MF1117-02

EPSON

CMOS 32-BIT SINGLE CHIP MICROCOMPUTER

E0C33A104 TECHNICAL MANUAL

E0C33A104 Technical Hardware

ENERGY SAVING EPSON

SEIKO EPSON CORPORATION

CONTENTS

1 Outline

The E0C33A104 is a Seiko Epson original 32-bit microcomputer that features low power and low-voltage operation. It is designed for portable equipment that needs advanced data processing.
The E0C33A104 consists of the E0C33000 32-bit RISC type CPU as the core, a bus control unit, a DMA controller, an interrupt controller, timers, serial interface circuits, A/D converter, D/A converter, and RAM. It also includes two oscillation circuits that generate high-speed and low-speed operating clocks allowing high-speed operation and low-power operation and a clock timer that provides excellent clock functions.
The E0C33A104 also provides a DSP function, by using the internal MAC (multiplication and accumulation) operation function with the A/D converter, it makes it possible to design simply speech recognition and voice synthesis systems.

1.1 Features

Core CPU

Seiko Epson original 32-bit RISC CPU E0C33000 built-in

- Basic instruction set: 105 instructions (16-bit fixed size)
- Sixteen 32-bit general-purpose register
- 32-bit ALU and 8-bit shifter
- Multiplication/division instructions and MAC (multiplication and accumulation) instruction are available
- 30 ns of minimum instruction execution time at 33 MHz operation

Internal memory

RAM:	2K bytes
ARAM:	4K bytes (used as internal RAM)

Internal peripheral circuits

Oscillation circuit:	High-speed (OSC3) oscillation circuit	33 MHz max.
		Crystal/ceramic oscillator or external clock input
	Low-speed (OSC1) oscillation circuit	32.768 kHz typ.
		Crystal oscillator or external clock input
Timers:	8-bit timer	4 channels
	16-bit timer	6 channels
	Watchdog timer	
	Clock timer	1 channel (with alarm function)
Serial interface:	2 channels (clock-synchronous system, asynchronous system and IrDA interface are selectable)	
A/D converter:	10 bits × 8 channels	
D/A converter:	8 bits × 2 channels	
DMA controller:	High-speed DMA 2 channels	
	Intelligent DMA 128 channels	
Interrupt controller:	Possible to invoke intelligent DMA	
	Input interrupt	6 types
	DMA controller interrupt	3 types
	16-bit programmable timer interrupt	16 types
	8-bit programmable timer interrupt	4 types
	Serial interface interrupt	6 types
	A/D converter interrupt	1 type
	Clock timer interrupt	1 type
General-purpose input and output ports:	Shared with the I/O pins for internal peripheral circuits	
	Input port	13 bits (built-in pull-up resistors are available)
	Output port	11 bits
	I/O port	15 bits (built-in pull-up resistors are available)

External bus interface

BCU (bus control unit) built-in

- 24-bit address bus (internal 28-bit processing)
- 16-bit data bus
  Data size is selectable from 8 bits and 16 bits in each area.
- Little-endian memory access
- Memory mapped I/O
- Chip enable and wait control circuits built-in
- DRAM direct interface function built-in
  Supports fast page mode and EDO page mode.
  Supports self-refresh and CAS-before RAS refresh.
- Supports burst ROM.

Operating conditions and power consumption

Operating voltage:	5 V ±10% or 3.3 V ±0.3 V		
Operating clock frequency:	Max. 33 MHz at 5 V operation		
	Max. 20 MHz at 3 V operation		
Operating temperature:	-20 to 70°C		
Power consumption:	During SLEEP	5 µW typ.	(5 V)
(Typ.)		4 µW typ.	(3.3 V)
	During HALT	200 mW typ.	(5 V, 33 MHz)
		40 mW typ.	(3.3 V, 20 MHz)
	During execution	400 mW typ.	(5 V, 33 MHz)
		100 mW typ.	(3.3 V, 20 MHz)

Note: The values of power consumption during execution were measured when a test program that consisted of 55% load instructions, 23% arithmetic operation instructions, 1% mac instruction, 12% branch instructions and 9% ext instruction was being continuously executed.

Supply form

QFP5-128pin, QFP15-128pin plastic package or die form

1.2 Block Diagram

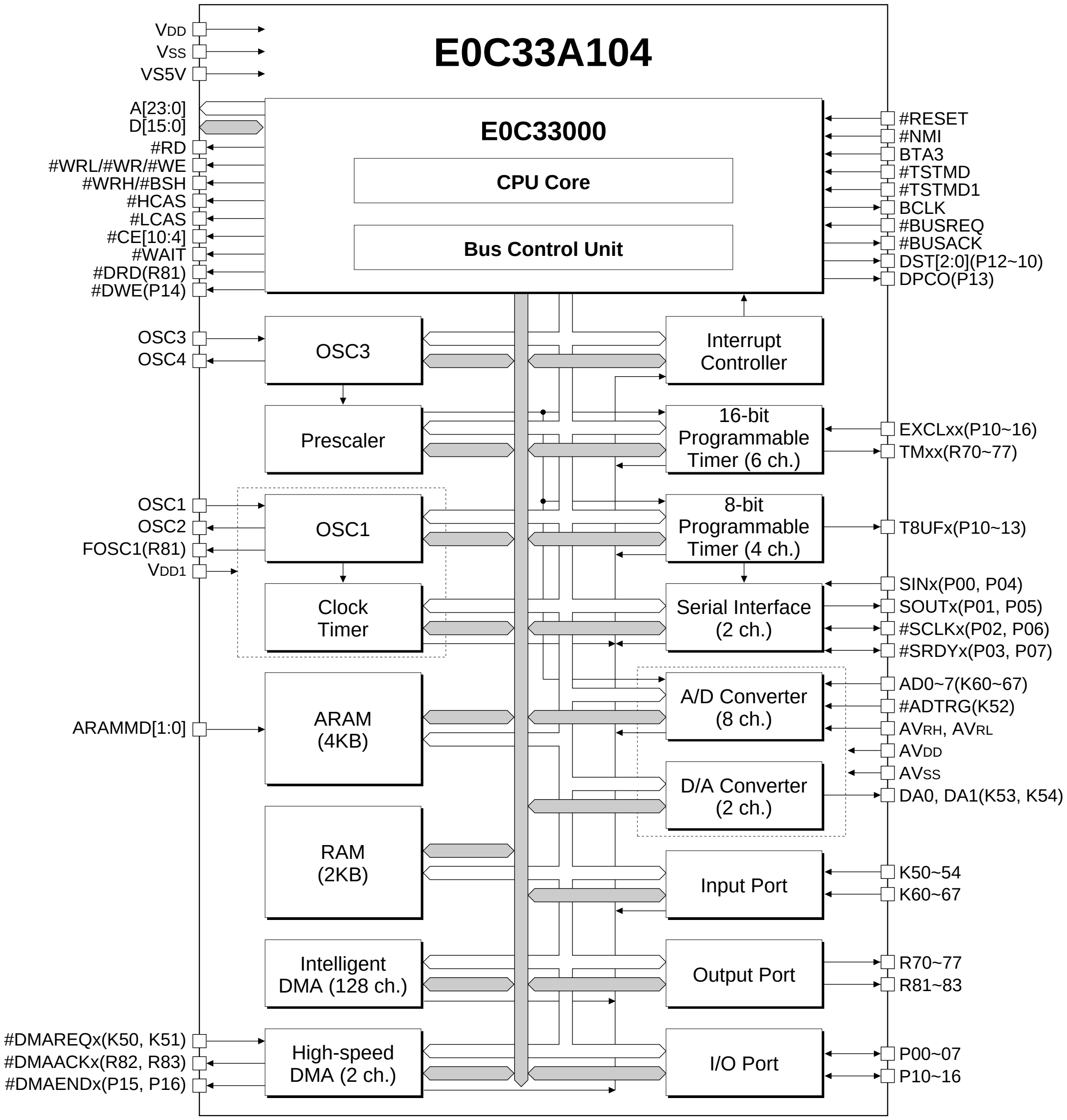
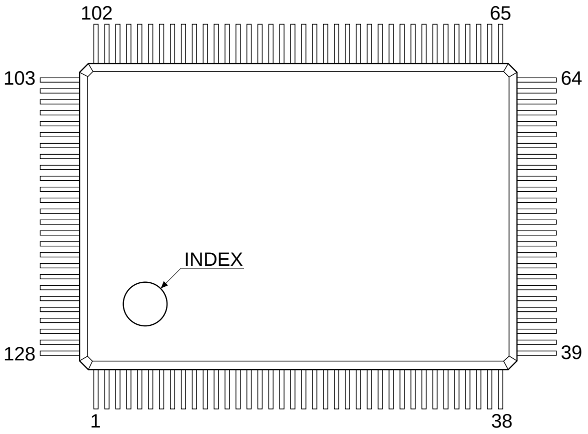


Figure 1.2.1 E0C33A104 Block Diagram

1.3 Pin Description

1.3.1 Pin Layout Diagram (plastic package)

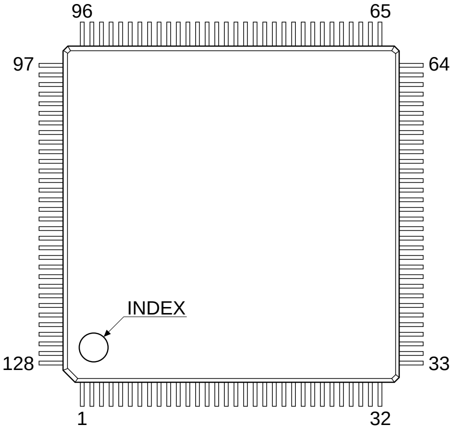
QFP5-128pin



No.	Pin name	No.	Pin name	No.	Pin name	No.	Pin name
1	#CE5/#CE15	33	OSC1	65	P03/#SRDY0	97	D1
2	#CE6	34	V_{DD1}	66	P04/SIN1	98	D0
3	#CE7/#RAS0/#CE13/#RAS2	35	K52/#ADTRG	67	P05/SOUT1	99	#WAIT
4	#CE8/#RAS1/#CE14/#RAS3	36	R70/TM00	68	P06/#SCLK1	100	ARAMMD1
5	#CE9/#CE17	37	R71/TM01	69	P07/#SRDY1	101	ARAMMD0
6	#CE10	38	R72/TM10	70	A7	102	#BUSACK
7	VS5V	39	R73/TM11	71	A6	103	#BUSREQ
8	#NMI	40	R74/TM21	72	A5	104	BCLK
9	V_{SS}	41	R75/TM31	73	A4	105	#HCAS
10	#TSTMD	42	R76/TM41	74	A3	106	#LCAS
11	#RESET	43	R77/TM51	75	A2	107	#WRH/#BSH
12	V_{DD}	44	V_{SS}	76	A1	108	#WRL/#WR/#WE
13	V_{SS}	45	V_{DD}	77	A0/#BSL	109	#RD
14	OSC4	46	P10/EXCL00/T8UF0/DST0	78	D15	110	V_{DD}
15	OSC3	47	P11/EXCL01/T8UF1/DST1	79	V_{SS}	111	A8
16	AV_{RH}	48	P12/EXCL10/T8UF2/DST2	80	D14	112	A9
17	AV_{RL}	49	P13/EXCL20/T8UF3/DPCO	81	D13	113	A10
18	AV_{SS}	50	P14/EXCL30/#BUSGET/#DWE	82	D12	114	A11
19	K67/AD7	51	P15/EXCL40/#DMAEND0	83	D11	115	A12
20	K66/AD6	52	P16/EXCL50/#DMAEND1	84	D10	116	A13
21	K65/AD5	53	BTA3	85	D9	117	A14
22	K64/AD4	54	K51/#DMAREQ1	86	D8	118	A15
23	K63/AD3	55	R83/#DMAACK1	87	D7	119	A16
24	K62/AD2	56	#TSTMD1	88	V_{DD}	120	A17
25	K61/AD1	57	R81/FOSC1/#DRD	89	V_{SS}	121	A18
26	K60/AD0	58	K50/#DMAREQ0	90	D6	122	A19
27	K53/DA0	59	R82/#DMAACK0	91	V_{DD}	123	A20
28	K54/DA1	60	P00/SIN0	92	V_{SS}	124	A21
29	AV_{DD}	61	V_{DD}	93	D5	125	A22
30	N.C.	62	V_{SS}	94	D4	126	A23
31	V_{SS}	63	P01/SOUT0	95	D3	127	V_{SS}
32	0SC2	64	P02/#SCLK0	96	D2	128	#CE4/#CE11

Figure 1.3.1 Pin Layout Diagram (QFP5-128pin)

QFP15-128pin



No.	Pin name	No.	Pin name	No.	Pin name	No.	Pin name
1	#CE8/#RAS1/#CE14/#RAS3	33	R70/TM00	65	P06/#SCLK1	97	ARAMMD1
2	#CE9/#CE17	34	R71/TM01	66	P07/#SRDY1	98	ARAMMD0
3	#CE10	35	R72/TM10	67	A7	99	#BUSACK
4	VS5V	36	R73/TM11	68	A6	100	#BUSREQ
5	#NMI	37	R74/TM21	69	A5	101	BCLK
6	VSS	38	R75/TM31	70	A4	102	#HCAS
7	#TSTMD	39	R76/TM41	71	A3	103	#LCAS
8	#RESET	40	R77/TM51	72	A2	104	#WRH/#BSH
9	VDD	41	VSS	73	A1	105	#WRL/#WR/#WE
10	VSS	42	VDD	74	A0/#BSL	106	#RD
11	OSC4	43	P10/EXCL00/T8UF0/DST0	75	D15	107	VDD
12	OSC3	44	P11/EXCL01/T8UF1/DST1	76	VSS	108	A8
13	AVRH	45	P12/EXCL10/T8UF2/DST2	77	D14	109	A9
14	AVRL	46	P13/EXCL20/T8UF3/DPCO	78	D13	110	A10
15	AVSS	47	P14/EXCL30/#BUSGET/#DWE	79	D12	111	A11
16	K67/AD7	48	P15/EXCL40/#DMAEND0	80	D11	112	A12
17	K66/AD6	49	P16/EXCL50/#DMAEND1	81	D10	113	A13
18	K65/AD5	50	BTA3	82	D9	114	A14
19	K64/AD4	51	K51/#DMAREQ1	83	D8	115	A15
20	K63/AD3	52	R83/#DMAACK1	84	D7	116	A16
21	K62/AD2	53	#TSTMD1	85	VDD	117	A17
22	K61/AD1	54	R81/FOSC1/#DRD	86	VSS	118	A18
23	K60/AD0	55	K50/#DMAREQ0	87	D6	119	A19
24	K53/DA0	56	R82/#DMAACK0	88	VDD	120	A20
25	K54/DA1	57	P00/SIN0	89	VSS	121	A21
26	AVDD	58	VDD	90	D5	122	A22
27	VSS	59	VSS	91	D4	123	A23
28	0SC2	60	P01/SOUT0	92	D3	124	VSS
29	0SC1	61	P02/#SCLK0	93	D2	125	#CE4/#CE11
30	N.C.	62	P03/#SRDY0	94	D1	126	#CE5/#CE15
31	VDD1	63	P04/SIN1	95	D0	127	#CE6
32	K52/#ADTRG	64	P05/SOUT1	96	#WAIT	128	#CE7/#RAS0/#CE13/#RAS2

Figure 1.3.2 Pin Layout Diagram (QFP15-128pin)

1.3.2 Pin Functions

Table 1.3.1 List of Pins for Power Supply System

Pin name	Pin No.		I/O	Pull-up	Function
	QFP5-128	QFP15-128			
VDD	12,45,61, 88,91,110	9,42,58, 85,88,107	–	–	Power supply (+)
VSS	9,13,31, 44,62,79, 89,92,127	6,10,27, 41,59,76, 86,89,124	–	–	Power supply (-)
VDD1	34	31	–	–	Low-speed oscillation system power supply (+)
AVDD	29	26	–	–	Analog system power supply (+)
AVSS	18	15	–	–	Analog system power supply (-)
AVRH	16	13	–	–	Analog system reference voltage (+)
AVRL	17	14	–	–	Analog system reference voltage (-)

Table 1.3.2 List of Pins for External Interface Signals

Pin name	Pin No.		I/O	Pull-up	Function
	QFP5-128	QFP15-128			
A[0]/#BSL	77	74	O	–	Address bus (A0) / Bus strobe (low byte)
A[23:1]	70–76, 111–126	67~73, 108~123	O	–	Address bus (A1 to A23)
D[15:0]	78,80–87, 90,93–98	75,77~84, 87,90~95	I/O	–	Data bus (D0 to D15)
#CE10	6	3	O	–	Area 10 chip enable
#CE9/#CE17	5	2	O	–	Area 9/17 chip enable
#CE8/#RAS1/ #CE14/#RAS3	4	1	O	–	Area 8/14 chip enable / DRAM row strobe
#CE7/#RAS0/ #CE13/#RAS2	3	128	O	–	Area 7/13 chip enable / DRAM row strobe
#CE6	2	127	O	–	Area 6 chip enable
#CE5/#CE15	1	126	O	–	Area 5/15 chip enable
#CE4/#CE11	128	125	O	–	Area 4/11 chip enable
#RD	109	106	O	–	Read signal
#WRL/#WR/#WE	108	105	O	–	Write (low byte) / Write / DRAM write
#WRH/#BSH	107	104	O	–	Write (high byte) / Bus strobe (high byte)
#HCAS	105	102	O	–	DRAM column address strobe (high byte)
#LCAS	106	103	O	–	DRAM column address strobe (low byte)
BCLK	104	101	O	–	Bus clock output
#BUSREQ	103	100	I	–	Bus release request
#BUSACK	102	99	O	–	Bus acknowledge
#WAIT	99	96	I	–	Wait cycle request

Table 1.3.3 List of I/O Pins for Input/Output Ports and Internal Peripheral Circuits

Pin name	Pin No.		I/O	Pull-up	Function
	QFP5-128	QFP15-128			
K50/#DMAREQ0	58	55	I	built-in	Input port / High-speed DMA request 0
K51/#DMAREQ1	54	51	I	built-in	Input port / High-speed DMA request 1
K52/#ADTRG	35	32	I	built-in	Input port / AD converter trigger input
K53/DA0	27	24	I(O)	built-in	Input port / DA converter output 0
K54/DA1	28	25	I(O)	built-in	Input port / DA converter output 1
K60/AD0	26	23	I	built-in	Input port / AD converter input 0
K61/AD1	25	22	I	built-in	Input port / AD converter input 1
K62/AD2	24	21	I	built-in	Input port / AD converter input 2
K63/AD3	23	20	I	built-in	Input port / AD converter input 3
K64/AD4	22	19	I	built-in	Input port / AD converter input 4
K65/AD5	21	18	I	built-in	Input port / AD converter input 5
K66/AD6	20	17	I	built-in	Input port / AD converter input 6
K67/AD7	19	16	I	built-in	Input port / AD converter input 7

Pin name	Pin No.		I/O	Pull-up	Function
	QFP5-128	QFP15-128			
R70/TM00	36	33	O	–	Output port (initial value = "0") /16-bit timer (timer 00) output
R71/TM01	37	34	O	–	Output port (initial value = "0") /16-bit timer (timer 01) output
R72/TM10	38	35	O	–	Output port (initial value = "0") /16-bit timer (timer 10) output
R73/TM11	39	36	O	–	Output port (initial value = "0") /16-bit timer (timer 11) output
R74/TM21	40	37	O	–	Output port (initial value = "0") /16-bit timer (timer 21) output
R75/TM31	41	38	O	–	Output port (initial value = "0") /16-bit timer (timer 31) output
R76/TM41	42	39	O	–	Output port (initial value = "0") /16-bit timer (timer 41) output
R77/TM51	43	40	O	–	Output port (initial value = "0") /16-bit timer (timer 51) output
R81/FOSC1/#DRD	57	54	O	–	Output port (initial value = "1") / Low-speed (OSC1) oscillation clock output/ DRAM read
R82/#DMAACK0	59	56	O	–	Output port (initial value = "1") / High-speed DMA acknowledge 0
R83/#DMAACK1	55	52	O	–	Output port (initial value = "1") / High-speed DMA acknowledge 1
P00/SIN0	60	57	I/O	built-in	I/O port / Serial IF Ch.0 data input
P01/SOUT0	63	60	I/O	built-in	I/O port / Serial IF Ch.0 data output
P02/#SCLK0	64	61	I/O	built-in	I/O port / Serial IF Ch.0 clock input/output
P03/#SRDY0	65	62	I/O	built-in	I/O port / Serial IF Ch.0 ready signal output
P04/SIN1	66	63	I/O	built-in	I/O port / Serial IF Ch.1 data input
P05/SOUT1	67	64	I/O	built-in	I/O port / Serial IF Ch.1 data output
P06/#SCLK1	68	65	I/O	built-in	I/O port / Serial IF Ch.1 clock input/output
P07/#SRDY1	69	66	I/O	built-in	I/O port / Serial IF Ch.1 ready signal output
P10/EXCL00/T8UF0/DST0	46	43	I/O	built-in	I/O port / Timer 00 event counter input / 8-bit timer 0 output / DST0 output
P11/EXCL01/T8UF1/DST1	47	44	I/O	built-in	I/O port / Timer 01 event counter input / 8-bit timer 1 output / DST1 output
P12/EXCL10/T8UF2/DST2	48	45	I/O	built-in	I/O port / Timer 10 event counter input / 8-bit timer 2 output / DST2 output
P13/EXCL20/T8UF3/DPCO	49	46	I/O	built-in	I/O port / Timer 20 event counter input / 8-bit timer 3 output / DPCO output
P14/EXCL30/#BUSGET/#DWE	50	47	I/O	built-in	I/O port / Timer 30 event counter input / #BUSGET signal output / DRAM write
P15/EXCL40/DMAEND0	51	48	I/O	built-in	I/O port / Timer 40 event counter input / End of high-speed DMA (Ch.0)
P16/EXCL50/DMAEND1	52	49	I/O	built-in	I/O port / Timer 50 event counter input / End of high-speed DMA (Ch.1)

Note 1 Follow the notes below for pins with a pull-up resistor (Kxx, Pxx).

1) The pull-up resistors can be turned on and off individually using the control registers.
2) All the pull-up resistors turn off at cold start. At hot start, they retain the previous status (on or off) before an initial reset.

Note 2 The output port pins (Rxx) output the initial value "0" ("1" for R81 to R83) at cold start. At hot start, they retain the previous status before an initial reset.

Table 1.3.4 List of Other Pins

Pin name	Pin No.		I/O	Pull-up	Function
	QFP5-128	QFP15-128			
OSC1	33	29	I	–	Low-speed (OSC1) oscillation input (32 kHz crystal oscillator or external clock input)
OSC2	32	28	O	–	Low-speed (OSC1) oscillation output
OSC3	15	12	I	–	High-speed (OSC3) oscillation input (crystal/ceramic oscillator or external clock input)
OSC4	14	11	O	–	High-speed (OSC3) oscillation output
#NMI	8	5	I	–	NMI request
BTA3	53	50	I	–	Boot address setup (Low: Area 10)
VS5V	7	4	I	–	Input threshold voltage setup (High: TTL, Low: CMOS)
ARAMMD[1:0]	100,101	97,98	I	–	ARAM mode setup
#TSTMD	10	7	I	–	Test input (Fixed at high during operation.)
#TSTMD1	56	53	I	–	Test input (Fixed at high during operation.)
#RESET	11	8	I	built-in	Initial reset input

2 Power Supply and I/O Level

This chapter explains the operating voltage and I/O levels of the E0C33A104.

2.1 Power Supply

2.1.1 Power Supply Pins

The E0C33A104 has the power supply pins shown in Table 2.1.1.

Table 2.1.1 Power Supply Pins

Pin name	Pin No.		Function
	QFP5-128	QFP15-128	
VDD	12,45,61,88,91,110	9,42,58,85,88,107	Power supply (+) for the internal logic circuits and I/O
VSS	9,13,31,44,62,79, 89,92,127	6,10,27,41,59,76, 86,89,124	Power supply (GND) for the internal logic circuits and I/O
VDD1	34	31	Power supply (+) for the low-speed (OSC1) oscillation circuit and the clock timer
AVDD	29	26	Power supply (+) for the internal analog circuits
AVSS	18	15	Power supply (GND) for the internal analog circuits

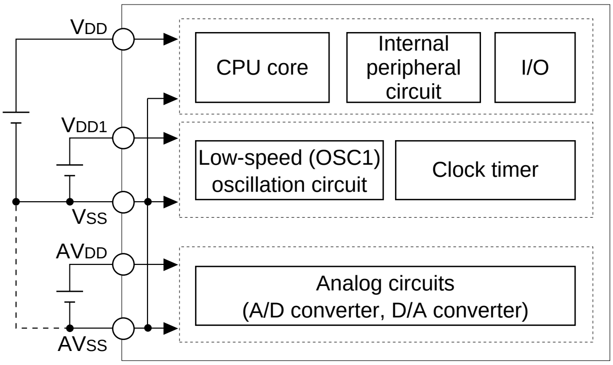

VDD
CPU core
Internal peripheral circuit
I/O
VDD1
Low-speed (OSC1) oscillation circuit
Clock timer
VSS
AVDD
Analog circuits
(A/D converter, D/A converter)
AVSS

Figure 2.1.1 Power Supply System

2.1.2 Operating Voltage (VDD)

The core CPU and internal peripheral circuit (excluding analog circuits, low-speed oscillation circuit and clock timer) operate with a voltage supplied between the VDD and VSS pins. The following two operating voltages can be used:

5 V system: VDD = 5.0 V ± 10% (VSS = GND)
3 V system: VDD = 3.3 V ± 0.3 V (VSS = GND)

Note: The E0C33A104 has six VDD pins and nine VSS pins. Be sure to supply the operating voltage to all the pins. Do not open any of them.

The operating clock frequency range is limited as shown in Table 2.1.2 according to the operating voltage to be supplied.

Table 2.1.2 Range of Operating Clock Frequency

Supply voltage (VDD)	Minimum operating clock frequency (OSC3)	Maximum operating clock frequency (OSC3)
5.0 V ± 10%	5 MHz	33 MHz
3.3 V ± 0.3 V	5 MHz	20 MHz

2.1.3 Power Supply for Clock (VDD1)

The VDD1 voltage is used for driving the low-speed (OSC1) oscillation circuit and the clock timer. VDD1 can be supplied separately from VDD, it makes it possible to drive the clock timer system when the core CPU and other internal peripheral circuits are turned off. The same power supply can also be used for both VDD and VDD1. The VSS pin is used for the ground common with VDD.
The following two voltage levels are enabled for VDD1 the same as VDD:

5 V system: VDD = 5.0 V ± 10% (VSS = GND)
3 V system: VDD = 3.3 V ± 0.3 V (VSS = GND)

However, satisfy the condition of VDD1 ≤ VDD. When VDD is 3.3 V, 5 V cannot be used for VDD1.

2.1.4 Power Supply for Analog Circuits (AVDD, AVSS)

The analog power supply pins (AVDD and AVSS) are provided separately from the VDD and VSS pins in order that the digital circuits do not affect the analog circuits (A/D converter and D/A converter). The AVDD pin is used to supply an analog power voltage and the AVSS pin is used as the analog ground.
Supply the same voltage level as the VDD to the AVDD pin.

AVDD = VDD, AVSS = VSS (GND)

Note: These pins should be set as AVDD = VDD, AVSS = VSS (GND) even if the analog circuits are not used.

The AVSS pin is electrically connected with the VSS pin on the chip substrate.
The reference voltage pins (AVRH, AVRL) are also provided for A/D conversion. Refer to Section 13.1, "A/D Converter", for the analog reference voltage.

Noise on the analog power lines decrease the A/D and D/A converting precision, so use a stabilized power supply and make the board pattern with consideration given to that. Refer to Chapter 17 for precautions in making a pattern.

2.2 I/O Level

For the output interface of the E0C33A104, the VDD voltage is used as high level and the VSS voltage as low level. For the input interface, the input threshold level can be set using the VS5V pin.

Table 2.2.1 Input Interface Level Setup Pin

Pin name	Pin No.		Pull-up	I/O	Function
	QFP5-128	QFP15-128			
VS5V	7	4	–	I	Input interface level setup High: TTL level Low: CMOS level

Note: The VS5V pin does not have a pull-up resistor, therefore it must be connected to VDD or VSS. Furthermore, when using a 3.3 V of operating voltage, do not set the VS5V pin to high.

3 Initial Reset

This chapter explains the initial reset for the E0C33A104.

3.1 Pins for Initial Reset

Table 3.1.1 shows the pins used for initial reset.

Table 3.1.1 Pins for Initial Reset

Pin name	Pin No.		Pull-up	I/O	Function
	QFP5-128	QFP15-128			
#RESET	11	8	Built-in	I	Initial reset input pin Low: Resets the CPU.
#NMI	8	5	–	I	NMI request pin This pin is also used for selecting a reset method. High: Cold start Low: Hot start
BTA3	53	50	–	I	Boot address setup pin Low: Area 10

The E0C33A104 is reset when the #RESET pin goes low and starts operating at the rising edge of the reset signal. The core CPU and internal peripheral circuits are initialized while the #RESET pin is low.

3.2 Cold Start and Hot Start

The E0C33A104 supports two initial reset methods: cold start and hot start. The #NMI pin is used with the #RESET pin to set this condition.

The differences between cold start and hot start are shown in Table 3.2.1.

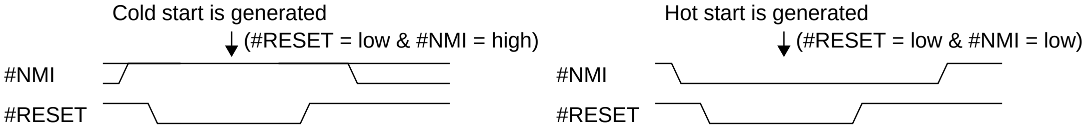
Table 3.2.1 Differences between Cold Start and Hot Start

Setup contents	Cold start	Hot start
Reset condition	#RESET = low & #NMI = high	#RESET = low & #NMI = low
CPU: PC	The vector at the boot address is loaded to the PC.	
CPU: PSR	All the PSR bits are reset to 0.	
CPU: Other registers	Undefined	
CPU: Operating clock	The CPU operates with the OSC3 clock.	
External bus status (0x40120–0x4013F)	Initialized	Status is retained.
Oscillation circuit	Both the OSC1 and OSC3 circuit start oscillating.	
I/O pin status (0x402C0–0x402DF)	Initialized	Status is retained.
Other peripheral circuit	Initialized or undefined	

Since cold start initializes all the internal peripheral circuits as well as the CPU, it is useful as a power-on reset.

Hot start initializes the CPU and peripheral circuits, but does not reset the bus control unit and the input, output and I/O port status. It is useful as a reset that maintains the external memory, external I/O and the port status.

The #NMI pin that specifies the reset method should be set following the timing chart shown in Figure 3.2.1.



#NMI must be set to high longer than the reset pulse width.

#NMI must be set to low longer than the reset pulse width.

(1) Cold start

(2) Hot start

Figure 3.2.1 Setup of #RESET and #NMI Pins

3.3 Power-on Reset

Be sure to reset (cold start) the E0C33A104 after turning on the power to start operating.
Since the #RESET pin has a built-in pull-up resister (approx. 160 kΩ), a power-on reset circuit is simply configured by connecting a capacitor as shown in Figure 3.3.1.

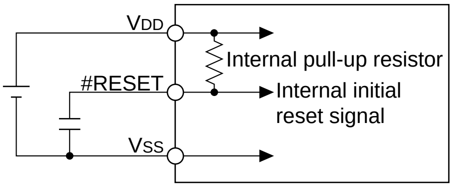


Figure 3.3.1 Power-on Reset Circuit

An initial reset (#RESET = low) turns the high-speed (OSC3) oscillation circuit on. The CPU starts operating with the OSC3 clock at the rising edge of the reset signal. The high-speed (OSC3) oscillation circuit takes time (10 ms max. under the standard condition in 3.3 V) for the oscillation to stabilize, therefore initial reset must be released after an appropriate oscillation-stabilization time has passed in order to start up the CPU without fault. The external capacitance should be decided so that the time constant of the capacitor and built-in pull-up resistor exceeds the oscillation-stabilization time.
Figure 3.3.2 shows a power-on reset timing chart.

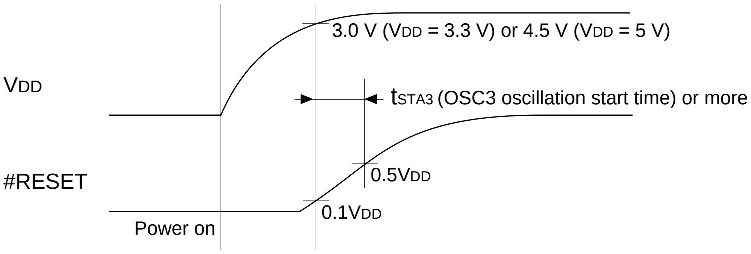


Figure 3.3.2 Power-on Reset Timing

Maintain the #RESET pin at 0.1•VDD or less (low level) after turning the power on until the supply voltage rises at least to the oscillation start voltage (3.0 V or 4.5 V). Furthermore, maintain the #RESET pin at 0.5•VDD or less until the high-speed (OSC3) oscillation circuit stabilizes oscillating.

Note: The OSC3 oscillation start time varies due to the elements used, board pattern and operating environment, therefore allow enough margin for the reset-release time. Refer to Section 18.8, "Oscillation Characteristics", in which an example of oscillation start time is provided.

3.4 Reset Pulse

A low pulse can be input to the #RESET pin for resetting the E0C33A104 being operated.
The minimum reset pulse width is provided in Section 18.7, "AC Characteristics". Be sure to input a pulse that has a pulse width longer than the minimum value.

To reset the E0C33A104 when the high-speed (OSC3) oscillation circuit is in off status, the pulse width must be extended until the oscillation stabilizes similarly to the power-on reset. Be aware that a short reset pulse may cause an operation error of the IC.

3.5 Boot Address

When the core CPU is initially reset, it reads the reset vector (program start address) from the boot address and loads the address to the PC (program counter). Then the CPU starts executing the program from the address when the #RESET pin goes high.
The boot address is decided according to the BTA3-pin setting.

Table 3.5.1 BTA3 Pin and Boot Address

Pin name	Pin No.		Pull-up	I/O	Setup level	Boot address
	QFP5-128	QFP15-128				
BTA3	53	50	–	I	Low	0x0C00000 (area 10)

Note: The BTA3 pin must be fixed at Low level.

The trap table in which trap vectors for interrupts and other trap factors are written also begins from the boot address by the default setting. (Refer to the "E0C33000 Core CPU Manual" for details of the trap table.)
The trap table base address can also be changed to a 1KB boundary address using the TTBR register (0x48134 to 0x48137).

3.6 Notes Related to Initial Reset

Core CPU

Since the all registers except for the PC and PSR are indeterminate at initial reset, they should be initialized by a program. In particular, the SP (stack pointer) must be initialized before accessing the stack area. NMI requests are disabled until any value is written to the SP. The initialization is necessary when the CPU is hot-started.

Internal RAM

The contents of the internal RAM are indeterminate at initial reset. Initialize the area to be used if necessary.

High-speed (OSC3) oscillation circuit

An initial reset activates the high-speed (OSC3) oscillation circuit and the CPU starts operating with the OSC3 clock after the initial reset is released. In order to prevent a malfunction of the CPU due to an unstabilized clock, the #RESET pin must be maintained at low until the OSC3 oscillation stabilizes when performing a power-on reset or resetting while the high-speed (OSC3) oscillation circuit is stopped. (Refer to Sections 3.3.)

Low-speed (OSC1) oscillation circuit

A power-on reset or an initial reset when the low-speed (OSC1) oscillation circuit is off starts the OSC1 oscillation. The low-speed (OSC1) oscillation circuit takes a longer stabilization time (3 sec max. under the standard condition) than the high-speed (OSC3) oscillation circuit. In order to prevent a malfunction due to an unstabilized clock, do not use the OSC1 clock until the stabilization time has passed.

BCU (external system interface)

Cold-start initializes the control registers for the BCU (bus control unit). Therefore, it is necessary to set up all the bus conditions.
Hot-start retains the previous bus conditions before an initial reset.

Input/output ports and input/output pins

Cold start initializes the control and data registers for the input, output and I/O ports.
The pull-up resistors in the input ports are disconnected.
The R70–R77 output ports output "0" (low level) and the R81–R83 ports output "1" (high level).
The I/O ports are set in input mode. The pull-up resistors in the I/O ports are disconnected.
Hot start retains the contents of the control registers and input/output pin status before an initial reset. However, when the pins are used for the internal peripheral circuits, it is necessary to set up the control registers of the peripheral circuit because they are initialized by an initial reset.

Other internal peripheral circuits

The control and data registers of peripheral circuits other than those listed above are initialized with the predefined values or become indeterminate regardless of the reset method (cold start or hot start). Therefore, it is necessary to set up the peripheral circuit conditions.
Refer to the I/O maps or explanation of each peripheral circuit section for initial settings of the peripheral circuits.

4 CPU and Operating Modes

This chapter explains the core CPU and the operating modes.

4.1 CPU

The E0C33A104 employs the E0C33000 32-bit RISC type CPU as the core CPU.
Since the E0C33A104 has a built-in multiplier, all instructions (105 instructions) in the E0C33000 instruction set including the MAC (multiplication and accumulation) instruction and the multiplication/division instructions are available.
All the internal registers of the E0C33000 can be used. The internal register can handle 28-bit addresses. However, the E0C33A104 has a 24-bit external address bus (A[0:23]), so the low-order 24 bits of address data can only be delivered to the external address bus.

Refer to the "E0C33000 Core CPU Manual" for details of the E0C33000.

Note: **MAC execution error**

<Descriptions of the problem>
The result of the MAC instructions may be incorrect when the following instructions appear right before the MAC instruction. The following instructions modify the %ALR and %AHR registers and this causes incorrect MAC results.
"ld.w %ALR, %Rs" "div*" "ld.w %ALR, %Rs" "mlt*"

<How to avoid this problem>
Do not place the above instructions right before the MAC instructions.

Load and data-size conversion instruction instruction error

<Descriptions of the problem>
CASE 1: When executing instructions in area 3 or area 0 (internal RAM), the following combination of instruction execution may result in incorrect register values to the destination register of the first instruction.

```
ld.ub  %r12, [%r9]      ld.ub  %r12, [%r9]        ld.h   %r12, [%r5]       ld.w  %r12, [%r6]
                     or                        or                       or
ld.h   %r10, %r12       ld.h   %r10, [%r12]       ld.b   [%r3], %r12       jp    %r12
```

The %r12 may be incorrect.
Notice that the source register of the second instruction is the destination register of the first instruction and the second instruction is a data-size conversion instruction, load/store instruction or branch instruction.

CASE 2: When turning on the CBR refresh cycles for the external DRAM access, the above instruction combination may result in incorrect register values even if the instructions are fetched from an external memory.

CASE 3: When the source register and destination register of the data-size conversion instruction is the same, the result may be incorrect.

CASE 4: When executing a data-size conversion instruction, the results become invalid and a debug environment like ICE will not work properly.

<How to avoid this problem>
There is a "FILTER" program in the CC33 software development package. Carefully go through the readme file in the CC33\utility\filter directory before using.

4.2 Standby Mode

The E0C33A104 supports three standby modes: two HALT modes and a SLEEP mode.
By setting the E0C33A104 in the standby mode, power consumption can greatly be reduced.

4.2.1 HALT Mode

When the CPU executes the halt instruction, it suspends the program execution and enters the HALT mode.
The E0C33A104 supports two types of HALT modes (basic HALT mode and HALT2 mode) and either can be selected using the HLT2OP (D3) / Clock option register (0x40150).
The CPU stops operating in basic HALT mode, so the amount of current consumption can be reduced. The internal peripheral circuits including the oscillation circuit keep operating in basic HALT mode.
HALT2 mode stops the external bus control functions including DMA and the bus clock as well as the CPU similar to basic HALT mode. Consequently, HALT2 mode realizes more power saving than the basic HALT mode.
The HALT mode is canceled by an initial reset or an interrupt including NMI. This mode is useful for saving power when waiting for an external input or completion of the peripheral circuit operations that do not need to execute the CPU.
The CPU transits to program execution status through trap processing when the HALT mode is canceled by an interrupt and executes the interrupt processing routine. The trap processing of the CPU saves the address of the instruction that follows the executed halt instruction into the stack. Therefore, when the interrupt processing routine is terminated by the reti instruction, the program flow returns to the instruction that follows the halt instruction.
Note that the HALT mode cannot be canceled with an interrupt factor except for reset and NMI if the PSR is set into interrupt disabled status.

4.2.2 SLEEP Mode

When the CPU executes the slp instruction, it suspends the program execution and enters SLEEP mode.
In SLEEP mode, the CPU and the internal peripheral circuits including the high-speed (OSC3) oscillation circuit stop operating. Thus SLEEP mode can greatly reduce current consumption in comparison to HALT mode. Moreover, the low-speed (OSC1) oscillation circuit and clock timer do not stop operating. The clock function keeps operating in SLEEP mode.
SLEEP mode is canceled by an initial reset or an interrupt (NMI, clock timer interrupt, external interrupt such as a key entry). Note that other interrupts by the internal peripheral circuits that use the OSC3 clock cannot be used for canceling SLEEP mode.
The CPU transits to program execution status through trap processing when the SLEEP mode is canceled by an interrupt and executes the interrupt processing routine. The trap processing of the CPU saves the address of the instruction that follows the executed slp instruction into the stack. Therefore, when the interrupt processing routine is terminated by the reti instruction, the program flow returns to the instruction that follows the slp instruction.
Note that SLEEP mode cannot be canceled with an interrupt factor except for reset and NMI if the PSR is set into interrupt disabled status.

4.2.3 Notes on Standby Mode

Interrupts

The standby mode can be canceled by an interrupt. Therefore, it is necessary to enable the interrupt to be used for canceling the standby mode before setting the CPU in the standby mode. It is also necessary to set the IE (interrupt enable) and IL (interrupt level) bits in the PSR to a condition that can accept the interrupt. Otherwise, the standby mode cannot be canceled even when an interrupt occurs. Refer to Chapter 8, "Interrupt", for interrupt settings.

Oscillation circuit

The high-speed (OSC3) oscillation circuit stops in SLEEP mode and restarts oscillating when SLEEP mode is canceled. If the CPU had operated with the OSC3 clock before entering SLEEP mode, the CPU restarts operating with the OSC3 clock immediately after canceling SLEEP mode. However, the OSC3 oscillation needs appropriate stabilization time (10 ms max. under the standard condition in 3.3 V). To restart the CPU after the oscillation stabilizes, a programmable interval can be inserted between cancellation of SLEEP mode and starting the CPU operation. Refer to Section 10.1, "Oscillation Circuits", for details.

The oscillation start time of the high-speed (OSC3) oscillation circuit varies according to the components to be used, board pattern and operating environment. The interval must be set to allow enough margin. Refer to Section 18.8, "Oscillation Characteristics", for examples of oscillation start time.

BCU (external bus)

When the CPU enters the standby mode, the BCU (bus control unit) stops after the current bus cycle has completed. All the chip enable signals are negated.
In basic HALT mode, the BCLK (bus clock) signal is output and DRAM refresh cycles are generated. DMA also operates.
In HALT2 or SLEEP mode, the BCLK signal stops, therefore DRAM refresh cycles cannot be generated and DMA stops.

Additional

The contents of the CPU registers and input/output port status are retained in the standby mode. Almost all control and data registers of the internal peripheral circuits are also retained, note, however, some registers may be changed at the transition to SLEEP mode. Refer to the section of each peripheral circuit for other precautions.

4.3 Test Mode

The E0C33A104 has two input pins for testing the IC: #TSTMD and #TSTMD1.

Table 4.3.1 Test Pins

Pin name	Pin No.		Pull-up	I/O	Function
	QFP5-128	QFP15-128			
#TSTMD	10	7	–	I	Test input (Fixed at high during normal operation.)
#TSTMD1	56	53	–	I	Test input (Fixed at high during normal operation.)

These pins must be fixed at high level during normal operation.

4.4 High-impedance Mode

It is possible to make E0C33A104 pins high-impedance status with the following method.
By holding the #RESET pin to "0" and pulling down the #TESTMD pin from "1" to "0", all the pins except for the OSC2, OSC4 and BCLK pins go into high-impedance status. This status will be maintained while both the #RESET and #TESTMD pins are "0".

Note:
- All pull up registers on the chip will be OFF including the #RESET pin.
- The high speed and low speed oscillators are both turned off in this mode.

4.5 Debug Mode

The E0C33A104 supports the debug mode.
The debug mode is a core CPU function, and realizes single step operation and break functions in the chip itself. Refer to the "E0C33000 Core CPU Manual" for details of the debug mode and the functions.

Area 2 in the memory map can only be accessed in the debug mode.
In the debug mode, the OSC3 clock is used as the CPU operating clock. Therefore, do not stop the high-speed (OSC3) oscillation circuit when using the debugging functions. Furthermore, only the CPU and BCU operate in the debug mode and other internal peripheral circuits (except for oscillation circuit) stop operating.

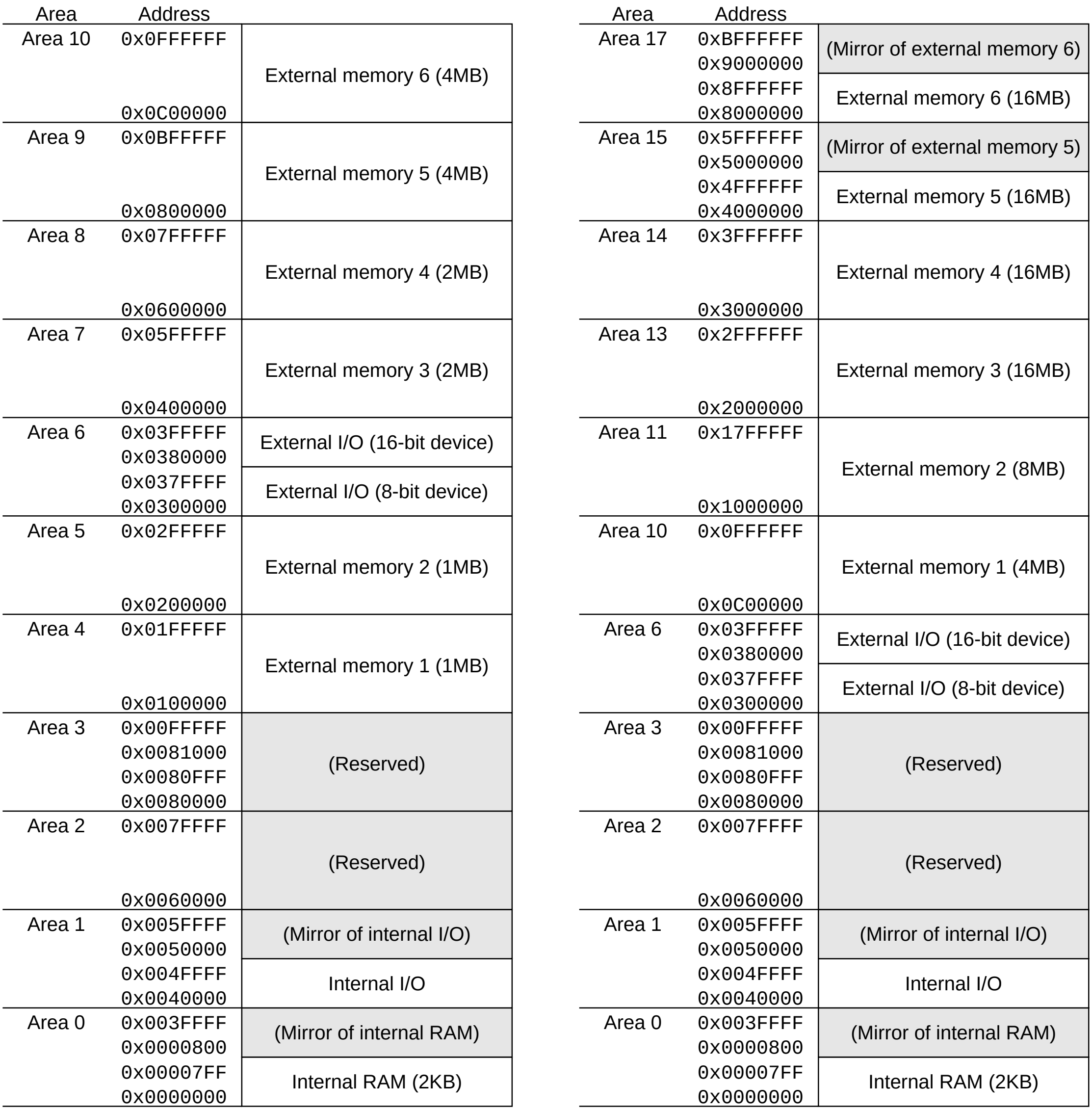
5 Address Space

This chapter shows the E0C33A104's memory map and the I/O map for the internal peripheral circuits.

5.1 Memory Map

The E0C33A104 has a 24-bit address bus allowing access to 16MB of linear address space (areas 0 to 10). Furthermore, the address space can be extended by remapping the #CE signals that are used to access areas 4 to 10 to the high-order areas.

Figure 5.1.1 shows the memory map configuration.



Area	Address	
Area 10	0x0FFFFFF 0x0C00000	External memory 6 (4MB)
Area 9	0x0BFFFFF 0x0800000	External memory 5 (4MB)
Area 8	0x07FFFFF 0x0600000	External memory 4 (2MB)
Area 7	0x05FFFFF 0x0400000	External memory 3 (2MB)
Area 6	0x03FFFFF 0x0380000	External I/O (16-bit device)
	0x037FFFF 0x0300000	External I/O (8-bit device)
Area 5	0x02FFFFF 0x0200000	External memory 2 (1MB)
Area 4	0x01FFFFF 0x0100000	External memory 1 (1MB)
Area 3	0x00FFFFF 0x0081000 0x0080FFF 0x0080000	(Reserved)
Area 2	0x007FFFF 0x0060000	(Reserved)
Area 1	0x005FFFF 0x0050000	(Mirror of internal I/O)
	0x004FFFF 0x0040000	Internal I/O
Area 0	0x003FFFF 0x0000800	(Mirror of internal RAM)
	0x00007FF 0x0000000	Internal RAM (2KB)

When #CE4–10 are used

Area	Address	
Area 17	0xBFFFFFF 0x9000000	(Mirror of external memory 6)
	0x8FFFFFF 0x8000000	External memory 6 (16MB)
Area 15	0x5FFFFFF 0x5000000	(Mirror of external memory 5)
	0x4FFFFFF 0x4000000	External memory 5 (16MB)
Area 14	0x3FFFFFF 0x3000000	External memory 4 (16MB)
Area 13	0x2FFFFFF 0x2000000	External memory 3 (16MB)
Area 11	0x17FFFFF 0x1000000	External memory 2 (8MB)
Area 10	0x0FFFFFF 0x0C00000	External memory 1 (4MB)
Area 6	0x03FFFFF 0x0380000	External I/O (16-bit device)
	0x037FFFF 0x0300000	External I/O (8-bit device)
Area 3	0x00FFFFF 0x0081000 0x0080FFF 0x0080000	(Reserved)
Area 2	0x007FFFF 0x0060000	(Reserved)
Area 1	0x005FFFF 0x0050000	(Mirror of internal I/O)
	0x004FFFF 0x0040000	Internal I/O
Area 0	0x003FFFF 0x0000800	(Mirror of internal RAM)
	0x00007FF 0x0000000	Internal RAM (2KB)

When #CE6, 10, 11, 13–15 and 17 are used

Figure 5.1.1 Memory Map

The internal memory configuration of the E0C33A104 changes according to the ARAM mode to be used. Refer to Chapter 6, "Internal Memory", for details.

For switching the #CE signals and setting the external memory areas, refer to Chapter 7, "External System Interface".

5.2 I/O Map for Internal Peripheral Circuits

Table 5.2.1 Internal I/O Map

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Power control register	**0040140 (B)**	D7	**CLKDT1**	System clock division ratio selection	CLKDT[1:0]			Division ratio	0	R/W	
		D6	**CLKDT0**		1	1		1/8	0		
					1	0		1/4			
					0	1		1/2			
					0	0		1/1			
		D5	**PSCON**	Prescaler On/Off control	1	On	0	Off	1	R/W	
		D4–3	–	reserved	–				0	–	Do not write 1.
		D2	**CLKCHG**	CPU operating clock switch	1	OSC3	0	OSC1	1	R/W	
		D1	**SOSC3**	High-speed (OSC3) oscillation On/Off	1	On	0	Off	1	R/W	
		D0	**SOSC1**	Low-speed (OSC1) oscillation On/Off	1	On	0	Off	1	R/W	
16-bit timer 0x clock control register	**0040147 (B)**	D7	**P16TON01**	16-bit timer 01 clock control	1	On	0	Off	0	R/W	
		D6	**P16TS012**	16-bit timer 01 clock division ratio selection	P16TS01[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D5	**P16TS011**		1	1	1	θ/4096	0		
		D4	**P16TS010**		1	1	0	θ/1024	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/16			
					0	0	0	θ/4			
		D3	**P16TON00**	16-bit timer 00 clock control	1	On	0	Off	0	R/W	
		D2	**P16TS002**	16-bit timer 00 clock division ratio selection	P16TS00[2:0]			Division ratio	0	R/W	θ: OSC3 clock 16-bit timer 0 can be used as a watchdog timer.
		D1	**P16TS001**		1	1	1	θ/4096	0		
		D0	**P16TS000**		1	1	0	θ/1024	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/16			
					0	0	0	θ/4			
16-bit timer 1x clock control register	**0040148 (B)**	D7	**P16TON11**	16-bit timer 11 clock control	1	On	0	Off	0	R/W	
		D6	**P16TS112**	16-bit timer 11 clock division ratio selection	P16TS11[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D5	**P16TS111**		1	1	1	θ/2048	0		
		D4	**P16TS110**		1	1	0	θ/512	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/8			
					0	0	0	θ/2			
		D3	**P16TON10**	16-bit timer 10 clock control	1	On	0	Off	0	R/W	
		D2	**P16TS102**	16-bit timer 10 clock division ratio selection	P16TS10[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D1	**P16TS101**		1	1	1	θ/2048	0		
		D0	**P16TS100**		1	1	0	θ/512	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/8			
					0	0	0	θ/2			

The meaning of the symbols described in [Init.] are listed below:

0, 1: Initial values that are set at initial reset.
(However, the registers for the bus and input/output ports are not initialized at hot start.)

X: Not initialized at initial reset.

–: Not set in the circuit.

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 2x clock control register	**0040149 (B)**	D7	**P16TON21**	16-bit timer 21 clock control	1	On	0	Off	0	R/W	
		D6	**P16TS212**	16-bit timer 21	P16TS21[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D5	**P16TS211**	clock division ratio selection	1	1	1	θ/4096	0		
		D4	**P16TS210**		1	1	0	θ/1024	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/16			
					0	0	0	θ/4			
		D3	**P16TON20**	16-bit timer 20 clock control	1	On	0	Off	0	R/W	
		D2	**P16TS202**	16-bit timer 20	P16TS20[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D1	**P16TS201**	clock division ratio selection	1	1	1	θ/4096	0		
		D0	**P16TS200**		1	1	0	θ/1024	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/16			
					0	0	0	θ/4			
16-bit timer 3x clock control register	**004014A (B)**	D7	**P16TON31**	16-bit timer 31 clock control	1	On	0	Off	0	R/W	
		D6	**P16TS312**	16-bit timer 31	P16TS31[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D5	**P16TS311**	clock division ratio selection	1	1	1	θ/2048	0		
		D4	**P16TS310**		1	1	0	θ/512	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/8			
					0	0	0	θ/2			
		D3	**P16TON30**	16-bit timer 30 clock control	1	On	0	Off	0	R/W	
		D2	**P16TS302**	16-bit timer 30	P16TS30[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D1	**P16TS301**	clock division ratio selection	1	1	1	θ/2048	0		
		D0	**P16TS300**		1	1	0	θ/512	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/8			
					0	0	0	θ/2			
16-bit timer 4x clock control register	**004014B (B)**	D7	**P16TON41**	16-bit timer 41 clock control	1	On	0	Off	0	R/W	
		D6	**P16TS412**	16-bit timer 41	P16TS41[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D5	**P16TS411**	clock division ratio selection	1	1	1	θ/4096	0		
		D4	**P16TS410**		1	1	0	θ/1024	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/16			
					0	0	0	θ/4			
		D3	**P16TON40**	16-bit timer 40 clock control	1	On	0	Off	0	R/W	
		D2	**P16TS402**	16-bit timer 40	P16TS40[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D1	**P16TS401**	clock division ratio selection	1	1	1	θ/4096	0		
		D0	**P16TS400**		1	1	0	θ/1024	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/16			
					0	0	0	θ/4			

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 5x	004014C	D7	P16TON51	16-bit timer 51 clock control	1	On	0	Off	0	R/W	
clock control	(B)	D6	P16TS512	16-bit timer 51	P16TS51[2:0]			Division ratio	0	R/W	θ: OSC3 clock
register		D5	P16TS511	clock division ratio selection	1	1	1	θ/2048	0		
		D4	P16TS510		1	1	0	θ/512	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/8			
					0	0	0	θ/2			
		D3	P16TON50	16-bit timer 50 clock control	1	On	0	Off	0	R/W	
		D2	P16TS502	16-bit timer 50	P16TS50[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D1	P16TS501	clock division ratio selection	1	1	1	θ/2048	0		
		D0	P16TS500		1	1	0	θ/512	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/8			
					0	0	0	θ/2			
8-bit timer 0/1	004014D	D7	P8TON1	8-bit timer 1 clock control	1	On	0	Off	0	R/W	
clock control	(B)	D6	P8TS12	8-bit timer 1	P8TS1[2:0]			Division ratio	0	R/W	θ: OSC3 clock
register		D5	P8TS11	clock division ratio selection	1	1	1	θ/4096	0		8-bit timer 1 can
		D4	P8TS10		1	1	0	θ/2048	0		generate the OSC3
					1	0	1	θ/1024			oscillation-stabilize
					1	0	0	θ/512			waiting period.
					0	1	1	θ/256			
					0	1	0	θ/128			
					0	0	1	θ/64			
					0	0	0	θ/32			
		D3	P8TON0	8-bit timer 0 clock control	1	On	0	Off	0	R/W	
		D2	P8TS02	8-bit timer 0	P8TS0[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D1	P8TS01	clock division ratio selection	1	1	1	θ/256	0		8-bit timer 0 can
		D0	P8TS00		1	1	0	θ/128	0		generate the DRAM
					1	0	1	θ/64			refresh clock.
					1	0	0	θ/32			
					0	1	1	θ/16			
					0	1	0	θ/8			
					0	0	1	θ/4			
					0	0	0	θ/2			
8-bit timer 2/3	004014E	D7	P8TON3	8-bit timer 3 clock control	1	On	0	Off	0	R/W	
clock control	(B)	D6	P8TS32	8-bit timer 3	P8TS3[2:0]			Division ratio	0	R/W	θ: OSC3 clock
register		D5	P8TS31	clock division ratio selection	1	1	1	θ/256	0		8-bit timer 3 can
		D4	P8TS30		1	1	0	θ/128	0		generate the clock for
					1	0	1	θ/64			the serial I/F Ch.1.
					1	0	0	θ/32			
					0	1	1	θ/16			
					0	1	0	θ/8			
					0	0	1	θ/4			
					0	0	0	θ/2			
		D3	P8TON2	8-bit timer 2 clock control	1	On	0	Off	0	R/W	
		D2	P8TS22	8-bit timer 2	P8TS2[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D1	P8TS21	clock division ratio selection	1	1	1	θ/4096	0		8-bit timer 2 can
		D0	P8TS20		1	1	0	θ/2048	0		generate the clock for
					1	0	1	θ/64			the serial I/F Ch.0.
					1	0	0	θ/32			
					0	1	1	θ/16			
					0	1	0	θ/8			
					0	0	1	θ/4			
					0	0	0	θ/2			
A/D clock	004014F	D7–4	–	–	–				–	–	0 when being read.
control register	(B)	D3	PSONAD	A/D converter clock control	1	On	0	Off	0	R/W	
		D2	PSAD2	A/D converter clock division ratio	P8TS0[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D1	PSAD1	selection	1	1	1	θ/256	0		
		D0	PSAD0		1	1	0	θ/128	0		
					1	0	1	θ/64			
					1	0	0	θ/32			
					0	1	1	θ/16			
					0	1	0	θ/8			
					0	0	1	θ/4			
					0	0	0	θ/2			

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Clock option register	**0040150 (B)**	D7–4	–	–	–				–	–	0 when being read.
		D3	**HLT2OP**	HALT clock option	1	On	0	Off	0	R/W	
		D2	**8T1ON**	OSC3-stabilize waiting function	1	Off	0	On	1	R/W	
		D1	–	reserved	–				0	–	Do not write 1.
		D0	**PF1ON**	OSC1 external output control	1	On	0	Off	0	R/W	
Clock timer Run/Stop register	**0040151 (B)**	D7–2	–	reserved	–				–	–	0 when being read.
		D1	**TCRST**	Clock timer reset	1	Reset	0	Invalid	X	W	0 when being read.
		D0	**TCRUN**	Clock timer Run/Stop control	1	Run	0	Stop	X	R/W	
Clock timer interrupt control register	**0040152 (B)**	D7	**TCISE2**	Clock timer interrupt factor selection	TCISE[2:0]			Interrupt factor	X	R/W	
		D6	**TCISE1**		1	1	1	None	X		
		D5	**TCISE0**		1	1	0	Day	X		
					1	0	1	Hour			
					1	0	0	Minute			
					0	1	1	1 Hz			
					0	1	0	2 Hz			
					0	0	1	8 Hz			
					0	0	0	32 Hz			
		D4	**TCASE2**	Clock timer alarm factor selection	TCASE[2:0]			Alarm factor	X	R/W	
		D3	**TCASE1**		1	X	X	Day	X		
		D2	**TCASE0**		X	1	X	Hour	X		
					X	X	1	Minute			
					0	0	0	None			
		D1	**TCIF**	Interrupt factor generation flag	1	Generated	0	Not generated	X	R/W	Reset by writing 1.
		D0	**TCAF**	Alarm factor generation flag	1	Generated	0	Not generated	X	R/W	Reset by writing 1.
Clock timer divider register	**0040153 (B)**	D7	**TCD7**	Clock timer data 1 Hz	1	High	0	Low	X	R	
		D6	**TCD6**	Clock timer data 2 Hz	1	High	0	Low	X	R	
		D5	**TCD5**	Clock timer data 4 Hz	1	High	0	Low	X	R	
		D4	**TCD4**	Clock timer data 8 Hz	1	High	0	Low	X	R	
		D3	**TCD3**	Clock timer data 16 Hz	1	High	0	Low	X	R	
		D2	**TCD2**	Clock timer data 32 Hz	1	High	0	Low	X	R	
		D1	**TCD1**	Clock timer data 64 Hz	1	High	0	Low	X	R	
		D0	**TCD0**	Clock timer data 128 Hz	1	High	0	Low	X	R	
Clock timer second register	**0040154 (B)**	D7–6	–	reserved	–				–	–	0 when being read.
		D5	**TCMD5**	Clock timer second counter data TCMD5 = MSB TCMD0 = LSB	0 to 59 seconds				X	R	
		D4	**TCMD4**						X		
		D3	**TCMD3**						X		
		D2	**TCMD2**						X		
		D1	**TCMD1**						X		
		D0	**TCMD0**						X		
Clock timer minute register	**0040155 (B)**	D7–6	–	reserved	–				–	–	0 when being read.
		D5	**TCHD5**	Clock timer minute counter data TCHD5 = MSB TCHD0 = LSB	0 to 59 minutes				X	R/W	
		D4	**TCHD4**						X		
		D3	**TCHD3**						X		
		D2	**TCHD2**						X		
		D1	**TCHD1**						X		
		D0	**TCHD0**						X		
Clock timer hour register	**0040156 (B)**	D7–5	–	reserved	–				–	–	0 when being read.
		D4	**TCDD4**	Clock timer hour counter data TCDD4 = MSB TCDD0 = LSB	0 to 23 hours				X	R/W	
		D3	**TCDD3**						X		
		D2	**TCDD2**						X		
		D1	**TCDD1**						X		
		D0	**TCDD0**						X		
Clock timer day (low-order) register	**0040157 (B)**	D7	**TCND7**	Clock timer day counter data (low-order 8 bits) TCND0 = LSB	0 to 65535 days				X	R/W	
		D6	**TCND6**						X		
		D5	**TCND5**						X		
		D4	**TCND4**						X		
		D3	**TCND3**						X		
		D2	**TCND2**						X		
		D1	**TCND1**						X		
		D0	**TCND0**						X		
Clock timer day (high-order) register	**0040158 (B)**	D7	**TCND15**	Clock timer day counter data (high-order 8 bits) TCND15 = MSB					X	R/W	
		D6	**TCND14**						X		
		D5	**TCND13**						X		
		D4	**TCND12**						X		
		D3	**TCND11**						X		
		D2	**TCND10**						X		
		D1	**TCND9**						X		
		D0	**TCND8**						X		

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Clock timer minute comparison register	**0040159 (B)**	D7–6	**–**	reserved	–				–	–	0 when being read.
		D5	**TCCH5**	Clock timer minute comparison data TCCH5 = MSB TCCH0 = LSB	0 to 59 minutes (Note) Can be set within 0–63.				X	R/W	
		D4	**TCCH4**						X		
		D3	**TCCH3**						X		
		D2	**TCCH2**						X		
		D1	**TCCH1**						X		
		D0	**TCCH0**						X		
Clock timer hour comparison register	**004015A (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**TCCD4**	Clock timer hour comparison data TCCD4 = MSB TCCD0 = LSB	0 to 23 hours (Note) Can be set within 0–31.				X	R/W	
		D3	**TCCD3**						X		
		D2	**TCCD2**						X		
		D1	**TCCD1**						X		
		D0	**TCCD0**						X		
Clock timer day comparison register	**004015B (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**TCCN4**	Clock timer day comparison data TCCN4 = MSB TCCN0 = LSB	0 to 31 days				X	R/W	Compared with TCND[4:0].
		D3	**TCCN3**						X		
		D2	**TCCN2**						X		
		D1	**TCCN1**						X		
		D0	**TCCN0**						X		
Power control protect register	**004015E (B)**	D7	**CLGP7**	Power control register protect flag	Writing 10010110 (0x96) removes the write protection of the power control register (0x40140). Writing another value set the write protection.				0	R/W	
		D6	**CLGP6**						0		
		D5	**CLGP5**						0		
		D4	**CLGP4**						0		
		D3	**CLGP3**						0		
		D2	**CLGP2**						0		
		D1	**CLGP1**						0		
		D0	**CLGP0**						0		
Low-speed oscillation control register	**004015F (B)**	D7	**TCAOFF**	OSC1 auto-off function	1	On	0	Off	0	R/W	
		D6–3	**–**	reserved	–				–	–	0 when being read.
		D2	**TCHVOF**	CPU core power On/Off flag	1	Off	0	On	0	R/W	
		D1	**CLGRON**	Feedback resistor On/Off flag	1	Off	0	On	0	R/W	
		D0	**–**	reserved	–				1	–	Writing 0 not allowed.
8-bit timer 0 control register	**0040160 (B)**	D7–3	**–**	reserved	–				–	–	0 when being read.
		D2	**PTOUT0**	8-bit timer 0 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET0**	8-bit timer 0 preset	1	Preset	0	Invalid	–	W	0 when being read.
		D0	**PTRUN0**	8-bit timer 0 Run/Stop control	1	Run	0	Stop	0	R/W	
8-bit timer 0 reload data register	**0040161 (B)**	D7	**RLD07**	8-bit timer 0 reload data RLD07 = MSB RLD00 = LSB	0 to 255				X	R/W	
		D6	**RLD06**						X		
		D5	**RLD05**						X		
		D4	**RLD04**						X		
		D3	**RLD03**						X		
		D2	**RLD02**						X		
		D1	**RLD01**						X		
		D0	**RLD00**						X		
8-bit timer 0 counter data register	**0040162 (B)**	D7	**PTD07**	8-bit timer 0 counter data PTD07 = MSB PTD00 = LSB	0 to 255				X	R	
		D6	**PTD06**						X		
		D5	**PTD05**						X		
		D4	**PTD04**						X		
		D3	**PTD03**						X		
		D2	**PTD02**						X		
		D1	**PTD01**						X		
		D0	**PTD00**						X		
8-bit timer 1 control register	**0040164 (B)**	D7–3	**–**	reserved	–				–	–	0 when being read.
		D2	**PTOUT1**	8-bit timer 1 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET1**	8-bit timer 1 preset	1	Preset	0	Invalid	–	W	0 when being read.
		D0	**PTRUN1**	8-bit timer 1 Run/Stop control	1	Run	0	Stop	0	R/W	
8-bit timer 1 reload data register	**0040165 (B)**	D7	**RLD17**	8-bit timer 1 reload data RLD17 = MSB RLD10 = LSB	0 to 255				X	R/W	
		D6	**RLD16**						X		
		D5	**RLD15**						X		
		D4	**RLD14**						X		
		D3	**RLD13**						X		
		D2	**RLD12**						X		
		D1	**RLD11**						X		
		D0	**RLD10**						X		
8-bit timer 1 counter data register	**0040166 (B)**	D7	**PTD17**	8-bit timer 1 counter data PTD17 = MSB PTD10 = LSB	0 to 255				X	R	
		D6	**PTD16**						X		
		D5	**PTD15**						X		
		D4	**PTD14**						X		
		D3	**PTD13**						X		
		D2	**PTD12**						X		
		D1	**PTD11**						X		
		D0	**PTD10**						X		

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit timer 2 control register	**0040168 (B)**	D7–3	**–**	reserved	–				–	–	0 when being read.
		D2	**PTOUT2**	8-bit timer 2 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET2**	8-bit timer 2 preset	1	Preset	0	Invalid	–	W	0 when being read.
		D0	**PTRUN2**	8-bit timer 2 Run/Stop control	1	Run	0	Stop	0	R/W	
8-bit timer 2 reload data register	**0040169 (B)**	D7	**RLD27**	8-bit timer 2 reload data	0 to 255				X	R/W	
		D6	**RLD26**	RLD27 = MSB					X		
		D5	**RLD25**	RLD20 = LSB					X		
		D4	**RLD24**						X		
		D3	**RLD23**						X		
		D2	**RLD22**						X		
		D1	**RLD21**						X		
		D0	**RLD20**						X		
8-bit timer 2 counter data register	**004016A (B)**	D7	**PTD27**	8-bit timer 2 counter data	0 to 255				X	R	
		D6	**PTD26**	PTD27 = MSB					X		
		D5	**PTD25**	PTD20 = LSB					X		
		D4	**PTD24**						X		
		D3	**PTD23**						X		
		D2	**PTD22**						X		
		D1	**PTD21**						X		
		D0	**PTD20**						X		
8-bit timer 3 control register	**004016C (B)**	D7–3	**–**	reserved	–				–	–	0 when being read.
		D2	**PTOUT3**	8-bit timer 3 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET3**	8-bit timer 3 preset	1	Preset	0	Invalid	–	W	0 when being read.
		D0	**PTRUN3**	8-bit timer 3 Run/Stop control	1	Run	0	Stop	0	R/W	
8-bit timer 3 reload data register	**004016D (B)**	D7	**RLD37**	8-bit timer 3 reload data	0 to 255				X	R/W	
		D6	**RLD36**	RLD37 = MSB					X		
		D5	**RLD35**	RLD30 = LSB					X		
		D4	**RLD34**						X		
		D3	**RLD33**						X		
		D2	**RLD32**						X		
		D1	**RLD31**						X		
		D0	**RLD30**						X		
8-bit timer 3 counter data register	**004016E (B)**	D7	**PTD37**	8-bit timer 3 counter data	0 to 255				X	R	
		D6	**PTD36**	PTD37 = MSB					X		
		D5	**PTD35**	PTD30 = LSB					X		
		D4	**PTD34**						X		
		D3	**PTD33**						X		
		D2	**PTD32**						X		
		D1	**PTD31**						X		
		D0	**PTD30**						X		
Watchdog timer write-protect register	**0040170 (B)**	D7	**WRWD**	EWD write protection	1	Write enabled	0	Write-protect	0	W	
		D6–0	**–**	–	–				–	–	Read: Invalid
Watchdog timer enable register	**0040171 (B)**	D7–2	**–**	–	–				–	–	Read: Invalid
		D1	**EWD**	Watchdog timer enable	1	NMI enabled	0	NMI disabled	0	W	
		D0	**–**	–	–				–	–	Read: Invalid
16-bit timer 00 control register	**0040180 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**MODE16**	16-bit timer 0 mode selection	1	16 bits	0	8 bits	0	R/W	
		D3	**CKSL00**	16-bit timer 00 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	**PTM00**	16-bit timer 00 clock output control	1	On	0	Off	0	R/W	∗1
		D1	**PSET00**	16-bit timer 00 preset	1	Preset	0	Invalid	0	W	0 when being read.
		D0	**PRUN00**	16-bit timer 00 Run/Stop control	1	Run	0	Stop	0	R/W	
16-bit timer 01 control register	**0040181 (B)**	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**CKSL01**	16-bit timer 01 input clock selection	1	External clock	0	Internal clock	0	R/W	∗1
		D2	**PTM01**	16-bit timer 01 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET01**	16-bit timer 01 preset	1	Preset	0	Invalid	0	W	0 when being read.∗1
		D0	**PRUN01**	16-bit timer 01 Run/Stop control	1	Run	0	Stop	0	R/W	∗1
16-bit timer 00 reload data register	**0040182 (B)**	D7	**RR007**	16-bit timer 00 reload data	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR006**	RR007 = MSB					X		
		D5	**RR005**	RR000 = LSB					X		
		D4	**RR004**						X		
		D3	**RR003**						X		
		D2	**RR002**						X		
		D1	**RR001**						X		
		D0	**RR000**						X		

∗1: Invalid in 16-bit mode (fixed at 0).

