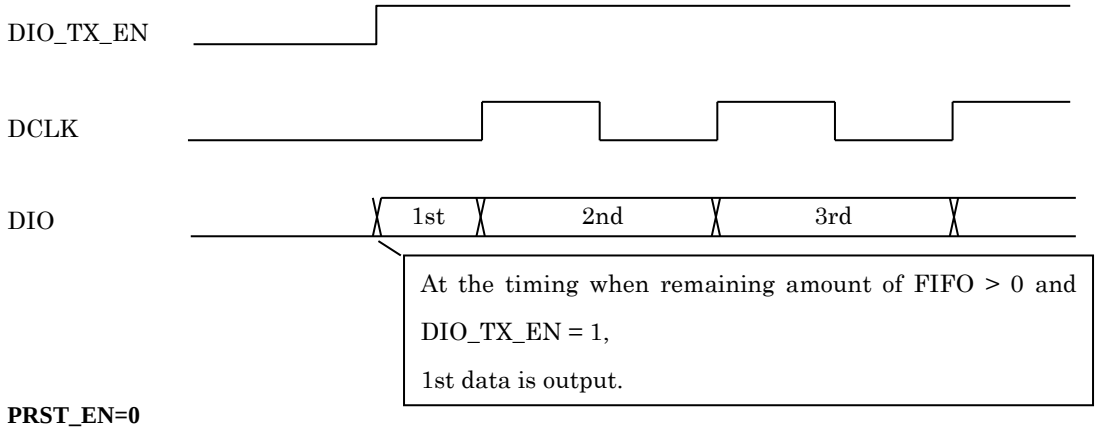
FIFO	Transmit/receive data (common to transmit/receive)

7.20.4 Functional Description

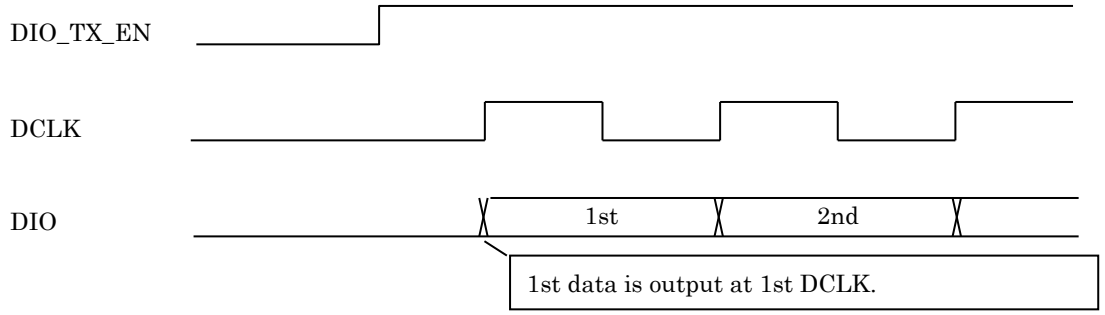
7.20.4.1 1st Data Transmit Timing Setting

1st data transmit timing can be switched by PRST_EN.

PRST_EN=1



PRST_EN=0



7.20.4.2 Received Data Matching Function

Confirms that the received data matches with the maximum of 32-bit pattern set in the MATCH_PAT register.

ON/OFF of this function can be controlled by the MATCH_EN register.

The length of matching pattern can be set to the range of 0 to 32 bits by the MATCH_LEN register.

Example: When MATCH_LEN = 0x10, MATCH_PAT[15:0] is enabled and the MATCH_PAT[31:16] setting become Don’t Care.

When the received data matches with MATCH_PAT, the MATCH interrupt occurs.

If this function is set to ON, data before matching is not stored in the receive FIFO.

7.20.5 Description of Operation

7.20.5.1 Transmit Operation

(1) Set PRST_EN, DIO_FAM, DCLK_EDGE_SEL, and F_EMPTY_TRG to any value.

(2) Set the transmit data in FIFO.

(3) Set DIO_TX_EN. DIO outputs the transmit data from MSB in synchronization with DCLK.

(4) When the EMPTY interrupt occurs, then set the transmit data.

(5) FIFO becomes empty and an UNDER interrupt occurs when the last data transmit is completed.

(6) When the transmit is completed, set DIO_TX_EN to 0.

7.20.5.2 Receive Operation

(1) Set MATCH_EN, DIO_FAM, DCLK_EDGE_SEL, MATCH_LEN, MATCH_PAT, and F_FULL_TRG to any value.

(2) Set DIO_RX_EN. DIO receives the received data from MSB in synchronization with DCLK and stores the data by 1 byte in FIFO. If MATCH_EN is set to 1, the received data is discarded without stored in FIFO until it matches with MATCH_PAT. When matched, a MATCH interrupt is notified and storing data in FIFO is started. If MATCH_EN is set to 0, all received data is stored in FIFO.

(3) When the FULL interrupt occurs, data is read from FIFO.

(4) When the reception of all data is completed, set DIO_RX_EN to 0.

The completion of received data is determined by the receive completion interrupt from RF-IC or by analyzing length from the received data. Since only data after the length can be stored in FIFO by setting SyncWord to MATCH_PAT, the analysis of length is made easy.

7-21.RAND_GEN

7.21.1 General Description

Pseudo-random number generation circuit. The features are shown below.

- The RAND length can be selected from RAND9, RAND15, and RAND23.
- Random number generation result can be output with the logical complement of 2.

7.21.2 List of Registers

Address [H]	Name	Symbol	R/W	Initial value [H]	Description
0x40044000	RAND_GEN control register 0	RAND_CR0	R/W	0x00000000	Controls the operation mode
0x40044004	RAND_GEN control register 1	RAND_CR1	R/W	0x00000000	Controls the operation mode
0x40044008	RAND_GEN random value register	RAND_VR	R	0x00000000	Displays the generated random value

7.21.3 Description of Registers

7.21.3.1 RAND_GEN Control Register 0: 0x40044000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																											RAND_COMP_ON	RAND_LEN		RAND_INIT	RAND_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

RAND_COMP_ON	Outputs the RAND generation result with the logical complement of 2. 0: disable 1: enable
RAND_LEN	Sets the RAND length. 00: RAND9 01: RAND15 10: RAND23 11: Do not use
RAND_INIT	Initializes the RAND generation circuit. 0: disable 1: enable

RAND_EN	Enables the RAND generation circuit. 0: disable 1: enable

7.21.3.2 RAND_GEN Control Register 1: 0x40044004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved	RAND_poly																														
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

RAND_POLY	Sets RAND POLY.

7.21.3.3 RAND_GEN Random Value Register: 0x40044008

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RAND_CALC_OUT																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

RAND_CALC_OUT	Stores the RAND generation result.

7.22.CLK_Timer

7.22.1 General Description

This function uses the high-speed clock to count a certain time period of the low-speed clock and stores the count result in a register. The frequency shift from the ideal clock can be determined from the count result (automatic correction of frequency is not supported).

The low-speed clock can be selected from low-speed CR oscillation circuit output, XTAL32kHz oscillation circuit output, and clock output from RF chip. The high-speed clock can be selected from high-speed CR oscillation circuit output, PLL output, and RF clock.

7.22.2 List of Registers

Address [H]	Name	Symbol	R/W	Initial value [H]	Description
0x40045000	CT control register	CT_CR	R/W	0x00000000	Controls the operation mode
0x40045004	CT timer register 0	CT_TR	R/W	0x00000000	Sets the timer value.
0x40045008	CT status register	CT_SR	R	0x00000000	Displays the count completion notification
0x4004500C	CT timer count value register	CT_TCR	R	0x00000000	Displays the count value
0x40045010	CT interrupt clear register	CT_TCL	R	0x00000000	Clears the interrupt.

7.22.3 Description of Registers

7.22.3.1 CT Control Register: 0x40045000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																														CLK_CAL_INT_MASK	CLK_CAL_START
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

CLK_CAL_INT_MASK	Masks the CLK_TIMER interrupt. 0: disable 1: enable
CLK_CAL_START	Enables CLK_TIMER. 0: disable 1: enable

7.22.3.2 CT Timer Register 0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																CLK_CAL_TIME															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

CLK_CAL_TIME	Sets the timer value equal to the number of cycles of the low-speed clock (such as XTAL32KHz).

7.22.3.3 CT Status Register: 0x40045008

	31–1	0
	Reserved	CLK_CAL_END
Initial value	0	0
R/W	R	R

CLK_CAL_END	Notifies the timer expiration.

7.22.3.4 CT Timer Count Value Register: 0x4004500C

	31–24	23–16	15–8	7–0
	Reserved	CLK_CAL		
Initial value	0	0	0	0
R/W	R	R	R	R

CLK_CAL	Stores the CLK count value.

7.22.3.5 CT Interrupt Clear Register: 0x40045010

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															CLK_CAL_CLEAR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

CLK_CAL_CLEAR	Clears the interrupt. * When this register is read, the interrupt is cleared. Always 0 is read when reading.

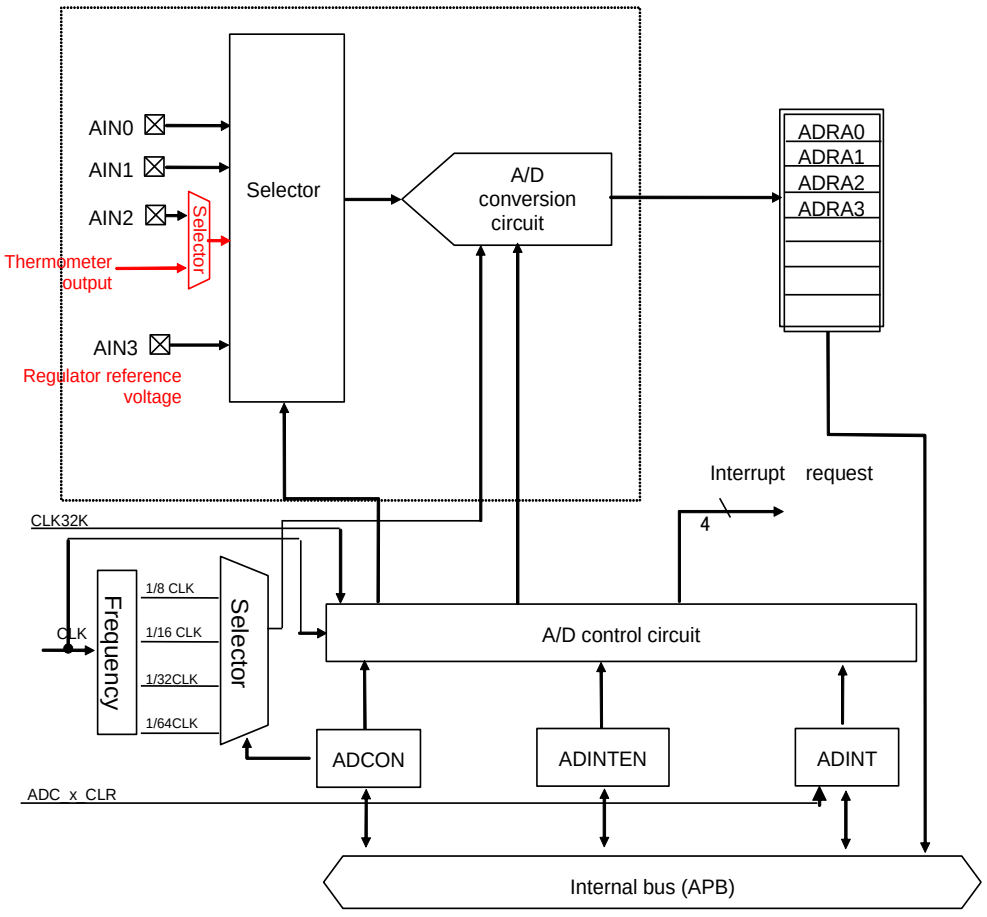
7.23. ADC

7.23.1 General Description

This function controls the 10-bit successive approximation type A/D converter.

The features are shown below.

- Programmable scan of up to three channels (CH0 to CH2) (The scan time and scan order can be set.)
  * Two channels when using the temperature sensor (CH2).
- Scan result notification (The scan completion is notified by an interrupt.)
- Averaging of A/D conversion data (The average value of A/D conversion results is displayed.)
- Calculation of CH0 to CH2 input voltage (It is assumed that the reference voltage output from the regulator at CH3 is monitored.)



7.23.2 List of Registers

Address [H]	Name	Symbol	R/W	Initial value [H]	Description
0x40070000	A/D control register	ADCON	R/W	0x0000_0000	Sets the operation mode.
0x40070004	A/D interrupt control register	ADINT	R/W	0x0000_0000	Notifies the interrupt.
0x40070008	A/D interrupt enable control register	ADINTEN	R/W	0x0000_0000	Sets the interrupt enable.
0x4007000C	A/D result register (AVE) CH0	ADRST0	R	0x0000_0000	Displays the A/D conversion result (average value) of CH0.
0x40070010	A/D result register (AVE) CH1	ADRST1	R	0x0000_0000	Displays the A/D conversion result (average value) of CH1.
0x40070014	A/D result register (AVE) CH2	ADRST2	R	0x0000_0000	Displays the A/D conversion result (average value) of CH2.
0x40070018	A/D result register (AVE) CH3	ADRST3	R	0x0000_0000	Displays the A/D conversion result (average value) of CH3.
0x4007001C	A/D data register 0	ADDT0	R/W	0xFFFF_FD30	Displays the A/D conversion result under the conditions of low temperature (-40 °C) and VDDIO_CPU = 3.3 V, which is stored in Flash in advance.
0x40070020	A/D data register 1	ADDT1	R/W	0xFFFF_FCEC	Displays the A/D conversion result under the conditions of ordinary temperature (25 °C) and VDDIO_CPU = 3.3V, which is stored in Flash in advance.
0x40070024	A/D data register 2	ADDT2	R/W	0xFFFF_FCAB	Displays the A/D conversion result under the conditions of high temperature (105 °C) and VDDIO_CPU = 3.3 V, which is stored in Flash in advance.
0x40070028	A/D data register 3	ADDT3	R/W	0xFFFF_FD75	Displays the A/D conversion result of CH3 under the conditions of VDDIO_CPU = 3.3 V, which is stored in Flash in advance.

[Note]

When ADCON is written during A/D conversion (ADRUN=1), all results of A/D conversion from ADRST0 to ADRST3 are not guaranteed.

7.23.3 Description of Registers

7.23.3.1 A/D Control Register: 0x40070000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	ADCALR3	ADCALR2	ADCALR1	ADCALR0	ADCAL3	ADCAL2	ADCAL1	ADCAL0	Reserved			EXTEMP_SEL	ADWAIT3	ADWAIT2	ADWAIT1	ADWAIT0	Reserved						ADCYC1	ADCYC0	ADCHG1	ADCHG0	ADAST	ADRUN	ADSSM3	ADSSM2	ADSSM1	ADSSM0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	RW	RW	RW	RW	RW	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

ADSSM

Specifies the scan channel(s) for selection or scan mode. When only one bit is set to "1", the register operates in selection mode, and when multiple bits are set to "1", the register operates in scan mode. In scan mode, A/D conversion is executed in ascending order of the channel number.

ADSSM				Channel to use	Order of channel selection/channel scan	Mode
3	2	1	0			
0	0	0	0	-	-	-
0	0	0	1	ch0	ch0	Selection
0	0	1	0	ch1	ch1	Selection
0	0	1	1	ch0,ch1	ch0→ch1	Scan
0	1	0	0	ch2	ch2	Selection
0	1	0	1	ch0,ch2	ch0→ch2	Scan
0	1	1	0	ch1,ch2	ch1→ch2	Scan
0	1	1	1	ch0,ch1,ch2	ch0→ch1→ch2	Scan
1	0	0	0	ch3	ch3	Selection
1	0	0	1	ch0,ch3	ch0→ch3	Scan
1	0	1	0	ch1,ch3	ch1→ch3	Scan
1	0	1	1	ch0,ch1,ch3	ch0→ch1→ch3	Scan
1	1	0	0	ch2,ch3	ch2→ch3	Scan
1	1	0	1	ch0,ch2,ch3	ch0→ch2→ch3	Scan
1	1	1	0	ch1,ch2,ch3	ch1→ch2→ch3	Scan
1	1	1	1	ch0,ch1,ch2,ch3	ch0→ch1→ch2→ch3	Scan

[Note]

The channel(s) to use should be changed when A/D conversion is stopped.

(The scan channel(s) cannot be changed unless ADRUN = "0")

ADRUN

Specifies the start/stop of A/D conversion.

0: Stops A/D conversion

1: Starts A/D conversion

How to stop A/D conversion differs depending on the setting value of ADAST described below.

When ADAST = 0: ADRUN is automatically cleared by hardware ("0" write) after a set of A/D conversions.

When ADAST = 1: It is necessary to write "0" to ADRUN after a set of A/D conversions.

To start the next A/D conversion, it is necessary to write "1" to ADRUN.

The previous A/D conversion result is held until the next time ADRUN is set to "1".

However, if the interrupt is not cleared at the next time ADRUN is set to "1", this ADRUN setting is disabled.

[Note]

If no bit of ADSSM is set to "1", A/D conversion is not started regardless of the setting value of ADRUN.

If ADSSM=4'h0 is set during conversion, A/D conversion is not terminated (it is necessary to set ADRUN = 0 in order to forcibly terminate the A/D conversion).

ADAST

It is possible to specify whether or not to perform the A/D conversion again after one cycle of A/D conversions for the specified channels. This is supported in both selection and scan modes.

<table>
<tr><td></td><td>0: Stops the conversion operation after A/D conversion for the specified channels
1: Starts A/D conversion again after A/D conversion for the specified channels</td></tr>
<tr><td>ADCHG</td><td>Specifies the number of conversion times of the A/D converter. One, four, eight, or sixteen times can be specified.
The average value of the specified number of conversions is output as the A/D conversion result.
00: One time
01: Four times
10: Eight times
11: Sixteen times</td></tr>
<tr><td>ADCYC</td><td>Selects the frequency of the operation clock of A/D converter.
[Note]
Do not change the operation clock during A/D conversion (ADRUN = 1).
Use the operation clock of 2.5 MHz or 1.25 MHz.
The minimum and maximum frequencies of the operation clock are 400 ns (2.5 MHz) and 800 ns (1.25 MHz) respectively, and 25 clocks are required for A/D conversion.
The A/D conversion time per channel is decided by the ADCYC setting value.
For example, when ADCYC[1:0] is set to "01", the conversion time is 800 ns × 25 = 20 µs.
Operation Clock and A/D Conversion Time Decided by ADCYC (Example)
<table>
<tr><td rowspan="2">CLK (MHz)</td><td colspan="2">ADC operation clock (min400ns / max800ns)</td><td colspan="2">A/D conversion time (min10µs / max 20µs)</td></tr>
<tr><td>ADCYC=00 (No division)</td><td>ADCYC=01 (2 dividing)</td><td>ADCYC=00 (No division)</td><td>ADCYC=01 (2 dividing)</td></tr>
<tr><td>2.5</td><td>400ns</td><td>800ns</td><td>10µs</td><td>20µs</td></tr>
</table>
When the clock cycle = 800 ns, the number of A/D conversion times = 16, and the selected channels = four channels, the A/D conversion time will have the maximum value, which is 800 [ns] x 25 [cycle] x (16 + 2 [times]) x 4 [channels] = 1.44 [ms].</td></tr>
<tr><td>ADWAIT</td><td>Specifies the interval time of detecting the A/D converter output value (wait time from the A/D conversion start).
8' h00=7.81 ms
8' h01=15.62 ms
8' h02=31.25 ms
8' h03=62.5 ms
8' h04=125 ms
8' h05=250 ms
8' h06=500 ms
8' h07=1000 ms
8' h08=2000 ms
8' h09=4000 ms</td></tr>
</table>

	8' h0a=8000 ms 8' h0b to 8' h0f=16000 ms The wait period includes the A/D conversion time. The interrupt process should be handled on the CPU side during the wait period. The value of the A/D result register described below is held during the wait period. Up to 16000 ms can be set for the wait time. Note that wait is not inserted between channels in the scan mode.
EXTEMP_SEL	Selects the CH2 input of ADC. When EXTEMP_SEL = 0: Selects the input from thermometer When EXTEMP_SEL = 1: Selects the input from external pin ADC2
ADCAL	Acquires the average of A/D converter output values when the corresponding bit is set to 1. Each bit (bits 24-27) corresponds to each channel (CH0-CH3). This register is only enabled when the number of A/D conversion times is set to four or more. 0: Stores the latest A/D conversion result 1: Stores the average value of A/D conversion results
ADCALR	Makes correction based on the A/D conversion result of CH3 when the corresponding bit is set to 1 (calculated up to the second decimal place and truncated at the third decimal place), and stores the calculation result in the A/D result register (ADRST0 to 3). Each bit (bits 28-31) corresponds to each channel (CH0-CH3). 0: Does not make correction. 1: Makes correction using the following formula. Correction formula: (CH0/1/2 conversion result) / (CH3 A/D conversion result)

7.23.3.2 A/D Interrupt Control Register: 0x40070004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												ADINT3	ADINT2	ADINT1	ADINT0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

ADINT	Indicates the completion of A/D conversion for each channel. ADINT0 0: A/D conversion of channel 0 is not completed 1: A/D conversion of channel 0 is completed ADINT1 0: A/D conversion of channel 1 is not completed 1: A/D conversion of channel 1 is completed ADINT2 0: A/D conversion of channel 2 is not completed 1: A/D conversion of channel 3 is completed ADINT3 0: A/D conversion of channel 3 is not completed 1: A/D conversion of channel 3 is completed This bit is cleared by writing "1" to it. Writing "0" is ignored.

7.23.3.3 A/D Interrupt Enable Control Register: 0x40070008

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												ADINTEN3	ADINTEN2	ADINTEN1	ADINTEN0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

ADINTEN	Indicates whether the interrupt for each channel is enabled. ADINTEN0 0: Interrupt for channel 0 disabled 1: Interrupt for channel 0 enabled ADINTEN1 0: Interrupt for channel 1 disabled 1: Interrupt for channel 1 enabled ADINTEN2 0: Interrupt for channel 2 disabled 1: Interrupt for channel 2 enabled ADINTEN3 0: Interrupt for channel 3 disabled 1: Interrupt for channel 3 enabled [Note] After converting a channel that is set to the interrupt disabled state, if the channel is changed to the interrupt enabled state without clearing the conversion status, an interrupt request is generated for the channel immediately after the change.

7.23.3.4 A/D Result Register 0: 0x4007000C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																		ADRST09	ADRST08	ADRST07	ADRST06	ADRST05	ADRST04	ADRST03	ADRST02	ADRST01	ADRST00	Reserved			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

ADRST0	This register stores the A/D conversion result (average value) of channel 0. If the number of conversion times of the A/D converter (one, four, eight, or sixteen times) is specified, the average value of the specified number of conversions is stored. After a set of A/D conversions, even if interrupt request is not enabled until the next A/D conversion is completed, the value of the next (latest) A/D conversion result is stored in this register (value is overwritten and updated with the latest A/D conversion value). [Note] (※1) Reserved bit for future extension. "0" is read when reading. Write "0" when writing. If the A/D control register is written during A/D conversion (ADRUN = 1), no A/D conversion result for ADRST0 to ADRST3 is guaranteed.

7.23.3.5 A/D Result Register 1: 0x40070010

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																		ADRST19	ADRST18	ADRST17	ADRST16	ADRST15	ADRST14	ADRST13	ADRST12	ADRST11	ADRST10	Reserved			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

ADRST1	This register stores the A/D conversion result (average value) of channel 1. If the number of conversion times of the A/D converter (one, four, eight, or sixteen times) is specified, the average value of the specified number of conversions is stored. After a set of A/D conversions, even if interrupt request is not enabled until the next A/D conversion is completed, the value of the next (latest) A/D conversion result is stored in this register (value is overwritten and updated with the latest A/D conversion value). [Note] (※1) Reserved bit for future extension. "0" is read when reading. Write "0" when writing. If the A/D control register is written during A/D conversion (ADRUN = 1), no A/D conversion result for ADRST0 to ADRST3 is guaranteed.

7.23.3.6 A/D Result Register 2: 0x40070014

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																		ADRST29	ADRST28	ADRST27	ADRST26	ADRST25	ADRST24	ADRST23	ADRST22	ADRST21	ADRST20	Reserved			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

ADRST2	This register stores the A/D conversion result (average value) of channel 2. If the number of conversion times of the A/D converter (one, four, eight, or sixteen times) is specified, the average value of the specified number of conversions is stored. After a set of A/D conversions, even if interrupt request is not enabled until the next A/D conversion is completed, the value of the next (latest) A/D conversion result is stored in this register (value is overwritten and updated with the latest A/D conversion value). [Note] (※1) Reserved bit for future extension. "0" is read when reading. Write "0" when writing. If the A/D control register is written during A/D conversion (ADRUN = 1), no A/D conversion result for ADRST0 to ADRST3 is guaranteed.

7.23.3.7 A/D Result Register 3: 0x40070018

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																		ADRST39	ADRST38	ADRST37	ADRST36	ADRST35	ADRST34	ADRST33	ADRST32	ADRST31	ADRST30	Reserved			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

ADRST3	This register stores the A/D conversion result (average value) of channel 3. If the number of conversion times of the A/D converter (one, four, eight, or sixteen times) is specified, the average value of the specified number of conversions is stored. After a set of A/D conversions, even if interrupt request is not enabled until the next A/D conversion is completed, the value of the next (latest) A/D conversion result is stored in this register (value is overwritten and updated with the latest A/D conversion value). [Note] (※1) Reserved bit for future extension. "0" is read when reading. Write "0" when writing. If the A/D control register is written during A/D conversion (ADRUN = 1), no A/D conversion result for ADRST0 to ADRST3 is guaranteed.

7.23.3.8 A/D Data Register 0: 0x4007001C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDT_STAT0	Reserved																					ADDT09	ADDT08	ADDT07	ADDT06	ADDT05	ADDT04	ADDT03	ADDT02	ADDT01	ADDT00
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

ADDT0	This register stores the A/D conversion result at low temperature (-40 ℃), which is stored in Flash.
ADDT_STAT0	This register indicates whether ADDT0 was updated from Flash. 0: Updated 1: Not updated

7.23.3.9 A/D Data Register 1: 0x40070020

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDT_STAT1	Reserved																					ADDT19	ADDT18	ADDT17	ADDT16	ADDT15	ADDT14	ADDT13	ADDT12	ADDT11	ADDT10
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

ADDT1	This register stores the A/D conversion result at ordinary temperature (25 ℃), which is stored in Flash.
ADDT_STAT1	This register indicates whether ADDT1 was updated from Flash. 0: Updated 1: Not updated

7.23.3.10 A/D Data Register 2: 0x40070024

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	ADDT_STAT0	Reserved																					ADDT29	ADDT28	ADDT27	ADDT26	ADDT25	ADDT24	ADDT23	ADDT22	ADDT21	ADDT20
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	0	1	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

ADDT2	This register stores the A/D conversion result at high temperature (105 ℃), which is stored in Flash.
ADDT_STAT2	This register indicates whether ADDT2 was updated from Flash. 0: Updated 1: Not updated

7.23.3.11 A/D Data Register 3: 0x40070028

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDT_STAT3	Reserved																					ADDT39	ADDT38	ADDT37	ADDT36	ADDT35	ADDT34	ADDT33	ADDT32	ADDT31	ADDT30
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

ADDT3	This register stores the A/D conversion result of CH3 when VDDIO_CPU = 3.3 V, which is stored in Flash.
ADDT_STAT3	This register indicates whether ADDT3 was updated from Flash. 0: Updated 1: Not updated

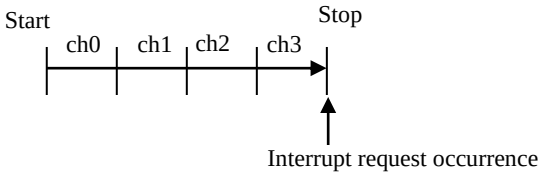
7.23.4 Functional Description

The A/D converter has two types of operation mode. The scan mode converts multiple selected channels sequentially, and the selection mode converts one selected channel.
These modes cannot be used at the same time. The operation mode should be changed when the A/D conversion operation is stopped.
If the operation mode is changed during A/D conversion (ADRUN = 1), no A/D conversion result is guaranteed.

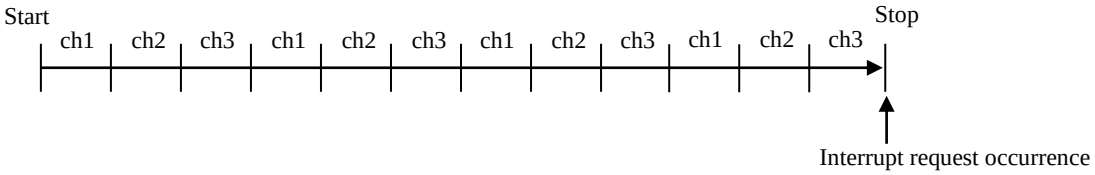
7.23.4.1 Scan Mode (When Multiple Channels Are Selected)

In scan mode (when multiple channels are selected), A/D conversion is performed for the channels selected from CH0 to CH3 in ascending order, starting at the channel with the smallest channel number.
In scan mode, A/D conversion is stopped when the A/D conversion of the last channel is completed after one cycle of the selected channels. It is also possible to automatically restart A/D conversion from the beginning channel and select the number of automatic restarts.

Operation examples of scan mode are shown below.



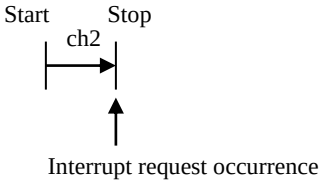
Operation example of scan mode (Scan channel: CH0 to CH3.)



