Toshiba Semiconductor and Storage - TMPM46BF10FG Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M4F
Core Size	32-Bit Single-Core
Speed	120MHz
Connectivity	EBI/EMI, I²C, IrDA, Microwire, SIO, SPI, SSI, SSP, UART/USART
Peripherals	DMA, LVD, POR, WDT
Number of I/O	71
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	514K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/toshiba-semiconductor-and-storage/tmpm46bf10fg

1.2 Block Diagram

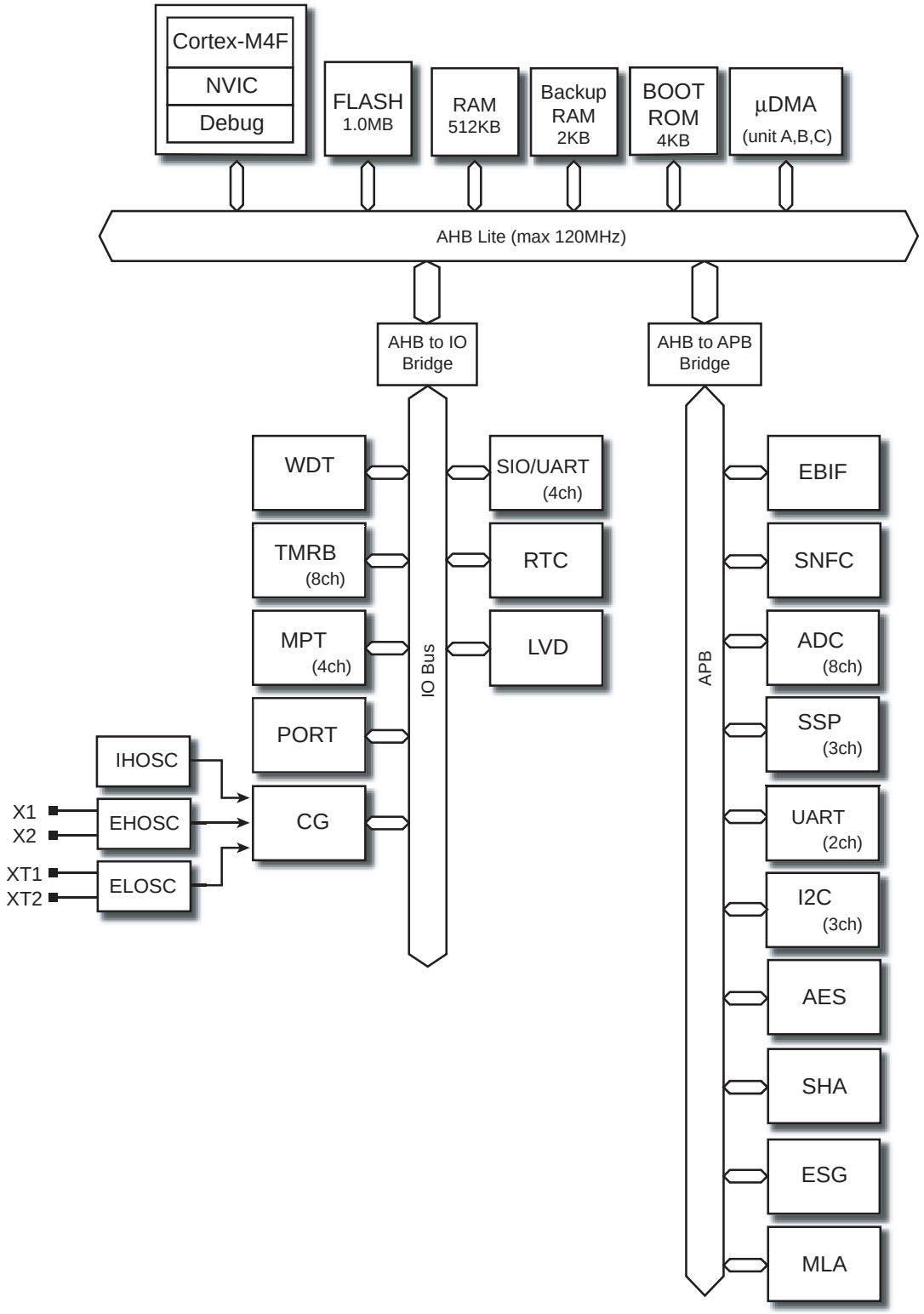


Figure 1-1 Block Diagram

5. Reset Operation

The following are sources of reset operation.

- RESET pin ($\overline{\text{RESET}}$)
- Releasing the STOP2 mode
- Low voltage detection circuit (LVD)
- Watch-dog timer (WDT)
- Application interrupt by CPU and a signal from the reset register bit <SYSRESETREQ>

To recognize a source of reset, check CGRSTFG in the clock generator register described in chapter of "Exception".

A reset by releasing the STOP2 mode is refer to the "Clock/Mode control".

A reset by low voltage detection circuit is refer to the "Low voltage detection circuit".

A reset by WDT is refer to the chapter on the "Watch-dog timer".

A reset by <SYSRESETREQ> is referred to "Cortex-M4 Technical Reference Manual".

Note: Once reset operation is done, internal RAM data is not assured.

5.1 Cold Reset

When turning-on power, $\overline{\text{RESET}}$ pin must be kept "Low".

When turning-on power, it is necessary to take a stable time of built-in regulator into consideration. In the TMPM46BF10FG, the internal regulator requires at least approximately 2ms to be stable. At cold reset, $\overline{\text{RESET}}$ pin must be kept "Low" for a duration of time sufficiently long enough for the internal regulator to be stable. Approximately 0.251ms after $\overline{\text{RESET}}$ pin becomes "High", internal reset will be released.