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 01 reload data register	**0040183 (B)**	D7	**RR017**	16-bit timer 01 reload data	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR016**	RR017 = MSB					X		
		D5	**RR015**	RR010 = LSB					X		
		D4	**RR014**						X		
		D3	**RR013**						X		
		D2	**RR012**						X		
		D1	**RR011**						X		
		D0	**RR010**						X		
16-bit timer 00 comparison data register	**0040184 (B)**	D7	**CR007**	16-bit timer 00 comparison data	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR006**	CR007 = MSB					X		
		D5	**CR005**	CR000 = LSB					X		
		D4	**CR004**						X		
		D3	**CR003**						X		
		D2	**CR002**						X		
		D1	**CR001**						X		
		D0	**CR000**						X		
16-bit timer 01 comparison data register	**0040185 (B)**	D7	**CR017**	16-bit timer 01 comparison data	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR016**	CR017 = MSB					X		
		D5	**CR015**	CR010 = LSB					X		
		D4	**CR014**						X		
		D3	**CR013**						X		
		D2	**CR012**						X		
		D1	**CR011**						X		
		D0	**CR010**						X		
16-bit timer 00 counter data register	**0040186 (B)**	D7	**TC007**	16-bit timer 00 counter data	0 to 255				X	R	Used as the low-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC006**	TC007 = MSB					X		
		D5	**TC005**	TC000 = LSB					X		
		D4	**TC004**						X		
		D3	**TC003**						X		
		D2	**TC002**						X		
		D1	**TC001**						X		
		D0	**TC000**						X		
16-bit timer 01 counter data register	**0040187 (B)**	D7	**TC017**	16-bit timer 01 counter data	0 to 255				X	R	Used as the high-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC016**	TC017 = MSB					X		
		D5	**TC015**	TC010 = LSB					X		
		D4	**TC014**						X		
		D3	**TC013**						X		
		D2	**TC012**						X		
		D1	**TC011**						X		
		D0	**TC010**						X		
16-bit timer 10 control register	**0040188 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**MODE16**	16-bit timer 1 mode selection	1	16 bits	0	8 bits	0	R/W	
		D3	**CKSL10**	16-bit timer 10 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	**PTM10**	16-bit timer 10 clock output control	1	On	0	Off	0	R/W	∗1
		D1	**PSET10**	16-bit timer 10 preset	1	Preset	0	Invalid	0	W	0 when being read.
		D0	**PRUN10**	16-bit timer 10 Run/Stop control	1	Run	0	Stop	0	R/W	
16-bit timer 11 control register	**0040189 (B)**	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**CKSL11**	16-bit timer 11 input clock selection	1	Invalid	0	Internal clock	0	R/W	∗2
		D2	**PTM11**	16-bit timer 11 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET11**	16-bit timer 11 preset	1	Preset	0	Invalid	0	W	0 when being read.∗1
		D0	**PRUN11**	16-bit timer 11 Run/Stop control	1	Run	0	Stop	0	R/W	∗1
16-bit timer 10 reload data register	**004018A (B)**	D7	**RR107**	16-bit timer 10 reload data	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR106**	RR107 = MSB					X		
		D5	**RR105**	RR100 = LSB					X		
		D4	**RR104**						X		
		D3	**RR103**						X		
		D2	**RR102**						X		
		D1	**RR101**						X		
		D0	**RR100**						X		
16-bit timer 11 reload data register	**004018B (B)**	D7	**RR117**	16-bit timer 11 reload data	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR116**	RR117 = MSB					X		
		D5	**RR115**	RR110 = LSB					X		
		D4	**RR114**						X		
		D3	**RR113**						X		
		D2	**RR112**						X		
		D1	**RR111**						X		
		D0	**RR110**						X		

∗1: Invalid in 16-bit mode (fixed at 0).

∗2: Invalid in 16-bit mode (fixed at 0). The timer does not work if "1" (external clock) is set in 8-bit mode.

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 10 comparison data register	**004018C (B)**	D7	**CR107**	16-bit timer 10 comparison data	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR106**	CR107 = MSB					X		
		D5	**CR105**	CR100 = LSB					X		
		D4	**CR104**						X		
		D3	**CR103**						X		
		D2	**CR102**						X		
		D1	**CR101**						X		
		D0	**CR100**						X		
16-bit timer 11 comparison data register	**004018D (B)**	D7	**CR117**	16-bit timer 11 comparison data	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR116**	CR117 = MSB					X		
		D5	**CR115**	CR110 = LSB					X		
		D4	**CR114**						X		
		D3	**CR113**						X		
		D2	**CR112**						X		
		D1	**CR111**						X		
		D0	**CR110**						X		
16-bit timer 10 counter data register	**004018E (B)**	D7	**TC107**	16-bit timer 10 counter data	0 to 255				X	R	Used as the low-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC106**	TC107 = MSB					X		
		D5	**TC105**	TC100 = LSB					X		
		D4	**TC104**						X		
		D3	**TC103**						X		
		D2	**TC102**						X		
		D1	**TC101**						X		
		D0	**TC100**						X		
16-bit timer 11 counter data register	**004018F (B)**	D7	**TC117**	16-bit timer 11 counter data	0 to 255				X	R	Used as the high-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC116**	TC117 = MSB					X		
		D5	**TC115**	TC110 = LSB					X		
		D4	**TC114**						X		
		D3	**TC113**						X		
		D2	**TC112**						X		
		D1	**TC111**						X		
		D0	**TC110**						X		
16-bit timer 20 control register	**0040190 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**MODE16**	16-bit timer 2 mode selection	1	16 bits	0	8 bits	0	R/W	
		D3	**CKSL20**	16-bit timer 20 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	**PTM20**	16-bit timer 20 clock output control	1	Invalid	0	Off	0	R/W	∗3
		D1	**PSET20**	16-bit timer 20 preset	1	Preset	0	Invalid	0	W	0 when being read.
		D0	**PRUN20**	16-bit timer 20 Run/Stop control	1	Run	0	Stop	0	R/W	
16-bit timer 21 control register	**0040191 (B)**	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**CKSL21**	16-bit timer 21 input clock selection	1	Invalid	0	Internal clock	0	R/W	∗2
		D2	**PTM21**	16-bit timer 21 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET21**	16-bit timer 21 preset	1	Preset	0	Invalid	0	W	0 when being read.∗1
		D0	**PRUN21**	16-bit timer 21 Run/Stop control	1	Run	0	Stop	0	R/W	∗1
16-bit timer 20 reload data register	**0040192 (B)**	D7	**RR207**	16-bit timer 20 reload data	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR206**	RR207 = MSB					X		
		D5	**RR205**	RR200 = LSB					X		
		D4	**RR204**						X		
		D3	**RR203**						X		
		D2	**RR202**						X		
		D1	**RR201**						X		
		D0	**RR200**						X		
16-bit timer 21 reload data register	**0040193 (B)**	D7	**RR217**	16-bit timer 21 reload data	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR216**	RR217 = MSB					X		
		D5	**RR215**	RR210 = LSB					X		
		D4	**RR214**						X		
		D3	**RR213**						X		
		D2	**RR212**						X		
		D1	**RR211**						X		
		D0	**RR210**						X		
16-bit timer 20 comparison data register	**0040194 (B)**	D7	**CR207**	16-bit timer 20 comparison data	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR206**	CR207 = MSB					X		
		D5	**CR205**	CR200 = LSB					X		
		D4	**CR204**						X		
		D3	**CR203**						X		
		D2	**CR202**						X		
		D1	**CR201**						X		
		D0	**CR200**						X		

∗1: Invalid in 16-bit mode (fixed at 0).

∗2: Invalid in 16-bit mode (fixed at 0). The timer does not work if "1" (external clock) is set in 8-bit mode.

∗3: This timer has no clock output terminal, so a clock cannot be output even if this bit is set to "1" (output On).

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 21 comparison data register	**0040195 (B)**	D7	**CR217**	16-bit timer 21 comparison data	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR216**	CR217 = MSB					X		
		D5	**CR215**	CR210 = LSB					X		
		D4	**CR214**						X		
		D3	**CR213**						X		
		D2	**CR212**						X		
		D1	**CR211**						X		
		D0	**CR210**						X		
16-bit timer 20 counter data register	**0040196 (B)**	D7	**TC207**	16-bit timer 20 counter data	0 to 255				X	R	Used as the low-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC206**	TC207 = MSB					X		
		D5	**TC205**	TC200 = LSB					X		
		D4	**TC204**						X		
		D3	**TC203**						X		
		D2	**TC202**						X		
		D1	**TC201**						X		
		D0	**TC200**						X		
16-bit timer 21 counter data register	**0040197 (B)**	D7	**TC217**	16-bit timer 21 counter data	0 to 255				X	R	Used as the high-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC216**	TC217 = MSB					X		
		D5	**TC215**	TC210 = LSB					X		
		D4	**TC214**						X		
		D3	**TC213**						X		
		D2	**TC212**						X		
		D1	**TC211**						X		
		D0	**TC210**						X		
16-bit timer 30 control register	**0040198 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**MODE16**	16-bit timer 3 mode selection	1	16 bits	0	8 bits	0	R/W	
		D3	**CKSL30**	16-bit timer 30 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	**PTM30**	16-bit timer 30 clock output control	1	Invalid	0	Off	0	R/W	∗3
		D1	**PSET30**	16-bit timer 30 preset	1	Preset	0	Invalid	0	W	0 when being read.
		D0	**PRUN30**	16-bit timer 30 Run/Stop control	1	Run	0	Stop	0	R/W	
16-bit timer 31 control register	**0040199 (B)**	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**CKSL31**	16-bit timer 31 input clock selection	1	Invalid	0	Internal clock	0	R/W	∗2
		D2	**PTM31**	16-bit timer 31 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET31**	16-bit timer 31 preset	1	Preset	0	Invalid	0	W	0 when being read.∗1
		D0	**PRUN31**	16-bit timer 31 Run/Stop control	1	Run	0	Stop	0	R/W	∗1
16-bit timer 30 reload data register	**004019A (B)**	D7	**RR307**	16-bit timer 30 reload data	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR306**	RR307 = MSB					X		
		D5	**RR305**	RR300 = LSB					X		
		D4	**RR304**						X		
		D3	**RR303**						X		
		D2	**RR302**						X		
		D1	**RR301**						X		
		D0	**RR300**						X		
16-bit timer 31 reload data register	**004019B (B)**	D7	**RR317**	16-bit timer 31 reload data	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR316**	RR317 = MSB					X		
		D5	**RR315**	RR310 = LSB					X		
		D4	**RR314**						X		
		D3	**RR313**						X		
		D2	**RR312**						X		
		D1	**RR311**						X		
		D0	**RR310**						X		
16-bit timer 30 comparison data register	**004019C (B)**	D7	**CR307**	16-bit timer 30 comparison data	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR306**	CR307 = MSB					X		
		D5	**CR305**	CR300 = LSB					X		
		D4	**CR304**						X		
		D3	**CR303**						X		
		D2	**CR302**						X		
		D1	**CR301**						X		
		D0	**CR300**						X		
16-bit timer 31 comparison data register	**004019D (B)**	D7	**CR317**	16-bit timer 31 comparison data	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR316**	CR317 = MSB					X		
		D5	**CR315**	CR310 = LSB					X		
		D4	**CR314**						X		
		D3	**CR313**						X		
		D2	**CR312**						X		
		D1	**CR311**						X		
		D0	**CR310**						X		

∗1: Invalid in 16-bit mode (fixed at 0).

∗2: Invalid in 16-bit mode (fixed at 0). The timer does not work if "1" (external clock) is set in 8-bit mode.

∗3: This timer has no clock output terminal, so a clock cannot be output even if this bit is set to "1" (output On).

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 30 counter data register	**004019E (B)**	D7	**TC307**	16-bit timer 30 counter data TC307 = MSB TC300 = LSB	0 to 255				X	R	Used as the low-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC306**						X		
		D5	**TC305**						X		
		D4	**TC304**						X		
		D3	**TC303**						X		
		D2	**TC302**						X		
		D1	**TC301**						X		
		D0	**TC300**						X		
16-bit timer 31 counter data register	**004019F (B)**	D7	**TC317**	16-bit timer 31 counter data TC317 = MSB TC310 = LSB	0 to 255				X	R	Used as the high-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC316**						X		
		D5	**TC315**						X		
		D4	**TC314**						X		
		D3	**TC313**						X		
		D2	**TC312**						X		
		D1	**TC311**						X		
		D0	**TC310**						X		
16-bit timer 40 control register	**00401A0 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**MODE16**	16-bit timer 4 mode selection	1	16 bits	0	8 bits	0	R/W	
		D3	**CKSL40**	16-bit timer 40 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	**PTM40**	16-bit timer 40 clock output control	1	Invalid	0	Off	0	R/W	∗3
		D1	**PSET40**	16-bit timer 40 preset	1	Preset	0	Invalid	0	W	0 when being read.
		D0	**PRUN40**	16-bit timer 40 Run/Stop control	1	Run	0	Stop	0	R/W	
16-bit timer 41 control register	**00401A1 (B)**	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**CKSL41**	16-bit timer 41 input clock selection	1	Invalid	0	Internal clock	0	R/W	∗2
		D2	**PTM41**	16-bit timer 41 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET41**	16-bit timer 41 preset	1	Preset	0	Invalid	0	W	0 when being read.∗1
		D0	**PRUN41**	16-bit timer 41 Run/Stop control	1	Run	0	Stop	0	R/W	∗1
16-bit timer 40 reload data register	**00401A2 (B)**	D7	**RR407**	16-bit timer 40 reload data RR407 = MSB RR400 = LSB	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR406**						X		
		D5	**RR405**						X		
		D4	**RR404**						X		
		D3	**RR403**						X		
		D2	**RR402**						X		
		D1	**RR401**						X		
		D0	**RR400**						X		
16-bit timer 41 reload data register	**00401A3 (B)**	D7	**RR417**	16-bit timer 41 reload data RR417 = MSB RR410 = LSB	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR416**						X		
		D5	**RR415**						X		
		D4	**RR414**						X		
		D3	**RR413**						X		
		D2	**RR412**						X		
		D1	**RR411**						X		
		D0	**RR410**						X		
16-bit timer 40 comparison data register	**00401A4 (B)**	D7	**CR407**	16-bit timer 40 comparison data CR407 = MSB CR400 = LSB	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR406**						X		
		D5	**CR405**						X		
		D4	**CR404**						X		
		D3	**CR403**						X		
		D2	**CR402**						X		
		D1	**CR401**						X		
		D0	**CR400**						X		
16-bit timer 41 comparison data register	**00401A5 (B)**	D7	**CR417**	16-bit timer 41 comparison data CR417 = MSB CR410 = LSB	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR416**						X		
		D5	**CR415**						X		
		D4	**CR414**						X		
		D3	**CR413**						X		
		D2	**CR412**						X		
		D1	**CR411**						X		
		D0	**CR410**						X		
16-bit timer 40 counter data register	**00401A6 (B)**	D7	**TC407**	16-bit timer 40 counter data TC407 = MSB TC400 = LSB	0 to 255				X	R	Used as the low-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC406**						X		
		D5	**TC405**						X		
		D4	**TC404**						X		
		D3	**TC403**						X		
		D2	**TC402**						X		
		D1	**TC401**						X		
		D0	**TC400**						X		

∗1: Invalid in 16-bit mode (fixed at 0).

∗2: Invalid in 16-bit mode (fixed at 0). The timer does not work if "1" (external clock) is set in 8-bit mode.

∗3: This timer has no clock output terminal, so a clock cannot be output even if this bit is set to "1" (output On).

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 41 counter data register	**00401A7 (B)**	D7	**TC417**	16-bit timer 41 counter data TC417 = MSB TC410 = LSB	0 to 255				X	R	Used as the high-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC416**						X		
		D5	**TC415**						X		
		D4	**TC414**						X		
		D3	**TC413**						X		
		D2	**TC412**						X		
		D1	**TC411**						X		
		D0	**TC410**						X		
16-bit timer 50 control register	**00401A8 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**MODE16**	16-bit timer 5 mode selection	1	16 bits	0	8 bits	0	R/W	
		D3	**CKSL50**	16-bit timer 50 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	**PTM50**	16-bit timer 50 clock output control	1	Invalid	0	Off	0	R/W	*3
		D1	**PSET50**	16-bit timer 50 preset	1	Preset	0	Invalid	0	W	0 when being read.
		D0	**PRUN50**	16-bit timer 50 Run/Stop control	1	Run	0	Stop	0	R/W	
16-bit timer 51 control register	**00401A9 (B)**	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**CKSL51**	16-bit timer 51 input clock selection	1	Invalid	0	Internal clock	0	R/W	*2
		D2	**PTM51**	16-bit timer 51 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET51**	16-bit timer 51 preset	1	Preset	0	Invalid	0	W	0 when being read.*1
		D0	**PRUN51**	16-bit timer 51 Run/Stop control	1	Run	0	Stop	0	R/W	*1
16-bit timer 50 reload data register	**00401AA (B)**	D7	**RR507**	16-bit timer 50 reload data RR507 = MSB RR500 = LSB	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR506**						X		
		D5	**RR505**						X		
		D4	**RR504**						X		
		D3	**RR503**						X		
		D2	**RR502**						X		
		D1	**RR501**						X		
		D0	**RR500**						X		
16-bit timer 51 reload data register	**00401AB (B)**	D7	**RR517**	16-bit timer 51 reload data RR517 = MSB RR510 = LSB	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR516**						X		
		D5	**RR515**						X		
		D4	**RR514**						X		
		D3	**RR513**						X		
		D2	**RR512**						X		
		D1	**RR511**						X		
		D0	**RR510**						X		
16-bit timer 50 comparison data register	**00401AC (B)**	D7	**CR507**	16-bit timer 50 comparison data CR507 = MSB CR500 = LSB	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR506**						X		
		D5	**CR505**						X		
		D4	**CR504**						X		
		D3	**CR503**						X		
		D2	**CR502**						X		
		D1	**CR501**						X		
		D0	**CR500**						X		
16-bit timer 51 comparison data register	**00401AD (B)**	D7	**CR517**	16-bit timer 51 comparison data CR517 = MSB CR510 = LSB	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR516**						X		
		D5	**CR515**						X		
		D4	**CR514**						X		
		D3	**CR513**						X		
		D2	**CR512**						X		
		D1	**CR511**						X		
		D0	**CR510**						X		
16-bit timer 50 counter data register	**00401AE (B)**	D7	**TC507**	16-bit timer 50 counter data TC507 = MSB TC500 = LSB	0 to 255				X	R	Used as the low-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC506**						X		
		D5	**TC505**						X		
		D4	**TC504**						X		
		D3	**TC503**						X		
		D2	**TC502**						X		
		D1	**TC501**						X		
		D0	**TC500**						X		
16-bit timer 51 counter data register	**00401AF (B)**	D7	**TC517**	16-bit timer 51 counter data TC517 = MSB TC510 = LSB	0 to 255				X	R	Used as the high-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC516**						X		
		D5	**TC515**						X		
		D4	**TC514**						X		
		D3	**TC513**						X		
		D2	**TC512**						X		
		D1	**TC511**						X		
		D0	**TC510**						X		

*1: Invalid in 16-bit mode (fixed at 0).

*2: Invalid in 16-bit mode (fixed at 0). The timer does not work if "1" (external clock) is set in 8-bit mode.

*3: This timer has no clock output terminal, so a clock cannot be output even if this bit is set to "1" (output On).

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Serial I/F Ch.0 transmit data register	**00401E0 (B)**	D7	**TXD07**	Serial I/F Ch.0 transmit data	0x0 to 0xFF(0x7F)				X	R/W	7-bit asynchronous mode does not use TXD07.
		D6	**TXD06**	TXD07(06) = MSB					X		
		D5	**TXD05**	TXD00 = LSB					X		
		D4	**TXD04**						X		
		D3	**TXD03**						X		
		D2	**TXD02**						X		
		D1	**TXD01**						X		
		D0	**TXD00**						X		
Serial I/F Ch.0 receive data register	**00401E1 (B)**	D7	**RXD07**	Serial I/F Ch.0 receive data	0x0 to 0xFF(0x7F)				X	R	7-bit asynchronous mode does not use RXD07 (fixed at 0).
		D6	**RXD06**	RXD07(06) = MSB					X		
		D5	**RXD05**	RXD00 = LSB					X		
		D4	**RXD04**						X		
		D3	**RXD03**						X		
		D2	**RXD02**						X		
		D1	**RXD01**						X		
		D0	**RXD00**						X		
Serial I/F Ch.0 status register	**00401E2 (B)**	D7–5	**–**	–	–				–	–	0 when being read.
		D4	**FER0**	Ch.0 flaming error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D3	**PER0**	Ch.0 parity error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D2	**OER0**	Ch.0 overrun error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D1	**TDBE0**	Ch.0 transmit data buffer empty	1	Empty	0	Buffer full	1	R	
		D0	**RDBF0**	Ch.0 receive data buffer full	1	Buffer full	0	Empty	0	R	
Serial I/F Ch.0 control register	**00401E3 (B)**	D7	**TXEN0**	Ch.0 transmit enable	1	Enabled	0	Disabled	0	R/W	
		D6	**RXEN0**	Ch.0 receive enable	1	Enabled	0	Disabled	0	R/W	
		D5	**EPR0**	Ch.0 parity enable	1	With parity	0	No parity	X	R/W	Valid only in asynchronous mode.
		D4	**PMD0**	Ch.0 parity mode selection	1	Odd	0	Even	X	R/W	
		D3	**STPB0**	Ch.0 stop bit selection	1	2 bits	0	1 bit	X	R/W	
		D2	**SSCK0**	Ch.0 input clock selection	1	#SCLK0	0	Internal clock	X	R/W	
		D1	**SMD01**	Ch.0 transfer mode selection	SMD0[1:0]		Transfer mode		X	R/W	
		D0	**SMD00**		1	1	8-bit asynchronous		X		
					1	0	7-bit asynchronous				
					0	1	Clock sync. Slave				
					0	0	Clock sync. Master				
Serial I/F Ch.0 IrDA register	**00401E4 (B)**	D7–5	**–**	–	–				–	–	0 when being read.
		D4	**DIVMD0**	Ch.0 async. clock division ratio	1	1/8	0	1/16	X	R/W	
		D3	**IRTL0**	Ch.0 IrDA I/F output logic inversion	1	Inverted	0	Direct	X	R/W	Valid only in asynchronous mode.
		D2	**IRRL0**	Ch.0 IrDA I/F input logic inversion	1	Inverted	0	Direct	X	R/W	
		D1	**IRMD01**	Ch.0 interface mode selection	IRMD0[1:0]		I/F mode		X	R/W	
		D0	**IRMD00**		1	1	reserved		X		
					1	0	IrDA 1.0				
					0	1	reserved				
					0	0	General I/F				
Serial I/F Ch.1 transmit data register	**00401E5 (B)**	D7	**TXD17**	Serial I/F Ch.1 transmit data	0x0 to 0xFF(0x7F)				X	R/W	7-bit asynchronous mode does not use TXD17.
		D6	**TXD16**	TXD17(16) = MSB					X		
		D5	**TXD15**	TXD10 = LSB					X		
		D4	**TXD14**						X		
		D3	**TXD13**						X		
		D2	**TXD12**						X		
		D1	**TXD11**						X		
		D0	**TXD10**						X		
Serial I/F Ch.1 receive data register	**00401E6 (B)**	D7	**RXD17**	Serial I/F Ch.1 receive data	0x0 to 0xFF(0x7F)				X	R	7-bit asynchronous mode does not use RXD17 (fixed at 0).
		D6	**RXD16**	RXD17(16) = MSB					X		
		D5	**RXD15**	RXD10 = LSB					X		
		D4	**RXD14**						X		
		D3	**RXD13**						X		
		D2	**RXD12**						X		
		D1	**RXD11**						X		
		D0	**RXD10**						X		
Serial I/F Ch.1 status register	**00401E7 (B)**	D7–5	**–**	–	–				–	–	0 when being read.
		D4	**FER1**	Ch.1 flaming error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D3	**PER1**	Ch.1 parity error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D2	**OER1**	Ch.1 overrun error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D1	**TDBE1**	Ch.1 transmit data buffer empty	1	Empty	0	Buffer full	1	R	
		D0	**RDBF1**	Ch.1 receive data buffer full	1	Buffer full	0	Empty	0	R	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Serial I/F Ch.1	**00401E8**	D7	**TXEN1**	Ch.1 transmit enable	1	Enabled	0	Disabled	0	R/W	
control register	**(B)**	D6	**RXEN1**	Ch.1 receive enable	1	Enabled	0	Disabled	0	R/W	
		D5	**EPR1**	Ch.1 parity enable	1	With parity	0	No parity	X	R/W	Valid only in
		D4	**PMD1**	Ch.1 parity mode selection	1	Odd	0	Even	X	R/W	asynchronous mode.
		D3	**STPB1**	Ch.1 stop bit selection	1	2 bits	0	1 bit	X	R/W	
		D2	**SSCK1**	Ch.1 input clock selection	1	#SCLK1	0	Internal clock	X	R/W	
		D1	**SMD11**	Ch.1 transfer mode selection	SMD1[1:0]		Transfer mode		X	R/W	
		D0	**SMD10**		1	1	8-bit asynchronous		X		
					1	0	7-bit asynchronous				
					0	1	Clock sync. Slave				
					0	0	Clock sync. Master				
Serial I/F Ch.1	**00401E9**	D7–5	**–**	–	–				–	–	0 when being read.
IrDA register	**(B)**	D4	**DIVMD1**	Ch.1 async. clock division ratio	1	1/8	0	1/16	X	R/W	
		D3	**IRTL1**	Ch.1 IrDA I/F output logic inversion	1	Inverted	0	Direct	X	R/W	Valid only in
		D2	**IRRL1**	Ch.1 IrDA I/F input logic inversion	1	Inverted	0	Direct	X	R/W	asynchronous mode.
		D1	**IRMD11**	Ch.1 interface mode selection	IRMD1[1:0]		I/F mode		X	R/W	
		D0	**IRMD10**		1	1	reserved		X		
					1	0	IrDA 1.0				
					0	1	reserved				
					0	0	General I/F				
A/D conversion	**0040240**	D7	**ADD7**	A/D converted data	0x0 to 0x3FF				0	R	
result (low-	**(B)**	D6	**ADD6**	(low-order 8 bits)					0		
order) register		D5	**ADD5**	ADD0 = LSB					0		
		D4	**ADD4**						0		
		D3	**ADD3**						0		
		D2	**ADD2**						0		
		D1	**ADD1**						0		
		D0	**ADD0**						0		
A/D conversion	**0040241**	D7–2	**–**	–					–	–	0 when being read.
result (high-	**(B)**	D1	**ADD9**	A/D converted data					0	R	
order) register		D0	**ADD8**	(high-order 2 bits) ADD9 = MSB					0		
A/D trigger	**0040242**	D7–6	**–**	–	–				–	–	0 when being read.
register	**(B)**	D5	**MS**	A/D conversion mode selection	1	Continuous	0	Normal	0	R/W	
		D4	**TS1**	A/D conversion trigger selection	TS[1:0]		Trigger		0	R/W	
		D3	**TS0**		1	1	#ADTRG pin		0		
					1	0	8-bit timer 0				
					0	1	16-bit timer 0				
					0	0	Software				
		D2	**CH2**	A/D conversion channel status	CH[2:0]			Channel	0	R	
		D1	**CH1**		1	1	1	AD7	0		
		D0	**CH0**		1	1	0	AD6	0		
					1	0	1	AD5			
					1	0	0	AD4			
					0	1	1	AD3			
					0	1	0	AD2			
					0	0	1	AD1			
					0	0	0	AD0			
A/D channel	**0040243**	D7–6	**–**	–	–				–	–	0 when being read.
register	**(B)**	D5	**CE2**	A/D converter	CE[2:0]			End channel	0	R/W	
		D4	**CE1**	end channel selection	1	1	1	AD7	0		
		D3	**CE0**		1	1	0	AD6	0		
					1	0	1	AD5			
					1	0	0	AD4			
					0	1	1	AD3			
					0	1	0	AD2			
					0	0	1	AD1			
					0	0	0	AD0			
		D2	**CS2**	A/D converter	CS[2:0]			Start channel	0	R/W	
		D1	**CS1**	start channel selection	1	1	1	AD7	0		
		D0	**CS0**		1	1	0	AD6	0		
					1	0	1	AD5			
					1	0	0	AD4			
					0	1	1	AD3			
					0	1	0	AD2			
					0	0	1	AD1			
					0	0	0	AD0			
A/D enable	**0040244**	D7–4	**–**	–	–				–	–	0 when being read.
register	**(B)**	D3	**ADF**	Conversion-complete flag	1	Completed	0	Run/Standby	0	R	Reset when ADD is read.
		D2	**ADE**	A/D enable	1	Enabled	0	Disabled	0	R/W	
		D1	**ADST**	A/D conversion control/status	1	Start/Run	0	Stop	0	R/W	
		D0	**OWE**	Overwrite error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D sampling register	**0040245 (B)**	D7–2	**–**	–	–	–	–	0 when being read.
		D1	**ST1**	Input signal sampling time setup	ST[1:0] Sampring time 1 1 9 clocks 1 0 7 clocks 0 1 5 clocks 0 0 3 clocks	1	R/W	Use with 9 clocks.
		D0	**ST0**			1		
D/A output control register	**0040246 (B)**	D7–2	**–**	–	–	–	–	0 when being read.
		D1	**DAE0**	DA0 output control	1 Output 0 High-Z	0	R/W	
		D0	**DAE1**	DA1 output control	1 Output 0 High-Z	0	R/W	
D/A operating voltage switch register	**0040247 (B)**	D7–2	**–**	–	–	–	–	0 when being read.
		D1	**–**	–	–	X	–	Undefined in read.
		D0	**SELV**	D/A operating voltage switch	1 5 V 0 3.3 V	0	R/W	
DA1 output data register	**0040248 (B)**	D7	**DADR17**	DA1 output data DADR17 = MSB DADR10 = LSB	0 to 255	0	R/W	
		D6	**DADR16**			0		
		D5	**DADR15**			0		
		D4	**DADR14**			0		
		D3	**DADR13**			0		
		D2	**DADR12**			0		
		D1	**DADR11**			0		
		D0	**DADR10**			0		
DA0 output data register	**0040249 (B)**	D7	**DADR07**	DA0 output data DADR07 = MSB DADR00 = LSB	0 to 255	0	R/W	
		D6	**DADR06**			0		
		D5	**DADR05**			0		
		D4	**DADR04**			0		
		D3	**DADR03**			0		
		D2	**DADR02**			0		
		D1	**DADR01**			0		
		D0	**DADR00**			0		
K6/K5 interrupt priority register	**0040260 (B)**	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**PK6L2**	K6[3:0], K5[4:0] interrupt level	0 to 7	X	R/W	
		D5	**PK6L1**			X		
		D4	**PK6L0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PK6H2**	K6[7:4] interrupt level	0 to 7	X	R/W	
		D1	**PK6H1**			X		
		D0	**PK6H0**			X		
16-bit timer 0–1, DMA interrupt priority register	**0040261 (B)**	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**P16T02**	16-bit timer 0–1 interrupt level	0 to 7	X	R/W	
		D5	**P16T01**			X		
		D4	**P16T00**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PDM2**	IDMA, high-speed DMA interrupt level	0 to 7	X	R/W	
		D1	**PDM1**			X		
		D0	**PDM0**			X		
16-bit timer 2–5 interrupt priority register	**0040262 (B)**	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**P16T42**	16-bit timer 4–5 interrupt level	0 to 7	X	R/W	
		D5	**P16T41**			X		
		D4	**P16T40**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**P16T22**	16-bit timer 2–3 interrupt level	0 to 7	X	R/W	
		D1	**P16T21**			X		
		D0	**P16T20**			X		
8-bit timer, serial I/F Ch.0 interrupt priority register	**0040263 (B)**	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**PSI002**	Serial interface Ch.0 interrupt level	0 to 7	X	R/W	
		D5	**PSI001**			X		
		D4	**PSI000**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**P8TM2**	8-bit timer 0–3 interrupt level	0 to 7	X	R/W	
		D1	**P8TM1**			X		
		D0	**P8TM0**			X		
Serial I/F Ch.1, A/D interrupt priority register	**0040264 (B)**	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**PAD2**	A/D converter interrupt level	0 to 7	X	R/W	
		D5	**PAD1**			X		
		D4	**PAD0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PSI012**	Serial interface Ch.1 interrupt level	0 to 7	X	R/W	
		D1	**PSI011**			X		
		D0	**PSI010**			X		
Clock timer interrupt priority register	**0040265 (B)**	D7–3	**–**	reserved	–	–	–	Writing 1 not allowed.
		D2	**PCTM2**	Clock timer interrupt level	0 to 7	X	R/W	
		D1	**PCTM1**			X		
		D0	**PCTM0**			X		

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
K6/K5 interrupt enable register	**0040270 (B)**	D7–6	**–**	reserved	–				–	–	0 when being read.
		D5	**EK5**	K5[4:0] input	1	Enabled	0	Disabled	0	R/W	
		D4	**EK6**	K6[3:0] input					0	R/W	
		D3	**EK64**	K64 input					0	R/W	
		D2	**EK65**	K65 input					0	R/W	
		D1	**EK66**	K66 input					0	R/W	
		D0	**EK67**	K67 input					0	R/W	
DMA interrupt enable register	**0040271 (B)**	D7–3	–	reserved	–				–	–	0 when being read.
		D2	**EIDMA**	IDMA	1	Enabled	0	Disabled	0	R/W	
		D1	**EHDM1**	High-speed DMA Ch.1					0	R/W	
		D0	**EHDM0**	High-speed DMA Ch.0					0	R/W	
16-bit timer 0–1 interrupt enable register	**0040272 (B)**	D7	**E16TC11**	16-bit timer 11 comparison match	1	Enabled	0	Disabled	0	R/W	
		D6	**E16TU11**	16-bit timer 11 underflow					0	R/W	
		D5	**E16TC10**	16-bit timer 10 comparison match					0	R/W	
		D4	**E16TU10**	16-bit timer 10 underflow					0	R/W	
		D3	**E16TC01**	16-bit timer 01 comparison match					0	R/W	
		D2	**E16TU01**	16-bit timer 01 underflow					0	R/W	
		D1	**E16TC00**	16-bit timer 00 comparison match					0	R/W	
		D0	**E16TU00**	16-bit timer 00 underflow					0	R/W	
16-bit timer 2–3 interrupt enable register	**0040273 (B)**	D7	**E16TC31**	16-bit timer 31 comparison match	1	Enabled	0	Disabled	0	R/W	
		D6	**E16TU31**	16-bit timer 31 underflow					0	R/W	
		D5	**E16TC30**	16-bit timer 30 comparison match					0	R/W	
		D4	**E16TU30**	16-bit timer 30 underflow					0	R/W	
		D3	**E16TC21**	16-bit timer 21 comparison match					0	R/W	
		D2	**E16TU21**	16-bit timer 21 underflow					0	R/W	
		D1	**E16TC20**	16-bit timer 20 comparison match					0	R/W	
		D0	**E16TU20**	16-bit timer 20 underflow					0	R/W	
16-bit timer 4–5 interrupt enable register	**0040274 (B)**	D7	**E16TC51**	16-bit timer 51 comparison match	1	Enabled	0	Disabled	0	R/W	
		D6	**E16TU51**	16-bit timer 51 underflow					0	R/W	
		D5	**E16TC50**	16-bit timer 50 comparison match					0	R/W	
		D4	**E16TU50**	16-bit timer 50 underflow					0	R/W	
		D3	**E16TC41**	16-bit timer 41 comparison match					0	R/W	
		D2	**E16TU41**	16-bit timer 41 underflow					0	R/W	
		D1	**E16TC40**	16-bit timer 40 comparison match					0	R/W	
		D0	**E16TU40**	16-bit timer 40 underflow					0	R/W	
8-bit timer interrupt enable register	**0040275 (B)**	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**E8TU3**	8-bit timer 3 underflow	1	Enabled	0	Disabled	0	R/W	
		D2	**E8TU2**	8-bit timer 2 underflow					0	R/W	
		D1	**E8TU1**	8-bit timer 1 underflow					0	R/W	
		D0	**E8TU0**	8-bit timer 0 underflow					0	R/W	
Serial I/F interrupt enable register	**0040276 (B)**	D7–6	**–**	reserved	–				–	–	0 when being read.
		D5	**ESTX1**	SIF Ch.1 transmit buffer empty	1	Enabled	0	Disabled	0	R/W	
		D4	**ESRX1**	SIF Ch.1 receive buffer full					0	R/W	
		D3	**ESERR1**	SIF Ch.1 receive error					0	R/W	
		D2	**ESTX0**	SIF Ch.0 transmit buffer empty					0	R/W	
		D1	**ESRX0**	SIF Ch.0 receive buffer full					0	R/W	
		D0	**ESERR0**	SIF Ch.0 receive error					0	R/W	
Clock timer, A/D interrupt enable register	**0040277 (B)**	D7–2	**–**	reserved	–				–	–	Writing 0 not allowed.
		D1	**ECTM**	Clock timer	1	Enabled	0	Disabled	0	R/W	
		D0	**EADE**	A/D converter					0	R/W	
K6/K5 interrupt factor flag register	**0040280 (B)**	D7–6	**–**	reserved	–				–	–	0 when being read.
		D5	**FK5**	K5[4:0] input	1	Factor is generated	0	No factor is generated	X	R/W	
		D4	**FK6**	K6[3:0] input					X	R/W	
		D3	**FK64**	K64 input					X	R/W	
		D2	**FK65**	K65 input					X	R/W	
		D1	**FK66**	K66 input					X	R/W	
		D0	**FK67**	K67 input					X	R/W	
DMA interrupt factor flag register	**0040281 (B)**	D7–3	**–**	reserved	–				–	–	0 when being read.
		D2	**FIDMA**	IDMA	1	Factor is generated	0	No factor is generated	X	R/W	
		D1	**FHDM1**	High-speed DMA Ch.1					X	R/W	
		D0	**FHDM0**	High-speed DMA Ch.0					X	R/W	
16-bit timer 0–1 interrupt factor flag register	**0040282 (B)**	D7	**F16TC11**	16-bit timer 11 comparison match	1	Factor is generated	0	No factor is generated	X	R/W	
		D6	**F16TU11**	16-bit timer 11 underflow					X	R/W	
		D5	**F16TC10**	16-bit timer 10 comparison match					X	R/W	
		D4	**F16TU10**	16-bit timer 10 underflow					X	R/W	
		D3	**F16TC01**	16-bit timer 01 comparison match					X	R/W	
		D2	**F16TU01**	16-bit timer 01 underflow					X	R/W	
		D1	**F16TC00**	16-bit timer 00 comparison match					X	R/W	
		D0	**F16TU00**	16-bit timer 00 underflow					X	R/W	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 2–3 interrupt factor flag register	**0040283 (B)**	D7	**F16TC31**	16-bit timer 31 comparison match	1	Factor is generated	0	No factor is generated	X	R/W	
		D6	**F16TU31**	16-bit timer 31 underflow					X	R/W	
		D5	**F16TC30**	16-bit timer 30 comparison match					X	R/W	
		D4	**F16TU30**	16-bit timer 30 underflow					X	R/W	
		D3	**F16TC21**	16-bit timer 21 comparison match					X	R/W	
		D2	**F16TU21**	16-bit timer 21 underflow					X	R/W	
		D1	**F16TC20**	16-bit timer 20 comparison match					X	R/W	
		D0	**F16TU20**	16-bit timer 20 underflow					X	R/W	
16-bit timer 4–5 interrupt factor flag register	**0040284 (B)**	D7	**F16TC51**	16-bit timer 51 comparison match	1	Factor is generated	0	No factor is generated	X	R/W	
		D6	**F16TU51**	16-bit timer 51 underflow					X	R/W	
		D5	**F16TC50**	16-bit timer 50 comparison match					X	R/W	
		D4	**F16TU50**	16-bit timer 50 underflow					X	R/W	
		D3	**F16TC41**	16-bit timer 41 comparison match					X	R/W	
		D2	**F16TU41**	16-bit timer 41 underflow					X	R/W	
		D1	**F16TC40**	16-bit timer 40 comparison match					X	R/W	
		D0	**F16TU40**	16-bit timer 40 underflow					X	R/W	
8-bit timer interrupt factor flag register	**0040285 (B)**	D7–4	–	reserved	–				–	–	0 when being read.
		D3	**F8TU3**	8-bit timer 3 underflow	1	Factor is generated	0	No factor is generated	X	R/W	
		D2	**F8TU2**	8-bit timer 2 underflow					X	R/W	
		D1	**F8TU1**	8-bit timer 1 underflow					X	R/W	
		D0	**F8TU0**	8-bit timer 0 underflow					X	R/W	
Serial I/F interrupt factor flag register	**0040286 (B)**	D7–6	–	reserved	–				–	–	0 when being read.
		D5	**FSTX1**	SIF Ch.1 transmit buffer empty	1	Factor is generated	0	No factor is generated	X	R/W	
		D4	**FSRX1**	SIF Ch.1 receive buffer full					X	R/W	
		D3	**FSERR1**	SIF Ch.1 receive error					X	R/W	
		D2	**FSTX0**	SIF Ch.0 transmit buffer empty					X	R/W	
		D1	**FSRX0**	SIF Ch.0 receive buffer full					X	R/W	
		D0	**FSERR0**	SIF Ch.0 receive error					X	R/W	
Clock timer, A/D interrupt factor flag register	**0040287 (B)**	D7–2	–	reserved	–				–	–	0 when being read.
		D1	**FCTM**	Clock timer	1	Factor is generated	0	No factor is generated	X	R/W	
		D0	**FADE**	A/D converter					X	R/W	
K6, DMA, 16-bit timer 0 IDMA request register	**0040290 (B)**	D7	**R16TC01**	16-bit timer 01 comparison match	1	IDMA request	0	Interrupt request	X	R/W	
		D6	**R16TU01**	16-bit timer 01 underflow					X	R/W	
		D5	**RHDM1**	High-speed DMA Ch.1					X	R/W	
		D4	**RHDM0**	High-speed DMA Ch.0					X	R/W	
		D3	**RK64**	K64 input					X	R/W	
		D2	**RK65**	K65 input					X	R/W	
		D1	**RK66**	K66 input					X	R/W	
		D0	**RK67**	K67 input					X	R/W	
16-bit timer 1–4 IDMA request register	**0040291 (B)**	D7	**R16TC41**	16-bit timer 41 comparison match	1	IDMA request	0	Interrupt request	X	R/W	
		D6	**R16TU41**	16-bit timer 41 underflow					X	R/W	
		D5	**R16TC31**	16-bit timer 31 comparison match					X	R/W	
		D4	**R16TU31**	16-bit timer 31 underflow					X	R/W	
		D3	**R16TC21**	16-bit timer 21 comparison match					X	R/W	
		D2	**R16TU21**	16-bit timer 21 underflow					X	R/W	
		D1	**R16TC11**	16-bit timer 11 comparison match					X	R/W	
		D0	**R16TU11**	16-bit timer 11 underflow					X	R/W	
16-bit timer 5, 8-bit timer, serial I/F Ch.0 IDMA request register	**0040292 (B)**	D7	**RSTX0**	SIF Ch.0 transmit buffer empty	1	IDMA request	0	Interrupt request	X	R/W	
		D6	**RSRX0**	SIF Ch.0 receive buffer full					X	R/W	
		D5	**R8TU3**	8-bit timer 3 underflow					X	R/W	
		D4	**R8TU2**	8-bit timer 2 underflow					X	R/W	
		D3	**R8TU1**	8-bit timer 1 underflow					X	R/W	
		D2	**R8TU0**	8-bit timer 0 underflow					X	R/W	
		D1	**R16TC51**	16-bit timer 51 comparison match					X	R/W	
		D0	**R16TU51**	16-bit timer 51 underflow					X	R/W	
Serial I/F Ch.1, A/D IDMA request register	**0040293 (B)**	D7–3	–	reserved	–				–	–	
		D2	**RADE**	A/D converter	1	IDMA request	0	Interrupt request	X	R/W	
		D1	**RSTX1**	SIF Ch.1 transmit buffer empty					X	R/W	
		D0	**RSRX1**	SIF Ch.1 receive buffer full					X	R/W	
Interrupt factor flag reset method select register	**004029F (B)**	D7–1	–	reserved	–				–	–	
		D0	**RSTONLY**	Interrupt factor flag reset method selection	1	Reset only	0	RD/WR	1	R/W	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
K5 function select register	**00402C0 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**CFK54**	K54 function selection	1	DA1	0	K54	0	W	Undefined in read.
		D3	**CFK53**	K53 function selection	1	DA0	0	K53	0	W	
		D2	**CFK52**	K52 function selection	1	#ADTRG	0	K52	0	W	
		D1	**CFK51**	K51 function selection	1	#DMAREQ1	0	K51	0	W	
		D0	**CFK50**	K50 function selection	1	#DMAREQ0	0	K50	0	W	
K5 input port data register	**00402C1 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**K54D**	K54 input port data	1	High	0	Low	–	R	
		D3	**K53D**	K53 input port data					–	R	
		D2	**K52D**	K52 input port data					–	R	
		D1	**K51D**	K51 input port data					–	R	
		D0	**K50D**	K50 input port data					–	R	
K5 interrupt select register	**00402C2 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**SIK54**	K54 interrupt selection	1	Interrupt enabled	0	Interrupt disabled	0	R/W	
		D3	**SIK53**	K53 interrupt selection					0	R/W	
		D2	**SIK52**	K52 interrupt selection					0	R/W	
		D1	**SIK51**	K51 interrupt selection					0	R/W	
		D0	**SIK50**	K50 interrupt selection					0	R/W	
K5 input comparison register	**00402C3 (B)**	D7–5	–	reserved	–				–	–	0 when being read.
		D4	**KCP54**	K54 input comparison	1	Interrupt is generated at falling edge.	0	Interrupt is generated at rising edge.	0	R/W	
		D3	**KCP53**	K53 input comparison					0	R/W	
		D2	**KCP52**	K52 input comparison					0	R/W	
		D1	**KCP51**	K51 input comparison					0	R/W	
		D0	**KCP50**	K50 input comparison					0	R/W	
K5 pull-up control register	**00402C4 (B)**	D7–5	–	reserved	–				–	–	0 when being read.
		D4	**KPU54**	K54 pull-up control	1	Pulled up	0	No pull-up	0	W	Undefined in read.
		D3	**KPU53**	K53 pull-up control					0	W	
		D2	**KPU52**	K52 pull-up control					0	W	
		D1	**KPU51**	K51 pull-up control					0	W	
		D0	**KPU50**	K50 pull-up control					0	W	
K6 function select register	**00402C5 (B)**	D7	**CFK67**	K67 function selection	1	AD7	0	K67	0	W	Undefined in read.
		D6	**CFK66**	K66 function selection	1	AD6	0	K66	0	W	
		D5	**CFK65**	K65 function selection	1	AD5	0	K65	0	W	
		D4	**CFK64**	K64 function selection	1	AD4	0	K64	0	W	
		D3	**CFK63**	K63 function selection	1	AD3	0	K63	0	W	
		D2	**CFK62**	K62 function selection	1	AD2	0	K62	0	W	
		D1	**CFK61**	K61 function selection	1	AD1	0	K61	0	W	
		D0	**CFK60**	K60 function selection	1	AD0	0	K60	0	W	
K6 input port data register	**00402C6 (B)**	D7	**K67D**	K67 input port data	1	High	0	Low	–	R	
		D6	**K66D**	K66 input port data					–	R	
		D5	**K65D**	K65 input port data					–	R	
		D4	**K64D**	K64 input port data					–	R	
		D3	**K63D**	K63 input port data					–	R	
		D2	**K62D**	K62 input port data					–	R	
		D1	**K61D**	K61 input port data					–	R	
		D0	**K60D**	K60 input port data					–	R	
K6 interrupt select register	**00402C7 (B)**	D7–4	–	reserved	–				–	–	0 when being read.
		D3	**SIK63**	K63 interrupt selection	1	Interrupt enabled	0	Interrupt disabled	0	R/W	
		D2	**SIK62**	K62 interrupt selection					0	R/W	
		D1	**SIK61**	K61 interrupt selection					0	R/W	
		D0	**SIK60**	K60 interrupt selection					0	R/W	
K6 input comparison register	**00402C8 (B)**	D7	**KCP67**	K67 input comparison	1	Interrupt is generated at falling edge.	0	Interrupt is generated at rising edge.	0	R/W	
		D6	**KCP66**	K66 input comparison					0	R/W	
		D5	**KCP65**	K65 input comparison					0	R/W	
		D4	**KCP64**	K64 input comparison					0	R/W	
		D3	**KCP63**	K63 input comparison					0	R/W	
		D2	**KCP62**	K62 input comparison					0	R/W	
		D1	**KCP61**	K61 input comparison					0	R/W	
		D0	**KCP60**	K60 input comparison					0	R/W	
K6 pull-up control register	**00402C9 (B)**	D7	**KPU67**	K67 pull-up control	1	Pulled up	0	No pull-up	0	W	Undefined in read.
		D6	**KPU66**	K66 pull-up control					0	W	
		D5	**KPU65**	K65 pull-up control					0	W	
		D4	**KPU64**	K64 pull-up control					0	W	
		D3	**KPU63**	K63 pull-up control					0	W	
		D2	**KPU62**	K62 pull-up control					0	W	
		D1	**KPU61**	K61 pull-up control					0	W	
		D0	**KPU60**	K60 pull-up control					0	W	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
R7 function select register	**00402CA (B)**	D7	**CFR77**	R77 function selection	1	TM51	0	R77	0	W	Undefined in read.
		D6	**CFR76**	R76 function selection	1	TM41	0	R76	0	W	
		D5	**CFR75**	R75 function selection	1	TM31	0	R75	0	W	
		D4	**CFR74**	R74 function selection	1	TM21	0	R74	0	W	
		D3	**CFR73**	R73 function selection	1	TM11	0	R73	0	W	
		D2	**CFR72**	R72 function selection	1	TM10	0	R72	0	W	
		D1	**CFR71**	R71 function selection	1	TM01	0	R71	0	W	
		D0	**CFR70**	R70 function selection	1	TM00	0	R70	0	W	
R7 output port data register	**00402CB (B)**	D7	**R77D**	R77 output port data	1	High	0	Low	0	R/W	
		D6	**R76D**	R76 output port data					0	R/W	
		D5	**R75D**	R75 output port data					0	R/W	
		D4	**R74D**	R74 output port data					0	R/W	
		D3	**R73D**	R73 output port data					0	R/W	
		D2	**R72D**	R72 output port data					0	R/W	
		D1	**R71D**	R71 output port data					0	R/W	
		D0	**R70D**	R70 output port data					0	R/W	
R8 function select register	**00402CC (B)**	D7–4	–	reserved	–				–	–	0 when being read.
		D3	**CFR83**	R83 function selection	1	#DMAACK1	0	R83	0	W	Undefined in read.
		D2	**CFR82**	R82 function selection	1	#DMAACK0	0	R82	0	W	
		D1	**CFR81**	R81 function selection	1	FOSC1	0	R81	0	W	
		D0	–	–	–				–	–	0 when being read.
R8 output port data register	**00402CD (B)**	D7–4	–	reserved	–				–	–	0 when being read.
		D3	**R83D**	R83 output port data	1	High	0	Low	1	R/W	
		D2	**R82D**	R82 output port data					1	R/W	
		D1	**R81D**	R81 output port data					1	R/W	
		D0	–	–	–				–	–	0 when being read.
P0 function select register	**00402D0 (B)**	D7	**CFP07**	P07 function selection	1	#SRDY1	0	P07	0	W	Undefined in read.
		D6	**CFP06**	P06 function selection	1	#SCLK1	0	P06	0	W	
		D5	**CFP05**	P05 function selection	1	SOUT1	0	P05	0	W	
		D4	**CFP04**	P04 function selection	1	SIN1	0	P04	0	W	
		D3	**CFP03**	P03 function selection	1	#SRDY0	0	P03	0	W	
		D2	**CFP02**	P02 function selection	1	#SCLK0	0	P02	0	W	
		D1	**CFP01**	P01 function selection	1	SOUT0	0	P01	0	W	
		D0	**CFP00**	P00 function selection	1	SIN0	0	P00	0	W	
P0 I/O port data register	**00402D1 (B)**	D7	**P07D**	P07 I/O port data	1	High	0	Low	0	R/W	
		D6	**P06D**	P06 I/O port data					0	R/W	
		D5	**P05D**	P05 I/O port data					0	R/W	
		D4	**P04D**	P04 I/O port data					0	R/W	
		D3	**P03D**	P03 I/O port data					0	R/W	
		D2	**P02D**	P02 I/O port data					0	R/W	
		D1	**P01D**	P01 I/O port data					0	R/W	
		D0	**P00D**	P00 I/O port data					0	R/W	
P0 I/O control register	**00402D2 (B)**	D7	**IOC07**	P07 I/O control	1	Output	0	Input	0	W	Undefined in read.
		D6	**IOC06**	P06 I/O control					0	W	
		D5	**IOC05**	P05 I/O control					0	W	
		D4	**IOC04**	P04 I/O control					0	W	
		D3	**IOC03**	P03 I/O control					0	W	
		D2	**IOC02**	P02 I/O control					0	W	
		D1	**IOC01**	P01 I/O control					0	W	
		D0	**IOC00**	P00 I/O control					0	W	
P0 pull-up control register	**00402D3 (B)**	D7	**IOU07**	P07 pull-up control	1	Pulled up	0	No pull-up	0	W	Undefined in read.
		D6	**IOU06**	P06 pull-up control					0	W	
		D5	**IOU05**	P05 pull-up control					0	W	
		D4	**IOU04**	P04 pull-up control					0	W	
		D3	**IOU03**	P03 pull-up control					0	W	
		D2	**IOU02**	P02 pull-up control					0	W	
		D1	**IOU01**	P01 pull-up control					0	W	
		D0	**IOU00**	P00 pull-up control					0	W	
P1 function select register	**00402D4 (B)**	D7	–	reserved	–				–	–	
		D6	**CFP16**	P16 function selection	1	EXCL50 #DMAEND1	0	P16	0	W	Undefined in read.
		D5	**CFP15**	P15 function selection	1	EXCL40 #DMAEND0	0	P15	0	W	
		D4	**CFP14**	P14 function selection	1	EXCL30 #BUSGET	0	P14	0	W	
		D3	**CFP13**	P13 function selection	1	EXCL20 T8UF3	0	P13	0	W	
		D2	**CFP12**	P12 function selection	1	EXCL10 T8UF2	0	P12	0	W	
		D1	**CFP11**	P11 function selection	1	EXCL01 T8UF1	0	P11	0	W	
		D0	**CFP10**	P10 function selection	1	EXCL00 T8UF0	0	P10	0	W	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P1 I/O port data register	**00402D5 (B)**	D7	**–**	reserved	–				–	–	Undefined in read.
		D6	**P16D**	P16 I/O port data	1	High	0	Low	0	R/W	
		D5	**P15D**	P15 I/O port data					0	R/W	
		D4	**P14D**	P14 I/O port data					0	R/W	
		D3	**P13D**	P13 I/O port data					0	R/W	
		D2	**P12D**	P12 I/O port data					0	R/W	
		D1	**P11D**	P11 I/O port data					0	R/W	
		D0	**P10D**	P10 I/O port data					0	R/W	
P1 I/O control register	**00402D6 (B)**	D7	**–**	reserved	–				–	–	
		D6	**IOC16**	P16 I/O control	1	Output	0	Input	0	W	Undefined in read.
		D5	**IOC15**	P15 I/O control					0	W	
		D4	**IOC14**	P14 I/O control					0	W	
		D3	**IOC13**	P13 I/O control					0	W	
		D2	**IOC12**	P12 I/O control					0	W	
		D1	**IOC11**	P11 I/O control					0	W	
		D0	**IOC10**	P10 I/O control					0	W	
P1 pull-up control register	**00402D7 (B)**	D7	**–**	reserved	–				–	–	
		D6	**IOU16**	P16 pull-up control	1	Pulled up	0	No pull-up	0	W	Undefined in read.
		D5	**IOU15**	P15 pull-up control					0	W	
		D4	**IOU14**	P14 pull-up control					0	W	
		D3	**IOU13**	P13 pull-up control					0	W	
		D2	**IOU12**	P12 pull-up control					0	W	
		D1	**IOU11**	P11 pull-up control					0	W	
		D0	**IOU10**	P10 pull-up control					0	W	
Port function extension register	**00402D8 (B)**	D7–3	**–**	reserved	–				–	–	Undefined in read.
		D2	**CFEX2**	R81 port extended function	1	#DRD	0	R81/FOSC1	0	W	Undefined in read.
		D1	**CFEX1**	P14 port extended function	1	#DWE	0	P14/#BUSGET	0	W	
		D0	**CFEX0**	P1[3:0] port extended function	1	DST/DPCO	0	P1x/T8UFx	0	W	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Areas 18–15	**0048120**	DF	–	reserved	–				–	–	0 when being read.
set-up register	**(HW)**	DE	**A18SZ**	Areas 18–17 device size selection	1	8 bits	0	16 bits	0	R/W	
		DD	**A18DF1**	Areas 18–17	A18DF[1:0]		Number of cycles		1	R/W	
		DC	**A18DF0**	output disable delay time	1	1	3.5		1		
					1	0	2.5				
					0	1	1.5				
					0	0	0.5				
		DB	–	reserved	–				–	–	0 when being read.
		DA	**A18WT2**	Areas 18–17 wait control	A18WT[2:0]			Wait cycles	1	R/W	
		D9	**A18WT1**		1	1	1	7	1		
		D8	**A18WT0**		1	1	0	6	1		
					1	0	1	5			
					1	0	0	4			
					0	1	1	3			
					0	1	0	2			
					0	0	1	1			
					0	0	0	0			
		D7	–	reserved	–				–	–	0 when being read.
		D6	**A16SZ**	Areas 16–15 device size selection	1	8 bits	0	16 bits	0	R/W	
		D5	**A16DF1**	Areas 16–15	A16DF[1:0]		Number of cycles		1	R/W	
		D4	**A16DF0**	output disable delay time	1	1	3.5		1		
					1	0	2.5				
					0	1	1.5				
					0	0	0.5				
		D3	–	reserved	–				–	–	0 when being read.
		D2	**A16WT2**	Areas 16–15 wait control	A16WT[2:0]			Wait cycles	1	R/W	
		D1	**A16WT1**		1	1	1	7	1		
		D0	**A16WT0**		1	1	0	6	1		
					1	0	1	5			
					1	0	0	4			
					0	1	1	3			
					0	1	0	2			
					0	0	1	1			
					0	0	0	0			
Areas 14–13	**0048122**	DF–9	–	reserved	–				–	–	0 when being read.
set-up register	**(HW)**	D8	**A14DRA**	Area 14 DRAM selection	1	Used	0	Not used	0	R/W	
		D7	**A13DRA**	Area 13 DRAM selection	1	Used	0	Not used	0	R/W	
		D6	**A14SZ**	Areas 14–13 device size selection	1	8 bits	0	16 bits	0	R/W	
		D5	**A14DF1**	Areas 14–13	A14DF[1:0]		Number of cycles		1	R/W	
		D4	**A14DF0**	output disable delay time	1	1	3.5		1		
					1	0	2.5				
					0	1	1.5				
					0	0	0.5				
		D3	–	reserved	–				–	–	0 when being read.
		D2	**A14WT2**	Areas 14–13 wait control	A14WT[2:0]			Wait cycles	1	R/W	
		D1	**A14WT1**		1	1	1	7	1		
		D0	**A14WT0**		1	1	0	6	1		
					1	0	1	5			
					1	0	0	4			
					0	1	1	3			
					0	1	0	2			
					0	0	1	1			
					0	0	0	0			
Areas 12–11	**0048124**	DF–7	–	reserved	–				–	–	0 when being read.
set-up register	**(HW)**	D6	**A12SZ**	Areas 12–11 device size selection	1	8 bits	0	16 bits	0	R/W	
		D5	**A12DF1**	Areas 12–11	A18DF[1:0]		Number of cycles		1	R/W	
		D4	**A12DF0**	output disable delay time	1	1	3.5		1		
					1	0	2.5				
					0	1	1.5				
					0	0	0.5				
		D3	–	reserved	–				–	–	0 when being read.
		D2	**A12WT2**	Areas 12–11 wait control	A18WT[2:0]			Wait cycles	1	R/W	
		D1	**A12WT1**		1	1	1	7	1		
		D0	**A12WT0**		1	1	0	6	1		
					1	0	1	5			
					1	0	0	4			
					0	1	1	3			
					0	1	0	2			
					0	0	1	1			
					0	0	0	0			

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Areas 10–9	**0048126**	DF–B	**–**	reserved	–				–	–	0 when being read.
set-up register	**(HW)**	DA	**A10BW1**	Areas 10–9	A10BW[1:0]			Wait cycles	0	R/W	
		D9	**A10BW0**	burst ROM	1	1		3	0		
				burst read cycle wait control	1	0		2			
					0	1		1			
					0	0		0			
		D8	**A10DRA**	Area 10 burst ROM selection	1	Used	0	Not used	0	R/W	
		D7	**A9DRA**	Area 9 burst ROM selection	1	Used	0	Not used	0	R/W	
		D6	**A10SZ**	Areas 10–9 device size selection	1	8 bits	0	16 bits	0	R/W	
		D5	**A10DF1**	Areas 10–9	A10DF[1:0]			Number of cycles	1	R/W	
		D4	**A10DF0**	output disable delay time	1	1		3.5	1		
					1	0		2.5			
					0	1		1.5			
					0	0		0.5			
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**A10WT2**	Areas 10–9 wait control	A10WT[2:0]			Wait cycles	1	R/W	
		D1	**A10WT1**		1	1	1	7	1		
		D0	**A10WT0**		1	1	0	6	1		
					1	0	1	5			
					1	0	0	4			
					0	1	1	3			
					0	1	0	2			
					0	0	1	1			
					0	0	0	0			
Areas 8–7	**0048128**	DF–9	**–**	reserved	–				–	–	0 when being read.
set-up register	**(HW)**	D8	**A8DRA**	Area 8 DRAM selection	1	Used	0	Not used	0	R/W	
		D7	**A7DRA**	Area 7 DRAM selection	1	Used	0	Not used	0	R/W	
		D6	**A8SZ**	Areas 8–7 device size selection	1	8 bits	0	16 bits	0	R/W	
		D5	**A8DF1**	Areas 8–7	A8DF[1:0]			Number of cycles	1	R/W	
		D4	**A8DF0**	output disable delay time	1	1		3.5	1		
					1	0		2.5			
					0	1		1.5			
					0	0		0.5			
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**A8WT2**	Areas 8–7 wait control	A8WT[2:0]			Wait cycles	1	R/W	
		D1	**A8WT1**		1	1	1	7	1		
		D0	**A8WT0**		1	1	0	6	1		
					1	0	1	5			
					1	0	0	4			
					0	1	1	3			
					0	1	0	2			
					0	0	1	1			
					0	0	0	0			
Areas 6–4	**004812A**	DF–E	**–**	reserved	–				–	–	0 when being read.
set-up register	**(HW)**	DD	**A6DF1**	Area 6	A6DF[1:0]			Number of cycles	1	R/W	
		DC	**A6DF0**	output disable delay time	1	1		3.5	1		
					1	0		2.5			
					0	1		1.5			
					0	0		0.5			
		DB	**–**	reserved	–				–	–	0 when being read.
		DA	**A6WT2**	Area 6 wait control	A6WT[2:0]			Wait cycles	1	R/W	
		D9	**A6WT1**		1	1	1	7	1		
		D8	**A6WT0**		1	1	0	6	1		
					1	0	1	5			
					1	0	0	4			
					0	1	1	3			
					0	1	0	2			
					0	0	1	1			
					0	0	0	0			
		D7	**–**	reserved	–				–	–	0 when being read.
		D6	**A5SZ**	Areas 5–4 device size selection	1	8 bits	0	16 bits	0	R/W	
		D5	**A5DF1**	Areas 5–4	A5DF[1:0]			Number of cycles	1	R/W	
		D4	**A5DF0**	output disable delay time	1	1		3.5	1		
					1	0		2.5			
					0	1		1.5			
					0	0		0.5			
		D3	**–**	reserved					–	–	0 when being read.
		D2	**A5WT2**	Areas 5–4 wait control	A5WT[2:0]			Wait cycles	1	R/W	
		D1	**A5WT1**		1	1	1	7	1		
		D0	**A5WT0**		1	1	0	6	1		
					1	0	1	5			
					1	0	0	4			
					0	1	1	3			
					0	1	0	2			
					0	0	1	1			
					0	0	0	0			