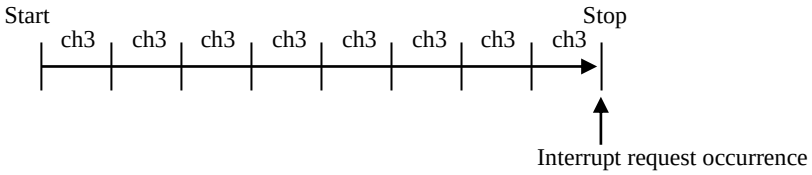
Operation example of scan mode (Scan channel: CH1 to CH3. Number of A/D conversion times = 4)

7.23.4.2 Selection Mode (When Only One Channel Is Selected)

In selection mode (when only one channel is selected), A/D conversion is performed for one channel selected from CH0 to CH3. Like scan mode, it is possible to automatically restart A/D conversion for the selected channel and select the number of automatic restarts.
Operation examples of selection mode are shown below.



Operation example of selection mode (Selected channel: CH2)



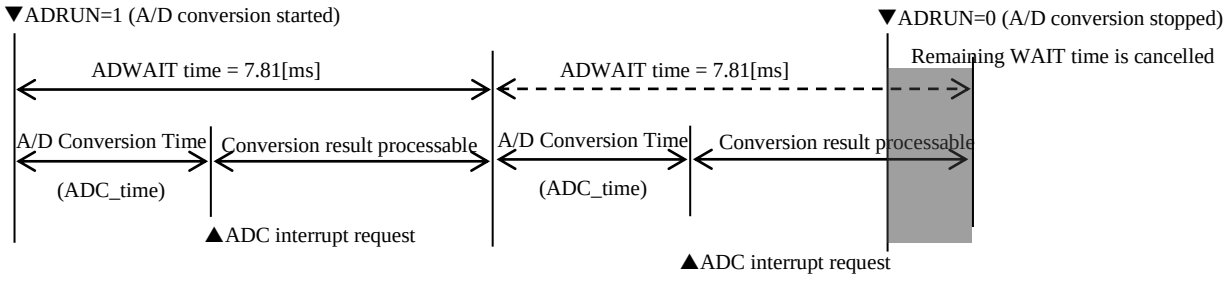
Operation example of selection mode (Selected channel: CH3. Number of A/D conversion times = 8)

7.23.4.3 A/D Conversion Time

The processing time taken for A/D conversion (ADC_time) is as follows:
ADC_time = (1/(2.5[MHz]/(2^ADCYC[1:0])) × 25[cycle] × ADSSM[3:0]) x ADCHG[1:0]

[Example]
ADCYC[1:0]=0x2 (Operation clock: 4 dividing)
ADSSM[3:0]=0x5 (CH used: CH0, CH2)
ADCHG[1:0]=0x2 (Conversions: 8 times)
ADWAIT[3:0]=0x0 (WAIT time from conversion start to output detection: 7.81 [ms])
Operation at ADAST=1 setting (when A/D conversion stops at second interrupt processing)

ADC_time = (1/2.5 MHz/4) × 25 × 2 × 8 = 640[us]



7.23.4.4 A/D Conversion Procedure

The follow procedure is used for calculating the input voltage.

(1) The current power supply voltage (Vddx) is calculated from the CH3 reference voltage A/D conversion result at power supply 3.3 V (ADDT3) and the current reference voltage A/D conversion result (ADRST3).
Vddx=3.3 x (ADDT3/ADRST3)

(2) The absolute value of the input voltage is calculated based on the CH0 to CH2 input voltage A/D conversion result (ADRST*) and the power supply voltage (Vddx) calculated in (1).
Vx=Vddx x (ADRST*/1023)
* Using a full scale code 1023 including quantization error for code calculation.

7.24. Thermometer (TEMP)

7.24.1 General Description

1-channel temperature sensor to measure the temperature in the chip (function to obtain the temperature information).
The features are shown below.

- Accuracy: ± 5 °C
- Converts the temperature to voltage and digitizes the converted voltage by using the 10-bit A/D converter.

* A/D conversion of thermometer conforms to the functions of ADC. A/D conversion of thermometer functions by selecting and operating channel 2 of ADC. For details, refer to the chapter about ADC.

7.24.2 List of Registers

Refer to the chapter about ADC.

7.24.3 Description of Registers

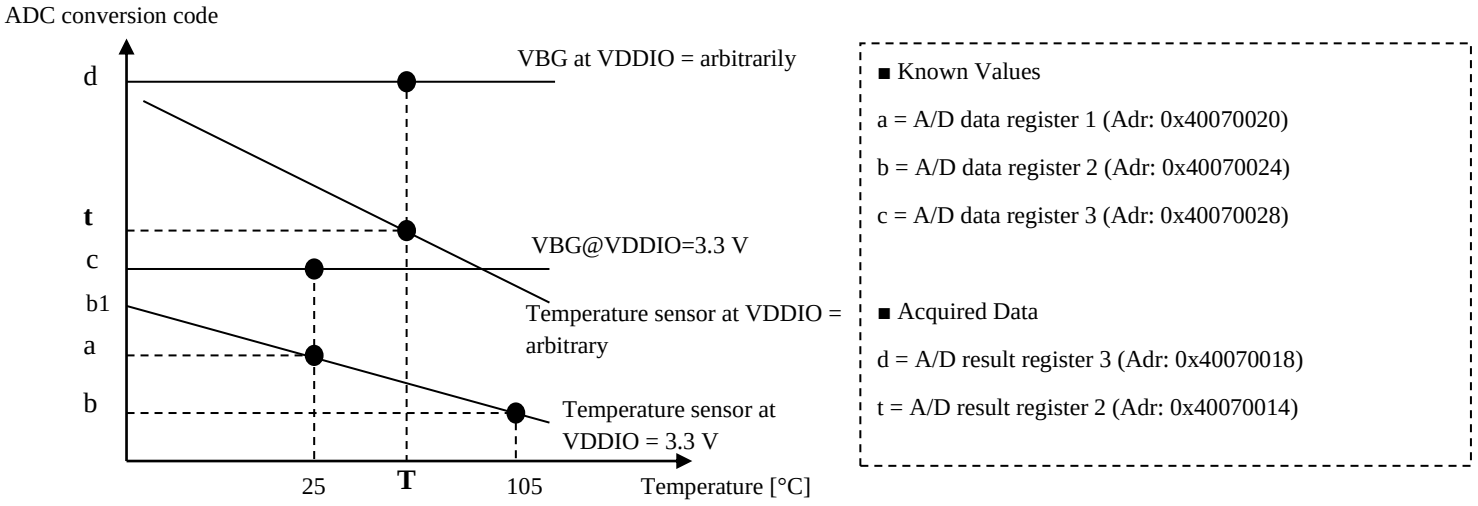
Refer to the chapter about ADC.

7.24.4 Functional Description

The ADC conversion codes at ordinary temperature (25 °C/VDDIO = 3.3 V), high temperature (105 °C/VDDIO = 3.3 V), and VBG (25 °C/VDDIO = 3.3 V) (named as a, b, and c, respectively) are written in the internal Flash in advance at shipment of LSI. These values stored in Flash are reflected to the A/D data register 1, A/D data register 2, and A/D data register 3 by the automatic load function at power-on.

Suppose we name the ADC conversion codes of arbitrary input (CH2) and VBG (CH3) at VDDIO = arbitrary voltage as t and d, respectively.
The temperature (T) at arbitrary VDDIO voltage can be calculated from these values a to d and t by using the calculation formula described below.
The calculation should be processed by customer software.

■ Temperature and ADC Conversion Code (Register Value)



■ Temperature Calculation Formula

$$a1 = \frac{b - a}{105 - 25}$$

$$\text{Temperature } \mathbf{T} = \frac{\frac{c}{d} * \mathbf{t} - b1}{a1}$$

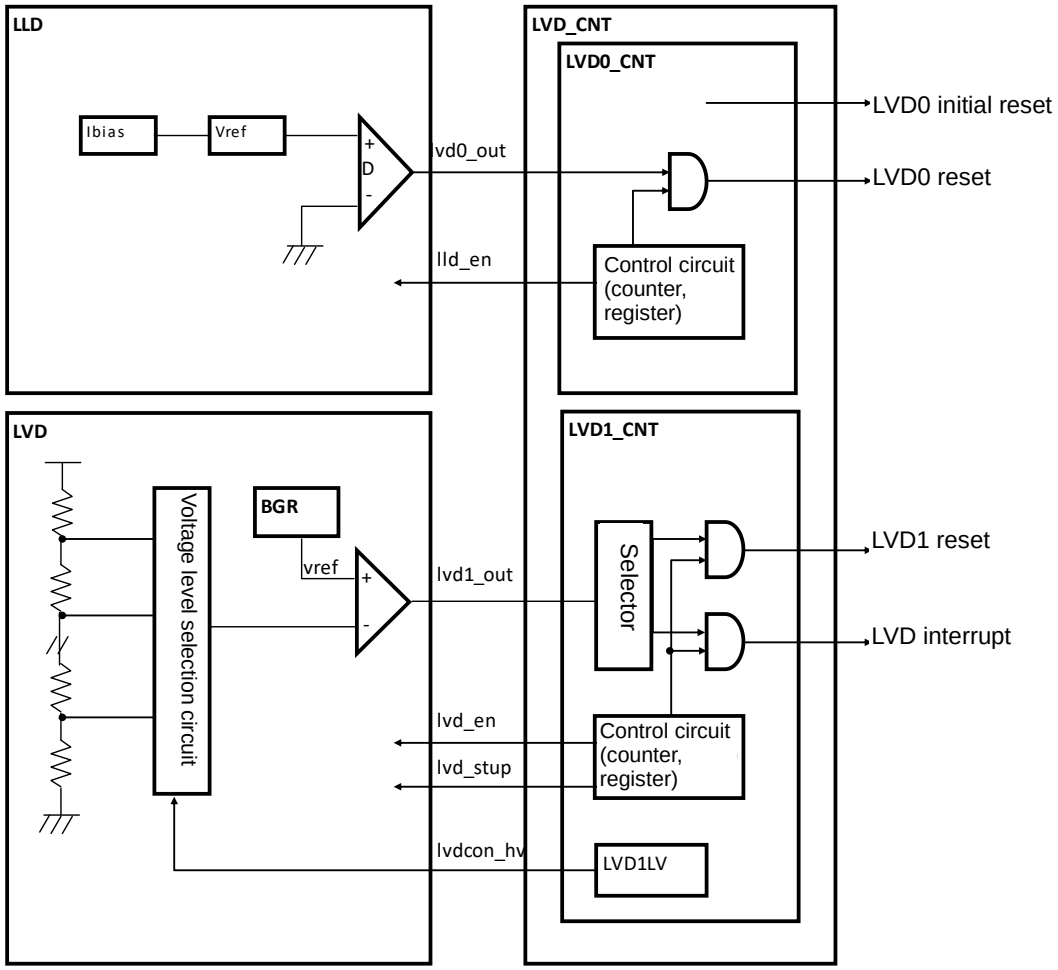
7.25. Low Voltage Detection (LVD)

7.25.1 General Description

Low voltage detection function.

The features are shown below.

- The voltage detection level can be set.
- After detection, interrupt notification or reset can be selected.
- Starts the reference voltage (VBG) periodically by using the dedicated low speed timer to compare and determine the voltage detection level.



7.25.2 List of Registers

Address [H]	Name	Symbol	R/W	Initial value [H]	Description
0x40090000	LVD0 control register	LVD0CON	R	0x0000_0000	Notify whether or not to detect LLD (LVD0).
0x40090004	LVD0 analog output stabilization wait time setting register	LVD0W	R/W	0x8000_0004	Sets the LLD (LVD0) stabilization wait time.
0x40090008	LVD0 enable register	ENLVD0	R/W	0x0000_0001	Sets whether to enable/disable the operation of LLD (LVD0).
0x4009000C	LVD1 control register	LVD1CON	R/W	0x0000_0001	Notify whether or not to detect LVD (LVD1).
0x40090010	LVD1 operation cycle setting register	LVD1P	R/W	0x0000_0000	Sets the operation mode of LVD (LVD1).
0x40090014	LVD1 analog output stabilization wait time setting register	LVD1W	R/W	0x0000_0002	Sets the LVD (LVD1) stabilization wait time.
0x40090018	LVD1 startup signal rise/fall setting register	LVD1STUP	R/W	0x0000_0100	Adjusts the timing of LVD (LVD1).
0x4009001C	LVD1 interrupt mask register	LVD1IMSK	R/W	0x0000_0000	Sets the interrupt mask of LVD (LVD1).
0x40090020	LVD1 interrupt status register	LVD1IST	R	0x0000_0000	Indicates the interrupt status (after masking) of LVD (LVD1).
0x40090024	LVD1 interrupt RAW status register	LVD1RIST	R	0x0000_0000	Indicates the interrupt status (before masking) of LVD (LVD1).
0x40090028	LVD1 interrupt clear register	LVD1ICLR	R	0x0000_0000	Clears the interrupt from LVD (LVD1).
0x4009002C	LVD1 enable register	ENLVD1	R/W	0x0000_0000	Sets whether to enable/disable the operation of LVD (LVD1).

7.25.3 Description of Registers

7.25.3.1 LVD0 Control Register: 0x40090000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved					LVD0RF	Reserved																									
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

LVD0RF	LVD0 voltage level detection enable flag. 1: Enabled 0: Disabled (waiting stabilization or OFF) This bit changes to "1" when the wait time set in LVD0W has been elapsed after setting ENLVD0 to "1". Be sure to check that this bit is "1" before stopping the clock supply to this block during LVD0 operation.

7.25.3.2 LVD0 analog output stabilization wait time setting register: 0x40090004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												LVD0W			
Initial value	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

LVD0W	Sets the LVD0 analog output stabilization wait time. (default 52us) Wait time = Setting value × APB clock cycle × 256 sec A value of up to about 100 µs can be set. Setting "0" is prohibited.

* Writing to this register address is only enabled when ENLVD0 = "0". It is ignored when ENLVD0 = "1".

* This register is reset only by the power-on reset.

7.25.3.3 LVD0 enable register: 0x40090008

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															ENLVD0
Initial value	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

ENLVD0	Enables the voltage detection 0 function. 1: Operate 0: Stop

* This register is reset only by the power-on reset.

7.25.3.4 LVD1 control register: 0x4009000C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved					LVD1RF	LVD1BF	LVD1F	Reserved							LVD1SMP	LVD1SLP	Reserved		LVD1MD		Reserved		LVD1SEL		Reserved				LVD1LV			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	R	R	RW	RW	R	R	RW	RW	R	R	R	R	RW	RW	RW	RW

LVD1RF	voltage level detection enable flag. 1: Enabled 0: Disabled (waiting stabilization or OFF)
LVD1BF	Voltage level detection BUSY flag. 1: Operating 0: Stopped (waiting operation cycle or OFF)
LVD1F	Voltage level detection flag 1: Detected 0: Not detected
LVD1SMP	Sets whether or not to sample the analog output. 1: Samples it with LVD clock. It is judged as detected that low voltage is detected continuously in two cycles. 0: Does not sample it. In intermittent operation mode, it is sampled regardless of the setting of this bit.
LVD1SLP	Sets the operation during sleep. 1: Continues the operation during sleep. 0: Automatically stops the operation at sleep and automatically restarts it after return from sleep.
LVD1MD	Sets the operation mode. 00: Continuous operation (default)

<table>
<tr><td></td><td>01: Intermittent operation
10: Single operation
11: Continuous operation (setting prohibited)</td></tr>
<tr><td>LVD1SEL</td><td>Sets whether to enable or disable interrupt/reset when it is detected that VDD is lower than the threshold voltage.
00: Reset disabled, interrupt disabled (default)
01: Reset enabled, interrupt disabled
10: Reset disabled, interrupt enabled
11: Reset enabled, interrupt disabled (setting prohibited)</td></tr>
<tr><td>LVD1LV</td><td>Sets the threshold voltage of LVD1. Set only the values listed below.
<table>
<tr><th>bit3</th><th>bit2</th><th>bit1</th><th>bit0</th><th>LVD1 threshold voltage [V]</th><th></th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1.80</td><td>Initial value</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1.95</td><td></td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>2.10</td><td></td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>2.25</td><td></td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>2.40</td><td></td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>2.55</td><td></td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>2.70</td><td></td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>2.85</td><td></td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>3.00</td><td></td></tr>
</table></td></tr>
</table>

* Writing to this register address is only enabled when ENLVD1 = "0". It is ignored when ENLVD1 = "1".

7.25.3.5 LVD1 operation cycle setting register: 0x40090010

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved												LVD1P																			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

LVD1P	Sets the operation cycle of LVD1. This register is only enabled in intermittent operation mode. Cycle = Setting value * Clock cycle. Up to 32 seconds at 32 kHz can be set. (512 seconds at 2 kHz) Setting "0" is prohibited.

* Writing to this register address is only enabled when ENLVD1 = "0". It is ignored when ENLVD1 = "1".

7.25.3.6 LVD1 analog output stabilization wait time setting register: 0x40090014

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																										LVD1W					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW

LVD1W	Sets the LVD1 analog output stabilization wait time. Wait time = Setting value * Cycle. 1.92 ms at frequency = 32 kHz. Setting "0" is prohibited.

* Writing to this register address is only enabled when ENLVD1 = "0". It is ignored when ENLVD1 = "1".

7.25.3.7 LVD1 startup signal rise/fall setting register: 0x40090018

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																		LVD1STUPF						Reserved		LVD1STUPR					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	R	R	RW	RW	RW	RW	RW	RW

LVD1STUPF	Sets the fall timing of LVD1 startup signal. When LVD1 startup signal width = LVD1STUPW, Setting value = LVD1STUPR + LVD1STUPW. 1.92 ms at frequency = 32 kHz. Setting "0" or LVD1STUPF > LVD1W is prohibited.
LVD1STUPR	Sets the delay time from start to startup of the LVD1 comparator. Delay time = Setting value * Cycle. 1.92 ms at frequency = 32 kHz. Set a value in the range of LVD1STUPR < LVD1W and LVD1STUPR < LVD1STUPF. When setting "0", the LVD1 comparator starts at the same time as enabled. The LVD1 startup signal is fixed to L when LVD1STUPR ≥ LVD1W. It is also fixed to L when LVD1STUPR ≥ LVD1STUPF.

* Writing to this register address is only enabled when ENLVD1 = "0". It is ignored when ENLVD1 = "1".

7.25.3.8 LVD1 interrupt mask register: 0x4009001C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															LVD1IMSK
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

LVD1IMSK	LVD1 interrupt mask 1: Mask 0: No mask

7.25.3.9 LVD1 interrupt status register: 0x40090020

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															LVD1IST
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

LVD1IST	LVD1 interrupt status (after masking) 1: Interrupt 0: No interrupt

7.25.3.10 LVD1 interrupt RAW status register: 0x40090024

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															LVD1RIST
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

LVD1RIST	LVD1 interrupt status (before masking) 1: Interrupt 0: No interrupt

7.25.3.11 LVD1 interrupt clear register: 0x40090028

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															LVD1ICLR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

LVD1ICLR	When this register is read, the LVD1 interrupt is cleared.

7.25.3.12 LVD1 enable register: 0x4009002C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															ENLVD1
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

ENLVD1	Enables the voltage detection function. 1: Operate 0: Stop

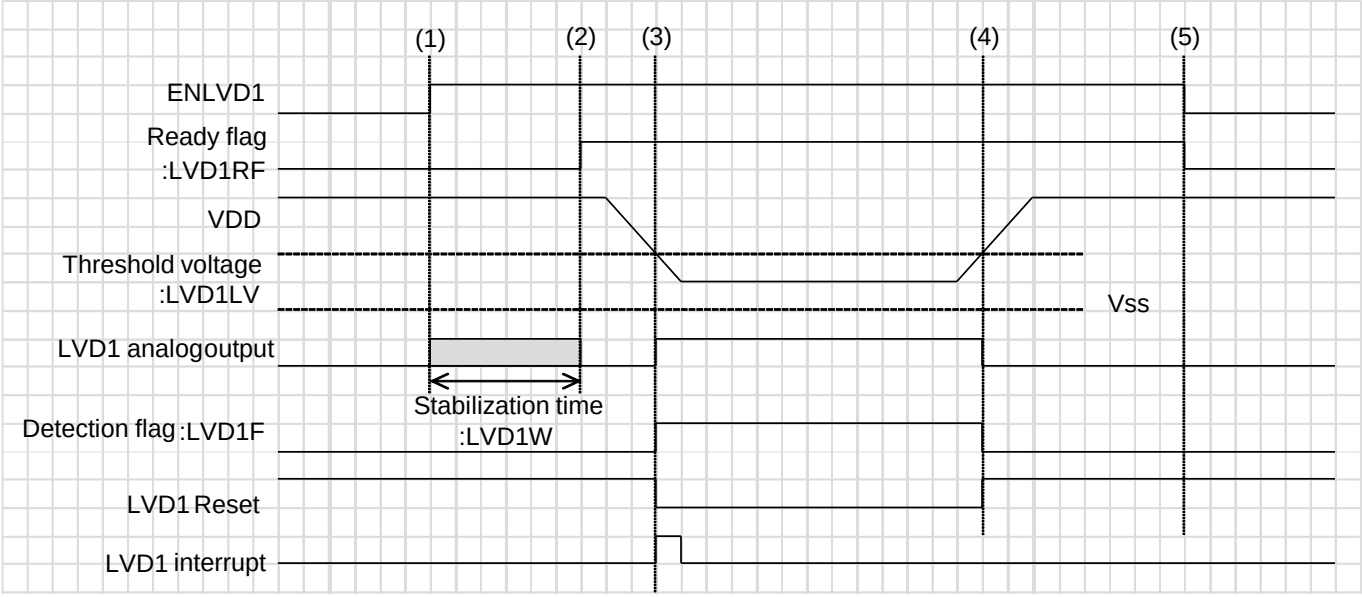
7.25.4 Functional Description

This function can be used to determine whether the power supply voltage is lower than the specified threshold voltage. The functions are shown below.

	LVD0(LLD)	LVD1(LVD)
At low voltage detection	Reset	Reset or interrupt can be selected.
Threshold voltage	Fixed value	Can be selected from 16 values
Operation Mode	Continuous operation	Can be selected from continuous operation mode, intermittent operation mode, and single operation mode.
Reset range at low voltage detection	All functions are reset except CPU_ST (same as when an interrupt is detected by WDT). The RF chip is also reset. However, this function (LVD) block itself is excluded from the reset.	
Operation during sleep	Can operate without clock	Continuous operation mode: Can operate without clock Intermittent operation mode: Can operate with only low-speed clock Single operation mode: Cannot operate

7.25.4.1 Continuous Operation Mode

In the continuous operation mode, the judgments is continuously made. A decrease of VDD can be detected fastest.



(1) ENLVD1 is set to "1" from the CPU.

(2) "1" is displayed on LVD1RF after the LVD1 analog output is stabilized (The time is set in LVD1W).

(3) When VDD becomes lower than the threshold voltage, "1" is displayed on the detection flag (VLSF).
When the reset is enabled, a LVD1 reset is generated.
When the interrupt is enabled, a LVD1 interrupt is generated. The interrupt is cleared by the CPU.

(4) When VDD becomes higher than the threshold voltage (LVD1LV), "0" is displayed on the detection flag (LVD1F).
The reset is released.

(5) ENLVD1 is set to "0" from the CPU to finish the detection.

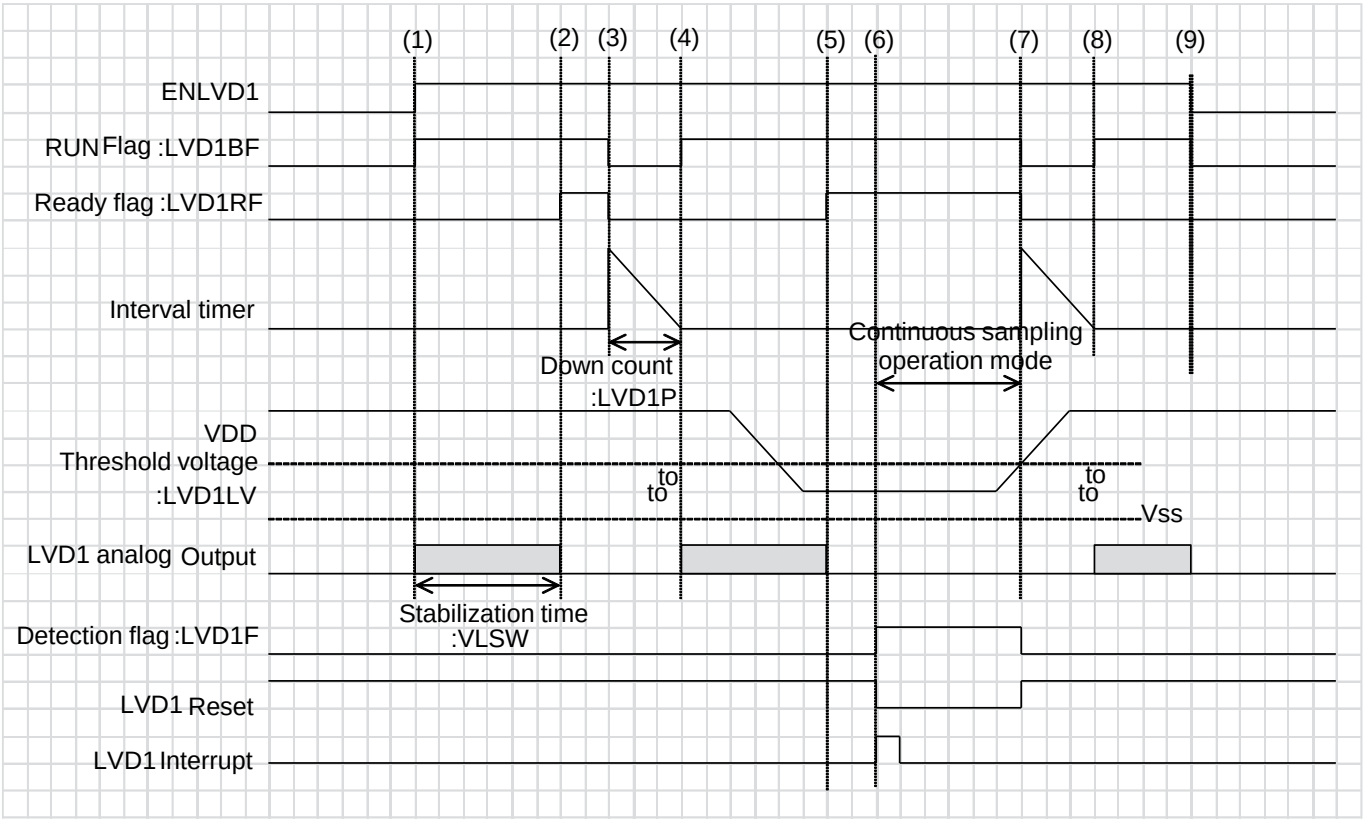
7.25.4.2 Intermittent Operation Mode

In the intermittent operation mode, the LVD1 judgment is periodically made by the hardware.

The hardware automatically performs start, judgment, and stop even when the CPU is in the sleep state, reducing the average current.

It automatically switches to the continuous sampling operation mode when it detects a low voltage.

When VDD exceeds the threshold voltage in the continuous sampling operation mode, it returns to the intermittent operation mode.



(1) ENLVD1 is set to "1" from the CPU. "1" is displayed on the LVD1BF flag.

(2) "1" is displayed on LVD1RF after the LVD1 analog output is stabilized (The time is set in LVD1W).

(3) After 32 kHz 2□ period, LVD1 is disabled, and LVD1BF and LVD1RF are cleared.
The operation cycle timer is set to LVD1P, and the down count starts.

(4) When the operation cycle timer becomes "0", LVD1 is enabled, and "1" is displayed on LVD1BF.

(5) "1" is displayed on LVD1RF after the LVD1 analog output is stabilized (The time is set in LVD1W).

(6) "1" is displayed on LVD1RF after the LVD1 analog output is stabilized (The time is set in LVD1W).
When VDD is lower than the threshold voltage, "1" is displayed on LVD1F.
When the reset is enabled, a LVD1 reset is generated.
When the interrupt is enabled, a LVD1 interrupt is generated. The interrupt is cleared by the CPU.
The continuous sampling operation mode is automatically turned on.

(7) “1” is displayed on LVD1RF after the LVD1 analog output is stabilized (The time is set in LVD1W).
When VDD becomes higher than the threshold voltage, “0” is displayed on the detection flag (LVD1F).
The reset is released.