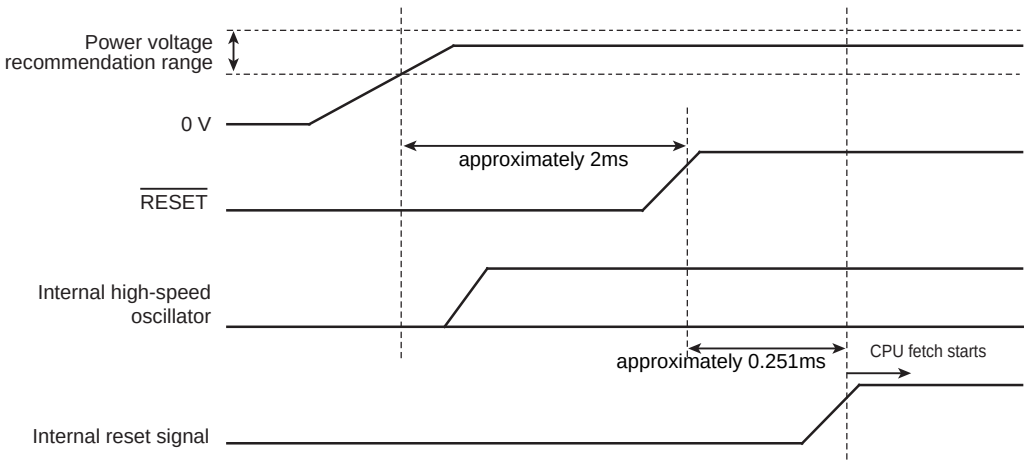


Figure 5-1 Cold Reset Operation Sequence

6.2 Registers

6.2.1 Register List

The following table shows the Clock/Mode control related registers and addresses.

For the Base Address, refer to the "Peripheral Base address list "of Chapter "Memory Map".

Register name		Address (Base+)
System control register	CGSYSCR	0x0000
Oscillation control register	CGOSCCR	0x0004
Standby control register	CGSTBYCR	0x0008
PLL selection register	CGPLLSEL	0x000C
Clock stop register A for peripheral	CGFSYSMSKA	0x0020
Clock stop register B for peripheral	CGFSYSMSKB	0x0024
Protect register	CGPROTECT	0x003C

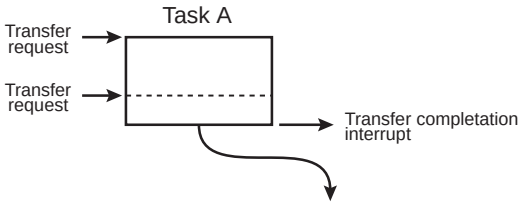
Bit	Bit Symbol	Type	Function
3	USGFAULT ACT	R/W	Usage Fault 0: Inactive 1: Active
2	−	R	Read as 0.
1	BUSFAULT ACT	R/W	Bus Fault 0: Inactive 1: Active
0	MEMFAULT ACT	R/W	Memory Management 0: Inactive 1: Active

Note: **You must clear or set the active bits with extreme caution because clearing and setting these bits does not repair stack contents.**

Preparation:

Prepare primary data and alternative data, and set "1" to the bits of the channels corresponding to both DMAxCfg<master_enable> and DMAxChnlEnableSet.

Task A: Primary data
<cycle_ctrl[2:0]> = "011"
(ping-pong mode)
<R_power[3:0]> = "0010"
(4 times)
<n_minus_1[9:0]> = "0x005" (6 times)



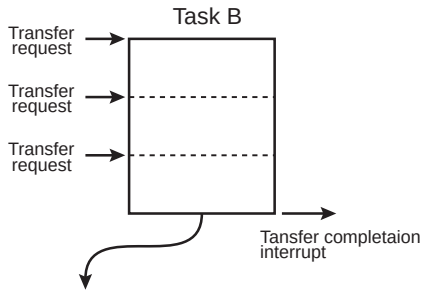
Receiving a transfer request, the DMA performs a transfer four times and performs arbitration.

If there is no other high-priority requests, the DMA performs remaining transfers twice toward a request for a transfer to the corresponding channels.

The DMA generates a transfer completion interrupt request and performs an arbitration.

After completing Task A, primary data for Task C can be set.

Task B: Alternative data
<cycle_ctrl[2:0]> = "011"
<R_power[3:0]> = "0010"
(4 times)
<n_minus_1[9:0]> = "0x00B" (12 times)



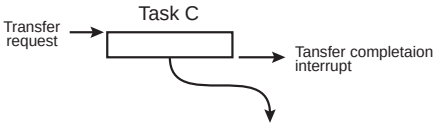
Receiving a transfer request, The DMA performs a transfer four times and performs arbitration.

If there is no other high-priority requests, The DMA performs transfers twice toward a request for a transfer to the corresponding channels.

The DMA generates a transfer completion interrupt request and performs an arbitration.

After completing Task B, alternative data for Task D can be set.

Task C: Primary data
<cycle_ctrl[2:0]> = "011"
<R_power[3:0]> = "0001"
(2 times)
<n_minus_1[9:0]> = "0x001" (2 times)

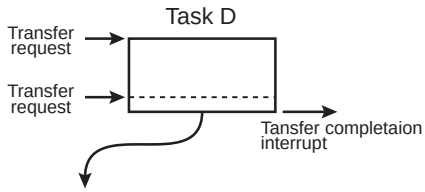


Receiving a transfer request, the DMA performs a transfer twice and performs arbitration.