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
TTBR write protect register	**004812D (B)**	D7	**TBRP7**	TTBR register write protect	Writing 01011001(0x59)				0	W	Undefined in read.
		D6	**TBRP6**		removes the TTBR (0x48134)				0		
		D5	**TBRP5**		write protection.				0		
		D4	**TBRP4**		Writing other data sets the				0		
		D3	**TBRP3**		write protection.				0		
		D2	**TBRP2**						0		
		D1	**TBRP1**						0		
		D0	**TBRP0**						0		
Bus control register	**004812E (HW)**	DF	**RBCLK**	BCLK output control	1	Fixed at H	0	Enabled	0	R/W	
		DE	–	reserved	–				0	–	Writing 1 not allowed.
		DD	**RBST8**	Burst ROM burst mode selection	1	8-successive	0	4-successive	0	R/W	
		DC	**REDO**	DRAM page mode selection	1	EDO	0	Fast page	0	R/W	
		DB	**RCA1**	Column address size selection	RCA[1:0]		Size		0	R/W	
		DA	**RCA0**		1	1	11		0		
					1	0	10				
					0	1	9				
					0	0	8				
		D9	**RPC2**	Refresh enable	1	Enabled	0	Disabled	0	R/W	
		D8	**RPC1**	Refresh method selection	1	Self-refresh	0	CBR-refresh	0	R/W	
		D7	**RPC0**	Refresh RPC delay setup	1	2.0	0	1.0	0	R/W	
		D6	**RRA1**	Refresh RAS pulse width selection	RRA[1:0]		Number of cycles		0	R/W	
		D5	**RRA0**		1	1	5		0		
					1	0	4				
					0	1	3				
					0	0	2				
		D4	–	reserved	–				0	–	Writing 1 not allowed.
		D3	**SBUSST**	External interface method selection	1	#BSL	0	A0	0	R/W	
		D2	**SEMAS**	External bus master setup	1	Existing	0	Nonexistent	0	R/W	
		D1	**SEPD**	External power-down control	1	Enabled	0	Disabled	0	R/W	
		D0	**SWAITE**	#WAIT enable	1	Enabled	0	Disabled	0	R/W	
DRAM timing set-up register	**0048130 (HW)**	DF–B	–	reserved	–				–	–	0 when being read.
		DA	–	reserved	–				0	–	Writing 1 not allowed.
		D9	**CEFUNC**	#CE pin function selection	1	High address	0	Normal	0	R/W	
		D8	**CRAS**	Successive RAS mode setup	1	Successive	0	Normal	0	R/W	
		D7	**RPRC1**	DRAM	RPRC[1:0]		Number of cycles		0	R/W	
		D6	**RPRC0**	RAS precharge cycles selection	1	1	4		0		
					1	0	3				
					0	1	2				
					0	0	1				
		D5	–	reserved	–				–	–	0 when being read.
		D4	**CASC1**	DRAM	CASC[1:0]		Number of cycles		0	R/W	
		D3	**CASC0**	CAS cycles selection	1	1	4		0		
					1	0	3				
					0	1	2				
					0	0	1				
		D2	–	reserved	–				–	–	0 when being read.
		D1	**RASC1**	DRAM	RASC[1:0]		Number of cycles		0	R/W	
		D0	**RASC0**	RAS cycles selection	1	1	4		0		
					1	0	3				
					0	1	2				
					0	0	1				
TTBR low-order register	**0048134 (HW)**	DF	**TTBR15**	Trap table base address [15:10]					0	R/W	
		DE	**TTBR14**						0		
		DD	**TTBR13**						0		
		DC	**TTBR12**						0		
		DB	**TTBR11**						0		
		DA	**TTBR10**						0		
		D9	**TTBR09**	Trap table base address [9:0]	Fixed at 0				0	R	0 when being read. Writing 1 not allowed.
		D8	**TTBR08**						0		
		D7	**TTBR07**						0		
		D6	**TTBR06**						0		
		D5	**TTBR05**						0		
		D4	**TTBR04**						0		
		D3	**TTBR03**						0		
		D2	**TTBR02**						0		
		D1	**TTBR01**						0		
		D0	**TTBR00**						0		

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
TTBR high-order register	**0048136 (HW)**	DF	**TTBR33**	Trap table base address [31:28]	Fixed at 0				0	R	0 when being read. Writing 1 not allowed.
		DE	**TTBR32**						0		
		DD	**TTBR31**						0		
		DC	**TTBR30**						0		
		DB	**TTBR2B**	Trap table base address [27:16]	The initial value is set according to the BTA3 pin status. BTA3 = "0": 0x0C0				←	R/W	
		DA	**TTBR2A**								
		D9	**TTBR29**								
		D8	**TTBR28**								
		D7	**TTBR27**								
		D6	**TTBR26**								
		D5	**TTBR25**								
		D4	**TTBR24**								
		D3	**TTBR23**								
		D2	**TTBR22**								
		D1	**TTBR21**								
		D0	**TTBR20**								
High-speed DMA Ch.0 low-order memory address register	**0048204 (HW)**	DF	**D0LADR15**	High-speed DMA Ch.0 memory address (low-order 16 bits)					X	R/W	
		DE	**D0LADR14**						X		
		DD	**D0LADR13**						X		
		DC	**D0LADR12**						X		
		DB	**D0LADR11**						X		
		DA	**D0LADR10**						X		
		D9	**D0LADR9**						X		
		D8	**D0LADR8**						X		
		D7	**D0LADR7**						X		
		D6	**D0LADR6**						X		
		D5	**D0LADR5**						X		
		D4	**D0LADR4**						X		
		D3	**D0LADR3**						X		
		D2	**D0LADR2**						X		
		D1	**D0LADR1**						X		
		D0	**D0LADR0**						X		
High-speed DMA Ch.0 high-order memory address register	**0048206 (HW)**	DF–C	–	reserved	–				–	–	Undefined in read.
		DB	**D0HADR11**	High-speed DMA Ch.0 memory address (high-order 12 bits)					X	R/W	
		DA	**D0HADR10**						X		
		D9	**D0HADR9**						X		
		D8	**D0HADR8**						X		
		D7	**D0HADR7**						X		
		D6	**D0HADR6**						X		
		D5	**D0HADR5**						X		
		D4	**D0HADR4**						X		
		D3	**D0HADR3**						X		
		D2	**D0HADR2**						X		
		D1	**D0HADR1**						X		
		D0	**D0HADR0**						X		
High-speed DMA Ch.0 transfer counter register	**0048208 (HW)**	DF	**D0LEN15**	High-speed DMA Ch.0 transfer counter					X	R/W	
		DE	**D0LEN14**						X		
		DD	**D0LEN13**						X		
		DC	**D0LEN12**						X		
		DB	**D0LEN11**						X		
		DA	**D0LEN10**						X		
		D9	**D0LEN9**						X		
		D8	**D0LEN8**						X		
		D7	**D0LEN7**						X		
		D6	**D0LEN6**						X		
		D5	**D0LEN5**						X		
		D4	**D0LEN4**						X		
		D3	**D0LEN3**						X		
		D2	**D0LEN2**						X		
		D1	**D0LEN1**						X		
		D0	**D0LEN0**						X		
High-speed DMA Ch.0 control register	**004820A (B)**	D7–3	–	reserved	–				–	–	Undefined in read.
		D2	**D0SIZ**	Ch.0 data size control	1	Half-word	0	Byte	0	R/W	
		D1	**D0INC**	Ch.0 address control	1	Inc	0	Dec	0	R/W	
		D0	**D0DIR**	Ch.0 transfer direction control	1	Memory write	0	Memory read	0	R/W	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
High-speed	**0048214**	DF	**D1LADR15**	High-speed DMA Ch.1					X	R/W	
DMA Ch.1	**(HW)**	DE	**D1LADR14**	memory address					X		
low-order		DD	**D1LADR13**	(low-order 16 bits)					X		
memory		DC	**D1LADR12**						X		
address		DB	**D1LADR11**						X		
register		DA	**D1LADR10**						X		
		D9	**D1LADR9**						X		
		D8	**D1LADR8**						X		
		D7	**D1LADR7**						X		
		D6	**D1LADR6**						X		
		D5	**D1LADR5**						X		
		D4	**D1LADR4**						X		
		D3	**D1LADR3**						X		
		D2	**D1LADR2**						X		
		D1	**D1LADR1**						X		
		D0	**D1LADR0**						X		
High-speed	**0048216**	DF–C	–	reserved	–				–	–	Undefined in read.
DMA Ch.1	**(HW)**	DB	**D1HADR11**	High-speed DMA Ch.1					X	R/W	
high-order		DA	**D1HADR10**	memory address					X		
memory		D9	**D1HADR9**	(high-order 12 bits)					X		
address		D8	**D1HADR8**						X		
register		D7	**D1HADR7**						X		
		D6	**D1HADR6**						X		
		D5	**D1HADR5**						X		
		D4	**D1HADR4**						X		
		D3	**D1HADR3**						X		
		D2	**D1HADR2**						X		
		D1	**D1HADR1**						X		
		D0	**D1HADR0**						X		
High-speed	**0048218**	DF	**D1LEN15**	High-speed DMA Ch.1					X	R/W	
DMA Ch.1	**(HW)**	DE	**D1LEN14**	transfer counter					X		
transfer		DD	**D1LEN13**						X		
counter		DC	**D1LEN12**						X		
register		DB	**D1LEN11**						X		
		DA	**D1LEN10**						X		
		D9	**D1LEN9**						X		
		D8	**D1LEN8**						X		
		D7	**D1LEN7**						X		
		D6	**D1LEN6**						X		
		D5	**D1LEN5**						X		
		D4	**D1LEN4**						X		
		D3	**D1LEN3**						X		
		D2	**D1LEN2**						X		
		D1	**D1LEN1**						X		
		D0	**D1LEN0**						X		
High-speed	**004821A**	D7–3	–	reserved	–				–	–	Undefined in read.
DMA Ch.1	**(B)**	D2	**D1SIZ**	Ch.1 data size control	1	Half-word	0	Byte	0	R/W	
control register		D1	**D1INC**	Ch.1 address control	1	Inc	0	Dec	0	R/W	
		D0	**D1DIR**	Ch.1 transfer direction control	1	Memory write	0	Memory read	0	R/W	
IDMA base	**0048230**	DF	**DBASEL15**	IDMA base address					0	R/W	
address low-	**(HW)**	DE	**DBASEL14**	low-order 16 bits					0		
order register		DD	**DBASEL13**	(Initial value: 0x0C003A0)					0		
		DC	**DBASEL12**						0		
		DB	**DBASEL11**						0		
		DA	**DBASEL10**						0		
		D9	**DBASEL9**						1		
		D8	**DBASEL8**						1		
		D7	**DBASEL7**						1		
		D6	**DBASEL6**						0		
		D5	**DBASEL5**						1		
		D4	**DBASEL4**						0		
		D3	**DBASEL3**						0		
		D2	**DBASEL2**						0		
		D1	**DBASEL1**						0		
		D0	**DBASEL0**						0		

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
IDMA base	**0048232**	DF–C	**–**	reserved	–				–	–	Undefined in read.
address	**(HW)**	DB	**DBASEH11**	IDMA base address					0	R/W	
high-order		DA	**DBASEH10**	high-order 12 bits					0		
register		D9	**DBASEH9**	(Initial value: 0x0C003A0)					0		
		D8	**DBASEH8**						0		
		D7	**DBASEH7**						1		
		D6	**DBASEH6**						1		
		D5	**DBASEH5**						0		
		D4	**DBASEH4**						0		
		D3	**DBASEH3**						0		
		D2	**DBASEH2**						0		
		D1	**DBASEH1**						0		
		D0	**DBASEH0**						0		
IDMA start	**0048234**	D7	**DSTART**	IDMA start	1	IDMA start	0	Stop	0	R/W	
register	**(B)**	D6–0	**DCHN**	IDMA channel number	0 to 127				0	R/W	
IDMA enable	**0048235**	D7–1	**–**	reserved	–				–	–	
register	**(B)**	D0	**DMAEN**	DMA enable	1	Enabled	0	Disabled	0	R/W	

6 Internal Memory

This chapter explains the internal memory configuration.

6.1 RAM

The E0C33A104 has a 2K-byte RAM built-in.

The internal RAM is a 32-bit sized device and is mapped from address 0x0 to address 0x7FF. Data can be read/written in 1 cycle regardless of data size (byte half-word or word).

Figure 6.1.1 shows the basic internal memory map (when ARAM described later is not used).

Area	Address		Bus cycle	Device size
Area 2	0x007FFFF 0x0060000	(Reserved) For CPU core or debug mode	Fixed at 3 cycles	16 bits
Area 1	0x005FFFF 0x0050000	(Mirror of internal I/O)	Fixed at 2 cycles	8,16 bits
	0x004FFFF 0x0040000	Internal I/O		
Area 0	0x003FFFF 0x0000800	(Mirror of internal RAM)	Fixed at 1 cycle	32 bits
	0x00007FF 0x0000000	Internal RAM (2KB)		

Figure 6.1.1 Basic Internal Memory Map

Area 2 is used in debug mode only and it cannot be accessed in user mode (normal program execution status).

∗ Mirror areas

The areas in which any device is not mapped in the internal memory space are used as mirror areas for the internal RAM and internal I/O memory. When the mirror area is specified, the original device is actually accessed.

For example, when the mirror area for the internal RAM (0x800 to 0x3FFFF) is specified, the internal RAM is accessed using the low-order 11 bits of address (0x0 to 0x7FF) as the valid address.

6.2 ARAM

The E0C33A104 contains a 4K-byte RAM as well as the internal RAM described in the previous section. This RAM is an additional memory of which the purpose of use is selectable by pin settings, and is called ARAM (additional RAM) to distinguish from the internal RAM.
The ARAM operating mode can be specified using the ARAMMD[1:0] pins as shown below.

Table 6.2.1 ARAM Mode Setup Pins

Pin name	Pin No.		Pull-up	I/O	Function
	QFP5-128	QFP15-128			
ARAMMD1	100	97	–	I	ARAM operating mode setup pins
ARAMMD0	101	98	–	I	ARAMMD1 ARAMMD0 ARAM mode Low Low Expanded RAM mode Low High Not allowed High Low Not allowed High High ARAM is not used.

When both the ARAMMD1 and ARAMMD0 pins are set to high, the ARAM is disconnected from the internal bus. The memory configuration in this case is shown in Figure 6.1.1.

Note: The ARAM mode can be switched while the IC is operating. However, the ARAM must not be accessed when switching. Basically, in order to prevent malfunction, use the ARAM under a condition that cannot change the input level of the setup pins.
Furthermore, when setting the pin to high level, be sure to connect the pin to VDD because the pin does not have a pull-up resistor.

6.2.1 Expanded RAM Mode

Expanded RAM mode is set when both the ARAMMD1 and ARAMMD0 pins are pulled down to low level. In this mode, the ARAM is mapped from address 0x800 to address 0x17FF and the area can be used the same as the internal RAM. Thus the internal RAM area is expanded into 6K bytes. The expanded area is 32-bit size and is accessed in the same condition as the internal RAM (1 cycle operation).

Area	Address		Bus cycle	Device size
Area 2	0x007FFFF 0x0060000	(Reserved) For CPU core or debug mode	Fixed at 3 cycles	16 bits
Area 1	0x005FFFF 0x0050000	(Mirror of internal I/O)	Fixed at 2 cycles	8,16 bits
	0x004FFFF 0x0040000	Internal I/O		
Area 0	0x003FFFF 0x0001800	(Mirror of ARAM)	Fixed at 1 cycle	32 bits
	0x00017FF 0x0000800	ARAM (4KB)		
	0x00007FF 0x0000000	Internal RAM (2KB)		

Figure 6.2.1 Internal Memory Map (Expanded RAM Mode)

The no-mapped area from address 0x1800 to address 0x3FFF becomes the mirror area for the ARAM. The internal RAM (0x0 to 0x7FF) cannot be accessed through this area.

7 External System Interface

The E0C33A104 provides an external system interface using its internal Bus Control Unit (BCU). The types and sizes of memory and peripheral I/O devices can be set for each area of the memory map and can be controlled directly by the BCU. This unit also supports a direct interface for DRAM and burst ROM. This chapter describes how to control the external system interface, and how it operates.

Note: The control registers of the external system interface shown in this chapter are mapped to the internal 16-bit I/O area. Therefore, the addresses of these control registers are indicated by half-word (16-bit) addresses unless otherwise specified. Note that the control registers can be accessed in bytes, half-words, or words.

7.1 Pin Assignment for External System Interface

7.1.1 I/O Pin List

Table 7.1.1 lists the pins used for the external system interface.

Table 7.1.1 I/O Pin List

Pin name	Pin No.		I/O	Pull-up	Function
	QFP5-128	QFP15-128			
A[0]/#BSL	77	74	O	–	Address bus (A0) / Bus strobe (Low-byte)
A[23:1]	70–76, 111–126	67–73, 108–123	O	–	Address bus (A1–A23)
D[15:0]	78,80–87, 90,93–98	75,77–84, 87,90–95	I/O	–	Data bus (D0–D15)
#CE10	6	3	O	–	Area 10 chip enable
#CE9/#CE17	5	2	O	–	Area 9/17 chip enable
#CE8/#RAS1/ #CE14/#RAS3	4	1	O	–	Area 8/14 chip enable / DRAM Row strobe
#CE7/#RAS0/ #CE13/#RAS2	3	128	O	–	Area 7/13 chip enable / DRAM Row strobe
#CE6	2	127	O	–	Area 6 chip enable
#CE5/#CE15	1	126	O	–	Area 5/15 chip enable
#CE4/#CE11	128	125	O	–	Area 4/11 chip enable
#RD	109	106	O	–	Read signal
#WRL/#WR/#WE	108	105	O	–	Write (Low-byte) / Write / DRAM write
#WRH/#BSH	107	104	O	–	Write (High-byte) / Bus strobe (High-byte)
#HCAS	105	102	O	–	DRAM column address strobe (High-byte)
#LCAS	106	103	O	–	DRAM column address strobe (Low-byte)
BCLK	104	101	O	–	Bus clock output
#BUSREQ	103	100	I	–	Bus release request
#BUSACK	102	99	O	–	Bus request acknowledge
#WAIT	99	96	I	–	Wait cycle request
#DRD/R81	57	54	O	–	DRAM read signal / Output port
#DWE/P14	50	47	O	–	DRAM write (Low-byte) / I/O port

7.1.2 Combination of System Bus Control Signals

The bus control signal pins that have two or more functions have their functionality determined when an interface method is selected by a program. The BCU contains an ordinary external system interface (an interface method is selectable) and a DRAM interface.

Table 7.1.2 Interface Selection

Interface type	Interface method	Control bit
External system interface	A0 system (default)	SBUSST(D3/0x4812E) = "0"
	#BSL system	SBUSST(D3/0x4812E) = "1"
DRAM interface	2CAS system (fixed)	None

These control bits are initialized to "0" at cold start.
When the E0C33A104 is hot-started, these bits retain their status before the chip was reset.

Table 7.1.3 shows combinations of control signals classified by each interface method.

Table 7.1.3 Combinations of Bus Control Signals

Pin No.		External system interface		DRAM interface
QFP5-128	QFP15-128	A0 system	#BSL system	2CAS system
77	74	A0	#BSL	–
108	105	#WRL	#WR	#WE
107	104	#WRH	#BSH	–
105	102	–	–	#HCAS
106	103	–	–	#LCAS
1–6,128	1–3, 125–128	#CEx	#CEx	#RASx*

∗ When using DRAM, the #CE output pins in areas 7−8 (or areas 13–14) function as the #RAS1−2 (#RAS3–4) pins.

7.2 External Memory Map and Chip Enable Switchover

The BCU of the E0C33A104 has a 24-bit external address bus (A[23:0]) and a 16-bit external data bus (D[15:0]), allowing an address space of up to 16MB to be accessed. By default, this address space is divided into 11 areas (areas 0 to 10) for management purposes. Of these, areas 4 to 10 are open to an external system, each provided with an independent chip-enable pin (#CE[10:4]).

The E0C33A104 is limited to 24 available pins for the address bus and 7 pins for the #CE output due to its package structure. However, the #CE[4:10] output pins can be switched using software between the #CE6, #CE[11:10], #CE[15:13], and #CE17 output pins. CEFUNC (D9) / DRAM timing set-up register (0x48130) is used for this switching.

Table 7.2.1 Switching of #CE Output

CEFUNC = "0"	CEFUNC = "1"
#CE4	#CE11
#CE5	#CE15
#CE6	#CE6
#CE7/#RAS0	#CE13/#RAS2
#CE8/#RAS1	#CE14/#RAS3
#CE9	#CE17
#CE10	#CE10

(Default: CEFUNC = "0")

The high-order areas that are made available for use by writing "1" to CEFUNC can be larger in size than the default low-order areas. For example, when using DRAM in default settings, the available space is 4MB in areas 7 and 8. However, if areas 13 and 14 are used, up to 32MB of DRAM can be used. The same applies to the other areas. Although the E0C33A104 has only 24 address output pins, it features 28-bit internal address processing.

Figure 7.2.1 shows a memory map for an external system.

Area	Address	
Area 10 (#CE10) SRAM type Burst ROM type 8 or 16 bits	0x0FFFFFF 0x0C00000	External memory 6 (4MB)
Area 9 (#CE9) SRAM type Burst ROM type 8 or 16 bits	0x0BFFFFF 0x0800000	External memory 5 (4MB)
Area 8 (#CE8/#RAS1) SRAM type DRAM type 8 or 16 bits	0x07FFFFF 0x0600000	External memory 4 (2MB)
Area 7 (#CE7/#RAS0) SRAM type DRAM type 8 or 16 bits	0x05FFFFF 0x0400000	External memory 3 (2MB)
Area 6 (#CE6) SRAM type	0x03FFFFF 0x0380000	External I/O (16-bit device)
	0x037FFFF 0x0300000	External I/O (8-bit device)
Area 5 (#CE5) SRAM type 8 or 16 bits	0x02FFFFF 0x0200000	External memory 2 (1MB)
Area 4 (#CE4) SRAM type 8 or 16 bits	0x01FFFFF 0x0100000	External memory 1 (1MB)

When #CE4–10 are used

Area	Address	
Area 17 (#CE17) SRAM type 8 or 16 bits	0xBFFFFFF 0x9000000	(Mirror of External memory 6)
	0x8FFFFFF 0x8000000	External memory 6 (16MB)
Area 15 (#CE15) SRAM type 8 or 16 bits	0x5FFFFFF 0x5000000	(Mirror of External memory 5)
	0x4FFFFFF 0x4000000	External memory 5 (16MB)
Area 14 (#CE14/#RAS3) SRAM type DRAM type 8 or 16 bits	0x3FFFFFF 0x3000000	External memory 4 (16MB)
Area 13 (#CE13/#RAS2) SRAM type DRAM type 8 or 16 bits	0x2FFFFFF 0x2000000	External memory 3 (16MB)
Area 11 (#CE11) SRAM type 8 or 16 bits	0x17FFFFF 0x1000000	External memory 2 (8MB)
Area 10 (#CE10) SRAM type Burst ROM type 8 or 16 bits	0x0FFFFFF 0x0C00000	External memory 1 (4MB)
Area 6 (#CE6) SRAM type	0x03FFFFF 0x0380000	External I/O (16-bit device)
	0x037FFFF 0x0300000	External I/O (8-bit device)

When #CE6, 10, 11, 13–15 and 17 are used

Figure 7.2.1 External System Memory Map

7.3 Setting External Bus Conditions

The type, size, and wait conditions of a device connected to the external bus can be individually set for each area using the control register (0x48126 to 0x48130). The following explains the available setup conditions individually for each area. For details on how to set the DRAM interface conditions, refer to Section 7.6, "DRAM Direct Interface".

The control register used to set bus conditions is initialized at cold start. Therefore, reset up this register using software according to the external device configuration and specifications.
When the E0C33A104 is hot-started, the setup contents and pins retain their previous status before a reset.

7.3.1 Setting Device Type and Size

Table 7.3.1 shows the types of devices that can be connected directly to each area.

Table 7.3.1 Device Type

(In case of CEFUNC = "0")

Area	SRAM type	DRAM type	Burst ROM type	Control bit
10	○	X	○	A10DRA(D8)/Areas 10–9 set-up register(0x48126)
9	○	X	○	A9DRA(D7)/Areas 10–9 set-up register(0x48126)
8	○	○	X	A8DRA(D8)/Areas 8–7 set-up register(0x48128)
7	○	○	X	A7DRA(D7)/Areas 8–7 set-up register(0x48128)
6	○	X	X	None
5	○	X	X	None
4	○	X	X	None

○: Can be connected X: Cannot be connected

(In case of CEFUNC = "1")

Area	SRAM type	DRAM type	Burst ROM type	Control bit
17	○	X	X	None
15	○	X	X	None
14	○	○	X	A14DRA(D8)/Areas 14–13 set-up register(0x48122)
13	○	○	X	A13DRA(D7)/Areas 14–13 set-up register(0x48122)
11	○	X	X	None
10	○	X	○	A10DRA(D8)/Areas 10–9 set-up register(0x48126)
6	○	X	X	None

○: Can be connected X: Cannot be connected

When connecting burst ROM or DRAM, write "1" to each corresponding control bit. These control bits are reset to "0" (SRAM type) at cold start.

The device size can be set to 8 or 16 bits once every two areas except for area 6. Area 6 alone has its first half (0x300000–0x37FFFF) fixed to an 8-bit device and the second half (0x380000–0x3FFFFF) fixed to a 16-bit device.

Table 7.3.2 Device Size Control Bits

Area	Control bit
17	A18SZ(DE)/Areas 18–15 set-up register(0x48120)
15	A16SZ(D6)/Areas 18–15 set-up register(0x48120)
14, 13	A14SZ(D6)/Areas 14–13 set-up register(0x48122)
11	A12SZ(D6)/Areas 12–11 set-up register(0x48124)
10, 9	A10SZ(D6)/Areas 10–9 set-up register(0x48126)
8, 7	A8SZ(D6)/Areas 8–7 set-up register(0x48128)
5, 4	A5SZ(D6)/Areas 6–4 set-up register(0x4812A)

At cold start, each area by default is set to 16 bits.
When using an 8-bit device, write "1" to the control bit.

Note: The BCU supports 16-bit burst ROM. Therefore, when connecting burst ROM to area 10 or area 9, do not set the device size to 8 bits (A10SZ = "1").

For differences in bus operation due to the device size and access data size, refer to Section 7.4.2, "Bus Operation of External Memory".

7.3.2 Setting SRAM Timing Conditions

The areas set for the SRAM allow wait cycles and output disable delay time to be set.

Number of wait cycles: 0 to 7 (incremented in units of one cycle)
Output disable delay time: 0.5, 1.5, 2.5, 3.5 cycles

This selection can be made once every two areas except for area 6.

Table 7.3.3 Timing Condition Setting Bits (for SRAM type)

Area	Number of wait cycles	Output disable delay time	Control register
17	A18WT[2:0](D[A:8])	A18DF[1:0](D[D:C])	Areas 18–15 set-up register(0x48120)
15	A16WT[2:0](D[2:0])	A16DF[1:0](D5:4])	Areas 18–15 set-up register(0x48120)
14, 13	A14WT[2:0](D[2:0])	A14DF[1:0](D5:4])	Areas 14–13 set-up register(0x48122)
11	A12WT[2:0](D[2:0])	A12DF[1:0](D5:4])	Areas 12–11 set-up register(0x48124)
10, 9	A10WT[2:0](D[2:0])	A10DF[1:0](D5:4])	Areas 10–9 set-up register(0x48126)
8, 7	A8WT[2:0](D[2:0])	A8DF[1:0](D[5:4])	Areas 8–7 set-up register(0x48128)
6	A6WT[2:0](D[A:8])	A6DF[1:0](D[D:C])	Areas 6–4 set-up register(0x4812A)
5, 4	A5WT[2:0](D[2:0])	A5DF[1:0](D[5:4])	Areas 6–4 set-up register(0x4812A)

At cold start, the number of wait cycles is set to 7 and the output disable delay time is set to 3.5 cycles. Reset up these parameters as necessary using software according to specifications of the connected device.
At hot start, these parameters retain their previous settings before a reset.

Wait cycles

When the number of wait cycles is set for an area using the control bit, the BCU extends the bus cycle for a duration equivalent to the wait cycles set when it accesses the area. Set the desired wait cycles according to the bus clock frequency and the external device's access time. Separately from the wait cycles set here, a wait request from an external device can also be accepted using the #WAIT pin. Since the settings of wait cycles using software are made once every two areas, use this external wait request function if you want the wait cycles to be controlled individually in each area or if you need 7 or more wait cycles. For an external wait request to be accepted, write "1" (default = "0") to SWAIT (D0) / Bus control register (0x4812E) to enable the #WAIT pin.
For timing charts for bus cycles and when wait cycles are inserted, refer to Section 7.5, "Bus Cycles in External System Interface".

If the number of wait cycles is set to 0 and no external wait is requested, the basic read cycle (read in byte or half-word) for the SRAM external device consists of one cycle. If wait cycles are set, because these cycles are added, the bus read cycle consists of [number of wait cycles + 1] (providing that there is no external wait).
On the other hand, the basic write cycle consists of at least two cycles. This does not change regardless of whether zero or one wait cycle is set. If the number of wait cycles set is 2 or more, the bus cycle is actually extended. In this case, the bus write cycle consists of [number of wait cycles + 1], as in the case of read cycles (providing that there is no external wait).

Output disable delay time

In cases when a device having a long output disable time is connected, if a read cycle for that device is followed by the next access, contention for the data bus may occur. (Due to the fact the read device's data bus is not placed in the high-impedance state.) The output disable delay time is provided to prevent such data bus contention. This is accomplished by inserting a specified number of cycles between a read cycle and the next bus operation.
Check the specifications of the device to be connected before setting the output disable delay time.
By default, the output disable delay time is inserted only in the following cases:

- when a read cycle from the external device that has had an output disable delay time set is followed by a write cycle performed by the CPU; and
- when a read cycle from the external device that has had an output disable delay time set is followed by a read cycle for a different area (including the internal device).

Conversely, no output disable delay time is inserted in the following conditions:

- immediately after a write cycle, and
- during a successive read from the same external device.

7.3.3 Setting Timing Conditions of Burst ROM

Wait cycles

If burst ROM is selected for area 10 or 9, the wait cycles to be inserted in the burst read cycle can be selected in a range from 0 to 3 cycles. A10BW[1:0] (D[A:9]) / Areas 10–9 set-up register (0x48126) is used for this selection. This selection is applied simultaneously to areas 10 and 9, so wait cycles can not be chosen individually for each area. The wait cycles set at cold start is 0.

Even for a burst read, the SRAM settings of wait cycles in the first bus operation are valid. (Refer to A10WT[2:0] in the foregoing section.)
The wait cycles set by A10BW[1:0] are inserted into the burst cycles after the first bus operation.

In addition, when burst ROM is selected, no wait cycles can be inserted into the read cycle via the #WAIT pin. For writing to an area that has had burst ROM selected, an SRAM write cycle is executed. In this case, both the SRAM settings of wait cycles and those input via the #WAIT pin are valid.

Burst mode

The burst mode can be selected between an eight-consecutive-burst and a four-consecutive-burst mode. RBST8 (DD) / Bus control register (0x4812E) is used for this selection. The eight-consecutive-burst mode is selected by writing "1" to RBST8 and the four-consecutive-burst mode is selected by setting the bit to "0". At cold start, the four-consecutive-burst mode is set by default.

7.4 Bus Operation

7.4.1 Data Arrangement in Memory

The E0C33 Family of devices handle data in bytes (8 bits), half-words (16 bits), and words (32 bits). When accessing data in memory, it is necessary to specify a boundary address that conforms to the data size involved. Specification of an invalid address causes an address error exception. For instructions (e.g., stack manipulation or branch instructions) that rewrite the SP (stack pointer) or PC (program counter), the specified addresses are forcibly modified to appropriate boundary addresses. Therefore, no address error exception occurs in this type of instruction. For details about the address error exception, refer to the "E0C33000 Core CPU Manual".
Table 7.4.1 shows the data arrangement in memory, classified by data type.

Table 7.4.1 Data Arrangement in Memory

Data type	Arranged location
Byte data	Byte boundary address (all addresses)
Half-word data	Half-word boundary address (A[0]="0")
Word data	Word boundary address (A[1:0]="00")

The half-word and word data in memory area accessed in little-endian format.

Table 7.4.2 Data Arrangement in Little-Endian Format

Data type	Boundary address	Boundary address + 1	Boundary address + 2	Boundary address + 3
Byte data	DB1[7:0]	(DB2[7:0])	(DB3[7:0])	(DB4[7:0])
Half-word data	DH1[7:0]	DH1[15:8]	(DH2[7:0])	(DH2[15:8])
Word data	DW1[7:0]	DW1[15:8]	DW1[23:16]	DW1[31:24]

To increase memory efficiency, try to locate the same type of data at continuous locations on exact boundary addresses in order to minimize invalid areas.

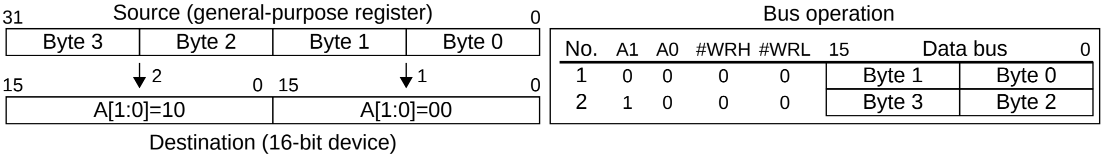
7.4.2 Bus Operation of External Memory

The external data bus is 16-bits wide. For this reason, more than one bus operation occurs depending on the device size and the data size of the instruction executed, as shown in Table 7.4.3.

Table 7.4.3 Number of Bus Operation Cycles

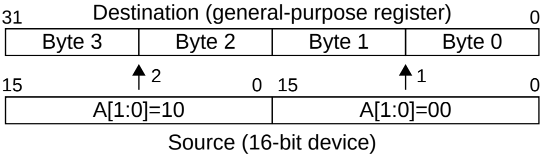
Data size to be accessed	Devise size	Number of bus operation cycles	Remarks
32 bits	16 bits	2	
16 bits	16 bits	1	
8 bits	16 bits	1	The low-order byte is accessed when the LSB of the address (A[0]) is "0" or the #BSL signal is L. The high-order byte is accessed when the LSB of the address (A[0]) is "1" or the #BSH signal is H.
32 bits	8 bits	4	The 8-bit device must be connected to the low-order 8 bits of the data bus.
16 bits	8 bits	2	The 8-bit device must be connected to the low-order 8 bits of the data bus.
8 bits	8 bits	1	The 8-bit device must be connected to the low-order 8 bits of the data bus.

The following diagram shows sample bus operations where the A0 system is used.



No.	A1	A0	#WRH	#WRL	Data bus 15	Data bus 0
1	0	0	0	0	Byte 1	Byte 0
2	1	0	0	0	Byte 3	Byte 2

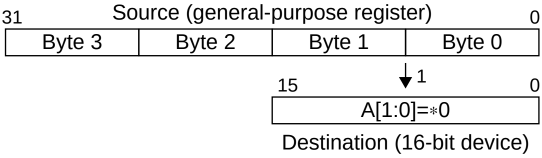
Figure 7.4.1 Word Data Writing to a 16-bit Device



Bus operation

No.	A1	A0	#WRH	#WRL	15 Data bus	0
1	0	0	1	1	Byte 1	Byte 0
2	1	0	1	1	Byte 3	Byte 2

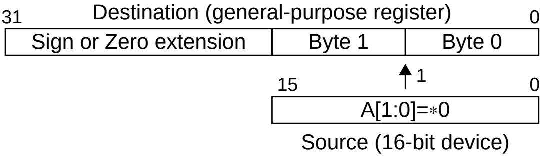
Figure 7.4.2 Word Data Reading from a 16-bit Device


31
Source (general-purpose register)
0
Byte 3
Byte 2
Byte 1
Byte 0
15
1
0
A[1:0]=*0
Destination (16-bit device)

Bus operation

No.	A1	A0	#WRH	#WRL	15 Data bus	0
1	*	0	0	0	Byte 1	Byte 0

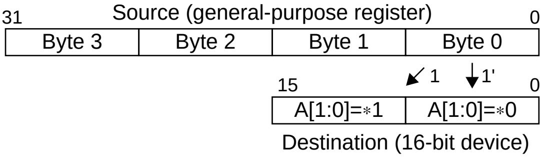
Figure 7.4.3 Half-word Data Writing to a 16-bit Device



Bus operation

No.	A1	A0	#WRH	#WRL	15 Data bus	0
1	*	0	1	1	Byte 1	Byte 0

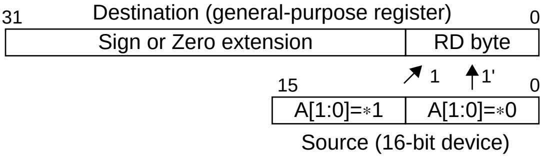
Figure 7.4.4 Half-word Data Reading from a 16-bit Device


31
Source (general-purpose register)
0
Byte 3
Byte 2
Byte 1
Byte 0
15
1
1'
0
A[1:0]=*1
A[1:0]=*0
Destination (16-bit device)

Bus operation

No.	A1	A0	#WRH	#WRL	15 Data bus	0
1	*	1	0	1	Byte 0	Data retained
1'	*	0	1	0	Data retained	Byte 0

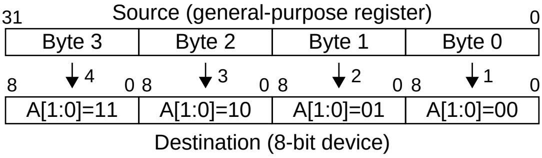
Figure 7.4.5 Byte Data Writing to a 16-bit Device



Bus operation

No.	A1	A0	#WRH	#WRL	15 Data bus	0
1	*	1	1	1	RD byte	Ignored
1'	*	0	1	1	Ignored	RD byte

Figure 7.4.6 Byte Data Reading from a 16-bit Device

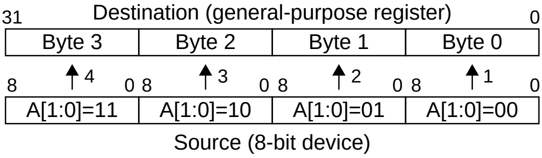


Bus operation

No.	A1	A0	#WRH	#WRL	15 Data bus	0
1	0	0	X	0	Data retained	Byte 0
2	0	1	X	0	Data retained	Byte 1
3	1	0	X	0	Data retained	Byte 2
4	1	1	X	0	Data retained	Byte 3

(X: Not connected/Unused)

Figure 7.4.7 Word Data Writing to an 8-bit Device

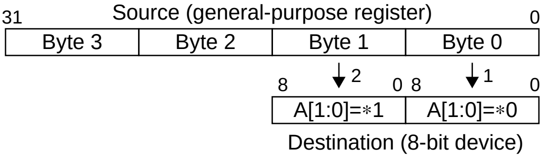


Bus operation

No.	A1	A0	#WRH	#WRL	15 Data bus	0
1	0	0	X	1	Ignored	Byte 0
2	0	1	X	1	Ignored	Byte 1
3	1	0	X	1	Ignored	Byte 2
4	1	1	X	1	Ignored	Byte 3

(X: Not connected/Unused)

Figure 7.4.8 Word Data Reading from an 8-bit Device

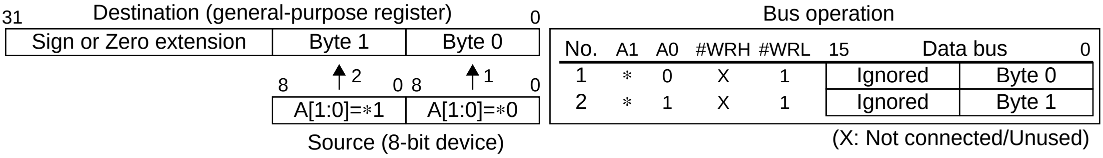


Bus operation

No.	A1	A0	#WRH	#WRL	15 Data bus	0
1	*	0	X	0	Data retained	Byte 0
2	*	1	X	0	Data retained	Byte 1

(X: Not connected/Unused)

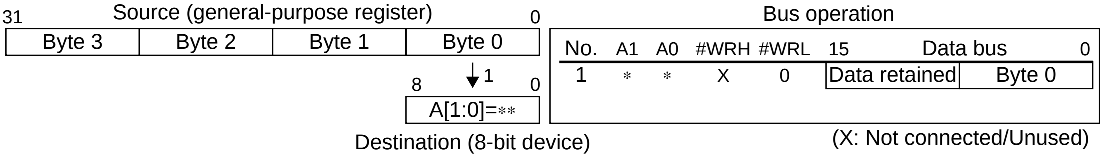
Figure 7.4.9 Half-word Data Writing to an 8-bit Device



No.	A1	A0	#WRH	#WRL	15 Data bus	0
1	*	0	X	1	Ignored	Byte 0
2	*	1	X	1	Ignored	Byte 1

(X: Not connected/Unused)

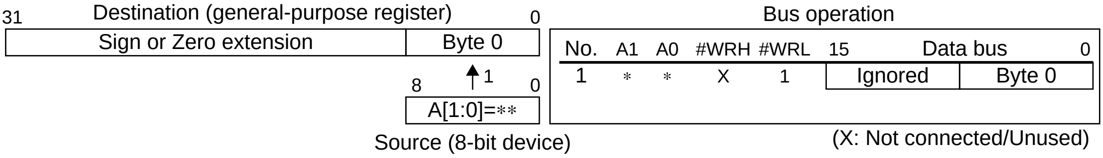
Figure 7.4.10 Half-word Data Reading from an 8-bit Device



No.	A1	A0	#WRH	#WRL	15 Data bus	0
1	*	*	X	0	Data retained	Byte 0

(X: Not connected/Unused)

Figure 7.4.11 Byte Data Writing to an 8-bit Device



No.	A1	A0	#WRH	#WRL	15 Data bus	0
1	*	*	X	1	Ignored	Byte 0

(X: Not connected/Unused)

Figure 7.4.12 Byte Data Reading from an 8-bit Device

7.4.3 Bus Clock

The CPU system clock is used to clock the bus. If the CPU is operated with the low-speed (OSC1) clock, the bus is also clocked with the unit.

If the CPU is operated with the high-speed (OSC3) clock, the operating frequency can be switched in four steps (1/8, 1/4, 1/2, or 1/1 of f_{OSC3}) by using CLKDT[1:0] (D[7:6]) / Power control register (0x40140). The setting of this division ratio also affects the bus clock.

For details about the settings of the system clock, refer to Section 10.1, "Oscillation Circuits".

The bus clock is also output from the BCLK pin to an external device.

Bus clock operation during standby is as follows:

Basic HALT mode: the BCU and bus clock continue operating. DRAM can be refreshed.
HALT2 mode: the BCU and bus clock are stopped.
SLEEP mode: the BCU and bus clock are stopped.

7.5 Bus Cycles in External System Interface

The following shows the basic bus cycles of the external system interface. For detailed timings, refer to Section 18.7, "AC Characteristics". For examples of timings in each device, refer to the Appendix.

7.5.1 SRAM Read Cycles

Basic read cycle with no wait mode

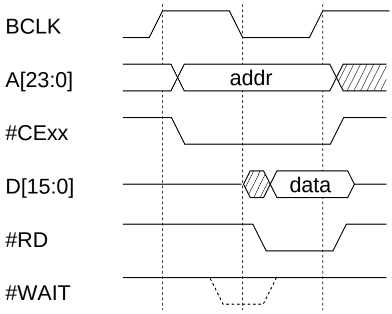


Figure 7.5.1 Basic Read Cycle with No Wait

Read cycle with wait mode

Example: When the BCU has no internal wait mode and 2 wait cycles via #WAIT pin are inserted

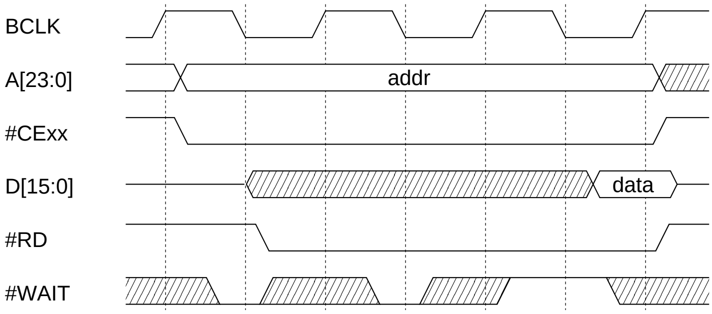


Figure 7.5.2 Read Cycle with Wait

The #WAIT signal is sampled at the falling edge of the transition of BCLK (bus clock) and when it is sampled on an inactive (high level), the read cycle is terminated.

Note: Insertion of wait cycles via the #WAIT pin is possible only when the device for bus conditions is set for SRAM, and SWAIT (D0) / Bus control register (0x4812E) is enabled for waiting.

The above example shows a read cycle when a wait mode is inserted via the #WAIT signal. A wait mode consisting of 0 to 7 cycles can also be inserted using the wait control bits. The settings of these bits can also be used in combination with the #WAIT signal. In this case as well, the #WAIT signal is sampled at the falling edge of the transition of BCLK. However, even when the #WAIT signal is inactive before the wait cycles set by the wait control bits are terminated, the read cycle is not terminated at that time.

7.5.2 SRAM Write Cycles

Basic write cycle with no wait mode

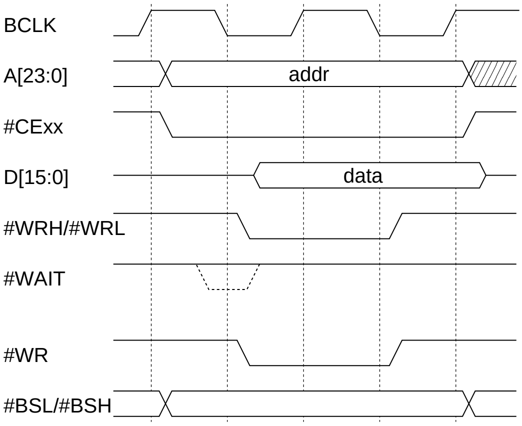


Figure 7.5.3 Basic Write Cycle with No Wait

Write cycle with wait mode

Example: When the BCU has no internal wait mode, and 1 wait cycle is inserted via the #WAIT pin

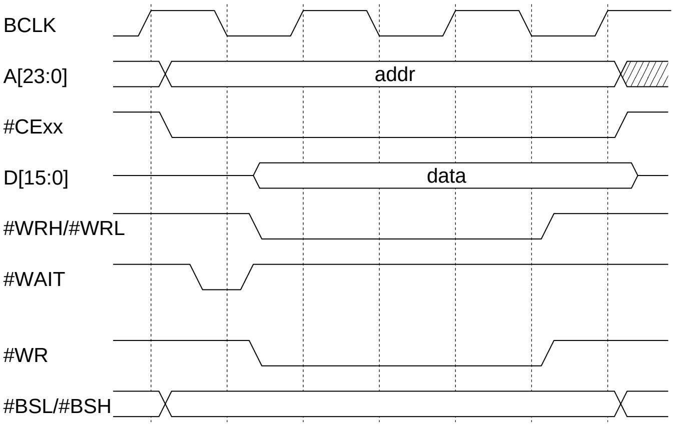


Figure 7.5.4 Write Cycle with Wait

The #WAIT signal is sampled at the falling edge of the transition of BCLK (bus clock), and the write cycle is terminated in the cycle immediately following the cycle in which the #WAIT signal was sampled in an inactive (high level).

Note: Insertion of wait cycles via the #WAIT pin is possible only when the device for bus conditions is set to SRAM and SWAIT (D0) / Bus control register (0x4812E) is enabled for waiting.

The above example shows a write cycle when a wait mode is inserted via the #WAIT signal. A wait mode consisting of 2 to 7 cycles can also be inserted using the wait control bits. The settings of these bits also can be used in combination with the #WAIT signal. In this case as well, the #WAIT signal is sampled at the falling edge of the transition of BCLK. However, even when the #WAIT signal is inactive before the wait cycles set by the wait control bits are terminated, the write cycle is not terminated at that time.

Note: The basic write cycle consists of at least two cycles. This does not change regardless of whether zero or one wait cycle is set by the wait control bits. If the number of wait cycles set is 2 or more, the bus cycle is actually extended. In this case, the bus write cycle consists of [number of wait cycles + 1], as in the case of read cycles (providing that there is no external wait).

7.5.3 Burst ROM Read Cycles

Burst read cycle

Example: When 4-consecutive-burst and 2-wait cycles are set during the first access

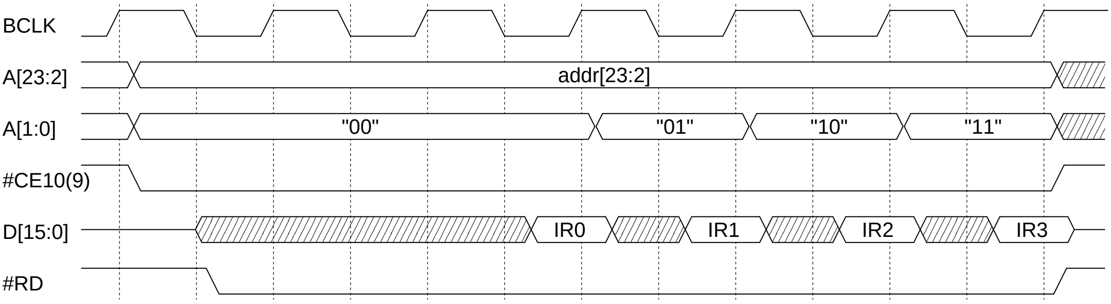


Figure 7.5.5 Burst Read Cycle

A burst read cycle occurs when area 10 or 9 is set for burst ROM and one of those areas is accessed for the following reasons:

1) Instruction fetch

The burst read cycle is executed as long as a instruction fetch from contiguous addresses continues until

A[2:1] = "11" (for 4-consecutive bursts); or
A[3:1] = "111" (for 8-consecutive bursts)

2) Word (32-bit) data readout

Note: A 16-bit output is supported for the burst ROM. Set the device size to 16 bits.

Wait cycles during burst read

In the first bus operation, 0 to 7 wait cycles can be inserted using the wait control bits A10WT[2:0] (D[2:0]) / Areas 10–9 set-up register (0x48126) in the same way as for ordinary SRAM. For the wait cycles to be inserted in the burst cycle that follows, use a dedicated wait control bits, A10BW[1:0], which is only used for reading bursts. The wait cycles can be set in the range from 0 to 3 using these bits.

Note that no wait cycle via the #WAIT pin can be inserted into the burst-read cycle.

Write cycle to burst ROM area

If area 10 or 9 is set for burst ROM, a SRAM write cycle is executed when a write to that area is attempted. In this case, wait cycles via the #WAIT pin can be inserted.

7.6 DRAM Direct Interface

7.6.1 Outline of DRAM Interface

The E0C33A104 incorporates a DRAM direct interface that allows DRAM to be connected directly to areas 8 and 7 or areas 14 and 13. This interface support the 2CAS interface method, so that column addresses can be set at between 8 and 11 bits. In addition, this interface supports a fast-page or an EDO-page mode (EDO DRAM directly connectable to areas) as well as random cycles. The refresh method (CAS-before-RAS refresh or self-refresh) and timing conditions (e.g., number of RAS/CAS cycles and number of precharge cycles) can be programmed using a control bit.

When selecting areas 8 and 7 or areas 14 and 13 to be used for DRAM, it depends on chip-enable settings using CEFUNC (D9) / DRAM timing set-up register (0x48130).
CEFUNC = "0": DRAM can be connected to areas 8 and 7 (default)
#CE8 and #CE7 function as #RAS0 and #RAS1, respectively.
CEFUNC = "1": DRAM can be connected to areas 14 and 13.
#CE14 and #CE13 function as #RAS2 and #RAS3, respectively.

Figure 7.6.1 shows a sample DRAM connection. Table 7.6.1 and Table 7.6.2 show examples of connectable DRAMs and typical configurations.

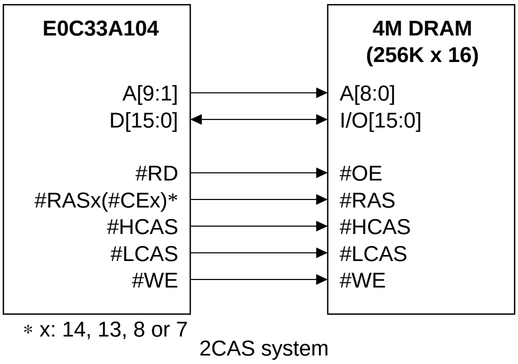


Figure 7.6.1 Sample DRAM Connection

Table 7.6.1 Connectable DRAM Example

DRAM	Number of devices	Number of Row bits	Number of Column bits	Memory size
1M (64K x 16)	1	8	8	128K bytes
4M (256K x 16)	1	9	9	512K bytes
16M (1M x 16)	1	12	8	2M bytes

Table 7.6.2 DRAM Configuration Example (areas 7 and 8 only)

	Area 7	Area 8	Total memory size	
1	I/O	DRAM (1M)	1M bits	(128K bytes)
2	I/O	DRAM (4M)	4M bits	(512K bytes)
3	I/O	DRAM (16M)	16M bits	(2M bytes)
4	DRAM (1M)	DRAM (1M)	2M bits	(256K bytes)
5	DRAM (4M)	DRAM (4M)	8M bits	(1M bytes)
6	DRAM (16M)	DRAM (16M)	32M bits	(4M bytes)

7.6.2 DRAM Setting Conditions

The DRAM interface allows the following conditions to be selected. Although DRAM can be used in areas 8 and 7 or areas 14 and 13, these condition are applied to all four areas and cannot be set individually for each area.

Table 7.6.3 DRAM Interface Parameters

Parameter	Selectable condition	Initial setting	Control bits
Page mode	EDO page mode or Fast page mode	Fast page mode	REDO(DC)/Bus control register(0x4812E)
RAS mode	Successive RAS mode or Normal mode	Normal mode	CRAS(D8)/DRAM timing set-up register(0x48130)
Column address size	8, 9, 10 or 11 bits	8 bits	RCA[1:0](D[B:A])/Bus control register(0x4812E)
Refresh enable	Enabled or Disabled	Disabled	RPC2(D9)/Bus control register(0x4812E)
Refresh method	Self-refresh or CBR refresh	CBR refresh	RPC1(D8)/Bus control register(0x4812E)
Refresh RPC delay	2.0 or 1.0	1.0	RPC0(D7)/Bus control register(0x4812E)
Refresh RAS pulse width	2, 3, 4 or 5 cycles	2 cycles	RRA[1:0](D[6:5])/Bus control register(0x4812E)
Number of RAS precharge cycles	1, 2, 3 or 4 cycles	1 cycle	RPRC[1:0](D[7:6])/DRAM timing set-up register(0x48130)
CAS cycle control	1, 2, 3 or 4 cycles	1 cycle	CASC[1:0](D[4:3])/DRAM timing set-up register(0x48130)
RAS cycle control	1, 2, 3 or 4 cycles	1 cycle	RASC[1:0](D[1:0])/DRAM timing set-up register(0x48130)

Page mode

The DRAM interface allows EDO DRAM to be connected directly. Therefore, the EDO-page mode is supported along with the fast-page mode.

Use REDO to choose the desired page mode that suits the DRAM to be used.

REDO = "1": EDO page mode
REDO = "0": Fast page mode (default)

Successive RAS mode

For applications that require high-speed DRAM access, the DRAM interface supports a successive RAS mode. In this mode, even when successive accesses to the DRAM are not requested by the CPU or DMA, the #RAS signal is kept low and operation is continued without inserting any precharge cycle. Therefore, when accessing the same page (row address) of the DRAM that has been accessed previously, the page mode remains active, allowing read/write to be performed at high speeds.

However, to maintain the rated AC characteristics, one idle cycle is inserted when access in the page mode is begun and when finished.

CRAS is used to set the successive RAS mode.

CRAS = "1": Successive RAS mode
CRAS = "0": Normal mode (default)

The successive RAS mode is suspended by one of the following causes:

- a refresh cycle has occurred;
- bus control is requested by an external bus master;
- the requested device and page are not compatible with DRAM memory; and
- the slp or halt instruction is executed.

If the successive RAS mode is suspended, a precharge cycle is inserted before the next bus cycle begins.

Note: When using the successive RAS mode, always be sure to use #DRD for the read signal and #DWE for the low-byte write signal.

Column address size

When accessing DRAM, addresses are divided into a row address and a column address as they are output. Choose the size of this column address using RCA, as shown below.

Table 7.6.4 Column Address Size

RCA1	RCA0	Column address size
1	1	11
1	0	10
0	1	9
0	0	8

The initial default size is 8 bits. Choose the desired size according to the address input pins of the DRAM to be used.

The row addresses output synchronously with falling edges of the #RAS signal are derived from the CPU's internal 28-bit addresses by logically shifting them to the right by an amount equal to the column address size. The MSB contains a 1. The column addresses are output to the address bus along with the falling edges of the #HCAS/#LCAS signal. These addresses are derived directly from the CPU's internal 28-bit addresses.

Figure 7.6.2 shows the contents of the row addresses thus output.

28-bit CPU internal address

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

(1) Row address when column address is set to 8 bits

T	T	T	T	T	T	T	T	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8

(2) Row address when column address is set to 9 bits

T	T	T	T	T	T	T	T	T	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9

(3) Row address when column address is set to 10 bits

T	T	T	T	T	T	T	T	T	T	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10

(4) Row address when column address is set to 11 bits

T	T	T	T	T	T	T	T	T	T	T	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11

T = "1", 0–27: Bit number of CPU internal address

Figure 7.6.2 Example of Row/Column Address Mapping

Refresh enable

Use RPC2 to enable or disable the internal refresh function.

RPC2 = "1": Enabled

RPC2 = "0": Disabled (default)

After choosing the desired refresh method using PPC1, write "1" to PPC2.

Refresh method

The DRAM interface supports both a CAS-before-RAS refresh cycle and a self-refresh cycle. Choose the desired method using RPC1.

RPC1 = "1": Self-refresh

RPC1 = "0": CAS-before-RAS refresh

The generation interval of the CAS-before-RAS refresh is determined by the underflow signal of an 8-bit programmable timer 0. Consequently, before the CAS-before-RAS refresh can be executed, the 8-bit programmable timer 0 must be set to obtain the necessary underflow timing. When this method is selected and RPC2 is enabled, the refresh cycle is generated each time the 8-bit programmable timer 0 underflows.

The self-refresh is started by writing "1" to RPC2 while RPC1 = "1" and is terminated by clearing RPC1 or RPC2 to "0".

If PPC1 is switched over when PPC2 = "1" (refresh enabled), an undesirable self-refresh cycle is generated. So be sure to clear PPC2 to "0" (refresh disabled) before selecting the refresh method.

Refresh RPC delay

Use RPC0 to set the RPC delay value of a refresh cycle (a delay time from the immediately preceding precharge to the fall of #CAS).

RPC0 = "1": 2 cycles
RPC0 = "0": 1 cycle

Refresh RAS pulse width

Use RRA to set the #RAS pulse width of a CAS-before-RAS refresh cycle.

Table 7.6.5 Refresh RAS Pulse Width

RRA1	RRA0	Pulse width
1	1	5 cycles
1	0	4 cycles
0	1	3 cycles
0	0	2 cycles

The initial default value is 2 cycles.

Number of RAS precharge cycles

Use RPRC to choose the number of RAS precharge cycles.

Table 7.6.6 Number of RAS Precharge Cycles

RPRC1	RPRC0	Number of cycles
1	1	4 cycles
1	0	3 cycles
0	1	2 cycles
0	0	1 cycle

The initial default value is 1 cycle.

CAS cycle control

Use CASC to choose the number of CAS cycles when accessing DRAM.

Table 7.6.7 Number of CAS Cycles

CASC1	CASC0	Number of cycles
1	1	4 cycles
1	0	3 cycles
0	1	2 cycles
0	0	1 cycle

The initial default value is 1 cycle.

RAS cycle control

Use RASC to choose the number of RAS cycles when accessing DRAM.

Table 7.6.8 Number of RAS Cycles

RASC1	RASC0	Number of cycles
1	1	4 cycles
1	0	3 cycles
0	1	2 cycles
0	0	1 cycle

The initial default value is 1 cycle.

Note: By writing a byte to the BCU register (address 0x0048130), the data will be written to both 0x0048130 and 0x0048131 addresses. When writing a byte to address 0x0048131, the register's data will not be modified.
Write in half-word or word units to address 0x0048130.

7.6.3 DRAM Read/Write Cycles

The following shows the basic bus cycles of DRAM. For detailed timings, refer to Section 18.7, "AC Characteristics". For examples of timings with each type of DRAM, refer to the Appendix.

The DRAM interface does not accept wait cycles inserted via the #WAIT pin.

DRAM random read cycle

Example: RAS: 1 cycle; CAS: 2 cycles; Precharge: 1 cycle

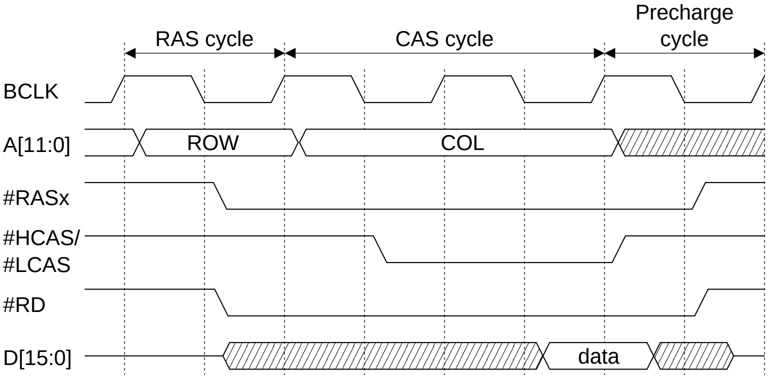


Figure 7.6.3 DRAM Random Read Cycle

DRAM read cycle (fast page mode)

Example: RAS: 1 cycle; CAS: 2 cycles; Precharge: 1 cycle

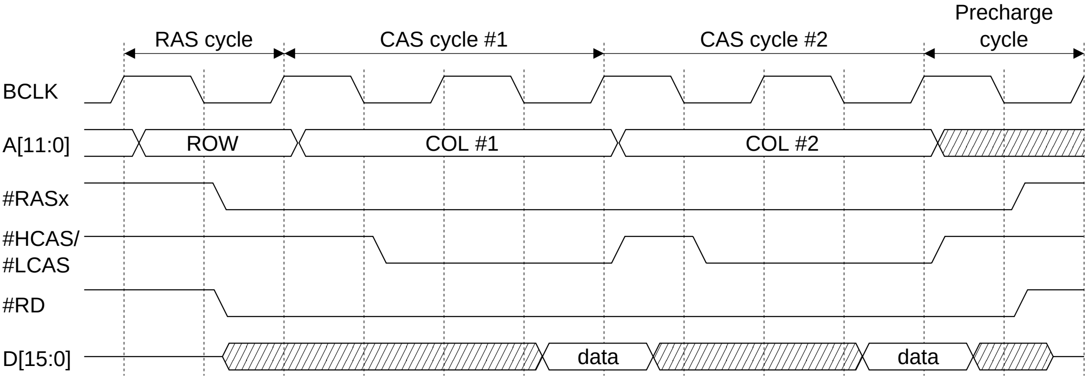


Figure 7.6.4 DRAM Read Cycle (fast page mode)

DRAM read cycle (EDO page mode)

Example: RAS: 1 cycle; CAS: 2 cycles; Precharge: 1 cycle

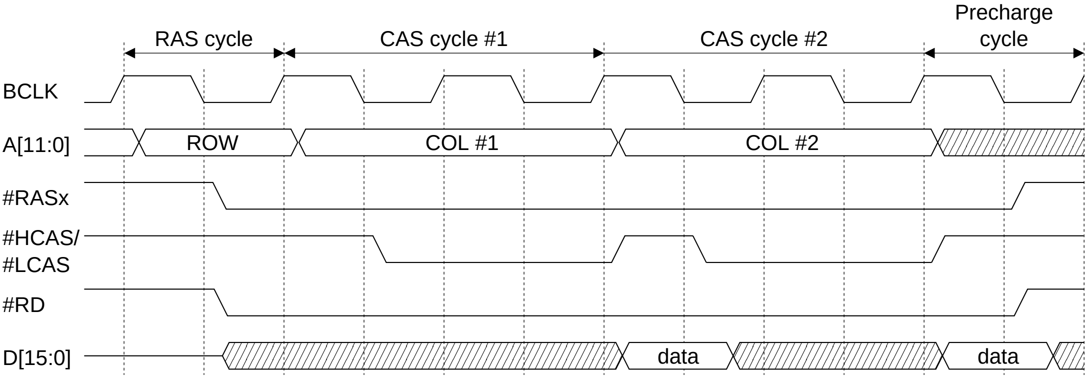


Figure 7.6.5 DRAM Read Cycle (EDO page mode)

The read timing in EDO page-mode lags 0.5 cycles behind that in fast page mode.