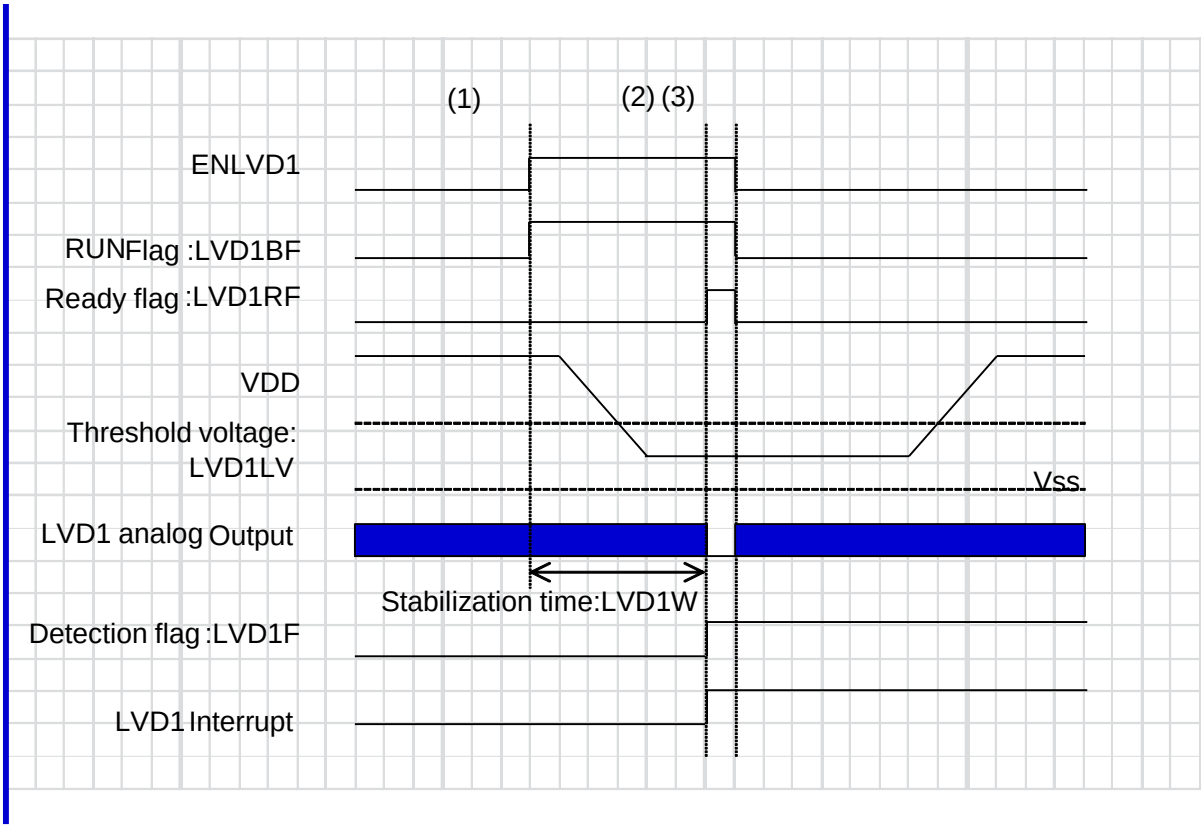
(8) Same as (2)

(9) ENLVD1 is set to "0" from the CPU to finish the detection.

7.25.4.3 Single Operation Mode

In the single operation mode, an LVD1 interrupt is notified when the comparison of voltage is completed (Note 1), and the hardware automatically disables the low voltage detection operation.

It is suitable for a case when the timing of the low voltage detection is controlled by software.



(1) ENLVD1 is set to "1" from the CPU. "1" is displayed on the VLSBF flag.

(2) "1" is displayed on LVD1RF after the LVD analog output is stabilized.
An LVD1 interrupt is notified (implying completion of judgment).
When VDD is less than the threshold voltage (LVD1LV), "1" is displayed on LVD1F.
* When the reset is enabled by LVD1SEL, a reset is generated.

(3) After 32 kHz 1□, ENLVD1 is automatically cleared.

Note 1) In the other modes (continuous and intermittent operation modes), the interrupt is notified when a low voltage is detected (VDD < threshold voltage).

7.26.DMAC

7.26.1 General Description

Direct memory access controller. Among peripherals, SPI and AES support DMA transfer.

The features are shown below.

- Four-channel (channel 0 to 3) DMA controller.
- Each channel includes a 16-stage FIFO (8-stage for channels 0 and 1) for source transfer and destination transfer.
- Supports the peripheral-to-memory transfer.
- Includes the hardware interface to handshake with SPI and AES.
- Supports up to 2048-byte block transfer.
- Supports the channel priority setting.
- Has one AHB master port.
- Supports increment/decrement of the transfer address and transfer to a single address.
- Supports multiple block transfer using a linked list.

7.26.2 List of Registers

Address	Size	Name	Symbol	32-bit symbol[8]	R/W	Description
BASE+0x000	64	Channel 0 source address register	DMACm_SAR0	H:- L:DMACm_SAR0	R/W	Sets the address of the transfer source of channel 0.
BASE+0x008	64	Channel 0 destination address register	DMACm_DAR0	H:- L:DMACm_DAR0	R/W	Sets the address of the transfer destination of channel 0.
BASE+0x010	64	Channel 0 linked list pointer register	DMACm_LLP0	H:- L:DMACm_LLP0	R/W	Sets the pointer to the linked list of channel 0.
BASE+0x018	64	Channel 0 control register	DMACm_CTL0	H:DMACm_CTL0_H L:DMACm_CTL0_L	R/W	Controls channel 0.
BASE+0x058	64	Channel 1 source address register	DMACm_SAR1	H:- L:DMACm_SAR1	R/W	Sets the address of the transfer source of channel 0.
BASE+0x060	64	Channel 1 destination address register	DMACm_DAR1	H:- L:DMACm_DAR1	R/W	Sets the address of the transfer destination of channel 1.
BASE+0x068	64	Channel 1 linked list pointer register	DMACm_LLP1	H:- L:DMACm_LLP1	R/W	Sets the pointer to the linked list of channel 1.
BASE+0x070	64	Channel 1 control register	DMACm_CTL1	H:DMACm_CTL1_H L:DMACm_CTL1_L	R/W	Controls channel 1.
BASE+0x098	64	Channel 1 configuration register	DMACm_CFG1	H:DMACm_CFG1_H L:DMACm_CFG1_L	R/W	Sets channel 1.
BASE+0x0B0	64	Channel 2 source address register	DMACm_SAR2	H:- L:DMACm_SAR2	R/W	Sets the address of the transfer source of channel 2.
BASE+0x0B8	64	Channel 2 destination address register	DMACm_DAR2	H:- L:DMACm_DAR2	R/W	Sets the address of the transfer destination of channel 2.
BASE+0x0C0	64	Channel 2 linked list pointer register	DMACm_LLP2	H:- L:DMACm_LLP2	R/W	Sets the pointer to the linked list of channel 2.

[8] Use the 32-bit symbols shown in this column to access the registers from software. "H:" indicates the higher 32 bits, and "L:" the lower 32 bits. When all the higher 32 bits are reserved fields and not used, use a 64-bit symbol to access the lower 32 bits.

BASE+0x0C8	64	Channel 2 control register	DMACm_CTL2	H:DMACm_CTL2_H L:DMACm_CTL2_L	R/W	Controls channel 2.
BASE+0x0F0	64	Channel 2 configuration register	DMACm_CFG2	H:DMACm_CFG2_H L:DMACm_CFG2_L	R/W	Sets channel 2.
BASE+0x108	64	Channel 3 source address register	DMACm_SAR3	H:- L:DMACm_SAR3	R/W	Sets the address of the transfer source of channel 3.
BASE+0x110	64	Channel 3 destination address register	DMACm_DAR3	H:- L:DMACm_DAR3	R/W	Sets the address of the transfer destination of channel 3.
BASE+0x118	64	Channel 3 linked list pointer register	DMACm_LLP3	H:- L:DMACm_LLP3	R/W	Sets the pointer to the linked list of channel 3.

BASE+0x120	64	Channel 3 control register	DMACm_CTL3	H:DMACm_CTL3_H L:DMACm_CTL3_L	R/W	Controls channel 3.
BASE+0x148	64	Channel 3 configuration register	DMACm_CFG3	H:DMACm_CFG3_H L:DMACm_CFG3_L	R/W	Sets channel 3.
BASE+0x2C0	32	Tfr interrupt RAW status register	DMACm_RAW_TFR	Same as the left	R	Indicates the RAW status of Tfr interrupt.
BASE+0x2C8	32	Block interrupt RAW status register	DMACm_RAW_BLOCK	Same as the left	R	Indicates the RAW status of Block interrupt.
BASE+0x2D0	32	SrcTran interrupt RAW status register	DMACm_RAW_SRC_TRAN	Same as the left	R	Indicates the RAW status of SrcTran interrupt.
BASE+0x2D8	32	DstTran interrupt RAW status register	DMACm_RAW_DST_TRAN	Same as the left	R	Indicates the RAW status of DstTran interrupt.
BASE+0x2E0	32	ERR interrupt status register	DMACm_RAW_ERR	Same as the left	R	Indicates the RAW status of Err interrupt.
BASE+0x2E8	32	Tfr interrupt status register	DMACm_STAT_TFR	Same as the left	R	Indicates the status of Tfr interrupt.
BASE+0x2F0	32	Block interrupt status register	DMACm_STAT_BLOCK	Same as the left	R	Indicates the status of Block interrupt.
BASE+0x2F8	32	SrcTran interrupt status register	DMACm_STAT_SRC_TRAN	Same as the left	R	Indicates the status of SrcTran interrupt.
BASE+0x300	32	DstTran interrupt status register	DMACm_STAT_DST_TRAN	Same as the left	R	Indicates the status of DstTran interrupt.

BASE+0x308	32	ERR interrupt status register	DMACm_STAT_ERR	Same as the left	R	Indicates the status of Err interrupt.
BASE+0x310	32	Tfr interrupt mask register	DMACm_MASK_TFR	Same as the left	R/W	Masks the Tfr interrupt.
BASE+0x318	32	Block interrupt mask register	DMACm_MASK_BLOCK	Same as the left	R/W	Masks the Block interrupt.
BASE+0x320	32	SrcTran interrupt mask register	DMACm_MASK_SRC_TRAN	Same as the left	R/W	Masks the SrcTran interrupt.
BASE+0x328	32	DstTran interrupt mask register	DMACm_MASK_DST_TRAN	Same as the left	R/W	Masks the DstTran interrupt.
BASE+0x330	32	ERR interrupt mask register	DMACm_MASK_ERR	Same as the left	R/W	Masks the Err interrupt.
BASE+0x338	32	Tfr interrupt clear register	DMACm_CLEAR_TFR	Same as the left	W	Clears the Tfr interrupt.
BASE+0x340	32	Block interrupt clear register	DMACm_CLEAR_BLOCK	Same as the left	W	Clears the Block interrupt.
BASE+0x348	32	SrcTran interrupt clear register	DMACm_CLEAR_SRC_TRAN	Same as the left	W	Clears the SrcTran interrupt.
BASE+0x350	32	DstTran interrupt clear register	DMACm_CLEAR_DST_TRAN	Same as the left	W	Clears the DstTran interrupt.
BASE+0x358	32	ERR interrupt clear register	DMACm_CLEAR_ERR	Same as the left	W	Clears the Err interrupt.
BASE+0x360	32	Interrupt status register	DMACm_STAT_INT	Same as the left	R	Indicates the status for each interrupt type.
BASE+0x368	32	Source software transfer request register	DMACm_REQ_SRC	Same as the left	R/W	Generates a transfer request of transfer source by software.

BASE+0x370	32	Destination software transfer request register	DMACm_REQ_DST	Same as the left	R/W	Generates a transfer request of transfer destination by software.
BASE+0x378	32	Source software single transfer request register	DMACm_SGL_REQ_SRC	Same as the left	R/W	Generates a single transfer request of transfer source by software.
BASE+0x380	32	Destination software single transfer request register	DMACm_SGL_REQ_DST	Same as the left	R/W	Generates a single transfer request of transfer destination by software.
BASE+0x388	32	Source software last transfer register	DMACm_LST_SRC	Same as the left	R/W	Generates the last transfer request of transfer source by software.
BASE+0x390	32	Destination software last transfer register	DMACm_LST_DST	Same as the left	R/W	Generates the last transfer request of transfer destination by software.
BASE+0x398	32	DMA configuration register	DMACm_CFG	Same as the left	R/W	Makes the settings for DMA.
BASE+0x3A0	32	DMA channel enable register	DMACm_CH_EN	Same as the left	R/W	Enables the DMA channel.
BASE+0x3A8	32	DMAID register	DMACm_ID	Same as the left	R	Indicates the DMA ID.

* The BASE address is 0x40150000.

7.26.3 Description of Registers

7.26.3.1. Channel n Source Address Register (DMACm_SARn): (n = 0: BASE + 0x000, n = 1: BASE + 0x058, n = 2: BASE + 0x0B0, n = 3: BASE + 0x108)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAR																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Indicates the transfer source address during DMA transfer. The value is updated each time after transfer from the transfer source.

7.26.3.2. Channel n Destination Address Register (DMACm_DARn): (n = 0: BASE + 0x008, n = 1: BASE + 0x060, n = 2: BASE + 0x0B8, n = 3: BASE + 0x110)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DAR																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Indicates the transfer destination address during DMA transfer. The value is updated each time after transfer to the transfer destination.

7.26.3.3. Channel n Linked List Pointer Register (DMACm_LLPn): (n = 0: BASE + 0x010, n = 1: BASE + 0x068, n = 2: BASE + 0x0C0, n = 3: BASE + 0x118)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	LOC																														LMS	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

LOC: Indicates the start address on the memory when the next linked list item is valid. The start addresses are always aligned to 32-bit boundaries, and their lower 2 bits are always 0. This field indicates bits 31-2 of the start address.

LMS: Indicates the AHB master port.

7.26.3.4. Channel n Control Register (DMACm_CTLn): (n = 0: BASE + 0x018, n = 1: BASE + 0x070, n = 2: BASE + 0x0C8, n = 3: BASE + 0x120)

	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	Reservedn																			DONE	BLOCK_TS											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved			LLP_SRC_EN	LLP_DST_EN	SMS		DMS		TT_FC			Reserved		Reserved		SRC_MSIZE			DEST_MSIZE			SINC		DINC		SRC_TR_WIDTH			DST_TR_WIDTH			INT_EN
Initial value	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1
R / W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	RW	RW	R	RW	RW	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

DONE	Indicates that a block transfer is completed.
BLOCK_TS	Sets the size of block transfer. When using DMAC as the flow controller, this field should be set before enabling the channel. It is represented as the number of single transfers.
LLP_SRC_EN	When this field is set to 1 and the LOC field of DMACm_LLPx is set to a value other than 0, the block chaining for transfer from the source is enabled.
LLP_DST_EN	When this field is set to 1 and the LOC field of DMACm_LLPx is set to a value other than 0, the block chaining for transfer to the destination is enabled.
SMS	Sets the source master interface.
DMS	Sets the destination master interface.
TT_FC	Sets the transfer type and flow controller. For details, please refer to "3.20.7.26.4.1Flow Controller".

SRC_MSIZE	Sets the burst length of burst transfer from the source. 000: 1 001: 4 010: 8 011: 16 1xx: Unavailable.
DEST_MSIZE	Sets the burst length of burst transfer to the destination. 000: 1 001: 4 010: 8 011: 16 1xx: Unavailable.
SINC	Sets the shift of the source address. 00: The address is incremented each time transfer is completed. 01: The address is decremented each time transfer is completed. 1x: The address does not shift.
DINC	Sets the shift of the destination address. 00: The address is incremented each time transfer is completed. 01: The address is decremented each time transfer is completed. 1x: The address does not shift.
SRC_TR_WIDTH	Sets the source transfer width. 000: 8bit 001: 16bit 010: 32bit Other setting: Unavailable.
DST_TR_WIDTH	Sets the destination transfer width. 000: 8bit 001: 16bit 010: 32bit Other setting: Unavailable.
INT_EN	Enables interrupt. All interrupt sources of the corresponding channels are enabled.

7.26.3.5. Channel n Source Status Address Register (DMACm_SSTATARn): (n = 0: BASE + 0x030, n = 1: BASE + 0x088, n = 2: BASE + 0x0E0, n = 3: BASE + 0x138)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SSTATAR																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Sets the address of the status register of the transfer source peripheral.

7.26.3.6. Channel n Destination Status Address Register (DMACm_DSTATARn): (n = 0: BASE + 0x038, n = 1: BASE + 0x090, n = 2: BASE + 0x0E8, n = 3: BASE + 0x140)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DSTATAR																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Sets the address of the status register of the transfer destination peripheral.

7.26.3.7. Channel n Destination Status Address Register (DMACm_DSTATARn): (n = 0: BASE+0x038, n=1: BASE+0x090, n=2: BASE+0x0E8; n=3: BASE+0x140)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DSTATAR																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Sets the address of the status register of the transfer destination peripheral.

7.26.3.8. Channel n Configuration Register (DMACm_CFGn): (n = 0: BASE + 0x040, n = 1: BASE + 0x098, n = 2: BASE + 0x0F0, n = 3: BASE + 0x148)

	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	Reserved																			DEST_PER		Reserved		SRC_PER		Reserved		PROTCTL			FIFO_MODE	FCMODE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	RELOAD_DST	RELOAD_SRC	Reserved										SRC_HS_POL	DST_HS_POL	Reserved						HS_SEL_SRC	HS_SEL_DST	FIFO_EMPTY	CH_SUSP	CH_PRIOR			Reserved				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	RW	RW	RW	RW	R	R	R	R	R

* The read value of a reserved field is undefined. Do not expect that a fixed value is read when designing software.

DEST_PER	Sets the hardware handshake I/F of the destination. For interface numbers, refer to Table 7-26-2 Correspondence between DMA Transfer Request Sources and Interface Numbers.
SRC_PER	Sets the hardware handshake I/F of the source. For interface numbers, refer to Table 7-26-2 Correspondence between DMA Transfer Request Sources and Interface Numbers.
PROTCTL	Sets HPROT[3:1] at AHB master transfer. Normally the initial value should be used[9] .
FIFO_MODE	Sets the condition of FIFO for starting burst transfer. 0: Starts the burst transfer when there is free space/data for one AHB transfer. 1: Starts the burst transfer when there is free space/data for one half or more of the FIFO stages.
FCMODE	Sets the condition for starting source transfer when the destination peripheral is set as the flow controller. 0: Starts the source transfer without waiting for a transfer request from the destination. 1: Suspends the source transfer until a transfer request from the destination occurs.

[9] HPROT[3], HPROT[2], and HPROT[1] indicate cacheable, bufferable, and privilege access respectively.
UxPlatform has no slave device whose operation is changed by HPROT[3:2]. HPROT[1] from DMAC is fixed to H for the privilege access to be always used regardless of the PROTCTL[1] value.

RELOAD_DST	Automatically reloads the value of the destination address register with the value that was set when multiple block transfer is started, each time a block transfer is completed, in multiple block transfer.
RELOAD_SRC	Automatically reloads the value of the source address register with the value that was set when multiple block transfer is started, each time a block transfer is completed, in multiple block transfer.
SRC_HS_POL	Sets the polarity of hardware handshake I/F of the source.
DST_HS_POL	Sets the polarity of hardware handshake I/F of the destination.
HS_SEL_SRC	Sets the transfer request source of source. 0: A request from the hardware handshake I/F is accepted. A request from software is ignored. 1: A request from software is accepted. A request from the hardware handshake I/F is ignored.
HS_SEL_DST	Sets the transfer request source of destination. 0: A request from the hardware handshake I/F is accepted. A request from software is ignored. 1: A request from software is accepted. A request from the hardware handshake I/F is ignored.
FIFO_EMPTY	Indicates that the FIFO of the corresponding channel is empty.
CH_SUSP	While this bit is set, transfer from all sources of the corresponding channel is inhibited.
CH_PRIOR	Sets the priority of the corresponding channel. The highest is 7, and the lowest is 0.

7.26.3.9. Interrupt RAW Status Register (DMACm_RAW_TFR: BASE + 0x2C0, DMACm_RAW_BLOCK: BASE + 0x2C8, DMACm_RAW_SRC_TRAN: 0x2D0, DMACm_RAW_DST_TRAN: 0x2D8, DMAC_RAW_ERR: BASE + 0x2E0)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												CH3	CH2	CH1	CH0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Indicates the interrupt status before masking of each channel.

Each channel corresponds to the following interrupt source.

CH	Interrupt source
0	Data transfer from RAM to SPI2 completed
1	Data transfer from SPI2 to RAM completed
2	Data transfer from RAM to AES completed
3	Data transfer from AES to RAM completed

7.26.3.10. Interrupt Status Register (DMACm_STAT_TFR: BASE + 0x2E8, DMACm_STAT_BLOCK: BASE + 0x2F0, DMACm_STAT_SRC_TRAN: 0x2F8, DMACm_STAT_DST_TRAN: 0x300, DMAC_STAT_ERR: BASE + 0x308)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												CH3	CH2	CH1	CH0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Indicates the interrupt status after masking of each channel.

7.26.3.11. Interrupt Mask Register (DMACm_MASK_TFR: BASE + 0x310, DMACm_MASK_BLOCK: BASE + 0x318, DMACm_MASK_SRC_TRAN: 0x320, DMACm_MASK_DST_TRAN: 0x328, DMAC_MASK_ERR: BASE + 0x330)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																				INT_MASK_WE				Reserved				CH3	CH2	CH1	CH0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	R	R	R	R	RW	RW	RW	RW

Masks an interrupt of each channel.

This bit is masked when 0 is written to it.

To update the CHn bit, write 1 to the INT_MASK_WE[n] bit at the same time.

7.26.3.12. Interrupt Clear Register (DMACm_CLEAR_TFR: BASE + 0x338, DMACm_CLEAR_BLOCK: BASE + 0x340, DMACm_CLEAR_SRC_TRAN: 0x348, DMACm_CLEAR_DST_TRAN: 0x350, DMAC_CLEAR_ERR: BASE + 0x358)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												CH3	CH2	CH1	CH0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Clears an interrupt source by writing 1 to this register.

7.26.3.13. **Interrupt Source Status Register (DMACm_STAT_INT: BASE + 0x360)**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																											ERR	DSTT	SRCT	BLOCK	TFR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Clears an interrupt source by writing 1 to this register.

Each bit corresponds to the following interrupt source:

Bit	Name	Interrupt source
00	TFR	Transfer completion interrupt
01	BLOCK	Block transfer completion interrupt
02	SRCT	Source transfer completion interrupt
03	DSTT	Destination transfer completion interrupt
04	ERR	Error interrupt

7.26.3.14. Source Software Transfer Request Register (DMACm_REQ_SRC: BASE + 0x368)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																				SRC_REQ_WE[3:0]				Reserved				SRC_REQ[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	R	R	R	R	RW	RW	RW	RW

Generates a source transfer request by software.

To generate a transfer request of channel N, set and write 1 to SRC_REQ[N] and SRC_REQ_WE[N].

If SRC_REQ_WE[N] is 0, no transfer request is generated even when SRC_REQ[N] is 1.

7.26.3.15. Destination Software Transfer Request Register (DMACm_REQ_DST: BASE + 0x370)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																				DST_REQ_WE[3:0]				Reserved				DST_REQ[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	R	R	R	R	RW	RW	RW	RW

Generates a destination transfer request by software.

To generate a transfer request of channel N, set and write 1 to DST_REQ[N] and DST_REQ_WE[N].

If DST_REQ_WE[N] is 0, no transfer request is generated even when DST_REQ[N] is 1.

7.26.3.16. Source Software Single Transfer Request Register (DMACm_SGL_REQ_SRC: BASE + 0x378)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																				SRC_SGLREQ_WE[3:0]				Reserved				SRC_SGLREQ[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	R	R	R	R	RW	RW	RW	RW

Generates a source single transfer request by software.

To generate a transfer request of channel N, set and write 1 to SRC_SGLREQ[N] and SRC_SGLREQ_WE[N].

If SRC_SGLREQ_WE[N] is 0, no transfer request is generated even when SRC_SGLREQ[N] is 1.

7.26.3.17. Destination Software Single Transfer Request Register (DMACm_SGL_REQ_DST: BASE + 0x380)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																				DST_SGLREQ_WE[3:0]				Reserved				DST_SGLREQ[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	R	R	R	R	RW	RW	RW	RW

Generates a destination single transfer request by software.

To generate a transfer request of channel N, set and write 1 to DST_SGLREQ[N] and DST_SGLREQ_WE[N].

If DST_SGLREQ_WE[N] is 0, no transfer request is generated even when DST_SGLREQ[N] is 1.

7.26.3.18. Source Software Last Transfer Register (DMACm_LST_SRC: BASE + 0x388)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																				LSTSRC_WE[3:0]				Reserved				LSTSRC[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	R	R	R	R	RW	RW	RW	RW

Indicates that the transfer request by software is the last transfer.

To set the last transfer for channel N, set and write 1 to LSTSRC[N] and LSTSRC_WE[N].

If LSTSRC_WE[N] is 0, the last transfer is not set even when LSTSRC[N] is 1.

After setting the last transfer by this register, a transfer request becomes the last one when it is generated by the source software transfer request register.

7.26.3.19. Destination Software Last Transfer Register (DMACm_LST_DST: BASE + 0x390)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																				LSTDST_WE[3:0]				Reserved				LSTDST[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	R	R	R	R	RW	RW	RW	RW

Indicates that the transfer request by software is the last transfer.

To set the last transfer for channel N, set and write 1 to LSTDST[N] and LSTDST_WE[N].

If LSTDST_WE[N] is 0, the last transfer is not set even when LSTDST[N] is 1.

After setting the last transfer by this register, a transfer request becomes the last one when it is generated by the destination software transfer request register.

7.26.3.20. DMA Configuration Register (DMACm_CFG: BASE + 0x398)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															DMA_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Enables the DMA controllers. 0: Disabled, 1: Enabled

7.26.3.21. DMA Channel Enable Register (DMACm_CH_EN: BASE + 0x3A0)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																				CH_EN_WE				Reserved				CH_EN			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	R	R	R	R	RW	RW	RW	RW

Enables the DMA channel.

To enable channel N, set and write 1 to CH_EN[N] and CH_EN_WE[N].

To disable channel N, set and write 0 to CH_EN[N] and 1 to CH_EN_WE[N].

If CH_EN_WE[N] is 0, the enable/disable of channel N does not change, regardless of CH_EN [N].

7.26.3.22. DMAID Register (DMACm_ID: BASE + 0x3A8)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	DMA_ID																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Indicates the ID value of this peripheral.

7.26.4 Description of Operation

7.26.4.1. Flow Controller

The device that controls the transfer block length is called "flow controller". At DMA transfer, the DMA controller or a source/destination peripheral can be specified as the flow controller.

Normally, when the transfer block length is already known, the DMA controller can work as the flow controller. In this case, write the block size to the BLOCK_TS field of the DMACm_CTLn register. On the other hand, when the transfer block size is not known before starting transfer, specify a source or destination peripheral as the flow controller.

Table 7-26-1 Relationship among Transfer Type, Flow Controller, and TT_FC shows the possible combinations of transfer type and flow controller, and TT/FC setting value for each combination.

This LSI supports only the transfer types from the memory (RAM) to the peripheral (AES or SPI2) and from the peripheral (AES or SPI2) to the memory (RAM).

Table 7-26-1 Relationship among Transfer Type, Flow Controller, and TT_FC

Transfer type	Flow Controller	DMACm_CTLn.TT_FC
Memory →Peripheral	DMAC	001
Peripheral → Memory	DMAC	010

7.26.4.2. Handshake Interface

Handshake interface is used to control the flow at transaction level. The operation of handshake interface varies depending on the flow controller.

Peripherals use handshake interface to notify DMAC that transfer is ready.

Peripherals other than memory device can request a DMA transfer via hardware or software handshake interface. Software can select hardware or software handshake interface for each channel. Software handshake interface is realized by the control register.

Table 7-26-2 Correspondence between DMA Transfer Request Sources and Interface Numbers shows the assignment of hardware handshake interface in this LSI.

Table 7-26-2 Correspondence between DMA Transfer Request Sources and Interface Numbers

Interface number	Request source	Description	Remarks
0	SPI2 transmit DMA	Data transfer from RAM to SPI2	
1	SPI2 receive DMA	Data transfer from SPI2 to RAM	
2	AES transmit DMA	Data transfer from RAM to AES	Set to ch2. *
3	AES receive DMA	Data transfer from AES to RAM	Set to ch3. *
4-15	Reserved		

* Use DEST_PER and SRC_PER of the channel n configuration register to set the interface number.

7.26.4.3. Transfer Data Size

The following table shows the relationship between the transfer data size and control register value.

Data size for source single transfer (Byte)	DMACm_CTLn.SRC_TR_WIDTH/8
Data size for source burst transfer (Byte)	DMACm_CTLn.SRC_MSIZE*(DMAC_CTLn.SRC_TR_WIDTH/8)
Data size for destination single transfer(Byte)	DMACm_CTLn.DST_TR_WIDTH/8
Data size for destination burst transfer(Byte)	DMACm_CTLn.DST_MSIZE*(DMAC_CTLn.DST_TR_WIDTH/8)
Block size (Byte) (DMAC flow controller)	DMACm_CTLn.BLOCK_TS*(DMAC_CTLn.SRC_TR_WIDTH/8)
Block size (Byte) (source peripheral)	Number of burst transfers * Data size for source burst transfer + Number of single transfers * Data size for source single transfer
Block size (Byte) (destination peripheral)	Number of burst transfers * Data size for destination burst transfer + Number of single transfers * Data size for destination single transfer

7.26.4.4. Peripheral Burst Transfer Request

To prevent FIFO overflow/underflow in a peripheral, it is necessary to set appropriate values for the FIFO size of the peripheral, the FIFO level at which the peripheral makes a DMA transfer request, and the burst transfer size of the DMA controller. This section describes an example of transmit FIFO and receive FIFO by a peripheral which communicates with outside (transmit/receive).

For transmit FIFO of the peripheral, a transmit FIFO overflow may occur when the empty size of transmit FIFO of the peripheral is less than the burst transfer size that is transferred by the DMA controller at one burst transfer request. Normally, the burst size (DMACm_CTRLn.DEST_MSIZE) of the DMA controller should be equal to or less than the empty size of transmit FIFO.

If a large empty size (small threshold value) is set to the transmit FIFO that makes a DMA transfer request in the peripheral, it is more likely that a transmit FIFO underflow occurs.[10] On the other hand, if a small empty size (large threshold value) is set to the transmit FIFO, the number of burst transfers for block transfer increases, and the bus efficiency is reduced. The underflow of transmit FIFO depends on the bandwidth guaranteed by DMA transfer. Please take into account another arbitration method in the bus master or DMA controller or the bus matrix arbitration method.