The DMA generates a transfer completion interrupt request and performs an arbitration.

After completing Task C, alternative data for Task E can be set.

Task D: Alternative data
<cycle_ctrl[2:0]> = "011"
<R_power[4:0]> = "0010"
(4 times)
<n_minus_1[9:0]> = "0x004" (5 times)

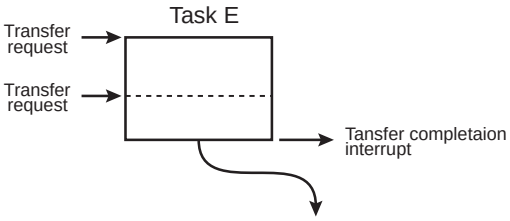


Receiving a transfer request, the DMA performs a transfer four times and performs arbitration.

If there is no other high-priority requests, the DMA performs a transfer once toward a request for a transfer to the corresponding channels.

The DMA generates a transfer completion interrupt request and performs an arbitration.

Task E: Primary data
<cycle_ctrl[2:0]> = "011"
<R_power[3:0]> = "0010"
(4 times)
<n_minus_1[9:0]> = "0x006" (7 times)

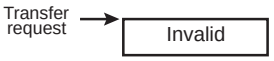


Receiving a transfer request, the DMA performs a transfer four times and performs arbitration.

If there is no other high-priority requests, the DMA performs transfers three times toward a request for a transfer to the corresponding channels.

The DMA generates a transfer completion interrupt request and performs an arbitration.

Final: Alternative data
<cycle_ctrl[2:0]> = "000"
(invalid)



Even receiving a transfer request, the operation stops because <cycle_ctrl[2:0]> is set to invalid.

(The operation can be also stopped by setting the <cycle_ctrl[2:0]> of Task E to normal mode "001".)

11.4.3 Error Correction Mode

As an error correction function, software correction and automatic correction can be selected with the ECC mode register (SNFCECCMOD <GOUTMODE>).

To use software correction, set "0" to <GOUTMODE>. To use automatic correction, set "1" to <GOUTMODE>.

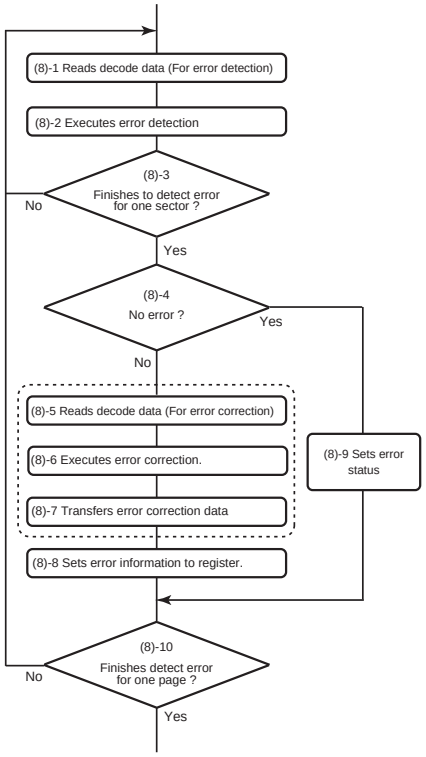
When the software correction is used, correct the software with the Sector x ECC error "n" position information register (SNFCSxEEnPI). (x: Sector number, n: Error number)

Table 11-5 shows the ECC mode at write/read operation.

Table 11-5 ECC mode at read/write operation.

Operation		ECC mode		
	The number of correction bits.	Decoding mode	BCH mode	Correction mode
		SNFCCSE<DECMODE> 0: Encoding 1: Decoding	SNFCECCMOD <SELBCH> 0: BCH4+CRC 1: BCH8	SNFCECCMOD <GOUTMODE> 0: Software correction (No correction data is output.) 1: Automatic correction (Correction data is output.)
Write	4 bits per 512 Bytes	0	0	-
	8 bits per 512 Bytes		1	
Read	4 bits per 512 Bytes, software correction	1	0	0
	4 bits per 512 Bytes, automatic correction			1
	8 bits per 512 Bytes, software correction		1	0
	8 bits per 512 Bytes, automatic correction			1

Table 11-13 (8) decoding (automatic correction)



(8)-1: Reads decode data (For error detection)
Read a data for error detection from RAM1.

(8)-2 : Executes error detection
Executes error detection every one byte.

(8)-3 : Finishes to detect error for one sector?
Executes error detection for one sector.

(8)-4 : No error?
If no error for one sector, error correction is not executed.
Only if errors exist, error correction is executed. (When automatic error correction)

(8)-5 : Reads decode data (For error correction)
Reads a data as same as error detection for error correction.

(8)-6 : Executes error correction.
Executes error correction every one byte.

(8)-7 : Transfers error correction data
Transfers the generated error correction data to RAM3.

(8)-8 : Set error information to registers.
Set a detected error information to registers.
- ECC decode state register (SNFCEDSx)
- ECC error n position information register (SNFCSxEEnPI)

If an error correction can not be executed by error information, an uncorrectable error detection interrupt.

(8)-9 : Sets error status.
Sets status flag in (8)-4.
- ECC Error status register (SNFCEES)

(8)-10 : Completes for one page?
Executes sectors for one page and completes to decode.

Note that when software correction is used, the processes ((8)-5 to 7) indicated by dashed lines are not executed. Based on error information, correct errors by software.