DRAM random write cycle

Example: RAS: 1 cycle; CAS: 2 cycles; Precharge: 1 cycle

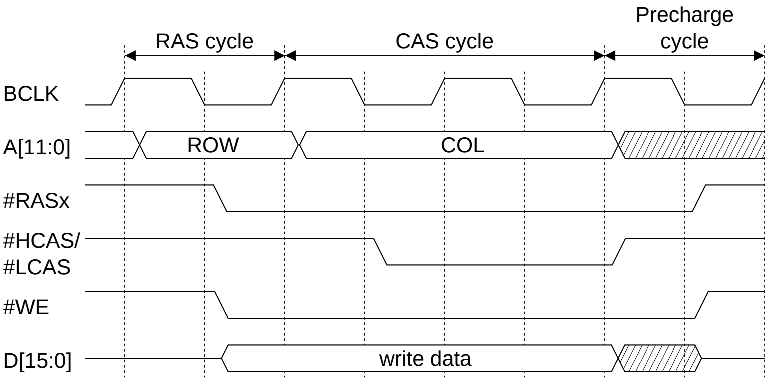


Figure 7.6.6 DRAM Random Write Cycle

DRAM write cycle (fast page or EDO page mode)

Example: RAS: 1 cycle; CAS: 2 cycles; Precharge: 1 cycle

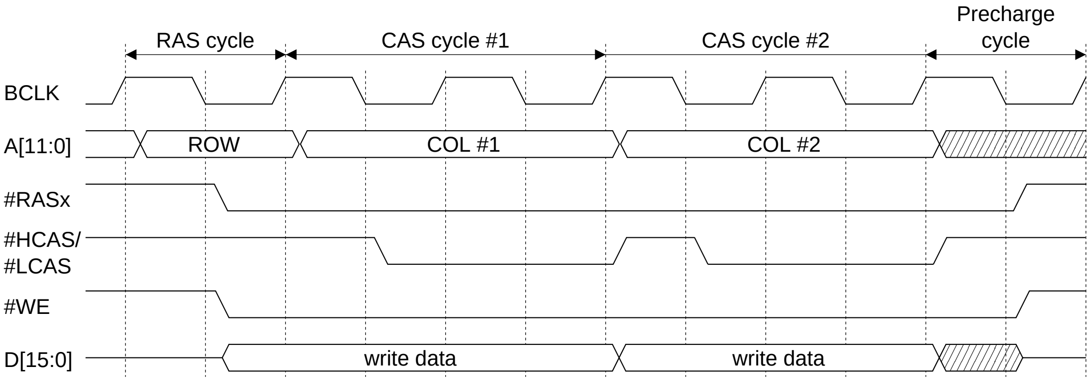


Figure 7.6.7 DRAM Write Cycle (fast page or EDO page mode)

Operation in successive RAS mode

Example: RAS: 2 cycles; CAS: 1 cycle; Precharge: 2 cycles

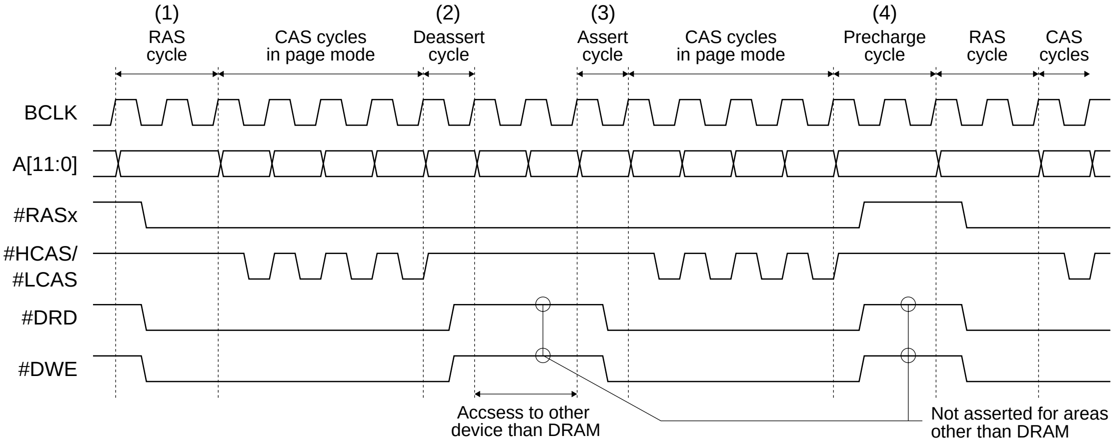


Figure 7.6.8 Operation in Successive RAS Mode

(1) When accessing the DRAM area, an ordinary RAS cycle is executed first.

(2) If access to the same DRAM is suspended during a page mode, #RASx remains asserted while some other device is accessed. In this case, a cycle to temporarily deassert #DRD/#DWE is inserted before accessing the other device.

(3) If access to the same page in the same DRAM area as in (1) is requested after (2), #DRD/#DWE is asserted back again to restart the page mode.

(4) A precharge cycle is executed when one of the following conditions that cause the page mode to suspend is encountered:
- access to different DRAM is requested;
- access to a different page in the same DRAM area is requested;
- access to some other device than DRAM is requested;
- CAS-before-RAS refresh is requested; and
- relinquishing of bus control is requested by an external bus master.

Note: When using the successive RAS mode, always be sure to use #DRD for the read signal and #DWE for the low-byte write signal.

7.6.4 DRAM Refresh Cycles

The DRAM interface supports a CAS-before-RAS refresh cycle and a self-refresh cycle.

CAS-before-RAS refresh cycle

Before performing a CAS-before-RAS refresh, set RPC2 to "1" while RPC1 = "0" in order to enable the DRAM refresh function. Once this is done, the BCU executes a CAS-before-RAS refresh by using the underflow signal that is output by the 8-bit programmable timer 0 as a trigger. Therefore, refresh generation timing can be programmed using the internal prescaler and 8-bit programmable timer 0.

For details on how to control the prescaler and 8-bit programmable timer 0, refer to Section 10.2, "Prescaler and Operating Clock for Peripheral Circuits", and Section 11.3, "8-Bit Programmable Timers".

Example: RPC delay: 1 cycle; Refresh RAS pulse width: 2 cycles; Precharge: 1 cycle

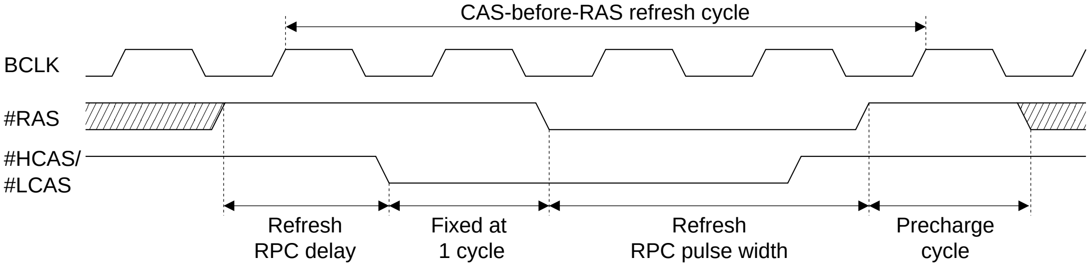


Figure 7.6.9 CAS-Before-RAS Refresh

When the refresh cycle is terminated, the #HCAS/#LCAS signal boot timing is 0.5 cycles before that of #RAS. Consequently, the pulse width of #HCAS/#LCAS is determined by the refresh RAS pulse width that was set using RRA. The number of precharge cycles after the refresh cycle is defined by the value that was set using RPRC, the same value that is used for both random cycles and page mode accesses.

Self-refresh

To support DRAM chips equipped with a self-refresh function, the E0C33A104 has a function to generate a self-refresh cycle.

To start a self-refresh cycle, set RPC2 to "1" after setting RPC1 to "1". To deactivate a self-refresh cycle, write "0" to RPC1 or RPC2.

Example: RPC delay: 1 cycle

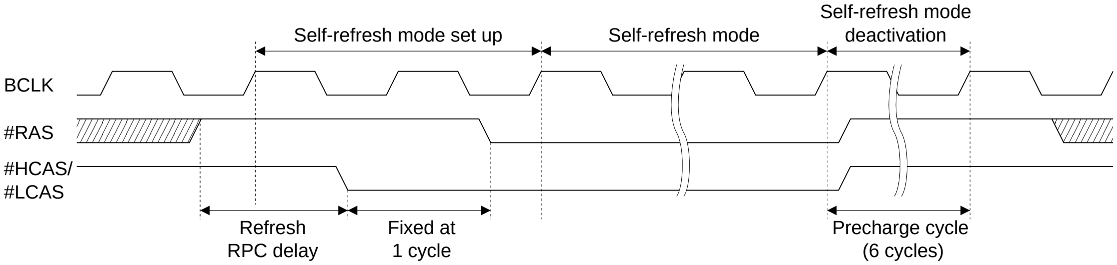


Figure 7.6.10 Self-Refresh

For a self-refresh function as well, the RPC delay is determined by setting RPC0 in the same way as for a CAS-before-RAS refresh.

The refresh RAS pulse width is determined by the timing at which the refresh is deactivated in software and is unaffected by settings of RRA.

#RAS and #HCAS/#LCAS are booted up simultaneously upon completion of a self-refresh and the precharge duration that follows is fixed at 6 cycles.

Normally, DRAM specifications require that the contents of all row addresses be refreshed within a certain time before and after a self-refresh. To meet this requirement, make sure a CAS-before-RAS refresh is executed by a program. In this case, set the 8-bit programmable timer 0 so that the contents of all row addresses are refreshed within a predetermined time.

Note: If read from or write to the DRAM under a self-refresh is attempted, the BCU keeps #RAS and #HCAS/#LCAS low as it executes a read/write cycle. Other bus signals than #RAS and #HCAS/#LCAS (e.g., address, data, and control signals) change their state according to the specified conditions. Since said attempt initiates an invalid access to the DRAM, do not read from or write to the DRAM during a self-refresh.

7.7 Releasing External Bus

The external bus is normally controlled by the CPU, but the E0C33A104 is designed to release control of the bus ownership to an external device. This function is enabled by writing "1" to SEMAS (D2) / Bus control register (0x4812E) (disabled by default). The #BUSREQ and #BUSACK pins are used for control of the bus ownership.

Sequence in which control of the bus is released

This sequence is described below.

1. The external bus master device requesting control of the bus ownership lowers the #BUSREQ pin.
2. The CPU keeps monitoring the status of the #BUSREQ pin, so that when this pin is lower, the CPU terminates the bus cycle being executed and places the signals listed below in high-impedance state one cycle later:
   A[23:0], D[15:0], #RD, #WRL, #WRH, #HCAS, #LCAS, #CExx
   Then the CPU lowers the #BUSACK pin to inform the external device that control of the bus ownership has been released.
3. One cycle later, the external bus starts its own bus cycle. The external bus master must hold the #BUSREQ pin low until the bus cycle is completed.
4. After completing the necessary bus cycles, the external bus master places the bus in high-impedance state and releases the #BUSREQ pin back high.
5. After confirming that the #BUSREQ pin is raised again, the CPU raises the #BUSACK pin one cycle later and resumes the processing that has been suspended.

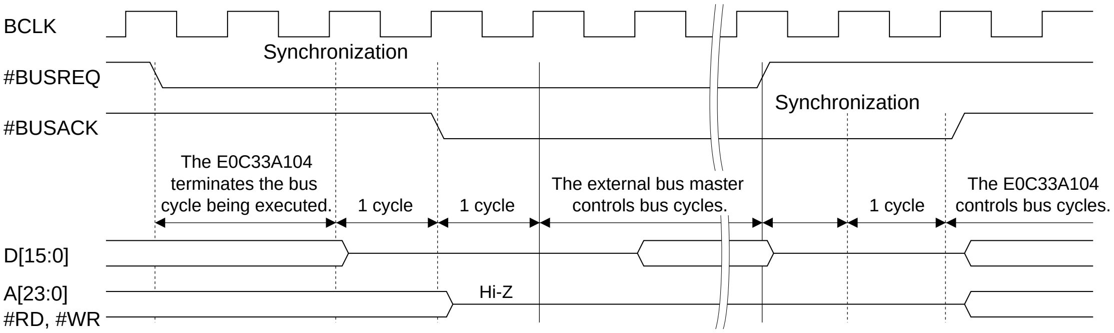

BCLK
Synchronization
#BUSREQ
#BUSACK
Synchronization
The E0C33A104 terminates the bus cycle being executed.
1 cycle
1 cycle
The external bus master controls bus cycles.
1 cycle
The E0C33A104 controls bus cycles.
D[15:0]
A[23:0]
#RD, #WR
Hi-Z

Figure 7.7.1 External Bus Release Timing

If control of the bus ownership is requested during a DMA transfer by the internal DMA controller, the DMA transfer under way is suspended at a break in data to accept the request for bus ownership control. The DMA transfer that has been kept pending is restarted when the CPU gains control of the bus ownership.

DRAM refresh when bus ownership control is released

In systems where DRAM is connected directly, a refresh request could arise while control of the bus ownership is released from the CPU. In such a case, take one of the corrective measures described below.

- Monitoring the output signal of the 8-bit programmable timer 0

The underflow signal (DRAM refresh request) of the 8-bit programmable timer 0 can be output from the P10 I/O port pin.
If a refresh request arises while the external bus master is monitoring this output, release #BUSREQ back high to drop the request for bus ownership control.
Start a DRAM refresh cycle when control of the bus ownership is returned to the CPU.

To direct the P10 pin in order to output the underflow signal of the 8-bit programmable timer 0, write "1" to CFP10 (D0) / P1 function select register (0x402D4 [Byte]) and IOC10 (D0) / P1 I/O control register (0x402D6 [Byte]). Also, to output the underflow signal to an external device, write "1" to PTOUT0 (D2) / 8-bit Timer 0 control register (0x40160 [Byte]). For details about output control, refer to Section 11.3, "8-Bit Programmable Timers".

- Monitoring the #BUSGET signal

The #BUSGET signal can be output from the P13 I/O port pin.
The #BUSGET signal is derived from logical sum of the following signals:
1. DRAM refresh request signal (output from the 8-bit programmable timer 0)
2. Interrupt request signal from the interrupt controller to the CPU
3. Startup request signal from the interrupt controller to the IDMA

If the #BUSGET signal is found to be active when the external bus master is monitoring it, release #BUSREQ back high to drop the request for bus ownership control.
When using the #BUSGET signal to only monitor a refresh request, set the interrupt controller in such a way that no interrupt request or IDMA startup request will be generated.

To direct the P13 pin for output of the #BUSGET signal, write "1" to CFP13 (D3) / P1 function select register (0x402D4 [Byte]) and IOC13 (D3) / P1 I/O control register (0x402D6 [Byte]).

7.8 Power-down Control by External Device

In addition to requesting the releasing of bus ownership control described above, it is possible to place the CPU in a HALT state by using the #BUSREQ signal. This allows the CPU to be stopped during bus operation by an external bus master in order to conserve power.
This function is enabled by writing "1" to SEPD (D1) / Bus control register (0x4812E).
If SEPD = "1", the CPU and the BCU stop operating when the #BUSREQ pin is lowered, thus entering a HALT state. This HALT state is not cleared by an interrupt from the internal peripheral circuits and remains set until the #BUSREQ pin is released back high. Unlike in the case of ordinary releasing of the bus by #BUSREQ, the address bus and bus control signals are not placed in high-impedance state.

For a DRAM refresh request that may arise in this HALT state, take one of the corrective measures described above.

7.9 I/O Memory of External System Interface

Table 7.9.1 shows the control bits of the external system interface. These I/O memories are mapped into the area (0x48000 and following addresses) used for the internal 16-bit peripheral circuits. However, these I/O memories can be accessed in bytes or words, as well as in half-words.

For the control bits of the external system interface pins assigned to the output and I/O ports, and for details on how to control the 8-bit programmable timer 0 in order to generate a DRAM refresh cycle, refer to each corresponding section in this manual.

Table 7.9.1 Control Bits of External System Interface

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Areas 18–15 set-up register	**0048120 (HW)**	DF	–	reserved	–				–	–	0 when being read.
		DE	**A18SZ**	Areas 18–17 device size selection	1	8 bits	0	16 bits	0	R/W	
		DD	**A18DF1**	Areas 18–17	A18DF[1:0]			Number of cycles	1	R/W	
		DC	**A18DF0**	output disable delay time	1	1		3.5	1		
					1	0		2.5			
					0	1		1.5			
					0	0		0.5			
		DB	–	reserved	–				–	–	0 when being read.
		DA	**A18WT2**	Areas 18–17 wait control	A18WT[2:0]			Wait cycles	1	R/W	
		D9	**A18WT1**		1	1	1	7	1		
		D8	**A18WT0**		1	1	0	6	1		
					1	0	1	5			
					1	0	0	4			
					0	1	1	3			
					0	1	0	2			
					0	0	1	1			
					0	0	0	0			
		D7	–	reserved	–				–	–	0 when being read.
		D6	**A16SZ**	Areas 16–15 device size selection	1	8 bits	0	16 bits	0	R/W	
		D5	**A16DF1**	Areas 16–15	A16DF[1:0]			Number of cycles	1	R/W	
		D4	**A16DF0**	output disable delay time	1	1		3.5	1		
					1	0		2.5			
					0	1		1.5			
					0	0		0.5			
		D3	–	reserved	–				–	–	0 when being read.
		D2	**A16WT2**	Areas 16–15 wait control	A16WT[2:0]			Wait cycles	1	R/W	
		D1	**A16WT1**		1	1	1	7	1		
		D0	**A16WT0**		1	1	0	6	1		
					1	0	1	5			
					1	0	0	4			
					0	1	1	3			
					0	1	0	2			
					0	0	1	1			
					0	0	0	0			
Areas 14–13 set-up register	**0048122 (HW)**	DF–9	–	reserved	–				–	–	0 when being read.
		D8	**A14DRA**	Area 14 DRAM selection	1	Used	0	Not used	0	R/W	
		D7	**A13DRA**	Area 13 DRAM selection	1	Used	0	Not used	0	R/W	
		D6	**A14SZ**	Areas 14–13 device size selection	1	8 bits	0	16 bits	0	R/W	
		D5	**A14DF1**	Areas 14–13	A14DF[1:0]			Number of cycles	1	R/W	
		D4	**A14DF0**	output disable delay time	1	1		3.5	1		
					1	0		2.5			
					0	1		1.5			
					0	0		0.5			
		D3	–	reserved	–				–	–	0 when being read.
		D2	**A14WT2**	Areas 14–13 wait control	A14WT[2:0]			Wait cycles	1	R/W	
		D1	**A14WT1**		1	1	1	7	1		
		D0	**A14WT0**		1	1	0	6	1		
					1	0	1	5			
					1	0	0	4			
					0	1	1	3			
					0	1	0	2			
					0	0	1	1			
					0	0	0	0			

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Areas 12–11	**0048124**	DF–7	**–**	reserved	–				–	–	0 when being read.
set-up register	**(HW)**	D6	**A12SZ**	Areas 12–11 device size selection	1	8 bits	0	16 bits	0	R/W	
		D5	**A12DF1**	Areas 12–11	A18DF[1:0]		Number of cycles		1	R/W	
		D4	**A12DF0**	output disable delay time	1	1	3.5		1		
					1	0	2.5				
					0	1	1.5				
					0	0	0.5				
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**A12WT2**	Areas 12–11 wait control	A18WT[2:0]			Wait cycles	1	R/W	
		D1	**A12WT1**		1	1	1	7	1		
		D0	**A12WT0**		1	1	0	6	1		
					1	0	1	5			
					1	0	0	4			
					0	1	1	3			
					0	1	0	2			
					0	0	1	1			
					0	0	0	0			
Areas 10–9	**0048126**	DF–B	**–**	reserved	–				–	–	0 when being read.
set-up register	**(HW)**	DA	**A10BW1**	Areas 10–9	A10BW[1:0]		Wait cycles		0	R/W	
		D9	**A10BW0**	burst ROM	1	1	3		0		
				burst read cycle wait control	1	0	2				
					0	1	1				
					0	0	0				
		D8	**A10DRA**	Area 10 burst ROM selection	1	Used	0	Not used	0	R/W	
		D7	**A9DRA**	Area 9 burst ROM selection	1	Used	0	Not used	0	R/W	
		D6	**A10SZ**	Areas 10–9 device size selection	1	8 bits	0	16 bits	0	R/W	
		D5	**A10DF1**	Areas 10–9	A10DF[1:0]		Number of cycles		1	R/W	
		D4	**A10DF0**	output disable delay time	1	1	3.5		1		
					1	0	2.5				
					0	1	1.5				
					0	0	0.5				
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**A10WT2**	Areas 10–9 wait control	A10WT[2:0]			Wait cycles	1	R/W	
		D1	**A10WT1**		1	1	1	7	1		
		D0	**A10WT0**		1	1	0	6	1		
					1	0	1	5			
					1	0	0	4			
					0	1	1	3			
					0	1	0	2			
					0	0	1	1			
					0	0	0	0			
Areas 8–7	**0048128**	DF–9	**–**	reserved	–				–	–	0 when being read.
set-up register	**(HW)**	D8	**A8DRA**	Area 8 DRAM selection	1	Used	0	Not used	0	R/W	
		D7	**A7DRA**	Area 7 DRAM selection	1	Used	0	Not used	0	R/W	
		D6	**A8SZ**	Areas 8–7 device size selection	1	8 bits	0	16 bits	0	R/W	
		D5	**A8DF1**	Areas 8–7	A8DF[1:0]		Number of cycles		1	R/W	
		D4	**A8DF0**	output disable delay time	1	1	3.5		1		
					1	0	2.5				
					0	1	1.5				
					0	0	0.5				
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**A8WT2**	Areas 8–7 wait control	A8WT[2:0]			Wait cycles	1	R/W	
		D1	**A8WT1**		1	1	1	7	1		
		D0	**A8WT0**		1	1	0	6	1		
					1	0	1	5			
					1	0	0	4			
					0	1	1	3			
					0	1	0	2			
					0	0	1	1			
					0	0	0	0			

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
Areas 6–4 set-up register	**004812A (HW)**	DF–E	**–**	reserved	–		–	–	0 when being read.
		DD	**A6DF1**	Area 6	A6DF[1:0]	Number of cycles	1	R/W	
		DC	**A6DF0**	output disable delay time	1 1	3.5	1		
					1 0	2.5			
					0 1	1.5			
					0 0	0.5			
		DB	**–**	reserved	–		–	–	0 when being read.
		DA	**A6WT2**	Area 6 wait control	A6WT[2:0]	Wait cycles	1	R/W	
		D9	**A6WT1**		1 1 1	7	1		
		D8	**A6WT0**		1 1 0	6	1		
					1 0 1	5			
					1 0 0	4			
					0 1 1	3			
					0 1 0	2			
					0 0 1	1			
					0 0 0	0			
		D7	**–**	reserved	–		–	–	0 when being read.
		D6	**A5SZ**	Areas 5–4 device size selection	1 8 bits	0 16 bits	0	R/W	
		D5	**A5DF1**	Areas 5–4	A5DF[1:0]	Number of cycles	1	R/W	
		D4	**A5DF0**	output disable delay time	1 1	3.5	1		
					1 0	2.5			
					0 1	1.5			
					0 0	0.5			
		D3	**–**	reserved			–	–	0 when being read.
		D2	**A5WT2**	Areas 5–4 wait control	A5WT[2:0]	Wait cycles	1	R/W	
		D1	**A5WT1**		1 1 1	7	1		
		D0	**A5WT0**		1 1 0	6	1		
					1 0 1	5			
					1 0 0	4			
					0 1 1	3			
					0 1 0	2			
					0 0 1	1			
					0 0 0	0			
Bus control register	**004812E (HW)**	DF	**RBCLK**	BCLK output control	1 Fixed at H	0 Enabled	0	R/W	
		DE	**–**	reserved	–		0	–	Writing 1 not allowed.
		DD	**RBST8**	Burst ROM burst mode selection	1 8-successive	0 4-successive	0	R/W	
		DC	**REDO**	DRAM page mode selection	1 EDO	0 Fast page	0	R/W	
		DB	**RCA1**	Column address size selection	RCA[1:0]	Size	0	R/W	
		DA	**RCA0**		1 1	11	0		
					1 0	10			
					0 1	9			
					0 0	8			
		D9	**RPC2**	Refresh enable	1 Enabled	0 Disabled	0	R/W	
		D8	**RPC1**	Refresh method selection	1 Self-refresh	0 CBR-refresh	0	R/W	
		D7	**RPC0**	Refresh RPC delay setup	1 2.0	0 1.0	0	R/W	
		D6	**RRA1**	Refresh RAS pulse width	RRA[1:0]	Number of cycles	0	R/W	
		D5	**RRA0**	selection	1 1	5	0		
					1 0	4			
					0 1	3			
					0 0	2			
		D4	**–**	reserved	–		0	–	Writing 1 not allowed.
		D3	**SBUSST**	External interface method selection	1 #BSL	0 A0	0	R/W	
		D2	**SEMAS**	External bus master setup	1 Existing	0 Nonexistent	0	R/W	
		D1	**SEPD**	External power-down control	1 Enabled	0 Disabled	0	R/W	
		D0	**SWAITE**	#WAIT enable	1 Enabled	0 Disabled	0	R/W	

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
DRAM timing set-up register	**0048130 (HW)**	DF–B	**–**	reserved	–		–	–	0 when being read.
		DA	**–**	reserved	–		0	–	Writing 1 not allowed.
		D9	**CEFUNC**	#CE pin function selection	1 High address	0 Normal	0	R/W	
		D8	**CRAS**	Successive RAS mode setup	1 Successive	0 Normal	0	R/W	
		D7	**RPRC1**	DRAM	RPRC[1:0]	Number of cycles	0	R/W	
		D6	**RPRC0**	RAS precharge cycles selection	1 1	4	0		
					1 0	3			
					0 1	2			
					0 0	1			
		D5	–	reserved	–		–	–	0 when being read.
		D4	**CASC1**	DRAM	CASC[1:0]	Number of cycles	0	R/W	
		D3	**CASC0**	CAS cycles selection	1 1	4	0		
					1 0	3			
					0 1	2			
					0 0	1			
		D2	–	reserved	–		–	–	0 when being read.
		D1	**RASC1**	DRAM	RASC[1:0]	Number of cycles	0	R/W	
		D0	**RASC0**	RAS cycles selection	1 1	4	0		
					1 0	3			
					0 1	2			
					0 0	1			

A18SZ: Areas 18–17 device size selection (DE) / Areas 18–15 set-up register (0x48120)
A16SZ: Areas 16–15 device size selection (D6) / Areas 18–15 set-up register (0x48120)
A14SZ: Areas 14–13 device size selection (D6) / Areas 14–13 set-up register (0x48122)
A12SZ: Areas 12–11 device size selection (D6) / Areas 12–11 set-up register (0x48124)
A10SZ: Areas 10–9 device size selection (D6) / Areas 10–9 set-up register (0x48126)
A8SZ: Areas 8–7 device size selection (D6) / Areas 8–7 set-up register (0x48128)
A5SZ: Areas 5–4 device size selection (D6) / Areas 6–4 set-up register (0x4812A)

Select the size of the device connected to each area.

Write "1":	8 bits
Write "0":	16 bits
Read:	Valid

A device size can be selected for every two areas.
An 8-bit size is selected by writing "1" to AxxSZ and a 16-bit size is selected by writing "0" to AxxSZ.
Area 6 has its first half (0x300000 through 0x37FFFF) fixed to an 8-bit device and the last half (0x380000 through 0x3FFFFF) fixed to a 16-bit device.
At cold start, these bits are set to "0" (16 bits). At hot start, these bits retain their status before being initialized.

A18DF1–A18DF0: Areas 18–17 output disable delay time (D[D:C]) / Areas 18–15 set-up register (0x48120)
A16DF1–A16DF0: Areas 16–15 output disable delay time (D[5:4]) / Areas 18–15 set-up register (0x48120)
A14DF1–A14DF0: Areas 14–13 output disable delay time (D[5:4]) / Areas 14–13 set-up register (0x48122)
A12DF1–A12DF0: Areas 12–11 output disable delay time (D[5:4]) / Areas 12–11 set-up register (0x48124)
A10DF1–A10DF0: Areas 10–9 output disable delay time (D[5:4]) / Areas 10–9 set-up register (0x48126)
A8DF1–A8DF0: Areas 8–7 output disable delay time (D[5:4]) / Areas 8–7 set-up register (0x48128)
A6DF1–A6DF0: Area 6 output disable delay time (D[D:C]) / Areas 6–4 set-up register (0x4812A)
A5DF1–A5DF0: Areas 5–4 output disable delay time (D[5:4]) / Areas 6–4 set-up register (0x4812A)

Set the output-disable delay time.

Table 7.9.2 Output Disable Delay Time

AxxDF1	AxxDF0	Delay time
1	1	3.5 cycles
1	0	2.5 cycles
0	1	1.5 cycles
0	0	0.5 cycles

When using a device that has a long output-disable time, set a delay time to ensure that no contention for the data bus occurs during the bus operation immediately after a device is read.
At cold start, these bits are set to "11" (3.5 cycles). At hot start, the bits retain their status before being initialized.

A18WT2–A18WT0: Areas 18–17 wait control (D[A:8]) / Areas 18–15 set-up register (0x48120)
A16WT2–A16WT0: Areas 16–15 wait control (D[2:0]) / Areas 18–15 set-up register (0x48120)
A14WT2–A14WT0: Areas 14–13 wait control (D[2:0]) / Areas 14–13 set-up register (0x48122)
A12WT2–A12WT0: Areas 12–11 wait control (D[2:0]) / Areas 12–11 set-up register (0x48124)
A10WT2–A10WT0: Areas 10–9 wait control (D[2:0]) / Areas 10–9 set-up register (0x48126)
A8WT2–A8WT0: Areas 8–7 wait control (D[2:0]) / Areas 8–7 set-up register (0x48128)
A6WT2–A6WT0: Area 6 wait control (D[A:8]) / Areas 6–4 set-up register (0x4812A)
A5WT2–A5WT0: Areas 5–4 wait control (D[2:0]) / Areas 6–4 set-up register (0x4812A)

Set the number of wait cycles to be inserted when accessing an SRAM device.
The values 0 through 7 written to the control bits equal the number of wait cycles inserted.
Note that the write cycle consists of a minimum of two cycles, so that a writing 0 or 1 is invalid.
When an SRAM device is connected, wait cycles derived via the #WAIT pin can also be inserted. In this case too, the wait cycles set by AxxWT are valid.
The DRAM read/write cycles do not have wait cycles inserted that are set by AxxWT or derived from the #WAIT pin.
The burst read cycle of a burst ROM (except for the first access) also does not have any wait cycle inserted. The first read cycle of a burst ROM and the write cycle to the burst ROM area have wait cycles inserted that are set by AxxWT . Wait cycles derived from the #WAIT pin also can be inserted in the cycle for writing to the burst ROM area.
At cold start, these bits are set to "111" (7 cycles). At hot start, the bits retain their status before being initialized.

A14DRA: Area 14 DRAM selection (D8) / Areas 14–13 set-up register (0x48122)
A13DRA: Area 13 DRAM selection (D7) / Areas 14–13 set-up register (0x48122)
A8DRA: Area 8 DRAM selection (D8) / Areas 8–7 set-up register (0x48128)
A7DRA: Area 7 DRAM selection (D7) / Areas 8–7 set-up register (0x48128)

Select the DRAM direct interface.

Write "1": DRAM is used
Write "0": DRAM is not used
Read: Valid

When DRAM is used by connecting it directly to the E0C33A104, write "1" to this bit. The ordinary SRAM interface is selected by writing "0" to the control bit.
The areas to which DRAM can be connected are areas 8 and 7 when the CEFUNC = "0", or areas 14 and 13 when the bit = "1".
At cold start, these bits are set to "0" (DRAM not used). At hot start, the bits retain their status before being initialized.

A10DRA: Area 10 burst ROM selection (D8) / Areas 10–9 set-up register (0x48126)
A9DRA: Area 9 burst ROM selection (D7) / Areas 10–9 set-up register (0x48126)

Set areas 10 and 9 for use of burst ROM.

Write "1": Burst ROM is used
Write "0": Burst ROM is not used
Read: Valid

When using burst ROM, write "1" to the control bit. The ordinary SRAM interface is selected by writing "0" to the bit.
Area 9 can only be used when the CEFUNC = "0".
At cold start, these bits are set to "0" (burst ROM not used). At hot start, the bits retain their status before being initialized.

A10BW1–A10BW0: Burst read cycle wait control (D[A:9]) / Areas 10–9 set-up register (0x48126)

Set the number of wait cycles inserted during a burst read.
The values 0 to 3 written to the bits constitute the number of wait cycles inserted. The contents set here are applied to both areas 10 and 9. The wait cycles set by AxxWT are inserted in the first read cycle of burst ROM and in the burst ROM write cycle. For the burst ROM write cycle, the wait cycles set via the #WAIT pin can also be used.
At cold start, A10BW is set to "0" (no wait cycle). At hot start, A10BW retains its status before being initialized.

RBCLK: BCLK output control (DF) / Bus control register (0x4812E)

Control the bus clock BCLK to enable or disable external output.

Write "1": Fixed at high level
Write "0": Output enabled
Read: Valid

To stop outputting the bus clock from the BCLK pin, write "1" to RBCLK. When the clock output is stopped, the BCLK pin is fixed at high level. The bus clock output from the BCLK pin is enabled by writing "0" to RBCLK.
The bus clock output from the BCLK pin also is stopped in the HALT2 and the SLEEP modes.
At cold start, the RBCLK is set to "0" (output enabled). At hot start, RBCLK retains its status before being initialized.

RBST8: Burst mode selection (DD) / Bus control register (0x4812E)

Set the operation mode during a burst read.

Write "1": 8-successive-burst mode
Write "0": 4-successive-burst mode
Read: Valid

The 8-successive-burst mode is selected by writing "1" to RBST8 and the 4-successive-burst mode is selected by writing "0" to RBST8. This setting is valid when areas 10 and 9 are set for burst ROM, and the setting is applied to both areas simultaneously.
At cold start, RBST8 is set to "0" (4-successive-burst mode). At hot start, RBST8 retains its status before being initialized.

REDO: Page mode selection (DC) / Bus control register (0x4812E)

Select the page mode of DRAM.

Write "1": EDO-page mode
Write "0": Fast-page mode
Read: Valid

When using EDO DRAM, write "1" to REDO to select the EDO-page mode.
The contents set here are applied to all of areas 14, 13, 8, and 7 that are set for DRAM.
At cold start, REDO is set to "0" (fast-page mode). At hot start, REDO retains its status before being initialized.

RCA1–RCA0: Column address size selection (D[B:A]) / Bus control register (0x4812E)

Select the column address size of DRAM.

Table 7.9.3 Column Address Size

RCA1	RCA0	Column address size
1	1	11
1	0	10
0	1	9
0	0	8

The contents set here are applied to all of areas 14, 13, 8, and 7 that are set for DRAM.
RCA can be read to obtain its set value.
At cold start, RCA is set to "0" (8 bits). At hot start, RCA retain its status before being initialized.

RPC2: Refresh enable (D9) / Bus control register (0x4812E)

Control the DRAM refresh function.

Write "1": Enabled
Write "0": Disabled
Read: Valid

When DRAM is connected directly, a refresh cycle is generated by writing "1" to RPC2. The internal refresh function is disabled by writing "0" to RPC2.
Since the BCU stops operating in the HALT2 and the SLEEP modes, no refresh cycle is generated regardless of how this bit is set.
At cold start, RPC2 is set to "0" (disabled). At hot start, RPC2 retains its status before being initialized.

RPC1: Refresh method selection (D8) / Bus control register (0x4812E)

Select the DRAM refresh method.

Write "1": Self-refresh
Write "0": CAS-before-RAS refresh
Read: Valid

To perform a CAS-before-RAS refresh, set RPC1 to "0" and then RPC2 to "1". This causes the underflow output signal of the 8-bit programmable timer 0 is fed to the DRAM interface, at which timing a refresh cycle is generated.
To start a self-refresh, set RPC1 to "1" and then RPC2 to "1". The self-refresh is disabled by writing "0" to RPC2.
The contents set here are applied to all of areas 14, 13, 8, and 7 that are set for DRAM.
At cold start, RPC1 is set to "0" (CAS-before-RAS refresh). At hot start, RPC1 retains its status before being initialized.

RPC0: Refresh RPC delay (D7) / Bus control register (0x4812E)

Set a RPC delay when at start of refresh.

Write "1": 2 cycles
Write "0": 1 cycle
Read: Valid

Set a time from the immediately preceding precharge to the falling transition of #CAS necessary in order to perform a refresh. This time is 2 cycles when RPC0 = "1" or 1 cycle when RPC0 = "0".
The contents set here are applied to all of areas 14, 13, 8, and 7 that are set for DRAM.
At cold start, RPC0 is set to "0" (1 cycle). At hot start, RPC0 retains its status before being initialized.

RRA1–RRA0: Refresh RAS pulse width selection (D[6:5]) / Bus control register (0x4812E)

Select the RAS pulse width of a CAS-before-RAS refresh.

Table 7.9.4 Refresh RAS Pulse Width

RRA1	RRA0	Pulse width
1	1	5 cycles
1	0	4 cycles
0	1	3 cycles
0	0	2 cycles

The contents set here are applied to all of areas 14, 13, 8, and 7 that are set for DRAM.
The RRA can be read to obtain their set value.
At cold start, RRA is set to "0" (2 cycles). At hot start, RRA retains its status before being initialized.

SBUSST: External interface method select register (D3) / Bus control register (0x4812E)

Select the interface method of an SRAM device.

Write "1":	#BSL system
Write "0":	A0 system
Read:	Valid

When using the #BSL system, write "1" to SBUSST.
The contents set here are applied to all areas that are set for the SRAM type.
At cold start, SBUSST is set to "0" (A0 system). At hot start, SBUSST retains its status before being initialized.

SEMAS: External bus master setup (D2) / Bus control register (0x4812E)

Specify whether an external bus master exists.

Write "1":	Existing
Write "0":	Nonexistent
Read:	Valid

A request for bus ownership control via the #BUSREQ pin is made acceptable by writing "1" to SEMAS. If the system does not have any external bus master, fix this register at "0".
At cold start, SEMAS is set to "0" (nonexistent). At hot start, SEMAS retains its status before being initialized.

SEPD: External power-down control (D1) / Bus control register (0x4812E)

Enable or disable the CPU's power-down control by an external bus master.

Write "1":	Enabled
Write "0":	Disabled
Read:	Valid

Power-down control via an external pin (#BUSREQ) is enabled by writing "1" to SEPD. If the #BUSREQ pin is lowered when external power-down control is thus enabled, the CPU is placed in a HALT state, allowing for reduction in power consumption.
At cold start, SEPD is set to "0" (disabled). At hot start, SEPD retains its status before being initialized.

SWAITE: #WAIT enable (D0) / Bus control register (0x4812E)

Enable or disable wait cycle control via the #WAIT pin.

Write "1":	Enabled
Write "0":	Disabled
Read:	Valid

A wait request from an SRAM device is made acceptable by writing "1" to SWAITE. The wait request signal input from the #WAIT pin is sampled at each falling edge of the bus clock when executing an SRAM read/write cycle. Wait cycles are inserted until the wait request signal is sampled and detected as high (inactive).
Wait control for 0 to 7 cycles can be accomplished by AxxWT without using the #WAIT pin. However, since the setting via AxxWT is applied to every two areas, the number of wait cycles may be controlled individually in each area or more than 7 wait cycles may be set. In such a case, use an external wait request via the #WAIT pin.
Wait requests from the #WAIT pin are ignored when SWAITE = "0".
The contents set here are applied to all areas that are set for SRAM, and are also effective for write cycles in the areas that are set for burst ROM.
At cold start, SWAITE is set to "0" (disabled). At hot start, SWAITE retains its status before being initialized.

CEFUNC: #CE pin function selection (D9) / DRAM timing set-up register (0x48130)

Change the #CE pin-assigned area.

Write "1":	High address
Write "0":	Normal
Read:	Valid

Table 7.9.5 #CE Output Assignment

CEFUNC = "0"	CEFUNC = "1"
#CE4	#CE11
#CE5	#CE15
#CE6	#CE6
#CE7/#RAS0	#CE13/#RAS2
#CE8/#RAS1	#CE14/#RAS3
#CE9	#CE17
#CE10	#CE10

(default: CEFUNC = "0")

The high-order areas that are made useful by writing "1" to CEFUNC can be larger than those of the default low-order areas.
At cold start, CEFUNC is set to "0" (normal). At hot start, CEFUNC retains its status before being initialized.

CRAS: Successive RAS mode (D8) / DRAM timing set-up register (0x48130)

Set the successive RAS mode.

Write "1":	Successive RAS mode
Write "0":	Normal mode
Read:	Valid

In systems using DRAM, the successive RAS mode is entered by writing "1" to CRAS. In this mode, read/write operations can be performed in page mode even when DRAM accesses do not occur back-to-back.
When using the successive RAS mode, be sure to use #DRD for the read signal and #DWE for the write signal for low-byte.
When CRAS = "0", random read/write cycles are used for non-successive DRAM accesses.
The contents set here are applied to all of areas 14, 13, 8, and 7 that are set for DRAM.
At cold start, CRAS is set to "0" (normal mode). At hot start, CRAS retains its status before being initialized.

RPRC1–RPRC0: Number of RAS precharge cycles (D[7:6]) / DRAM timing set-up register (0x48130)

Select the number of precharge cycles during a DRAM access.

Table 7.9.6 Number of RAS Precharge Cycles

RPRC1	RPRC0	Number of cycles
1	1	4 cycles
1	0	3 cycles
0	1	2 cycles
0	0	1 cycle

The contents set here are applied to all of areas 14, 13, 8, and 7 that are set for DRAM.
At cold start, RPRC is set to "0" (1 cycle). At hot start, RPRC retains its status before being initialized.

CASC1–CASC0: Number of CAS cycles (D[4:3]) / DRAM timing set-up register (0x48130)

Select the number of CAS cycles during a DRAM access.

Table 7.9.7 Number of CAS Cycles

CASC1	CASC0	Number of cycles
1	1	4 cycles
1	0	3 cycles
0	1	2 cycles
0	0	1 cycle

The contents set here are applied to all of areas 14, 13, 8, and 7 that are set for DRAM.
At cold start, CASC is set to "0" (1 cycle). At hot start, CASC retains its status before being initialized.

RASC1–RASC0: Number of RAS cycles (D[1:0]) / DRAM timing set-up register (0x48130)

Select the number of RAS cycles during a DRAM access.

Table 7.9.8 Number of RAS Cycles

RASC1	RASC0	Number of cycles
1	1	4 cycles
1	0	3 cycles
0	1	2 cycles
0	0	1 cycle

The contents set here are applied to all of areas 14, 13, 8, and 7 that are set for DRAM.
At cold start, RASC is set to "0" (1 cycle). At hot start, RASC retains its status before being initialized.

7.10 Programming Note

By writing a byte to the BCU register (address 0x0048130), the data will be written to both 0x0048130 and 0x0048131 addresses. When writing a byte to address 0x0048131, the register's data will not be modified.
Write in half-word or word units to address 0x0048130.

8 Interrupt

The E0C33A104 contains an interrupt controller, making it possible to control all interrupts generated by the internal peripheral circuits. This chapter explains the functions of this interrupt controller centering around the method for controlling maskable interrupts. For details about the various causes and conditions under which interrupts are generated, refer to the description of each peripheral circuit in this manual.

8.1 Outline of Interrupt Functions

8.1.1 Maskable Interrupts

The E0C33A104 has 11 systems and 45 kinds of maskable interrupts available, as shown in the table below.

Table 8.1.1 List of Maskable Interrupts

No.	Vector number (Address)	Interrupt system (Peripheral circuit)	Interrupt factor	IDMA Ch.	Priority
0	16(Base+64)	Input port K64–K67	K67 port, input rising or falling edge	1	High ↑
1	17(Base+68)		K66 port, input rising or falling edge	2	
2	18(Base+72)		K65 port, input rising or falling edge	3	
3	19(Base+76)		K64 port, input rising or falling edge	4	
4	20(Base+80)	Input port K60–K63, K50–K54	K60–K63 ports, input rising or falling edge	–	
5	21(Base+84)		K50–K54 ports, input rising or falling edge	–	
6	22(Base+88	reserved	–	–	
7	23(Base+92)		–	–	
8	24(Base+96)	High-speed DMA • IDMA	High-speed DMA Ch.0, end of transfer	5	
9	25(Base+100)		High-speed DMA Ch.1, end of transfer	6	
10	26(Base+104)		Intelligent DMA, end of transfer	–	
11	27(Base+108)	reserved	–	–	
12	28(Base+112)	16-bit programmable timer Timers 00, 01, 10, 11	Timer 00 underflow	–	
13	29(Base+116)		Timer 00 compare match	–	
14	30(Base+120)		Timer 01 underflow	7	
15	31(Base+124)		Timer 01 compare match	8	
16	32(Base+128)		Timer 10 underflow	–	
17	33(Base+132)		Timer 10 compare match	–	
18	34(Base+136)		Timer 11 underflow	9	
19	35(Base+140)		Timer 11 compare match	10	
20	36(Base+144)	16-bit programmable timer Timers 20, 21, 30, 31	Timer 20 underflow	–	
21	37(Base+148)		Timer 20 compare match	–	
22	38(Base+152)		Timer 21 underflow	11	
23	39(Base+156)		Timer 21 compare match	12	
24	40(Base+160)		Timer 30 underflow	–	
25	41(Base+164)		Timer 30 compare match	–	
26	42(Base+168)		Timer 31 underflow	13	
27	43(Base+172)		Timer 31 compare match	14	
28	44(Base+176)	16-bit programmable timer Timers 40, 41, 50, 51	Timer 40 underflow	–	
29	45(Base+180)		Timer 40 compare match	–	
30	46(Base+184)		Timer 41 underflow	15	
31	47(Base+188)		Timer 41 compare match	16	
32	48(Base+192)		Timer 50 underflow	–	
33	49(Base+196)		Timer 50 compare match	–	
34	50(Base+200)		Timer 51 underflow	17	
35	51(Base+204)		Timer 51 compare match	18	
36	52(Base+208)	8-bit programmable timer	Timer 0 underflow	19	
37	53(Base+212)		Timer 1 underflow	20	
38	54(Base+216)		Timer 2 underflow	21	
39	55(Base+220)		Timer 3 underflow	22	
40	56(Base+224)	Serial interface Ch.0	Receive error	–	
41	57(Base+228)		Receive buffer full	23	
42	58(Base+232)		Transmit buffer empty	24	
43	59(Base+236)	reserved	–	–	
44	60(Base+240)	Serial interface Ch.1	Receive error	–	
45	61(Base+244)		Receive buffer full	25	
46	62(Base+248)		Transmit buffer empty	26	
47	63(Base+252)	reserved	–	–	
48	64(Base+256)	A/D converter	A/D converter, end of conversion	27	
49	65(Base+260)	Clock timer	Falling edge of 32 Hz, 8 Hz, 2 Hz or 1 Hz signal 1-minuet, 1-hour or specified time count up	–	↓ Low

Contents of table

"No." indicates an interrupt number.

"Vector number (Address)" indicates the trap table's vector number. The numerals in parentheses show an offset (in bytes) from the starting address (Base) of the trap table. The starting address (Base) of the trap table by default is the boot address. Depending on how the #BTA3 pin is set at an initial reset, this address is 0x80000 (#BTA3 = High) or 0xC00000 (#BTA3 = Low). This address can be changed using the TTBR register (0x48134 to 0x48137). (For details on how to change, refer to Section 8.2.)

For details about the trap table contents including exception factors, etc., refer to the "E0C33000 Core CPU Manual".

"Interrupt system (Peripheral circuit)" indicates that interrupt levels can be programmed for each peripheral circuit written.

"Interrupt factor" indicates the factor of the interrupt occurring in each interrupt system.

"IDMA Ch." indicates that an interrupt factor which has a numeric value in this column can start up the intelligent DMA (IDMA) to transfer data when an interrupt factor occurs. The numeric value indicates the IDMA's channel number. Interrupt factors that do not have a numeric value here cannot start up the IDMA.

"Priority" indicates the priority of interrupts in cases when all interrupt systems are set to the same interrupt level. If two or more interrupt factors occur simultaneously, interrupt requests are accepted in order of highest priority. Interrupt priority varies depending on the interrupt levels set in each interrupt system. However, the priorities of interrupt factors in the same interrupt system are fixed in the order that they are written here.

Maskable interrupt generating conditions

A maskable interrupt to the CPU occurs when all of the conditions described below are met.

- The interrupt enable register for the interrupt factor that has occurred is set to "1".
- The IE (Interrupt Enable) bit of the Processor Status Register (PSR) in the CPU is set to "1".
- The interrupt factor that has occurred has a higher priority level than the value that is set in the PSR's Interrupt Level (IL). (The interrupt levels can be set using the interrupt priority register in each interrupt system.)
- No other trap factor having higher priority, such as NMI, has occurred.

When an interrupt factor occurs, the corresponding interrupt factor flag is set to "1" and the flag remains set until it is reset in the software program. Therefore, in no cases can the generated interrupt factor be inadvertently cleared even if the above conditions are not met when the interrupt factor has occurred. The interrupt will occur when the above conditions are met.

If two or more maskable interrupt factors occur simultaneously, the interrupt factor that has the highest priority is allowed to signal an interrupt request to the CPU. The other interrupts with lower priorities are kept pending until the above conditions are met.

The PSR and interrupt control register will be detailed later.
For details about interrupt factor generating conditions, refer to the description of each peripheral circuit in this manual.

8.1.2 Interrupt Factors and Intelligent DMA

Several interrupt factors can be set so that they can invoke IDMA startup. When one of these interrupt factors occurs, IDMA is started up before an interrupt request to the CPU. The interrupt request to the CPU is generated after IDMA is completed. (The interrupt request can be disabled by a program.)
IDMA is always started up regardless of how the PSR is set. For details, refer to Section 8.4, "IDMA Request Register and IDMA Invocation".

8.1.3 Nonmaskable Interrupt (NMI)

The nonmaskable interrupt (NMI) can be generated by pulling the #NMI pin low or using the internal watchdog timer. The vector number of NMI is 7, with the vector address set to the trap table's starting address + 28 bytes.
This interrupt is prioritized over other interrupts and is unconditionally accepted by the CPU.
However, since this interrupt may operate erratically if it occurs before the stack pointer (SP) is set up, it is masked in hardware until a write to the SP is completed after an initial reset.

8.1.4 Interrupt Processing by the CPU

The CPU keeps sampling interrupt requests every cycle. When the CPU accepts an interrupt request, it enters trap processing after completing execution of the instruction that was being executed.
The following lists the contents executed in trap processing.

(1) The PSR and the current program counter (PC) value are saved to the stack.
(2) The IE bit of the PSR is reset to "0" (following maskable interrupts are disabled).
(3) The IL of the PSR is set to the priority level of the accepted interrupt (NMI does not have its interrupt level changed).
(4) The vector of the generated interrupt factor is loaded into the PC, thus executing the interrupt processing routine.

Thus, once an interrupt is accepted, all maskable interrupts that may follow are disabled in (2). Multiple interrupts can also be handled by setting the IE bit to "1" in the interrupt processing routine. In this case, since the IL has been changed in (3), only an interrupt that has a higher priority than that of the currently processed interrupt is accepted.
When the interrupt processing routine is terminated by the reti instruction, the PSR is restored to its previous status before the interrupt has occurred. The program restarts processing after branching to the instruction next to the one that was being executed when the interrupt occurred.

8.1.5 Clearing Standby Mode by Interrupts

The standby modes (HALT and SLEEP) are cleared by an NMI or a maskable interrupt.
All maskable interrupts can be used to clear HALT mode. However, if the bus clock has stopped in HALT2 mode, a DMA interrupt cannot be used.
In SLEEP mode, since the high-speed (OSC3) oscillation circuit is deactivated, interrupts from the peripheral circuits that operate with the OSC3 clock cannot be used.

Interrupts that can be used to clear basic HALT mode:	NMI and all maskable interrupts
Interrupts that can be used to clear HALT2 mode:	NMI and all maskable interrupts (except DMA interrupts)
Interrupts that can be used to clear SLEEP mode:	NMI, input port interrupts, and clock timer interrupts

Clearing of the standby modes is accomplished by an interrupt request to the CPU. Therefore, this requires that the PSR be set in such a way that the requested interrupt will be accepted, and that the interrupt enable register for the interrupt factor be set to accept the interrupt.
When standby mode is cleared and the CPU has accepted the interrupt, it returns to the instruction next to the halt or slp instruction after executing the interrupt processing routine.

Note: If the interrupt factor used to restart from the standby mode has been set to invoke the IDMA, the IDMA is started up by that interrupt.
In the case of SLEEP mode, the high-speed (OSC3) oscillation circuit also starts operating.
If an interrupt to be generated upon completion of IDMA is disabled at the setting of the IDMA side, no interrupt request is signaled to the CPU. Therefore, the CPU remains idle until the next interrupt request is generated.

8.2 Trap Table

The E0C33A104 allows the base (starting) address of the trap table to be set by the TTBR register.

TTBR0 (D[9:0]) / TTBR low-order register (0x48134):	Trap table base address [9:0] (fixed at "0")
TTBR1 (D[F:A]) / TTBR low-order register (0x48134):	Trap table base address [15:10]
TTBR2 (D[B:0]) / TTBR high-order register (0x48136):	Trap table base address [27:16]
TTBR3 (D[F:C]) / TTBR high-order register (0x48136):	Trap table base address [31:28] (fixed at "0")

After an initial reset, the TTBR register is set to the boot address determined by the BTA3 pin status.
BTA3 = low: 0x0C00000

Therefore, even when the trap table position is changed, it is necessary that at least the reset vector be written to the above address.

TTBR0 and TTBR3 are read-only bits which are fixed at "0". Therefore, the trap table starting address always begins with a 1KB boundary address.
The TTBR register is normally write-protected to prevent them from being inadvertently rewritten. To remove this write protection function, another register, TBRP (D[F:8]) / TTBR write-protect register (0x4812D [byte]), is provided. A write to the TTBR register is enabled by writing "0x59" to TBRP and is disabled back again by a write to the most significant byte of the TTBR register (0x48137). Consequently, a write to the TTBR register needs to begin with the low-order half-word first. However, since an occurrence of NMI or the like between writes of the low-order and high-order half-words would cause a malfunction, it is recommended that the register be written in words.

8.3 Control of Maskable Interrupts

8.3.1 Structure of the Interrupt Controller

The interrupt controller is configured as shown in Figure 8.3.1.

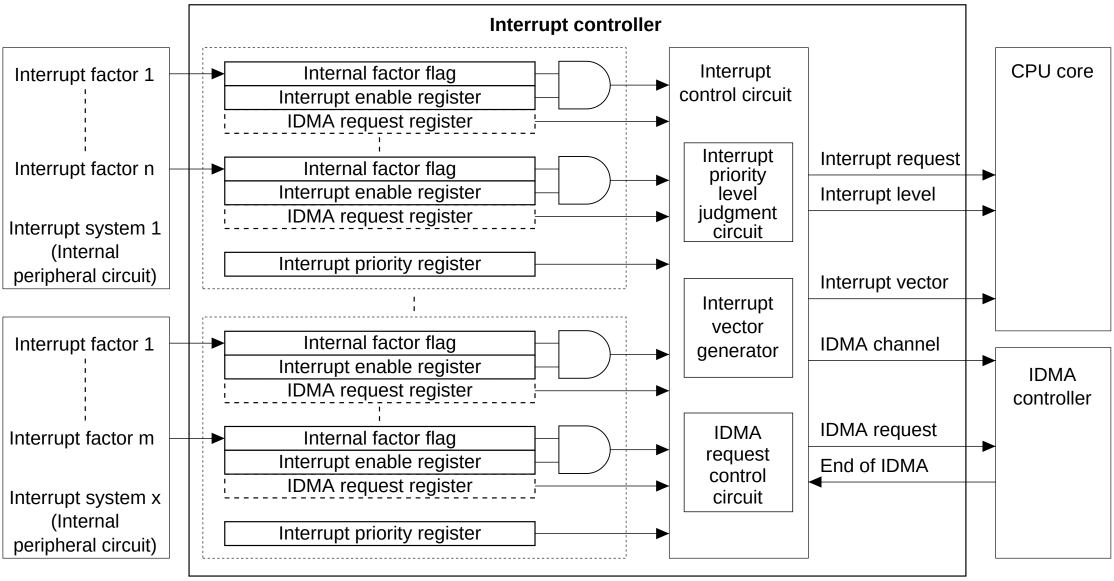

Interrupt controller
Interrupt factor 1
Interrupt factor n
Interrupt system 1 (Internal peripheral circuit)
Interrupt factor 1
Interrupt factor m
Interrupt system x (Internal peripheral circuit)
Internal factor flag
Interrupt enable register
IDMA request register
Interrupt priority register
Interrupt control circuit
Interrupt priority level judgment circuit
Interrupt vector generator
IDMA request control circuit
Interrupt request
Interrupt level
Interrupt vector
IDMA channel
IDMA request
End of IDMA
CPU core
IDMA controller

Figure 8.3.1 Configuration of Interrupt Controller

The following sections explain the functions of the registers used to control interrupts.

8.3.2 Processor Status Register (PSR)

The PSR is a special register incorporated in the core CPU and contains control bits to enable or disable an interrupt request to the CPU.

Interrupt Enable (IE) bit: PSR[4]

This bit is used to enable or disable an interrupt request to the CPU. When this bit is set to "1", the CPU is enabled to accept a maskable interrupt request. When this bit is reset to "0", no maskable interrupt request is accepted by the CPU.

When the CPU accepts an interrupt request (or some other trap occurs), it saves the PSR to the stack and resets the IE bit to "0". Consequently, no maskable interrupt request occurring thereafter will be accepted unless the IE bit is set to "1" in software program or the interrupt (trap) processing routine is terminated by the reti instruction.

The IE bit is initialized to "0" (interrupts disabled) by an initial reset.

Interrupt Level (IL): PSR[11:8]

The IL bits disable the interrupts whose priorities are below the set interrupt level. For example, if the interrupt level set in the IL is 3, the interrupts whose priorities are set below 3 in the interrupt priority register (described later) are not accepted by the CPU even if the IE bit is set to "1". The IL and the interrupt priority register together allow you to control the interrupt priorities in each interrupt system. For details about the interrupt levels, refer to Section 8.3.4, "Interrupt Priority Register and Interrupt Levels".

When the CPU accepts a maskable interrupt request, it saves the PSR to the stack and sets the IL to the accepted interrupt's priority level. Therefore, even when the IE bit is set to "1" in the interrupt processing routine, no interrupts whose priority levels are equal or below that of the interrupt currently being processed are accepted unless the IL is rewritten.

The IL is restored to its previous status when the interrupt processing routine is terminated by the reti instruction.

The IL is rewritten for only maskable interrupts and not for any other traps (except a reset).
The IL is set to level 0 (that is, all interrupts above level 1 are enabled) by an initial reset.

Note: As the E0C33000 Core CPU function, the IL allows interrupt levels to be set in the range of 0 to 15. However, since the interrupt priority register in the E0C33A104 consists of three bits, interrupt levels in each interrupt system can only be set for up to 8.

8.3.3 Interrupt Factor Flag and Interrupt Enable Register

An interrupt factor flag and an interrupt enable register are provided for each maskable interrupt factor.

Interrupt factor flag

The interrupt factor flag is set to "1" when the corresponding interrupt factor occurs. Reading the flag enables you to determine what caused an interrupt, making it unnecessary to resort to the CPU's trap processing. The interrupt factor flag is reset only by writing data in software. Note that the method by which this flag is reset can be selected from the software application using either of the two methods described below. This selection is accomplished using RSTONLY (D0) / Interrupt factor flag reset method select register (0x4029F).

- **Reset-only method (default)**

  This method is selected (RSTONLY = "1") when initially reset.
  With this method, the interrupt factor flag is reset by writing "1". Although multiple interrupt factor flags are located at the same address of the interrupt control register, the interrupt factor flags for which "0" has been written can be neither set nor reset. Therefore, this method ensures that only a specific factor flag is reset. However, when using read-modify-write instructions (e.g., bset, bclr, or bnot), note that an interrupt factor flag that has been set to "1" is reset by writing.

  In this method, no interrupt factor flag can be set in the software application.

- **Read/write method**

  This method is selected by writing "0" to RSTONLY.
  When this method is used, interrupt factor flags can be read and written as for other registers. Therefore, the flag is reset by writing "0" and set by writing "1". In this case, all factor flags for which "0" has been written are reset. Even in a read-modify-write operation, an interrupt factor can occur between the read and the write, so be careful when using this method.

Since interrupt factor flags are not initialized by an initial reset, be sure to reset them before enabling interrupts.

Note: Even when a maskable interrupt request is accepted by the CPU and control branches off to the interrupt processing routine, the interrupt factor flag is not reset. Consequently, if control is returned from the interrupt processing routine by the reti instruction without resetting the interrupt factor flag in a program, the same interrupt factor occurs again.

For details about interrupt factor generating conditions, refer to the description of each peripheral circuit in this manual.

Interrupt enable register

This register controls the output of an interrupt request to the CPU. Only when the interrupt enable bit of this register is set to "1" can an interrupt request to the CPU be enabled by an occurrence of the corresponding interrupt factor. If the bit is set to "0", no interrupt request is made to the CPU even when the corresponding interrupt factor occurs.

Settings of the interrupt enable register do not affect the operation of interrupt factor flags, so when an interrupt factor occurs the interrupt factor flag is set to "1" even if the corresponding interrupt enable bit is set to "0".

When initially reset, the interrupt enable register is set to "0" (interrupts are disabled). Since this register can be accessed for read instructions, its setup status can be checked at any time.

In cases when IDMA is started up by occurrence of an interrupt factor or when clearing standby mode (HALT or SLEEP mode) too, the corresponding interrupt enable bit must be set to "1".

Note: When an interrupt is disabled using the interrupt enable bit (0x0040270–0x0040277) at the same time the corresponding interrupt factor occurs, the interrupt processing read a wrong vector from [TTBR+284(decimal)] (reserved vector number). Therefore the program does not jump to the correct interrupt service routine.

To avoid this problem, an interrupt enable bit in ITC should be reset by the following procedure:

1) Clear the IE bit in PSR.

2) Execute the NOP instruction

3) Reset the interrupt enable bit in ITC.

Example:

```
ld.w    %r0,%psr
ext     0x3ff
and     %r0,0x2f
ld.w    %psr,%r0        ; clear IE
nop
ld.b    [%r1],%r2       ; %r1=0x00040274, %r2=00000000
                        ; disable interrupt
```

The interrupt controller outputs an interrupt request to the CPU when the following conditions are met:

- An interrupt factor has occurred and the interrupt factor flag is set to "1".
- The bit of the interrupt enable register for the interrupt factor that has occurred is set to "1" (interrupt enable).

If two or more interrupt factors occur simultaneously, the interrupt factor that has the highest priority is allowed to signal an interrupt request to the CPU. (See the following section.)

When these conditions are met, the interrupt controller outputs an interrupt request signal to the CPU along with the setup content (interrupt level) of the interrupt priority register for the generated interrupt system and its vector number.

These signals remain asserted until the interrupt factor flag is reset to "0" or the corresponding bit of the interrupt enable register is set to "0" (interrupts are disabled) in the software application or until some other interrupt factor of higher priority occurs. They are not cleared if the CPU simply accepts the interrupt request.

8.3.4 Interrupt Priority Register and Interrupt Levels

The interrupt priority register is a 3-bit register provided for each interrupt system. It allows the interrupt levels of a given interrupt system to be set in the range of 0 to 7. The default priorities shown in Table 8.1.1 can be modified according to system requirements by this setting.

The value set in this register is used by the interrupt controller and the CPU as described below.

Roles of the interrupt priority register in the interrupt controller

If two or more interrupt factors that have been enabled by the interrupt enable register occur simultaneously, the interrupt factor in the interrupt system whose interrupt priority register contains the greatest value is allowed by the interrupt controller to signal an interrupt request to the CPU.

If an interrupt factor occurs in two or more interrupt systems having the same value, the interrupt priority is resolved according to the default priorities in Table 8.1.1. Interrupt factors in the same interrupt system also have their priorities resolved according to the order in Table 8.1.1.

Other interrupt factors are kept pending until all interrupts of higher priority are accepted by the CPU.

When outputting an interrupt request signal to the CPU, the interrupt controller outputs the content of the interrupt priority register to the CPU along with it.

If when outputting an interrupt request signal another interrupt factor of higher priority occurs, the interrupt controller changes the vector number and interrupt level to those of the new interrupt factor before they are output to the CPU. The first interrupt request is left pending.

Roles of the interrupt priority register in CPU processing

The CPU compares the content of the interrupt priority register received from the interrupt controller with the interrupt level that is set in the IL of the PSR to determine whether or not to accept the interrupt request.

IE bit = "1" & IL < interrupt priority register: the interrupt request is accepted
IE bit = "1" & IL ≥ interrupt priority register: the interrupt request is rejected

Before interrupts can be controlled by an interrupt level, the interrupt disabling level must be written to the IL. For example, if the value written to the IL is 3, only the interrupts whose interrupt levels written in the interrupt priority register are 4 or more will be accepted.

When an interrupt is accepted, the interrupt level that is set in its interrupt priority register is written to the IL. As a result, the interrupt requests below that interrupt level can no longer be accepted.

If the interrupt priority register for an interrupt is set to "0", the interrupt is disabled. However, invoking IDMA by means of an interrupt factor works fine.

Notes:
- As the E0C33000 Core CPU function, the IL allows interrupt levels to be set in the range of 0 to 15. However, since the interrupt priority register in the E0C33A104 consists of three bits, interrupt levels in each interrupt system can only be set for up to 8.
- Multiple interrupts can also be handled by rewriting the interrupt level to the IL in the interrupt processing routine. However, if the interrupt level of the IL is set below the current level and the IE is set to enable interrupts before resetting the interrupt factor flag after an interrupt has occurred, the same interrupt may occur again.

8.4 IDMA Request Register and IDMA Invocation

The interrupt factors for which IDMA channel numbers are written in Table 8.1.1 have the function to invoke the intelligent DMA (IDMA).
Before IDMA can be invoked by the occurrence of an interrupt factor, the corresponding bit of the IDMA request register must be set to "1". Then when an interrupt factor occurs, the interrupt request to the CPU is made pending and the corresponding IDMA channel is invoked. The DMA transfer is performed according to the control information of that IDMA channel.
When invoking IDMA, the corresponding bit of the interrupt enable register must be set to enable the interrupt. An IDMA invocation request is accepted even when the internal PSR of the CPU is set to disable interrupts. It is also necessary that DMA be enabled on the IDMA side by setting the control information required for DMA transfer.

The interrupt request that has been kept pending can be generated after completion of the DMA transfer.
For this to be possible, the interrupt request must be rendered acceptable by the interrupt controller and the CPU's internal PSR while at the same time the interrupt is enabled by IDMA's control information (DINTEM = "1").
However, if the transfer counter set for the selected IDMA channel does not reach the terminal count of 0 after the number of transfers set have been performed, the interrupt factor flag is reset and no interrupt request is generated.
The transfer counter is decremented by 1 for each transfer performed.
If the transfer counter is decremented to 0, the interrupt factor flag is not reset and the interrupt request is generated.
If an interrupt request is generated, the corresponding bit of the IDMA request register is cleared to "0". Therefore, this register must be set up again in order for IDMA to be invoked when an interrupt factor occurs next time as well.
To ensure that no unwanted IDMA request occurs, this setup must be performed before enabling interrupts after resetting the interrupt factor flag.
If an interrupt request is not generated, the corresponding bit of the IDMA request register remains set to "1" without being cleared.

Note that although IDMA has a function to generate an interrupt at completion of DMA transfer, if IDMA has been invoked by an interrupt factor, no interrupt is generated by IDMA for the completion of transfer even when DINTEM is set for interrupt enable. The IDMA interrupt is valid only when IDMA has been invoked by the software application.
For details about IDMA, refer to Section 9.2, "Intelligent DMA".

The IDMA request register is not initialized by an initial reset. To ensure that no unwanted IDMA invocation request may occur, be sure to initialize this register in the software application.

8.5 I/O Memory of Interrupt Controller

Table 8.5.1 shows the control bits of the interrupt controller.