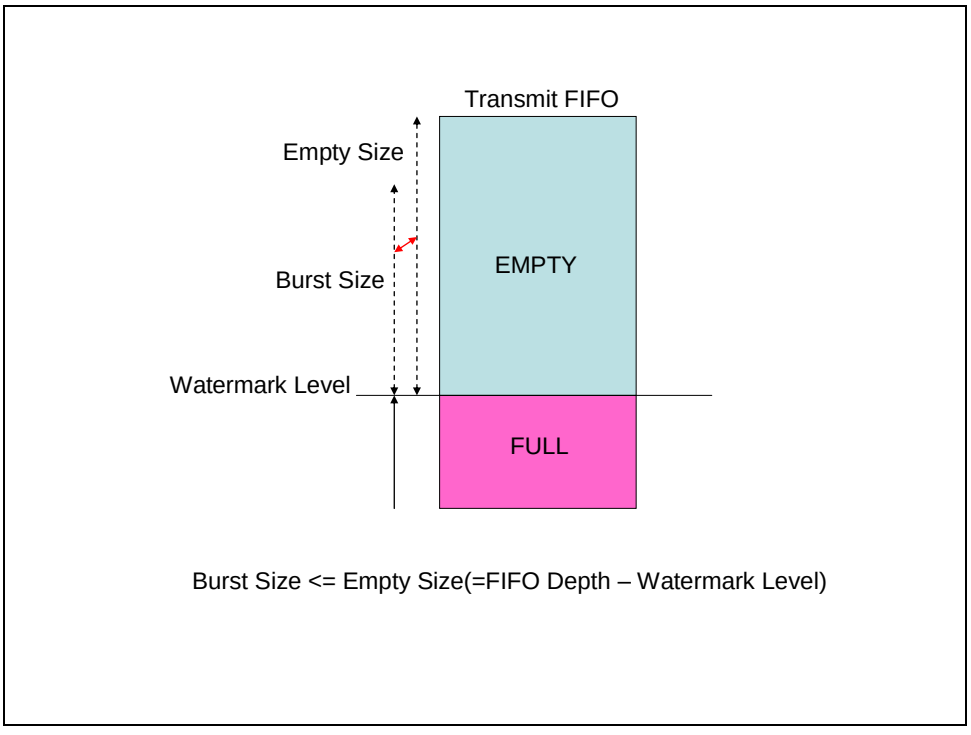


Figure 7-10 Relationship between Empty Size Transmit FIFO and Burst Size

[10] For example, an underflow may occur in SPI operating in slave mode.

For receive FIFO of the peripheral, a receive FIFO underflow may occur when the data size of receive FIFO of the peripheral is less than the burst transfer size that is transferred by the DMA controller at one burst transfer request. Normally, the burst size (DMACm_CTRLn.SRC_MSIZE) of the DMA controller should be equal to or less than the data size of receive FIFO.

If a large data size is set to the receive FIFO that makes a DMA transfer request in the peripheral, it is more likely that a receive FIFO overflow occurs. On the other hand, if a small data size is set to the receive FIFO, the number of burst transfers for block transfer increases, and the bus efficiency is reduced. The overflow of receive FIFO depends on the bandwidth guaranteed by DMA transfer. Please take into account another arbitration method in the bus master or DMA controller or the bus matrix arbitration method.

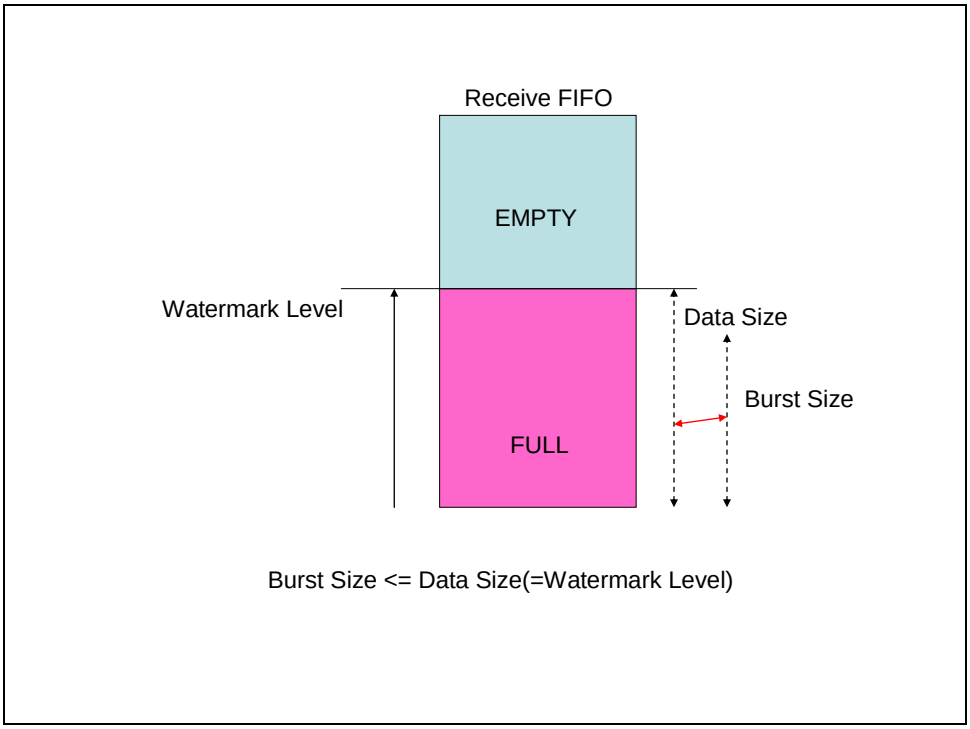


Figure 7-11 Relationship between Data Size Receive FIFO and Burst Size

7.26.4.5. Multiple Block Transfer

This DMA controller supports block chaining (linked list), auto-reload of channel register, and multiple block transfer with successive blocks.

7.26.4.5.1. Block Chaining (Linked List)

A linked list pointer (LLP) points the next linked list item (LLI) placed on the system memory. An LLI is a register set (block descriptor) that describes the next block. If block chaining is enabled, the DMA controller fetches an LLI each time when a block transfer is started.

An LLI consists of the following:

1. SARn
2. DARn
3. LLPn
4. CTLn
5. SSTATn
6. DSTATn

To enable block chaining, the linked list should be placed in the upper part of the memory.

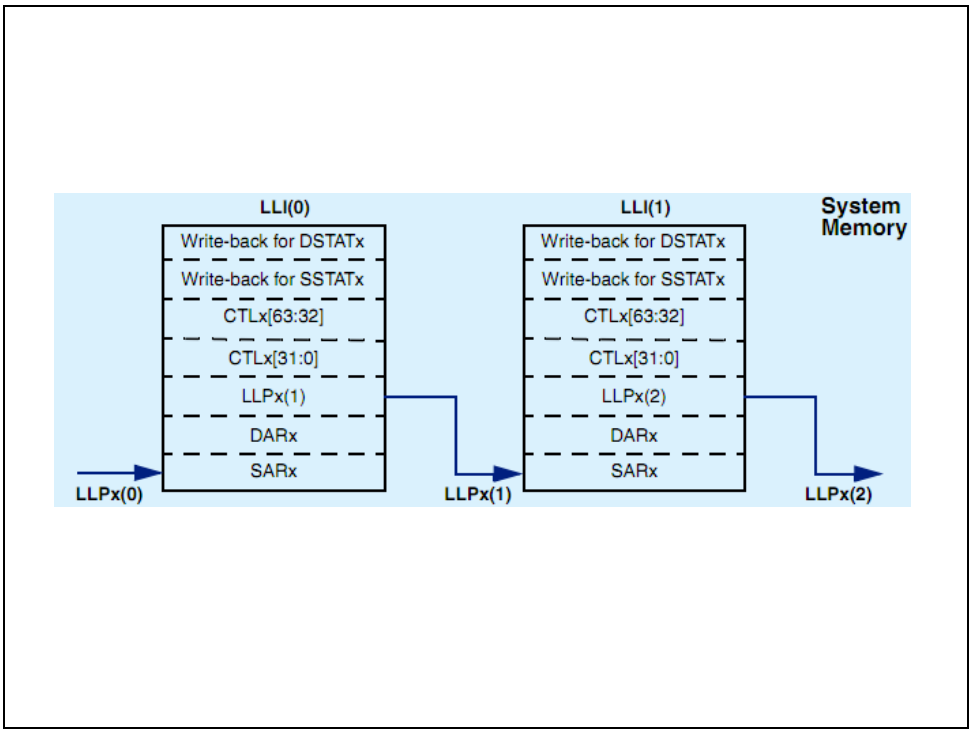


Figure 7-12 Linked List Schematic Diagram

7.26.4.5.2. Auto-reload of Channel Register

When a block transfer is completed, the register value at the first enabled channel is automatically reloaded.

7.26.4.6. Arbitration of AHB Master Interface

Each DMA channel has two request signals, source transfer request and destination transfer request.

Source transfer and destination transfer are arbitrated separately in the bus. Once the source or destination acquires the bus privilege, AHB transfer is started.

The arbitration circuit determines which transfer request (number of channels x 2) to grant the bus privilege. Each channel has a programmable priority. A transfer request can be generated at arbitrary timing, but the bus privilege is granted after AHB transfer is completed. If a transfer request with a higher priority is generated while executing a transfer with a lower priority, AHB transfer may be completed to switch from the transfer with a lower priority to the one with a higher priority.

To prevent a certain channel from occupying the bus, the maximum burst length can be set in the MAX_ABRST field of the DMACm_CFGn register.

If there is only one transfer request with the highest priority, the bus privilege of AHB is granted to it. If there are multiple transfer requests with the highest priority, the transfer request with the lowest channel number takes precedence.

7.27.Flash DMA

7.27.1 General Description

Flash DMA controls data write/verify of the Flash area in behalf of CPU to improve the throughput.
For example, while CPU is executing a program in Bank 0, Flash DMA writes an update program to Bank 1 and verifies it. The CPU performance during write operation is maintained by the multilayer AHB.

[Notices]
FlashDMA can be used only in 2 bank mode.
FlashDMA can control only Bank where the CPU does not execute any program. Therefore, FlashDMA can control Bank 1 while the CPU executes a program in Bank 0, and Bank 0 while the CPU executes a program in Bank 1.
When the Flash area is rewritten by FlashDMA, the FlashDMA register should be set so that the following address areas are not contained.
If these areas are specified as rewritable areas, the processing is terminated with an error (notifying the Flash ROM address exceeding transfer size specification error).

■ Non-rewritable Area

0x1C03_E7FC to 0x1C03_E7FF

0x1C03_FFFC to 0x1C03_FFFF

0x1803_FFFC to 0x1803_FFFF

7.27.2 List of Registers

Address [H]	Name	Symbol	R/W	Initial value [H]	Description
0x40160000	FlashDMA setting register 0	FDMA0	R/W	0x0000_0004	Sets the operation mode.
0x40160004	FlashDMA setting register 1	FDMA1	R	0x0000_0000	Notifies the interrupt.
0x40160008	FlashDMA setting register 2	FDMA2	R	0x0000_0000	Clears the interrupt.
0x4016000C	FlashDMA setting register 3	FDMA3	R	0x0000_0000	Notifies an error interrupt.
0x40160010	FlashDMA setting register 4	FDMA4	R	0x0000_0000	Clears an error interrupt.
0x40160014	FlashDMA setting register 5	FDMA5	R/W	0x0000_0000	Sets the start address of SRAM.
0x40160018	FlashDMA setting register 6	FDMA6	R/W	0x0000_0000	Sets the start address of FLASH.
0x4016001C	FlashDMA setting register 7	FDMA7	R/W	0x0000_0000	Sets the address size to be written to FLASH.

7.27.3 Description of Registers

7.27.3.1 FlashDMA setting register 0: 0x40160000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												FDMA_INT_MASK	FDMA_MODE		FDMA_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

FDMA_INT_MASK	Masks the interrupt. 0: No mask 1: Mask
FDMA_MODE	Sets the operation mode. 2' b00: Write only 2' b01: Verify only 2' b10: ALL 2' b11: Do not use
FDMA_EN	Starts FlashDMA. This is automatically cleared when the processing is stopped or completed. 1' b0: Standby (processing completed) * Setting it to 0 during start-up forcibly terminates FlashDMA operation. 1' b1: Start activation (during start)

7.27.3.2 FlashDMA setting register 1: 0x40160004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															FDMA_STAT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

FDMA_STAT	Indicates the interrupt status. 1: Indicates that an interrupt has been generated. 0: Indicates that no interrupt has been generated.

7.27.3.3 FlashDMA setting register 2: 0x40160008

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															FDMA_EOI
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

FDMA_EOI	When this register is read, the interrupt is cleared

7.27.3.4 FlashDMA setting register 3: 0x4016000C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																							FDMA_ERR_STAT								
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

FDMA_ERR_STAT	Indicates the error status. [0]: Transfer size 0 specification error (Start-up request is made with a size 0 specified by the size address) * 0x4016001C SIZE_ADDR[15:2] should be set to other than 0. [1]: SRAM start address specification error (Start-up request is made after a value outside the SRAM address range was written to the SRAM address register) * 0x40160014 SRAM_ADDR[31:2] should be set to an address in the SRAM area. [2]: SRAM address exceeding transfer size specification error (Start-up request is made when the size address was written where the end address determined by it exceeds the end address of SRAM) * The sum of 0x40160014 SRAM_ADDR[31:2] and 0x4016001C SIZE_ADDR[15:2] should not exceed the SRAM area addresses. [3]: Flash ROM start address specification error (Start-up request is made after a value outside the Flash-ROM address range was written to the Flash-ROM address register) * 0x40160018 FLASH_ADDR[31:2] should be set to an address in the FLASH ROM area excluding the security area. [4]: Flash ROM address exceeding transfer size specification error (Start-up request is made when the size address was written where the end address determined by it exceeds the end address of Flash) * The sum of 0x40160018 FLASH_ADDR[31:2] and 0x4016001C SIZE_ADDR[15:2] should not exceed the FLASH ROM area addresses excluding the security area. [5]: Error of Flash ROM running at transfer (flc_fewidle_o is L in the F_JDG1 state)

	* Possible causes include malfunction due to noise and IC hardware. If there is still trouble, contact us. [6]: Error of Flash ROM not started at transfer (flc_fewidle_o does not change to L from H in the F_JDG2 state) * Possible causes include malfunction due to noise and IC hardware. If there is still trouble, contact us. [7]: Flash Cont. error detected (an error signal (flc_fewerr_o) is detected from FlashCnt) * Possible causes include malfunction due to noise and IC hardware. If there is still trouble, contact us. [8]: Verify error (Verify results in mismatch) * For Verify only (FDMA_MODE=2' b01), check the address setting, etc. for an error. For ALL (FDMA_MODE=2' b10), possible causes include malfunction due to noise and chip hardware. If there is still trouble, contact us.

7.27.3.5 FlashDMA setting register 4: 0x40160010

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																								FDMA_ERR_EOI							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

FDMA_ERR_EOI	When this register is read, the error status is cleared.

7.27.3.6 FlashDMA setting register 5: 0x40160014

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	SRAM_ADDR																														Fixed to 0	Fixed to 0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	R

SRAM_ADDR	Specifies the start address of SRAM.

7.27.3.7 FlashDMA setting register 6: 0x40160018

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	FLASH_ADDR																														Fixed to 0	Fixed to 0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	R

FLASH_ADDR	Specifies the start address of FLASH ROM.

7.27.3.8 FlashDMA setting register 7: 0x4016001C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																SIZE_ADDR														Fixed to 0	Fixed to 0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	R

SIZE_ADDR	Specifies the address size to be written to FLASH ROM from SRAM.

7.27.4 Functional Description

Example of Flash write control by Flash DMA (solid line: data signal, broken line: control signal)

(1) CPU starts Flash DMA

(2) Flash DMA controls the read of RAM

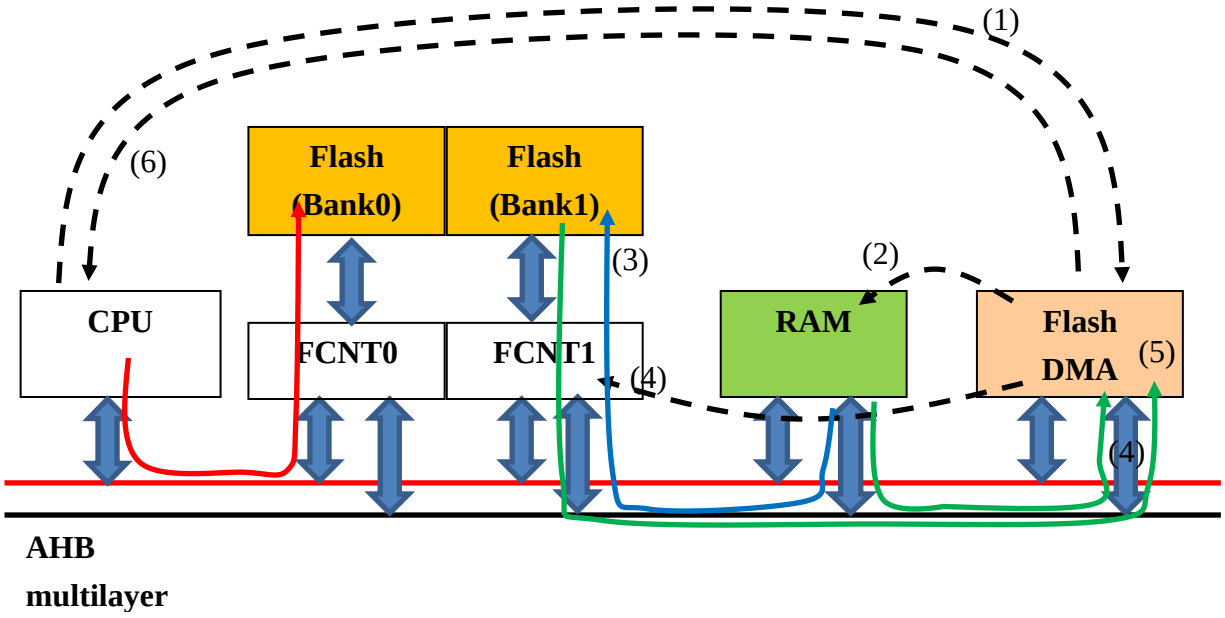
(3) Data transfer from RAM to Flash (write)

* Steps (2) to (3) are repeated

(4) Flash DMA controls the read of data from Flash and RAM

(5) Flash DMA compares (verify) data from Flash and RAM

(6) Flash DMA writes to CPU and notifies a Verify completion interrupt



7.28.AES

7.28.1 General Description

The AES (Advanced Encrypt Standard) encryption method prescribed by NIST (US National Institute of Standards and Technology) is provided.

The features are shown below.

- Conforms to FIPS PUB197 prescribed by NIST (US National Institute of Standards and Technology)
- Supports the encryption key size of 128/192/256 bits
- Supports the following encryption use modes:
  - Electronic Code Book (ECB)
  - Cipher Block Chaining (CBC)
  - Cipher Feedback (CFB)
  - Output Feedback(OFB)
  - Counter (CTR)
  - Counter with CBC-MAC (CCM)
  - Galois/Counter Mode (GCM)
- Encrypts/decrypts in 12/14/16 clock cycles (encryption key size of 128/192/256 bits)
- Simultaneous authentication TAG generation and encryption/decryption (CCM/GCM)
- Reduces execution time with two-stage input buffer
- Supports DMAC

7.28.2 List of Registers

Address [H]	Name	Symbol	R/W	Initial value [H]	Description
0x40170000	AES control register	AES_CTL	R/W	0x0000_0000	Enables the AES block.
0x40170004	AES mode setting register	AES_MOD	R/W	0x0000_0000	Sets the operation mode.
0x40170008	GCM-CCM control register	AES_GCCM_CTL	R/W	0x0000_0000	Controls the CCM/GCM mode.
0x4017000C	AES status register	AES_ST	R/W	0x0000_0000	Displays the AES status.
0x40170010	Interrupt RAW status register	AES_RIST	R	0x0000_0000	Displays the interrupt status (before mask).
0x40170014	Interrupt status register	AES_IST	R	0x0000_0000	Displays the interrupt status (after mask).
0x40170018	Interrupt mask register	AES_IMSK	R/W	0x0000_0000	Masks the interrupt.
0x4017001C	Interrupt clear register	AES_ICLR	R	0x0000_0000	Clears an interrupt.
0x40170020	AES_BLKDONE interrupt clear register	AES_CLR_BLKDONE	R	0x0000_0000	Clears an AES_BLKDONE interrupt.
0x40170024	AES_DONE interrupt clear register	AES_CLR_DONE	R	0x0000_0000	Clears an AES_DONE interrupt.
0x40170028	Encryption key setting register 0	AES_KEY0	W	0x0000_0000	Sets the encryption key [31:0].
0x4017002C	Encryption key setting register 1	AES_KEY1	W	0x0000_0000	Sets the encryption key [63:32].
0x40170030	Encryption key setting register 2	AES_KEY2	W	0x0000_0000	Sets the encryption key [95:64].
0x40170034	Encryption key setting register 3	AES_KEY3	W	0x0000_0000	Sets the encryption key [127:96].
0x40170038	Encryption key setting register 4	AES_KEY4	W	0x0000_0000	Sets the encryption key [159:128].
0x4017003C	Encryption key setting register 5	AES_KEY5	W	0x0000_0000	Sets the encryption key [191:160].
0x40170040	Encryption key setting register 6	AES_KEY6	W	0x0000_0000	Sets the encryption key [223:192].

0x40170044	Encryption key setting register 7	AES_KEY7	W	0x0000_0000	Sets the encryption key [255:224].
0x40170048	HASH sub key setting register 0	AES_HKEY0	R/W	0x0000_0000	Sets the HASH sub key [31:0].
0x4017004C	HASH sub key setting register 1	AES_HKEY1	R/W	0x0000_0000	Sets the HASH sub key [63:32].
0x40170050	HASH sub key setting register 2	AES_HKEY2	R/W	0x0000_0000	Sets the HASH sub key [95:64].
0x40170054	HASH sub key setting register 3	AES_HKEY3	R/W	0x0000_0000	Sets the HASH sub key [127:96].
0x40170058	IV setting register 0	AES_IV0	R/W	0x0000_0000	Sets the Initialization Vector [31:0].
0x4017005C	IV setting register 1	AES_IV1	R/W	0x0000_0000	Sets the Initialization Vector [63:32].
0x40170060	IV setting register 2	AES_IV2	R/W	0x0000_0000	Sets the Initialization Vector [95:64].
0x40170064	IV setting register 3	AES_IV3	R/W	0x0000_0000	Sets the Initialization Vector [127:96].
0x40170068	CTRIV setting register 0	AES_CTRIV0	R/W	0x0000_0000	Sets the counter initial value [31:0].
0x4017006C	CTRIV setting register 1	AES_CTRIV1	R/W	0x0000_0000	Sets the counter initial value [63:32].
0x40170070	CTRIV setting register 2	AES_CTRIV2	R/W	0x0000_0000	Sets the counter initial value [95:64].
0x40170074	CTRIV setting register 3	AES_CTRIV3	R/W	0x0000_0000	Sets the counter initial value [127:96].
0x40170078	Input data register	AES_IDATA	R/W	0x0000_0000	Sets the input data [31:0].
0x4017007C	Output data register	AES_ODATA	R	0x0000_0000	Displays the AES conversion result [31:0].
0x40170080	Output data 2 register 0	AES_ODATA20	R	0x0000_0000	Displays the AES conversion result [31:0].

0x40170084	Output data 2 register 1	AES_ODATA21	R	0x0000_0000	Displays the AES conversion result [63:32].
0x40170088	Output data 2 register 2	AES_ODATA22	R	0x0000_0000	Displays the AES conversion result [95:64].
0x4017008C	Output data 2 register 3	AES_ODATA23	R	0x0000_0000	Displays the AES conversion result [127:96].

7.28.3 Description of Registers

7.28.3.1 AES control register: 0x40170000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																													DMA_RX_EN	DMA_TX_EN	AES_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

DMA_RX_EN	Enables DMA transfer of input data. 0: Disabled 1: Enabled
DMA_TX_EN	Enables DMA transfer of output data. 0: Disabled 1: Enabled
AES_EN	Enables AES conversion. 0: Disabled 1: Enabled Cleared by completion of CCM or GCM mode (AES_DONE)

7.28.3.2 AES mode setting register: 0x40170004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																					STASIN			OPMOD				KEYSIZE		AUTHMOD	CIPHMOD
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

STASIN	Switches the STATE allocation For description of the function, refer to 7.28.4.1 "STATE Configuration".
OPMOD	Selects the encryption use mode. 0x0 : ECB 0x1 : CBC 0x2 : CFB 0x3 : OFB 0x4 : CTR 0x5 : CCM 0x6 : GCM 0x7 : GHASH 0x11-0x15 : Reserved
KEYSIZE	Sets the encryption key length. 00 : 128bit 01 : 192bit 10 : 256bit 11: Setting prohibited (128 bit)

AUTHMOD	Authentication TAG generation mode For description of the function, refer to 7.28.4.5.
CIPHMOD	Selects encryption/decryption 0 : Encryption 1 : Decryption

7.28.3.3 GCM-CCM control register: 0x40170008

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PLEN																ALEN															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PLEN	Sets the number of bytes of data encrypted/decrypted in GCM/CCM mode.
ALEN	Sets the number of bytes of data that is authenticated (not encrypted/decrypted) in GCM/CCM mode only.

7.28.3.5 AES status register: 0x4017000C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved									STMON							Reserved		OBRP		IBWP		IBRP							AESBUSY	AESOBFL	AESIBFL
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

STMON	Displays the internal state of AES block.
OBRP	Read pointer monitor of ODATA 00 : ODATA0 01 : ODATA1 10 : ODATA2 11 : ODATA3
IBWP	Write pointer monitor of IDATA 00 : IDATA0 01 : IDATA1 10 : IDATA2 11 : IDATA3
IBRP	Read pointer monitor of IDATA 00 : IDATA0 01 : IDATA1 10 : IDATA2 11 : IDATA3
AESBUSY	Displays the AES conversion state 0: Conversion completed or being stopped

	1: Converting
AESOBFL	Indicates that ODATA has unread conversion results. It is cleared by 4-word read. 0: No data 1: Unread conversion results Writing "1" to this bit clears the following register. •OBRP (read pointer of ODATA)
AESIBFL	Disables IDATA input 0: Input to IDATA enabled. 1: Input to IDATA disabled. This indicates that the input buffer is filled with unprocessed AES data. Writing "1" to this bit clears the following registers. •IBWP (write pointer of IDATA) •IBRP (read pointer of IDATA) •AESIBFL (FULL status of write buffer) •AESOBFL (FULL status of read buffer)

7.28.3.6 Interrupt RAW status register: 0x40170010

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														AES_DONE	AES_BLKDONE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

AES_DONE	This bit is set when encryption/decryption of CCM or GCM is completed. 0: Not completed or stopped 1: Completed
AES_BLKDONE	This bit is set when encryption/decryption by block (128 bits) is completed. 0: Not completed or stopped 1: Completed

7.28.3.7 Interrupt status register: 0x40170014

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																														R_AES_DONE	R_AESBLK_DONE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

R_AES_DONE	Indicates the interrupt status.
R_AES_BLKDONE	For details of each bit, refer to the interrupt RAW status register. The masked interrupt status is read from this register.
	To clear the interrupt source, read the interrupt clear register corresponding to the appropriate interrupt source.

7.28.3.8 interrupt mask register: 0x40170018

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																														M_AES_DONE	M_AESBLK_DONE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

M_AES_DONE	Masks the AES_DONE interrupt. 0: No mask 1: Mask
M_AES_BLKDONE	Masks the AES_BLKDONE interrupt. 0: No mask 1: Mask

7.28.3.9 Interrupt clear register: 0x4017001C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															CLR_AESINTR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

CLR_AESINTR	When this register is read, the AES interrupt (AES_BLKDONE, AES_CONE) is cleared.

7.28.3.10 AESBLK_DONE interrupt clear register: 0x40170020

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															CLR_AESBLK
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

CLR_AESBLK_DONE	When this register is read, the AES_BLKDONE interrupt is cleared.

7.28.3.11 AES_DONE interrupt clear register: 0x40170024

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															CLR_AES_DONE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

CLR_AES_DONE	When this register is read, the AES_DONE interrupt is cleared.

7.28.3.12 Encryption Key Setting Register x: 0x40170028 - 0x40170044

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

KEY	Sets the encryption key.

7.28.3.13 HASH Sub Key Setting Register x: 0x40170048 - 0x40170054

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HKEY																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

HKEY	Sets the HASH sub key.

7.28.3.14 IV Setting Register x: 0x40170058 - 0x40170064

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IV																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

IV	Sets the Initialization Vector.

7.28.3.15 CTRIV Setting Register x: 0x40170068 - 0x40170074

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CTRIV																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

CTRIV	Sets the counter initial value of CTR/CCM/GCM mode. The lower 16 bits are automatically incremented each time AES conversion of block is completed.

7.28.3.16 Input Data Register x: 0x40170078

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IDATA																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

IDATA	Sets the input data.

7.28.3.17 Output Data Register x: 0x4017007C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ODATA																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

ODATA	Displays the AES conversion result.

7.28.3.18 Output Data 2 Register x: 0x40170080 - 0x4017008C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ODATA2																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R / W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

ODATA2	Displays the AES conversion result. It displays the decryption result of authentication TAG when decryption is performed in CCM/GCM mode.

7.28.4 Functional Description

7.28.4.1 About Encryption Key Length

The encryption key length can be set in the AES_MOD.KEYSIZE register. The following table shows the encryption key setting registers used to set the encryption key length.