16.3.5 TBxMOD (Mode Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	TBCPM		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	TBCP	-	-	TBCLE	TBCLK		
After reset	0	1	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-11	−	R	Read as "0".
10-8	TBCPM[2:0]	R/W	Sets capture timing by TBxIN0/1 and up-counter clearing timing. 000: Disabled 001: TBxIN0↑ TBxIN1↑ Captures a counter value on rising edge of TBxIN0 input into Capture register 0 (TBxCP0). Captures a counter value on rising edge of TBxIN1 input into Capture register 1 (TBxCP1). 010: TBxIN0↑ TBxIN0↓ Captures a counter value on rising edge of TBxIN0 input into Capture register 0 (TBxCP0). Captures a counter value on falling edge of TBxIN0 input into Capture register 1 (TBxCP1). 011: TBxFF0↑ TBxFF0↓ Captures a counter value on rising edge of TBxFF0 input into Capture register 0 (TBxCP0). Captures a counter value on falling edge of TBxFF0 input into Capture register 1 (TBxCP1). 100: Clears up-counter on TBxIN1↑ 101: Captures a counter value on TBxIN0↑ into Capture register 0(TBxCP0); clears up-counter on TBxIN1↑ If capture timing and up-counter clearing timing are same, capturing is performed first, and then up-counter is cleared. 110 to 111:Reserved
7	−	R/W	Write "0".
6	TBCP	W	Capture control by software 0: Capture by software 1: Don’t care When "0" is written, the capture register 0 (TBxCP0) captures a count value. Read as "1".
5-4	−	R	Read as "0".
3	TBCLE	R/W	Up-counter control 0: Disables clearing of the up-counter. 1: Enables clearing of the up-counter. Clears and controls the up-counter. When "0" is written, it disables clearing of the up-counter. When "1" is written, it clears up-counter when up-counter matches with timer regsiter1 (TBxRG1).
2-0	TBCLK[2:0]	R/W	Selects the TMRBx source clock. 000: TBxIN pin input 001: φT1 010: φT4 011: φT16 100: φT32 101: φT64 110: φT128 111: φT256

16.5.3 Programmable Pulse Generation (PPG) Output Mode

Square wave can be output in any frequency and duty. The output pulse can be either low-active or high-active.

TBxFF0 is reversed when the up-counter matches the set value of TBxRG0 and TBxRG1. TBxFF0 can be output from TBxOUT pin.

Note that the set value of TBxRG0 and TBxRG1 must be satisfy the following requirement.

Set value of TBxRG0 < Set value of TBxRG1

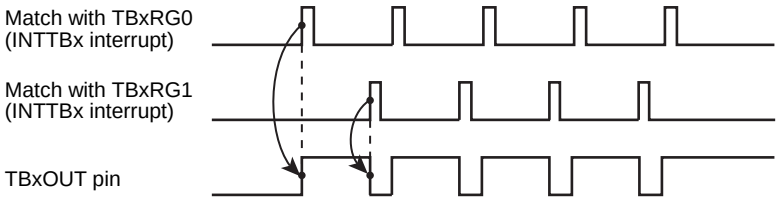


Figure 16-2 Example of programmable pulse generation output

In this mode, by enabling the double buffering, the value of register buffer 0 and 1 are shifted into TBxRG0 and TBxRG1 when UC matches the value of TBxRG1.

This makes possible to modify frequency and duty without a concern of a change timing of TBxRG0 and TBxRG1.