Table 8.5.1 Control Bits of Interrupt Controller

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
K6/K5 interrupt priority register	0040260 (B)	D7	–	reserved	–				–	–	0 when being read.
		D6	**PK6L2**	K6[3:0], K5[4:0] interrupt level	0 to 7				X	R/W	
		D5	**PK6L1**						X		
		D4	**PK6L0**						X		
		D3	–	reserved	–				–	–	0 when being read.
		D2	**PK6H2**	K6[7:4] interrupt level	0 to 7				X	R/W	
		D1	**PK6H1**						X		
		D0	**PK6H0**						X		
16-bit timer 0–1, DMA interrupt priority register	0040261 (B)	D7	–	reserved	–				–	–	0 when being read.
		D6	**P16T02**	16-bit timer 0–1 interrupt level	0 to 7				X	R/W	
		D5	**P16T01**						X		
		D4	**P16T00**						X		
		D3	–	reserved	–				–	–	0 when being read.
		D2	**PDM2**	IDMA, high-speed DMA interrupt level	0 to 7				X	R/W	
		D1	**PDM1**						X		
		D0	**PDM0**						X		
16-bit timer 2–5 interrupt priority register	0040262 (B)	D7	–	reserved	–				–	–	0 when being read.
		D6	**P16T42**	16-bit timer 4–5 interrupt level	0 to 7				X	R/W	
		D5	**P16T41**						X		
		D4	**P16T40**						X		
		D3	–	reserved	–				–	–	0 when being read.
		D2	**P16T22**	16-bit timer 2–3 interrupt level	0 to 7				X	R/W	
		D1	**P16T21**						X		
		D0	**P16T20**						X		
8-bit timer, serial I/F Ch.0 interrupt priority register	0040263 (B)	D7	–	reserved	–				–	–	0 when being read.
		D6	**PSI002**	Serial interface Ch.0 interrupt level	0 to 7				X	R/W	
		D5	**PSI001**						X		
		D4	**PSI000**						X		
		D3	–	reserved	–				–	–	0 when being read.
		D2	**P8TM2**	8-bit timer 0–3 interrupt level	0 to 7				X	R/W	
		D1	**P8TM1**						X		
		D0	**P8TM0**						X		
Serial I/F Ch.1, A/D interrupt priority register	0040264 (B)	D7	–	reserved	–				–	–	0 when being read.
		D6	**PAD2**	A/D converter interrupt level	0 to 7				X	R/W	
		D5	**PAD1**						X		
		D4	**PAD0**						X		
		D3	–	reserved	–				–	–	0 when being read.
		D2	**PSI012**	Serial interface Ch.1 interrupt level	0 to 7				X	R/W	
		D1	**PSI011**						X		
		D0	**PSI010**						X		
Clock timer interrupt priority register	0040265 (B)	D7–3	–	reserved	–				–	–	Writing 1 not allowed.
		D2	**PCTM2**	Clock timer interrupt level	0 to 7				X	R/W	
		D1	**PCTM1**						X		
		D0	**PCTM0**						X		
K6/K5 interrupt enable register	0040270 (B)	D7–6	–	reserved	–				–	–	0 when being read.
		D5	**EK5**	K5[4:0] input	1	Enabled	0	Disabled	0	R/W	
		D4	**EK6**	K6[3:0] input					0	R/W	
		D3	**EK64**	K64 input					0	R/W	
		D2	**EK65**	K65 input					0	R/W	
		D1	**EK66**	K66 input					0	R/W	
		D0	**EK67**	K67 input					0	R/W	
DMA interrupt enable register	0040271 (B)	D7–3	–	reserved	–				–	–	0 when being read.
		D2	**EIDMA**	IDMA	1	Enabled	0	Disabled	0	R/W	
		D1	**EHDM1**	High-speed DMA Ch.1					0	R/W	
		D0	**EHDM0**	High-speed DMA Ch.0					0	R/W	
16-bit timer 0–1 interrupt enable register	0040272 (B)	D7	**E16TC11**	16-bit timer 11 comparison match	1	Enabled	0	Disabled	0	R/W	
		D6	**E16TU11**	16-bit timer 11 underflow					0	R/W	
		D5	**E16TC10**	16-bit timer 10 comparison match					0	R/W	
		D4	**E16TU10**	16-bit timer 10 underflow					0	R/W	
		D3	**E16TC01**	16-bit timer 01 comparison match					0	R/W	
		D2	**E16TU01**	16-bit timer 01 underflow					0	R/W	
		D1	**E16TC00**	16-bit timer 00 comparison match					0	R/W	
		D0	**E16TU00**	16-bit timer 00 underflow					0	R/W	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 2–3 interrupt enable register	**0040273 (B)**	D7	**E16TC31**	16-bit timer 31 comparison match	1	Enabled	0	Disabled	0	R/W	
		D6	**E16TU31**	16-bit timer 31 underflow					0	R/W	
		D5	**E16TC30**	16-bit timer 30 comparison match					0	R/W	
		D4	**E16TU30**	16-bit timer 30 underflow					0	R/W	
		D3	**E16TC21**	16-bit timer 21 comparison match					0	R/W	
		D2	**E16TU21**	16-bit timer 21 underflow					0	R/W	
		D1	**E16TC20**	16-bit timer 20 comparison match					0	R/W	
		D0	**E16TU20**	16-bit timer 20 underflow					0	R/W	
16-bit timer 4–5 interrupt enable register	**0040274 (B)**	D7	**E16TC51**	16-bit timer 51 comparison match	1	Enabled	0	Disabled	0	R/W	
		D6	**E16TU51**	16-bit timer 51 underflow					0	R/W	
		D5	**E16TC50**	16-bit timer 50 comparison match					0	R/W	
		D4	**E16TU50**	16-bit timer 50 underflow					0	R/W	
		D3	**E16TC41**	16-bit timer 41 comparison match					0	R/W	
		D2	**E16TU41**	16-bit timer 41 underflow					0	R/W	
		D1	**E16TC40**	16-bit timer 40 comparison match					0	R/W	
		D0	**E16TU40**	16-bit timer 40 underflow					0	R/W	
8-bit timer interrupt enable register	**0040275 (B)**	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**E8TU3**	8-bit timer 3 underflow	1	Enabled	0	Disabled	0	R/W	
		D2	**E8TU2**	8-bit timer 2 underflow					0	R/W	
		D1	**E8TU1**	8-bit timer 1 underflow					0	R/W	
		D0	**E8TU0**	8-bit timer 0 underflow					0	R/W	
Serial I/F interrupt enable register	**0040276 (B)**	D7–6	**–**	reserved	–				–	–	0 when being read.
		D5	**ESTX1**	SIF Ch.1 transmit buffer empty	1	Enabled	0	Disabled	0	R/W	
		D4	**ESRX1**	SIF Ch.1 receive buffer full					0	R/W	
		D3	**ESERR1**	SIF Ch.1 receive error					0	R/W	
		D2	**ESTX0**	SIF Ch.0 transmit buffer empty					0	R/W	
		D1	**ESRX0**	SIF Ch.0 receive buffer full					0	R/W	
		D0	**ESERR0**	SIF Ch.0 receive error					0	R/W	
Clock timer, A/D interrupt enable register	**0040277 (B)**	D7–2	**–**	reserved	–				–	–	Writing 0 not allowed.
		D1	**ECTM**	Clock timer	1	Enabled	0	Disabled	0	R/W	
		D0	**EADE**	A/D converter					0	R/W	
K6/K5 interrupt factor flag register	**0040280 (B)**	D7–6	**–**	reserved	–				–	–	0 when being read.
		D5	**FK5**	K5[4:0] input	1	Factor is generated	0	No factor is generated	X	R/W	
		D4	**FK6**	K6[3:0] input					X	R/W	
		D3	**FK64**	K64 input					X	R/W	
		D2	**FK65**	K65 input					X	R/W	
		D1	**FK66**	K66 input					X	R/W	
		D0	**FK67**	K67 input					X	R/W	
DMA interrupt factor flag register	**0040281 (B)**	D7–3	**–**	reserved	–				–	–	0 when being read.
		D2	**FIDMA**	IDMA	1	Factor is generated	0	No factor is generated	X	R/W	
		D1	**FHDM1**	High-speed DMA Ch.1					X	R/W	
		D0	**FHDM0**	High-speed DMA Ch.0					X	R/W	
16-bit timer 0–1 interrupt factor flag register	**0040282 (B)**	D7	**F16TC11**	16-bit timer 11 comparison match	1	Factor is generated	0	No factor is generated	X	R/W	
		D6	**F16TU11**	16-bit timer 11 underflow					X	R/W	
		D5	**F16TC10**	16-bit timer 10 comparison match					X	R/W	
		D4	**F16TU10**	16-bit timer 10 underflow					X	R/W	
		D3	**F16TC01**	16-bit timer 01 comparison match					X	R/W	
		D2	**F16TU01**	16-bit timer 01 underflow					X	R/W	
		D1	**F16TC00**	16-bit timer 00 comparison match					X	R/W	
		D0	**F16TU00**	16-bit timer 00 underflow					X	R/W	
16-bit timer 2–3 interrupt factor flag register	**0040283 (B)**	D7	**F16TC31**	16-bit timer 31 comparison match	1	Factor is generated	0	No factor is generated	X	R/W	
		D6	**F16TU31**	16-bit timer 31 underflow					X	R/W	
		D5	**F16TC30**	16-bit timer 30 comparison match					X	R/W	
		D4	**F16TU30**	16-bit timer 30 underflow					X	R/W	
		D3	**F16TC21**	16-bit timer 21 comparison match					X	R/W	
		D2	**F16TU21**	16-bit timer 21 underflow					X	R/W	
		D1	**F16TC20**	16-bit timer 20 comparison match					X	R/W	
		D0	**F16TU20**	16-bit timer 20 underflow					X	R/W	
16-bit timer 4–5 interrupt factor flag register	**0040284 (B)**	D7	**F16TC51**	16-bit timer 51 comparison match	1	Factor is generated	0	No factor is generated	X	R/W	
		D6	**F16TU51**	16-bit timer 51 underflow					X	R/W	
		D5	**F16TC50**	16-bit timer 50 comparison match					X	R/W	
		D4	**F16TU50**	16-bit timer 50 underflow					X	R/W	
		D3	**F16TC41**	16-bit timer 41 comparison match					X	R/W	
		D2	**F16TU41**	16-bit timer 41 underflow					X	R/W	
		D1	**F16TC40**	16-bit timer 40 comparison match					X	R/W	
		D0	**F16TU40**	16-bit timer 40 underflow					X	R/W	
8-bit timer interrupt factor flag register	**0040285 (B)**	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**F8TU3**	8-bit timer 3 underflow	1	Factor is generated	0	No factor is generated	X	R/W	
		D2	**F8TU2**	8-bit timer 2 underflow					X	R/W	
		D1	**F8TU1**	8-bit timer 1 underflow					X	R/W	
		D0	**F8TU0**	8-bit timer 0 underflow					X	R/W	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Serial I/F interrupt factor flag register	**0040286 (B)**	D7–6	**–**	reserved	–				–	–	0 when being read.
		D5	**FSTX1**	SIF Ch.1 transmit buffer empty	1	Factor is generated	0	No factor is generated	X	R/W	
		D4	**FSRX1**	SIF Ch.1 receive buffer full					X	R/W	
		D3	**FSERR1**	SIF Ch.1 receive error					X	R/W	
		D2	**FSTX0**	SIF Ch.0 transmit buffer empty					X	R/W	
		D1	**FSRX0**	SIF Ch.0 receive buffer full					X	R/W	
		D0	**FSERR0**	SIF Ch.0 receive error					X	R/W	
Clock timer, A/D interrupt factor flag register	**0040287 (B)**	D7–2	**–**	reserved	–				–	–	0 when being read.
		D1	**FCTM**	Clock timer	1	Factor is generated	0	No factor is generated	X	R/W	
		D0	**FADE**	A/D converter					X	R/W	
K6, DMA, 16-bit timer 0 IDMA request register	**0040290 (B)**	D7	**R16TC01**	16-bit timer 01 comparison match	1	IDMA request	0	Interrupt request	X	R/W	
		D6	**R16TU01**	16-bit timer 01 underflow					X	R/W	
		D5	**RHDM1**	High-speed DMA Ch.1					X	R/W	
		D4	**RHDM0**	High-speed DMA Ch.0					X	R/W	
		D3	**RK64**	K64 input					X	R/W	
		D2	**RK65**	K65 input					X	R/W	
		D1	**RK66**	K66 input					X	R/W	
		D0	**RK67**	K67 input					X	R/W	
16-bit timer 1–4 IDMA request register	**0040291 (B)**	D7	**R16TC41**	16-bit timer 41 comparison match	1	IDMA request	0	Interrupt request	X	R/W	
		D6	**R16TU41**	16-bit timer 41 underflow					X	R/W	
		D5	**R16TC31**	16-bit timer 31 comparison match					X	R/W	
		D4	**R16TU31**	16-bit timer 31 underflow					X	R/W	
		D3	**R16TC21**	16-bit timer 21 comparison match					X	R/W	
		D2	**R16TU21**	16-bit timer 21 underflow					X	R/W	
		D1	**R16TC11**	16-bit timer 11 comparison match					X	R/W	
		D0	**R16TU11**	16-bit timer 11 underflow					X	R/W	
16-bit timer 5, 8-bit timer, serial I/F Ch.0 IDMA request register	**0040292 (B)**	D7	**RSTX0**	SIF Ch.0 transmit buffer empty	1	IDMA request	0	Interrupt request	X	R/W	
		D6	**RSRX0**	SIF Ch.0 receive buffer full					X	R/W	
		D5	**R8TU3**	8-bit timer 3 underflow					X	R/W	
		D4	**R8TU2**	8-bit timer 2 underflow					X	R/W	
		D3	**R8TU1**	8-bit timer 1 underflow					X	R/W	
		D2	**R8TU0**	8-bit timer 0 underflow					X	R/W	
		D1	**R16TC51**	16-bit timer 51 comparison match					X	R/W	
		D0	**R16TU51**	16-bit timer 51 underflow					X	R/W	
Serial I/F Ch.1, A/D IDMA request register	**0040293 (B)**	D7–3	**–**	reserved	–				–	–	
		D2	**RADE**	A/D converter	1	IDMA request	0	Interrupt request	X	R/W	
		D1	**RSTX1**	SIF Ch.1 transmit buffer empty					X	R/W	
		D0	**RSRX1**	SIF Ch.1 receive buffer full					X	R/W	
Interrupt factor flag reset method select register	**004029F (B)**	D7–1	**–**	reserved	–				–	–	
		D0	**RSTONLY**	Interrupt factor flag reset method selection	1	Reset only	0	RD/WR	1	R/W	
TTBR write protect register	**004812D (B)**	D7	**TBRP7**	TTBR register write protect	Writing 01011001(0x59) removes the TTBR (0x48134) write protection. Writing other data sets the write protection.				0	W	Undefined in read.
		D6	**TBRP6**						0		
		D5	**TBRP5**						0		
		D4	**TBRP4**						0		
		D3	**TBRP3**						0		
		D2	**TBRP2**						0		
		D1	**TBRP1**						0		
		D0	**TBRP0**						0		
TTBR low-order register	**0048134 (HW)**	DF	**TTBR15**	Trap table base address [15:10]					0	R/W	
		DE	**TTBR14**						0		
		DD	**TTBR13**						0		
		DC	**TTBR12**						0		
		DB	**TTBR11**						0		
		DA	**TTBR10**						0		
		D9	**TTBR09**	Trap table base address [9:0]	Fixed at 0				0	R	0 when being read. Writing 1 not allowed.
		D8	**TTBR08**						0		
		D7	**TTBR07**						0		
		D6	**TTBR06**						0		
		D5	**TTBR05**						0		
		D4	**TTBR04**						0		
		D3	**TTBR03**						0		
		D2	**TTBR02**						0		
		D1	**TTBR01**						0		
		D0	**TTBR00**						0		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
TTBR high-order register	**0048136 (HW)**	DF	**TTBR33**	Trap table base address [31:28]	Fixed at 0	0	R	0 when being read. Writing 1 not allowed.
		DE	**TTBR32**			0		
		DD	**TTBR31**			0		
		DC	**TTBR30**			0		
		DB	**TTBR2B**	Trap table base address [27:16]	The initial value is set according to the BTA3 pin status. BTA3 = "0": 0x0C0	←	R/W	
		DA	**TTBR2A**					
		D9	**TTBR29**					
		D8	**TTBR28**					
		D7	**TTBR27**					
		D6	**TTBR26**					
		D5	**TTBR25**					
		D4	**TTBR24**					
		D3	**TTBR23**					
		D2	**TTBR22**					
		D1	**TTBR21**					
		D0	**TTBR20**					

The following collectively explains the basic functions of each control register/bit. For details about individual interrupt systems and the contents classified by an interrupt factor, refer to the descriptions of the peripheral circuits in this manual.

Pxxx2–Pxxx0: Interrupt priority register

Set the priority levels of each interrupt system in the range of 0 to 7.
If this register is set below the IL value of the PSR, no interrupt is generated. The value of this register when initially reset is indeterminate.

Exxx: Interrupt enable register

Enable or disable interrupt generation to the CPU.

Write "1": Interrupt enabled
Write "0": Interrupt disabled
Read: Valid

Interrupts are enabled when the corresponding bits of this register are set to "1" and are disabled when the bits are set to "0".
For the interrupt factors used to request IDMA invocation or clear the standby mode, the corresponding interrupt enable register bit must be set for interrupt enable.
When initially reset, this register is set to "0" (interrupt disabled).

Fxxx: Interrupt factor flag

Indicate the status of interrupt factors generated.

When read

Read "1": Interrupt factor generated
Read "0": No interrupt factor generated

When written using the reset-only method (default)

Write "1": Factor flag is reset
Write "0": Invalid

When written using the read/write method

Write "1": Factor flag is set
Write "0": Factor flag is reset

The interrupt factor flag is set to "1" when an interrupt factor occurs in each peripheral circuit.
If the following conditions are met at this time, an interrupt is generated to the CPU:
1. The corresponding bit of the interrupt enable register is set to "1".
2. No other interrupt request of higher priority has occurred.
3. The IE bit of the PSR is set to "1" (interrupt enabled).
4. The corresponding interrupt priority register is set to a level higher than the CPU's interrupt level (IL).

When using an interrupt factor to request IDMA, note that even when the above conditions are met, no interrupt request to the CPU is generated for the interrupt factor that has occurred. If interrupts are enabled at the setting of IDMA, an interrupt is generated under the above conditions after the data transfer by IDMA is completed.

The interrupt factor flag is always set to "1" when an interrupt factor occurs no matter how the interrupt enable and interrupt priority registers are set.
In order for the next interrupt to be accepted after interrupt generation, the interrupt factor flag must be reset and the PSR must be set up again (by setting the IL below the level indicated by the interrupt priority register and setting the IE bit to "1" or executing the reti instruction).
The interrupt factor flag can only be reset by a write instruction in the software application. If the PSR is again set up to accept interrupts (or the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt may occur again. Note also that the value to be written to reset the flag is "1" when using the reset-only method (RSTONLY = "1") and "0" when using the read/write method (RSTONLY = "0"). Be careful not to confuse these two conditions.
The interrupt factor flag becomes indeterminate when initially reset, so be sure to reset the flag in the software application.

Rxxx: IDMA request register

Specify whether or not to invoke IDMA when an interrupt factor occurs.

Write "1": IDMA is requested
Write "0": Interrupt is requested
Read: Valid

If a bit of this register is set to "1", IDMA is invoked when the corresponding interrupt factor occurs and the programmed data transfer is performed. If the register bit is set to "0", regular interrupt processing is performed, without ever invoking IDMA.
For details about IDMA, refer to Section 9.2, "Intelligent DMA".
If interrupts are enabled on the IDMA side and the transfer counter reaches the terminal count of 0 after completion of DMA transfer, the IDMA request register is reset to "0" and an interrupt request for the interrupt factor that enabled IDMA invoking is generated.
The value of this register becomes indeterminate when initially reset, so be sure to reset it in the software application.

RSTONLY: Interrupt factor flag reset method selection (D0) / Interrupt factor flag reset method select register (0x4029F)

Select the method for resetting the interrupt factor flag.

Write "1": Reset-only method
Write "0": Read/write method
Read: Valid

With the reset-only method, the interrupt factor flag is reset by writing "1".
The interrupt factor flags for which "0" has been written can neither be set nor reset. Therefore, this method ensures that only a specific factor flag is reset. However, when using read-modify-write instructions (e.g., bset, bclr, or bnot), note that an interrupt factor flag that has been set to "1" is reset by writing. This method cannot be used to set any interrupt factor flag in the software application.
The read/write method is selected by writing "0" to RSTONLY. When this method is selected, interrupt factor flags can be read and written as for other registers. Therefore, the flag is reset by writing "0" and set by writing "1". In this case all factor flags for which "0" has been written are reset. Even in a read-modify-write operation, an interrupt factor can occur between read and write instructions, so be careful when using this method.
After an initial reset, RSTONLY is set to "1" (reset-only method).

TBRP7–TBRP0: TTBR register write protection ([D[7:0]) / TTBR write-protect register (0x4812D[B])

Remove write protection for the TTBR register.

Write 0x59:	Write protection is removed
Write not the above:	No operation (write protected)
Read:	Valid

Before writing to the TTBR register, set TBRP to "0x59" to remove the write protection. Then when data is written to the most significant byte (0x48137) of the TTBR, the register once again becomes write-protected.
After an initial reset, TBRP is set to "0x0" (write protected).

TTBR09–TTBR00: Trap table base address [9:0] (D[9:0]) / TTBR low-order register (0x48134[HW])
TTBR15–TTBR10: Trap table base address [15:10] (D[F:A]) / TTBR low-order register (0x48134[HW])
TTBR2B–TTBR20: Trap table base address [27:16] (D[B:0]) / TTBR high-order register (0x48136[HW])
TTBR33–TTBR30: Trap table base address [31:28] (D[F:C]) / TTBR high-order register (0x48136[HW])

Set the starting address of the trap table.
TTBR0 and TTBR3 are read-only registers and are fixed to "0". For this reason, the trap table starting address always begins with a 1KB boundary address.
The TTBR registers normally are write-protected to prevent them from being inadvertently rewritten. To remove this write protect function, another register, TBRP (D[F:8]) / TTBR write-protect register (0x4812D), is provided.
A write to the TTBR register is enabled by writing "0x59" to TBRP and is disabled back again by a write to the most significant byte of the TTBR register (0x48137). Consequently, writes to the TTBR register need to begin with the low-order half-word first. However, since occurrences of NMI and the like between writes of the low-order and high-order half-words cause malfunctions, it is recommended that the register be written in words.
After an initial reset, the TTBR register is set to the boot address that is determined by the BTA3 pin status (BTA3 = low: 0x0C00000).

8.6 Programming Notes

(1) In cases when an interrupt factor that is used for restarting from the standby mode has been set to invoke IDMA, IDMA is started up by the interrupt at its occurrence. In SLEEP mode, the high-speed (OSC3) oscillation circuit also starts operating. However, if an interrupt to be generated upon completion of IDMA is disabled at the setting of IDMA side, no interrupt request is signaled to the CPU. Therefore, the CPU remains idle until the next interrupt request is generated.

(2) As the E0C33000 Core CPU function, the IL allows interrupt levels to be set in the range of 0 to 15. However, since the interrupt priority register in the E0C33A104 consists of three bits, interrupt levels in each interrupt system can only be set for up to 8.

(3) When the reset-only method is used to reset the interrupt factor flag (by writing "1"), if a read-modify-write instruction (e.g., bset, bclr, or bnot) is executed, the other interrupt factor flags at the same address that have been set to "1" are reset by a write. This requires caution. In cases when the read/write method is used to reset the interrupt factor flag (by writing "0"), all factor flags for which "0" has been written are reset. When a read-modify-write operation is performed, an interrupt factor may occur between reads and writes, so be careful when using this method.

(4) After an initial reset, the interrupt factor flags, IDMA request registers, and interrupt priority registers all become indeterminate. To prevent unwanted interrupts or IDMA requests from being generated inadvertently, be sure to reset these flags and registers in the software application.

(5) To prevent another interrupt from being generated for the same factor again after generation of an interrupt, be sure to reset the interrupt factor flag before enabling interrupts and setting the PSR again or executing the reti instruction.

(6) When an interrupt is disabled using the interrupt enable bit (0x0040270–0x0040277) at the same time the corresponding interrupt factor occurs, the interrupt processing read a wrong vector from [TTBR+284(decimal)] (reserved vector number). Therefore the program does not jump to the correct interrupt service routine.
To avoid this problem, an interrupt enable bit in ITC should be reset by the following procedure:
1) Clear the IE bit in PSR.
2) Execute the NOP instruction
3) Reset the interrupt enable bit in ITC.

Example:

```
ld.w   %r0,%psr
ext    0x3ff
and    %r0,0x2f
ld.w   %psr,%r0   ; clear IE
nop
ld.b   [%r1],%r2  ; %r1=0x00040274, %r2=00000000
                  ; disable interrupt
```

9 DMA Controller

The E0C33A104 incorporates a DMA controller, thus providing support for two types of DMA transfers: high-speed DMA, in which DMA transfer between external devices is performed at high speed; and intelligent DMA in which transfer conditions and the like can be programmed. This chapter describes the functions of each DMA and how to control them.

9.1 High-Speed DMA

9.1.1 Functional Outline of High-Speed DMA

The E0C33A104 contains two channels of high-speed DMA that allows data transfers between external memory and an external device to be performed directly on the external system interface.

Since the control registers required for the DMA function are built into the chip, DMA requests for data transfer can be responded to instantaneously. In addition, data transfers that are normally accomplished by executing data read and write operations back-to-back are executed on the external bus collectively at one time, thus further speeding up the transfer operation.

Unlike intelligent DMA, which has transfer information in memory, this DMA method is not capable of elaborate programming of transfer conditions but allows data transfers to be performed in minimum cycles.

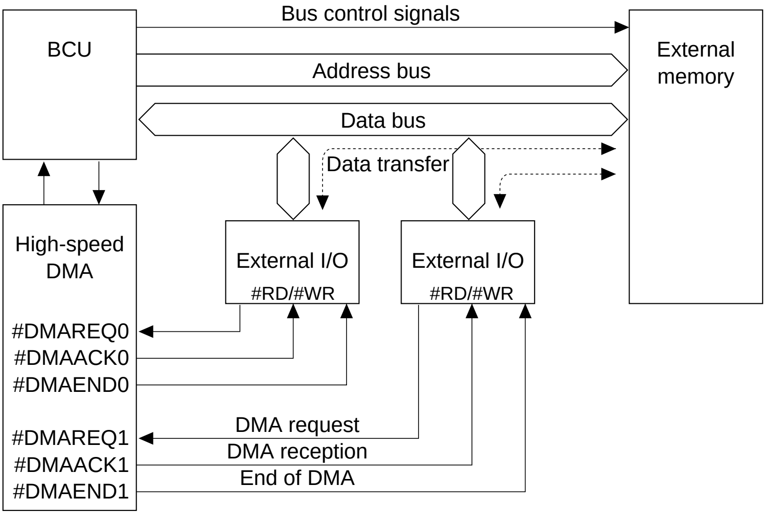


Figure 9.1.1 High-Speed DMA between External Device and External Memory

Note: Channels 0 and 1 are configured in the same way and have the same functionality. Signal and control bit names are assigned channel numbers 0 or 1 to distinguish them from other channels. In this manual, however, channel numbers 0 and 1 are designated with an "x" except where the two must be distinguished, as the explanation is the same for both channels.

9.1.2 I/O Pins of High-Speed DMA

Table 9.1.1 lists the I/O pins used for high-speed DMA.

Table 9.1.1 I/O Pins of High-Speed DMA

Pin name	Pin No.		I/O	Pull-up	Function	Function select bit
	QFP5-128	QFP15-128				
K50/ #DMAREQ0	58	55	I	Built-in	Input port / DMA request input for high-speed DMA Ch.0	CFK50(D0) / K5 function select register (0x402C0)
K51/ #DMAREQ1	54	51	I	Built-in	Input port / DMA request input for high-speed DMA Ch.1	CFP51(D1) / K5 function select register (0x402C0)
R82/ #DMAACK0	59	56	O	–	Output port / Acknowledge signal output of high-speed DMA Ch.0	CFR82(D2) / R8 function select register (0x402CC)
R83/ #DMAACK1	55	52	O	–	Output port / Acknowledge signal output of high-speed DMA Ch.1	CFR83(D3) / R8 function select register (0x402CC)
P15/EXCL40/ #DMAEND0	51	48	I/O	Built-in	I/O port / Timer 40 event counter input / End-of-transfer signal output of high-speed DMA Ch.0	CFP15(D5) / P1 function select register (0x402D4)
P16/EXCL50/ #DMAEND1	52	49	I/O	Built-in	I/O port / Timer 50 event counter input / End-of-transfer signal output of high-speed DMA Ch.1	CFP16(D6) / P1 function select register (0x402D4)

#DMAREQx (DMA request input pin)

This pin is used to input a DMA request signal from an external peripheral circuit. A falling edge of this input signal invokes high-speed DMA. One data transfer operation is performed by this trigger. #DMAREQ0 and #DMAREQ1 respectively are an input pin for channel 0 and channel 1.

#DMAACKx (DMA acknowledge signal output pin)

This signal is output to indicate that a DMA request has been acknowledged by the DMA controller.
The I/O device that is the source or destination of transfer outputs data to the external bus or takes in data from the external data synchronously with this signal.
#DMAACK0 and #DMAACK1 are input pins for channels 0 and 1 respectively.
For the output timing of the #DMAACKx signal, refer to Section 9.1.4.

#DMAENDx (End-of-transfer signal output pin)

This signal is output to indicate that the number of data transfer operations that is set in the control register have been completed. #DMAEND0 and #DMAEND1 are an input pins for channels 0 and 1 respectively.
For the output timing of the #DMAENDx signal, refer to Section 9.1.4.

Method for setting high-speed DMA I/O pins

As shown in Table 9.1.1, the pins used for high-speed DMA are shared with input ports, output ports, and I/O ports. At cold start, all of these are set as input, output, and I/O port pins (function select register = "0"). According to the high-speed DMA channels to be used, set the corresponding pin function select bit by writing "1". Since the function select register is a write-only register (read data is indeterminate), bit operation instructions (e.g., bset, bclr, or bnot) that accompany a read-modify-write operation cannot be used to rewire this register. Use an ordinary store instruction for this purpose. At hot start, the register retains the previous status before a reset.
In addition, setup of the #DMAENDx pin further requires setting the I/O port's I/O control bit IOC15 (D5) or IOC16 (D6) / P1 I/O control register (0x402D6) by writing "1" in order to direct the pin for output. If this pin is directed for input, it functions as a 16-bit programmable timer's event counter input and cannot be used to output the #DMAENDx signal. At cold start, this pin is set for input. At hot start, it retains the previous status.

Input pin pull-up resistors

Since the #DMAREQx pin is shared with an input port pin, it contains a pull-up resistor. The pull-up control bit for each pin can be used to determine whether this pull-up resistor is used.
K50 (for #DMAREQ0) pull-up control: KPU50 (D0) / K5 pull-up control register (0x402C4)
K51 (for #DMAREQ1) pull-up control: KPU51 (D1) / K5 pull-up control register (0x402C4)

At cold start, the pull-up control register is set to "0" so that the pull-up resistor is disconnected from the input line. If you want the #DMAREQx pin to be pulled up, set the KPU5x by writing "1". Note that since the pull-up control register is a write-only register (read data is indeterminate), bit operation instructions (e.g., bset, bclr, or bnot) cannot be used to rewrite this register. Use an ordinary store instruction for this purpose.
At hot start, the register retains the previous status before a reset.

9.1.3 Setting High-Speed DMA

In order for high-speed DMA to be used, the following settings must be made before initiating data transfers.

1. Set input and output pins
2. Set the external memory address
3. Set transfer conditions
4. Set the interrupt/IDMA
5. Enable DMA transfer

The contents of each setting is described below. For settings for the input and output pins, refer to the preceding section. For the settings for the interrupt/IDMA, refer to Section 9.1.5, "Interrupt Function of High-Speed DMA".

Note: Always make sure settings 2 to 5 are made when the DMA controller is idle (DMAEN = "0").

Setting external memory address

In high-speed DMA, data transfer is performed between the internal memory and an I/O device connected to the external system interface. Therefore, it is necessary to specify the external memory address that is the source or destination of transfer. For this address specification, use the registers shown below:

Channel 0 memory address
High-order 12 bits: D0HADR (D[11:0]) / High-speed DMA Ch.0 high-order memory address register (0x48206)
Low-order 16 bits: D0LADR (D[15:0]) / High-speed DMA Ch.0 low-order memory address register (0x48204)

Channel 1 memory address
High-order 12 bits: D1HADR (D[11:0]) / High-speed DMA Ch.1 high-order memory address register (0x48216)
Low-order 16 bits: D1LADR (D[15:0]) / High-speed DMA Ch.1 low-order memory address register (0x48214)

Notes:
- Always be sure to write or read to and from the memory address register in words (32 bits). If this register is written in bytes (8 bits) or half-words (16 bits), the data may become indeterminate. The same applies when reading this register too; the data that is read out is indeterminate.
- No address of the internal memory can be specified for the destination or source of transfer. The address specified here must always be that of an external memory connected to the external system interface. If an internal memory address is specified, DMA may not operate properly.
- When the DMA controller is enabled (DMAEN = "1"), no data can be written to the memory address register. Also, the data read from the register at this time is indeterminate. Always make sure the DMA controller is disabled (DMAEN = "0") before writing or reading to and from the memory address register.

For I/O devices, do not specify an address bus. The devices are accessed directly by the #DMAACKx signal, so it is unnecessary to specify an address.

Setting transfer conditions

- **Transfer data size**

  Choose the size of data to be transferred by one DMA operation using the control bits shown below.
  Channel 0 data size control: D0SIZ (D2) / High-speed DMA Ch.0 control register (0x4820A)
  Channel 1 data size control: D1SIZ (D2) / High-speed DMA Ch.1 control register (0x4821A)

  The half-word size (16 bits) is selected by writing "1" and the byte size (8 bits) is selected by writing "0".

- **Address increment/decrement**

  Specify whether the address needs to be incremented or decremented for each transfer performed by using the control bits shown below.
  Channel 0 address INC/DEC control: D0INC (D1) / High-speed DMA Ch.0 control register (0x4820A)
  Channel 1 address INC/DEC control: D1INC (D1) / High-speed DMA Ch.1 control register (0x4821A)

  INC (increment) is selected by writing "1" and DEC (decrement) is selected by writing "0".
  When one DMA transfer operation is completed, the content of the memory address register is incremented or decremented according to the data size that has been set by DxSIZ (by 2 when half-word is selected or by 1 when byte is selected).

- **Direction of transfer**

  Choose the direction of data transfer using the registers shown below.
  Channel 0 transfer direction control: D0DIR (D0) / High-speed DMA Ch.0 control register (0x4820A)
  Channel 1 transfer direction control: D1DIR (D0) / High-speed DMA Ch.1 control register (0x4821A)

  Memory write operations (data transfer from I/O device to memory) are specified by writing "1" and memory read operations (data transfer from memory to I/O device) are specified by writing "0".

- **Transfer counter**

  A DMA transfer that conforms to the above transfer conditions is performed only once by a trigger input to the #DMAREQx pin. Use the transfer counter to set the number of times a transfer needs to be repeated to send a series of data.
  Channel 0 transfer counter: D0LEN (D[15:0]) / High-speed DMA Ch.0 transfer counter register (0x0048208)
  Channel 1 transfer counter: D1LEN (D[15:0]) / High-speed DMA Ch.1 transfer counter register (0x0048218)

  When one data transfer operation is completed, the transfer counter is decremented by 1. When the counter reaches 0, a pulse indicating the completion of transfer is output to the #DMAENDx pin. An interrupt factor is generated simultaneously.

Note: The control registers used to set the transfer data size, address increment/decrement, and transfer direction can be accessed for a write instruction even during a DMA transfer. However, the contents of any changes made are not reflected until the next transfer is performed. A write to the transfer counter during a transfer is ignored and the data read from the counter during a transfer is indeterminate.

Enabling/disabling DMA transfer

The DMA controller is enabled by writing "1" to the DMA enable bit DMAEN (D0) / DMA enable register (0x48235) so that a DMA request input from the #DMAREQx pin can be accepted. However, the memory address and transfer conditions must always be set correctly before enabling a DMA transfer.
When DMAEN is set to "0", DMA requests from the #DMAREQx pin are no longer accepted.

DMAEN enables data transfers in channels 0 and 1 by high-speed DMA and those by intelligent DMA simultaneously.
If when using two channels of high-speed DMA you want data transfer to be enabled in only one channel, redirect the #DMAREQx pin on the channel you want to be disabled for transfer in order to inhibit a DMA request from being input from that pin to the DMA controller.
If you want a transfer by intelligent DMA to be disabled, use the interrupt controller's IDMA request register to inhibit IDMA from being invoked.

9.1.4 Operation of High-Speed DMA

When you write "1" is written to DMAEN after setting the memory address and transfer conditions, the DMA controller is activated standing by for input from the #DMAREQx pin.

The following explains the operation of high-speed DMA from when DMA is requested by an external I/O device to when the transfer is completed.

(1) DMA request from an external I/O device

When a low pulse is input from an external I/O device to the #DMAREQx pin, operation of high-speed DMA is triggered by the falling edge of the pulse.

(2) #DMAACKx signal output and bus operation

When the E0C33A104 accepts the DMA request, it outputs a low-level pulse from the #DMAACKx pin and starts bus operation for the external memory at the same time.

The contents of this bus operation are as follows:

- **Data transfer from I/O device to memory**

  The address that has been set in the memory address register is output to the address bus.
  A write operation is performed under the interface conditions set on the area to which the external memory at the destination of transfer belongs. The data bus is left floating.

  The external I/O device outputs the transfer data onto the data bus using the #DMAACKx signal as the read signal. The external memory takes in this data using the write signal.

- **Data transfer from memory to an I/O device**

  The address that has been set in the memory address register is output to the address bus.
  A read operation is performed under the interface conditions set on the area to which the external memory at the source of transfer belongs.

  The external memory outputs the transfer data onto the data bus using the read signal.
  The external I/O device takes in the data from the data bus using the #DMAACKx signal as the write signal.

If the transfer data size is 16 bits and the I/O device is an 8-bit device, two bus operations are performed. Otherwise, transfer is completed in one bus operation.

(3) Updating the memory address and transfer counter

When the bus operation in (2) is completed, the content of the address register is incremented or decremented by an amount equal to the transfer data size according to the settings of DxINC.

The transfer counter also is decremented.

(4) If the transfer counter is not 0

If after being decremented the transfer counter is not 0, the DMA controller returns to a standby status.

(5) If the transfer counter is 0

An end-of-transfer signal is output from the #DMAENDx pin indicating that a specified number of transfers has been completed. At the same time, an interrupt factor for the completion of high-speed DMA is generated.

Note that even when the transfer counter is 0, a new DMA request is accepted when generated and the counter is sequentially decremented from "0xFFFF" as long as the DMA controller remains active. Therefore, the transfer counter and the address register must be set back again using an end-of-transfer interrupt or the like. In this case, be sure to disable DMA transfer (DMAEN = "0") first before changing the counter and register.

High-speed DMA timing chart

(1) SRAM

Example: When 2 (RD)/1 (WR) wait cycles are inserted

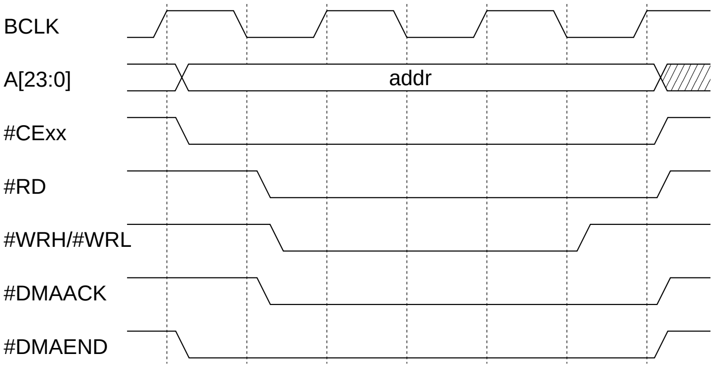


Figure 9.1.2 #DMAACK/#DMAEND Signal Output Timing (SRAM)

(2) Burst ROM

Example: When 4-consecutive-burst and 2-wait cycles are set during the first access

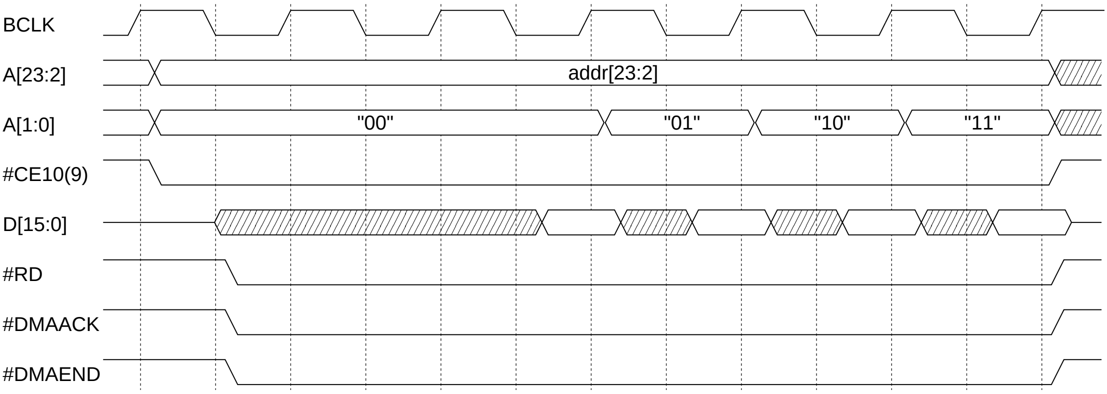


Figure 9.1.3 #DMAACK/#DMAEND Signal Output Timing (Burst ROM)

(3) DRAM

Example: Page mode, RAS: 1 cycle; CAS: 2 cycles; Precharge: 1 cycle

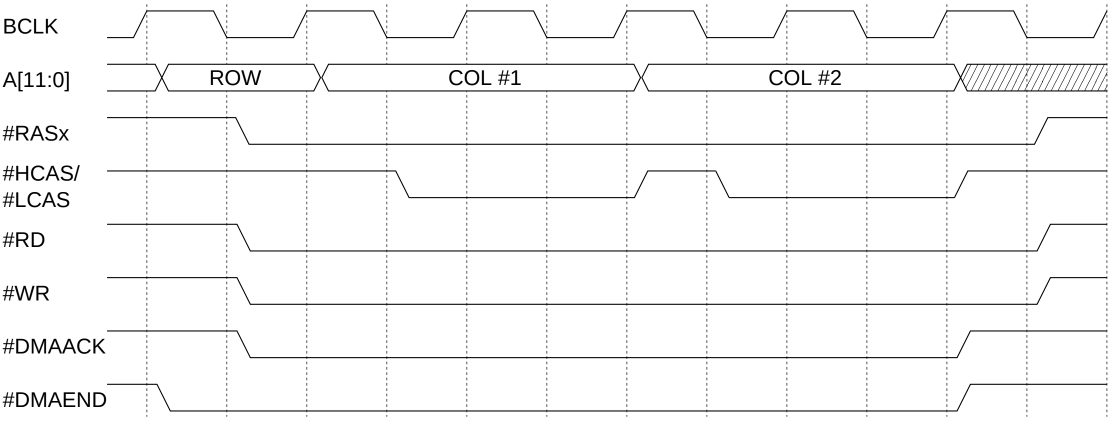


Figure 9.1.4 #DMAACK/#DMAEND Signal Output Timing (DRAM)

9.1.5 Interrupt Function of High-Speed DMA

The DMA controller allows you to generate an interrupt or invoke IDMA when the transfer counter in each high-speed DMA channel reaches 0.

Control registers of the interrupt controller

Table 9.1.2 shows the control registers of the interrupt controller that are provided for each channel.

Table 9.1.2 Control Registers of Interrupt Controller

Channel	Interrupt factor flag	Interrupt enable register	Interrupt priority register	IDMA request register
Ch.0	FHDM0(D0/0x40281)	EHDM0(D0/0x40271)	PDM[2:0](D[2:0]/0x40261)	RHDM0(D4/0x40290)
Ch.1	FHDM1(D1/0x40281)	EHDM1(D1/0x40271)		RHDM1(D5/0x40290)

The DMA controller sets the interrupt factor flag to "1" when the transfer counter reaches 0 after completing a series of high-speed DMA transfers. If the corresponding bit of the interrupt enable register is set to "1" at this time, an interrupt request is generated.

Interrupts can be disabled by leaving the interrupt enable register bit set to "0". The interrupt factor flag is always set to "1" when the data transfer in each channel is completed no matter what value the interrupt enable register bit is set to. (This is true even when it is set to "0".)

The interrupt priority register sets an interrupt priority level (0 to 7). An interrupt request to the CPU is accepted only when there is no other interrupt request of higher priority.

Furthermore, it is only when the PSR's IE bit = "1" (interrupt enable) and the set value of IL is smaller than the high-speed DMA interrupt level which is set in the interrupt priority register that the CPU actually accepts a high-speed DMA interrupt.

For details about the interrupt control register and for the device operation when an interrupt occurs, refer to Chapter 8, "Interrupt".

Intelligent DMA

Intelligent DMA (IDMA) can be invoked by an end-of-transfer interrupt factor of high-speed DMA.

The following shows the IDMA channels set in high-speed DMA:

	IDMA channel
Channel 0 end-of-transfer interrupt:	0x05
Channel 1 end-of-transfer interrupt:	0x06

Before IDMA can be invoked, the corresponding bit of the IDMA request register must be set to "1". Settings of transfer conditions on the IDMA side are also required.

If the IDMA request register bit is set to "1", IDMA is invoked by generation of an interrupt factor. No interrupt request is generated at this point in time. An interrupt request is generated after the transfer by IDMA is completed. IDMA can be set otherwise so that only IDMA transfer is performed and no interrupt is generated.

For details about data transfer by IDMA and for information on interrupt control after the completion of IDMA transfer, refer to Section 9.2, "Intelligent DMA".

Trap vector

The trap vector addresses for interrupt factors in each channel are set by default as follows:

	BTA3 = low
Channel 0 end-of-transfer interrupt:	0x0C00060
Channel 1 end-of-transfer interrupt:	0x0C00064

Note that the trap table base address can be modified using the TTBR registers (0x48134 to 0x48137).

9.1.6 I/O Memory of High-Speed DMA

Table 9.1.3 shows the control bits of high-speed DMA.

Table 9.1.3 Control Bits of High-Speed DMA

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
K5 function select register	**00402C0 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**CFK54**	K54 function selection	1	DA1	0	K54	0	W	Undefined in read.
		D3	**CFK53**	K53 function selection	1	DA0	0	K53	0	W	
		D2	**CFK52**	K52 function selection	1	#ADTRG	0	K52	0	W	
		D1	**CFK51**	K51 function selection	1	#DMAREQ1	0	K51	0	W	
		D0	**CFK50**	K50 function selection	1	#DMAREQ0	0	K50	0	W	
K5 pull-up control register	**00402C4 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**KPU54**	K54 pull-up control	1	Pulled up	0	No pull-up	0	W	Undefined in read.
		D3	**KPU53**	K53 pull-up control					0	W	
		D2	**KPU52**	K52 pull-up control					0	W	
		D1	**KPU51**	K51 pull-up control					0	W	
		D0	**KPU50**	K50 pull-up control					0	W	
R8 function select register	**00402CC (B)**	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**CFR83**	R83 function selection	1	#DMAACK1	0	R83	0	W	Undefined in read.
		D2	**CFR82**	R82 function selection	1	#DMAACK0	0	R82	0	W	
		D1	**CFR81**	R81 function selection	1	FOSC1	0	R81	0	W	
		D0	**–**	–	–				–	–	0 when being read.
P1 function select register	**00402D4 (B)**	D7	**–**	reserved	–				–	–	
		D6	**CFP16**	P16 function selection	1	EXCL50 #DMAEND1	0	P16	0	W	Undefined in read.
		D5	**CFP15**	P15 function selection	1	EXCL40 #DMAEND0	0	P15	0	W	
		D4	**CFP14**	P14 function selection	1	EXCL30 #BUSGET	0	P14	0	W	
		D3	**CFP13**	P13 function selection	1	EXCL20 T8UF3	0	P13	0	W	
		D2	**CFP12**	P12 function selection	1	EXCL10 T8UF2	0	P12	0	W	
		D1	**CFP11**	P11 function selection	1	EXCL01 T8UF1	0	P11	0	W	
		D0	**CFP10**	P10 function selection	1	EXCL00 T8UF0	0	P10	0	W	
P1 I/O control register	**00402D6 (B)**	D7	**–**	reserved	–				–	–	
		D6	**IOC16**	P16 I/O control	1	Output	0	Input	0	W	Undefined in read.
		D5	**IOC15**	P15 I/O control					0	W	
		D4	**IOC14**	P14 I/O control					0	W	
		D3	**IOC13**	P13 I/O control					0	W	
		D2	**IOC12**	P12 I/O control					0	W	
		D1	**IOC11**	P11 I/O control					0	W	
		D0	**IOC10**	P10 I/O control					0	W	
16-bit timer 0–1, DMA interrupt priority register	**0040261 (B)**	D7	**–**	reserved	–				–	–	0 when being read.
		D6	**P16T02**	16-bit timer 0–1 interrupt level	0 to 7				X	R/W	
		D5	**P16T01**						X		
		D4	**P16T00**						X		
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**PDM2**	IDMA, high-speed DMA interrupt level	0 to 7				X	R/W	
		D1	**PDM1**						X		
		D0	**PDM0**						X		
DMA interrupt enable register	**0040271 (B)**	D7–3	**–**	reserved	–				–	–	0 when being read.
		D2	**EIDMA**	IDMA	1	Enabled	0	Disabled	0	R/W	
		D1	**EHDM1**	High-speed DMA Ch.1					0	R/W	
		D0	**EHDM0**	High-speed DMA Ch.0					0	R/W	
DMA interrupt factor flag register	**0040281 (B)**	D7–3	**–**	reserved	–				–	–	0 when being read.
		D2	**FIDMA**	IDMA	1	Factor is generated	0	No factor is generated	X	R/W	
		D1	**FHDM1**	High-speed DMA Ch.1					X	R/W	
		D0	**FHDM0**	High-speed DMA Ch.0					X	R/W	
K6, DMA, 16-bit timer 0 IDMA request register	**0040290 (B)**	D7	**R16TC01**	16-bit timer 01 comparison match	1	IDMA request	0	Interrupt request	X	R/W	
		D6	**R16TU01**	16-bit timer 01 underflow					X	R/W	
		D5	**RHDM1**	High-speed DMA Ch.1					X	R/W	
		D4	**RHDM0**	High-speed DMA Ch.0					X	R/W	
		D3	**RK64**	K64 input					X	R/W	
		D2	**RK65**	K65 input					X	R/W	
		D1	**RK66**	K66 input					X	R/W	
		D0	**RK67**	K67 input					X	R/W	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
High-speed DMA Ch.0 low-order memory address register	**0048204 (HW)**	DF	**D0LADR15**	High-speed DMA Ch.0 memory address (low-order 16 bits)					X	R/W	
		DE	**D0LADR14**						X		
		DD	**D0LADR13**						X		
		DC	**D0LADR12**						X		
		DB	**D0LADR11**						X		
		DA	**D0LADR10**						X		
		D9	**D0LADR9**						X		
		D8	**D0LADR8**						X		
		D7	**D0LADR7**						X		
		D6	**D0LADR6**						X		
		D5	**D0LADR5**						X		
		D4	**D0LADR4**						X		
		D3	**D0LADR3**						X		
		D2	**D0LADR2**						X		
		D1	**D0LADR1**						X		
		D0	**D0LADR0**						X		
High-speed DMA Ch.0 high-order memory address register	**0048206 (HW)**	DF–C	–	reserved	–				–	–	Undefined in read.
		DB	**D0HADR11**	High-speed DMA Ch.0 memory address (high-order 12 bits)					X	R/W	
		DA	**D0HADR10**						X		
		D9	**D0HADR9**						X		
		D8	**D0HADR8**						X		
		D7	**D0HADR7**						X		
		D6	**D0HADR6**						X		
		D5	**D0HADR5**						X		
		D4	**D0HADR4**						X		
		D3	**D0HADR3**						X		
		D2	**D0HADR2**						X		
		D1	**D0HADR1**						X		
		D0	**D0HADR0**						X		
High-speed DMA Ch.0 transfer counter register	**0048208 (HW)**	DF	**D0LEN15**	High-speed DMA Ch.0 transfer counter					X	R/W	
		DE	**D0LEN14**						X		
		DD	**D0LEN13**						X		
		DC	**D0LEN12**						X		
		DB	**D0LEN11**						X		
		DA	**D0LEN10**						X		
		D9	**D0LEN9**						X		
		D8	**D0LEN8**						X		
		D7	**D0LEN7**						X		
		D6	**D0LEN6**						X		
		D5	**D0LEN5**						X		
		D4	**D0LEN4**						X		
		D3	**D0LEN3**						X		
		D2	**D0LEN2**						X		
		D1	**D0LEN1**						X		
		D0	**D0LEN0**						X		
High-speed DMA Ch.0 control register	**004820A (B)**	D7–3	–	reserved	–				–	–	Undefined in read.
		D2	**D0SIZ**	Ch.0 data size control	1	Half-word	0	Byte	0	R/W	
		D1	**D0INC**	Ch.0 address control	1	Inc	0	Dec	0	R/W	
		D0	**D0DIR**	Ch.0 transfer direction control	1	Memory write	0	Memory read	0	R/W	
High-speed DMA Ch.1 low-order memory address register	**0048214 (HW)**	DF	**D1LADR15**	High-speed DMA Ch.1 memory address (low-order 16 bits)					X	R/W	
		DE	**D1LADR14**						X		
		DD	**D1LADR13**						X		
		DC	**D1LADR12**						X		
		DB	**D1LADR11**						X		
		DA	**D1LADR10**						X		
		D9	**D1LADR9**						X		
		D8	**D1LADR8**						X		
		D7	**D1LADR7**						X		
		D6	**D1LADR6**						X		
		D5	**D1LADR5**						X		
		D4	**D1LADR4**						X		
		D3	**D1LADR3**						X		
		D2	**D1LADR2**						X		
		D1	**D1LADR1**						X		
		D0	**D1LADR0**						X		

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
High-speed DMA Ch.1 high-order memory address register	**0048216 (HW)**	DF–C	–	reserved	–				–	–	Undefined in read.
		DB	**D1HADR11**	High-speed DMA Ch.1 memory address (high-order 12 bits)					X	R/W	
		DA	**D1HADR10**						X		
		D9	**D1HADR9**						X		
		D8	**D1HADR8**						X		
		D7	**D1HADR7**						X		
		D6	**D1HADR6**						X		
		D5	**D1HADR5**						X		
		D4	**D1HADR4**						X		
		D3	**D1HADR3**						X		
		D2	**D1HADR2**						X		
		D1	**D1HADR1**						X		
		D0	**D1HADR0**						X		
High-speed DMA Ch.1 transfer counter register	**0048218 (HW)**	DF	**D1LEN15**	High-speed DMA Ch.1 transfer counter					X	R/W	
		DE	**D1LEN14**						X		
		DD	**D1LEN13**						X		
		DC	**D1LEN12**						X		
		DB	**D1LEN11**						X		
		DA	**D1LEN10**						X		
		D9	**D1LEN9**						X		
		D8	**D1LEN8**						X		
		D7	**D1LEN7**						X		
		D6	**D1LEN6**						X		
		D5	**D1LEN5**						X		
		D4	**D1LEN4**						X		
		D3	**D1LEN3**						X		
		D2	**D1LEN2**						X		
		D1	**D1LEN1**						X		
		D0	**D1LEN0**						X		
High-speed DMA Ch.1 control register	**004821A (B)**	D7–3	–	reserved	–				–	–	Undefined in read.
		D2	**D1SIZ**	Ch.1 data size control	1	Half-word	0	Byte	0	R/W	
		D1	**D1INC**	Ch.1 address control	1	Inc	0	Dec	0	R/W	
		D0	**D1DIR**	Ch.1 transfer direction control	1	Memory write	0	Memory read	0	R/W	
IDMA enable register	**0048235 (B)**	D7–1	–	reserved	–				–	–	
		D0	**DMAEN**	DMA enable	1	Enabled	0	Disabled	0	R/W	

CFK51–CFK50: K5[1:0] pin function selection (D[1:0]) / K5 function select register (0x402C0)

Set the #DMAREQx pin of high-speed DMA.

Write "1": #DMAREQx input
Write "0": Input port
Read: Invalid

To use high-speed DMA in channel 0, set the K50 pin for #DMAREQ0 by writing "1" to CFK50. Similarly, when using high-speed DMA in channel 1, set the K51 pin for #DMAREQ1 by writing "1" to CFK51.
If this bit is set to "0", the pin is set for an input port.
Since the CFK register is a write-only register (read data is indeterminate), bit operation instructions (e.g., bset, bclr, or bnot) cannot be used to rewrite this register. Use an ordinary store instruction.
At cold start, CFK5x is set to "0" (input port). At hot start, CFK5x retains the previous status before an initial reset.

KPU51–KPU50: K5[1:0] pull-up control (D[1:0]) / K5 pull-up control register (0x402C4)

Control the pull-up resistor of each input pin.

Write "1": Pull-up resistor enabled
Write "0": Pull-up resistor unavailable
Read: Invalid

When a bit in the pull-up register is set to "1", the pull-up resistor included in the corresponding pin is enabled and the pin is pulled up. If a bit in the register is set to "0", the corresponding pin is not pulled up.
To pull up the #DMAREQx pin, set the corresponding bit of the KPU register to "1".
Since the KPU register is a write-only register (read data is indeterminate), bit operation instructions (e.g., bset, bclr, or bnot) cannot be used to rewrite this register. Use an ordinary store instruction.
At cold start, KPU5x is set to "0" (pull-up resistor unavailable). At hot start, KPU5x retains the previous status before an initial reset.

CFR83–CFR82: R8[3:2] pin function selection (D[3:2]) / R8 function select register (0x402CC)

Set the #DMAACKx pin of high-speed DMA.

Write "1":	#DMAACKx output
Write "0":	Output port
Read:	Invalid

To use high-speed DMA in channel 0, set the R82 pin for #DMAACK0 by writing "1" to CFR82. Similarly, when using high-speed DMA in channel 1, set the R83 pin for #DMAACK1 by writing "1" to CFR83.
If CFR8x is set to "0", the pin is set for an output port.
Since the CFR register is a write-only register (read data is indeterminate), bit operation instructions (e.g., bset, bclr, or bnot) cannot be used to rewrite this register. Use an ordinary store instruction.
At cold start, CFR8x is set to "0" (output port). At hot start, CFR8x retains the previous status before an initial reset.

CFP16–CFP15: P1[6:5] pin function selection (D[6:5]) / P1 function select register (0x402D4)

Set the #DMAENDx pin of high-speed DMA.

Write "1":	#DMAENDx output
Write "0":	I/O port
Read:	Invalid

To use high-speed DMA in channel 0, set the P15 pin for #DMAEND0 by writing "1" to CFP15. Similarly, when using high-speed DMA in channel 1, set the P16 pin for #DMAEND1 by writing "1" to CFP16. Furthermore, direct these pins for output by writing "1" to the corresponding I/O control register.
If CFP1x is set to "0", the pin is set for an I/O port.
Since the CFP register is a write-only register (read data is indeterminate), bit operation instructions (e.g., bset, bclr, or bnot) cannot be used to rewrite this register. Use an ordinary store instruction.
At cold start, CFP1x is set to "0" (I/O port). At hot start, CFP1x retains the previous status before an initial reset.

IOC16–IOC15: P1[6:5] port I/O control (D[6:5]) / P1 I/O control register (0x402D6)

Direct the I/O port for input or output.

Write "1":	Output mode
Write "0":	Input mode
Read:	Invalid

To use the #DMAEND0 pin (channel 0), direct the pin for output by writing "1" to IOC15; to use the #DMAEND1 pin (channel 1), direct the pin for output by writing "1" to IOC16. If these pins are set for input, the P15 and P16 pins do not function as the #DMAENDx output pins even when CFP15 and CFP16 are set to "1".
Since the IOC register is a write-only register (read data is indeterminate), bit operation instructions (e.g., bset, bclr, or bnot) cannot be used to rewrite this register. Use an ordinary store instruction.
At cold start, IOC1x is set to "0" (input mode). At hot start, IOC1x retains the previous state before an initial reset.

DMAEN: DMA enable (D0) / DMA enable register (0x48235)

Enable a DMA transfer.

Write "1":	Enabled
Write "0":	Disabled
Read:	Valid

A data transfer by high-speed DMA or intelligent DMA (IDMA) is enabled by writing "1" to this bit.
High-speed DMA is placed in a state ready to accept a DMA request from the #DMAREQx pin.
DMA transfer is disabled by writing "0" to this bit.
Be sure to disable DMA transfers (DMAEN = "0") before setting the memory address and transfer conditions.
At initial reset, DMAEN is set to "0" (disabled).

D0LADR[15:0]: Ch.0 memory address A[16:0]
(D[F:0]) / High-speed DMA Ch.0 low-order memory address register (0x48204[HW])
D0HADR[11:0]: Ch.0 memory address A[27:16]
(D[B:0]) / High-speed DMA Ch.0 high-order memory address register (0x48206[HW])
D1LADR[15:0]: Ch.1 memory address A[16:0]
(D[F:0]) / High-speed DMA Ch.1 low-order memory address register (0x48214[HW])
D1HADR[11:0]: Ch.1 memory address A[27:16]
(D[B:0]) / High-speed DMA Ch.1 high-order memory address register (0x48216[HW])

Specify the external memory address at the destination or source of transfer.
Use DxLADR to set the 16 low-order bits of the address and DxHADR to set the 12 high-order bits.
Make sure the address is written and read in words (32 bits) collectively for the high-order and low-order bits. If the address is written and read in bytes (8 bits) or half-words (16 bits), the written or read data becomes indeterminate.
Furthermore, be sure to disable DMA transfers (DMAEN = "0") before writing or reading to and from these registers.
These registers are incremented or decremented (as set by DxINC) according to the transfer data size each time a DMA transfer in the corresponding channel is performed.
At initial reset, these registers are not initialized.

D0LEN[15:0]: Ch.0 transfer counter (D[F:0]) / High-speed DMA Ch.0 transfer counter register (0x48208[HW])
D1LEN[15:0]: Ch.1 transfer counter (D[F:0]) / High-speed DMA Ch.1 transfer counter register (0x48218[HW])

Set the data transfer count.
This counter is decremented each time a DMA transfer in the corresponding channel is performed. When the counter reaches 0, an end-of-transfer signal is output from the #DMAENDx pin. An interrupt factor is generated at the same time.
Even when the counter is 0, a DMA request is accepted and the counter is decremented to "0xFFFF".
Be sure to disable DMA transfers (DMAEN = "0") before writing and reading to and from the counter.
At initial reset, these counters are not initialized.

D0SIZ: Ch.0 data size control (D2) / High-speed DMA Ch.0 control register (0x4820A)
D1SIZ: Ch.1 data size control (D2) / High-speed DMA Ch.1 control register (0x4821A)

Select the data size to be transferred.

Write "1":	Half-word (16 bits)
Write "0":	Byte (8 bits)
Read:	Valid

The transfer data size is set to 16 bits by writing "1" to DxSIZ and set to 8 bits by writing "0".
If DxSIZ is rewritten during a DMA transfer, the transfer being executed is unaffected by this change. The altered content takes effect beginning with the next transfer performed.
At initial reset, DxSIZ is set to "0" (8 bits).

D0INC: Ch.0 address Inc/Dec control (D1) / High-speed DMA Ch.0 control register (0x4820A)
D1INC: Ch.1 address Inc/Dec control (D1) / High-speed DMA Ch.1 control register (0x4821A)

Control the incrementing or decrementing of the memory address.

Write "1":	Incremented
Write "0":	Decremented
Read:	Valid

If DxINC is set to "1", the memory address is incremented by an amount equal to the data size set by DxSIZ each time one data transfer operation is completed. If DxINC is set to "0", the memory address is decremented.
If DxINC is rewritten during a DMA transfer, the transfer being executed is unaffected by this change. The altered content takes effect beginning with the next transfer performed.
At initial reset, DxINC is set to "0" (decremented).

D0DIR: Ch.0 transfer direction control (D0) / High-speed DMA Ch.0 control register (0x4820A)
D1DIR: Ch.1 transfer direction control (D0) / High-speed DMA Ch.1 control register (0x4821A)

Control the direction of data transfer.

Write "1":	Memory write (I/O to memory)
Write "0":	Memory read (memory to I/O)
Read:	Valid

Data transfer from an external I/O device to external memory is performed by writing "1" to DxDIR. Data transfer from external memory to an external I/O is performed by writing "0".
If DxDIR is rewritten during a DMA transfer, the transfer being executed is unaffected by this change. The altered content takes effect beginning with the next transfer performed.
At initial reset, DxDIR is set to "0" (memory to I/O).

PDM2–PDM0: DMA interrupt level (D[2:0]) / 16-bit timer 0–1, DMA interrupt priority register (0x40261)

Set the priority level of an end-of-DMA interrupt in the range of 0 to 7.
At initial reset, this register becomes indeterminate.

EHDM0: Ch.0 interrupt enable (D0) / DMA interrupt enable register (0x40271)
EHDM1: Ch.1 interrupt enable (D1) / DMA interrupt enable register (0x40271)

Enable or disable interrupt generation to the CPU.

Write "1":	Interrupt enabled
Write "0":	Interrupt disabled
Read:	Valid

EHDM0 and EHDM1 are the interrupt enable bits for high-speed DMA channels 0 and 1, respectively. Interrupts are enabled when EHDMx is set to "1" and disabled when EHDMx is set to "0".
At initial reset, EHDMx is set to "0" (interrupts are disabled).

FHDM0: Ch.0 interrupt factor flag (D0) / DMA interrupt factor flag register (0x40281)
FHDM1: Ch.1 interrupt factor flag (D1) / DMA interrupt factor flag register (0x40281)

Indicate the occurrence status of high-speed DMA interrupt factor.

When read

Read "1":	Interrupt factor generated
Read "0":	No interrupt factor generated

When written using the reset-only method (default)

Write "1":	Factor flag is reset
Write "0":	Invalid

When written using the read/write method

Write "1":	Factor flag is set
Write "0":	Factor flag is reset

FHDM0 and FHDM1 are the Interrupt factor flags for high-speed DMA channels 0 and 1, respectively. These flags are set to "1" when the transfer counter reaches 0.
An interrupt to the CPU is generated if the following conditions are met at this time:
1. The corresponding interrupt enable register is set to "1".
2. No other interrupt request of higher priority is generated.
3. The IE bit of the PSR is set to "1" (interrupt enable).
4. The corresponding interrupt priority register is set to a level higher than the CPU's interrupt level (IL).
When using an interrupt factor to request IDMA, note that even when the above conditions are met, no interrupt request to the CPU is generated for the interrupt factor that has occurred. If interrupts are enabled at the setting of the IDMA side, an interrupt is generated under the above conditions after the data transfer by IDMA is completed.
The Interrupt factor flag is always set to "1" when an interrupt factor occurs no matter how the interrupt enable and interrupt priority registers are set.

In order for the next interrupt to be accepted after interrupt generation, the Interrupt factor flag must be reset and the PSR must be set up again (by setting the IL below the level indicated by the interrupt priority register and setting the IE bit to "1" or executing the reti instruction).
The interrupt factor flag can only be reset by a write instruction in the software application. If the PSR is again set up to accept interrupts (or the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt may occur again. Note also that the value to be written to reset the flag is "1" when using the reset-only method (RSTONLY = "1") and "0" when using the read/write method (RSTONLY = "0"). Be careful not to confuse these two cases.
The FHDMx flag becomes indeterminate when initially reset, so be sure to reset the flag in the software application.

RHDM0: Ch.0 IDMA request (D4) / K6, DMA, 16-bit timer 0 DMA request register (0x40290)
RHDM1: Ch.1 IDMA request (D5) / K6, DMA, 16-bit timer 0 DMA request register (0x40290)

Specify whether IDMA need to be invoked when an interrupt factor occurs.

Write "1":	IDMA request
Write "0":	Interrupt request
Read:	Valid

RHDM0 and RHDM1 are the IDMA request registers for high-speed DMA channels 0 and 1, respectively. If the register is set to "1", IDMA is invoked when an interrupt factor occurs, thus performing a programmed data transfer. If the register is set to "0", regular interrupt processing is performed without ever invoking IDMA.
For details about IDMA, refer to Section 9.2, "Intelligent DMA".
These registers become indeterminate when initially reset, so be sure to initialize the registers in the software application.

9.1.7 Programming Notes

(1) When setting the memory address and transfer conditions, always make sure the DMA controller is inactive (DMAEN = "0").

(2) Make sure the memory address register is always accessed for read and write operations in words (32 bits). If written in bytes (8 bits) or half-words (16 bits), the data becomes indeterminate. Similarly, when the register is read, indeterminate data is read out.

(3) No address of the internal memory can be specified for the destination or source of transfer. The address specified here must always be that of an external memory that is connected to the external system interface. If an internal memory address is specified, DMA may not operate properly.

(4) Since the pin function select registers (CFK, CFR, CFP), I/O control register (IOC), and pull-up control register (KPU) are write-only registers (read data is indeterminate), bit operation instructions (e.g., bset, bclr, or bnot) cannot be used to rewrite these registers. Use an ordinary store instruction.

(5) After an initial reset, the interrupt factor flag (FHDMx) and IDMA request register (RHDMx) become indeterminate. Always be sure to reset the flag and register to prevent interrupts or IDMA requests from being generated inadvertently.

(6) To prevent an interrupt from being generated repeatedly for the same cause, be sure to reset the interrupt factor flag before setting up the PSR again or executing the reti instruction.

(7) High-speed DMA is given higher priority over IDMA (intelligent DMA) and the CPU. However, since high-speed DMA and IDMA share the same circuit, high-speed DMA cannot gain the bus ownership while an IDMA transfer is under way. Requests for high-speed DMA invocation that have occurred during an IDMA transfer are kept pending until the IDMA transfer is completed.
A request for IDMA invocation or an interrupt request that has occurred during a high-speed DMA transfer are accepted after completion of the high-speed DMA transfer.

9.2 Intelligent DMA

9.2.1 Functional Outline of Intelligent DMA

The E0C33A104 contains an intelligent DMA (IDMA), a function that allows control information to be programmed in RAM. Up to 128 channels can be programmed, including 27 channels that are invoked by an interrupt factor that occurs in some internal peripheral circuit.
Although an additional overhead for loading and storing control information in RAM may be incurred, this intelligent DMA supports such functions as successive transfers, block transfers, and linking to another IDMA.

IDMA is invoked by an interrupt factor that occurs in some internal peripheral circuit or a software trigger, thereby performing a data transfer according to the control information in RAM. When the transfer is completed, IDMA can generate an interrupt or invoke another IDMA according to link settings.