Encryption key length	AES_KEY0 to 3	AES_KEY4 to 5	AES_KEY6 to 7
128 bits	Need to be set	-	-
192 bits	Need to be set	Need to be set	-
256 bits	Need to be set	Need to be set	Need to be set

7.28.4.2 Conversion Cycle

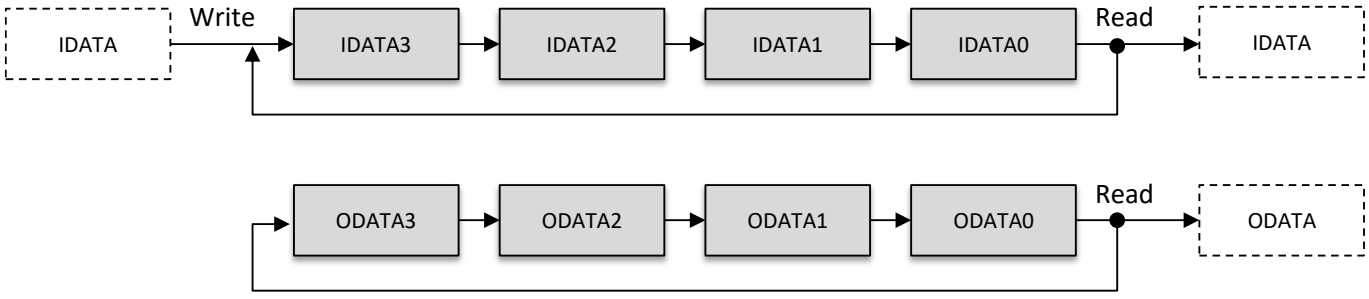
The following table shows the number of conversion cycles in each encryption use mode.

Encryption use mode	Conversion cycle (cycle/block)		
	AES-128	AES-192	AES-256
ECB/CBC/CFB/OFB/CTR/GCM/GHASH/MULT/CTR+GHASH	12	14	16
CCM/CTR+CBC	24	28	32

7.28.4.3 IDATA/ODATA Access

IDATA/ODATA is the beginning address of a 4-word (128-bit) barrel shifter.

The schematic diagram is shown below.



When 4-word data is written to IDATA, the first written word is written to IDATA0, and the last one is written to IDATA3. At reading, the first read data indicates IDATA0, and the last one indicates IDATA3. Same for when reading ODATA.

7.28.4.4 STATE Configuration

The following figures show the correspondence between STATE defined in FIPS PUB197 and IDATAx register.
The correspondence between STATE and IDATAx can be switched by STASIN.
The same allocation is applied to ODATA.

Example: STASIN=000 (default)

[31:24]	$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
[23:16]	$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
[15:8]	$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
[7:0]	$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$
	IDATA0	IDATA1	IDATA2	IDATA3

STASIN[0] changes the BYTE assignment.

Example: STASIN=001

[7:0]	$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
[15:8]	$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
[23:16]	$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
[31:24]	$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$
	IDATA0	IDATA1	IDATA2	IDATA3

STASIN[1] changes the block assignment of IDATA0 to IDATA3.

Example: STASIN=010

[31:24]	$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
[23:16]	$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
[15:8]	$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
[7:0]	$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$
	IDATA3	IDATA2	IDATA1	IDATA0

STASIN[2] changes the order of bits.

Example: STASIN=100

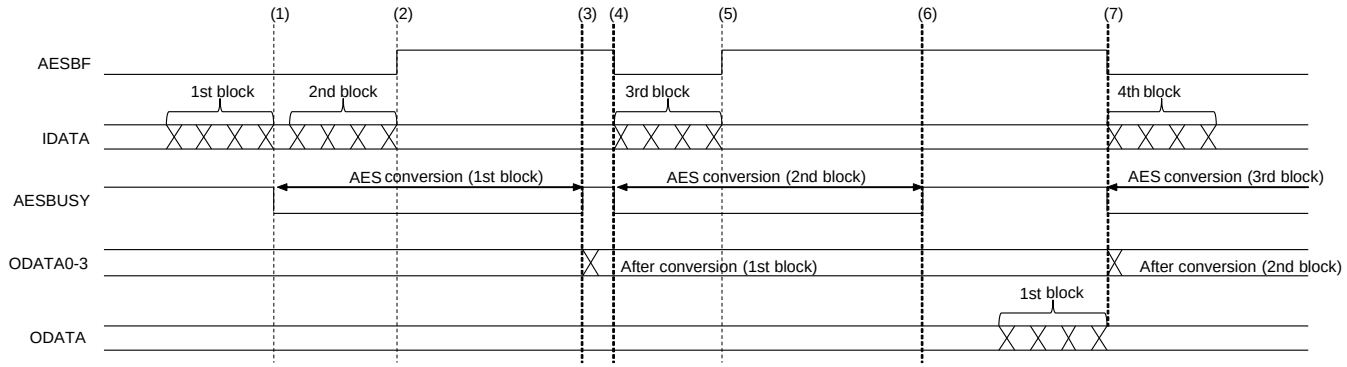
[24:31]	$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
[16:23]	$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
[8:15]	$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
[0:7]	$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$
	IDATA0	IDATA1	IDATA2	IDATA3

STASIN[0], STASIN[1], and STASIN[2] can be used in combinations.

7.28.4.5 Authentication TAG Generation Mode

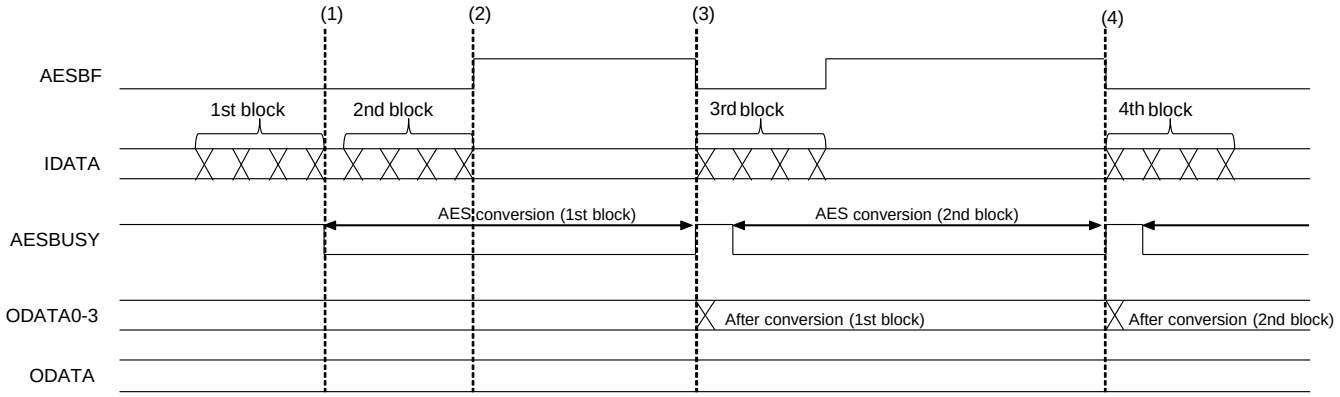
The authentication TAG generation mode (AUTHMOD = 1) is optimum when only the final result of multiple block encryption is required, such as when generating an authentication TAG.
If AUTHMOD = 0, the next block cannot be set until the conversion result of the previous block is read. On the other hand, if AUTHMOD = 1, the next block can be set when the conversion of the previous block is completed. When using DMAC, if AUTHMOD = 0, a transfer request of ODATA is generated each time a block conversion is completed. On the other hand, if AUTHMOD = 1, a transfer request of ODATA is only generated after the last data (the last data of transfer to IDATA by DMAC) is converted.

•Timing of IDATA setting (when AUTHMOD = 0)



(1) When 4-word data is written from IDATA, AES conversion is started.
(2) When 4-word data is written again, the input buffer becomes full, and AESBF changes to "1".
(3) When the conversion of the 1st block is completed, AESBUSY changes to "1", and the converted data is stored in the internal ODATA0-3.
(4) As the data of the 2nd block is already set, AESBUSY changes to "0" in one cycle, and the conversion of the 2nd block is started. When the conversion of the 2nd block is started, AESBF changes to "0", which allows the next block to be set.
(5) When the 3rd block is written, the input buffer becomes full, and AESBF changes to "1".
(6) The conversion of the 2nd block is completed, and AESBUSY changes to "1". However, the conversion result of the 2nd block is not stored in ODATA0-3 until the conversion result of the 1st block is read, and the conversion of the 3rd block is not started.
(7) When the converted data of the 1st block has been read from ODATA, the converted data of the 2nd block is stored in ODATA0-3, and the conversion of the 3rd block is started. In addition, AESBF changes to "0", which allows the next block set to be set.

•Timing of IDATA setting (when AUTHMOD = 1)



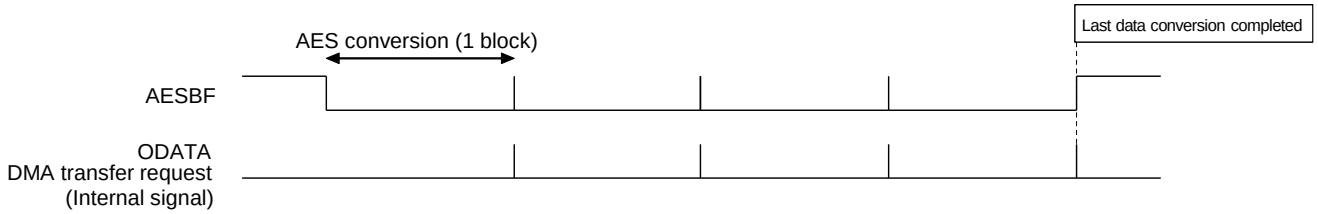
(1) When 4-word data is written from IDATA, AES conversion is started.

(2) When 4-word data is written again, the input buffer becomes full, and AESBF changes to "1".

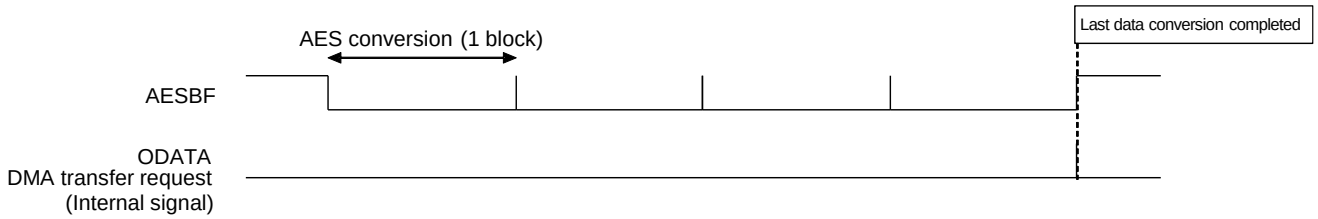
(3) When the conversion of the 1st block is completed, AESBUSY changes to "1", and the converted data is stored in the internal ODATA0-3. In addition, at the conversion completion of the 1st block, AESBF changes "0", which allows the input of the next block, without needing to read the converted data from ODATA.

(4) When the conversion of the 2nd clock is completed, ODATA0-3 is overwritten with the converted data of the 2nd block, whether or not the 1st block is read.

•Timing of transfer request to DMAC (AUTHMOD = 0)



•Timing of transfer request to DMAC (AUTHMOD = 1)



7.28.4.6 Input/Output Block Size

The following table shows the input data block size and output data block size for each encryption use mode.

Encryption use mode	Encryption/decryption	Input (IDATA)	Output (ODATA)
CCM	Encryption	Authentication data size (ALEN) + Plain text/encrypted text size (PLEN)	Encrypted text/plain text size (PLEN)+ 1 (authentication TAG)
	Decryption	Authentication data size (ALEN) + Plain text/encrypted text size (PLEN) + 1 (authentication TAG before decryption)	Encrypted text/plain text size (PLEN)+ 1 (authentication TAG)
GCM	Encryption	Authentication data size (ALEN) + Plain text/encrypted text size (PLEN) + 1 (bit width of authentication data and plain text)	Encrypted text/plain text size (PLEN)+ 1 (authentication TAG)
	Decryption	Authentication data size (ALEN) + Plain text/encrypted text size (PLEN) + 1 (authentication TAG before decryption) + 1 (bit width of authentication data and plain text)	Encrypted text/plain text size (PLEN)+ 1 (authentication TAG)
Other	Encryption/decryption	Plain text/encrypted text size	Same for input

7.28.4.7 ECB

Executes the Electronic Code Book (ECB) described in NIST Special Publication 800-38A.

•Input

IDATA : Plain text or encrypted text

KEY : Encryption key

•Output

ODATA : Encrypted text or plain text

7.28.4.8 CBC

Executes the Cipher Block Chaining (CBC) described in NIST Special Publication 800-38A.

•Input

IV : Initial vector

IDATA : Plain text or encrypted text

KEY : Encryption key

•Output

ODATA : Encrypted text or plain text

While AES_EN = 1 is maintained, the output result of the previous block encryption is used for the initial vector of the second and subsequent block encryptions.

7.28.4.9 CFB

Executes the Cipher Feedback (CFB) described in NIST Special Publication 800-38A.

•Input

IV : Initial vector

IDATA : Plain text or encrypted text

KEY : Encryption key

•Output

ODATA : Encrypted text or plain text

While AES_EN = 1 is maintained, the output result of the previous block encryption is used for the initial vector of the second and subsequent block encryptions.

Note) The bit width to feedback is fixed to 128 bits. In other than 128 bits, IV should be reset each time a block encryption is completed.

7.28.4.10 OFB

Executes the Output Feedback (OFB) described in NIST Special Publication 800-38A.

•Input

IV : Initial vector

IDATA : Plain text or encrypted text
KEY : Encryption key

•Output
ODATA : Encrypted text or plain text

While AES_EN = 1 is maintained, the output result of the previous block encryption (result before XOR with plain text) is used for the initial vector of the second and subsequent block encryptions.

7.28.4.11 CTR

Executes the Counter (CTR) described in NIST Special Publication 800-38A.

•Input
CTRIV : Counter initial value
IDATA : Plain text or encrypted text
KEY : Encryption key

•Output
ODATA : Encrypted text or plain text

While AES_EN = 1 is maintained, the counter value is incremented each time a block encryption is completed.

7.28.4.12 CCM

In CCM mode described in NIST Special Publication 800-38C, encryption, decryption and authentication TAG generation in CCM are simultaneously executed.

[Encryption]

•Input

IV : Initial vector of CBC

CTRIV : Counter initial value of CTR

IDATA : Authentication data, plain text

KEY : Encryption key

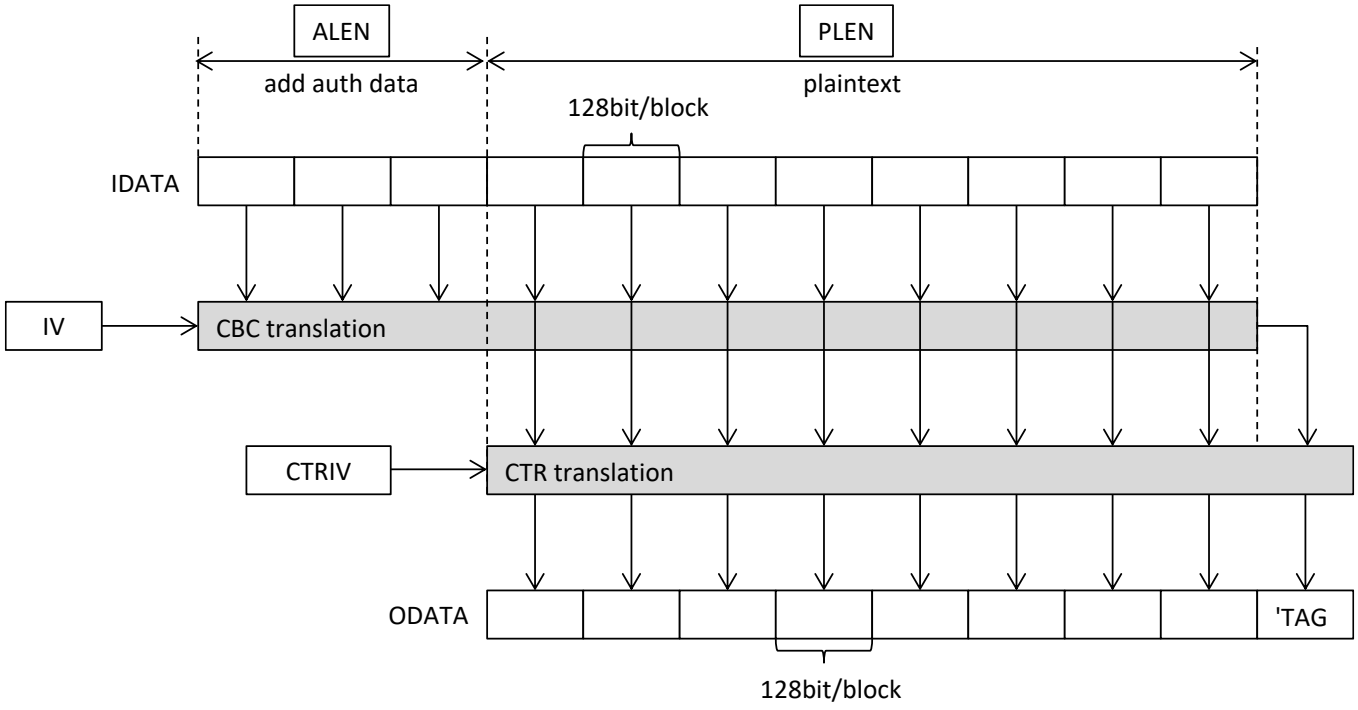
ALEN : Number of bytes of authentication data

PLEN : Number of bytes of plain text

•Output

ODATA : Encrypted text, authentication TAG (final data)

The schematic diagram of CCM mode (encryption) is shown below.



IDATA has the data for authentication TAG generation (data to execute only CBC) and the data to be encrypted (data to execute CBC and CTR), in this order.
ALEN has the number of bytes of the data for TEG generation.
PLEN has the data to be encrypted.
The hardware changes the internal operation depending on the values set to ALEN and PLEN.
IV has the initial vector of CBC.
IV2 has the counter initial value of CTR. The counter value is automatically incremented when the conversion of block is completed.
ODATA has the output encrypted data and TAG. The final output data indicates TAG.
The initial vector of CBC and the counter value of CTR are automatically changed by the hardware until the conversions of the number of bytes set by ALEN and PLEN are completed.
Reading of the last plaintext data triggers the encryption of authentication TAG.

[Decryption]

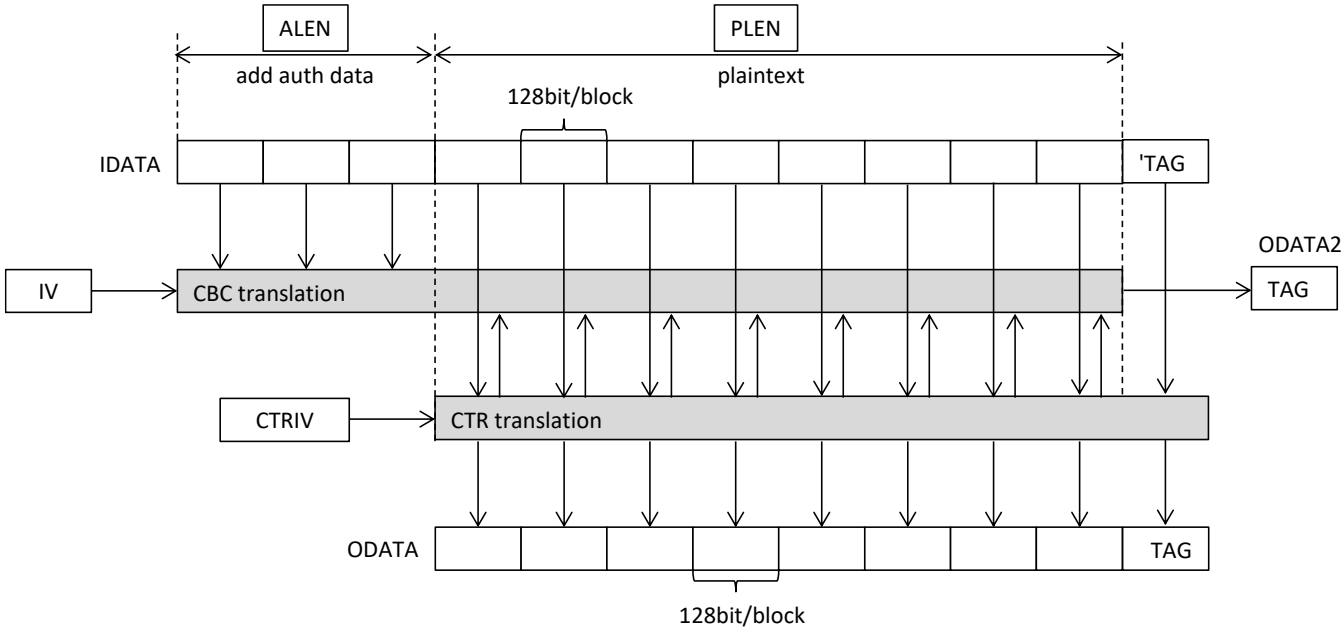
•Input

IV	: Initial vector of CBC
CTRIV	: Counter initial value of CTR
IDATA	: Authentication data, encrypted text, and authentication TAG
KEY	: Encryption key
ALEN	: Number of bytes of authentication data
PLEN	: Number of bytes of encrypted text

•Output

ODATA	: Decrypted text, decrypted data of the encrypted authentication TAG (last data)
ODATA2	: Authentication TAG generated from the decrypted text

The schematic diagram of CCM mode (decryption) is shown below.



The last ODATA data indicates the data decrypted from the encrypted authentication TAG. The last ODATA2 data indicates the authentication TAG generated from the decrypted message.

7.28.4.13 GCM

Simultaneously executes encryption, decryption and authentication TAG generation in Galois/Counter Mode (GCM) described in NIST Special Publication 800-38D.

[Encryption]

•Input

IV : Initial vector of GHASH

CTRIV : Counter initial value of CTR

IDATA : Authentication data, plain text, and their bit widths

KEY : Encryption key

HKEY : HASH sub key of GHASH

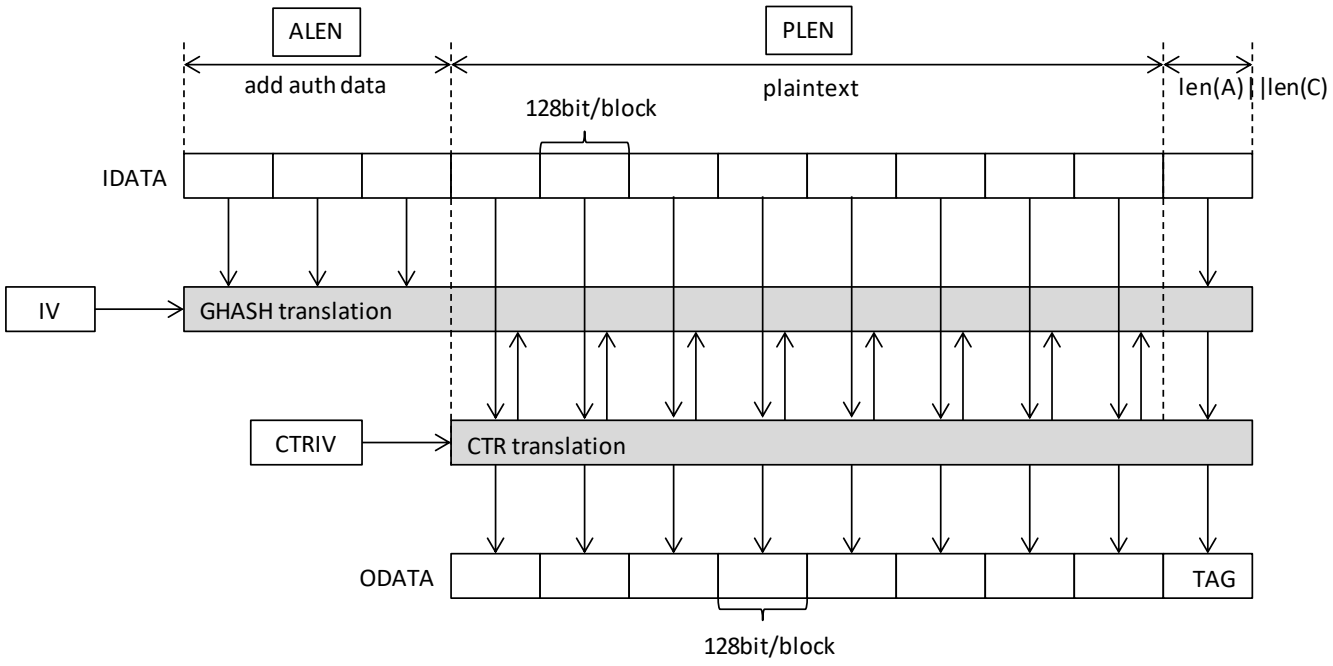
ALEN : Number of bytes of authentication data

PLEN : Number of bytes of plain text

•Output

ODATA : Encrypted text, authentication TAG (final data)

The schematic diagram of GCM mode (encryption) is shown below.



IDATA has the data for authentication TAG generation (data to execute only GHASH), data to be encrypted (data to execute GHASH and CTR), and bit width information of data for authentication TAG generation and data to be encrypted, in this order.
ALEN has the number of blocks of the data for TEG generation.
PLEN has the number of blocks of the data to be encrypted.
The hardware changes the internal operation depending on the values set to ALEN and PLEN.
IV has the initial vector of GHASH. For details of the initial vector, refer to the section about GHASH.
CTRIV has the counter initial value of CTR. The counter value is automatically incremented when the conversion of block is completed.
ODATA has the output encrypted data and TAG. The final output data indicates TAG.
The initial vector of GHASH and the counter value of CTR are automatically changed by the hardware until the conversions of the number of bytes set by ALEN and PLEN are completed.

[Decryption]

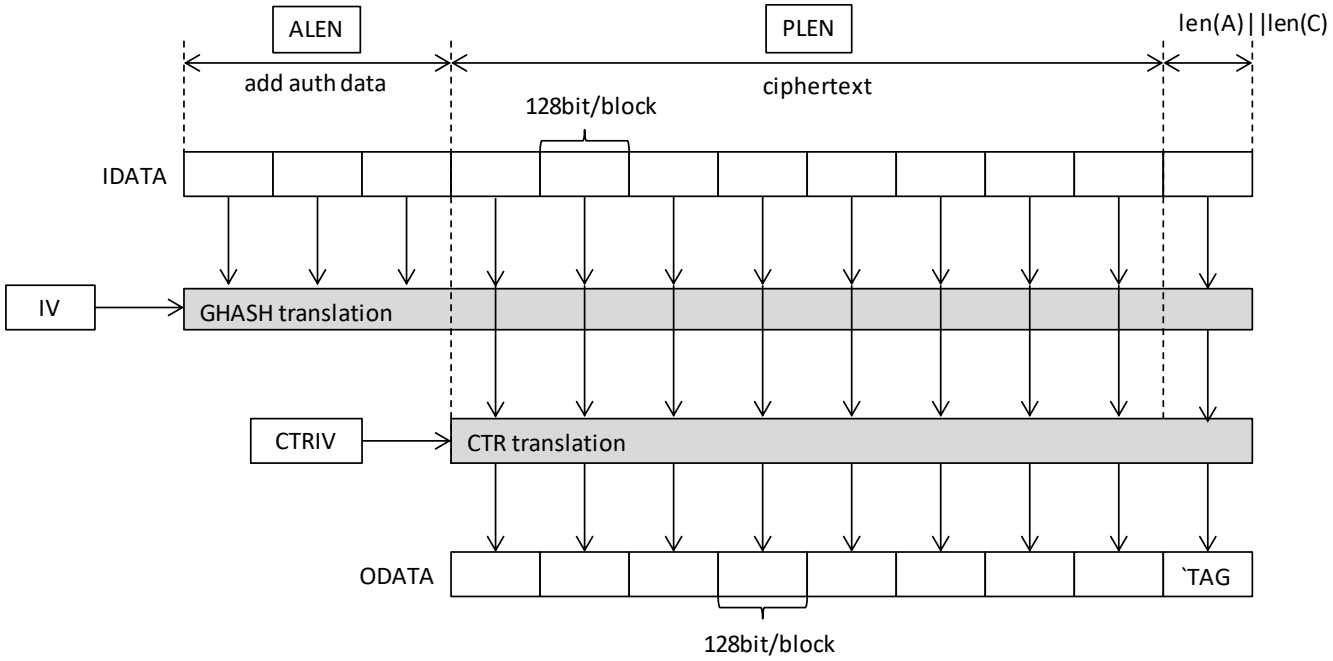
•Input

IV	: Initial vector of GHASH
CTRIV	: Counter initial value of CTR
IDATA	: Authentication data, encrypted text, authentication TAG, and bit widths of authentication data and encrypted text
KEY	: Encryption key
HKEY	: HASH sub key of GHASH
ALEN	: Number of bytes of authentication data
PLEN	: Number of bytes of encrypted text

•Output

ODATA	: Decrypted text, authentication TAG generated from the decrypted text (last data)

The schematic diagram of GCM mode (decryption) is shown below.



The last ODATA data indicates the authentication TAG generated from the decrypted message.

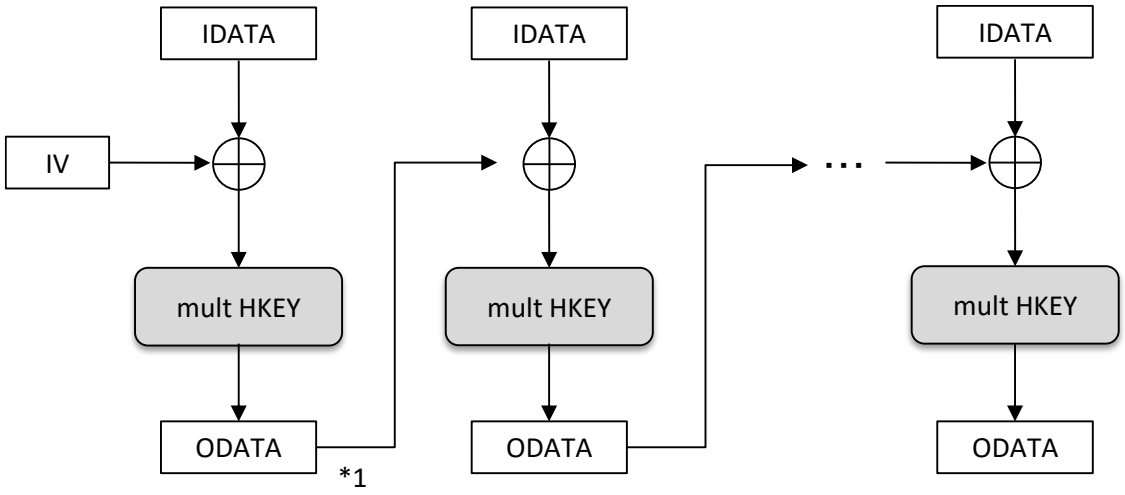
7.28.4.14 GHASH

Performs the GHASH conversion described in NIST Special Publication 800-38D.