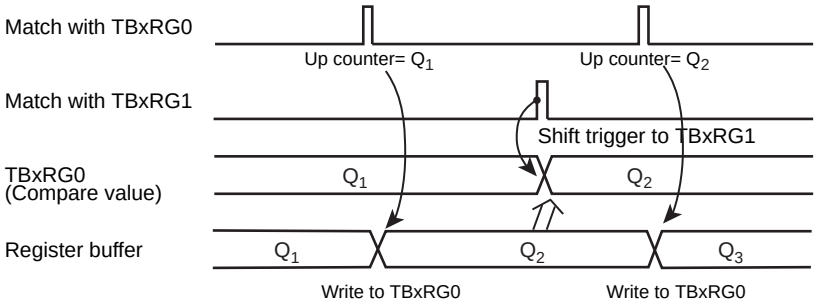


Figure 16-3 Register buffer operation

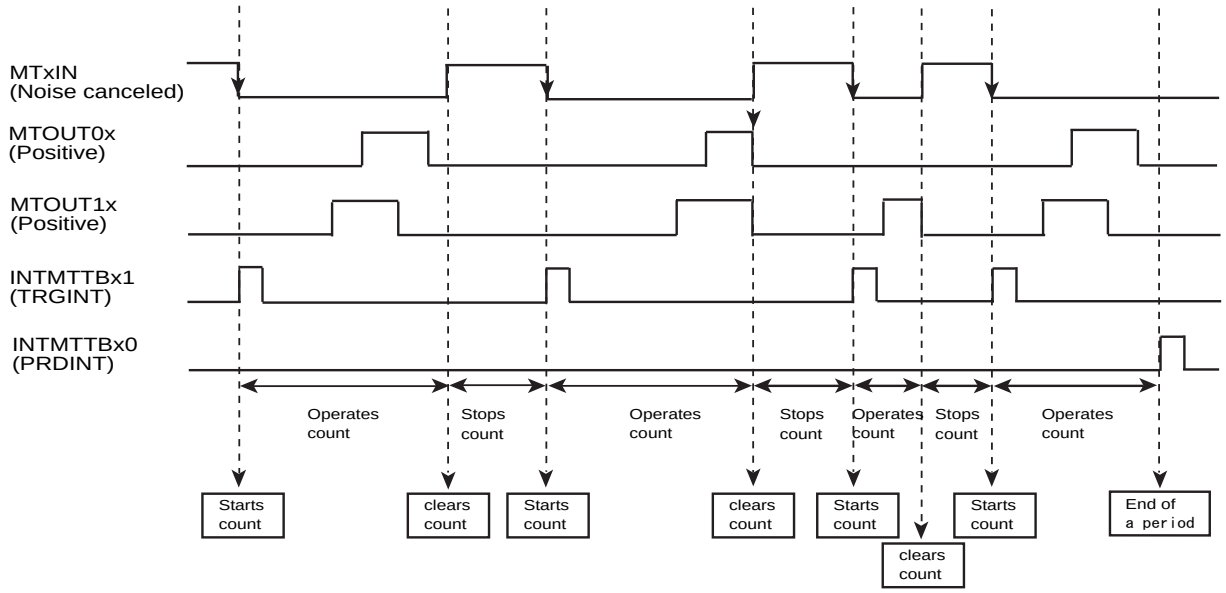


Figure 17-20 Trigger constant acceptance

When <IGTRGM>="1" is set, input edge at MTxOUT0/1x output in non-active is accepted and cleared to stop.

If input edge at MTxOUT0/1x output in active, the counter does not immediately stops. It continues to count until MTxOUT0/1x output becomes non-active. When MTxOUT0/1x output is non-active, if trigger signal is not in active level, the counter is cleared to stop and waits next start trigger signal.

If the counter operates when both MTxOUT0 and MTxOUT1 are enabled, both outputs must be in non-active. Otherwise triggers are not accepted.

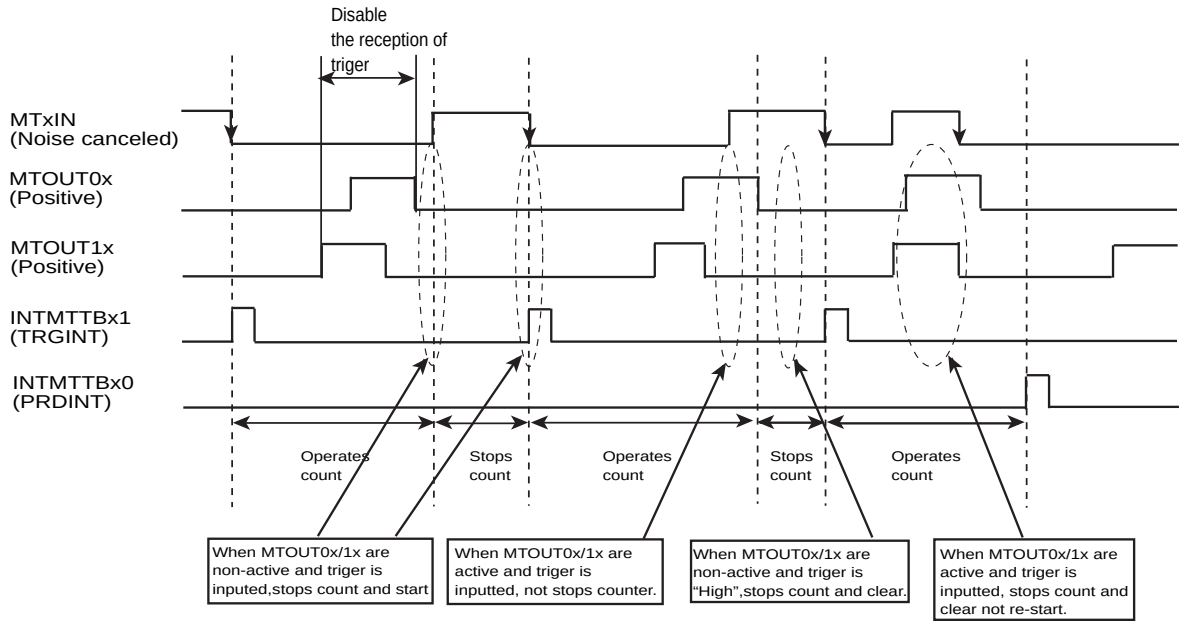


Figure 17-21 Prohibit accessing during active level

18.3.6 SCxMOD1 (Mode Control Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	I2SC	FDPX		TXE	SINT			-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	I2SC	R/W	IDLE 0: Stop 1: Operate Specifies operation in the IDLE mode.
6-5	FDPX[1:0]	R/W	Transfer mode setting 00: Transfer prohibited 01: Half duplex (Receive) 10: Half duplex (Transmit) 11: Full duplex Configures the transfer mode in the I/O interface mode. And when FIFO is enabled, specify the configuration of FIFO. In UART mode, specify the only configuration of FIFO.
4	TXE	R/W	Transmit control (Note1)(Note2) 0 :Disabled 1: Enabled This bit enables transmission and is valid for all the transfer modes.
3-1	SINT[2:0]	R/W	Interval time of continuous transmission (For I/O interface mode) 000: None 001: 1 x SCLK cycle 010: 2 x SCLK cycle 011: 4 x SCLK cycle 100: 8 x SCLK cycle 101: 16 x SCLK cycle 110: 32 x SCLK cycle 111: 64 x SCLK cycle This parameter is valid only for the I/O interface mode when SCLK output mode is selected. In other modes, this parameter has no meaning. Specifies the interval time of continuous transmission when double buffering or FIFO is enabled in the I/O interface mode.
0	−	R/W	Write a "0".

Note 1: Specify the all mode control registers first and then enable the <TXE>.

Note 2: Do not stop the transmit operation (by setting <TXE> to "0") when data is being transmitted.