9.2.2 Programming Control Information

The intelligent DMA operates according to the control information prepared in RAM. The control information can be stored in either internal RAM or external RAM should the necessary area be allocated.
The control information is 3 words (12 bytes) per channel in size, and must be located at contiguous addresses beginning with the base address that is set in the software application as the starting address of channel 0.
Consequently, an area of 384 words (1,536 bytes) in RAM is required in order for all of 128 channels to be used.

The following explains how to set the base address and the contents of control information. Before using IDMA, make each the settings described below.

Note: Be sure to disable DMA transfers (DMAEN = "0") before setting the base address and control information.

Setting the base address

Set the starting address of control information (starting address of channel 0) in the IDMA base address register.
16 low-order bits: DBASEL[15:0](D[15:0]) / IDMA base address low-order register (0x48230)
12 high-order bits: DBASEH[11:0](D[11:0]) / IDMA base address high-order register (0x48232)

When initially reset, the base address is set to 0x0C003A0.

Notes:
- The address you set in the IDMA base address register must always be a word (32-bit) boundary address.
- The IDMA base address registers cannot be accessed in bytes for read and write. The registers must be accessed in words for read/write operations to address 0x48230, and in half-words for read/write operations to addresses 0x48230 and 0x48232. Write operations in half-words must be performed in order of 0x48230 and 0x48232. Read operations in half-words may be performed in any order.
- Writes to the IDMA base address register during a DMA transfer are ignored. When the register is read during a DMA transfer, the read data is indeterminate.

Control information

Write the control information for the IDMA channels used to RAM.
The addresses at which the control information of each channel is placed are determined by the base address and a channel number.

Starting address of channel = base address + (channel number × 12 [bytes])

The contents of control information (3 words) in each channel are shown in the table below.

Table 9.2.1 IDMA Control Information

Word	Bit	Name	Function
1st	D31	LNKEN	IDMA link enable "1" = Enabled, "0" = Disabled
	D30–24	LNKCHN[6:0]	IDMA link field
	D23–8	TC[15:0]	Transfer counter (block transfer mode) Transfer counter - high-order 16 bits (single or successive transfer mode)
	D7–0	BLKLEN[7:0]	Block size (block transfer mode) Transfer counter - low-order 8 bits (single or successive transfer mode)
2nd	D31	DINTEN	End-of-transfer interrupt enable "1" = Enabled, "0" = Disabled
	D30	DATSIZ	Data size control "1" = Half-word, "0" = Byte
	D29–28	SRINC[1:0]	Source address control SRINC1 SRINC0 Setting contents 1 1 Address incremented (In block transfer mode, the transfer address is updated without reset using the initial value.) 1 0 Address incremented (In block transfer mode, the transfer address is updated with the initial value.) 0 1 Address decremented (In block transfer mode, the transfer address is updated without reset using the initial value.) 0 0 Address fixed
	D27–0	SRADR[27:0]	Source address
3rd	D31–30	DMOD[1:0]	Transfer mode (Do not set to "11".) DMOD1 DMOD0 Setting contents 1 0 Block transfer mode 0 1 Successive transfer mode 0 0 Single transfer mode
	D29–28	DSINC[1:0]	Destination address control DSINC1 DSINC0 Setting contents 1 1 Address incremented (In block transfer mode, the transfer address is updated without reset using the initial value.) 1 0 Address incremented (In block transfer mode, the transfer address is updated with the initial value.) 0 1 Address decremented (In block transfer mode, the transfer address is updated without reset using the initial value.) 0 0 Address fixed
	D27–0	DSADR[27:0]	Destination address

LNKEN: IDMA link enable (D31/1st Word)

If this bit remains set (= "1"), the IDMA channel that is set in the IDMA link field is invoked after the completion of a DMA transfer in this channel. DMA transfers in multiple channels can be performed successively by merely triggering the first channel to be executed. There is no limit to the number of channels linked. Set this link in order of the IDMA channels you want to be executed.

If this bit is "0", IDMA is completed by merely executing a DMA transfer in this channel.

LNKCHN[6:0]: IDMA link field (D[30:24]/1st Word)

If you want IDMA to be linked, set the channel numbers (0 to 127) to be executed next.

The data in this field is valid only when LINKEN = "1".

TC[15:0]: Transfer counter (D[23:8]/1st Word)

In block transfer mode, a transfer count can be specified using up to 16 bits. Set this value here. In single transfer and successive transfer modes, a transfer count can be specified using up to 24 bits. Set a 16-bit high-order value here.

BLKLEN[7:0]: Block size/transfer counter (D[7:0]/1st Word)

In block transfer mode, set the size of a block that is transferred in one operation (in units of DATSIZ). In single transfer and successive transfer modes, set an 8-bit low-order value for the transfer count here.

Note: The transfer count and block size thus set are decremented according to the transfers performed. If the transfer count or block size is set to 0, it is decremented to all Fs by the first transfer performed. This means that you have set the maximum value that is determined by the number of bits available.

DINTEN: End-of-transfer interrupt enable (D31/2nd Word)

If this bit is left set (= "1"), when the transfer counter reaches 0, an interrupt request to the CPU is generated based on the interrupt factor flag by which IDMA has been invoked.
If this bit is "0", no interrupt request to the CPU is generated even when the transfer counter has reached 0.

DATSIZ: Data size control (D30/2nd Word)

Set the unit size of data to be transferred.
A half-word size (16 bits) is assumed if this bit is "1" and a byte size (8 bits) is assumed if this bit is "0".

SRINC[1:0]: Source address control (D[29:28]/2nd Word)

Set the source address updating format.
If the format is set for "address fixed" (00), the source address is not changed by a data transfer performed. Even when transferring multiple data, the transfer data is always read from the same address.
If the format is set for "address increment" (11 or 10) in single and successive transfer modes, the source address is incremented by an amount equal to the data size set by DATSIZ when one data transfer is completed. If the format is set for "address decrement" (01), the source address is decremented in the same way.
In block transfer mode too, the source address is incremented or decremented when one data unit is transferred. However, if the set format is "10", the source address that has been incremented during a block transfer recycles back to the initial value when the block transfer is completed.

SRADR[27:0]: Source address (D[27:0]/2nd Word)

Use these bits to set the starting address at the source of transfer. The content set here is updated according to the setting of SRINC.

DMOD[1:0]: Transfer mode (D[31:30]/3rd Word)

Use these bits to set the desired transfer mode.
The transfer modes are outlined below (to be detailed later):

- **Single transfer mode (00)**

  In this mode, a transfer operation invoked by one trigger is completed after transferring one unit of data of the size set by DATSIZ. If data transfer need to be performed a number of times as set by the transfer counter, an equal number of triggers are required.

- **Successive transfer mode (01)**

  In this mode, data transfer operations are performed by one trigger a number of times as set by the transfer counter. The transfer counter is decremented to 0 each time data is transferred.

- **Block transfer mode (10)**

  In this mode, a transfer operation invoked by one trigger is completed after transferring one block of data of the size set by BLKELN. If a block transfer need to be performed a number of times as set by the transfer counter, an equal number of triggers are required.

DSINC[1:0]: Destination address control (D[29:28]/3rd Word)

Set the destination address update format.

If the format is set for "address fixed" (00), the destination address is not changed by the performance of a data transfer operation. Even when transferring multiple data, the transfer data is always written to the same address.

If the format is set for "address increment" (11 or 10) in single and successive transfer modes, the destination address is incremented by an amount equal to the data size set by DATSIZ when one data transfer is completed. If the format is set for "address decrement" (01), the destination address is decremented in the same way.

In block transfer mode as well, the destination address is incremented or decremented when one data unit is transferred. However, if the set format is "10", the destination address that has been incremented during a block transfer recycles back to the initial value when the block transfer is completed.

DSADR[27:0]: Destination address (D[27:0]/3rd Word)

Use these bits to set the starting address at the destination of transfer. The content set here is updated according to the setting of DSINC.

Since the control information is placed in RAM, it can be rewritten. However, before rewriting the content of this information, make sure that no DMA transfer is generated in the channel whose information you are going to rewrite.

9.2.3 IDMA Invocation

The triggers by which IDMA is invoked have the following three causes:

1. Interrupt factor in an internal peripheral circuit
2. Trigger in the software application
3. Link setting

Enabling/disabling DMA transfer

The DMA controller is enabled by writing "1" to the DMA enable bit DMAEN (D0) / DMA enable register (0x48235), and is ready to accept the triggers described above. However, before enabling a DMA transfer, be sure to set the base address and the control information for the channel to be invoked correctly. If DMAEN is set to "0", no IDMA invocation request is accepted.

DMAEN enables data transfers by IDMA and high-speed DMA simultaneously.

If you want a data transfer by high-speed DMA to be disabled, redirect the #DMAREQx pin on the channel you want to be disabled for input in order to inhibit a DMA request generated by an external I/O device from being input from that pin to the DMA controller. If you want high-speed DMA to be enabled and IDMA to be disabled, set the interrupt controller so that it will not generate IDMA.

IDMA invocation by an interrupt factor in internal peripheral circuits

Some internal peripheral circuits that have an interrupt generating function can invoke IDMA by an interrupt factor in that circuit. The IDMA channel numbers corresponding to such IDMA invocation are predetermined. The relationship between the interrupt factors that have this function and the IDMA channels is shown in Table 9.2.2.

Table 9.2.2 Interrupt Factors Used to Invoke IDMA

Peripheral circuit	Interrupt factor	IDMA Ch.	IDMA request register
Input port	K67 port input	1	RK67(D0/0x40290)
	K66 port input	2	RK66(D1/0x40290)
	K65 port input	3	RK65(D2/0x40290)
	K64 port input	4	RK64(D3/0x40290)
High-speed DMA	Ch.0, end of transfer	5	RHDM0(D4/0x40290)
	Ch.1, end of transfer	6	RHDM1(D5/0x40290)
16-bit programmable timer	Timer 01 underflow	7	R16TU01(D6/0x40290)
	Timer 01 compare match	8	R16TC01(D7/0x40290)
	Timer 11 underflow	9	R16TU11(D0/0x40291)
	Timer 11 compare match	10	R16TC11(D1/0x40291)
	Timer 21 underflow	11	R16TU21(D2/0x40291)
	Timer 21 compare match	12	R16TC21(D3/0x40291)
	Timer 31 underflow	13	R16TU31(D4/0x40291)
	Timer 31 compare match	14	R16TC31(D5/0x40291)
	Timer 41 underflow	15	R16TU41(D6/0x40291)
	Timer 41 compare match	16	R16TC41(D7/0x40291)
	Timer 51 underflow	17	R16TU51(D0/0x40292)
	Timer 51 compare match	18	R16TC51(D1/0x40292)
8-bit programmable timer	Timer 0 underflow	19	R8TU0(D2/0x40292)
	Timer 1 underflow	20	R8TU1(D3/0x40292)
	Timer 2 underflow	21	R8TU2(D4/0x40292)
	Timer 3 underflow	22	R8TU3(D5/0x40292)
Serial interface	Ch.0 receive buffer full	23	RSRX0(D6/0x40292)
	Ch.0 transmit buffer empty	24	RSTX0(D7/0x40292)
	Ch.1 receive buffer full	25	RSRX1(D0/0x40293)
	Ch.1 transmit buffer empty	26	RSTX1(D1/0x40293)
A/D converter	End of A/D conversion	27	RADE(D2/0x40293)

These interrupt factors are used in common for interrupt requests and IDMA invocation requests.
To invoke IDMA upon the occurrence of an interrupt factor, set the corresponding bit of the IDMA request register shown in the table by writing "1". Then when an interrupt factor occurs, an interrupt request to the CPU is kept pending and the corresponding IDMA channel is invoked.
The interrupt factor flag that has been set to "1" remains set until the DMA transfer invoked by it is completed. If the following two conditions are met when one DMA transfer is completed, an interrupt request is generated without resetting the interrupt factor flag.

- The transfer counter has reached 0.
- DINTEN in control information is set to "1" (interrupt enabled).

In this case, the IDMA request register is cleared to "0". Therefore, if IDMA needs to be invoked when an interrupt factor occurs next time, this register must be set up again. To prevent unwanted IDMA requests from being generated, this setting must be performed before enabling interrupts and after resetting the interrupt factor flag.

If the above conditions are not met, the interrupt factor flag is reset when the DMA transfer is completed, so that no interrupt is generated. In this case, the IDMA request register bit is not cleared and remains set to "1".

If the IDMA request register bit is left reset to "0", the relevant interrupt factor generates an interrupt request and not a IDMA request.

Even when using an interrupt factor to request IDMA invocation, the corresponding interrupt enable register bit must be set to "1" (interrupt enabled). If this bit is set to "0", neither an interrupt nor an IDMA invocation request is generated.

If IDMA has been invoked by an interrupt factor, an interrupt upon completion of transfer by IDMA is not generated even when DINTEN is set for interrupt enable.

IDMA invocation by a trigger in the software application

All IDMA channels for which control information is set, including those corresponding to interrupt factors described above, can be invoked by a trigger in the software application.

The following bits are used for this control:

IDMA channel number set-up:	DCHN[6:0] (D[6:0]) / IDMA start register (0x48234)
IDMA start control:	DSTART (D7) / IDMA start register (0x48234)

When the IDMA channel number to be invoked (0 to 127) is written to DCHN and DSTART is set to "1", the specified IDMA channel starts a DMA transfer.

DSTART remains set (= "1") during a DMA transfer and is reset to "0" in hardware when one DMA transfer operation is completed.

Do not modify these bits during a DMA transfer.

If DINTEN is set to "1" (interrupt enabled), an interrupt factor for the completion of IDMA transfer is generated when one DMA transfer is completed.

IDMA invocation by link setting

If LNKEN in the control information is set to "1" (link enabled), the IDMA channel that is set in the IDMA link field "LNKCHN" is invoked successively after a DMA transfer in the link-enabled channel is completed. An interrupt request by one of linked channels is generated after transfers in all linked channels are completed.

IDMA invocation request during a DMA transfer

Additional IDMA invocation requests generated during a DMA transfer by IDMA are kept pending until the DMA transfer that was being executed at the time is completed. Since an invocation request is not cleared, new requests will be accepted when the DMA transfer under execution is completed.

IDMA invocation request when DMA transfer is disabled

An IDMA invocation request generated when DMAEN is "0" (DMA transfer disabled) is kept pending until DMAEN is set to "1". Since an invocation request is not cleared, it is accepted when DMA transfer is enabled.

Monitoring DMA transfer status

If an IDMA transfer is started by a software trigger, DSTART is reset to "0" when one transfer is completed. DCHN does not change.

When an IDMA transfer is started by a hardware trigger, the status of the channel that has generated an IDMA transfer completion interrupt can be checked by reading the interrupt vector or the transfer counter.

Simultaneous generation of a software trigger and a hardware trigger

When a software trigger and a hardware trigger are generated simultaneously, the IDMA transfer by the software trigger starts first. The IDMA transfer by the hardware trigger is kept pending and it will start after the transfer by the software trigger has finished.

9.2.4 Operation of IDMA

IDMA has three transfer modes, in each of which data transfer operates differently. Furthermore, an interrupt factor is processed differently depending on the type of trigger. The following describes the operation of IDMA in each transfer mode and how an interrupt factor is processed for each type of trigger.

Single transfer mode

The channels for which DMOD in control information is set to "00" operate in single transfer mode. In this mode, a transfer operation invoked by one trigger is completed after transferring one data unit of the size set by DATSIZ. If a data transfer needs to be performed a number of times as set by the transfer counter, an equal number of triggers are required.

The operation of IDMA in single transfer mode is shown by the flow chart in Figure 9.2.1.

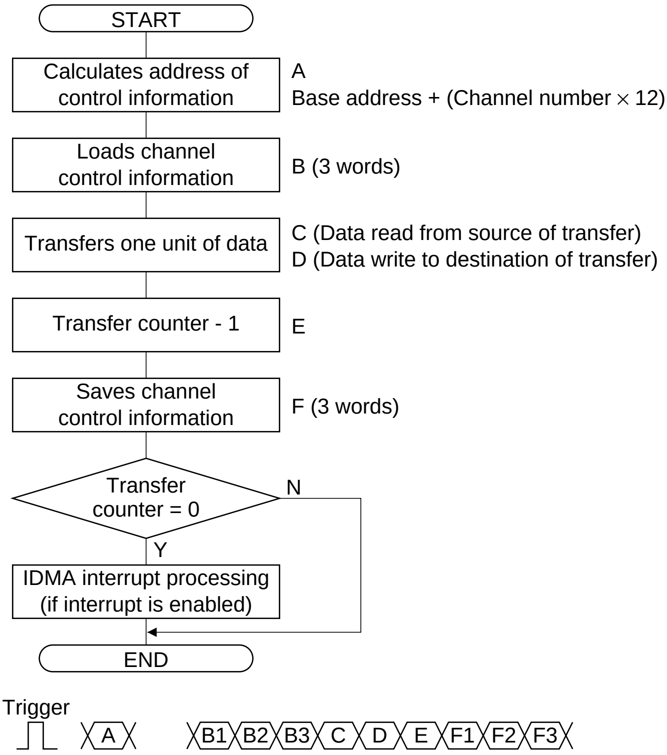

START
Calculates address of control information
A
Base address + (Channel number × 12)
Loads channel control information
B (3 words)
Transfers one unit of data
C (Data read from source of transfer)
D (Data write to destination of transfer)
Transfer counter - 1
E
Saves channel control information
F (3 words)
Transfer counter = 0
N
Y
IDMA interrupt processing (if interrupt is enabled)
END
Trigger
A
B1
B2
B3
C
D
E
F1
F2
F3

Figure 9.2.1 Operation Flow in Single Transfer Mode

(1) When a trigger is accepted, the address for control information is calculated from the base address and channel number.
(2) Control information is read from the calculated address into the internal temporary register of the E0C33A104.
(3) Data of the size set in the control information is read from the source address.
(4) The read data is written to the destination address.
(5) The address is incremented or decremented and the transfer counter is decremented.
(6) The modified control information is written to RAM.
(7) An interrupt factor is processed before completing IDMA.

Successive transfer mode

The channels for which DMOD in control information is set to "01" operate in successive transfer mode. In this mode, a data transfer is performed by one trigger a number of times as set by the transfer counter. The transfer counter is decremented to "0" by one transfer executed.

The operation of IDMA in successive transfer mode is shown by the flow chart in Figure 9.2.2.

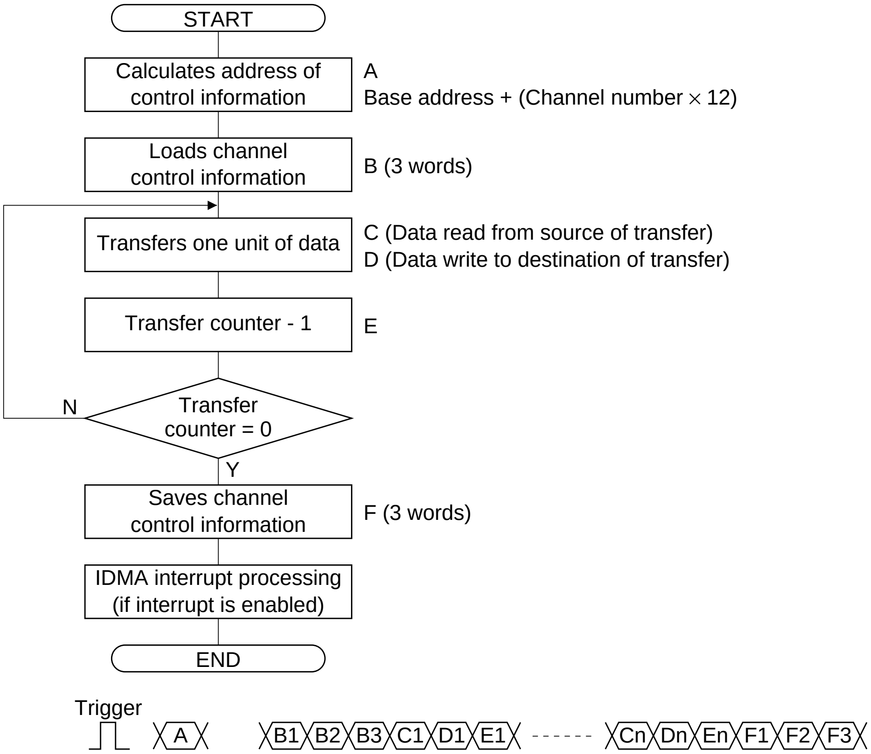

START
Calculates address of control information
A
Base address + (Channel number × 12)
Loads channel control information
B (3 words)
Transfers one unit of data
C (Data read from source of transfer)
D (Data write to destination of transfer)
Transfer counter - 1
E
Transfer counter = 0
N
Y
Saves channel control information
F (3 words)
IDMA interrupt processing (if interrupt is enabled)
END
Trigger
A
B1 B2 B3 C1 D1 E1 ------ Cn Dn En F1 F2 F3

Figure 9.2.2 Operation Flow in Successive Transfer Mode

(1) When a trigger is accepted, the address for control information is calculated from the base address and channel number.

(2) Control information is read from the calculated address into the internal temporary register of the E0C33A104.

(3) Data of the size set in the control information is read from the source address.

(4) The read data is written to the destination address.

(5) The address is incremented or decremented and the transfer counter is decremented.

(6) Steps (3) to (5) are repeated until the transfer counter reaches 0.

(7) The modified control information is written to RAM.

(8) An interrupt factor is processed before completing IDMA.

Block transfer mode

The channels for which DMOD in control information is set to "10" operate in block transfer mode. In this mode, a transfer operation invoked by one trigger is completed after transferring one block of data of the size set by BLKLEN. If a block transfer needs to be performed a number of times as set by the transfer counter, an equal number of triggers are required.

The operation of IDMA in block transfer mode is shown by the flow chart in Figure 9.2.3.

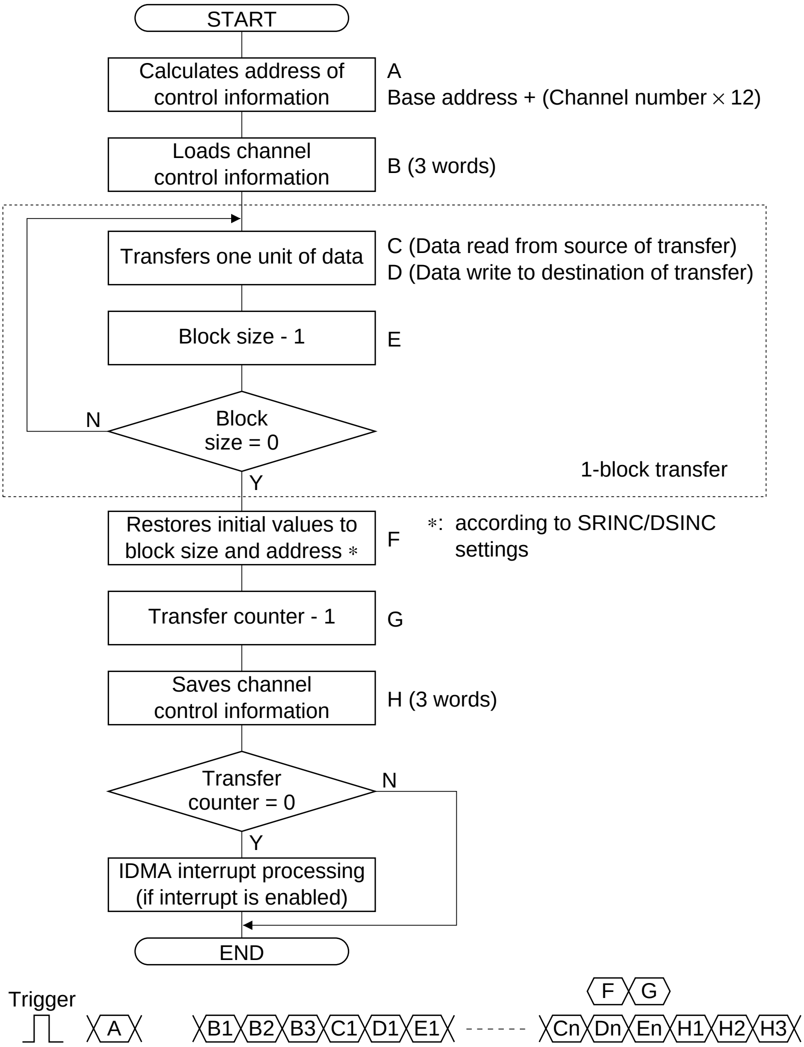

START
Calculates address of control information
A
Base address + (Channel number × 12)
Loads channel control information
B (3 words)
Transfers one unit of data
C (Data read from source of transfer)
D (Data write to destination of transfer)
Block size - 1
E
N
Block size = 0
Y
1-block transfer
Restores initial values to block size and address ∗
F
∗: according to SRINC/DSINC settings
Transfer counter - 1
G
Saves channel control information
H (3 words)
Transfer counter = 0
N
Y
IDMA interrupt processing (if interrupt is enabled)
END
Trigger
A
B1 B2 B3 C1 D1 E1
Cn Dn En H1 H2 H3
F G

Figure 9.2.3 Operation Flow in Block Transfer Mode

(1) When a trigger is accepted, the address for control information is calculated from the base address and channel number.
(2) Control information is read from the calculated address into the internal temporary register of the E0C33A104.
(3) Data of the size set in the control information is read from the source address.
(4) The read data is written to the destination address.
(5) The address is incremented or decremented and BLKLEN is decremented.
(6) Steps (3) to (5) are repeated until BLKLEN reaches 0.
(7) If SRINK and DSINC are "10", the address is recycled to the initial value.
(8) The transfer counter is decremented.
(9) The modified control information is written to RAM.
(10) An interrupt factor is processed before completing IDMA.

Processing of interrupt factors by type of trigger

- **When invoked by an interrupt factor**

The interrupt factor flag by which IDMA has been invoked remains set even during a DMA transfer. If the transfer counter is decremented to 0 and DINTEN = "1" (interrupt enable) when one DMA transfer is completed, the interrupt factor that has invoked IDMA is not reset and an interrupt request is generated. At the same time, the IDMA request register is cleared to "0".

In this case, the interrupt factor flag for the completion of transfer by IDMA itself is not set.

If the transfer counter is not 0 or DINTEN = "0" (interrupt disable), the interrupt factor flag is reset upon completion of transfer. In this case, the IDMA request register is not cleared.

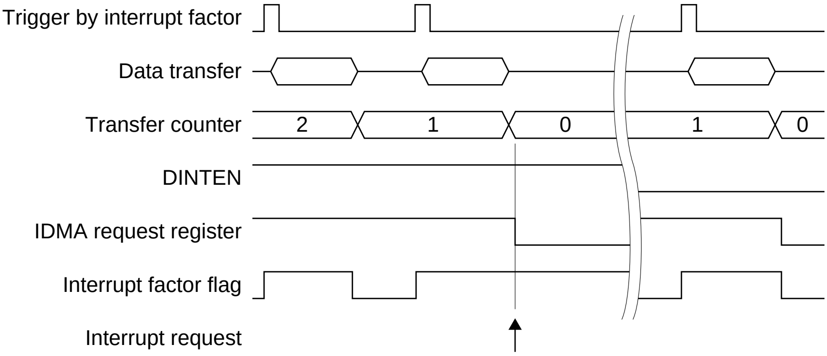


Figure 9.2.4 Operation when Invoked by Interrupt Factor

- **When invoked by a software trigger**

If the transfer counter is decremented to 0 and DINTEN = "1" (interrupt enable) when one DMA transfer is completed, the interrupt factor flag for the completion of IDMA transfer is set, thereby generating an interrupt request.

If the transfer counter is not 0 or DINTEN = "0" (interrupt disable), the interrupt factor flag for the completion of IDMA transfer is not set.

If an interrupt factor flag is set during a software-triggered transfer, a IDMA invocation request by that interrupt factor flag is kept pending until the DMA transfer under execution is completed. The DMA transfer under execution does not reset the generated interrupt factor flag.

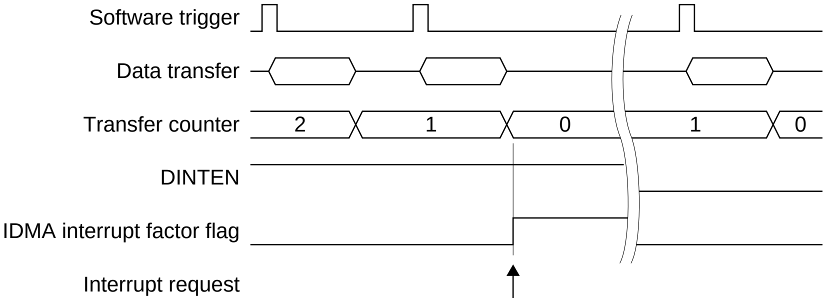


Figure 9.2.5 Operation when Invoked by Software Trigger

9.2.5 Linking

If the IDMA channel number to be executed next is set in the IDMA link field "LNKCHN" of control information and LNKEN is set to "1" (link enabled), DMA successive transfer in that IDMA channel can be performed.
An example of link setting is shown in Figure 9.2.6.

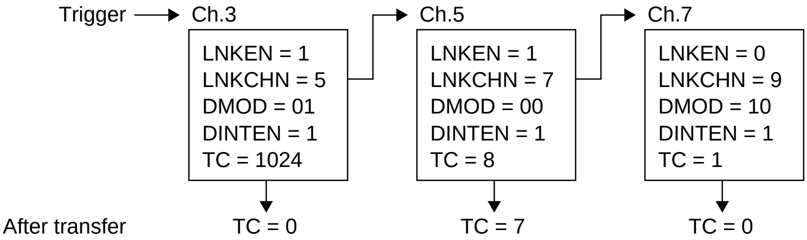


Figure 9.2.6 Example of Link Setting

For the above example, IDMA operates as described below.

• For trigger in hardware

(1) The IDMA channel 3 is invoked by an interrupt factor and the DMA transfer that is set is performed. Since the IDMA is operating in successive transfer mode and the transfer counter is decremented to 0 and DINTEN is set to "1", the interrupt factor flag by which the channel 3 has been invoked remains set.

(2) Next, a DMA transfer is performed via the linked IDMA channel 5. Channel 5 is set for single transfer mode and the transfer counter in this transfer is decremented by 1.

(3) Finally, a DMA transfer in IDMA channel 7 is performed. Although the channel 7 is set for block transfer mode, the transfer counter is decremented to 0 when the transfer is completed because the number of transfers to be performed is 1.

(4) Since the interrupt factor flag that has invoked IDMA channel 3 in (1) remains set, an interrupt is generated when the IDMA transfer (channel 7) in (3) is completed. The DINTEN settings and transfer results of channels 5 and 7 do not affect the interrupt factor flag of channel 3.

• For trigger in the software application

(1) The IDMA channel 3 is invoked by a trigger in the software application and the DMA transfer that is set is performed. Since the IDMA is operating in successive transfer mode and the transfer counter is decremented to 0 and DINTEN is set to "1", an interrupt factor flag for the completion of IDMA transfer is set when the transfer is completed.

(2) Next, a DMA transfer is performed in the linked IDMA channel 5. The channel 5 is set for the single transfer mode and the transfer counter in this transfer is decremented by 1.

(3) Finally, a DMA transfer in IDMA channel 7 is performed. Although channel 7 is set for the block transfer mode, the transfer counter is decremented to 0 when the transfer is completed because the number of transfers to be performed is 1. The completion of this transfer also causes the interrupt factor flag for the completion of IDMA transfer to be set to "1". However, the interrupt factor flag for the completion of IDMA transfer has already been set when the transfer is completed in (1) above.

(4) Since the interrupt factor flag for the completion of IDMA transfer is set, an interrupt request is generated here. In cases when IDMA has been invoked by a trigger in the software application, if the transfer counter in any one of the linked channels is decremented to 0 and DINTEN for that channel is set to "1", an interrupt request for the completion of IDMA transfer is generated when a transfer operation in each of the linked channels is completed. The channel in which an interrupt request has been generated can be verified by reading out the transfer counter.

Transfer operations in each channel are performed as described earlier.

9.2.6 Interrupt Function of Intelligent DMA

IDMA can generate an interrupt that causes invocation of IDMA and an interrupt for the completion of IDMA transfer itself.

Interrupt when invoked by an interrupt factor

If the corresponding bits of the IDMA request and interrupt enable registers are left set (= "1"), assertion of an interrupt request is kept pending even when the enabled interrupt factor has occurred and the IDMA channel assigned to that interrupt factor is invoked. If DINTEN in control information is left set (= "1"), the pending interrupt request can be output when the DMA transfer is completed and the transfer counter is decremented to 0. In this case, the IDMA request register is cleared, and the interrupt factor flag for the completion of IDMA transfer itself is not set.

If the transfer counter is not 0 or DINTEN = "0" (interrupt disabled), the interrupt factor flag is set when the transfer is completed. In this case, the IDMA request register is not cleared.

For details about the interrupt factors that can be used to invoke IDMA and the interrupt control registers, refer to the descriptions of the peripheral circuits in this manual.

Note that the priority levels of interrupt factors are set by the interrupt priority register. (Refer to Chapter 8, "Interrupt".) However, when compared between IDMA and interrupt requests, IDMA is given higher priority over the other. Consequently, even when an interrupt factor occurring during an IDMA transfer has higher priority than the interrupt factor that invoked the IDMA transfer, an interrupt request for it or a new IDMA invocation request is not accepted until after the current IDMA transfer is completed. Since the interrupt level (IL) is rewritten to that of the interrupt being processed and remains so during interrupt processing, an IDMA invocation request triggered by an interrupt factor whose priority is below that level is not accepted. If you want IDMA to be invoked by an interrupt factor of lower priority during interrupt processing, the interrupt factor flag for the interrupt being processed must be reset in the interrupt processing routine to rewrite the IL level.

Software-triggered interrupts

If the transfer counter is decremented to 0 and DINTEN = "1" (interrupt enable) when one DMA transfer operation is completed, the interrupt factor flag for the completion of IDMA transfer is set, thereby generating an interrupt request. If the transfer counter is not 0 or DINTEN = "0" (interrupt disabled), the interrupt factor flag for the completion of IDMA transfer is not set.

Control register of interrupt controller

The following registers are used to control an interrupt for the completion of IDMA transfer:

Interrupt factor flag:	FIDMA(D2) / DMA interrupt factor flag register (0x40281)
Interrupt enable:	EIDMA(D2) / DMA interrupt enable register (0x40271)
Interrupt level:	PDM[2:0](D[2:0]) / 16-bit timer 0–1, DMA interrupt priority register (0x40261)

When a DMA transfer in the IDMA channel invoked by a trigger in the software application or subsequent link is completed and the transfer counter is decremented to 0, the interrupt factor flag for the completion of IDMA transfer is set to "1". However, this requires as a precondition that interrupt be enabled (DINTEN = "1") in the control information for that channel. If the interrupt enable register bit remains set (= "1") when the flag is set, an interrupt request is generated.

Interrupts can be disabled by leaving the interrupt enable register bit cleared (= "0"). Use the interrupt priority register to set interrupt priority levels (0 to 7). An interrupt request to the CPU is accepted on condition that no other interrupt request of higher priority is generated.

Furthermore, it is only when the PSR's IE bit = "1" (interrupt enabled) and the set value of IL is smaller than the IDMA interrupt level which is set by the interrupt priority register that the CPU actually accepts an IDMA interrupt request.

For details about these interrupt control registers, and for information on device operation when an interrupt occurs, refer to Chapter 8, "Interrupt".

Trap vector

The trap vector address for an interrupt upon completion of IDMA transfer by default is set as follows:
BTA3 = low: 0x0C00068

The trap table base address can be changed using the TTBR registers (0x48134 to 0x48137).

9.2.7 I/O Memory of Intelligent DMA

Table 9.2.3 shows the control bits of IDMA.

Table 9.2.3 Control Bits of IDMA

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 0–1, DMA interrupt priority register	**0040261 (B)**	D7	**–**	reserved	–				–	–	0 when being read.
		D6	**P16T02**	16-bit timer 0–1 interrupt level	0 to 7				X	R/W	
		D5	**P16T01**						X		
		D4	**P16T00**						X		
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**PDM2**	IDMA, high-speed DMA interrupt level	0 to 7				X	R/W	
		D1	**PDM1**						X		
		D0	**PDM0**						X		
DMA interrupt enable register	**0040271 (B)**	D7–3	**–**	reserved	–				–	–	0 when being read.
		D2	**EIDMA**	IDMA	1	Enabled	0	Disabled	0	R/W	
		D1	**EHDM1**	High-speed DMA Ch.1					0	R/W	
		D0	**EHDM0**	High-speed DMA Ch.0					0	R/W	
DMA interrupt factor flag register	**0040281 (B)**	D7–3	**–**	reserved	–				–	–	0 when being read.
		D2	**FIDMA**	IDMA	1	Factor is generated	0	No factor is generated	X	R/W	
		D1	**FHDM1**	High-speed DMA Ch.1					X	R/W	
		D0	**FHDM0**	High-speed DMA Ch.0					X	R/W	
IDMA base address low-order register	**0048230 (HW)**	DF	**DBASEL15**	IDMA base address low-order 16 bits (Initial value: 0x0C003A0)					0	R/W	
		DE	**DBASEL14**						0		
		DD	**DBASEL13**						0		
		DC	**DBASEL12**						0		
		DB	**DBASEL11**						0		
		DA	**DBASEL10**						0		
		D9	**DBASEL9**						1		
		D8	**DBASEL8**						1		
		D7	**DBASEL7**						1		
		D6	**DBASEL6**						0		
		D5	**DBASEL5**						1		
		D4	**DBASEL4**						0		
		D3	**DBASEL3**						0		
		D2	**DBASEL2**						0		
		D1	**DBASEL1**						0		
		D0	**DBASEL0**						0		
IDMA base address high-order register	**0048232 (HW)**	DF–C	**–**	reserved	–				–	–	Undefined in read.
		DB	**DBASEH11**	IDMA base address high-order 12 bits (Initial value: 0x0C003A0)					0	R/W	
		DA	**DBASEH10**						0		
		D9	**DBASEH9**						0		
		D8	**DBASEH8**						0		
		D7	**DBASEH7**						1		
		D6	**DBASEH6**						1		
		D5	**DBASEH5**						0		
		D4	**DBASEH4**						0		
		D3	**DBASEH3**						0		
		D2	**DBASEH2**						0		
		D1	**DBASEH1**						0		
		D0	**DBASEH0**						0		
IDMA start register	**0048234 (B)**	D7	**DSTART**	IDMA start	1	IDMA start	0	Stop	0	R/W	
		D6–0	**DCHN**	IDMA channel number	0 to 127				0	R/W	
IDMA enable register	**0048235 (B)**	D7–1	**–**	reserved	–				–	–	
		D0	**DMAEN**	DMA enable	1	Enabled	0	Disabled	0	R/W	

DBASEL[15:0]: IDMA base address A[15:0] (D[F:0]) / IDMA base address low-order register (0x48230[HW])
DDBASEH[11:0]: IDMA base address A[27:16] (D[B:0]) / IDMA base address high-order register (0x48232[HW])

Specify the starting address of the control information to be placed in RAM.
Use DBASEL to set the 16 low-order bits of the address and DBASEH to set the 12 high-order bits.
The address to be set in these registers must always be a word (32-bit) boundary address.
These registers cannot be read or written in bytes. The registers must be accessed in words for read/write operations to address 0x48230, and in half-words for read/write operations to addresses 0x48230 and 0x48232. Write operations in half-words must be performed in order of 0x48230 and 0x48232. Read operations in half-words may be performed in any order.
Write operations to the IDMA base address registers during a DMA transfer are ignored. When the register is read during a DMA transfer, the read data is indeterminate.
At initial reset, the base address is set to 0xC003A0.

DMAEN: DMA enable (D0) / DMA enable register (0x48235)

Enable a DMA transfer.

Write "1":	Enabled
Write "0":	Disabled
Read:	Valid

A data transfer operation by intelligent DMA and high-speed DMA is enabled by writing "1" to DMAEN.
DMA transfer is disabled by writing "0" to DMAEN.
Be sure to disable DMA transfers (DMAEN = "0") before you set the base address and transfer conditions.
At initial reset, DMAEN is set to "0" (disabled).

DCHN[6:0]: IDMA channel number (D[6:0]) / IDMA start register (0x48234)

Set the channel numbers (0 to 127) to be invoked by a trigger in the software application.
At initial reset, DCHN is set to "0".

DSTART: IDMA start (D7) / IDMA start register (0x48234)

Use this register for a trigger in the software application and for monitoring the operation of IDMA.

When written

Write "1":	IDMA started
Write "0":	Invalid

When read

Read "1":	IDMA operating (only when invoked by software trigger)
Read "0":	IDMA inactive

When DSTART is set to "1", it functions as a trigger in the software application, invoking the IDMA channel that is set in the DCHN register.
DSTART can be read as "1" while IDMA is operating only when IDMA has invoked by a software trigger. When IDMA has invoked by the hardware via ITC, DSTART is read as "0" during IDMA transfer.
At initial reset, DSTART is set to "0".

PDM2–PDM0: DMA interrupt level (D[2:0]) / 16-bit timer 0–1, DMA interrupt priority register (0x40261)

Set the priority level of the interrupt upon completion of IDMA transfer in the range of 0 to 7.
At initial reset, the contents of this register are indeterminate.

EIDMA: IDMA interrupt enable (D2) / DMA interrupt enable register (0x40271)

Enable or disable occurrence of an interrupt to the CPU.

Write "1": Interrupt enabled
Write "0": Interrupt disabled
Read: Valid

This bit controls the interrupt generated upon completion of IDMA transfer. The interrupt is enabled by setting this register to "1" and disabled by setting this register to "0".
At initial reset, EIDMA is set to "0" (interrupt disable).

FIDMA: IDMA interrupt factor flag (D2) / DMA interrupt factor flag register (0x40281)

Indicate the occurrence status of an IDMA interrupt request.

When read
Read "1": Interrupt factor occurred
Read "0": No interrupt factor occurred

When written using reset-only method (default)
Write "1": Interrupt factor flag is reset
Write "0": Invalid

When written using the read/write method
Write "1": Interrupt factor flag is set
Write "0": Interrupt factor flag is reset

This flag is set to "1" when one DMA transfer initiated by a software trigger or subsequent link is completed and the transfer counter is decremented to 0. However, this requires as a precondition that interrupts be enabled in control information (DINTEN = "1").
At this time, an interrupt to the CPU is generated if the following conditions are met:
1. The corresponding interrupt enable register bit is set to "1".
2. No interrupt request of higher priority is generated.
3. The IE bit of the PSR is set to "1" (interrupt enable).
4. The corresponding interrupt priority register is set to a level higher than the CPU's interrupt level (IL).

In order for the next interrupt to be accepted after interrupt generation, the interrupt factor flag must be reset and the PSR must be set up again (by setting the IL below the level indicated by the interrupt priority register and setting the IE bit to "1" or executing a reti instruction).
The interrupt factor flag can only be reset by a write instruction in the software application. If the PSR is set up again to accept interrupts (or the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt may occur again. Note also that the value to be written to reset the flag is "1" when using the reset-only method (RSTONLY = "1") and "0" when using the read/write method (RSTONLY = "0"). Be careful not to confuse these two cases.
This flag becomes indeterminate when initially reset, so be sure to reset it in the software application.

9.2.8 Programming Notes

(1) Before setting the IDMA base address, be sure to disable DMA transfers (DMAEN = "0"). Writes to the IDMA base address register during a DMA transfer are invalid. Also, when the register is read during a DMA transfer, the data is indeterminate. When setting or rewriting control information for each channel, make sure that DMA transfers will not occur in any channel.

(2) The address that is set in the IDMA base address register must always be a word (32-bit) boundary address.

(3) The IDMA base address register cannot be read or written in bytes. The register must be accessed in words for read/write to address 0x48230, and in half-words for read/write to addresses 0x48230 and 0x48232. Write operations in half-words must be performed in order of 0x48230 and 0x48232. Read operations in half-words may be performed in any order.

(4) After an initial reset, the interrupt factor flag (FIDMA) becomes indeterminate. To prevent unwanted interrupts from occurring, be sure to reset the flag in a program.

(5) Once an interrupt occurs, be sure to reset the interrupt factor flag (FIDMA) before setting up the PSR again or executing the reti instruction. This ensures that an interrupt will not be generated for the same factor.

(6) During an IDMA transfer, high-speed DMA requests and interrupt requests (even those of higher priority) are kept pending. These requests are accepted after the completion of the transfer.
Since the interrupt level (IL) is rewritten to that of the interrupt being processed and remains so during interrupt processing, an IDMA invocation request triggered by an interrupt factor whose priority is below that level is not accepted. If you want IDMA to be invoked by an interrupt factor of lower priority during interrupt processing, the interrupt factor flag for the interrupt being processed must be reset in the interrupt processing routine and rewrite the IL level.

(7) If an IDMA channel is set so that it generates an interrupt when the IDMA transfer has completed, the IDMA controller clears the IDMA request flag in ITC when the transfer counter becomes "0". This IDMA operation does not clear the interrupt factor flag in ITC, so ITC can generate an interrupt to the CPU.
An IDMA request flag register (byte-size-register) contains several IDMA request flags and the status of the IDMA request flags may be modified by a load/store operation. The CPU performs a read-modify-write operation in byte units for this modification.
Therefore, if an IDMA request flag is cleared due to the end of the transfer while the CPU is setting another IDMA request flag in the same register, the cleared IDMA request flag may be set again by the CPU's read-modify-write operation.
To avoid this problem, do not activate two or more IDMA channels assigned in the same IDMA request register at once. This problem does not occur if only one IDMA channel in an IDMA request flag register is enabled at a time.

(8) When "0" is written to bit 0 of address 0x004029f, interrupt flags in ITC will be modified by the read/write operation of the CPU (the default value is "1", reset-only method). With this setting, an IDMA transfer may not clear the interrupt flag, so the IDMA channel may be activated repeatedly with the same interrupt factor.
To avoid this problem, do not set "0" to bit 0 of address 0x004029f when using IDMA.

(9) If an IDMA trigger which has a higher interrupt level than that of the CPU happens when an interrupt occurs and the program jumps to the interrupt service routine, the interrupt level of the interrupt service routine may be set to the IDMA's interrupt level which is higher than the correct value.
When enabling nested interrupts by setting IE = 1 in PSR, the nested interrupt request from ITC may be masked by this incorrect interrupt level in PSR.
To enable nested interrupts in the interrupt service routine when IDMA is working, set the correct interrupt level to PSR at the same time as setting IE = 1 in the PSR. In other words, over-write the interrupt level by software in the interrupt service routine when enabling nested interrupts.

10 Oscillation Circuits and Clock Control

This chapter describes the method for controlling the oscillation circuits and the method for controlling the operating clocks of the CPU and internal peripheral circuits.

10.1 Oscillation Circuits

10.1.1 Configuration of Oscillation Circuits

The E0C33A104 uses a twin-clock structure consisting of two types of oscillation circuits (OSC3 and OSC1).

The high-speed (OSC3) oscillation circuit generates the main clock for the CPU and internal peripheral circuits (e.g., DMA, serial interface, programmable timer, and A/D converter). The low-speed (OSC1) oscillation circuit generates a subclock (32.768 kHz, Typ.) for the clock timer and for operating the CPU at a low clock speed in order to reduce current consumption.

Figure 10.1.1 shows the configuration of the oscillation circuits.

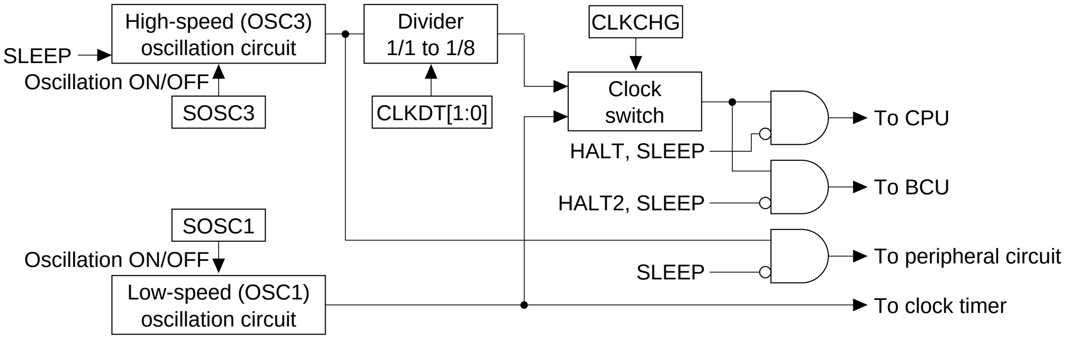


Figure 10.1.1 Configuration of Oscillation Circuits

After an initial reset, the output (OSC3 clock) of the high-speed (OSC3) oscillation circuit is set for the CPU operating clock.

The CPU operating clock can be switched to the output (OSC1 clock) of the low-speed (OSC1) oscillation circuit in a program. Furthermore, each oscillation circuit can be stopped in a program.

If the OSC3 clock is unnecessary such as when performing clock processing only, set the OSC1 clock for operation of the CPU and turn off the high-speed (OSC3) oscillation circuit in order to reduce current consumption. In addition, when SLEEP mode is set, the high-speed (OSC3) oscillation circuit is turned off, greatly reducing current consumption (no internal units except for the clock timer need to be operated).

10.1.2 I/O Pins of Oscillation Circuits

Table 10.1.1 lists the I/O pins of the oscillation circuits.

Table 10.1.1 I/O Pins of Oscillation Circuits

Pin name	Pin No.		Pull-up	I/O	Function
	QFP5-128	QFP15-128			
OSC1	33	29	–	I	Low-speed (OSC1) oscillation input pin Crystal oscillation or external clock input
OSC2	32	28	–	O	Low-speed (OSC1) oscillation output pin Crystal oscillation (open when external clock is used)
OSC3	15	12	–	I	High-speed (OSC3) oscillation input pin Crystal/ceramic oscillation or external clock input
OSC4	14	11	–	O	High-speed (OSC3) oscillation output pin Crystal/ceramic oscillation (open when external clock is used)

10.1.3 High-Speed (OSC3) Oscillation Circuit

The high-speed (OSC3) oscillation circuit generates the main clock for the CPU and internal peripheral circuits (e.g., DMA, serial interface, programmable timer, and A/D converter).
This circuit can be a crystal or a ceramic oscillation circuit. Optionally an external clock source can be used.
Figure 10.1.2 shows the structure of the high-speed (OSC3) oscillation circuit.

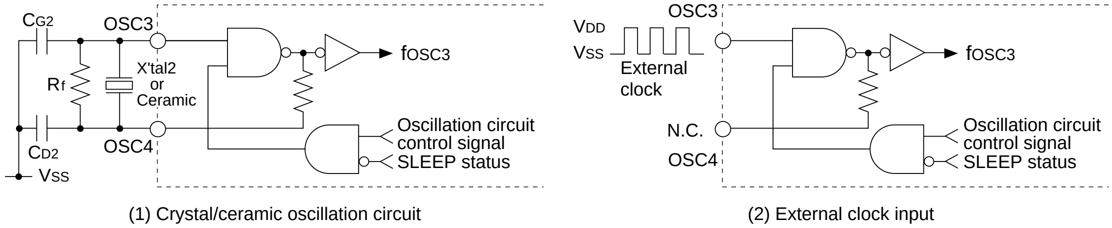


Figure 10.1.2 High-Speed (OSC3) Oscillation Circuit

When using a crystal or a ceramic oscillation for this circuit, connect a crystal (X'tal2) or ceramic (Ceramic) resonator and feedback resistor (Rf) between the OSC3 and OSC4 pins, and two capacitors (C_{G2}, C_{D2}) between the OSC3 pin and V_{SS} and the OSC4 pin and V_{SS}, respectively.

Note: The high-speed (OSC3) oscillation circuit may not generate the clock signals if a fundamental crystal resonator is NOT being used. Use a fundamental resonator (MA-306 made by Seiko Epson, etc.) for the high-speed (OSC3) oscillation circuit.

When an external clock is used, leave the OSC4 pin open and input a square-wave clock to the OSC3 pin.
The range of oscillation frequencies is shown in Table 10.1.2. This frequency range also applies when an external clock is used.

Table 10.1.2 OSC3 Clock Frequency Range

Supply voltage (V_{DD})	Frequency range
3.3 V	5 MHz to 20 MHz
5.0 V	5 MHz to 33 MHz

For details on oscillation characteristics and the external clock input characteristics, refer to Chapter 18, "Electrical Characteristics".

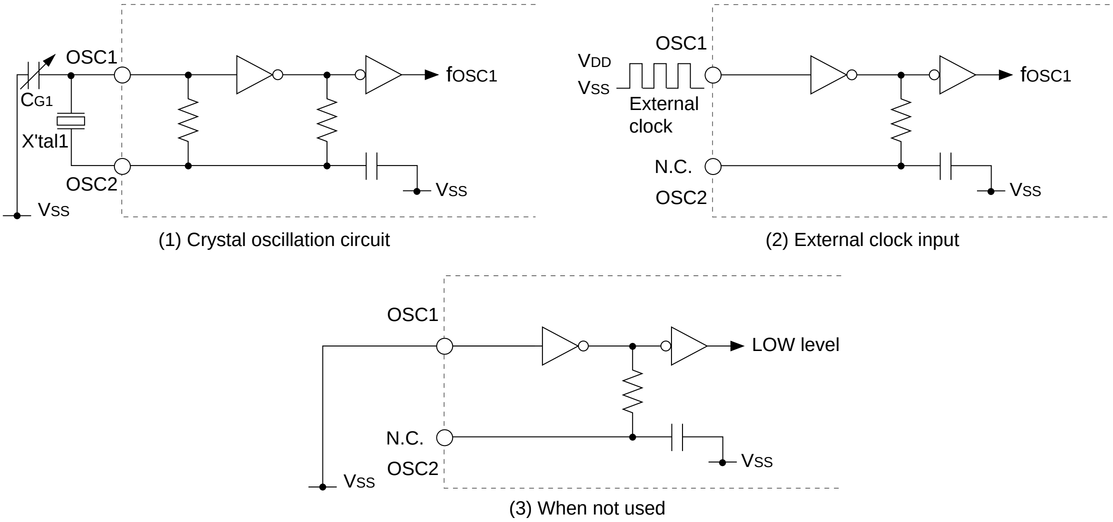
10.1.4 Low-Speed (OSC1) Oscillation Circuit

The low-speed (OSC1) oscillation circuit generates a 32.768-kHz (Typ.) subclock.

The OSC1 clock output by this circuit is used as the source clock for the clock timer. It can also be used as a clock for the low-speed (low-power) operation of the CPU (switchable in a program).

A crystal oscillation or an external clock input can be selected as the type of oscillation circuit in a program.

Figure 10.1.3 shows the structure of the low-speed (OSC1) oscillation circuit.



(1) Crystal oscillation circuit

(2) External clock input

(3) When not used

Figure 10.1.3 Low-Speed (OSC1) Oscillation Circuit

When using a crystal oscillation for this circuit, connect a crystal resonator X'tal1 (32.768 kHz, Typ.) between the OSC1 and OSC2 pins and a trimmer capacitor CG1 (5 to 25 pF, total value including the board capacitance) between the OSC1 pin and VSS.

When an external clock source is used, leave the OSC2 pin open and input a square-wave clock to the OSC1 pin.

If the low-speed (OSC1) oscillation circuit is not used, connect the OSC1 pin to VSS and leave the OSC2 pin open. In this case, select external clock input as the type of oscillation circuit.

Use CLGRON (D1) / Low-speed oscillation control register (0x4015F) to select the type of oscillation circuit (whether a feedback resistor is used or not). CLGRON is set to "0" at initial reset, so the type of oscillation circuit is a crystal oscillation using a feedback resistor. When external clock is used or the low-speed (OSC1) oscillation circuit is not used, write "1" to CLGRON to disconnect the feedback resistor. Furthermore, be sure to write "1" to the D0 bit of the low-speed oscillation control register (0x4015F) when changing to the register.

The low-speed (OSC1) oscillation circuit is powered by VDD1 voltage (3.3 V or 5 V, VDD1 ≤ VDD) that is separate from the system power supply VDD (refer to Section 2.1.3, " Power Supply for Clock").

The oscillation frequency is 32.768 kHz (Typ.). Use a crystal resonator or external clock that oscillates at this frequency. No other frequency can be used for clock applications.

For details on oscillation characteristics and the external clock input characteristics, refer to Chapter 18, "Electrical Characteristics".

10.1.5 Controlling Oscillation

Each oscillation circuit can be turned on or off using the following bits:
SOSC3 (D1) / Power control register (0x40140): Turns OSC3 on or off
SOSC1 (D0) / Power control register (0x40140): Turns OSC1 on or off

The oscillation circuit is turned off by writing "0" to the corresponding control bit and turned back on again by writing "1". Both the control bits are set to "1" at initial reset, so the oscillation circuits are turned on.

Notes:
- The oscillation circuit used for the CPU operating clock cannot be turned off. In this case, writing "0" to the control bit is ignored. Note also that writing to these control bits is allowed only when the power-control register protection flag is set to "0b10010110" (refer to Section 10.1.7).
- Immediately after the oscillation circuit is turned on, a certain period of time is required for oscillation to stabilize (for OSC3 using a 3.3-V crystal resonator, this time is 10 ms max.; for OSC1, this time is 3 sec max. For details, refer to Section 18.8, "Oscillation Characteristics"). To prevent the device from operating erratically, do not use the clock until its oscillation has stabilized.

The high-speed (OSC3) oscillation circuit turns off when the CPU is set in SLEEP mode. (Refer to Section 10.1.8.) The low-speed (OSC1) oscillation circuit can also be turned off through the use of an auto-OFF function that is actuated by a clock timer interrupt. (Refer to Section 11.1.4.)

10.1.6 Setting and Switching Over the CPU Operating Clock

Setting the CPU operating clock frequency

When operating the CPU with the high-speed (OSC3) clock, the operating frequency can be switched over in four steps. Use CLKDT[1:0] (D[7:6]) / Power control register (0x40140) for this switchover.

Table 10.1.3 Setting of CPU Operating Clock

CLKDT1	CLKDT0	Division ratio
1	1	$f_{OSC3}/8$
1	0	$f_{OSC3}/4$
0	1	$f_{OSC3}/2$
0	0	$f_{OSC3}/1$

The clock thus set becomes the system clock, which is used as the CPU operating clock and the bus clock.
At initial reset, the division ratio is set to $f_{OSC3}/1$, so the CPU is operated directly by the high-speed (OSC3) oscillation clock. Since the device's current consumption can be decreased by reducing the CPU operating speed, switch over the operating frequency as necessary.
This setting is effective only for the high-speed (OSC3) clock, and has no effect when the low-speed (OSC1) clock is used as the system clock.

Note: Writing to CLKDT[1:0] is effective only when the power-control register protection flag is set to "0b10010110". (Refer to Section 10.1.7.)

Switching over the CPU operating clock

After an initial reset, the CPU starts operating using the OSC3 clock. All internal peripheral circuits also operate.
In cases in which some peripheral circuits (e.g., programmable timer, serial interface, and A/D converter) that are clocked by the OSC3 clock do not need to be operate and the CPU can process its jobs at a low clock speed, the CPU operating clock can be switched to the OSC1 clock, thereby reducing current consumption. Use CLKCHG (D2) / Power control register (0x40140) to switch over the operating clock.

Procedure for switching over from the OSC3 clock to the OSC1 clock

1. Turn on the low-speed (OSC1) oscillation circuit (by writing "1" to SOSC1).
2. Wait until the OSC1 oscillation stabilizes (three seconds or more).
3. Change the CPU operating clock (by writing "0" to CLKCHG).
4. Turn off the high-speed (OSC3) oscillation circuit (by writing "0" to SOSC3).

∗ Steps 1 and 2 are required only when the low-speed (OSC1) oscillation circuit is inactive.

Notes:
- Use separate instructions to switch from OSC3 to OSC1 and turn the OSC3 oscillation off. If these operations are processed simultaneously using one instruction, the CPU may operate erratically.
- Make sure the operation of the peripheral circuits, such as the programmable timer, A/D converter, and serial interface, which are clocked by the OSC3 oscillation circuit, is terminated before the OSC3 oscillation is turned off in order to prevent them from operating erratically.

Procedure for switching over from the OSC1 clock to the OSC3 clock

1. Turn on the high-speed (OSC3) oscillation circuit (by writing "1" to SOSC3).
2. Wait until the OSC3 oscillation stabilizes (10 ms or more for a 3.3-V crystal resonator).
3. Switch over the CPU operating clock (by writing "1" to CLKCHG).

Note: The operating clock switchover by CLKCHG is effective only when both oscillation circuits are on and the power-control register protection flag is set to "0b10010110". (Refer to Section 10.1.7.)

10.1.7 Power-Control Register Protection Flag

The power-control register (SOSC1, SOSC3, CLKCHG, CLKDT[1:0]) at address 0x40140, which is used to control the oscillation circuits and the CPU operating clock, is normally disabled against writing in order to prevent it from malfunctioning due to unnecessary writing.

To enable this register for writing, the power-control register protection flag CLGP[7:0] (D[7:0]) / Power-control protection register (0x4015E) must be set to "0b10010110". Note that this setting allows for the power-control register (0x40140) to be written to only once, so all bits of CLGP[7:0] are cleared to "0" when this address is written to. Therefore, CLGP[7:0] must be set to "0b10010110" each time the power-control register (0x40140) is written to. The flag CLGP[7:0] does not affect the readout from the power-control register (0x40140).

10.1.8 Operation in Standby Mode

In HALT mode, which is entered by executing the halt instruction, the high-speed (OSC3) and low-speed (OSC1) oscillation circuits both retain their status before HALT mode is entered. Under normal conditions, therefore, there is no need to control the oscillation circuits before entering or after exiting HALT mode.

The high-speed (OSC3) oscillation circuit stops operating after SLEEP mode is entered, which is done by executing the slp (sleep) instruction. If the high-speed (OSC3) oscillation circuit was operating before SLEEP mode was entered, it automatically starts oscillating again after SLEEP mode is exited.
In addition, if the CPU was operating using the OSC3 clock before SLEEP mode was entered, the CPU starts operating using the OSC3 clock again even after SLEEP mode is exited. The high-speed (OSC3) oscillation circuit requires 10 ms max. (when using a 3.3-V crystal resonator) for its oscillation to stabilize after oscillation starts. To prevent the CPU from operating erratically upon restart during this period, the E0C33A104 is designed to allow the OSC3 clock supply to the CPU to be disabled in the hardware after SLEEP mode is exited. Use 8T1ON (D2) / Clock option register (0x40150) to select this function. Use 8-bit programmable timer 1 to set the waitting time before clock supply is started.
The processing procedure and the operations to be performed when this function is used are as follows:

1. Disable the 8-bit programmable timer 1 interrupt.
2. Preset the initial count to 8-bit programmable timer 1.
   Set a value that will provide an ample stabilization waiting time. It is also necessary to set the input clock for 8-bit programmable timer 1 using the prescaler.
3. Enable the interrupt used to exit SLEEP mode.
   Before enabling the interrupt, be sure to reset the interrupt factor flag.
4. Write "0" to 8TION (turn on the function for waiting until the oscillation stabilizes after exiting SLEEP mode).
5. Activate 8-bit programmable timer 1 to start counting.
6. Enter SLEEP mode using the slp instruction.
   :
   SLEEP mode
   :
7. Exit SLEEP mode using an NMI, input port, or timer interrupt.
8. The high-speed (OSC3) oscillation circuit starts oscillating when SLEEP mode is exited. 8-bit programmable timer 1 also is made to start counting using the OSC3 clock.
9. 8-bit programmable timer 1 underflows.
   The operating clock supply to the CPU is begun by the underflow signal, so that the CPU restarts.

For details on how to control the 8-bit programmable timer, prescaler, and interrupts, refer to the description of each item in this manual.

Note: The function for waiting until the high-speed (OSC3) oscillation is stabilized by 8TION is effective only when SLEEP mode is exited.
Writing to 8TION is effective only when the power-control register protection flag is set to "0b10010110". (Refer to Section 10.1.7.)

10.1.9 OSC1 Clock Output to External Devices

The low-speed (OSC1) oscillation clock can be output from the FOSC1 (R81) pin to external devices.

Table 10.1.4 OSC1 Clock Output Pin

Pin name	Pin No.		I/O	Pull-up	Function	Function select bit
	QFP5-128	QFP15-128				
R81/FOSC1/ #DRD	57	54	O	–	Output port / Low-speed (OSC1) oscillation clock output / #DRD signal output	CFR81(D1) / R8 function select register (0x402CC) CFEX2 (D2) / Port function extension register (0x402D8)

Setting the clock output pin

The pin used to output the OSC1 clock to external devices is shared with the R81 output port and the DRAM read signal #DRD.

At cold start, it is set for the output port (CFR81 = "0" and CFEX2 = "0"). When using the clock output function, write "1" to CFR81 and "0" to CFEX2 (refer to Section 14.2, "Output Ports"). Since the function select register and the port function extension register are write-only registers (data is indeterminate when read), bit operation instructions (e.g., bset, bclr, or bnot) that accompany a read-modify-write operation cannot be used for rewriting. Use ordinary storage instructions for this purpose.

At hot start, the pin retains its pre-reset status.

Output control

To start clock output, write "1" to PF1ON (D0) / Clock option register (0x40150). The clock output is stopped by writing "0".

At initial reset, PF1ON is set to "0" (output disabled).

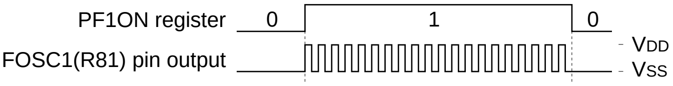


Figure 10.1.4 OSC1 Clock Output

10.1.10 I/O Memory of Oscillation Circuits

Table 10.1.5 lists the control bits of oscillation circuits.

Table 10.1.5 Control Bits of Oscillation Circuits

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Power control register	**0040140 (B)**	D7	**CLKDT1**	System clock division ratio selection	CLKDT[1:0]		Division ratio		0	R/W	
		D6	**CLKDT0**		1	1	1/8		0		
					1	0	1/4				
					0	1	1/2				
					0	0	1/1				
		D5	**PSCON**	Prescaler On/Off control	1	On	0	Off	1	R/W	
		D4–3	–	reserved	–				0	–	Do not write 1.
		D2	**CLKCHG**	CPU operating clock switch	1	OSC3	0	OSC1	1	R/W	
		D1	**SOSC3**	High-speed (OSC3) oscillation On/Off	1	On	0	Off	1	R/W	
		D0	**SOSC1**	Low-speed (OSC1) oscillation On/Off	1	On	0	Off	1	R/W	
Clock option register	**0040150 (B)**	D7–4	–	–	–				–	–	0 when being read.
		D3	**HLT2OP**	HALT clock option	1	On	0	Off	0	R/W	
		D2	**8T1ON**	OSC3-stabilize waiting function	1	Off	0	On	1	R/W	
		D1	–	reserved	–				0	–	Do not write 1.
		D0	**PF1ON**	OSC1 external output control	1	On	0	Off	0	R/W	
Power control protect register	**004015E (B)**	D7	**CLGP7**	Power control register protect flag	Writing 10010110 (0x96) removes the write protection of the power control register (0x40140). Writing another value set the write protection.				0	R/W	
		D6	**CLGP6**						0		
		D5	**CLGP5**						0		
		D4	**CLGP4**						0		
		D3	**CLGP3**						0		
		D2	**CLGP2**						0		
		D1	**CLGP1**						0		
		D0	**CLGP0**						0		
Low-speed oscillation control register	**004015F (B)**	D7	**TCAOFF**	OSC1 auto-off function	1	On	0	Off	0	R/W	
		D6–3	–	reserved	–				–	–	0 when being read.
		D2	**TCHVOF**	CPU core power On/Off flag	1	Off	0	On	0	R/W	
		D1	**CLGRON**	Feedback resistor On/Off flag	1	Off	0	On	0	R/W	
		D0	–	reserved	–				1	–	Writing 0 not allowed.
R8 function select register	**00402CC (B)**	D7–4	–	reserved	–				–	–	0 when being read.
		D3	**CFR83**	R83 function selection	1	#DMAACK1	0	R83	0	W	Undefined in read.
		D2	**CFR82**	R82 function selection	1	#DMAACK0	0	R82	0	W	
		D1	**CFR81**	R81 function selection	1	FOSC1	0	R81	0	W	
		D0	–	–	–				–	–	0 when being read.
Port function extension register	**00402D8 (B)**	D7–3	–	reserved	–				–	–	Undefined in read.
		D2	**CFEX2**	R81 port extended function	1	#DRD	0	R81/FOSC1	0	W	Undefined in read.
		D1	**CFEX1**	P14 port extended function	1	#DWE	0	P14/#BUSGET	0	W	
		D0	**CFEX0**	P1[3:0] port extended function	1	DST/DPCO	0	P1x/T8UFx	0	W	

SOSC1: Low-speed (OSC1) oscillation control (D0) / Power control register (0x40140)

Turns the low-speed (OSC1) oscillation on or off.