•Input
HKEY : HASH sub key
IV : Initial vector
IDATA : Input data

•Output
ODATA : GHASH(HKEY, IV, IDATA)

The schematic diagram of GHASH mode is shown below.



*1: While AES_EN = 1 is maintained, the value of ODATA is used for the initial vector of the second and subsequent blocks.

7.28.5 Description of Operation

7.28.5.1 Control Example (without DMAC)

A control example when not using DMAC is shown below.

(1) Set the AES_MOD register.
(2) Set the input information required for the encryption use mode to execute.
(IV, CTRIV, KEY, HKEY, ALEN, PLEN)
(3) Set AES_CTL. DMA_TX_EN=DMA_RX_EN=0, AES_EN=1.
(4) Check that AES_BLKDONE is "0".
(5) Set the input data to IDATA. When 4-word write is completed, AES conversion is started.
(6) After checking that AES_BLKDONE is "1", read ODATA.
(7) Clear AES_BLKDONE.
(8) Repeat steps (5) to (7) for the number of bytes of input data.
When all conversions are completed, negate AES_EN.

Caution)

- In CCM mode, after the encryption of the last plaintext data is completed, reading ODATA triggers the encryption of authentication TAG.
- In CCM or GCM mode, AES_EN is automatically negated when the conversion is completed.

7.28.5.2 Control Example (DMAC)

A control example when using DMAC is shown below.

(1) Set the AES_MOD register.
(2) Set the input information required for the encryption use mode to execute.
(IV, CTRIV, KEY, HKEY, ALEN, PLEN)
(3) Set AES_CTL. DMA_TX_EN=DMA_RX_EN=1, AES_EN=1.
(4) Check that AES_BLKDONE is "0".
(5) Set the input data to RAM.
(6) Set the register of DMAC. The following settings are recommended:
- Set IDATA to the destination address of CH2.

- Set ODATA to the source address of CH3.
- The data transfer width should be a word (32 bit).
- The burst length of the burst transfer should be 4 words.

(7) Start DMAC transfer.

(8) After DMAC transfer is completed, read AES_BLKDONE or AES_DONE to confirm that AES conversion is completed, and negate AES_EN.

7.29.MODE_CNT

7.29.1 General Description

Mode control: Function used to control the operation mode (Flash bank configuration, PLL/clock dividing, etc.).

7.29.2 List of Registers

Address [H]	Name	Symbol	R/W	Initial value [H]	Description
0x40050000	WDT setting register		RW	0x00000001	WDT setting
0x40050004	TIMER setting register		RW	0x00000000	Timer setting
0x4005000C	PLL setting register 0		RW	0x00081359	PLL setting 0
0x40050010	PLL setting register 1		RW	0x00000000	PLL setting 1
0x40050018	CLKGEN setting register 0		RW	0x00000000	PLL setting 0
0x4005001C	CLKGEN setting register 1		RW	0x00000000	PLL setting 1
0x40050020	CLKGEN setting register 2		RW	0x00000000	PLL setting 2
0x40050024	CLKGEN setting register 3		RW	0x00000000	PLL setting 3
0x40050028	CLKGEN setting register 4		RW	0x00000000	PLL setting 4
0x4005002C	CLKGEN setting register 5		RW	0x00000000	PLL setting 5
0x40050030	CLKGEN setting register 6		RW	0x00000000	PLL setting 6
0x40050034	CLKGEN setting register 7		RW	0x00000000	PLL setting 7
0x40050038	CLKGEN setting register 8		RW	0x00000007	PLL setting 8
0x4005003C	CLKGEN setting register 9		RW	0x00000056	PLL setting 9
0x40050040	Deep sleep control register		RW	0x00000000	Deep sleep setting
0x40050044	FLASH-ROM deep standby control register		RW	0x00000000	FLASH-ROM deep standby control
0x40050048	FLASH WakeUp time setting register		RW	0x0000000E	FLASH WakeUp time setting
0x4005004C	Power-off release interval setting register		RW	0x00004B00	Power-off release interval setting
0x40050050	MODE_CNT interrupt mask register		RW	0x00000000	MODE_CNT interrupt mask
0x40050054	MODE_CNT interrupt status register		RW	0x00000000	MODE_CNT interrupt status
0x40050058	MODE_CNT RAW interrupt status register		RW	0x00000000	MODE_CNT RAW

					interrupt status
0x4005005C	High-speed CR stabilization completion interrupt clear register		RW	0x00000000	High-speed CR stabilization completion interrupt clear
0x40050060	Low-speed CR stabilization completion interrupt clear register		RW	0x00000000	Low-speed CR stabilization completion interrupt clear
0x40050064	XTAL32kHz stabilization completion interrupt clear register		RW	0x00000000	XTAL32kHz stabilization completion interrupt clear
0x40050068	PLL start completion interrupt clear register		RW	0x00000000	PLL start completion interrupt clear
0x4005006C	MODE_CNT interrupt clear register		RW	0x00000000	MODE_CNT interrupt clear
0x40050070	Clock status register		RW	0x00000007	Clock status
0x40050074	DMON monitor register		RW	0x00000000	DMON monitor control
0x40050094	TEST setting register 2		RW	0x00000000	For clock output
0x40050130	IO setting register 0		RW	0x00000101	IO attribute forced setting
0x40050134	IO setting register 1		RW	0x00000181	IO attribute forced setting
0x40050138	IO setting register 2		RW	0x00000081	IO attribute forced setting
0x4005013C	IO setting register 3		RW	0x00000081	IO attribute forced setting
0x40050140	IO setting register 4		RW	0x00000081	IO attribute forced setting
0x40050144	IO setting register 5		RW	0x00000181	IO attribute forced setting
0x40050148	IO setting register 6		RW	0x00000181	IO attribute forced setting
0x4005014C	IO setting register 7		RW	0x00000181	IO attribute forced setting
0x40050150	IO setting register 8		RW	0x00000181	IO attribute forced setting

0x40050154	IO setting register 9		RW	0x000000C0	IO attribute forced setting
0x40050158	IO setting register 10		RW	0x00000181	IO attribute forced setting
0x4005015C	IO setting register 11		RW	0x000000C0	IO attribute forced setting
0x40050160	IO setting register 12		RW	0x000000C0	IO attribute forced setting
0x40050164	IO setting register 13		RW	0x00000281	IO attribute forced setting
0x40050168	IO setting register 14		RW	0x00000281	IO attribute forced setting
0x4005016C	IO setting register 15		RW	0x00000281	IO attribute forced setting
0x40050170	IO setting register 16		RW	0x00000281	IO attribute forced setting
0x40050174	IO setting register 17		RW	0x00000281	IO attribute forced setting
0x40050178	IO setting register 18		RW	0x00000281	IO attribute forced setting
0x4005017C	IO setting register 19		RW	0x00000281	IO attribute forced setting
0x40050180	IO setting register 20		RW	0x00000281	IO attribute forced setting
0x40050184	IO setting register 21		RW	0x00000281	IO attribute forced setting
0x40050188	IO setting register 22		RW	0x00000281	IO attribute forced setting
0x4005018C	IO setting register 23		RW	0x00000281	IO attribute forced setting
0x40050190	IO setting register 24		RW	0x00000281	IO attribute forced setting
0x40050194	IO setting register 25		RW	0x00000281	IO attribute forced setting

7.29.3 Description of Registers

7.29.3.1 WDT setting register: 0x40050000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															WDT_RST_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

WDT_RST_EN	Enables the reset request from WDT. 0: disable 1: enable

7.29.3.2 Timer setting register: 0x40050004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																														CASCADE_EN1	CASCADE_EN0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

CASCADE_EN1	Sets TIMERD and TIMERE to the cascade mode. 0: disable 1: enable
CASCADE_EN0	Sets TIMERB and TIMERC to the cascade mode. 0: disable 1: enable

7.29.3.3 Reserved register: 0x40050008

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																														~~FLASH_BANK_EN~~	Reserved
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Reserved	Reserved

7.29.3.4 PLL setting register 0: 0x4005000C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																TEG_MODE	PLL_PDN0	PLL_BYPASS0	PLL_DVCO0											PLL_DREF0	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1	0	1	0	1	1	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

TEG_MODE	TEG setting 0: Commercial mode 1: TEG mode * Normally use the initial value (0).
PLL_PDN0	PLL0 power down setting 0: Power-down operation 1: Operation mode
PLL_BYPASS0	PLL0 bypass setting 0: Operation mode (non-bypass) 1: IREF/PLLOUT bypass * Normally use the initial value (0).

PLL_DVCO0	PLL0 counter output multiplying setting 00000000001: 1 multiplying 00000000010: 2 multiplying 00000000011: 3 multiplying 00000000100: 4 multiplying ••• 11111111110: 2046 multiplying 11111111111: 2047 multiplying 00000000000: 2048 multiplying
PLL_DREF0	PLL0 counter input dividing setting 01: 1 dividing 10: 2 dividing 11: 3 dividing 00: 4 dividing

The PLL output frequency is obtained from the formula below using PLL_DVCO0, PLL_DREF0, and PLL_DIV (7.29.3.45 0x400500C8).

fout = (fin x PLL_DVCO0 ÷ PLL_DREF0) ÷ PLL_DIV

For an input of XTAL32KHz, PLL_DREF0 = 0x01 (1 dividing) is recommended. If 0x01 is not set, jitter may increase.

Below is an example of setting the PLL output frequency.

[Register initial value]

PLL_DVCO0 = 1238 multiplying

, PLL_DREF0 = 1 dividing, PLL_DIV = 1 dividing

PLL_IREF_SEL = XTAL32kHz

PLL output = (32.768 KHz x 1238 multiplying ÷ 1 dividing) ÷ 1 dividing = 40.567 MHz

[40 MHz output]

PLL_DVCO0 = 1221 multiplying, PLL_DREF0 = 1 dividing, PLL_DIV = 1 dividing

PLL_IREF_SEL = XTAL32KHz

PLL output = (32.768 KHz x 1221 multiplying ÷ 1 dividing) ÷ 1 dividing = 40.009 MHz

[5 MHz output]

PLL_DVCO0 = 1221 multiplying, PLL_DREF0 = 1 dividing, PLL_DIV = 8 dividing

PLL_IREF_SEL = XTAL32KHz

PLL output = (32.768 KHz x 1221 multiplying ÷ 1 dividing) ÷ 8 dividing = 5.001 MHz

7.29.3.5 PLL setting register 1: 0x40050010

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																														PLL_IREF_SEL	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

PLL_IREF_SEL	PLL reference clock selection 00: XTAL32kHz 01: Low-speed CR 10:RF(ML7396B) 11: High-speed CR

7.29.3.6 CLKGEN setting register 0: 0x40050018

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																											SUB_SRC_SEL	Reserved	MAIN_SRC_SEL		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	R	RW	RW	RW

MAIN_SRC_SEL	Main clock source selection 000: High-speed CR 001: PLL 010: RF(ML7396B) 011: XTAL32kHz 100: Low-speed CR Other: Setting prohibited
SUB_SRC_SEL	Sub clock source selection 0: XTAL32kHz 1: Low-speed CR

7.29.3.7 CLKGEN setting register 1: 0x4005001C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																									EXTTIMER_CLK_SEL	TIMERE_CLK_SEL	TIMERD_CLK_SEL	TIMERC_CLK_SEL	TIMERB_CLK_SEL	TIMERA_CLK_SEL	WDT_CLK_SEL
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW

WDT_CLK_SEL	WDT clock source selection 0: Main clock 1: Sub clock
TIMERA_CLK_SEL	TIMERA clock source selection 0: Main clock 1: Sub clock
TIMERB_CLK_SEL	TIMERB clock source selection 0: Main clock 1: Sub clock

TIMERC_CLK_SEL	TIMERC clock source selection 0: Main clock 1: Sub clock
TIMERD_CLK_SEL	TIMERD clock source selection 0: Main clock 1: Sub clock
TIMERE_CLK_SEL	TIMERE clock source selection 0: Main clock 1: Sub clock
EXT_TIMER_CLK_SEL	EXT_TIMER clock source selection 0: Main clock 1: Sub clock

7.29.3.8 CLKGEN setting register 2: 0x40050020

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												CLKT_L_SEL		CLKT_H_SEL	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

CLKT_H_SEL	High-speed clock for CLK_TIMER selection 00: PLL 01: RF 10: High-speed CR Other: Setting prohibited
CLKT_L_SEL	Low-speed clock for CLK_TIMER selection 00: Low-speed CR 01: XTAL32kHz 10: RF Other: Setting prohibited

7.29.3.9 CLKGEN setting register 3: 0x40050024

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				RAND_DIV				WDT_DIV				GPIO_DIV				SSIS_DIV				UART_DIV				I2C_DIV				FCLK_DIV			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

RAND_DIV	Dividing setting 0000: 1 dividing 0001: 2 dividing 0010: 4 dividing 0011: 8 dividing 0100: 16 dividing 0101: 32 dividing 0110: 64 dividing
WDT_DIV	Dividing setting 0000: 1 dividing 0001: 2 dividing 0010: 4 dividing 0011: 8 dividing 0100: 16 dividing 0101: 32 dividing 0110: 64 dividing * When the WDT clock source is set to sub clock (CLKGEN setting register 1: 0x4005001C), up to 16 dividing can be set. When the WDT clock source is set to sub clock with 0101 or 0110, the clock operates at 1 dividing.

GPIO_DIV	Dividing setting 0000: 1 dividing 0001: 2 dividing 0010: 4 dividing 0011: 8 dividing 0100: 16 dividing 0101: 32 dividing 0110: 64 dividing
SSIS_DIV	Dividing setting 0000: 1 dividing 0001: 2 dividing 0010: 4 dividing 0011: 8 dividing 0100: 16 dividing 0101: 32 dividing 0110: 64 dividing
UART_DIV	Dividing setting 0000: 1 dividing 0001: 2 dividing 0010: 4 dividing 0011: 8 dividing 0100: 16 dividing 0101: 32 dividing 0110: 64 dividing
I2C_DIV	Dividing setting 0000: 1 dividing 0001: 2 dividing 0010: 4 dividing 0011: 8 dividing 0100: 16 dividing 0101: 32 dividing 0110: 64 dividing

FCLK_DIV	Dividing setting 0000: 1 dividing 0001: 2 dividing 0010: 4 dividing 0011: 8 dividing 0100: 16 dividing 0101: 32 dividing 0110: 64 dividing

7.29.3.10 CLKGEN setting register 4: 0x40050028

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	SUB_CLK_DIV				Reserved				EXTTMR_DIV				TMRE_DIV				TMRD_DIV				TMRC_DIV				TMRB_DIV				TMRA_DIV			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

SUB_CLK_DIV	Dividing setting 0000: 1 dividing 0001: 2 dividing 0010: 4 dividing 0011: 8 dividing 0100: 16 dividing
EXTTMR_DIV	Dividing setting 0000: 1 dividing 0001: 2 dividing 0010: 4 dividing 0011: 8 dividing 0100: 16 dividing 0101: 32 dividing 0110: 64 dividing * When the EXT_TIMER clock source is set to sub clock (CLKGEN setting register 1: 0x4005001C), up to 16 dividing can be set. When the EXT_TIMER clock source is set to sub clock with 0101 or 0110, the clock operates at 1 dividing.

TMRE_DIV	Dividing setting 0000: 1 dividing 0001: 2 dividing 0010: 4 dividing 0011: 8 dividing 0100: 16 dividing 0101: 32 dividing 0110: 64 dividing * When the TIMERE clock source is set to sub clock (CLKGEN setting register 1: 0x4005001C), up to 16 dividing can be set. When the TIMERE clock source is set to sub clock with 0101 or 0110, the clock operates at 1 dividing.
TMRD_DIV	Dividing setting 0000: 1 dividing 0001: 2 dividing 0010: 4 dividing 0011: 8 dividing 0100: 16 dividing 0101: 32 dividing 0110: 64 dividing * When the TIMERD clock source is set to sub clock (CLKGEN setting register 1: 0x4005001C), up to 16 dividing can be set. When the TIMERD clock source is set to sub clock with 0101 or 0110, the clock operates at 1 dividing.
TMRC_DIV	Dividing setting 0000: 1 dividing 0001: 2 dividing 0010: 4 dividing 0011: 8 dividing 0100: 16 dividing * When the TIMERC clock source is set to sub clock (CLKGEN setting register 1: 0x4005001C), up to 16 dividing can be set. When the TIMERC clock source is set to sub clock with 0101 or 0110, the clock operates at 1 dividing.

TMRB_DIV	Dividing setting 0000: 1 dividing 0001: 2 dividing 0010: 4 dividing 0011: 8 dividing 0100: 16 dividing 0101: 32 dividing 0110: 64 dividing * When the TIMERB clock source is set to sub clock (CLKGEN setting register 1: 0x4005001C), up to 16 dividing can be set. When the TIMERB clock source is set to sub clock with 0101 or 0110, the clock operates at 1 dividing.
TMRA_DIV	Dividing setting 0000: 1 dividing 0001: 2 dividing 0010: 4 dividing 0011: 8 dividing 0100: 16 dividing 0101: 32 dividing 0110: 64 dividing * When the TIMERA clock source is set to sub clock (CLKGEN setting register 1: 0x4005001C), up to 16 dividing can be set. When the TIMERA clock source is set to sub clock with 0101 or 0110, the clock operates at 1 dividing.

7.29.3.11 CLKGEN setting register 5: 0x4005002C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved												LVD_DIV				Reserved			ADC_DIV	Reserved											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	R	R	R	RW	R	R	R	R	R	R	R	R	R	R	R	R

LVD_DIV	Dividing setting 0000: 1 dividing 0001: 2 dividing 0010: 4 dividing 0011: 8 dividing 0100: 16 dividing
ADC_DIV	Dividing setting 0: 16 dividing (default) 1: 1 dividing

7.29.3.12 CLKGEN setting register 6: 0x40050030

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																											FCNTCLK_DIV				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

FCNTCLK_DIV	Flash control timing generation clock (FCNTCLK) dividing setting FCNTCLK = Main clock/FCNTCLK_DIV When set to 0, 1 dividing (no dividing) is set. Adjust this register to set FCNTCLK to 2 MHz.

FCNTCLK_DIV	Dividing value (DEC)
00000	1
00001	1
00010	2
00011	3
00100	4
00101	5
00110	6
00111	7
01000	8
01001	9
01010	10
01011	11
01100	12
01101	13
01110	14
01111	15
10000	16
10001	17
10010	18
10011	19
10100	20
10101	21
10110	22
10111	23
11000	24
11001	25
11010	26
11011	27
11100	28
11101	29
11110	30
11111	31

7.29.3.13 CLKGEN setting register 7: 0x40050034

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																											STCLK_DIV				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

STCLK_DIV	SysTick timer clock dividing setting SysTick timer clock = Main clock/STCLK_DIV/2 When set to 0, 2 dividing is set. Adjust this register to set the SysTick timer clock to 1 MHz.

STCLK_DIV	Dividing value (DEC)
00000	2
00001	2
00010	4
00011	6
00100	8
00101	10
00110	12
00111	14
01000	16
01001	18
01010	20
01011	22
01100	24
01101	26
01110	28
01111	30
10000	32
10001	34
10010	36
10011	38
10100	40
10101	42
10110	44
10111	46
11000	48
11001	50
11010	52
11011	54
11100	56
11101	58
11110	60
11111	62

7.29.3.14 CLKGEN setting register 8: 0x40050038

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																													CR40M_ON	CR32K_ON	XTAL32K_ON
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

CR40M_ON	High-speed CR operation enable 0: Disabled 1: Enabled
CR32K_ON	Low-speed CR operation enable 0: Disabled 1: Enabled
XTAL32K_ON	XTAL32kHz operation enable 0: Disabled 1: Enabled

7.29.3.15 CLKGEN setting register 9: 0x4005003C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																							PLL_WTCR			XTAL32K_WTCR		CR32K_WTCR		CR40M_WTCR	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW

PLL_WTCR	PLL stabilization wait time adjustment 111: 18000 pll reference clock cycles 110: 9000 pll reference clock cycles 101: 4500 pll reference clock cycles 100: 2250 pll reference clock cycles 011: 100 pll reference clock cycles 010: 50 pll reference clock cycles 001: 25 pll reference clock cycles (default) 000: 20 pll reference clock cycles PLL stabilization wait time = PLL reference clock cycle x PLL_WTCR setting cycle count
XTAL32K_WTCR	XTAL32kHz clock stabilization wait time adjustment 11:100 ms 10:31.25 ms 01:3.125 ms(default) 00:312.5 us
CR32K_WTCR	Low-speed CR clock stabilization wait time adjustment 11: 30 low-speed CR clock cycles 10: 20 low-speed CR clock cycles 01: 10 low-speed CR clock cycles (default) 00: 5 low-speed CR clock cycles Low-speed CR clock stabilization wait time = Low-speed CR clock cycle x CR32K_WTCR setting cycle count

CR40M_WTCR	High-speed CR clock stabilization wait time adjustment 11: 4000 high-speed CR clock cycles 10: 1200 high-speed CR clock cycles (default) 01: 50 high-speed CR clock cycles 00: 5 high-speed CR clock cycles High-speed CR clock stabilization wait time = High-speed CR clock cycle x CR40M_WTCR setting cycle count

* The stabilization wait time does not include the time before the oscillation circuit starts the clock output.

7.29.3.16 Deep sleep control register: 0x40050040

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																			RF_PSO	LOGIC_PSO	FLASH_PSO	SRAM2_PSO	SRAM1_PSO	Reserved								SUBCLK_STP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	RW

SUBCLK_STP	32 kHz oscillation circuit stop (SLEEP) enable (XTAL32kHz is set to OFF during DEEPSLEEP state) 0: Disabled (= 32 kHz oscillation circuit ON) 1: Enabled (= 32 kHz oscillation circuit OFF)
SRAM1_PS0	SRAM1 power OFF (SLEEP) enable (SRAM1 is powered OFF during DEEPSLEEP state) 0: Disabled (= SRAM1 is powered ON) 1: Enabled (= SRAM1 is powered OFF)
SRAM2_PS0	SRAM2 power OFF (SLEEP) enable (SRAM2 is powered OFF during DEEPSLEEP state) 0: Disabled (= SRAM2 is powered ON) 1: Enabled (= SRAM2 is powered OFF)
FLASH_PS0	FLASH power OFF (SLEEP) enable (FLASH is powered OFF during DEEPSLEEP state) 0: Disabled (= FLASH is powered ON) 1: Enabled (= FLASH is powered OFF) * When this bit is set to 1, ensure that LOGIC_PS0 bit is set to 1 at the same time.

LOGIC_PS0	LOGIC power OFF (SLEEP) enable (Some LOGIC is powered OFF during DEEPSLEEP state) 0: Disabled (= Some LOGIC is powered ON) 1: Enabled (= Some LOGIC is powered OFF) * When FLASH_PS0 bit is set to 1, ensure that this bit is set to 1 at the same time.
RF_PS0	RF power OFF (SLEEP) enable (RF is powered OFF during DEEPSLEEP state) 0: Disabled (= RF is powered ON) 1: Enabled (= RF is powered OFF)

7.29.3.17 FLASH-ROM deep standby control register: 0x40050044

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												FLASH11_DPSTB		FLASH10_DPSTB		FLASH01_DPSTB		FLASH00_DPSTB	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

FLASH00_DPSTB	FLASH00 deep standby setting 1: Deep standby 0: Operable
FLASH01_DPSTB	FLASH01 deep standby setting 1: Deep standby 0: Operable
FLASH10_DPSTB	FLASH10 deep standby setting 1: Deep standby 0: Operable
FLASH11_DPSTB	FLASH11 deep standby setting 1: Deep standby 0: Operable

When returning from deep standby, the read value of the corresponding bit changes to "0" when the return time (10 μs) of FLASH has

elapsed after FLASHn_DPSTB was set to "0" from "1". Confirm that the read value of the target bit has changed to "0" before accessing FLASH that was switched to deep standby mode by this register.

This register controls the FLASH-ROM operation modes independent of the LSI operation modes (Active/SLEEP/DEEPSLEEP).
When only some FLASH-ROM areas are used, the register can reduce consumption current by putting unused areas into the deep standby state. The table below lists the operation modes and the FLASHn operation modes appropriate for register settings.

Operation mode	**Register setting**	**FLASHn operation mode**	**Current consumed by FLASHn (Typ. reference value)**
Active	FLASHn_DPSTB=0 (0x40050044)	Code execution area: Active Other areas: Standby	1.5 mA@40 MHz 200 uA
	FLASHn_DPSTB=1 (0x40050044)	Deep standby	3 uA
Sleep	FLASHn_DPSTB=0 (0x40050044)	Standby	200 uA
	FLASHn_DPSTB=1 (0x40050044)	Deep standby	3 uA
DeepSleep	FLASH_PSO=0 (0x40050040)	Deep standby	3 uA
	FLASH_PSO=1 (0x40050040)	Shutdown	0 uA

* n = 00, 01, 10, 11

7.29.3.18 FLASH WakeUp time setting register: 0x40050048

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																											FLC_STARTW				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

FLC_STARTW	FLASH WakeUp time setting Wakeup time = Setting value x 800 ns (default 11.2 µs)

7.29.3.20 MODE_CNT interrupt mask register: 0x40050050

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												PLL_DONE_MSK	XTAL32K_DONE_MSK	CR32K_DONE_MSK	CR40M_DONE_MSK
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

PLL_DONE_MSK	Masks the PLL stabilization completion interrupt. 1: Mask 0: No mask
XTAL32K_DONE_MSK	Masks the XTAL32kHz clock stabilization completion interrupt. 1: Mask 0: No mask
CR32K_DONE_MSK	Masks the low-speed CR clock stabilization completion interrupt. 1: Mask 0: No mask

CR40M_DONE_MSK	Masks the high-speed CR clock stabilization completion interrupt. 1: Mask 0: No mask

7.29.3.21 MODE_CNT interrupt status register: 0x40050054

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												PLL_DONE	XTAL32K_DONE	CR32K_DONE	CR40M_DONE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

PLL _DONE	PLL stabilization completion flag after masking 0: During start or stop 1: Stabilized operation
XTAL32K _DONE	XTAL32kHz stabilization completion flag after masking 0: During start or stop 1: Stabilized operation
CR32K _DONE	Low-speed CR stabilization completion flag after masking 0: During start or stop 1: Stabilized operation
CR40M _DONE	High-speed CR stabilization completion flag after masking 0: During start or stop 1: Stabilized operation

7.29.3.22 MODE_CNT RAW interrupt status register: 0x40050058

	31–04	03	02	01	00
	Reserved	PLL_DONE_RAW	XTAL32K_DONE_RAW	CR32K_DONE_RAW	CR40M_DONE_RAW
Initial value	0	0	0	0	0
R/W	R	R	R	R	R

PLL _DONE_RAW	PLL stabilization completion flag before masking 0: During start or stop 1: Stabilized operation
XTAL32K _DONE_RAW	XTAL32kHz stabilization completion flag before masking 0: During start or stop 1: Stabilized operation
CR32K _DONE_RAW	Low-speed CR stabilization completion flag before masking 0: During start or stop 1: Stabilized operation

CR40M _DONE_RAW	High-speed CR stabilization completion flag before masking 0: During start or stop 1: Stabilized operation

7.29.3.23 High-speed CR stabilization completion interrupt clear register: 0x4005005C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															CR40M_DONE_CLR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

CR40M_DONE_CLR	Clears the high-speed CR stabilization completion interrupt. * When this register is read, the interrupt is cleared. Always 0 is read when reading.

7.29.3.24 Low-speed CR stabilization completion interrupt clear register: 0x40050060

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															CR32K_DONE_CLR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

CR32K_DONE_CLR	Clears the low-speed CR stabilization completion interrupt. * When this register is read, the interrupt is cleared. Always 0 is read when reading.

7.29.3.25 XTAL32K stabilization completion interrupt clear register: 0x40050064

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															XTAL32K_DONE_CLR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

XTAL32K_DONE_CLR	Clears the XTAL32K stabilization completion interrupt. * When this register is read, the interrupt is cleared. Always 0 is read when reading.

7.29.3.26 PLL start completion interrupt clear register: 0x40050068

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															PLL_DONE_CLR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

PLL_DONE_CLR	Clears the PLL stabilization completion interrupt. * When this register is read, the interrupt is cleared. Always 0 is read when reading.

7.29.3.27 MODE_CNT interrupt clear register: 0x4005006C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																															MODE_CNT_CLR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

MODE_CNT_CLR	Clears all interrupts of MODE_CNT. * When this register is read, the interrupt is cleared. Always 0 is read when reading.

7.29.3.28 Clock status register: 0x40050070

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												PLL_DONE_FLG	XTAL32K_DONE_FLG	CR32K_DONE_FLG	CR40M_DONE_FLG
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1*	1*	1*
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

(*) The initial value of hardware is 0.