18.16.1.3 Transmit and Receive (Full-duplex)

(1) Clock Output Mode

- If double buffers are disabled (SCxMOD2<WBUF> = "0")

  Clock is output when the CPU writes data to the transmit buffer.

  Subsequently, a data is shifted into receive buffer and the INTRXx is generated. Concurrently, a data written to the transmit buffer is output from the SCxTXD pin, the INTTXx is generated when transmission of all data has been completed. Then, the clock output stops.

  The next round of data transmission and reception starts when the data is read from the receive buffer and the next transmit data is written to the transmit buffer by the CPU. The order of reading the receive buffer and writing to the transmit buffer can be freely determined. Data transmission is resumed only when both conditions are satisfied.

- If double buffers are enabled (SCxMOD2<WBUF> = "1")

  Clock is outputted when the CPU writes data to the transmit buffer.

  A data is shifted into the receive shift register, moved to the receive buffer, and the INTRXx is generated. While a data is received, a transmit data is output from the SCxTXD pin. When all data are sent out, the INTTXx is generated and the next data is moved from the transmit buffer to the transmit shift register.

  If the transmit buffer has no data to be moved to the transmit buffer (SCxMOD2<TBEMP> = "1") or when the receive buffer is full (SCxMOD2<RBFLL> = "1"), the clock output stops. When both conditions, receive data is read and transmit data is written, are satisfied, the clock output is resumed and the next round of data transmission and reception is started.

Bit	Bit Symbol	Type	Function
2	SIRLP	RW	IrDA encoding mode select for transmitting 0 bit : 0 : 0 bit are transmitted as an active high pulse of 3/16th of the bit period. 1 : 0 bit are transmitted with a pulse width that is 3 times the period of the IrLPBaud16 input signal. <SIRLP> selects IrDA encoding mode. When this bit is cleared to 0, 0 bits of the IrDA transmission data are transmitted as an active high pulse (UTxIROUT) with a width of 3/16th of the bit period. When this bit is set to 1, 0 bits of the IrDA transmission data are transmitted with a pulse width that is 3 times the period of the IrLPBaud16 input signal. Setting this bit can reduce power consumption but might decrease transmission distances.
1	SIREN	RW	SIR enable 0 : Disabled 1 : Enabled When this bit is set to 1, the IrDA circuit is enabled. To use the UART, the <UARTEN> must be set to 1. When the IrDA circuit is enabled, the UTxIROUT and UTxIRIN pins are enabled. The UTxTXD pin remains in the marking state (set to 1). Signal transitions on the UTxRXD pin or modem status input have no effect. When IrDA circuit is disabled, UTxIROUT remains cleared to 0 (no light pulse is generated) and the UTxIRIN pin has no effect.
0	UARTEN	R/W	UART permission setting 0 : Disabled 1 : Enabled When <UARTEN> is set to "0", UART is disabled. If UART is disabled during transmission or reception, UART stops after current on-going data transmission or reception is complete. When <UARTEN> is set to "1", data is sent or received using UART function or SIR function according to a value of <SIREN>.

19.3.13 UARTxRIS (UART Raw Interrupt Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	OERIS	BERIS	PERIS
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	FERIS	RTRIS	TXRIS	RXRIS	DSRRMIS	DCDRMIS	CTSRMIS	RIRMIS
After reset	0	0	0	0	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-11	−	R	Read as an undefined value.
10	OERIS	R	Overrun error interrupt status : 0: No interrupt request 1: Interrupt request.
9	BERIS	R	Break error interrupt status : 0: No interrupt request 1: Interrupt request
8	PERIS	R	Parity error interrupt status : 0: No interrupt request 1: Interrupt request
7	FERIS	R	Framing error interrupt status : 0: No interrupt request 1: Interrupt request
6	RTRIS	R	Receive timeout interrupt status : 0: No interrupt request 1: Interrupt request
5	TXRIS	R	Transmit interrupt status : 0: No interrupt request 1: Interrupt request
4	RXRIS	R	Receive interrupt status : 0: No interrupt request 1: Interrupt request
3	DSRRMIS	R	Read as an undefined value. 0: No interrupt request 1: Interrupt request
2	DCDRMIS	R	DCD modem raw interrupt status 0: No interrupt request 1: Interrupt request
1	CTSRMIS	R	CTS modem interrupt status : 0: No interrupt request 1: Interrupt request
0	RIRMIS	R	RIN modem raw interrupt status 0: No interrupt request 1: Interrupt request

Note: All the bits, except the modem raw status interrupt bits (bits 3 to 0), are cleared to 0 when reset. Themodem status bit are undefined after reset.

The slave device ignores a slave address and general calls sent from the master and returns non-acknowledgment. the INTI2Cx interrupt request are not generated.

In master mode, the bit of I2CxCR1<NOACK> is ignored and has no effect on operation.

Note: if I2CxCR1<NOACK> is cleared to "0" during data transfer in slave mode, I2CxCR1<NOACK> remains "1" and an acknowledge signal is returned for the transferred data.

20.4.3 Setting the Acknowledgement Mode

Setting I2CxCR1<ACK> to "1" selects the acknowledge mode. When operating as a master, the I2C adds one clock for acknowledgment signal. In slave mode, the clock for acknowledgement signals is counted. In transmitter mode, the I2C releases the SDAx pin during clock cycle to receive acknowledgement signals from the receiver. In receiver mode, the I2C pulls the SDAx pin to the "Low" level during the clock cycle and generates acknowledgement signals. Also in slave mode, if a general-call address is received, the I2C pulls the SDAx pin to the "Low" level during the clock cycle and generates acknowledgement signals.

However, the second byte is necessary to be controlled by software to generate an ACK signal on the contents of the second byte.