Write "1": OSC1 oscillation turned on
Write "0": OSC1 oscillation turned off
Read: Valid

The oscillation of the low-speed (OSC1) oscillation circuit is stopped by writing "0" to SOSC1, and started again by writing "1".
Since a duration of maximum three seconds is required for oscillation to stabilize after the oscillation has been restarted, at least this length of time must pass before the OSC1 clock can be used.
Writing to SOSC1 is allowed only when CLGP[7:0] is set to "0b10010110". Note also that if the CPU is operating using the OSC1 clock, writing "0" to SOSC1 is ignored and the oscillation is not turned off.
At initial reset, SOSC1 is set to "1" (OSC1 oscillation turned on).

SOSC3: High-speed (OSC3) oscillation control (D1) / Power control register (0x40140)

Turns the high-speed (OSC3) oscillation on or off.

Write "1": OSC3 oscillation turned on
Write "0": OSC3 oscillation turned off
Read: Valid

The oscillation of the high-speed (OSC3) oscillation circuit is stopped by writing "0" to SOSC3, and started again by writing "1".
Since a duration of maximum 10 ms (for a 3.3-V crystal resonator) is required for oscillation to stabilize after the oscillation has been restarted, at least this length of time must pass before the OSC3 clock can be used.
Writing to SOSC3 is allowed only when CLGP[7:0] is set to "0b10010110". Note also that if the CPU is operating using the OSC3 clock, writing "0" to SOSC3 is ignored and the oscillation is not turned off.
At initial reset, SOSC3 is set to "1" (OSC3 oscillation turned on).

CLKCHG: CPU operating clock switch (D2) / Power control register (0x40140)

Selects the CPU operating clock.

Write "1": OSC3 clock
Write "0": OSC1 clock
Read: Valid

The OSC3 clock is selected as the CPU operating clock by writing "1" to CLKCHG, and OSC1 is selected by writing "0". The operating clock can be switched over in this way only when both the high-speed (OSC3) and low-speed (OSC1) oscillation circuits are on. In addition, writing to CLKCHG is effective only when CLGP[7:0] is set to "0b10010110". Immediately after the oscillation circuit has started oscillating, wait for the oscillation to stabilize before switching over the CPU operating clock.
At initial reset, CLKCHG is set to "1" (OSC3 clock).

CLKDT1–CLKDT0: CPU operating frequency selection (D[7:6]) / Power control register (0x40140)

Select the CPU operating clock frequency.

Table 10.1.6 Setting of CPU Operating Clock

CLKDT1	CLKDT0	Division ratio
1	1	$f_{OSC3}/8$
1	0	$f_{OSC3}/4$
0	1	$f_{OSC3}/2$
0	0	$f_{OSC3}/1$

This setting is effective when the CPU is operated using the high-speed (OSC3) clock and has no effect on the low-speed (OSC1) clock. Writing to CLKDT[1:0] is allowed only when CLGP[7:0] is set to "0b10010110".
At initial reset, CLKDT is set to "0" (f_{OSC3}/1).

8TION: High-speed (OSC3) oscillation waiting function (D2) / Clock option register (0x40150)

Sets the function for waiting until the high-speed (OSC3) oscillation stabilizes after SLEEP mode is exited.

Write "1": On
Write "0": Off
Read: Valid

After SLEEP mode is exited, the high-speed (OSC3) oscillation waiting function is effective by writing "1" to 8TION. For this function to be used, the waiting time must be set in 8-bit programmable timer 1 to allow it to start counting before entering SLEEP mode. After SLEEP mode is exited, the OSC3 clock is not supplied to the CPU until 8-bit programmable timer 1 underflows. This function will not work when 8TION is set to "0".
The high-speed (OSC3) oscillation waiting function is effective only when SLEEP mode is exited.
Writing to 8TION is effective only when CLGP[7:0] is set to "0b10010110".
When writing to 8TION, always be sure to write "0" to the reserved bits at address 0x40150.
At initial reset, 8TION is set to "0" (Off).

HLT2OP: HALT clock option (D3) / Clock option register (0x40150)

Select a HALT condition (basic mode or HALT2 mode).

Write "1": HALT2 mode
Write "0": Basic mode
Read: Valid

When "1" is written to HLT2OP, the CPU will enter HALT2 mode when the HALT instruction is executed. When "0" is written, the CPU will enter basic mode.
Writing to HLT2OP is allowed only when CLGP[7:0] is set to "0b10010110".
At initial reset, HLT2OP is set to "0" (basic mode).
The following shows the operating status in HALT mode (basic mode and HALT2 mode) and SLEEP mode.

Table 10.1.7 Operating Status in Standby Mode

Standby mode		Operating status	Reactivating factor
HALT mode	Basic mode	(1) The CPU clock is stopped. (CPU stop status) (2) BCU clock is supplied. (BCU run status) (3) Clocks for the peripheral circuits maintain the status before entering HALT mode. (run or stop) (4) The high-speed oscillation circuit maintains the status before entering HALT mode.	(1) Reset, NMI (2) Enabled (not masked) interrupt factors
	HALT2 mode	(1) The CPU clock is stopped. (CPU stop status) (2) BCU clock is stopped. (BCU stop status) (3) Clocks for the peripheral circuits maintain the status before entering HALT mode. (run or stop) (4) The high-speed oscillation circuit maintains the status before entering HALT mode.	
SLEEP mode		(1) The CPU clock is stopped. (CPU stop status) (2) BCU clock is stopped. (BCU stop status) (3) Clocks for the peripheral circuits are stopped. (4) The high-speed oscillation circuit is stopped.	(1) Reset, NMI (2) Enabled (not masked) input port interrupt factors (3) Clock timer interrupt when the low-speed oscillation circuit is being operated

PF1ON: OSC1 external output control (D0) / Clock option register (0x40150)

Turns the low-speed (OSC1) clock output to external devices on or off.

Write "1": On
Write "0": Off
Read: Valid

The low-speed (OSC1) clock is output from the FOSC1 pin to an external device by writing "1" to PF1ON.
However, for this setting to be effective, the R81 pin must be set for the FOSC1 pin by CFR81 and CFEX2.
The clock output is disabled by writing "0".
Writing to PF1ON is allowed only when CLGP[7:0] is set to "0b10010110".
At initial reset, PF1ON is set to "0" (Off).

CLGP7–CLGP0:Power-control register protection flag ([D[7:0]) / Power control protection register (0x4015E)

These bits remove the protection against writing to addresses 0x40140 and 0x40150.

Write "0b1010110": Write protection removed
Write other than the above: No operation (write-protected)
Read: Valid

Before writing to address 0x40140 or 0x40150, set CLGP[7:0] to "0b10010110" to remove the protection against writing to that address. This clearing of write protection is effective for only one writing, so the bits are cleared to "0b00000000" by one writing. Therefore, CLGP[7:0] must be set each time the protected address is written to.
At initial reset, CLGP is set to "0b00000000" (write-protected).

CLGRON: Feedback resistor ON/OFF flag (D1) / Low-speed oscillation control register (0x4015F)

Turns the internal feedback resistor of the low-speed (OSC1) oscillation circuit on or off.

Write "1":	Off
Write "0":	On
Read:	Valid

The feedback resistor of the low-speed (OSC1) oscillation circuit is disconnected by writing "1" to CLGRON. As a result, the circuit is configured for external clock input. If CLGRON = "0", the feedback resistor is effective and the OSC1 can be used as a crystal oscillation circuit.
At initial reset, CLGRON is set to "0" (On; crystal oscillation).

CFR81: R81 function selection (D1) / R8 function select register (0x402CC)

Selects the pin function of the R81 output port.

Write "1":	OSC1 clock output pin
Write "0":	Output port pin
Read:	Invalid

The R81 pin is set for OSC1 clock output (FOSC1) by writing "1" to CFR81.
As the CFR register is a write-only register (data is indeterminate when read), bit operation instructions (e.g., bset, bclr, or bnot) cannot be used for rewriting. Use ordinary storage instructions for this purpose.
At cold start, CFR81 is set to "0" (output port pin). At hot start, CFR81 retains its status from before the initial reset.

CFEX2: R81 extended function (D2) / Port function extension register (0x402D8)

Sets whether the function of the R81 pin is to be extended.

Write "1":	#DRD output pin
Write "0":	R81/FOSC1 output pin
Read:	Invalid

When CFEX2 is set to "1", the R81 pin functions as a DRAM read-signal #DRD output pin. When CFEX2 = "0", the CFR81 register becomes effective, so the settings of this register determine whether the R81 pin functions as an R81 output port or a FOUT1 output pin.
Since the CFEX2 register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.
At cold start, CFEX2 is set to "0" (R81/FOSC1 output pin). At hot start, CFEX2 retains its state from prior to the initial reset.

10.1.11 Programming Notes

(1) At initial reset, the low-speed (OSC1) oscillation circuit is configured for a crystal oscillation. For external clock input, use CLGRON to change the circuit configuration. This setting is required even for hot starts.

(2) Immediately after the oscillation circuit is turned on, a certain period of time is required for oscillation to stabilize (for OSC3 using a 3.3-V crystal resonator, this time is 10 ms max.; for OSC1, this time is 3 sec max. For details, refer to Section 18.8, "Oscillation Characteristics"). To prevent the device from operating erratically, do not use the clock until its oscillation has stabilized.
In particular, if the CPU is set in SLEEP mode during operation using the OSC3 clock, the high-speed (OSC3) oscillation circuit is turned off during in SLEEP mode and starts oscillating again after SLEEP mode is exited. To prevent the CPU from operating erratically at restart due to an unstable OSC3 clock, set a sufficient stabilization waiting time in 8-bit programmable timer 1 to turn on the oscillation stabilization waiting function after SLEEP mode is exited before entering SLEEP mode.

(3) The oscillation circuit used for the CPU operating clock cannot be turned off.

(4) The CPU operating clock can only be switched over when both oscillation circuits are on. Furthermore, when turning off an oscillation circuit that has become unnecessary as a result of the CPU operating clock switchover, be sure to use separate instructions for switchover and oscillation turnoff. If these two operations are processed simultaneously using one instruction, the CPU may operate erratically.

(5) If the high-speed (OSC3) oscillation circuit is turned off or the CPU operating clock is set for the OSC1 clock, all peripheral circuits operated using the OSC3 clock will be inactive.

(6) If the OSC3 clock is unnecessary, use the OSC1 clock to operate the CPU and turn the high-speed (OSC3) oscillation circuit off. This helps reduce current consumption.

10.2 Prescaler and Operating Clock for Peripheral Circuits

10.2.1 Configuration of Prescaler

The E0C33A104 contains a prescaler that divides the high-speed (OSC3) oscillation clock to generate the clocks for the internal peripheral circuits. The prescaler division ratio can be selected for each peripheral circuit in a program. A clock control circuit to control the clock supply to each peripheral circuit is also included.
The following are the peripheral circuits that use the output clock:

- 16-bit programmable timers 5 to 0
- 8-bit programmable timers 3 to 0
- A/D converter

Figure 10.2.1 shows the configuration of the prescaler.
For details on control of each peripheral circuit, refer to each corresponding chapter in this manual.

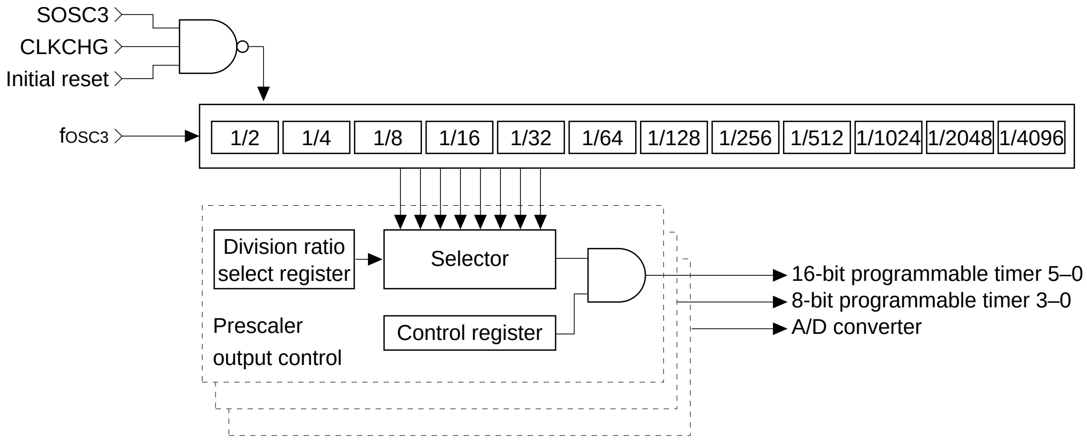


Figure 10.2.1 Configuration of Prescaler and Clock Control Circuit

10.2.2 Source Clock

The prescaler operates using the OSC3 clock generated by the high-speed (OSC3) oscillation circuit as its source clock. If the high-speed (OSC3) oscillation circuit is turned off (SOSC3 = "0"), the prescaler is in a reset state and does not perform the dividing operation. The same applies when the CPU is in SLEEP mode, because the high-speed (OSC3) oscillation circuit is inactive. In addition, when OSC1 is selected for the CPU operating clock (CLKCHG = "0"), clock supply to the prescaler is stopped.
Therefore, the 16-bit and 8-bit programmable timers and the A/D converter are operational only when the OSC3 clock is set for the CPU operating clock. The same also applies to the serial interface, as it uses the clock that outputs an 8-bit programmable timer (except when an external clock is used).

For details on how to control the high-speed (OSC3) oscillation circuit and how to switch over the CPU operating clock, refer to Section 10.1, "Oscillation Circuits".
At initial reset, the OSC3 clock is selected for the CPU operating clock.

The OSC3 clock is supplied to the prescaler by writing "1" to PSCON (D5/0x40140). At initial reset, PSCON is set to "1", so the prescaler is in an operating state. If all of said peripheral circuits can be turned off, stop the prescaler by writing "0" to PSCON. This helps to reduce current consumption.

10.2.3 Selecting Division Ratio and Output Control for Prescaler

The prescaler has registers for selecting the division ratio and clock output control separately for each peripheral circuit described above, allowing each peripheral circuit to be controlled.

The prescaler's division ratio can be selected from among eight ratios set for each peripheral circuit through the use of the division ratio selection register. The divided clock is output to the corresponding peripheral circuit by writing "1" to the clock control register.

Table 10.2.1 Control Bits of the Clock Control Registers

Peripheral circuit	Division ratio selection bit	Clock control bit
16-bit programmable timer 00	P16TS00[2:0] (D[2:0]/0x40147)∗1	P16TON00 (D3/0x40147)
16-bit programmable timer 01	P16TS01[2:0] (D[6:4]/0x40147)∗1	P16TON01 (D7/0x40147)
16-bit programmable timer 10	P16TS10[2:0] (D[2:0]/0x40148)∗2	P16TON10 (D3/0x40148)
16-bit programmable timer 11	P16TS11[2:0] (D[6:4]/0x40148)∗2	P16TON11 (D7/0x40148)
16-bit programmable timer 20	P16TS20[2:0] (D[2:0]/0x40149)∗1	P16TON20 (D3/0x40149)
16-bit programmable timer 21	P16TS21[2:0] (D[6:4]/0x40149)∗1	P16TON21 (D7/0x40149)
16-bit programmable timer 30	P16TS30[2:0] (D[2:0]/0x4014A)∗2	P16TON30 (D3/0x4014A)
16-bit programmable timer 31	P16TS31[2:0] (D[6:4]/0x4014A)∗2	P16TON31 (D7/0x4014A)
16-bit programmable timer 40	P16TS40[2:0] (D[2:0]/0x4014B)∗1	P16TON40 (D3/0x4014B)
16-bit programmable timer 41	P16TS41[2:0] (D[6:4]/0x4014B)∗1	P16TON41 (D7/0x4014B)
16-bit programmable timer 50	P16TS50[2:0] (D[2:0]/0x4014C)∗2	P16TON50 (D3/0x4014C)
16-bit programmable timer 51	P16TS51[2:0] (D[6:4]/0x4014C)∗2	P16TON51 (D7/0x4014C)
8-bit programmable timer 0	P8TS0[2:0] (D[2:0]/0x4014D)∗3	P8TON0 (D3/0x4014D)
8-bit programmable timer 1	P8TS1[2:0] (D[6:4]/0x4014D)∗4	P8TON1 (D7/0x4014D)
8-bit programmable timer 2	P8TS2[2:0] (D[2:0]/0x4014E)∗5	P8TON2 (D3/0x4014E)
8-bit programmable timer 3	P8TS3[2:0] (D[6:4]/0x4014E)∗3	P8TON3 (D7/0x4014E)
A/D converter	PSAD[2:0] (D[2:0]/0x4014F)∗3	PSONAD (D3/0x4014F)

∗1 to ∗5: See Table 10.2.2.

Table 10.2.2 Division Ratio

Bit setting	7	6	5	4	3	2	1	0
∗1	f_{OSC3}/4096	f_{OSC3}/1024	f_{OSC3}/256	f_{OSC3}/128	f_{OSC3}/64	f_{OSC3}/32	f_{OSC3}/16	f_{OSC3}/4
∗2	f_{OSC3}/2048	f_{OSC3}/512	f_{OSC3}/256	f_{OSC3}/128	f_{OSC3}/64	f_{OSC3}/32	f_{OSC3}/8	f_{OSC3}/2
∗3	f_{OSC3}/256	f_{OSC3}/128	f_{OSC3}/64	f_{OSC3}/32	f_{OSC3}/16	f_{OSC3}/8	f_{OSC3}/4	f_{OSC3}/2
∗4	f_{OSC3}/4096	f_{OSC3}/2048	f_{OSC3}/1024	f_{OSC3}/512	f_{OSC3}/256	f_{OSC3}/128	f_{OSC3}/64	f_{OSC3}/32
∗5	f_{OSC3}/4096	f_{OSC3}/2048	f_{OSC3}/64	f_{OSC3}/32	f_{OSC3}/16	f_{OSC3}/8	f_{OSC3}/4	f_{OSC3}/2

(f_{OSC3} = OSC3 oscillation frequency)

Current consumption can be reduced by turning off the clock output to the peripheral circuits that are unused among those listed above.

Note: In the following cases, the prescaler output clock may contain a hazard:

- If, when a clock is output, its division ratio is changed
- When the clock output is switched between on and off
- When the high-speed (OSC3) oscillation circuit is turned off or the CPU operating clock is switched over

Before performing these operations, make sure the 16-bit and 8-bit programmable timers and the A/D converter are turned off.

10.2.4 I/O Memory of Prescaler

Table 10.2.3 shows the control bits of the prescaler.

Table 10.2.3 Control Bits of Prescaler

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Power control register	0040140 (B)	D7	**CLKDT1**	System clock division ratio selection	CLKDT[1:0]			Division ratio	0	R/W	
		D6	**CLKDT0**		1	1		1/8	0		
					1	0		1/4			
					0	1		1/2			
					0	0		1/1			
		D5	**PSCON**	Prescaler On/Off control	1	On	0	Off	1	R/W	
		D4–3	–	reserved	–				0	–	Do not write 1.
		D2	**CLKCHG**	CPU operating clock switch	1	OSC3	0	OSC1	1	R/W	
		D1	**SOSC3**	High-speed (OSC3) oscillation On/Off	1	On	0	Off	1	R/W	
		D0	**SOSC1**	Low-speed (OSC1) oscillation On/Off	1	On	0	Off	1	R/W	
16-bit timer 0x clock control register	0040147 (B)	D7	**P16TON01**	16-bit timer 01 clock control	1	On	0	Off	0	R/W	
		D6	**P16TS012**	16-bit timer 01 clock division ratio selection	P16TS01[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D5	**P16TS011**		1	1	1	θ/4096	0		
		D4	**P16TS010**		1	1	0	θ/1024	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/16			
					0	0	0	θ/4			
		D3	**P16TON00**	16-bit timer 00 clock control	1	On	0	Off	0	R/W	
		D2	**P16TS002**	16-bit timer 00 clock division ratio selection	P16TS00[2:0]			Division ratio	0	R/W	θ: OSC3 clock 16-bit timer 0 can be used as a watchdog timer.
		D1	**P16TS001**		1	1	1	θ/4096	0		
		D0	**P16TS000**		1	1	0	θ/1024	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/16			
					0	0	0	θ/4			
16-bit timer 1x clock control register	0040148 (B)	D7	**P16TON11**	16-bit timer 11 clock control	1	On	0	Off	0	R/W	
		D6	**P16TS112**	16-bit timer 11 clock division ratio selection	P16TS11[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D5	**P16TS111**		1	1	1	θ/2048	0		
		D4	**P16TS110**		1	1	0	θ/512	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/8			
					0	0	0	θ/2			
		D3	**P16TON10**	16-bit timer 10 clock control	1	On	0	Off	0	R/W	
		D2	**P16TS102**	16-bit timer 10 clock division ratio selection	P16TS10[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D1	**P16TS101**		1	1	1	θ/2048	0		
		D0	**P16TS100**		1	1	0	θ/512	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/8			
					0	0	0	θ/2			

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 2x	**0040149**	D7	**P16TON21**	16-bit timer 21 clock control	1	On	0	Off	0	R/W	
clock control	**(B)**	D6	**P16TS212**	16-bit timer 21	P16TS21[2:0]			Division ratio	0	R/W	θ: OSC3 clock
register		D5	**P16TS211**	clock division ratio selection	1	1	1	θ/4096	0		
		D4	**P16TS210**		1	1	0	θ/1024	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/16			
					0	0	0	θ/4			
		D3	**P16TON20**	16-bit timer 20 clock control	1	On	0	Off	0	R/W	
		D2	**P16TS202**	16-bit timer 20	P16TS20[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D1	**P16TS201**	clock division ratio selection	1	1	1	θ/4096	0		
		D0	**P16TS200**		1	1	0	θ/1024	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/16			
					0	0	0	θ/4			
16-bit timer 3x	**004014A**	D7	**P16TON31**	16-bit timer 31 clock control	1	On	0	Off	0	R/W	
clock control	**(B)**	D6	**P16TS312**	16-bit timer 31	P16TS31[2:0]			Division ratio	0	R/W	θ: OSC3 clock
register		D5	**P16TS311**	clock division ratio selection	1	1	1	θ/2048	0		
		D4	**P16TS310**		1	1	0	θ/512	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/8			
					0	0	0	θ/2			
		D3	**P16TON30**	16-bit timer 30 clock control	1	On	0	Off	0	R/W	
		D2	**P16TS302**	16-bit timer 30	P16TS30[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D1	**P16TS301**	clock division ratio selection	1	1	1	θ/2048	0		
		D0	**P16TS300**		1	1	0	θ/512	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/8			
					0	0	0	θ/2			
16-bit timer 4x	**004014B**	D7	**P16TON41**	16-bit timer 41 clock control	1	On	0	Off	0	R/W	
clock control	**(B)**	D6	**P16TS412**	16-bit timer 41	P16TS41[2:0]			Division ratio	0	R/W	θ: OSC3 clock
register		D5	**P16TS411**	clock division ratio selection	1	1	1	θ/4096	0		
		D4	**P16TS410**		1	1	0	θ/1024	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/16			
					0	0	0	θ/4			
		D3	**P16TON40**	16-bit timer 40 clock control	1	On	0	Off	0	R/W	
		D2	**P16TS402**	16-bit timer 40	P16TS40[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D1	**P16TS401**	clock division ratio selection	1	1	1	θ/4096	0		
		D0	**P16TS400**		1	1	0	θ/1024	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/16			
					0	0	0	θ/4			

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 5x clock control register	004014C (B)	D7	**P16TON51**	16-bit timer 51 clock control	1	On	0	Off	0	R/W	
		D6	**P16TS512**	16-bit timer 51 clock division ratio selection	P16TS51[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D5	**P16TS511**		1	1	1	θ/2048	0		
		D4	**P16TS510**		1	1	0	θ/512	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/8			
					0	0	0	θ/2			
		D3	**P16TON50**	16-bit timer 50 clock control	1	On	0	Off	0	R/W	
		D2	**P16TS502**	16-bit timer 50 clock division ratio selection	P16TS50[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D1	**P16TS501**		1	1	1	θ/2048	0		
		D0	**P16TS500**		1	1	0	θ/512	0		
					1	0	1	θ/256			
					1	0	0	θ/128			
					0	1	1	θ/64			
					0	1	0	θ/32			
					0	0	1	θ/8			
					0	0	0	θ/2			
8-bit timer 0/1 clock control register	004014D (B)	D7	**P8TON1**	8-bit timer 1 clock control	1	On	0	Off	0	R/W	
		D6	**P8TS12**	8-bit timer 1 clock division ratio selection	P8TS1[2:0]			Division ratio	0	R/W	θ: OSC3 clock 8-bit timer 1 can generate the OSC3 oscillation-stabilize waiting period.
		D5	**P8TS11**		1	1	1	θ/4096	0		
		D4	**P8TS10**		1	1	0	θ/2048	0		
					1	0	1	θ/1024			
					1	0	0	θ/512			
					0	1	1	θ/256			
					0	1	0	θ/128			
					0	0	1	θ/64			
					0	0	0	θ/32			
		D3	**P8TON0**	8-bit timer 0 clock control	1	On	0	Off	0	R/W	
		D2	**P8TS02**	8-bit timer 0 clock division ratio selection	P8TS0[2:0]			Division ratio	0	R/W	θ: OSC3 clock 8-bit timer 0 can generate the DRAM refresh clock.
		D1	**P8TS01**		1	1	1	θ/256	0		
		D0	**P8TS00**		1	1	0	θ/128	0		
					1	0	1	θ/64			
					1	0	0	θ/32			
					0	1	1	θ/16			
					0	1	0	θ/8			
					0	0	1	θ/4			
					0	0	0	θ/2			
8-bit timer 2/3 clock control register	004014E (B)	D7	**P8TON3**	8-bit timer 3 clock control	1	On	0	Off	0	R/W	
		D6	**P8TS32**	8-bit timer 3 clock division ratio selection	P8TS3[2:0]			Division ratio	0	R/W	θ: OSC3 clock 8-bit timer 3 can generate the clock for the serial I/F Ch.1.
		D5	**P8TS31**		1	1	1	θ/256	0		
		D4	**P8TS30**		1	1	0	θ/128	0		
					1	0	1	θ/64			
					1	0	0	θ/32			
					0	1	1	θ/16			
					0	1	0	θ/8			
					0	0	1	θ/4			
					0	0	0	θ/2			
		D3	**P8TON2**	8-bit timer 2 clock control	1	On	0	Off	0	R/W	
		D2	**P8TS22**	8-bit timer 2 clock division ratio selection	P8TS2[2:0]			Division ratio	0	R/W	θ: OSC3 clock 8-bit timer 2 can generate the clock for the serial I/F Ch.0.
		D1	**P8TS21**		1	1	1	θ/4096	0		
		D0	**P8TS20**		1	1	0	θ/2048	0		
					1	0	1	θ/64			
					1	0	0	θ/32			
					0	1	1	θ/16			
					0	1	0	θ/8			
					0	0	1	θ/4			
					0	0	0	θ/2			
A/D clock control register	004014F (B)	D7–4	–	–	–				–	–	0 when being read.
		D3	**PSONAD**	A/D converter clock control	1	On	0	Off	0	R/W	
		D2	**PSAD2**	A/D converter clock division ratio selection	P8TS0[2:0]			Division ratio	0	R/W	θ: OSC3 clock
		D1	**PSAD1**		1	1	1	θ/256	0		
		D0	**PSAD0**		1	1	0	θ/128	0		
					1	0	1	θ/64			
					1	0	0	θ/32			
					0	1	1	θ/16			
					0	1	0	θ/8			
					0	0	1	θ/4			
					0	0	0	θ/2			

PSCON: Prescaler on/off control (D5) / Power control register (0x40140)

Turns the prescaler on or off.

Write "1": On
Write "0": Off
Read: Valid

The OSC3 clock is input to the prescaler by writing "1" to PSCON, thereby starting a dividing operation. However, before this can take effect, OSC3 must be set for the CPU operating clock.
The prescaler is turned off by writing "0". If the peripheral circuits do not need to be operated, write "0" to this bit to reduce current consumption.
At initial reset, PSCON is set to "1" (On).

P16TS00[2:0]: 16-bit timer 00 clock division ratio (D[2:0]) / 16-bit timer 0x clock control register (0x40147)
P16TS01[2:0]: 16-bit timer 01 clock division ratio (D[6:4]) / 16-bit timer 0x clock control register (0x40147)
P16TS10[2:0]: 16-bit timer 10 clock division ratio (D[2:0]) / 16-bit timer 1x clock control register (0x40148)
P16TS11[2:0]: 16-bit timer 11 clock division ratio (D[6:4]) / 16-bit timer 1x clock control register (0x40148)
P16TS20[2:0]: 16-bit timer 20 clock division ratio (D[2:0]) / 16-bit timer 2x clock control register (0x40149)
P16TS21[2:0]: 16-bit timer 21 clock division ratio (D[6:4]) / 16-bit timer 2x clock control register (0x40149)
P16TS30[2:0]: 16-bit timer 30 clock division ratio (D[2:0]) / 16-bit timer 3x clock control register (0x4014A)
P16TS31[2:0]: 16-bit timer 31 clock division ratio (D[6:4]) / 16-bit timer 3x clock control register (0x4014A)
P16TS40[2:0]: 16-bit timer 40 clock division ratio (D[2:0]) / 16-bit timer 4x clock control register (0x4014B)
P16TS41[2:0]: 16-bit timer 41 clock division ratio (D[6:4]) / 16-bit timer 4x clock control register (0x4014B)
P16TS50[2:0]: 16-bit timer 50 clock division ratio (D[2:0]) / 16-bit timer 5x clock control register (0x4014C)
P16TS51[2:0]: 16-bit timer 51 clock division ratio (D[6:4]) / 16-bit timer 5x clock control register (0x4014C)
P8TS0[2:0]: 8-bit timer 0 clock division ratio (D[2:0]) / 8-bit timer 0/1 clock control register (0x4014D)
P8TS1[2:0]: 8-bit timer 1 clock division ratio (D[6:4]) / 8-bit timer 0/1 clock control register (0x4014D)
P8TS2[2:0]: 8-bit timer 2 clock division ratio (D[2:0]) / 8-bit timer 2/3 clock control register (0x4014E)
P8TS3[2:0]: 8-bit timer 3 clock division ratio (D[6:4]) / 8-bit timer 2/3 clock control register (0x4014E)
PSAD[2:0]: A/D converter clock division ratio (D[2:0]) / A/D clock control register (0x4014F)

Select a clock for each peripheral circuit.
The desired division ratio can be selected from among the eight ratios shown on the I/O map. Note that the division ratio differs for each peripheral circuit.
These bits can also be read out.
At initial reset, all of these bits are set to "0b000" (highest frequency available).

P16TON00: 16-bit timer 00 clock control (D3) / 16-bit timer 0x clock control register (0x40147)
P16TON01: 16-bit timer 01 clock control (D7) / 16-bit timer 0x clock control register (0x40147)
P16TON10: 16-bit timer 10 clock control (D3) / 16-bit timer 1x clock control register (0x40148)
P16TON11: 16-bit timer 11 clock control (D7) / 16-bit timer 1x clock control register (0x40148)
P16TON20: 16-bit timer 20 clock control (D3) / 16-bit timer 2x clock control register (0x40149)
P16TON21: 16-bit timer 21 clock control (D7) / 16-bit timer 2x clock control register (0x40149)
P16TON30: 16-bit timer 30 clock control (D3) / 16-bit timer 3x clock control register (0x4014A)
P16TON31: 16-bit timer 31 clock control (D7) / 16-bit timer 3x clock control register (0x4014A)
P16TON40: 16-bit timer 40 clock control (D3) / 16-bit timer 4x clock control register (0x4014B)
P16TON41: 16-bit timer 41 clock control (D7) / 16-bit timer 4x clock control register (0x4014B)
P16TON50: 16-bit timer 50 clock control (D3) / 16-bit timer 5x clock control register (0x4014C)
P16TON51: 16-bit timer 51 clock control (D7) / 16-bit timer 5x clock control register (0x4014C)
P8TON0: 8-bit timer 0 clock control (D3) / 8-bit timer 0/1 clock control register (0x4014D)
P8TON1: 8-bit timer 1 clock control (D7) / 8-bit timer 0/1 clock control register (0x4014D)
P8TON2: 8-bit timer 2 clock control (D3) / 8-bit timer 2/3 clock control register (0x4014E)
P8TON3: 8-bit timer 3 clock control (D7) / 8-bit timer 2/3 clock control register (0x4014E)
PSONAD: A/D converter clock control (D3) / A/D clock control register (0x4014F)

Control the clock supply to each peripheral circuit.

Write "1": On
Write "0": Off
Read: Valid

The clock selected using the division ratio setup bits is output to the corresponding peripheral circuit by writing "1" to these bits. However, before this can take effect, the CPU operating clock must be set for OSC3.
The clock is not output by writing "0". If the peripheral circuits do not need to be operated, write "0" to these bits. This helps to reduce current consumption.
At initial reset, all of these bits are set to "0" (Off).

10.2.5 Programming Notes

(1) The prescaler operates only when the high-speed (OSC3) oscillation circuit is active and the OSC3 clock is set for the CPU operating clock. Note that the 16-bit and 8-bit programmable timers and the A/D converter do not operate when the high-speed (OSC3) oscillation circuit is inactive or when the OSC1 clock is selected for the CPU operating clock.

(2) In the following cases, the prescaler output clock may contain a hazard:
• If, during outputting of a clock, its division ratio is changed
• When the clock output is switched between on and off
• When the high-speed (OSC3) oscillation circuit is turned off or the CPU operating clock is switched over
Before performing these operations, make sure the 16-bit and 8-bit programmable timers and the A/D converter are turned off.

(3) When the 16-bit and 8-bit programmable timers and the A/D converter do not need to be operated, turn off the clock supply to those peripheral circuits. This helps to reduce current consumption.

11 Timers

The E0C33A104 contains four types of timers. This chapter explains the functions of each timer and how to control them.

11.1 Clock Timer

11.1.1 Configuration of Clock Timer

The E0C33A104 contains a clock timer that uses the low-speed (OSC1) oscillation circuit as the clock source. It consists of an 8-bit binary counter that is clocked by a 256-Hz signal derived from fOSC1, and second, minute, hour, and day counters, allowing all data (128 Hz to 1 Hz, seconds, minutes, hours, and day) to be read out in a software. It can also generate an interrupt using a 32-Hz, 8-Hz, 2-Hz, or 1-Hz (1-second) signal or when a one-minute, one-hour, or one-day count is up, in addition to generating an alarm at a specified time (minute or hour) or day.
The low-speed (OSC1) oscillation circuit and the clock timer can be kept operating even when the CPU and other internal peripheral circuits are placed in standby mode (HALT or SLEEP).
Normally, this clock timer should be used for a clock and various other clocking functions.
Figure 11.1.1 shows the structure of the clock timer.

Note: Since the clock timer is driven by a clock originating from the low-speed (OSC1) oscillation circuit, this timer cannot be used unless the low-speed (OSC1) oscillation circuit (32.768 kHz, Typ.) is used.

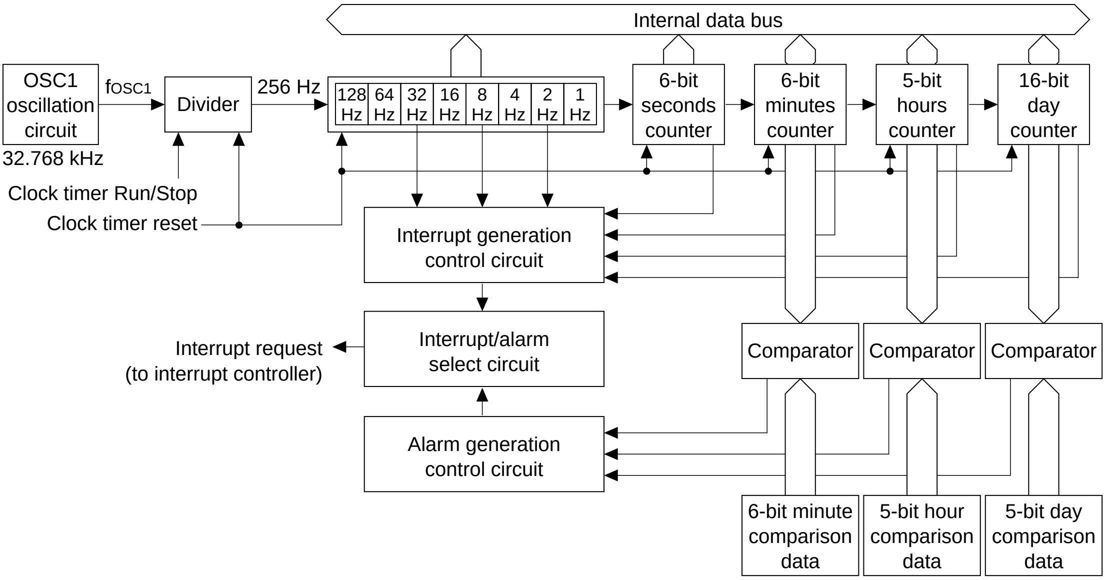


Figure 11.1.1 Structure of Clock Timer

11.1.2 Control and Operation of the Clock Timer

Initial setting

At initial reset, the clock timer's counter data, setup contents of alarms, and control bits including RUN/STOP, are not initialized. (This does not include the CPU core power on/off flag TCHVOF or OSC1 auto-off flag TCAOFF.)

Therefore, when using the clock timer, initialize it as follows:

1. Before you start setting up, stop the clock timer and disable the clock timer interrupt.
2. Reset the counters.
3. Preset the minute, hour, and day data (only when necessary).
4. Select an interrupt factor.
5. Select the alarm function.
6. Enable the interrupt.
7. Start the clock timer.

The following shows how to set and control each of the above. For details on interrupt control, refer to Section 11.1.3, "Interrupt Function".

Resetting the counters

Each counter of the clock timer can only be reset to "0" in the software. Note that they are not reset by an initial reset or the auto-off function.

To reset the clock timer, write "1" to TCRST (D1) / Clock timer Run/Stop register (0x40151). Note, however, that this reset input is accepted only when the clock timer is inactive, and is ignored when the timer is operating.

Notes:
- The clock timer reset bit TCRST and the clock timer RUN/STOP control bit TCRUN are located at the same address (0x40151). However, the clock timer cannot be reset at the same time it is set to RUN by writing "1" to both. In this case, the reset input is ignored and the timer starts counting up from the counter values then in effect. Always make sure TCRUN = "0" before resetting the timer.
- When the counters are cleared as the clock timer is reset, an interrupt may be generated depending on the timer settings. Therefore, first disable the clock timer interrupt before resetting the clock timer, and after resetting the clock timer, reset the interrupt factor flag, interrupt factor generation flag, and alarm factor generation flag.

Presetting minute, hour, and day data

The clock timer's minute, hour, and day counters have a data preset function, enabling the desired time and day to be set.

Table 11.1.1 Presetting the Counters

Counter	Data register	Preset value
Minute counter	TCHD[5:0] (D[5:0]) / Clock timer minute register (0x40155)	0 to 59
Hour counter	TCDD[4:0] (D[4:0]) / Clock timer hour register (0x40156)	0 to 23
Day counter	TCND[15:0] (D[7:0]) / Clock timer day (high-order) register (0x40158) (D[7:0]) / Clock timer day (low-order) register (0x40157)	0 to 65535

When using the clock timer as an RTC, be sure to set these counter values before starting operating of the clock timer. For the day counter, set a number of days starting from the reference day (e.g., January 1, 1990).

RUN/STOP the clock timer

The clock timer starts counting when "1" is written to TCRUN (D0) / Clock timer Run/Stop register (0x40151) and stops counting when "0" is written.

When the clock timer is made to RUN, the 256-Hz clock input is enabled at a falling edge of the low-speed (OSC1) oscillation clock pulse, and the 8-bit binary counter counts up at each falling edge of this 256-Hz clock. Figure 11.1.2 shows the operation of the 8-bit binary counter.

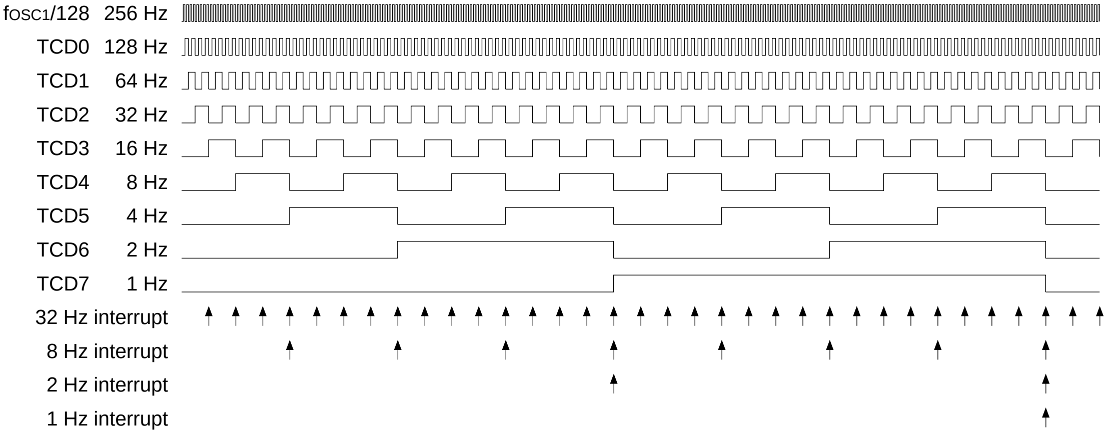


Figure 11.1.2 Timing Chart of 8-Bit Binary Counter

The 8-bit binary counter outputs a 1-Hz signal in its final stage.

The second counter counts the 1-Hz signal thus output. When it counts 60 seconds, the counter outputs a 60-second signal and is reset to 0 seconds.

Similarly, the minute and hour counters count 60 minutes and 24 hours, respectively, using the signals output by each preceding counter.

The day counter is a 16-bit binary counter and can count up to 65,536 days using the 24-hour signal output by the hour counter.

One of the following signals output by each counter can be selected to generate an interrupt:
32 Hz, 8 Hz, 2 Hz, 1 Hz (1 second), 1 minute, 1 hour, 1 day

If "0" is written to TCRUN, the clock timer is stopped at a rising edge of the low-speed (OSC1) oscillation clock to prevent device malfunction caused by the concurrent termination of counting (falling edge of the 256-Hz clock).

Even when the clock timer is stopped, each counter retains the data set at that point. When the timer is made to RUN again while in that state, each counter restarts counting from the retained value.

Reading out counter data

The data in each counter can be read out in a software as binary data.

Table 11.1.2 Reading Out Counter Data

Counter	Counter data
1 Hz to 128 Hz	TCD[7:0] (D[7:0]) / Clock timer divider register (0x40153)
Second counter	TCMD[5:0] (D[5:0]) / Clock timer second counter (0x40154)
Minute counter	TCHD[5:0] (D[5:0]) / Clock timer minute counter (0x40155)
Hour counter	TCDD[4:0] (D[4:0]) / Clock timer hour counter (0x40156)
Day counter	TCND[15:0] (D[7:0]) / Clock timer day (high-order) counter (0x40158) (D[7:0]) / Clock timer day (low-order) counter (0x40157)

Data is read directly from the counter during operation. For this reason, a counter can overflow while reading data from each counter, so the data thus read may not be exact. For example, if the 8-bit binary counter is read at 0xFF and then overflows before reading the next seconds counter, the value of the seconds counter is its count plus the one second that has elapsed since the 8-bit binary counter was read. To prevent this problem, try reading out each counter several times and make sure data has not been modified.

Setting alarm function

The clock timer has an alarm function, enabling an interrupt to be generated at a specified time and day. This specification can be made in minutes, hours, and days for each alarm or a combination of multiple alarms. Use TCASE[2:0] (D[4:2) / Clock timer interrupt control register (0x40152) for this specification.

Table 11.1.3 Alarm Factor Selection

TCASE2	TCASE1	TCASE0	Alarm factor
X	X	1	Minutes alarm
X	1	X	Hours alarm
1	X	X	Day alarm
0	0	0	None

For example, if TCASE is set to "001", only a minutes alarm is enabled and an alarm is generated at a specified minute every hour. If TCASE is set to "111", an alarm is generated on each specified day at each specified hour and minute. If alarms are not to be used, set TCASE to "000".
An interrupt can be generated every minute, every hour, and every day through the use of the counter's interrupt function instead of the alarm function.
To specify a day, hours, and minutes, use the registers shown below:

To specify minutes:	TCCH[5:0] (D[5:0]) / Minute-comparison data register (90x40159)	0 to 59 minutes*
To specify hours:	TCCD[4:0] (D[4:0]) / Hour-comparison data register (0x4015A)	0 to 23 hours*
To specify day:	TCCN4[4:0] (D[4:0]) / Day-comparison data register 0x4015B)	0 to 31 days after

* The minute-comparison data register (6 bits) and hour-comparison data register (5 bits) can be set for up to 63 minutes and 31 hours, respectively. Note that even when the data set in these registers exceeds 59 minutes or 23 hours, the data is not considered invalid.

The values set in these registers are compared with those of each counter, and when they match, the alarm factor generation flag TCAF (D0) / Clock timer interrupt control register (0x40152) is set to "1". If clock timer interrupts have been enabled using the interrupt controller, an interrupt is generated when the flag is set. The day-comparison data register is a 5-bit register, and its value is compared with the five low-order bits of the day counter. Therefore, an alarm can be generated for up to 31 days after the register is set.

11.1.3 Interrupt Function

Clock timer interrupt factors

The clock timer can generate an interrupt using a 32-Hz, 8-Hz, 2-Hz, 1-Hz (1-second), 1-minute, 1-hour, or 1-day signal. The interrupt factor to be used from among these signals can be selected using the interrupt factor selection bit TCISE[2:0] (D[7:5]) / Clock timer interrupt control register (0x40152).

Table 11.1.4 Selecting Interrupt Factor

TCISE2	TCISE1	TCISE0	Interrupt factor
1	1	1	None
1	1	0	1 day
1	0	1	1 hour
1	0	0	1 minute
0	1	1	1 Hz
0	1	0	2 Hz
0	0	1	8 Hz
0	0	0	32 Hz

An interrupt factor is generated at intervals of a selected signal (each falling edge of the signal).
If interrupts based on these signals are not to be used, set TCISE to "111".
When a selected interrupt factor is generated, the interrupt factor generation flag TCIF (D1) / Clock timer interrupt control register (0x40152) is set to "1". At the same time, the clock timer interrupt factor flag FCTM (D1) / Clock timer, A/D interrupt factor flag register (0x40287) also is set to "1". At this time, if the interrupt conditions set by the interrupt control registers are met, an interrupt to the CPU is generated.

An interrupt can be generated on a specified alarm day at a specified time as described in the preceding section. Interrupts generated by a signal and those generated by an alarm can both be used. However, since the clock timer has only one interrupt factor flag, it is the same interrupt that is generated by the timer. Therefore, if both types of interrupts are used, when an interrupt occurs, read the interrupt factor generation flag TCIF and alarm factor generation flag TCAF to determine which factor has generated the interrupt.
Once the factor generation flag is set to "1", it remains set until it is reset by writing "1" in the software. After confirming that the flag is set, write "1" to reset it.
The interrupt factor generation flag TCIF and alarm factor generation flag TCAF should be reset after at least 4 ms have passed from generation of an interrupt or an alarm.

Note: To prevent generation of an unwanted interrupt, disable the clock timer interrupt before selecting the interrupt and alarm factors. Then, before reenabling the interrupt, reset each factor generation flag and the interrupt factor flag.

Control registers of the interrupt controller

The following lists the clock timer interrupt control registers:

Interrupt factor flag:	FCTM (D1) / Clock timer, A/D interrupt factor flag register (0x40287)
Interrupt enable:	ECTM (D1) / Clock timer, A/D interrupt enable register (0x40277)
Interrupt level:	PCTM[2:0] (D[2:0]) / Clock timer interrupt priority register (0x40265)

When an interrupt factor occurs, the clock timer sets the interrupt actor flag to "1" as described above. At this time, if the interrupt enable register bit is set to "1", an interrupt request is generated.
Interrupts can be disabled by leaving the interrupt enable register bit reset to "0". The interrupt factor flag is always set to "1" when an interrupt factor is generated, regardless of the setting of the interrupt enable register (even when it is set to "0").
The interrupt priority register sets the priority levels (0 to 7) of interrupts. An interrupt request to the CPU is accepted on the condition that no other interrupt request has been generated that is of a higher priority.
It is only when the PSR's IE bit = "1" (interrupts enabled) and the set value of the IL is smaller than the clock timer interrupt level set by the interrupt priority register that a clock timer interrupt request is actually accepted by the CPU.
For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, refer to Section 8, "Interrupt".
Note that the clock timer interrupt factor does not have a function to invoke an intelligent DMA.

Trap vectors

The trap vector addresses for the clock-timer interrupt by default are set as shown below:
BTA3 = low: 0x0C00074

The trap table base address can be changed using the TTBR registers (0x48134 to 0x48137).

11.1.4 OSC1 Auto-Off Function

The clock timer has a function to stop the low-speed (OSC1) oscillation circuit by generating a selected alarm factor. This is effective for automatically stopping the clock timer when a certain time has elapsed after the CPU stopped.

To use this function, set the alarm setup registers to generate an alarm on a day and at a time at which the clock timer is to be automatically stopped, and write "1" to the OSC1 auto-off flag TCAOFF (D7) / Low-speed oscillation circuit control register (0x4015F).

The low-speed (OSC1) oscillation circuit and the clock timer are turned off on a specified day and at a specified time. Even after the clock timer has stopped, the counter values at the time of the stoppage are retained providing that the power-supply VDD1 remains on. When the device is initially reset while in this state, the low-speed (OSC1) oscillation circuit starts oscillating again, but the clock timer does not restart. It retains the counter values at the time of the stoppage. The OSC1 auto-off flag is set to "0" (auto-off disabled) at initial reset.

Furthermore, even when the clock timer is automatically stopped while the power supply VDD to the CPU and other peripheral circuits is turned off, the clock timer interrupt request signal remains active. Consequently, a clock-timer interrupt is generated when interrupts are enabled after the CPU has restarted. If this interrupt is unnecessary, reset the interrupt factor flag before enabling interrupts.

11.1.5 Retention of System (CPU) Power-Supply On/Off Data

To ensure that the clock timer can operate even when the system power supply is turned off, the power supply to the clock timer and the low-speed (OSC1) oscillation circuit can be disconnected from the system power supply.

To enable the above operation, write "1" to TCHVOF (D2) / Low-speed oscillation control register (0x4015F) before turning the system power supply off. This ensures that all clock timer control signals from the VDD block are set at a low value, so that the clock timer can operate without being affected by the system. Writing "1" to TCHVOF during normal operation causes the clock timer to become uncontrollable, so be careful.

11.1.6 Examples of Use of Clock Timer

The following shows examples of use of the clock timer and how to control the timer in each case.

To use the clock timer as a timer/counter

(1) Example in which while the CPU is inactive, the clock timer is kept operating in order to start again the CPU after a specified length of time has elapsed (e.g., three days):

1. Make sure the low-speed (OSC1) oscillation circuit is oscillating stably (SOSC1 = "1").
   Wait for approximately three seconds after the oscillation starts for its oscillation to stabilize.
2. Disable the clock timer interrupt using the interrupt controller (ECTM = "0").
3. Stop the clock timer and set "3 days" in the day-comparison register (TCRUN = "0", TCCN = "3").
4. Choose a "day-specified alarm" using the alarm-factor select bit and set "none" in the interrupt-factor select bit (TCASE = "100", TCISE = "111").
5. Reset the interrupt factor and alarm factor generation flags (FCTM = "0", TCAF = "0").
6. Reenable the clock timer interrupt using the interrupt controller (ECTM = "1").
7. Switch the CPU operating clock to the low-speed (OSC1) clock (CLKCHG = "0").
8. Turn off the high-speed (OSC3) oscillation circuit (SOSC3 = "0").
9. Reset the clock timer (TCRST = "0").
10. Start the clock timer (TCRUN = "1").
11. Execute the halt instruction to stop the CPU.

    :

    Wait until an interrupt is generated by a day-specified alarm from the clock timer. When an interrupt occurs, the CPU starts up using the OSC1 clock.

    :
12. If necessary, turn on the high-speed (OSC3) oscillation circuit and change the CPU operating clock back to the OSC3 clock.

In the above example, if the device is reset before a three-day period has elapsed, the device operates as follows:

• The CPU starts up using the OSC3 clock.

• The clock timer counters are not reset. They remain in the RUN state.

The time during which the CPU has been idle can be checked by reading out the clock timer counters.

(2) Example in which while the CPU is inactive (the system power supply is turned off), the clock timer is kept operating, and when this state continues for a specified length of time (e.g., three days), the low-speed (OSC1) oscillation circuit and the clock timer are automatically stopped:

1. Make sure the low-speed (OSC1) oscillation circuit is oscillating stably (SOSC1 = "1").
   Wait for approximately three seconds after the oscillation starts for its oscillation to stabilize.
2. Disable the clock timer interrupt using the interrupt controller (ECTM = "0").
3. Stop the clock timer and set "3 days" in the day-comparison register (TCRUN = "0", TCCN = "3").
4. Select a "day-specified alarm" using the alarm factor select bit and set "none" in the interrupt factor select bit (TCASE = "100", TCISE = "111").
5. Reset the interrupt factor and alarm factor generation flags (FCTM = "0", TCAF = "0").
6. Reenable the clock timer interrupt using the interrupt controller (ECTM = "1").
7. Switch the CPU operating clock to the low-speed (OSC1) clock (CLKCHG = "0").
8. Turn off the high-speed (OSC3) oscillation circuit (SOSC3 = "0").
9. Reset the clock timer (TCRST = "0").
10. Start the clock timer (TCRUN = "1").
11. Set the CPU core power-supply on/off flag to off to allow the system to stop (TCHVOF = "1").
12. Stop the system and turn off the system power supply.

    :

    The clock timer continues operating. When this state continues for three days, the low-speed (OSC1) oscillation circuit and the clock timer are made to stop by the auto-off function, which is triggered by a day-specified alarm.

    :

13. Start up the CPU by resetting the device.

In the above example, if the device is reset before a three-day period has elapsed, the device operates as follows:

- The CPU starts up using the OSC3 clock.
- The clock timer counters are not reset. They remain in the RUN state.
- The alarm function is not cleared, but the auto-off function is cleared by a reset.

The time during which the CPU has been idle can be checked by reading out the clock timer counters.

For using the clock timer as RTC

Example in which the clock timer is kept operating and an alarm is generated at 10:00 A.M. every day:

1. Disable the clock timer interrupt using the interrupt controller (ECTM = "0").
2. Stop the clock timer (TCRUN = "0").
3. Reset the clock timer (TCRST = "1").
4. Set the current day and time in the minute (TCHD), hour (TCDD), and day (TCND) counters. For the day counter, set a number of days starting from the reference day (e.g., January 1, 1990). When the count is read, it is converted into the current date by the software.
5. Set "10:00" in the hour-compare register (TCCD = "0x0A").
6. Select an a "hour-specified alarm" using the alarm factor select bit, and set "none" in the interrupt factor select bit (TCASE = "010", TCISE = "111").
7. Reset the interrupt factor and alarm-factor generation flags (FCTM = "1", TCAF = "0").
8. Reenable the clock timer interrupt using the interrupt controller (ECTM = "1").
9. Start the clock timer (TCRUN = "1").

    :

    The clock timer is made to generate an interrupt at 10:00 every day by an hour-specified alarm.

    :

In the above example, if any interrupt factor other than an alarm is selected, an interrupt is also generated by that interrupt factor. To determine which factor caused the interrupt generated, read the interrupt factor generation flag TCIF and alarm factor generation flag TCAF. If TCAF is set to 1, the interrupt has been caused by an alarm. If you select an interrupt factor (other than a 1-day factor) along with the hour-specified alarm, the selected interrupt factor occurs at the same time as the alarm factor.

11.1.7 I/O Memory of Clock Timer

Table 11.1.5 shows the control bits of the clock timer.

Table 11.1.5 Control Bits of Clock Timer

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Clock timer Run/Stop register	0040151 (B)	D7–2	–	reserved	–				–	–	0 when being read.
		D1	TCRST	Clock timer reset	1	Reset	0	Invalid	X	W	0 when being read.
		D0	TCRUN	Clock timer Run/Stop control	1	Run	0	Stop	X	R/W	
Clock timer interrupt control register	0040152 (B)	D7	TCISE2	Clock timer interrupt factor selection	TCISE[2:0]			Interrupt factor	X	R/W	
		D6	TCISE1		1	1	1	None	X		
		D5	TCISE0		1	1	0	Day	X		
					1	0	1	Hour			
					1	0	0	Minute			
					0	1	1	1 Hz			
					0	1	0	2 Hz			
					0	0	1	8 Hz			
					0	0	0	32 Hz			
		D4	TCASE2	Clock timer alarm factor selection	TCASE[2:0]			Alarm factor	X	R/W	
		D3	TCASE1		1	X	X	Day	X		
		D2	TCASE0		X	1	X	Hour	X		
					X	X	1	Minute			
					0	0	0	None			
		D1	TCIF	Interrupt factor generation flag	1	Generated	0	Not generated	X	R/W	Reset by writing 1.
		D0	TCAF	Alarm factor generation flag	1	Generated	0	Not generated	X	R/W	Reset by writing 1.
Clock timer divider register	0040153 (B)	D7	TCD7	Clock timer data 1 Hz	1	High	0	Low	X	R	
		D6	TCD6	Clock timer data 2 Hz	1	High	0	Low	X	R	
		D5	TCD5	Clock timer data 4 Hz	1	High	0	Low	X	R	
		D4	TCD4	Clock timer data 8 Hz	1	High	0	Low	X	R	
		D3	TCD3	Clock timer data 16 Hz	1	High	0	Low	X	R	
		D2	TCD2	Clock timer data 32 Hz	1	High	0	Low	X	R	
		D1	TCD1	Clock timer data 64 Hz	1	High	0	Low	X	R	
		D0	TCD0	Clock timer data 128 Hz	1	High	0	Low	X	R	
Clock timer second register	0040154 (B)	D7–6	–	reserved	–				–	–	0 when being read.
		D5	TCMD5	Clock timer second counter data	0 to 59 seconds				X	R	
		D4	TCMD4	TCMD5 = MSB					X		
		D3	TCMD3	TCMD0 = LSB					X		
		D2	TCMD2						X		
		D1	TCMD1						X		
		D0	TCMD0						X		
Clock timer minute register	0040155 (B)	D7–6	–	reserved	–				–	–	0 when being read.
		D5	TCHD5	Clock timer minute counter data	0 to 59 minutes				X	R/W	
		D4	TCHD4	TCHD5 = MSB					X		
		D3	TCHD3	TCHD0 = LSB					X		
		D2	TCHD2						X		
		D1	TCHD1						X		
		D0	TCHD0						X		
Clock timer hour register	0040156 (B)	D7–5	–	reserved	–				–	–	0 when being read.
		D4	TCDD4	Clock timer hour counter data	0 to 23 hours				X	R/W	
		D3	TCDD3	TCDD4 = MSB					X		
		D2	TCDD2	TCDD0 = LSB					X		
		D1	TCDD1						X		
		D0	TCDD0						X		
Clock timer day (low-order) register	0040157 (B)	D7	TCND7	Clock timer day counter data	0 to 65535 days				X	R/W	
		D6	TCND6	(low-order 8 bits)					X		
		D5	TCND5	TCND0 = LSB					X		
		D4	TCND4						X		
		D3	TCND3						X		
		D2	TCND2						X		
		D1	TCND1						X		
		D0	TCND0						X		
Clock timer day (high-order) register	0040158 (B)	D7	TCND15	Clock timer day counter data					X	R/W	
		D6	TCND14	(high-order 8 bits)					X		
		D5	TCND13	TCND15 = MSB					X		
		D4	TCND12						X		
		D3	TCND11						X		
		D2	TCND10						X		
		D1	TCND9						X		
		D0	TCND8						X		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock timer minute comparison register	**0040159 (B)**	D7–6	–	reserved	–	–	–	0 when being read.
		D5	**TCCH5**	Clock timer minute comparison data TCCH5 = MSB TCCH0 = LSB	0 to 59 minutes (Note) Can be set within 0–63.	X	R/W	
		D4	**TCCH4**			X		
		D3	**TCCH3**			X		
		D2	**TCCH2**			X		
		D1	**TCCH1**			X		
		D0	**TCCH0**			X		
Clock timer hour comparison register	**004015A (B)**	D7–5	–	reserved	–	–	–	0 when being read.
		D4	**TCCD4**	Clock timer hour comparison data TCCD4 = MSB TCCD0 = LSB	0 to 23 hours (Note) Can be set within 0–31.	X	R/W	
		D3	**TCCD3**			X		
		D2	**TCCD2**			X		
		D1	**TCCD1**			X		
		D0	**TCCD0**			X		
Clock timer day comparison register	**004015B (B)**	D7–5	–	reserved	–	–	–	0 when being read.
		D4	**TCCN4**	Clock timer day comparison data TCCN4 = MSB TCCN0 = LSB	0 to 31 days	X	R/W	Compared with TCND[4:0].
		D3	**TCCN3**			X		
		D2	**TCCN2**			X		
		D1	**TCCN1**			X		
		D0	**TCCN0**			X		
Low-speed oscillation control register	**004015F (B)**	D7	**TCAOFF**	OSC1 auto-off function	1 On 0 Off	0	R/W	
		D6–3	–	reserved	–	–	–	0 when being read.
		D2	**TCHVOF**	CPU core power On/Off flag	1 Off 0 On	0	R/W	
		D1	**CLGRON**	Feedback resistor On/Off flag	1 Off 0 On	0	R/W	
		D0	–	reserved	–	1	–	Writing 0 not allowed.
Clock timer interrupt priority register	**0040265 (B)**	D7–3	–	reserved	–	–	–	Writing 1 not allowed.
		D2	**PCTM2**	Clock timer interrupt level	0 to 7	X	R/W	
		D1	**PCTM1**			X		
		D0	**PCTM0**			X		
Clock timer, A/D interrupt enable register	**0040277 (B)**	D7–2	–	reserved	–	–	–	Writing 0 not allowed.
		D1	**ECTM**	Clock timer	1 Enabled 0 Disabled	0	R/W	
		D0	**EADE**	A/D converter		0	R/W	
Clock timer, A/D interrupt factor flag register	**0040287 (B)**	D7–2	–	reserved	–	–	–	0 when being read.
		D1	**FCTM**	Clock timer	1 Factor is generated 0 No factor is generated	X	R/W	
		D0	**FADE**	A/D converter		X	R/W	

TCRST: Clock timer reset (D1) / Clock timer Run/Stop register (0x40151)

Resets the clock timer.

Write "1":	The clock timer is reset
Write "0":	Invalid
Read:	Always "0"

The clock timer is reset by writing "1" to TCRST when the timer is inactive. All timer counters are cleared to "0". The clock timer cannot be reset when in the RUN state, nor can it be reset at the same time it is made to RUN through the execution of one write to address 0x40151. (The clock timer is started, but not reset.) In this case, first reset the clock timer and then use another instruction to RUN the clock timer. When the counters are cleared as the clock timer is reset, an interrupt may be generated, depending on the register settings. Therefore, before resetting the clock timer, first disable the clock timer interrupt, and after resetting the clock timer, reset the interrupt factor flag and the interrupt factor and alarm factor generation flags.

Writing "0" to TCRST results in No Operation. Since this TCRST is a write-only bit, its value when read is always "0".

The clock timer is not reset by an initial reset.

TCRUN: Clock timer RUN/STOP control (D0) / Clock timer Run/Stop register (0x40151)

Controls the RUN/STOP of the clock timer.

Write "1":	RUN
Write "0":	STOP
Read:	Valid

The clock timer is made to start counting by writing "1" to the TCRUN register and made to stop by writing "0". The timer data is retained even in the STOP state. The timer can also be made to start counting from the retained data by changing its state from STOP to RUN.

The TCRUN register is not initialized at initial reset.

TCD7–TCD0: 1–128 Hz counter data (D[7:0]) / Clock timer divider register (0x40153)
TCMD5–TCMD0: Second counter data (D[5:0]) / Clock timer second register (0x40154)
TCHD5–TCHD0: Minute counter data (D[5:0]) / Clock timer minute register (0x40155)
TCDD4–TCDD0: Hour counter data (D[4:0]) / Clock timer hour register (0x40156)
TCND15–TCND0: Day counter data (D[7:0]) / Clock timer day (high-order) register (0x40158)
(D[7:0]) / Clock timer day (low-order) register (0x40157)

Data can be read out from each counter.
The minute, hour, and day counters allow data to be written to, in addition to being read out.
The 1–128 Hz counter and seconds counter are read-only, so writing to these registers is ignored.
The unused high-order bits at each address of the second, minute, and hour counter data are always "0" when read out.
The counter data is not initialized at initial reset.

TCCH5–TCCH0: Minute-comparison data (D[5:0]) / Clock timer minute-comparison register (0x40159)
TCCD4–TCCD0: Hour-comparison data (D[4:0]) / Clock timer hour-comparison register (0x4015A)
TCCN4–TCCN0: Day-comparison data (D[4:0]) / Clock timer day-comparison register (0x4015B)

Set a day on which and a time at which an alarm is to be generated.
The comparison data register corresponding to the alarm factor selected using the TCASE register is compared with the counter data, and when the data matches, an alarm interrupt request is generated.
The day-comparison data is compared with the 5 low-order bits of the day counter.
Each register can be read out.
Rewriting to these registers must be performed when no interrupt and alarm have generated or after at least 4 ms have passed from generation of interrupt or alarm.
These registers are not initialized at initial reset.

TCAOFF: OSC1 auto-off flag (D7) / Low-speed oscillation control register (0x4015F)

Sets the clock timer's auto-off function.

Write "1": Enabled
Write "0": Disabled
Read: Valid

The OSC1 auto-off function is enabled by writing "1" to TCAOFF, in which case, when the set alarm interrupt factor occurs, the low-speed (OSC1) oscillation circuit and the clock timer are stopped. The clock timer retains the counter data even when inactive.
Set an alarm before writing "1" to TCAOFF.
The OSC1 auto-off function is disabled by writing "0", in which case the low-speed (OSC1) oscillation circuit and the clock timer are not stopped even when an alarm interrupt factor occurs.
At initial reset, TCAOFF is set to "0" (disabled).

TCHVOF: CPU core power-supply ON/OFF flag (D2) / Low-speed oscillation control register (0x4015F)

Disconnects the system power supply.

Write "1": Off
Write "0": On
Read: Valid

The clock timer and the low-speed (OSC1) oscillation circuit are disconnected from the system power supply by writing "1" to TCHVOF, in which case the clock timer operates without being effected by control signals from the system power supply. This function allows the system power supply to be turned off while the clock timer continues operating. Control signals for the clock timer and the low-speed (OSC1) oscillation circuit are made effective by writing "0" to TCHVOF.
At initial reset, TCHVOF is set to "0" (On).

TCISE2–TCISE0: Interrupt factor selection (D[7:5]) / Clock timer interrupt control register (0x40152)

Selects the factor for which the clock timer interrupt is to be generated.

Table 11.1.6 Selecting Interrupt Factor

TCISE2	TCISE1	TCISE0	Interrupt factor
1	1	1	None
1	1	0	1 day
1	0	1	1 hour
1	0	0	1 minute
0	1	1	1 Hz
0	1	0	2 Hz
0	0	1	8 Hz
0	0	0	32 Hz

When the clock timer interrupt is enabled, an interrupt is generated cyclically at each falling edge of the selected signal. If you the interrupt caused by these factors is not be used set TCISE to "111".
Rewriting to these bits must be performed when no interrupt and alarm have generated or after at least 4 ms have passed from generation of interrupt or alarm.
TCISE is not initialized at initial reset.

TCASE2–TCASE0: Alarm factor select register (D[4:2]) / Clock timer interrupt control register (0x40152)

Selects the factor for which an alarm is to be generated.