PLL_DONE_FLG	PLL stabilization completion flag 0: During start or stop 1: Stabilized operation
XTAL32K_DONE_FLG	XTAL32kHz stabilization completion flag 0: During start or stop 1: Stabilized operation
CR32K_DONE_FLG	Low-speed CR stabilization completion flag 0: During start or stop 1: Stabilized operation

CR40M_DONE_FLG	High-speed CR stabilization completion flag 0: During start or stop 1: Stabilized operation

7.29.3.29 DMON Monitor Register: 0x40050074

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															DMON_CPU
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

DMON_CPU	DMON_CPU pin monitor

7.29.3.34 TMP Setting Register 0: 0x40050088

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																Reserved							TEMP_EN_OVR	Reserved							TEMP_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	R	R	R	R	R	R	R	RW	R	R	R	R	R	R	R	RW

TEMP_EN_OVR	Enables (overwrites) TEMP_EN. 0: disable 1: enable
TEMP_EN	Thermometer enable 0: disable 1: enable

7.29.3.37 TEST Setting Register 2: 0x40050094

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																											MON_CLK_SEL		MON_CLK_DIV		MON_CLK_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

MON_CLK_SEL	Selects the clock for the monitor output from GPIOA1. 00: XTAL32K 01: OSC32K 10: OSC40M 11: PLL
MON_CLK_DIV	Divides the clock for the monitor output from GPIOA1. 00: 8 dividing 01: 4 dividing 10: 2 dividing 11: no division (through)

MON_CLK_EN	Enables the clock for the monitor output from GPIOA1. 0: Disabled 1: Enabled

7.29.3.45 PLL Setting Register 2: 0x400500C8

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																												PLL_DIV		Reserved	
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW

PLL_DIV	PLL output dividing setting 00: 1 dividing 01: 2 dividing 10: 4 dividing 11: 8 dividing

7.29.3.52 IO setting register 0 to 25: 0x40050130 to 0x40050194

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	Reserved																						set_io_en	set_io[8](IE)	Reserved	set_io[6](OE)	Reserved		set_io[3](PU)	set_io[2](PD)	Reserved	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*	*	*
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

set_io_en	set_io[8] and set_io[6] enable (* This function is disabled for IO setting register 2 [TEST_CPU pin]) For details, see the generation logic of each pin control signal.
set_io[8](IE)	Input buffer enable 0: disable 1: enable
set_io[6](OE)	Output buffer enable 0: disable 1: enable
set_io[3](PU)	Weak pull-up resistor enable 0: disable 1: enable

set_io[2](PD)	Weak pull-down resistor enable 0: disable 1: enable

* The reserved bits of this register should be used with the initial values.

This register controls the following IO pins with their initial values.

Register name	Target pin	Address	Initial value
IO setting register 0	SWCK	0x40050130	0x00000101
IO setting register 1	SWD	0x40050134	0x00000181
IO setting register 2	TEST_CPU	0x40050138	0x00000081
IO setting register 3	MODE0	0x4005013C	0x00000081
IO setting register 4	MODE1	0x40050140	0x00000081
IO setting register 5	SINTN_CPU (LSI internal pin)	0x40050144	0x00000181
IO setting register 6	DMON_CPU (LSI internal pin)	0x40050148	0x00000181
IO setting register 7	DCLK_CPU (LSI internal pin)	0x4005014C	0x00000181
IO setting register 8	DIO_CPU (LSI internal pin)	0x40050150	0x00000181
IO setting register 9	SDI_CPU (LSI internal pin)	0x40050154	0x000000C0
IO setting register 10	SDO_CPU (LSI internal pin)	0x40050158	0x00000181
IO setting register 11	SCEN_CPU (LSI internal pin)	0x4005015C	0x000000C0
IO setting register 12	SCLK_CPU (LSI internal pin)	0x40050160	0x000000C0
IO setting register 13	GPIOA0	0x40050164	0x00000281
IO setting register 14	GPIOA1	0x40050168	0x00000281
IO setting register 15	GPIOA2	0x4005016C	0x00000281
IO setting register 16	GPIOA3	0x40050170	0x00000281
IO setting register 17	GPIOA4	0x40050174	0x00000281
IO setting register 18	GPIOA5	0x40050178	0x00000281
IO setting register 19	GPIOA6	0x4005017C	0x00000281
IO setting register 20	GPIOA7	0x40050180	0x00000281
IO setting register 21	GPIOA8	0x40050184	0x00000281
IO setting register 22	GPIOA9	0x40050188	0x00000281
IO setting register 23	GPIOA10	0x4005018C	0x00000281
IO setting register 24	GPIOA11	0x40050190	0x00000281
IO setting register 25	GPIOA12	0x40050194	0x00000281

This register controls the following IO pins with their control signal generation logics.

Pin name: GPIOAn (n=0-12)

	set_io_en=1	Other
IE	set_io[8]	Conform to 7.11.5 General-purpose Port Setting
OE	set_io[6]	Conform to 7.11.5 General-purpose Port Setting
PU	set_io[3]	set_io[3]
PD	set_io[2]	set_io[2]

Pin name: SWCK/SWD/TEST_CPU/MODE0/MODE1

	set_io_en=1	Other
IE	set_io[8]	Conform to 7.11.5 General-purpose Port Setting
OE	set_io[6]	Conform to 7.11.5 General-purpose Port Setting
PU	set_io[3]	set_io[3]
PD	set_io[2]	set_io[2]

Pin name: SINTN_CPU

	set_io_en=1	set_io_en=0	
		During DeepSleep and RF_PSO=1	All conditions except the left
IE	set_io[8]	set_io[8]	set_io[8]
OE	set_io[6]	set_io[6]	set_io[6]
PU	set_io[3]	1	set_io[3]
PD	set_io[2]	0	set_io[2]

Pin name: DMON_CPU

	set_io_en=1	set_io_en=0	
		During DeepSleep and RF_PSO=1	All conditions except the left
IE	set_io[8]	set_io[8]	set_io[8]
OE	set_io[6]	set_io[6]	set_io[6]
PU	set_io[3]	0	set_io[3]
PD	set_io[2]	1	set_io[2]

Pin name: DIO_CPU

	set_io_en=1	set_io_en=0		
		During DeepSleep and RF_PSO=1	All conditions except the left	
			PU request from DIO block	All conditions except the left
IE	set_io[8]	set_io[8]	1/0(DIO)	1/0(DIO)
OE	set_io[6]	set_io[6]	1/0(DIO)	1/0(DIO)
PU	set_io[3]	0	1	set_io[3]
PD	set_io[2]	1	0	set_io[2]

Pin name: DCLK_CPU

	set_io_en=1	set_io_en=0	
		During DeepSleep and RF_PSO=1	All conditions except the left
IE	set_io[8]	set_io[8]	set_io[8]
OE	set_io[6]	set_io[6]	set_io[6]
PU	set_io[3]	0	set_io[3]
PD	set_io[2]	1	set_io[2]

Pin name: SCLK_CPU

	set_io_en=1	set_io_en=0
IE	set_io[8]	1/0(SPI2_SCK)
OE	set_io[6]	1/0(SPI2_SCK)
PU	set_io[3]	set_io[3]
PD	set_io[2]	set_io[2]

Pin name: SCEN_CPU

	set_io_en=1	set_io_en=0
IE	set_io[8]	1/0(SPI2_SSN)
OE	set_io[6]	1/0(SPI2_SSN)
PU	set_io[3]	set_io[3]
PD	set_io[2]	set_io[2]

Pin name: SDI_CPU

	set_io_en=1	set_io_en=0
IE	set_io[8]	1/0(SPI2_MOSI)
OE	set_io[6]	1/0(SPI2_MOSI)
PU	set_io[3]	set_io[3]
PD	set_io[2]	set_io[2]

Pin name: SDO_CPU

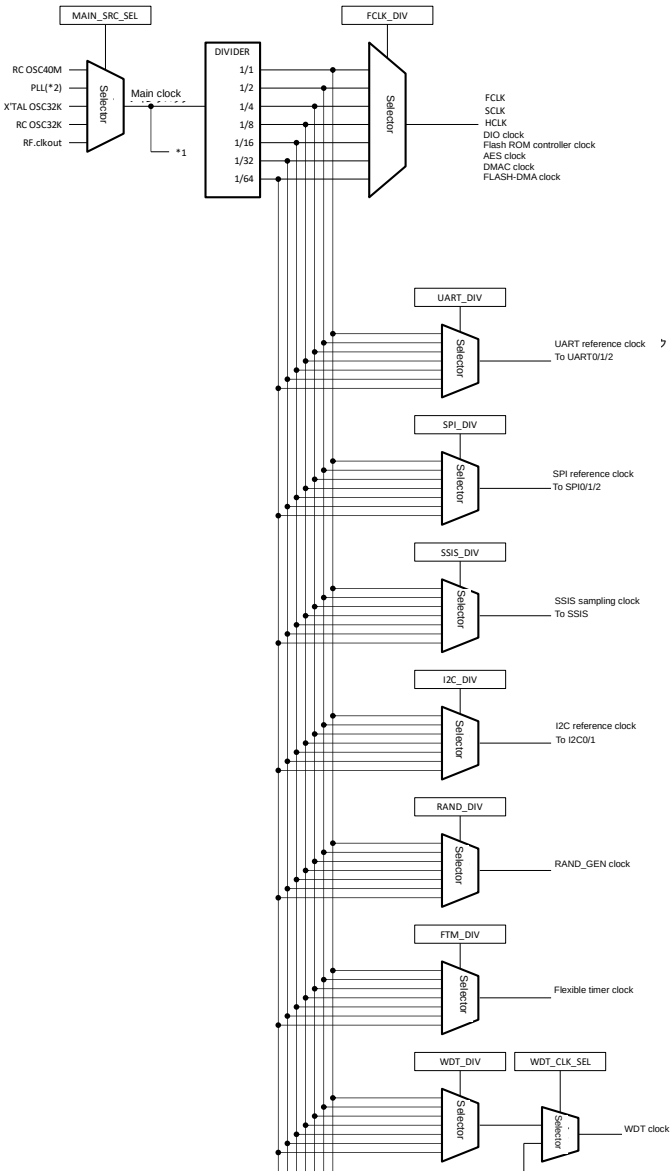
	set_io_en=1	set_io_en=0	
		During DeepSleep and RF_PSO=1	All conditions except the left
IE	set_io[8]	set_io[8]	1/0(SPI2_MISO)
OE	set_io[6]	set_io[6]	1/0(SPI2_MISO)
PU	set_io[3]	0	set_io[3]
PD	set_io[2]	1	set_io[2]

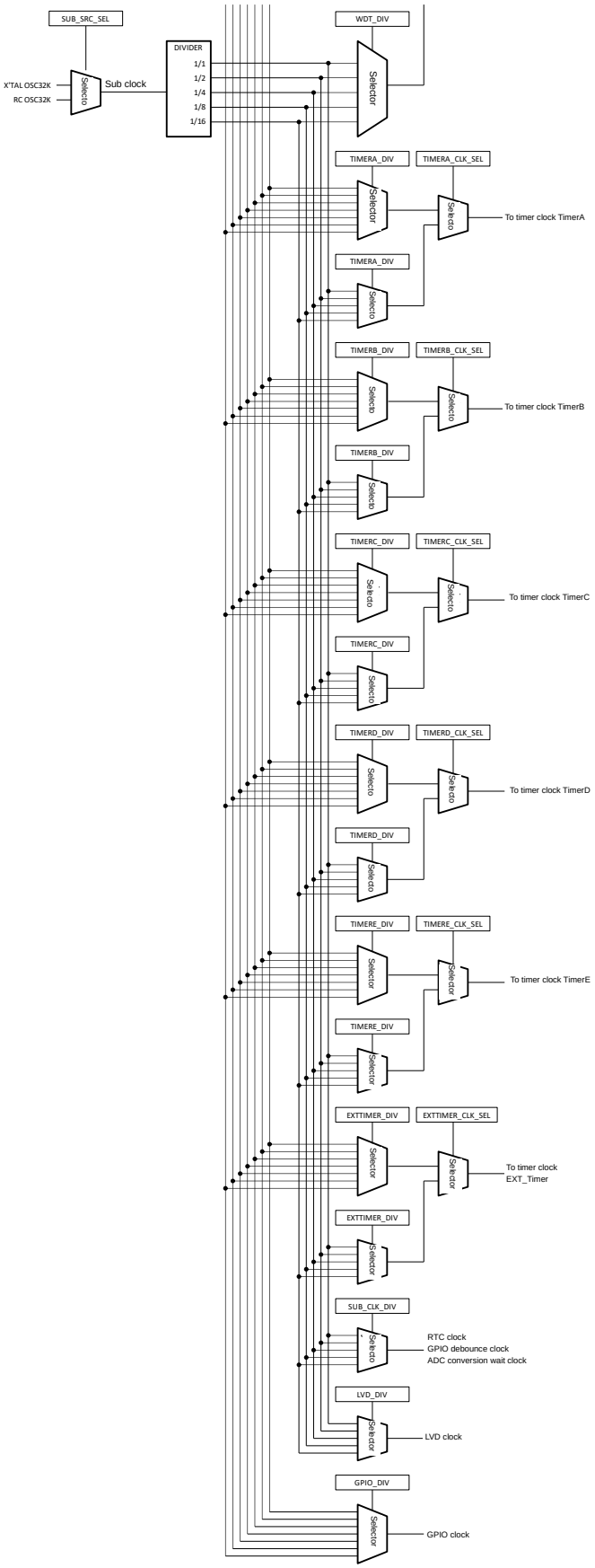
7.29.4 Clock Control

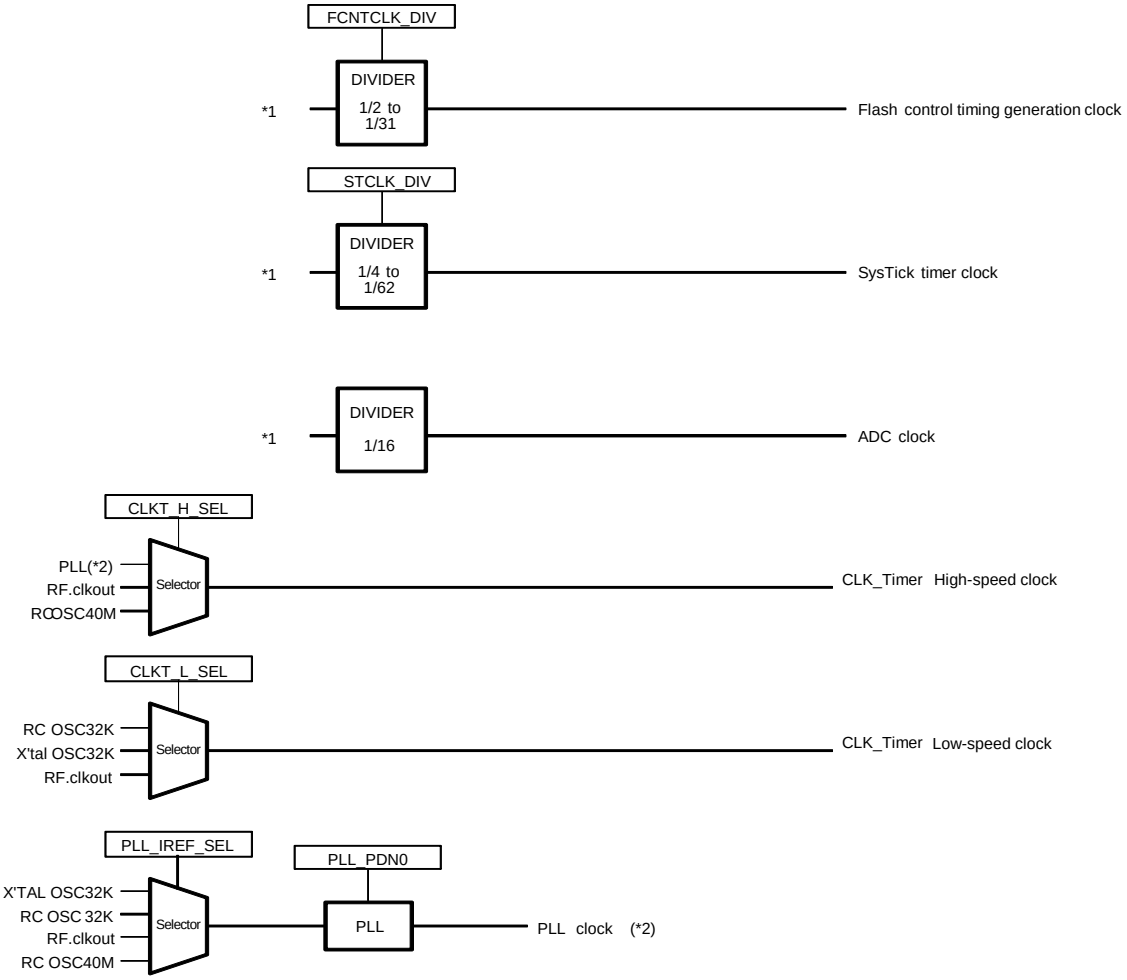
7.29.4.1 Clock System Diagram

Here is the clock system diagram. Register names are indicated in "□" in the diagram.

[Note] 1 to 2048 dividing and 1 to 512 dividing can be set for SPI and FTM respectively.



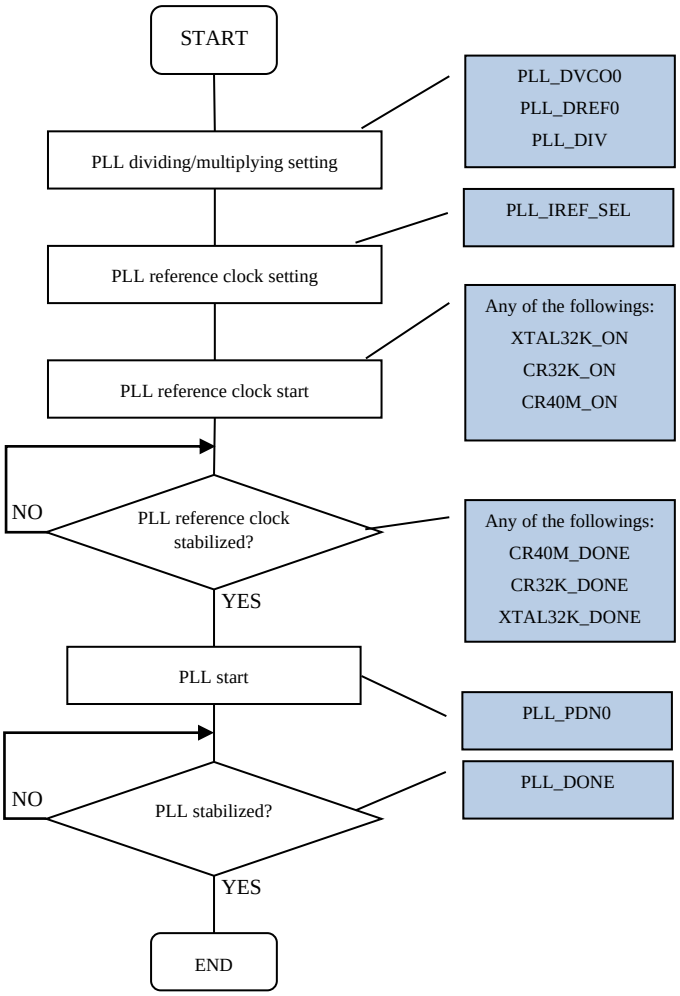

SUB_SRC_SEL
X'TAL OSC32K
RC OSC32K
Selector
Sub clock
DIVIDER
1/1
1/2
1/4
1/8
1/16
WDT_DIV
Selector
TIMERA_DIV
TIMERA_CLK_SEL
To timer clock TimerA
TIMERB_DIV
TIMERB_CLK_SEL
To timer clock TimerB
TIMERC_DIV
TIMERC_CLK_SEL
To timer clock TimerC
TIMERD_DIV
TIMERD_CLK_SEL
To timer clock TimerD
TIMERE_DIV
TIMERE_CLK_SEL
To timer clock TimerE
EXTTIMER_DIV
EXTTIMER_CLK_SEL
To timer clock
EXT_Timer
SUB_CLK_DIV
RTC clock
GPIO debounce clock
ADC conversion wait clock
LVD_DIV
LVD clock
GPIO_DIV
GPIO clock


FCNTCLK_DIV
DIVIDER
1/2 to
1/31
*1
Flash control timing generation clock
STCLK_DIV
DIVIDER
1/4 to
1/62
*1
SysTick timer clock
DIVIDER
1/16
*1
ADC clock
CLKT_H_SEL
PLL(*2)
RF.clkout
RCOSC40M
Selector
CLK_Timer High-speed clock
CLKT_L_SEL
RC OSC32K
X'tal OSC32K
RF.clkout
Selector
CLK_Timer Low-speed clock
PLL_IREF_SEL
X'TAL OSC32K
RC OSC 32K
RF.clkout
RC OSC40M
Selector
PLL_PDN0
PLL
PLL clock (*2)

7.29.4.2 PLL Control

Here is the control flow of PLL.

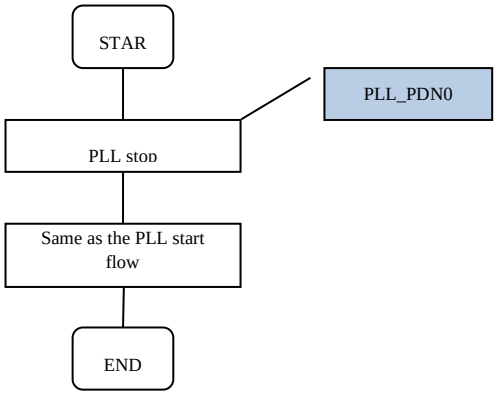
PLL Start Flow



The dividing/multiplying value or reference clock frequency of PLL should be changed after PLL is stopped.

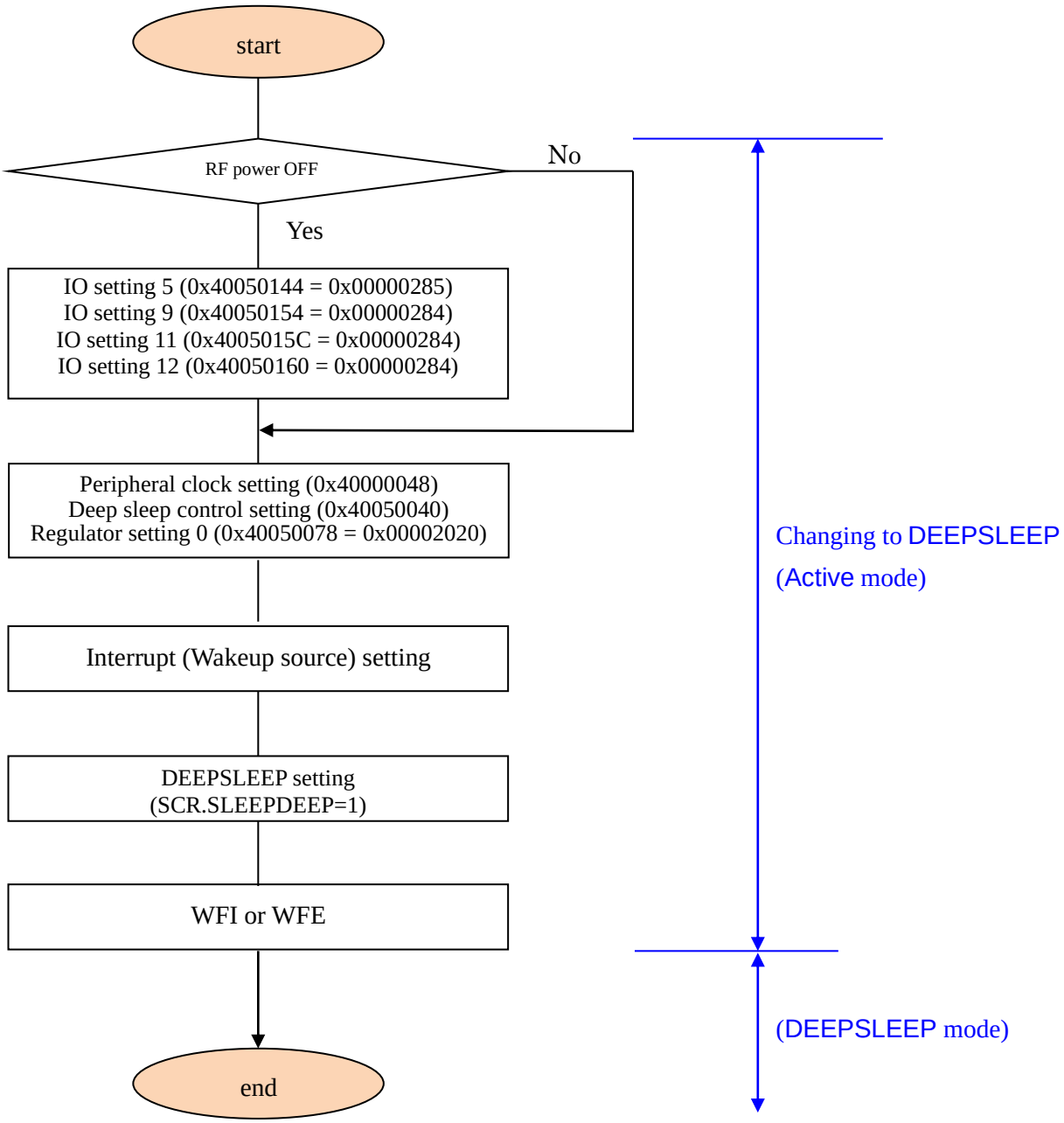
Here is the flow for changing the multiplying/dividing value of PLL.

PLL Setting Change Flow



7.29.4.3 DeepSleep/Sleep Control

■ Procedure for Changing to DEEPSLEEP Mode

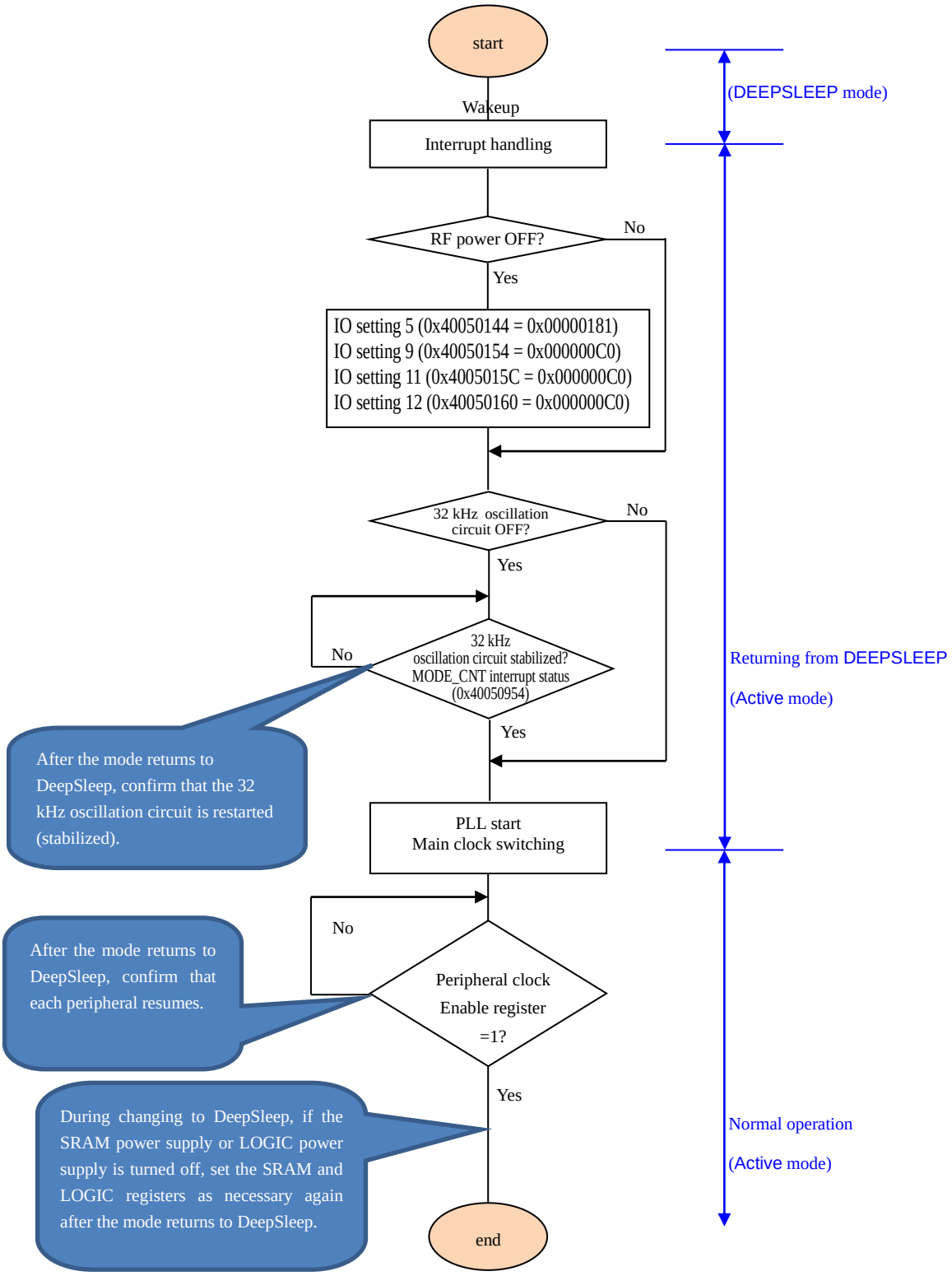


■ Procedure for Returning from DEEPSLEEP Mode

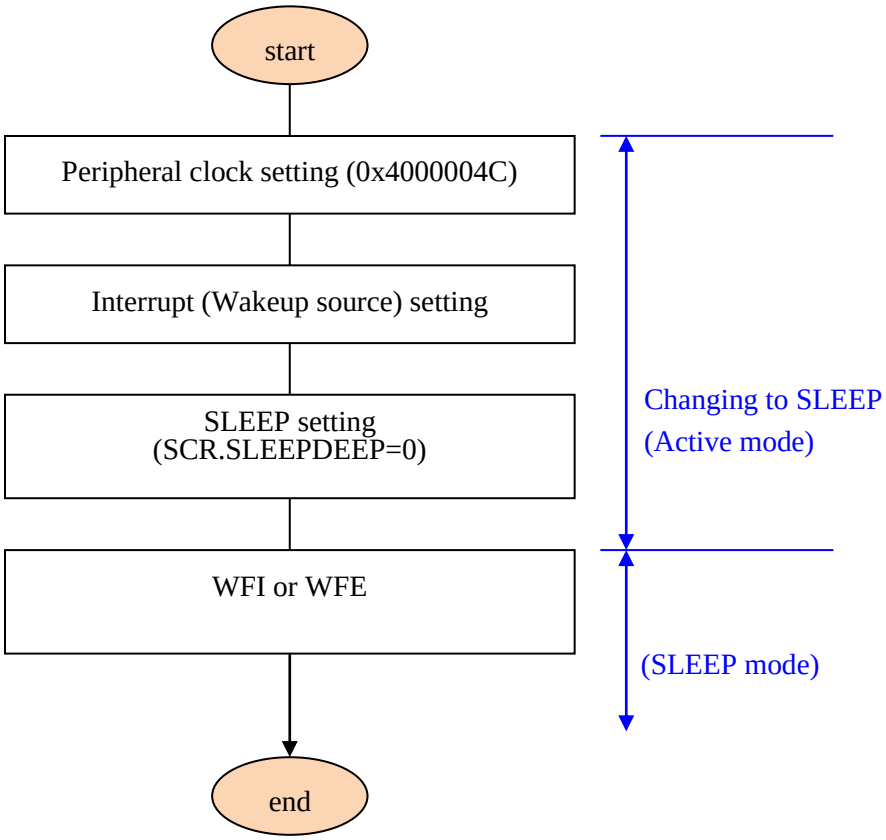
Changing to the DEEPSLEEP resets the following registers for clock control, and it starts in the high-speed CR after returning.