By setting <ACK> to "0", the non-acknowledgment mode is activated. When operating as a master, the I2C does not generate clock for acknowledgement signals. In slave mode, the clock for acknowledgement signals is not counted.

20.4.4 Setting the Number of Bits per Transfer

I2CxCR1<BC[2:0]> specifies the number of bits of the next data to be transmitted or received.

Under the start condition, <BC[2:0]> is set to "000", causing a slave address and the direction bit to be transferred in a packet of eight bits. At other times, <BC[2:0]> keeps a previously programmed value.

Note: A slave address must be trasmitted / received with I2CxCR1<ACK> set to "1". if I2CxCR1<ACK> is cleared, the slave address match detection and direction bit detection cannot be performed properly.

20.4.5 Slave Addressing and Address Recognition Mode

Setting "0" to I2CxAR<ALS> and a slave address in I2CxAR<SA[6:0]> sets addressing format, and then the I2C recognizes a slave address transmitted by the master device and receives data in the addressing format.

If <ALS> is set to "1", the I2C does not recognize a slave address and receives data in the free data format. In the case of free data format, a slave address and a direction bit are not recognized; they are recognized as data immediately after generation of the start condition.

21.3.10 SSPxICR (Interrupt clear register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After Reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After Reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After Reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	RTIC	RORIC
After Reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-2	−	W	Write as "0".
1	RTIC	W	Clear the time-out interrupt flag: 0: Invalid 1: Clear
0	RORIC	W	Clear the overrun interrupt flag: 0: Invalid 1: Clear

21.3.11 SSPxDMACR (DMA control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After Reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After Reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After Reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	TXDMAE	RXDMAE
After Reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0	0

Bit	Bit Symbol	Type	Function
31-2	−	W	Write as "0".
1	TXDMAE	R/W	Transmit FIFO DMA control: 0:Disable 1:Enable
0	RXDMAE	R/W	Transmit FIFO DMA control: 0:Disable 1:Enable

22.2 Configuration

Figure 22-1 shown the block diagram of the AD converter.

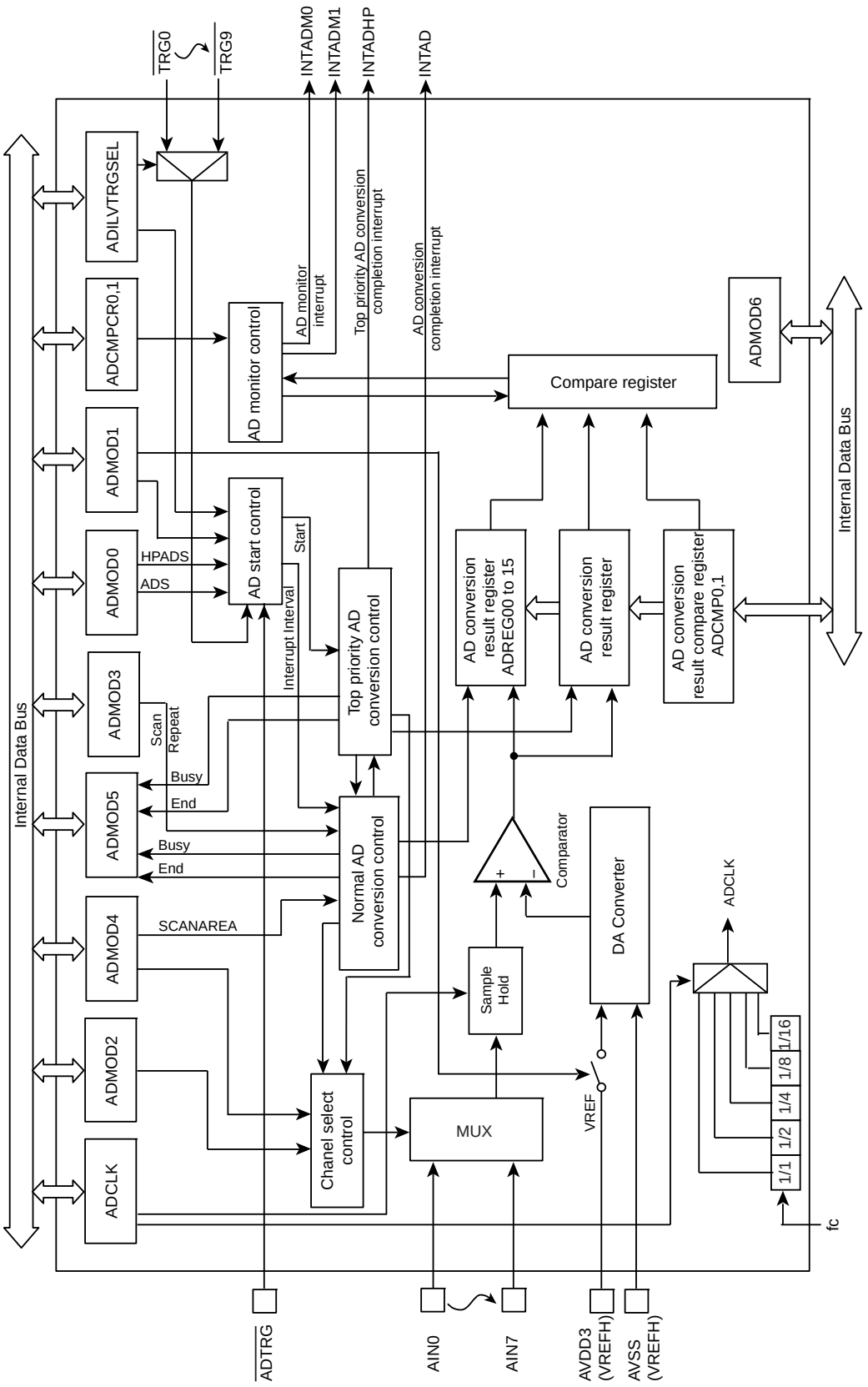


Figure 22-1 Block Diagram

22.3 Registers

22.3.1 Register list

The AD converter is controlled by the Mode Setting Registers (ADMOD0 through ADMOD6).