MAIN_SRC_SEL(0x40050018), CR40M_ON(0x40050038), PLL_PDN0(0x4005000C)

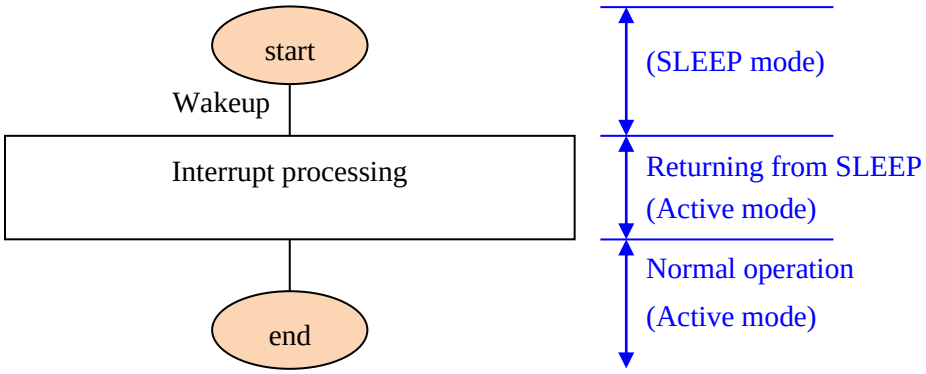
Set the clock again after returning.



■ Procedure for Changing to SLEEP Mode



■ Procedure for Returning from SLEEP Mode



7.30.Micro Trace Buffer(MTB)

7.30.1 General Description

The features of Micro Trace Buffer(MTB) are shown below.

- Provides the program execution trace function of Cortex®-M0+ processor.
- Shares SRAM space as the trace information storage buffer.
- The trace buffer capacity to be used by MTB can be changed by software.

For details of MTB, refer to the following document provided by ARM®:

CoreSight MTB-M0+ Technical Reference Manual

In this product, the SFR area of MTB is assigned to the 4 KB area starting at the following address:

0xF0002000

The entire area of SRAM space starting at address 0x20000000 is used as RAM area of MTB[11] .
By default, the area is 16 KB (AWIDTH = 14).

[11] When the trace function of MTB is enabled, the RAM access from CPU may be held if trace information to be stored to RAM conflicts with RAM access from CPU, degrading CPU performance.

7.31.System ROM Table

7.31.1 General Description

The system ROM table indicates debug components (watch point, break point, trace buffer, etc.) implemented in MCU connected to the debugger.

Normally, the debugger recognizes the addresses stored in the system ROM table by using the Debug Base Address register [12]that exists at the given address within DAP (Debug Access Port). The system ROM table includes the entry that indicates the storage address of Cortex®-M0+ ROM Table and the one that indicates the storage address of Micro Trace Buffer (MTB) ROM Table, to be used to recognize Cortex®-M0+ (and the debug components of Cortex®-M0+) and MTB.

* When MTB is not implemented, the system ROM table is not implemented.

The Debug Base Address register of DAP points the ROM Table of Cortex®-M0+.

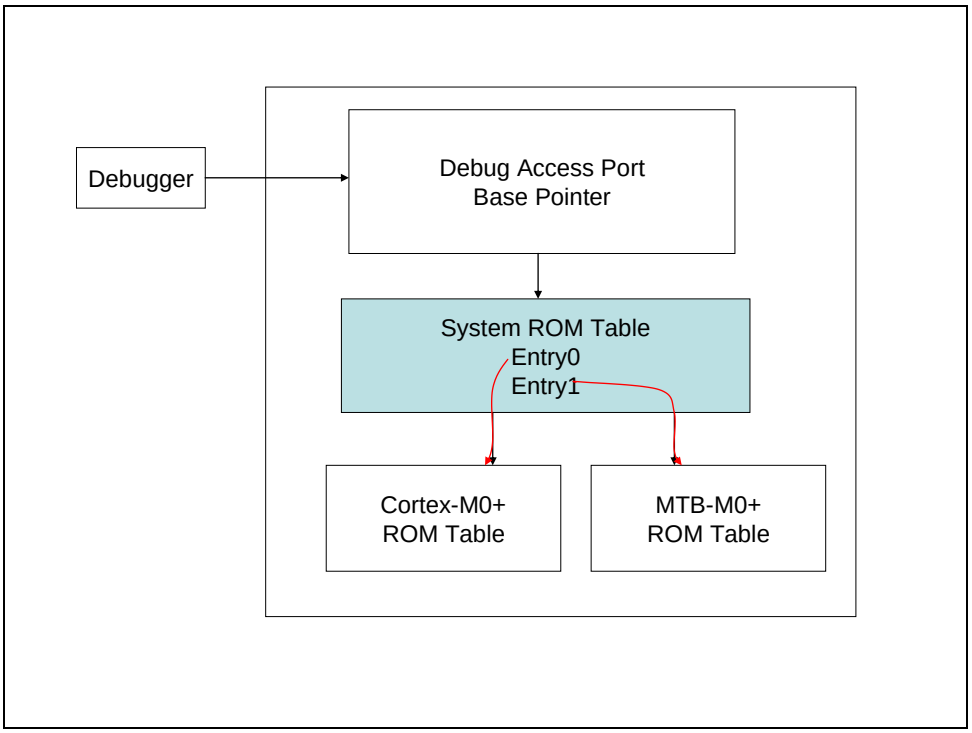


Figure 7-30 Relationship between System ROM Table and Peripheral Functions

[12] MEM-AP: Debug Base Address (BASE) register implemented at address 0xF8

7.31.2 Address Map

This product implements a 32-bit format system ROM table. The system ROM table is placed in 0xF0000000 and higher on the system address map. The details of the system ROM table implemented by this product are shown below.

Offset address	Name	Value	Remarks
0x000	Table Entry0	0xF00FF003	[31:12]: = Offset from the system ROM table to Cortex®-M0+ (0xE00FF000) [11:2]:=Reserved [1]:= Format. Fixed to 1 (32-bit format). [0]:=Entry Present. Fixed to 1
0x004	Table Entry1	0x00002003	[31:12]: = Offset from the system ROM table to MTB-M0+ (0xF0002000) [11:2]:=Reserved [1]:=Format. Fixed to 1 (32-bit format). [0]:= Entry Present. Fixed to 1 (MTB-M0+ implemented).
0x008	-	0x00000000	Bit 0 (Entry Present) is 0, indicating that no more entry exists[13] .
:			
0xFCC	MEMTYPE	0x00000001	[31:1]:=Reserved [0]:=System memory present. Indicates that the system memory exists on the bus to which the ROM table is connected.
0xFD0	Peripheral ID4	0x0000000X	[31:8]:=Reserved [7:4]:= Number of bytes. 0x0 fixed (1 byte). [3:0]:=JEP 106 Continuation Code
0xFD4	Peripheral ID5	0x00000000	[31:0]:=Reserved
0xFD8	Peripheral ID6	0x00000000	[31:0]:=Reserved
0xFDC	Peripheral ID7	0x00000000	[31:0]:=Reserved
0xFE0	Peripheral ID0	0x000000XX	[31:8]:=Reserved [7:0]:=Part Number[7:0]

[13] Bit 0 becomes 0 which indicates that no valid entry exists. Other bits are undefined (implementation-dependent).

0xFE4	Peripheral ID1	0x000000XX	[31:8]:=Reserved [7:4]:=JEP106 Identity Code[3:0] [3:0]:=Part Number[11:8]
0xFE8	Peripheral ID2	0x000000XX	[31:8]:=Reserved [7:4]:= Revision[3:0]. Indicates the revision of this system ROM table. Fixed to 0x0. [3]:= Indicates that JEP 106 Identity Code is used. Fixed to 1. [2:0]:= JEP 106 Identity Code[6:4]
0xFEC	Peripheral ID3	0x000000X0	[31:8]:=Reserved [7:4]:=RevAnd. Normally, this indicates the revision number by ECO. [3:0]:=Customer Modified. Fixed to 0x0.
0xFF0	Component ID0	0x0000000D	[31:8]:=Reserved [7:0] :=Preamble byte0. Fixed to 0x0D.
0xFF4	Component ID1	0x00000010	[31:8]:=Reserved [7:4] :=Component Class. Fixed to 0x1 (ROM Table). [3:0] := Preamble. Fixed to 0x0.
0xFF8	Component ID2	0x00000005	[31:8]:=Reserved [7:0] := Preamble byte 2. Fixed to 0x05.
0xFFC	Component ID3	0x000000B1	[31:8]:=Reserved [7:0] := Preamble byte 3. Fixed to 0xB1.

7.31.3 JEDEC JEP-106

JEDEC JEP-106 Identity Code and Continuation Code are required in the system ROM table. JEP-106 codes in UxPlatform are as follows:

Field	Value	Remarks
JEP-106 Identity Code[6:0]	0x2F	Manufacturer's Identification Code of the Rohm group registered in JEDEC JEP106. It is usually 0x2F. It may be necessary to change this value when deploying a product under another company's brand, such as OEM products. It should comply with the brand policy of LSI.
Continuation Code[3:0]	0x1	Continuation Code of the Rohm group registered in JEDEC JEP106. It is usually 0x1. It may be necessary to change this value when deploying a product under another company's brand, such as OEM products. It should comply with the brand policy of LSI.

7.31.4 Part Number

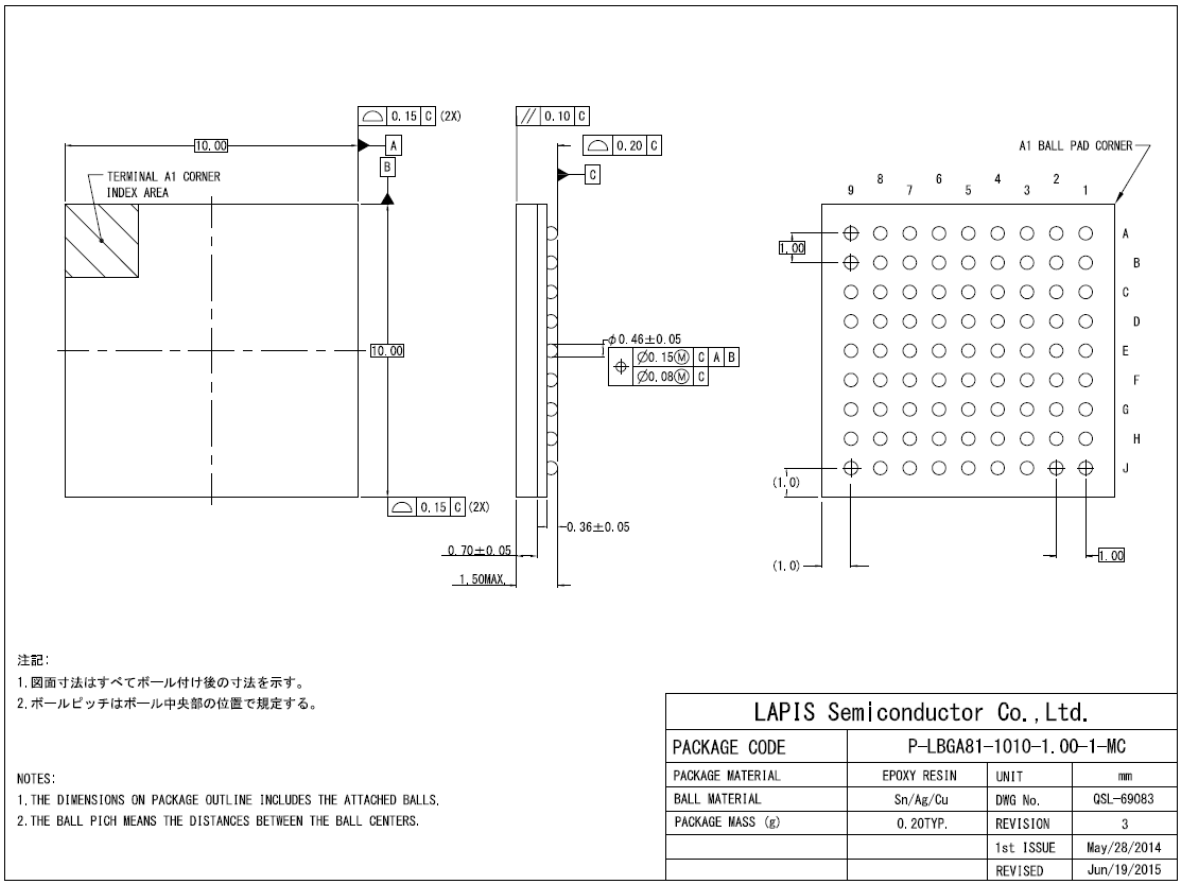
Part Number is a unique identification number assigned to each debug component. The number must be unique in the Rohm group. Be sure to use the value given from the platform development department.
This system ROM table has the following Part Number:

Part Number[11:0]	Corresponding component	Remarks
0x000	UxPlatform Cortex®-M0+ System ROM Table	When you want to change the UxPlatform configuration, contact the platform development department.

■ 8. Examples of Application Circuit

Please refer to design guide (FEXL7416N-060DG).

■ 9. Package External Dimensions/Footprint Pattern



LAPIS Semiconductor Co.,Ltd.			
PACKAGE CODE	P-LBGA81-1010-1.00-1-MC		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
BALL MATERIAL	Sn/Ag/Cu	DWG No.	QSL-69083
PACKAGE MASS (g)	0.20TYP.	REVISION	3
		1st ISSUE	May/28/2014
		REVISED	Jun/19/2015

Remarks for surface mount type package

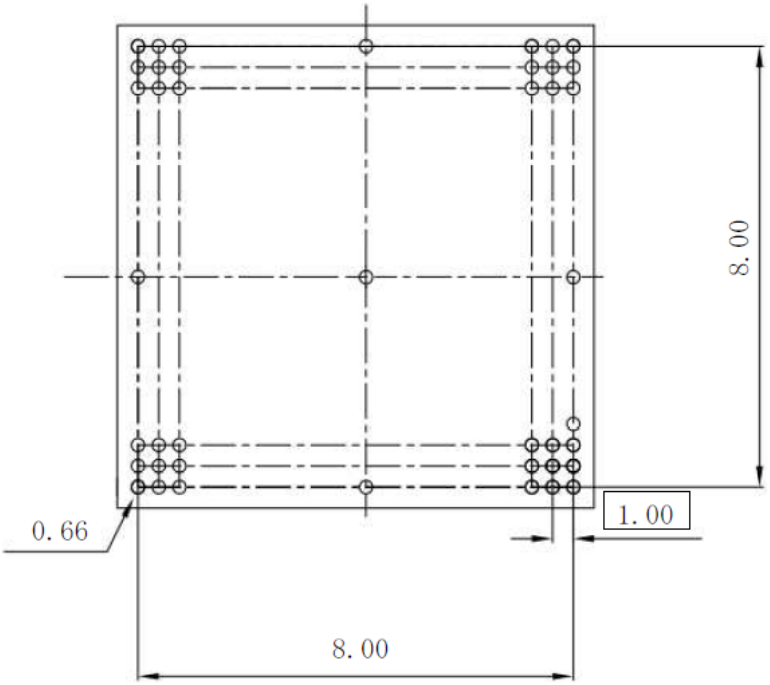
The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

P-LBGA81-1010-1.00-1-MC

参考図

半田付け部端子存在範囲図

Mounting area for package lead soldering to PC boards



［単位：mm］

実装基板のフットパターンの設計の際には、実装の容易さ、接続の信頼性、配線の引き回し、半田ブリッジ発生のないことなどを十分考慮してください。
フットパターンの最適な設計は基板材質、使用する半田ペースト種類、厚み、半田付け方法などによって変わってきます。従って、本パッケージの端子の存在し得る範囲を「半田付け部端子存在範囲図」として示しますので、フットパターン設計の参考資料としてください。

When laying out PC boards, it is important to design the foot pattern so as to give consideration to ease of mounting, bonding, positioning of parts, reliability, wiring, and elimination of solder bridges.
The optimum design for the foot pattern varies with the materials of the substrate, the sort and thickness of used soldering paste, and the way of soldering. Therefore when laying out the foot pattern on the PC boards, refer to this figure which mean the mounting area that the package leads are allowable for soldering to PC boards.

LAPIS Semiconductor Co., Ltd.

■ 10. Revision History

Version	Revision description	Before change	After change	Date	Remarks
1.0	Ver. 1	-	-	September 7, 2015	
2.0	The I/O of the SPI data input/output pin 1 (MISO) has been corrected.	8, 10	8, 10	October 14, 2015	
	The pin handling of ADC0-3, SWCK, and SWD has been added.	12	12		
	The typical value of the RC clock 1 is has been corrected. The maximum and minimum values of the RC clock frequency have been added.	14	14		
	The unit has been corrected.	22	22		
	The maximum value of SCK-SSN lag time (tLAG) has been corrected.	23	23		
	The caution has been added.	27	28		
	The maximum conversion time has been added.	28	29		
	The condition for accuracy, maximum and minimum values of temperature gradient, and maximum conversion time have been added.	29	30		
	The low voltage detection characteristic has been added.	-	31		
	The description of Source Clock has been corrected.	36	37		
	The description of the clock selection has been corrected.	38	39		
	The description of the setting register has been corrected.	44	45		
	The description of the RF interrupt source has been added.	59	60		
	The list of system control registers has been corrected.	66	67		
	The initial value of the peripheral clock and enable register has been corrected.	76	77		

	The caution in DEEPSLEEP mode has been added.	78	79		
	The description has been corrected.	81	82		
	The initial value of Component Parameter Register has been corrected.	86	87		
	The description of the SIRE bit has been added.	98	99		
	The initial value of the Component Parameter register has been corrected.	119	120		
	The initial value of the UART Component Version register has been corrected.	121	122		
	The description of the initial value of the SPI control register has been added.	125	126		
	The information has been added that the initial value of the MSTR bit is Master for only SPI2.	129	130		
	The initial value of the SPIn_BRR register has been corrected.	130	131		
	The description has been added that these registers are enabled for only SPI2.	135-136	136-137		
	The mistakes in the examples of the initial setting flow diagram and register initial setting have been corrected.	151	152		
	The initial value of the interrupt mask register has been corrected.	154	155		
	The description of the receive FIFO threshold level register bit has been corrected.	162	163		
	The initial value of the interrupt mask register has been corrected.	165	166		
	The initial value of the WDT current counter value register has been corrected.	175	176		
	The description of the INTRPORT_SEL register has been added.	182	183		
	The description of BER_MODE has been added.	185	186		
	The list of registers has been corrected.	195	196		

	The initial value of the timer 1 current value register has been corrected.	208	206		
	The initial value of the timer 1 current value register has been corrected.	209	207		
	The initial value of the timer n current value register has been corrected.	214	212		
	The initial value of the timer n current value register has been corrected.	215	213		
	The register name has been corrected (n -> 0).	241-	239-		
	The description of the FTMEN bit has been added.	253	251		
	The description of the FTMDIS bit has been added.	254	252		
	The initial value has been corrected.	265-266	263-264		
	The description of the operation in 1 bank mode has been added.	298	296		
	The flash ROM area controlled by the flash ROM controller has been added.	299	297		
	The R/W attribute of PROTUNLOCKWE has been corrected.	317	315		
	The description of PROTLOCKKEYSTA has been corrected.	318	316		
	The description of using DIO has been added.	340	338		
	The R/W attribute of DIO_IMSK has been corrected.	340	338		
	The R/W attribute of F_O_CLR has been corrected.	350	348		
	The R/W attribute of F_U_CLR has been corrected.	351	349		
	The R/W attribute of MATCH_CLR has been corrected.	352	350		
	The R/W attribute of ICLR has been corrected.	353	351		

Version	Revision description	Before change	After change	Date	Remarks
2.0	The summary has been corrected.	359	357	October 14, 2015	
	The CT timer register has been added to the list of registers.	359	357		
	The CT timer register has been added.	-	359		
	The description of ADRUN has been corrected.	366	365		
	The caution about clock switching processing during AD conversion has been added.	367	366		
	The description about how to use ADC has been added.	-	380		
	The initial values in the list of LVD registers have been corrected.	385	384		
	The initial value of the LVD0 enable register has been corrected.	388	387		
	The initial value of the LVD1 control register has been corrected. The threshold voltage for each setting value has been added.	389-390	388-389		
	The description of the low voltage detection (LVD) operation has been added.	-	399-402		
	The list of DMAC registers has been corrected.	401-405	404-407		
	The initial value of the channel n control register has been corrected.	409	411		
	The description of the interrupt RAW status register has been added.	416	418		
	The description of the interrupt source status register has been added.	420	422		
	The R/W attribute of the DMA channel enable register has been corrected.	428	430		

	The description of the flow controller has been corrected.	430	432		
	The description of assignment of the handshake interface has been added.	431	433		
	The description of the FDMA_EN bit has been added.	440	442		
	The description of the FlashDMA setting register 3 has been added.	443	445-446		
	The bit width of FDMA_ERR_EOI in the register list has been corrected from [3:0] to [8:0].	444	447		
	The register cleared by AESOBFL and AESIBFL has been corrected.	457	460		
	The functional description of the AES has been added.	476, 479-480	479, 482-483		
	In the list of registers, the initial values have been corrected, and the registers added (clock status, DMON monitor, PLL setting 2).	485-486	488-490		
	The initial value of the PLL setting register 0 has been corrected.	490	494		
	The adc_div bit has been added to the CLKGEN setting register 5.	501	505		
	The descriptions of the FLASH_PS0 bit and LOGIC_PS0 bit have been corrected.	507-508	511-512		
	The description of the FLASH-ROM deep standby control register has been corrected.	510	514		
	The clock status register has been added.	-	526-527		
	The register for monitoring the DMON_CPU pin has been added.	-	528		
	The PLL setting register 2 has been added.	-	529		

	The initial value of the IO setting register has been corrected, and its description has been added.	522	530-532		
	In the clock system diagram, the dividing values have been corrected, and the caution has been added.	523	533		
	The PLL_DIV setting has been added to the PLL control procedure.	526	536		
	The description of RF clock control has been added.	-	539		
	The external dimension has been updated (tolerance information has been added).	535	546		
	The footprint pattern has been renamed.	536	547		
	CR40M and CR32K have been renamed to high-speed CR and low-speed CR.	Entire document	Entire document		

Version	Revision description	Before change	After change	Date	Remarks
3.0	The minimum and maximum values of the SLow clock 32.768 kHz crystal oscillator frequency have been added.	14	14	2016/3/30	
	The definitions of high-level input voltage/low-level input voltage of the CXIN pin have been deleted.	16	16		
	The maximum value of consumption current has been added.	16	32		
	The maximum values of input leakage current and Tri-state output leakage current have been changed.	16	16		
	The description of (*5) has been corrected.	16	16		
	The maximum value of low-level output voltage has been corrected.	16	16		
	The RF characteristics of 915 MHz band and 868 MHz band have been deleted.	19-21	19-21		
	The typical value of the minimum receiver sensitivity has been corrected.	21	21		
	Temperature gradient in Temperature Sensor Characteristics has been deleted.	30	30		
	The minimum and maximum values of the LVD detection error have been corrected.	31	31		
	The R/W attribute has been corrected (R -> R/W).	80	80		
	The initial value has been corrected (0x00 -> 0x60).	87,101	87,101		

	The initial value has been corrected (0x00 -> 0x60).	88,111	88,111		
	The register name has been corrected (UARTn_IER -> UARTn_IIR).	93	93		
	The description of the register has been corrected.	121	121		
	The wrong R/W attribute of the SPIn_ID register has been corrected (R/W -> R). The description of the register has been corrected.	127	127		
	The initial value has been corrected.	131	131		
	The R/W attribute has been corrected (R -> R/W).	133	133		
	The summary of WDT has been corrected.	176	176		
	The description of the timer control procedure has been corrected.	211	211		
	The description of the ADJ30S bit has been added.。	226	226		
	The RESTART function has been removed.	265,273 289-290	265,273 289-290		
	The description of the bit 16 has been corrected (BANK_SEL -> BANK_MODE).	319	320		
	The description of the bit of the CT timer register has been corrected.	359	360		
	The temperature condition has been corrected (from 85 °C to 105 °C).	363,382	364,383		
	The R/W attribute of LVD0W has been corrected (R -> R/W).	386	387		
	The description of the LVD1LV bits of the LVD1 control register has been corrected.	389	391		
	The bit 18 and bit 17 have been corrected to Reserved.	412	414		

	The description of the bit has been corrected.	422	424		
	The description of [Note] has been added.	440	443		
	The bit width of the error status clear has been corrected from [7:0] to [8:0].	447	450		
	The TEST setting register 2 has been added.	-	535		
	The description of the PLL setting register 0 has been added.	495	498		
	The R/W attribute of the Reserved bit has been corrected (R/W -> R).	501	504		
	The dividing values of the CLKGEN setting register have been corrected.	501-505	504-510		
	The R/W attribute of the SUB_CLK_DIV bit has been corrected (R -> R/W).	503	507		
	The starting and stopping procedures in DeepSleep mode have been corrected.	537	543-544		
	The resolution has been reviewed to read values easier.	546	552		
4.0	Add Product Name,application	1	1	2023/11/1	
	The description of [Note] has been updated.	572	572		
5.0	The description of [Note] has been updated.	572	572	2024/1/10	

Notes

1) When using LAPIS Technology Products, refer to the latest product information and ensure that usage conditions (absolute maximum ratings*1, recommended operating conditions, etc.) are within the ranges specified. LAPIS Technology disclaims any and all liability for any malfunctions, failure or accident arising out of or in connection with the use of LAPIS Technology Products outside of such usage conditions specified ranges, or without observing precautions. Even if it is used within such usage conditions specified ranges, semiconductors can break down and malfunction due to various factors. Therefore, in order to prevent personal injury, fire or the other damage from break down or malfunction of LAPIS Technology Products, please take safety at your own risk measures such as complying with the derating characteristics, implementing redundant and fire prevention designs, and utilizing backups and fail-safe procedures.

   *1: Absolute maximum ratings: a limit value that must not be exceeded even momentarily.

2) The Products specified in this document are not designed to be radiation tolerant.

3) Descriptions of circuits, software and other related information in this document are provided only to illustrate the standard operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. And the peripheral conditions must be taken into account when designing circuits for mass production. LAPIS Technology disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, and other related information.

4) No license, expressly or implied, is granted hereby under any intellectual property rights or other rights of LAPIS Technology or any third party with respect to LAPIS Technology Products or the information contained in this document (including but not limited to, the Product data, drawings, charts, programs, algorithms, and application examples, etc.). Therefore, LAPIS Technology shall have no responsibility whatsoever for any dispute, concerning such rights owned by third parties, arising out of the use of such technical information.

5) LAPIS Technology intends our Products to be used in a way indicated in this document. Please be sure to contact a ROHM sales office if you consider the use of our Products in different way from original use indicated in this document. For use of our Products in medical systems, please be sure to contact a LAPIS Technology representative and must obtain written agreement. Do not use our Products in applications which may directly cause injuries to human life, and which require extremely high reliability, such as aerospace equipment, nuclear power control systems, and submarine repeaters, etc. LAPIS Technology disclaims any and all liability for any losses and damages incurred by you or third parties arising by using the Product for purposes not intended by us without our prior written consent.

6) All information contained in this document is subject to change for the purpose of improvement, etc. without any prior notice. Before

purchasing or using LAPIS Technology Products, please confirm the latest information with a ROHM sales office. LAPIS Technology has used reasonable care to ensure the accuracy of the information contained in this document, however, LAPIS Technology shall have no responsibility for any damages, expenses or losses arising from inaccuracy or errors of such information.

7) Please use the Products in accordance with any applicable environmental laws and regulations, such as the RoHS Directive. LAPIS Technology shall have no responsibility for any damages or losses resulting non-compliance with any applicable laws or regulations.

8) When providing our Products and technologies contained in this document to other countries, you must abide by the procedures and provisions stipulated in all applicable export laws and regulations, including without limitation the US Export Administration Regulations and the Foreign Exchange and Foreign Trade Act.

9) Please contact a ROHM sales office if you have any questions regarding the information contained in this document or LAPIS Technology's Products.

10) This document, in part or in whole, may not be reprinted or reproduced without prior consent of LAPIS Technology.

(Note) “LAPIS Technology” as used in this document means LAPIS Technology Co., Ltd.

LAPIS Technology Co.,Ltd.

2-4-8 Shinyokohama, Kouhoku-ku, Yokohama 222-8575, Japan

https://www.lapis-tech.com/en/

LTSZ08023・01・002