For the base address, refer to "Address lists of peripheral functions" of Chapter "Memory Map".

Registers		Address (Base+)
Clock Setting Register	ADCLK	0x0000
Mode Setting Register 0	ADMOD0	0x0004
Mode Setting Register 1	ADMOD1	0x0008
Mode Setting Register 2	ADMOD2	0x000C
Mode Setting Register 3	ADMOD3	0x0010
Mode Setting Register 4	ADMOD4	0x0014
Mode Setting Register 5	ADMOD5	0x0018
Mode Setting Register 6	ADMOD6	0x001C
Monitoring Setting Register 0	ADCMPCR0	0x0024
Monitoring Setting Register 1	ADCMPCR1	0x0028
AD Conversion Result Compare Register 0	ADCMP0	0x002C
AD Conversion Result Compare Register 1	ADCMP1	0x0030
AD Conversion Result Register 0	ADREG00	0x0034
AD Conversion Result Register 1	ADREG01	0x0038
AD Conversion Result Register 2	ADREG02	0x003C
AD Conversion Result Register 3	ADREG03	0x0040
AD Conversion Result Register 4	ADREG04	0x0044
AD Conversion Result Register 5	ADREG05	0x0048
AD Conversion Result Register 6	ADREG06	0x004C
AD Conversion Result Register 7	ADREG07	0x0050
Top-priority Conversion Result Register	ADREGSP	0x0074

Registers		Address (Base+)
Trigger Selection Register	ADILVTRGSEL	0x0010

Note: Access the registers by using word (32 bit) reads and word writes.

Table 26-23 Communication Rules of Flash memory Chip Erase and Protect Bit Erase

No	Transfer direction	Transfer data	Description
1	C→T	Operation command data (0x40)	Sends Flash memory chip erase and protect bit erase command data (0x40).
2	C←T	ACK response to operation command Normal: 0x40 Abnormal: 0x41 Communication error: 0x48	ACK response data to the operation command. First, checks if 3rd byte of receive data has errors. (UART mode only) If receive errors exist, sends a ACK response data 0xX8 that means abnormal communication sand waits for a next operation command (3rd byte). Upper 4 bits of transmit data are undefined. (same as upper 4 bits of immediately before operation command data.) Note that in the I/O interface, receive error check is not performed. Then, if the 3rd byte of receive data corresponds to either operation command data in Table 26-11, receive data is echoed back. If the data does not correspond to the command in Table 26-11, sends a ACK response data 0xX1 that means operation command errors, and waits for next operation command. (3rd byte) Upper 4 bits of transmit data are undefined. (Upper 4 bits of immediate before operation command data are used.)
3	C→T	Password data (12 bytes)	Transmit data which is the same as password data area of Flash memory from the controller. However; if password requirement of Flash memory is set to "no" (data: 0xFF), the target does not conduct password verification, so that dummy data can be used as a password. For details of password data area, refer to "26.3.5.3 Password Determination".
4	C→T	CHECK SUM value of transmit data (No.3)	Transmit a CHECK SUM value of transmit data (No.3) from the controller. For a method of CHECK SUM calculation, refer to "26.3.5.4 CHECK SUM Calculation".
5	C←T	ACK response to CHECK SUM value Normal: 0x40 Abnormal: 0x41 Communication error: 0x48	The target checks receive data and responses ACK response data. If receive error exists, the target responses ACK response data 0x48 indicating abnormal communication, and then returns to the initial state waiting for operation command data. If receive error does not exist, the target checks a CHECK SUM value. If checking is failed, the target responses ACK response data 0x41 indicating abnormal communication, and then returns to the initial state waiting for operation command data. If checking is succeeded, the target performs password checking. If password requirement is set to "no", the target transmits ACK response data 0x40 indicating normal. If password requirement is set to "need password", the target checks the password. If password checking is failed, the target responses ACK response data 0x41 indicating abnormal communication, and then returns to the initial state waiting for operation command data. If password checking is succeeded, the target responses ACK response data 0x40 indicating normal and then it waits next data.
6	C→T	Erase enable command data (0x54)	Transmit erase enable command data (0x54) from the controller.
7	C←T	ACK response to erase enable command Normal: 0x54 Abnormal: 0x51 Communication error: 0x58	The target checks receive data and responses ACK response data. If receive error exists, the target responses ACK response data 0x58 indicating communication error, and then returns to the initial state waiting for operation command data. If receive error does not exist, the target checks erase enable command (0x54). If checking is failed, the target responses ACK response data 0x51 indicating abnormal communication, and then returns to the initial state waiting for operation command data. If checking is succeeded, the target responses ACK response data 0x54 indicating normal, and then chip erase process is performed.
8	C←T	ACK response to erase command Normal: 0x4F Abnormal: 0x4C Abort chip erase command: 0x47	The target responses the result of chip erase process. If any problems occur, the target responses ACK response data (0x4F) indicating normal. If an blank check error occurs, the target responses ACK response data (0x4C) indicating abnormal. If chip erase command is aborted, the target responses ACK response data (0x47) indicating abort and then returns to the initial state waiting for operation command data.