Table 11.1.7 Selecting Alarm Factor

TCASE2	TCASE1	TCASE0	Alarm factor
X	X	1	Minute alarm
X	1	X	Hour alarm
1	X	X	Day alarm
0	0	0	None

Use the TCASE2, TCASE1, and TCASE0 bits to select a day, hour, and minute alarm, respectively. It is therefore possible to select multiple alarm factors. When one of these bits is set to "1", the contents of the comparison data register that corresponds to the selected alarm factor is compared with the counter. If the comparison data of all selected alarm factors matches the counter data, an alarm interrupt request is generated. The comparison data register from which the alarm factor is unselected by writing "0" is not compared with the counter data.
Rewriting to these bits must be performed when no interrupt and alarm have generated or after at least 4 ms have passed from generation of interrupt or alarm.
TCASE is not initialized at initial reset.

TCIF: Interrupt factor generation flag (D1) / Clock timer interrupt control register (0x40152)

Indicates whether an interrupt factor has occurred.

Read "1": Interrupt factor has occurred
Read "0": No interrupt factor has occurred
Write "1": Flag is reset
Write "0": Invalid

TCIF is set to "1" when an interrupt factor selected using TCISE occurs. Since there is only one source for the clock timer interrupt, use this flag to differentiate it from interrupts caused by an alarm.
Once set to "1", TCIF remains set until it is reset by writing "1".
Rewriting to this bit must be performed when no interrupt and alarm have generated or after at least 4 ms have passed from generation of interrupt or alarm.
TCIF is not initialized at initial reset.

TCAF: Alarm factor generation flag (D0) / Clock timer interrupt control register (0x40152)

Indicates whether an alarm factor has occurred.

Read "1":	Alarm factor has occurred
Read "0":	No alarm factor has occurred
Write "1":	Flag is reset
Write "0":	Invalid

TCAF is set to "1" when all alarm factors selected using the TCASE register occur. Since there is only one source for the clock timer interrupt, use this flag to differentiate it from interrupts due to other interrupt factors.
Once set to "1", TCAF remains set until it is reset by writing "1".
Rewriting to this bit must be performed when no interrupt and alarm have generated or after at least 4 ms have passed from generation of interrupt or alarm.
TCAF is not initialized at initial reset.

PCTM2–PCTM0: Clock timer interrupt level (D[2:0]) / Clock timer interrupt priority register (0x40265)

Sets the priority level of the clock timer interrupt between 0 and 7.
At initial reset, PCTM becomes indeterminate.

ECTM: Clock timer interrupt enable (D1) / Clock timer, A/D interrupt enable register (0x40277)

Enables or disables generation of an interrupt to the CPU.

Write "1":	Interrupt enabled
Write "0":	Interrupt disabled
Read:	Valid

This bit controls the clock timer interrupt. The interrupt is enabled by setting ECTM to "1" and is disabled by setting it to "0".
At initial reset, ECTM is set to "0" (interrupt disabled).

FCTM: Clock timer interrupt factor flag (D1) / Clock timer, A/D interrupt factor flag register (0x40287)

Indicates whether the clock timer interrupt factor has occurred.

When read

Read "1":	Interrupt factor has occurred
Read "0":	No interrupt factor has occurred

When written using the reset-only method (default)

Write "1":	Interrupt factor flag is reset
Write "0":	Invalid

When written using the read/write method

Write "1":	Interrupt flag is set
Write "0":	Interrupt flag is reset

FCTM is set to "1" when the selected interrupt factor or alarm factor occurs.
At this time, if the following conditions are met, an interrupt to the CPU is generated:
1. The corresponding interrupt enable register bit is set to "1".
2. No other interrupt request of a higher interrupt priority is generated.
3. The IE bit of the PSR is set to "1" (interrupt enabled).
4. The corresponding interrupt priority register is set to a value higher than the CPU interrupt level (IL).
The interrupt factor flag is always set to "1" when an interrupt factor occurs, no matter how the interrupt enable and interrupt priority registers are set.
For the next interrupt to be accepted after an interrupt has occurred, it is necessary that the interrupt factor flag be reset, and that the PSR be set again (by setting the IE bit to "1" after setting the IL to a value lower than the level indicated by the interrupt priority register, or by executing the reti instruction).

The interrupt factor flag can be reset only by writing to it in the software. Note that if the PSR is set again to accept generated interrupts (or if the reti instruction is executed) without the interrupt factor flag being reset, the same interrupt occurs again. Note also that the value to be written to reset the flag is "1" when the reset-only method (RSTONLY = "1") is used, and "0" when the read/write method (RSTONLY = "0") is used.
The FCTM flag becomes indeterminate at initial reset, so be sure to reset it in the software.

11.1.8 Programming Notes

(1) The low-speed (OSC1) oscillation circuit, which is the clock source for the clock timer, requires a muxmum of three seconds for its oscillation to stabilize after it is started up. Therefore, immediately after power-on, wait until the oscillation stabilizes before starting the clock timer.

(2) At initial reset, the clock timer counter data, the setup contents of alarms, and control bits, including RUN/STOP, are not initialized. (This does not include the CPU core power on/off flag TCHVOF and OSC1 auto-off flag TCAOFF.) Therefore, always initialize the clock timer in the software following power-on.

(3) The clock timer reset bit TCRST and the clock timer RUN/STOP control bit TCRUN are located at the same address (0x40151). However, the clock timer cannot be reset at the same time it is set to RUN by writing "1" to both. In this case, the reset input is ignored and the timer starts counting up from the counter values then in effect. When resetting the timer, always make sure TCRUN = "0" (timer stopped).

(4) When the counters are cleared as the clock timer is reset, an interrupt may be generated depending on the register settings. Therefore, before resetting the clock timer, first disable the clock timer interrupt and, after resetting the clock timer, reset the interrupt factor flag and the interrupt factor generation and alarm factor generation flags.

(5) To prevent generation of an unwanted interrupt, disable the clock timer interrupt before selecting the interrupt and alarm factors. Then, before reenabling the interrupt, reset each factor generation flag and the interrupt factor flag.

(6) The interrupt factor flag (FCTM) becomes indeterminate at initial reset. To prevent generation of an unwanted interrupt, be sure to reset the flag in a program.

(7) To prevent regeneration of interrupts with the same factor after an interrupt has occurred, be sure to reset the interrupt factor flag (FCTM) before setting the PSR again or executing the reti instruction.

(8) When a clock timer control register (address 0x40152, 0x40159, 0x4015A or 0x4015B) is rewritten within 4 msec after the clock timer's alarm or periodic interrupt has occurred, the same interrupt may occur again even though the interrupt factor flag in ITC and the factor generation flag in the clock timer are cleared in the clock timer interrupt service routine.
To avoid this problem, wait at least 4 msec after the clock timer interrupt has occurred before writing data to a clock timer control register (address 0x40152, 0x40159, 0x4015A or 0x4015B).
When using only one of the clock timer interrupt functions (alarm or periodic), this problem may be avoided since it is not necessary to reset the clock timer interrupt factor generation flag in address 0x40152. It is possible to use only the clock timer interrupt factor flag in ITC (bit 1 of 0x40287).

11.2 Watchdog Timer

11.2.1 Configuration of Watchdog Timer

The E0C33A104 incorporates a watchdog timer function to detect the CPU's crash.

This function is implemented through the use of the 16-bit programmable timer 0. When this function is enabled, an NMI (nonmaskable interrupt) is generated by an underflow signal from the 16-bit programmable timer 0 (generating intervals can be set through the use of software). The 16-bit programmable timer 0 set in the software so as not to generate the NMI, making it possible to detect a program crash that may not pass through this processing routine.

Figure 11.2.1 shows the block diagram of the watchdog timer.

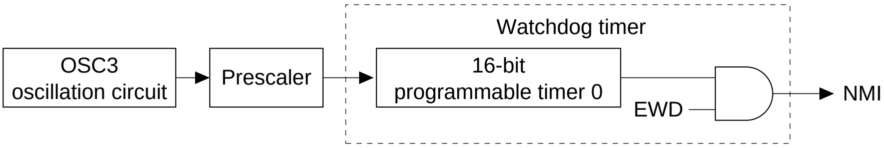


Figure 11.2.1 Watchdog Timer Block Diagram

11.2.2 Control of Watchdog Timer

Setting the operating clock and NMI generating interval

The watchdog timer is operated by the prescaler's output clock, which is generated from the high-speed (OSC3) oscillation clock. Therefore, the watchdog timer function cannot be used when the high-speed (OSC3) oscillation circuit is inactive or the CPU is operating using the low-speed (OSC1) oscillation clock.

The NMI is generated every time the 16-bit programmable timer 0 underflows. Therefore, this timing or interval is determined by the prescaler's P16TS00[2:0] (D[2:0]) / 16-bit timer 0x clock control register (0x40147), and the reload data that is preset in the 16-bit programmable timer 0, RR00[7:0] (D[7:0]) / 16-bit timer 00 reload data register (0x40182) and RR01[7:0] (D[7:0]) / 16-bit timer 01 reload data register (0x40183).

The 16-bit programmable timer can also be used in an 8-bit mode. In such a case, the 16-bit timer 01 functions as the watchdog timer. The NMI generating interval during the 8-bit mode is determined by the prescaler's P16TS01[2:0] (D[6:4]) / 16-bit timer 0x clock control register (0x40147) and the reload data that is preset in the 16-bit programmable timer 01, RR01[7:0] (D[7:0]) / 16-bit timer 01 reload data register (0x40183).

The explanation below applies to the 16-bit programmable timer 0.

The NMI generating interval is calculated using the following equation:

$$\text{NMI generating interval} = \frac{\text{RR0} + 1}{f_{\text{OSC3}} \times \text{dr}} \quad \text{[sec.]}$$

f_{OSC3}	High-speed (OSC3) oscillation frequency [Hz]
dr	Prescaler's division ratio set by the P16TS00 register in 16-bit mode (1/4096, 1/1024, 1/256, 1/128, 1/64, 1/32, 1/16, 1/4) Prescaler's division ratio set by the P16TS01 register in 8-bit mode (1/4096, 1/1024, 1/256, 1/128, 1/64, 1/32, 1/16, 1/4)
RR0	Set value of the RR0[1:0] register in 16-bit mode (0 to 65,535) Set value of the RR01 register in 8-bit mode (0 to 255)

For details on how to control the prescaler and the 16-bit programmable timer 0, refer to Section 10.2, "Prescaler and Operating Clock for Peripheral Circuits", and Section 11.4, "16-Bit Programmable Timers".

Setting the watchdog timer function

To use the watchdog timer function, enable the NMI that is generated by an underflow signal from the 16-bit programmable timer 0. For this purpose, use EWD (D1) / Watchdog timer enable register (0x40171).
The NMI is enabled by writing "1" to EWD. At initial reset, EWD is set to "0", so generation of the NMI is disabled.
To prevent an unwanted NMI from being generated by erroneous writing to EWD, this register is normally write-protected. To write-enable EWD, write "1" to WRWD (D7) / Watchdog timer write-protect register (0x40170). Only one writing to EWD is enabled in this way by the WRWD bit. When data is written to EWD after it is write-enabled, the WRWD bit is reset back to "0", thus making EWD write-protected again.

For the 16-bit programmable timer 0, set an appropriate preset value to make it start operating.

If the watchdog timer function is not to be used, set EWD to "0" and do not change it.

Presetting the watchdog timer

When using the watchdog timer, prepare a routine to preset the 16-bit programmable timer 0 before an NMI is generated in a location where it will be periodically processed. Make sure this routine is processed within the NMI generation interval described above.
The 16-bit programmable timer 0 (in 16-bit mode) and the 01 (in 8-bit mode) are preset by writing "1" to PSET00 (D1) / 16-bit timer 00 control register (0x40180) and PSET01 (D1) / 16-bit timer 01 control register (0x40181), respectively. At this point, the content of the reload data register is preset to the counter, and the timer starts counting the NMI generation interval over again from that point.

If the watchdog timer is not preset within the set interval for any reason, the CPU is made to enter trap processing by an NMI and starts executing the processing routine indicated by the NMI vector.
The NMI trap vector address is set by default as follows:
BTA3 = low: 0x0C0001C

The trap table base address can be changed using the TTBR registers (0x48134 to 0x48137).

11.2.3 Operation in Standby Modes

During HALT mode

In HALT mode (basic mode or HALT2 mode), the high-speed (OSC3) oscillation circuit, prescaler, and watchdog timer are operating. Consequently, if HALT mode continues beyond the NMI generation interval, HALT mode is cleared by the NMI.
To disable the watchdog timer in HALT mode, set EWD to "0" before executing the halt instruction or turn off the 16-bit programmable timer 0.
If the NMI is disabled by EWD, the 16-bit programmable timer 0 continues counting even in HALT mode. To reenable the NMI after clearing HALT mode, preset the 16-bit programmable timer 0 in advance.
If HALT mode was entered after the 16-bit programmable timer 0 was turned off, preset the timer before restarting it.

During SLEEP mode

In SLEEP mode, the high-speed (OSC3) oscillation circuit is turned off. Therefore, the watchdog timer also stops operating. To prevent generation of an unwanted NMI after clearing SLEEP mode, preset the 16-bit programmable timer 0 before executing the slp instruction. In addition, disable generation of the NMI by EWD as necessary.

11.2.4 I/O Memory of Watchdog Timer

Table 11.2.1 shows the control bits of the watchdog timer.

Table 11.2.1 Control Bits of Watchdog Timer

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Watchdog timer write-protect register	**0040170 (B)**	D7	**WRWD**	EWD write protection	1	Write enabled	0	Write-protect	0	W	
		D6–0	–	–	–				–	–	Read: Invalid
Watchdog timer enable register	**0040171 (B)**	D7–2	–	–	–				–	–	Read: Invalid
		D1	**EWD**	Watchdog timer enable	1	NMI enabled	0	NMI disabled	0	W	
		D0	–	–	–				–	–	Read: Invalid

WRWD: EWD write protection (D7) / Watchdog timer write-protect register (0x40170)

Enables writing to the EWD register.

Write "1": Writing enabled
Write "0": Writing disabled
Read: Invalid

The EWD bit is write-protected to prevent unwanted modifications. Writing to this bit is enabled for only one writing by setting WRWD to "1". WRWD is reset back to "0" by writing to EWD, so EWD is write-protected again.

If WRWD is reset to "0" when EWD is write-enabled (WRWD = "1"), EWD becomes write-protected again.

At initial reset, WRWD is set to "0" (write-protected).

EWD: NMI enable (D1) / Watchdog timer enable register (0x40171)

Controls the generation of a nonmaskable interrupt (NMI) by the watchdog timer.

Write "1": NMI is enabled
Write "0": NMI is disabled
Read: Invalid

The watchdog timer's interrupt signal is masked by writing "0" to EWD, so a nonmaskable interrupt (NMI) to the CPU is not generated. If EWD is set to "1", an NMI is generated when the 16-bit programmable timer 0 underflows.

Writing to EWD is valid only when WRWD = "1".

Even when EWD is set to "0", the 16-bit programmable timer 0 does not stop counting. Therefore, if the NMI has been temporarily disabled, be sure to preset the 16-bit programmable timer 0 before setting the EWD register back to "1".

At initial reset, EWD is set to "0" (NMI disabled).

11.2.5 Programming Notes

(1) If the watchdog timer's NMI is enabled, the watchdog timer must be preset in the software before the 16-bit programmable timer 0 underflows.

(2) Even when EWD is set to "0", the 16-bit programmable timer 0 does not stop counting. Therefore, if the NMI has been temporarily disabled, be sure to preset the 16-bit programmable timer 0 before setting EWD back to "1".

11.3 8-Bit Programmable Timers

11.3.1 Configuration of 8-Bit Programmable Timer

The E0C33A104 contains four channels of 8-bit programmable timers (timers 0 to 3).
Figure 11.3.1 shows the structure of the 8-bit programmable timer.

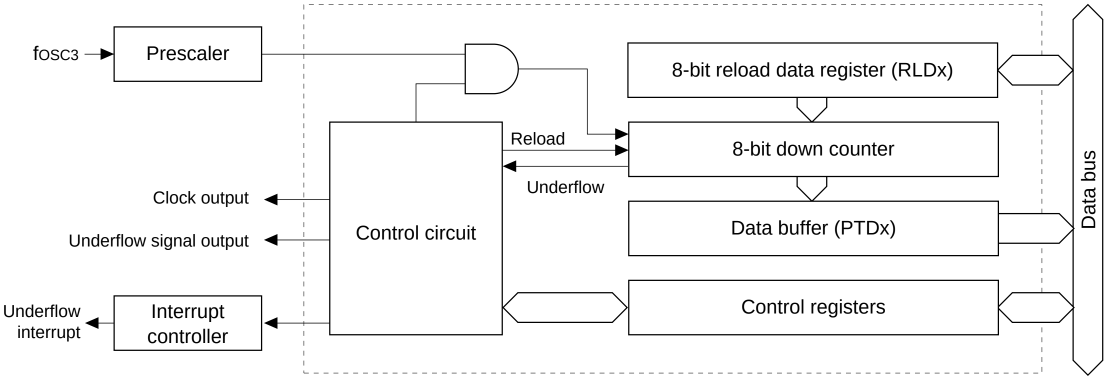


Figure 11.3.1 Structure of 8-Bit Programmable Timer

Each timer consists of an 8-bit presentable counter and can output a clock generated by the counter's underflow signal to the internal peripheral circuits or external devices. The output clock cycle can be selected from a wide range of cycles by setting the preset data that can be set in the software and the input clock in the prescaler.

11.3.2 Output Pins of 8-Bit Programmable Timers

Table 11.3.1 shows the pins that are used to output the underflow signals of the 8-bit programmable timers to external devices.

Table 11.3.1 Output Pins of 8-Bit Programmable Timers

Pin name	Pin No.		I/O	Pull-up	Function	Function select bit
	QFP5-128	QFP15-128				
P10/EXCL00/ T8UF0	46	46	I/O	Built-in	I/O port / Timer 00 event counter input / 8-bit timer 0 output	CFP10(D0)/P1 function select register (0x402D4)
P11/EXCL01/ T8UF1	47	47	I/O	Built-in	I/O port / Timer 01 event counter input / 8-bit timer 1 output	CFP11(D1/P1 function select register (0x402D4)
P12/EXCL10/ T8UF2	48	48	I/O	Built-in	I/O port / Timer 10 event counter input / 8-bit timer 2 output	CFP12(D2/P1 function select register (0x402D4)
P13/EXCL20/ T8UF3	49	49	I/O	Built-in	I/O port / Timer 20 event counter input / 8-bit timer 3 output	CFP13(D3/P1 function select register (0x402D4)

T8UFx (output pin of the 8-bit programmable timer)

This pin outputs a clock divided in each 8-bit programmable timer. The pulse width is equal to that of input clock of the 8-bit programmable timer (prescaler output). Therefore, the pulse width varies according to the prescaler setting.

How to set the output pins of the 8-bit programmable timer

All pins used by the 8-bit programmable timers are shared with I/O ports and the event counter inputs of the 16-bit programmable timers. At cold start, all these pins are set for I/O ports P1x (function select bit CFP1x = "0"). When using the clock output function of the 8-bit programmable timer, select the desired timer and write "1" to the function select bit CFP1x for the corresponding pin. The function select register is a write-only register (read data is indeterminate), so bit operation instructions (bset, bclr, and bnot) that accompany a read-modify-write operation cannot be used to rewrite this register. Use ordinary storage instructions for this purpose. At hot start, these pins retain their status from prior to the reset.

Then, after setting the above, write "1" to the I/O port's I/O control bit IOC1x (D[3:0]) / P1 I/O control register (0x402D6) to set to output mode. In input mode, the pin functions as the 16-bit programmable timer's event counter input and cannot be used to output a clock of the 8-bit programmable timer. At cold start, the register is set to input mode. At hot start, the register retains its status from prior to the reset.

Pull-up resistors

Each pin contains a pull-up resistor. However, if the pins are set for the output pins of the 8-bit programmable timer, the pull-up resistors are disconnected and do not serve their function.

11.3.3 Uses of 8-Bit Programmable Timers

The down-counter of the 8-bit programmable timer cyclically outputs an underflow signal according to the preset data that is set in the software. This underflow signal is used to generate an interrupt request to the CPU or to control the internal peripheral circuits. In addition, this signal can be output to external devices.

Furthermore, each 8-bit programmable timer generates a clock from the underflow signal by dividing it by 2, and the resulting clock is output to a specific internal peripheral circuit.

CPU interrupt request/IDMA invocation request

Each timer's underflow condition can be used as an interrupt factor to output an interrupt request to the CPU. Therefore, an interrupt can be generated at an interval that is set in the software.

This interrupt factor also can be used to invoke IDMA.

Clock output to external devices

The underflow signal can be output from the chip to the outside. This output can be used to control external devices. The output pins of each timer are described in the preceding section.

Control of and clock supply to internal peripheral circuits

The following describes the functions controlled by the underflow signal from the 8-bit programmable timer and the internal peripheral circuits that use the timer's output clock.

8-bit programmable timer 0

- DRAM refresh

  When the E0C33A104 has a DRAM directly connected to its external bus, the underflow signal from timer 0 can be used as a DRAM refresh request signal. This enables the intervals of the refresh cycle to be programmed.

  To use this function, write "1" to the BCU's control bit RPC (D9) / Bus control register (0x4812E) to enable the DRAM refresh.

- A/D conversion start trigger

  The A/D converter enables a trigger for starting the A/D conversion to be selected from among four available types. One of these is the underflow signal of the 8-bit programmable timer 0. This makes it possible to perform the A/D conversion at programmable intervals.

  To use this function, write "10" to the A/D converter control bit TS[1:0] (D[4:3]) / A/D trigger register (0x40242) to select the 8-bit programmable timer 0 as the trigger.

8-bit programmable timer 1

- Oscillation stabilization wait time of the high-speed (OSC3) oscillation circuit

  When SLEEP mode is cleared by an external interrupt, the high-speed (OSC3) oscillation circuit starts oscillating. To prevent the CPU from being operated erratically by an unstable clock before the oscillation stabilizes, the E0C33A104 enables setting of the waiting time before the CPU starts operating after SLEEP is cleared. Use the 8-bit programmable timer 1 to generate this waiting time. If the 8-bit programmable timer 1 is set so that the timer is actuated when the high-speed (OSC3) oscillation circuit starts oscillating the timer and, after the oscillation stabilization time elapses, an underflow signal is generated, then the CPU can be started up by that underflow signal.

  To use this function, write "0" to the oscillation circuit control bit 8TION (D2) / Clock option register (0x40150) to enable the oscillation stabilization waiting function.

8-bit programmable timer 2

- Clock supply to the Ch.0 serial interface

  When using the Ch.0 serial interface in the clock-synchronized master mode or the internal clock-based asynchronous mode, the output clock derived from the underflow signal of the 8-bit programmable timer 2 by dividing it by 2 is supplied to the serial interface as its operating clock. This enables the transfer rate of the serial interface to be programmed.

  To use this function, write "0" to the serial interface control bit SSCK0 (D2) / Serial I/F Ch.0 control register (0x401E3) to select the internal clock.

8-bit programmable timer 3

- Clock supply to the Ch.1 serial interface

  When using the Ch.1 serial interface in the clock-synchronized master mode or the internal clock-based asynchronous mode, the output clock derived from the underflow signal of the 8-bit programmable timer 3 by dividing it by 2 is supplied to the serial interface as its operating clock. This enables the transfer rate of the serial interface to be programmed.

  To use this function, write "0" to the serial interface control bit SSCK1 (D2) / Serial I/F Ch.1 control register (0x401E8) to select the internal clock.

11.3.4 Control and Operation of 8-Bit Programmable Timer

With the 8-bit programmable timer, the following settings must first be made before it starts counting:

1. Setting the output pin (only when necessary)
2. Setting the input clock
3. Setting the preset data (initial counter value)
4. Setting the interrupt/IDMA

Setting of an output pin is necessary only when the output clock of the 8-bit programmable timer is supplied to external devices. For details on how to set the pin, refer to Section 11.3.2, "Output Pins of 8-Bit Programmable Timers".
For details on how to set interrupts and IDMA, refer to Section 11.3.6, "8-Bit Programmable Timer Interrupts and DMA".

Note: The 8-bit programmable timers 0 through 3 all operate in the same way during counting, and the structure of their control registers is also the same. The control bit names are assigned the numerals "0" through "3" to denote the timer numbers. Since all these timers have common functions, timer numbers here are represented it is by "x" unless necessary to specify a timer number.

Setting the input clock

The 8-bit programmable timer is operated by the prescaler's output clock. The prescaler's division ratio can be selected for each timer.

	Division ratio select bit	Clock control bit	Register
8-bit timer 0:	P8TS0[2:0] (D2:0])	P8TON0 (D3)	8-bit timer 0/1 clock control register (0x4014D)
8-bit timer 1:	P8TS1[2:0] (D6:4])	P8TON1 (D7)	8-bit timer 0/1 clock control register (0x4014D)
8-bit timer 2:	P8TS2[2:0] (D2:0])	P8TON2 (D3)	8-bit timer 2/3 clock control register (0x4014E)
8-bit timer 3:	P8TS3[2:0] (D6:4])	P8TON3 (D7)	8-bit timer 2/3 clock control register (0x4014E)

Note that the division ratios differ for each timer.

Table 11.3.2 Input Clock Selection

Timer	P8TSx = 7	P8TSx = 6	P8TSx = 5	P8TSx = 4	P8TSx = 3	P8TSx = 2	P8TSx = 1	P8TSx = 0
Timer 0	f_{OSC3}/256	f_{OSC3}/128	f_{OSC3}/64	f_{OSC3}/32	f_{OSC3}/16	f_{OSC3}/8	f_{OSC3}/4	f_{OSC3}/2
Timer 1	f_{OSC3}/4096	f_{OSC3}/2048	f_{OSC3}/1024	f_{OSC3}/512	f_{OSC3}/256	f_{OSC3}/128	f_{OSC3}/64	f_{OSC3}/32
Timer 2	f_{OSC3}/4096	f_{OSC3}/2048	f_{OSC3}/64	f_{OSC3}/32	f_{OSC3}/16	f_{OSC3}/8	f_{OSC3}/4	f_{OSC3}/2
Timer 3	f_{OSC3}/256	f_{OSC3}/128	f_{OSC3}/64	f_{OSC3}/32	f_{OSC3}/16	f_{OSC3}/8	f_{OSC3}/4	f_{OSC3}/2

f_{OSC3}: OSC3 oscillation frequency

The selected clock is output from the prescaler to the 8-bit programmable timer by writing "1" to P8TONx.

Notes:

- The 8-bit programmable timer operates only when the prescaler is operating. The prescaler generates a clock for each timer from the OSC3 oscillation clock by dividing it as set in the register. When the CPU is operating using the low-speed (OSC1) clock, the prescaler is inactive, so the 8-bit programmable timer cannot be used. (Refer to Section 10.2, "Prescaler and Operating Clock for Peripheral Circuits".)
- The data set in the prescaler is the division ratio for the OSC3 oscillation frequency. If the CPU operating clock is generated from the OSC3 oscillation clock by dividing it using the CLKDT[1:0] register (D[7:6]/0x40140), do not use a clock that is faster than the CPU operating clock.
- When setting an input clock, make sure the 8-bit programmable timer is turned off.

Setting preset data (initial counter value)

Each timer has an 8-bit down-counter and a reload data register. The reload data register RLDx is used to set the initial value of the down-counter of each timer.
Timer 0 reload data: RLD0[7:0] (D[7:0]) / 8-bit timer 0 reload data register (0x40161)
Timer 1 reload data: RLD1[7:0] (D[7:0]) / 8-bit timer 1 reload data register (0x40165)
Timer 2 reload data: RLD2[7:0] (D[7:0]) / 8-bit timer 2 reload data register (0x40169)
Timer 3 reload data: RLD3[7:0] (D[7:0]) / 8-bit timer 3 reload data register (0x4016D)
The reload data registers can be read and written. At initial reset, the reload data registers are not initialized.

The data written to this register is preset in the down-counter, and the counter starts counting down from the preset value.
Data is thus preset in the down-counter in the following two cases:

1. **When it is preset in the software**
   Presetting in the software is performed using the preset control bit PSETx. When this bit is set to "1", the content of the reload data register is loaded into the down-counter at that point.
   Timer 0 preset: PSET0 (D1) / 8-bit timer 0 control register (0x40160)
   Timer 1 preset: PSET1 (D1) / 8-bit timer 1 control register (0x40164)
   Timer 2 preset: PSET2 (D1) / 8-bit timer 2 control register (0x40168)
   Timer 3 preset: PSET3 (D1) / 8-bit timer 3 control register (0x4016C)

2. **When the down-counter underflown during counting**
   Since the reload data is preset in the down-counter upon underflow, its underflow cycle is determined by the value that is set in the reload data register. This underflow signal controls each function described in the preceding section.

Before starting the 8-bit programmable timer, set the initial value in the reload data register and use the PSETx bit to preset the data in the down-counter.

The underflow cycle is determined by the prescaler setting and the reload data. The relationship between these two parameters is expressed by the following equation:

$$\text{Under flow cycle} = \frac{\text{RLDx} + 1}{f_{\text{OSC3}} \times \text{dr}} \text{ [sec.]}$$

f_{OSC3}: High-speed (OSC3) oscillation frequency [Hz]
dr: Prescaler division ratio set by P8TSx
RLDx: Set value of the RLDx register (0 to 255)

Timer RUN/STOP control

Each timer has a PTRUNx bit to control RUN/STOP.
Timer 0 RUN/STOP control: PTRUN0 (D0) / 8-bit timer 0 control register (0x40160)
Timer 1 RUN/STOP control: PTRUN1 (D0) / 8-bit timer 1 control register (0x40164)
Timer 2 RUN/STOP control: PTRUN2 (D0) / 8-bit timer 2 control register (0x40168)
Timer 3 RUN/STOP control: PTRUN3 (D0) / 8-bit timer 3 control register (0x4016C)

The timer is initiated to start counting down by writing "1" to PTRUNx. Writing "0" to PTRUNx disables the clock input and causes the timer to stop counting.
This RUN/STOP control does not affect the counter data. Even when the timer has stopped counting, the counter retains its count so that it can start counting again from that point.

When the terminal count is reached and the counter underflows, the initial value is reloaded from the reload data register into the counter.

When both the timer RUN/STOP control bit (PTRUNx) and the timer preset bit (PSETx) are set to "1" at the same time, the timer starts counting after presetting the reload register value into the counter.

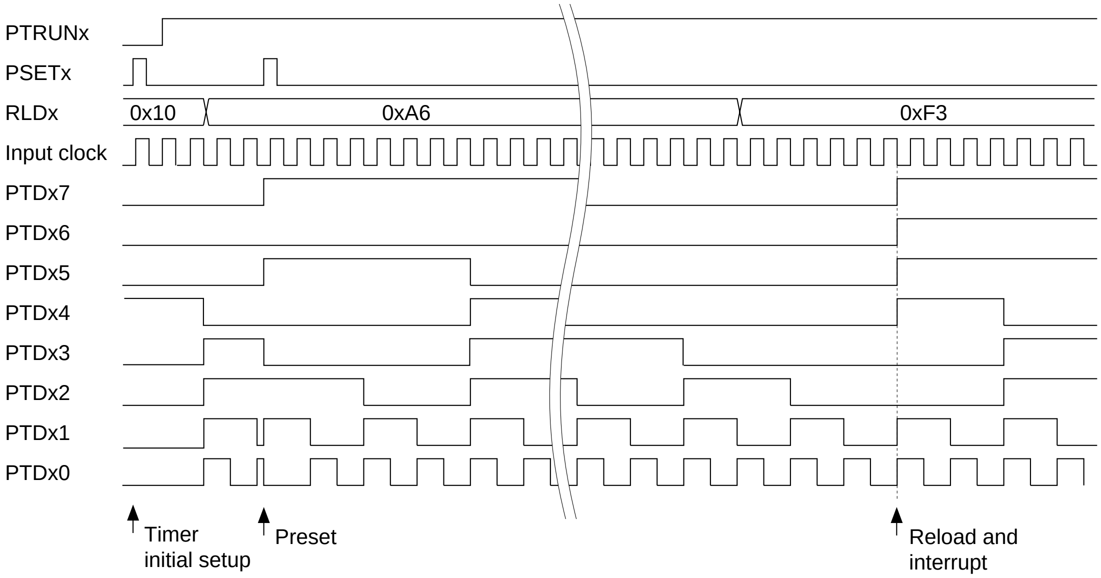


Figure 11.3.2 Basic Operation Timing of Counter

Reading out counter data

The counter data is read out via a PTDx data buffer. The counter data can be read out at any time.

Timer 0 data: PTD0[7:0] (D[7:0]) / 8-bit timer 0 counter data register (0x40162)
Timer 1 data: PTD1[7:0] (D[7:0]) / 8-bit timer 1 counter data register (0x40166)
Timer 2 data: PTD2[7:0] (D[7:0]) / 8-bit timer 2 counter data register (0x4016A)
Timer 3 data: PTD3[7:0] (D[7:0]) / 8-bit timer 3 counter data register (0x4016E)

11.3.5 Control of Clock Output

When outputting an underflow signal of the 8-bit programmable timer to external devices, or when supplying a clock generated by the underflow signal to the serial interface, it is necessary to control the clock output of the timer.

Timer 0 clock output control: PTOUT0 (D2) / 8-bit timer 0 control register (0x40160)
Timer 1 clock output control: PTOUT1 (D2) / 8-bit timer 1 control register (0x40164)
Timer 2 clock output control: PTOUT2 (D2) / 8-bit timer 2 control register (0x40168)
Timer 3 clock output control: PTOUT3 (D2) / 8-bit timer 3 control register (0x4016C)

To output the underflow signal/clock, write "1" to PTOUTx. If an output pin has been set (see Section 11.3.2), the underflow signal is output from that pin.

The same applies when timer 2 or 3 has been set as the clock source of the serial interface. A clock generated from the underflow signal by dividing it by 2 is output to the serial interface through this control. The clock output is turned off by writing "0" to PTOUTx, and the external output is fixed at "0" and the internal clock output is fixed at "1".

Figure 11.3.3 shows the waveforms of the output signals.

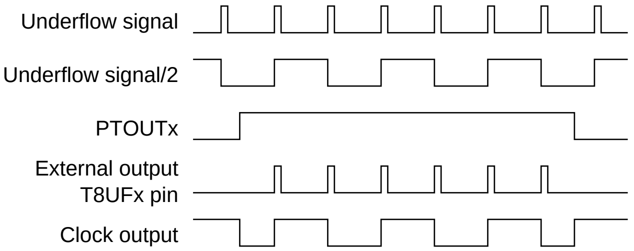


Figure 11.3.3 8-Bit Programmable Timer Output Waveform

The underflow signal's pulse width (duration of the high period) is equal to that of the timer's input clock (prescaler's output).

8-bit timer external output (P10–P13 ports)

1) After an initial reset (cold start), the ports (P10–P13) are set to general-purpose I/O ports with input mode (Hi-Z status).
2) The port (P10–P13) outputs "0" when it is set to the 8-bit timer output (timer output is off status).
3) The timer output is left as "0" when the timer output is turned on after setting the input clock and timer initial value.
4) When an underflow occurs after starting the timer, the port outputs a pulse with the same width as the 8-bit timer input clock pulse (prescaler's output).

11.3.6 8-Bit Programmable Timer Interrupts and DMA

The 8-bit programmable timer has a function to generate an interrupt based on the underflow state of each timer. The timing at which an interrupt is generated is shown in Figure 11.3.2 in the preceding section.

Control registers of the interrupt controller

Table 11.3.3 shows the interrupt controller's control register provided for each timer.

Table 11.3.3 Control Registers of Interrupt Controller

Timer	Interrupt factor flag	Interrupt enable register	Interrupt priority register	IDMA request register
Timer 0	F8TU0(D0/0x40285)	E8TU0(D0/0x40275)	P8TM[2:0](D[2:0]/0x40263)	R8TU0(D2/0x40292)
Timer 1	F8TU1(D1/0x40285)	E8TU1(D1/0x40275)		R8TU1(D3/0x40292)
Timer 2	F8TU2(D2/0x40285)	E8TU2(D2/0x40275)		R8TU2(D4/0x40292)
Timer 3	F8TU3(D3/0x40285)	E8TU3(D3/0x40275)		R8TU3(D5/0x40292)

When the timer underflows, the corresponding interrupt factor flag is set to "1". If the interrupt enable register bit corresponding to that interrupt factor flag has been set to "1", an interrupt request is generated.

An interrupt caused by a timer can be disabled by leaving the interrupt enable register bit for that timer set to "0". The interrupt factor flag is set to "1" whenever the timer underflows, regardless of how the interrupt enable register is set (even when it is set to "0").

The interrupt priority register sets an interrupt priority level (0 to 7) for the four timers as one interrupt source. Within 8-bit programmable timers, timer 0 has the highest priority and timer 3 the lowest. An interrupt request to the CPU is accepted on the condition that no other interrupt request of a higher priority has been generated.

It is only when the PSR's IE bit = "1" (interrupts enabled) and the set value of the IL is smaller than the timer interrupt level set by the interrupt priority register, that a timer interrupt request is actually accepted by the CPU.

For details on these interrupt control registers and device operation when an interrupt has occurred, refer to Chapter 8, "Interrupt".

Intelligent DMA

The underflow interrupt factor of each timer can also invoke intelligent DMA (IDMA). This enables memory-to-memory DMA transfers to be performed cyclically.

The following shows the IDMA channel numbers set to each timer:

	IDMA channel
Timer 0:	0x13
Timer 1:	0x14
Timer 2:	0x15
Timer 3:	0x16

For IDMA to be invoked, one of the IDMA request bits shown in Table 11.3.3 must be set to "1" in advance. Transfer conditions, etc. must also be set on the IDMA side in advance.

If the IDMA request bit is set to "1", IDMA is invoked through generation of an interrupt factor. No interrupt request is generated at that point. An interrupt request is generated after the DMA transfer is completed. The registers can also be set so as not to generate an interrupt, with only a DMA transfer performed.

For details on DMA transfers and interrupt control upon completion of DMA transfer, refer to Section 9.2, "Intelligent DMA".

Trap vectors

The trap vector addresses for individual underflow interrupt factors are set by default as shown below:

	BTA3 = low
Timer 0 underflow interrupt:	0x0C000D0
Timer 1 underflow interrupt:	0x0C000D4
Timer 2 underflow interrupt:	0x0C000D8
Timer 3 underflow interrupt:	0x0C000DC

The base address of the trap table can be changed using the TTBR register (0x48134 to 0x48137).

11.3.7 I/O Memory of 8-Bit Programmable Timers

Table 11.3.4 shows the control bits of the 8-bit programmable timers.

For details on the I/O memory of the prescaler used to set a clock, refer to Section 10.2.4, "I/O Memory of Prescaler".

Table 11.3.4 Control Bits of 8-Bit Programmable Timer

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit timer 0 control register	0040160 (B)	D7–3	–	reserved	–				–	–	0 when being read.
		D2	**PTOUT0**	8-bit timer 0 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET0**	8-bit timer 0 preset	1	Preset	0	Invalid	–	W	0 when being read.
		D0	**PTRUN0**	8-bit timer 0 Run/Stop control	1	Run	0	Stop	0	R/W	
8-bit timer 0 reload data register	0040161 (B)	D7	**RLD07**	8-bit timer 0 reload data RLD07 = MSB RLD00 = LSB	0 to 255				X	R/W	
		D6	**RLD06**						X		
		D5	**RLD05**						X		
		D4	**RLD04**						X		
		D3	**RLD03**						X		
		D2	**RLD02**						X		
		D1	**RLD01**						X		
		D0	**RLD00**						X		
8-bit timer 0 counter data register	0040162 (B)	D7	**PTD07**	8-bit timer 0 counter data PTD07 = MSB PTD00 = LSB	0 to 255				X	R	
		D6	**PTD06**						X		
		D5	**PTD05**						X		
		D4	**PTD04**						X		
		D3	**PTD03**						X		
		D2	**PTD02**						X		
		D1	**PTD01**						X		
		D0	**PTD00**						X		
8-bit timer 1 control register	0040164 (B)	D7–3	–	reserved	–				–	–	0 when being read.
		D2	**PTOUT1**	8-bit timer 1 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET1**	8-bit timer 1 preset	1	Preset	0	Invalid	–	W	0 when being read.
		D0	**PTRUN1**	8-bit timer 1 Run/Stop control	1	Run	0	Stop	0	R/W	
8-bit timer 1 reload data register	0040165 (B)	D7	**RLD17**	8-bit timer 1 reload data RLD17 = MSB RLD10 = LSB	0 to 255				X	R/W	
		D6	**RLD16**						X		
		D5	**RLD15**						X		
		D4	**RLD14**						X		
		D3	**RLD13**						X		
		D2	**RLD12**						X		
		D1	**RLD11**						X		
		D0	**RLD10**						X		
8-bit timer 1 counter data register	0040166 (B)	D7	**PTD17**	8-bit timer 1 counter data PTD17 = MSB PTD10 = LSB	0 to 255				X	R	
		D6	**PTD16**						X		
		D5	**PTD15**						X		
		D4	**PTD14**						X		
		D3	**PTD13**						X		
		D2	**PTD12**						X		
		D1	**PTD11**						X		
		D0	**PTD10**						X		
8-bit timer 2 control register	0040168 (B)	D7–3	–	reserved	–				–	–	0 when being read.
		D2	**PTOUT2**	8-bit timer 2 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET2**	8-bit timer 2 preset	1	Preset	0	Invalid	–	W	0 when being read.
		D0	**PTRUN2**	8-bit timer 2 Run/Stop control	1	Run	0	Stop	0	R/W	
8-bit timer 2 reload data register	0040169 (B)	D7	**RLD27**	8-bit timer 2 reload data RLD27 = MSB RLD20 = LSB	0 to 255				X	R/W	
		D6	**RLD26**						X		
		D5	**RLD25**						X		
		D4	**RLD24**						X		
		D3	**RLD23**						X		
		D2	**RLD22**						X		
		D1	**RLD21**						X		
		D0	**RLD20**						X		
8-bit timer 2 counter data register	004016A (B)	D7	**PTD27**	8-bit timer 2 counter data PTD27 = MSB PTD20 = LSB	0 to 255				X	R	
		D6	**PTD26**						X		
		D5	**PTD25**						X		
		D4	**PTD24**						X		
		D3	**PTD23**						X		
		D2	**PTD22**						X		
		D1	**PTD21**						X		
		D0	**PTD20**						X		

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit timer 3 control register	**004016C (B)**	D7–3	–	reserved	–				–	–	0 when being read.
		D2	**PTOUT3**	8-bit timer 3 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET3**	8-bit timer 3 preset	1	Preset	0	Invalid	–	W	0 when being read.
		D0	**PTRUN3**	8-bit timer 3 Run/Stop control	1	Run	0	Stop	0	R/W	
8-bit timer 3 reload data register	**004016D (B)**	D7	**RLD37**	8-bit timer 3 reload data RLD37 = MSB RLD30 = LSB	0 to 255				X	R/W	
		D6	**RLD36**						X		
		D5	**RLD35**						X		
		D4	**RLD34**						X		
		D3	**RLD33**						X		
		D2	**RLD32**						X		
		D1	**RLD31**						X		
		D0	**RLD30**						X		
8-bit timer 3 counter data register	**004016E (B)**	D7	**PTD37**	8-bit timer 3 counter data PTD37 = MSB PTD30 = LSB	0 to 255				X	R	
		D6	**PTD36**						X		
		D5	**PTD35**						X		
		D4	**PTD34**						X		
		D3	**PTD33**						X		
		D2	**PTD32**						X		
		D1	**PTD31**						X		
		D0	**PTD30**						X		
8-bit timer, serial I/F Ch.0 interrupt priority register	**0040263 (B)**	D7	–	reserved	–				–	–	0 when being read.
		D6	**PSI002**	Serial interface Ch.0 interrupt level	0 to 7				X	R/W	
		D5	**PSI001**						X		
		D4	**PSI000**						X		
		D3	–	reserved	–				–	–	0 when being read.
		D2	**P8TM2**	8-bit timer 0–3 interrupt level	0 to 7				X	R/W	
		D1	**P8TM1**						X		
		D0	**P8TM0**						X		
8-bit timer interrupt enable register	**0040275 (B)**	D7–4	–	reserved	–				–	–	0 when being read.
		D3	**E8TU3**	8-bit timer 3 underflow	1	Enabled	0	Disabled	0	R/W	
		D2	**E8TU2**	8-bit timer 2 underflow					0	R/W	
		D1	**E8TU1**	8-bit timer 1 underflow					0	R/W	
		D0	**E8TU0**	8-bit timer 0 underflow					0	R/W	
8-bit timer interrupt factor flag register	**0040285 (B)**	D7–4	–	reserved	–				–	–	0 when being read.
		D3	**F8TU3**	8-bit timer 3 underflow	1	Factor is generated	0	No factor is generated	X	R/W	
		D2	**F8TU2**	8-bit timer 2 underflow					X	R/W	
		D1	**F8TU1**	8-bit timer 1 underflow					X	R/W	
		D0	**F8TU0**	8-bit timer 0 underflow					X	R/W	
16-bit timer 5, 8-bit timer, serial I/F Ch.0 IDMA request register	**0040292 (B)**	D7	**RSTX0**	SIF Ch.0 transmit buffer empty	1	IDMA request	0	Interrupt request	X	R/W	
		D6	**RSRX0**	SIF Ch.0 receive buffer full					X	R/W	
		D5	**R8TU3**	8-bit timer 3 underflow					X	R/W	
		D4	**R8TU2**	8-bit timer 2 underflow					X	R/W	
		D3	**R8TU1**	8-bit timer 1 underflow					X	R/W	
		D2	**R8TU0**	8-bit timer 0 underflow					X	R/W	
		D1	**R16TC51**	16-bit timer 51 comparison match					X	R/W	
		D0	**R16TU51**	16-bit timer 51 underflow					X	R/W	
P1 function select register	**00402D4 (B)**	D7	–	reserved	–				–	–	
		D6	**CFP16**	P16 function selection	1	EXCL50 #DMAEND1	0	P16	0	W	Undefined in read.
		D5	**CFP15**	P15 function selection	1	EXCL40 #DMAEND0	0	P15	0	W	
		D4	**CFP14**	P14 function selection	1	EXCL30 #BUSGET	0	P14	0	W	
		D3	**CFP13**	P13 function selection	1	EXCL20 T8UF3	0	P13	0	W	
		D2	**CFP12**	P12 function selection	1	EXCL10 T8UF2	0	P12	0	W	
		D1	**CFP11**	P11 function selection	1	EXCL01 T8UF1	0	P11	0	W	
		D0	**CFP10**	P10 function selection	1	EXCL00 T8UF0	0	P10	0	W	
P1 I/O control register	**00402D6 (B)**	D7	–	reserved	–				–	–	
		D6	**IOC16**	P16 I/O control	1	Output	0	Input	0	W	Undefined in read.
		D5	**IOC15**	P15 I/O control					0	W	
		D4	**IOC14**	P14 I/O control					0	W	
		D3	**IOC13**	P13 I/O control					0	W	
		D2	**IOC12**	P12 I/O control					0	W	
		D1	**IOC11**	P11 I/O control					0	W	
		D0	**IOC10**	P10 I/O control					0	W	

CFP13–CFP10: P1[3:0] pin function selection (D[3:0]) / P1 function select register (0x402D4)

Selects the pin that is used to output a timer underflow signal to external devices.

Write "1": Underflow signal output pin
Write "0": I/O port pin
Read: Invalid

Select the pin used to output a timer underflow signal to external devices from among P10 through P13 by writing "1" to the corresponding bit, CFP10 through CFP13. P10 through P13 correspond to timers 0 through 3, respectively. If "0" is written to CFP1x, the pin is set for an I/O port.
At cold start, CFP1x is set to "0" (I/O port). At hot start, the bit retains its state from prior to the initial reset.

IOC13–IOC10: P1[3:0] port I/O control (D[3:0]) / P1 I/O control register (0x402D6)

Sets input or output mode for P10 through P13.

Write "1": Output mode
Write "0": Input mode
Read: Invalid

If a pin chosen from among P10 through P13 is used to output an underflow signal, write "1" to the corresponding I/O control bit to set it to output mode. If the pin is set to input mode, even if its CFP1x is set to "1", it functions as the event counter input pin of a 16-bit programmable timer cannot be used to output a timer underflow signal.
Since the IOC register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite the register. Use ordinary storage instructions for this purpose.
At cold start, IOC1x is set to "0" (input mode). At hot start, the bit retains its state from prior to the initial reset.

RLD07–RLD00: Timer 0 reload data (D[7:0]) / 8-bit timer 0 reload data register (0x40161)
RLD17–RLD10: Timer 1 reload data (D[7:0]) / 8-bit timer 1 reload data register (0x40165)
RLD27–RLD20: Timer 2 reload data (D[7:0]) / 8-bit timer 2 reload data register (0x40169)
RLD37–RLD30: Timer 3 reload data (D[7:0]) / 8-bit timer 3 reload data register (0x4016D)

Set the initial counter value of each timer.
The reload data set in this register is loaded into each counter, and the counter starts counting down beginning with this data, which is used as the initial count.
There are two cases in which the reload data is loaded into the counter: when data is preset after "1" is written to PSETx, or when data is automatically reloaded upon counter underflow.
At initial reset, RLD is not initialized.

PTD07–PTD00: Timer 0 counter data (D[7:0]) / 8-bit timer 0 counter data (0x40162)
PTD17–PTD10: Timer 1 counter data (D[7:0]) / 8-bit timer 1 counter data (0x40166)
PTD27–PTD20: Timer 2 counter data (D[7:0]) / 8-bit timer 2 counter data (0x4016A)
PTD37–PTD30: Timer 3 counter data (D[7:0]) / 8-bit timer 3 counter data (0x4016E)

The 8-bit programmable timer data can be read out from these bits.
These bits function as buffers that retain the counter data when read out, enabling the data to be read out at any time.
At initial reset, PTD is not initialized.

PSET0: Timer 0 preset (D1) / 8-bit timer 0 control register (0x40160)
PSET1: Timer 1 preset (D1) / 8-bit timer 1 control register (0x40164)
PSET2: Timer 2 preset (D1) / 8-bit timer 2 control register (0x40168)
PSET3: Timer 3 preset (D1) / 8-bit timer 3 control register (0x4016C)

Preset the reload data in the counter.

Write "1":	Preset
Write "0":	Invalid
Read:	Always "0"

The reload data of RLDx is preset in the counter of timer x by writing "1" to PSETx. If the counter is preset when in a RUN state, the counter starts counting immediately after the reload data is preset.
If the counter is preset when in a STOP state, the reload data that has been preset is retained.
Writing "0" results in No Operation.
Since PSETx is a write-only bit, its content when read is always "0".

PTRUN0: Timer 0 RUN/STOP control (D0) / 8-bit timer 0 control register (0x40160)
PTRUN1: Timer 1 RUN/STOP control (D0) / 8-bit timer 1 control register (0x40164)
PTRUN2: Timer 2 RUN/STOP control (D0) / 8-bit timer 2 control register (0x40168)
PTRUN3: Timer 3 RUN/STOP control (D0) / 8-bit timer 3 control register (0x4016C)

Controls the counter's RUN/STOP states.

Write "1":	RUN
Write "0":	STOP
Read:	Valid

The counter of each timer starts counting down when "1" written to PTRUNx, and stops counting when "0" is written.
While in a STOP state, the counter retains its count until it is preset with reload data or placed in a RUN state.
When the state is changed from STOP to RUN, the counter can restart counting beginning with the retained count.
At initial reset, PTRUNx is set to "0" (STOP).

PTOUT0: Timer 0 clock output control register (D2) / 8-bit timer 0 control register (0x40160)
PTOUT1: Timer 1 clock output control register (D2) / 8-bit timer 1 control register (0x40164)
PTOUT2: Timer 2 clock output control register (D2) / 8-bit timer 2 control register (0x40168)
PTOUT3: Timer 3 clock output control register (D2) / 8-bit timer 3 control register (0x4016C)

Controls the clock output of each timer.

Write "1":	On
Write "0":	Off
Read:	Valid

The underflow signal of timer x is output from the external output pin set by CFP1x by writing "1" to PTOUTx.
When using timer 2 or 3 as the clock source of the serial interface, a clock generated from the underflow signal by dividing it by 2 is output to the corresponding channel of the serial interface.
The clock output is turned off by writing "0" to PTOUT, and the external output is fixed at "0" and the internal clock output is fixed at "1".
At initial reset, PTOUT is set to "0" (off).

P8TM2–P8TM0: 8-bit timer interrupt level (D[2:0]) / 8-bit timer, serial I/F Ch.0 interrupt priority register (0x40263)

Set the priority level of the 8-bit programmable timer interrupt in the range of 0 to 7.
At initial reset, the content of the P8TM register becomes indeterminate.

E8TU0: Timer 0 interrupt enable (D0) / 8-bit timer interrupt enable register (0x40275)
E8TU1: Timer 1 interrupt enable (D1) / 8-bit timer interrupt enable register (0x40275)
E8TU2: Timer 2 interrupt enable (D2) / 8-bit timer interrupt enable register (0x40275)
E8TU3: Timer 3 interrupt enable (D3) / 8-bit timer interrupt enable register (0x40275)

Enables or disables generation of an interrupt to the CPU.

Write "1": Interrupt enabled
Write "0": Interrupt disabled
Read: Valid

E8TUx is the interrupt enable bit which controls the interrupt generated by each timer. The interrupt set to "1" by this bit is enabled, and the interrupt set to "0" by this bit is disabled.
At initial reset, E8TUx is set to "0" (interrupt disabled).

F8TU0: Timer 0 interrupt factor flag (D0) / 8-bit timer interrupt factor flag register (0x40285)
F8TU1: Timer 1 interrupt factor flag (D1) / 8-bit timer interrupt factor flag register (0x40285)
F8TU2: Timer 2 interrupt factor flag (D2) / 8-bit timer interrupt factor flag register (0x40285)
F8TU3: Timer 3 interrupt factor flag (D3) / 8-bit timer interrupt factor flag register (0x40285)

Indicates the interrupt generation status of the 8-bit programmable timer.

When read
Read "1": Interrupt factor has occurred
Read "0": No interrupt factor has occurred

When written using the reset-only method (default)
Write "1": Interrupt factor flag is reset
Write "0": Invalid

When written using the read/write method
Write "1": Interrupt flag is set
Write "0": Interrupt flag is reset

F8TUx is the interrupt factor flag corresponding to each timer. It is set to "1" when the counter underflows.
At this time, if the following conditions are met, an interrupt to the CPU is generated:
1. The corresponding interrupt enable register bit is set to "1".
2. No other interrupt request of a higher priority has been generated.
3. The IE bit of the PSR is set to "1" (interrupts enabled).
4. The value set in the corresponding interrupt priority register is higher than the interrupt level (IL) of the CPU.
When using the interrupt factor of the 8-bit programmable timer to request IDMA, note that even when the above conditions are met, no interrupt request to the CPU is generated for the interrupt factor that has occurred. If interrupts are enabled at the setting of IDMA, an interrupt is generated under the above conditions after the data transfer by IDMA is completed.

The interrupt factor flag is set to "1" whenever interrupt generation conditions are met, regardless of how the interrupt enable and interrupt priority registers are set.
If the next interrupt is to be accepted after an interrupt has occurred, it is necessary that the interrupt factor flag be reset, and that the PSR be set again (by setting the IE bit to "1" after setting the IL to a value lower than the level indicated by the interrupt priority register, or by executing the reti instruction).
The interrupt factor flag can be reset only by writing to it in the software. Note that if the PSR is set again to accept interrupts generated (or if the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt occurs again. Note also that the value to be written to reset the flag is "1" when the reset-only method (RSTONLY = "1") is used, and "0" when the read/write method (RSTONLY = "0") is used.
At initial reset, the content of F8TUx becomes indeterminate, so be sure to reset it in the software.

R8TU0: Timer 0 IDMA request (D2) / 16-bit timer 5, 8-bit timer, serial I/F Ch.0 IDMA request register (0x40292)
R8TU1: Timer 1 IDMA request (D3) / 16-bit timer 5, 8-bit timer, serial I/F Ch.0 IDMA request register (0x40292)
R8TU2: Timer 2 IDMA request (D4) / 16-bit timer 5, 8-bit timer, serial I/F Ch.0 IDMA request register (0x40292)
R8TU3: Timer 3 IDMA request (D5) / 16-bit timer 5, 8-bit timer, serial I/F Ch.0 IDMA request register (0x40292)

Specifies whether IDMA is to be invoked at the occurrence of an interrupt factor.

Write "1":	IDMA request
Write "0":	Interrupt request
Read:	Valid

R8TUx is the IDMA request bit for each timer. If this bit is set to "1", IDMA is invoked when an interrupt factor occurs, and thus programmed data transfers are performed. If the bit is set to "0", normal interrupt processing is performed and IDMA is not invoked.
For details on IDMA, refer to Section 9.2, "Intelligent DMA".
At initial reset, the content of R8TUx becomes indeterminate, so be sure to initialize it in the software.

11.3.8 Programming Notes

(1) The 8-bit programmable timer operates only when the prescaler is operating. The prescaler generates a clock for each timer from the OSC3 oscillation clock by dividing it as set using the register. When the CPU is operating using the low-speed (OSC1) clock, the prescaler is inactive, so the 8-bit programmable timer also cannot be used.

(2) The data set in the prescaler is the division ratio for the OSC3 oscillation frequency. If the CPU operating clock is generated from the OSC3 oscillation clock by dividing it using CLKDT[1:0] (D[7:6]) / Power control register (0x40140), do not use a clock that is faster than the CPU operating clock.

(3) When setting an input clock, make sure the 8-bit programmable timer is turned off.

(4) Since the underflow interrupt condition and the timer output status are undefined after an initial reset, the counter initial value should be set to the 8-bit timer before resetting the interrupt factor flag or turning the timer output on.

(5) The function select register (CFP) is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, and bnot) cannot be used to rewrite this register. Use ordinary storage instructions for this purpose.

(6) After an initial reset, the interrupt factor flag (F8TUx) and IDMA request register (R8TUx) become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset this flag and register in the software.

(7) To prevent another interrupt from being generated again by the same factor after an interrupt has occurred, be sure to reset the interrupt factor flag (F8TUx) before setting the PSR again or executing the reti instruction.

11.4 16-Bit Programmable Timers

11.4.1 Configuration of 16-Bit Programmable Timer

The E0C33A104 contains six systems of 16-bit programmable timers (timers 0 to 5). Each timer can be used as a 16-bit × 1-channel or 8-bit × 2-channel programmable timer by switching over the operation modes. They also have an event counter function using an input port pin.

Note: On the following pages, each timer is identified as timer x when used as a 16-bit × 1-channel timer and as timers x0 and x1 when used as 8-bit × 2-channel timers (x = 0 to 5). The functions and control register structures of 16-bit programmable timers 0 to 5 are almost the same. Control bit names are assigned numerals "0" to "5" denoting timer numbers, and "0" or "1" denoting channel numbers. Since explanations are common to all timers, timer numbers are represented by "x" or "xx" unless it is necessary to specify a timer number.

Figure 11.4.1 shows the structure of one channel of the 16-bit programmable timer.

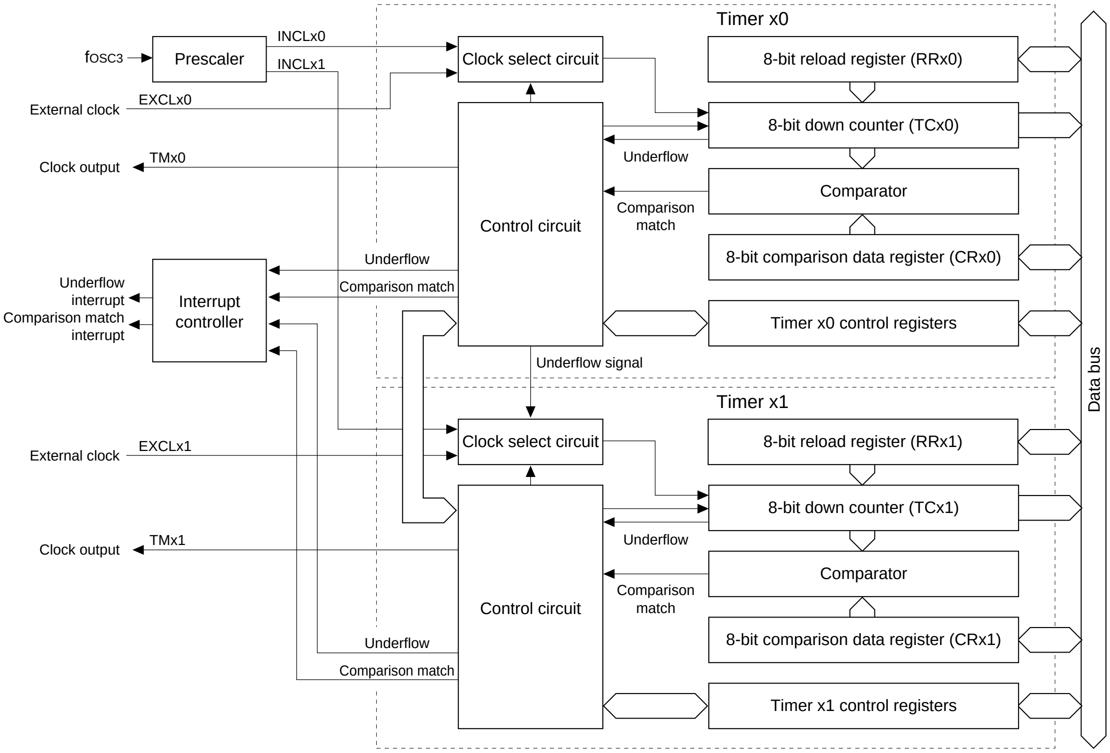


Figure 11.4.1 Structure of 16-Bit Programmable Timer

In each timer, two 8-bit down-counters, as well as 8-bit reload data registers (RRx0, RRx1) and 8-bit comparison data registers (CRx0, CRx1), are provided for each down-counter. The reload data register is used to set the initial value in the down-counter. The comparison data register is used to store the data to be compared with the content of the down-counter. These registers allow interrupt generating intervals and the timer's output clock frequency and duty ratio to be programmed.

11.4.2 I/O Pins of 16-Bit Programmable Timers

Table 11.4.1 shows the input/output pins used for the 16-bit programmable timers.

Table 11.4.1 I/O Pins of 16-Bit Programmable Timer

Pin name	Pin No.		I/O	Pull-up	Function	Function select bit
	QFP5-128	QFP15-128				
R70/TM00	36	33	O	–	Output port / Timer 00 clock output	CFR70(D0) / R7 function select register (0x402CA)
R71/TM01	37	34	O	–	Output port / Timer 01 clock output	CFR71(D1) / R7 function select register (0x402CA)
R72/TM10	38	35	O	–	Output port / Timer 10 clock output	CFR72(D2) / R7 function select register (0x402CA)
R73/TM11	39	36	O	–	Output port / Timer 11 clock output	CFR73(D3) / R7 function select register (0x402CA)
R74/TM21	40	37	O	–	Output port / Timer 21 clock output	CFR74(D4) / R7 function select register (0x402CA)
R75/TM31	41	38	O	–	Output port / Timer 31 clock output	CFR75(D5) / R7 function select register (0x402CA)
R76/TM41	42	39	O	–	Output port / Timer 41 clock output	CFR76(D6) / R7 function select register (0x402CA)
R77/TM51	43	40	O	–	Output port / Timer 51 clock output	CFR77(D7) / R7 function select register (0x402CA)
P10/EXCL00/ T8UF0	46	43	I/O	Built-in	I/O port / Timer 00 event counter input / 8-bit timer 0 output	CFP10(D0) / P1 function select register (0x402D4)
P11/EXCL01/ T8UF1	47	44	I/O	Built-in	I/O port / Timer 01 event counter input / 8-bit timer 1 output	CFP11(D1) / P1 function select register (0x402D4)
P12/EXCL10/ T8UF2	48	45	I/O	Built-in	I/O port / Timer 10 event counter input / 8-bit timer 2 output	CFP12(D2) / P1 function select register (0x402D4)
P13/EXCL20/ T8UF3	49	46	I/O	Built-in	I/O port / Timer 20 event counter input / 8-bit timer 3 output	CFP13(D3) / P1 function select register (0x402D4)
P14/EXCL30/ #BUSGET/ #DWE	50	47	I/O	Built-in	I/O port / Timer 30 event counter input / #BUSGET signal output / #DWE signal output	CFP14(D4) / P1 function select register (0x402D4)
P15/EXCL40/ #DMAEND0	51	48	I/O	Built-in	I/O port / Timer 40 event counter input / High-speed DMA Ch.0 end signal output	CFP15(D5) / P1 function select register (0x402D4)
P16/EXCL50/ #DMAEND1	52	49	I/O	Built-in	I/O port / Timer 50 event counter input / High-speed DMA Ch.1 end signal output	CFP16(D6) / P1 function select register (0x402D4)

TMxx (output pin of the 16-bit programmable timer)

This pin outputs a clock generated by the timer (00, 01, 10, 11, 21, 31, 41, or 51).

EXCLxx (event counter input pin)

When using the timer (00, 01, 10, 20, 30, 40, or 50) as an event counter, input count pulses from an external source to this pin.

How to set the input/output pins of 16-bit programmable timers

All clock output pins used by the 16-bit programmable timers are shared with output ports. At cold start, all these pins are set for output port pins R7x (function select bit CFR7x = "0"), and output a low-level signal. When using the clock output function of the 16-bit programmable timer, select the desired timer and write "1" to the function select bit CFR7x for the corresponding pin. At hot start, these pins retain their status before from prior to the reset.

All event-counter input pins are shared with I/O-port pins. At cold start, all these pins are set for I/O-port pins P1x (function select bit CFP1x = "0"). When using the event counter function, select the desired timer and write "1" to the function select bit CFP1x for the corresponding pin.
Note that these pins are also shared with output pins for the 8-bit programmer timers, etc. When the input/output pins are set in input mode, they function as event counter inputs. Therefore, it is necessary to set the I/O port's I/O control IOC1x (D[6:0]/0x402D6) to "0" in advance. At cold start, these pins are set in input mode. At hot start, they retain their status from prior to the reset.

Since the function select and I/O control registers are write-only registers (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) that accompany a read-modify-write operation cannot be used to rewrite these registers. Use ordinary storage instructions. At hot start, they retain their status from prior to the reset.

Pull-up resistors of input pins

Since the event counter input pins are shared with I/O ports, they contain a pull-up resistor. Whether these pull-up resistors are to be used can be specified for each pin through the use of the pull-up control register.
Pull-up resistor control for P16 to P10: IOU1[6:0] (D[6:0]) / P1 pull-up control register (0x402D7)

At cold start, the pull-up control register is set to "0", with the pull-up resistor disconnected from the input line. If you want an input pin to be pulled up, write "1" to IOU1x for the corresponding pin. Since the pull-up control register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite the register. Use ordinary storage instructions.
At hot start, the register retains its status from prior to the reset.

11.4.3 Uses of 16-Bit Programmable Timers

The down-counters of the 16-bit programmable timer cyclically output an underflow signal and a comparison-match signal in accordance with the preset data and comparison data that are set in the software. These signals are used to generate an interrupt request to the CPU or control the internal peripheral circuits. A clock generated from these signals can also be output to external devices.

CPU interrupt request/IDMA invocation request

Each timer's underflow and comparison match (matching of counter and comparison data) can be used as an interrupt factor to generate an interrupt request to the CPU. Therefore, an interrupt can be generated at an interval that is set in the software.
Furthermore, this interrupt factor can also be used to invoke IDMA (timer x1 only when using the timer as 8-bit × 2-channel timers).

Clock output to external devices

A clock generated from the underflow and comparison-match signals can be output from the chip to the outside. The clock cycle is determined by the underflow signal, and the duty ratio is determined by the comparison match signal. This output can be used to control external devices. The output pins of each timer are described in the preceding section.

A/D converter start trigger

The A/D converter allows a trigger to start the A/D conversion to be selected from among four available types. One is the underflow signal from the 16-bit programmable timer 0. This makes it possible to perform the A/D conversion at programmable intervals.
To use this function, write "01" to the A/D converter control TS[1:0] (D[4:3]) / A/D trigger register (0x40242) to select the 16-bit programmable timer 0 as the trigger.

Watchdog timer

The 16-bit programmable timer 0 can be used as a watchdog timer to monitor CPU crash. In this case, the underflow signal of this timer serves as an NMI request signal to the CPU.
To use this function, write "1" to the watchdog timer control bit EWD (D1) / Watchdog timer enable register (0x40171) to enable the NMI. For details on how to control the watchdog timer, refer to Section 11.2, "Watchdog Timer".

11.4.4 Operation Modes of 16-Bit Programmable Timers

Each timer can be used as 2-channel 8-bit timers or a 1-channel 16-bit timer. Two operation modes corresponding to each timer are provided, and the desired mode can be selected using the 8/16-bit mode select bit MODE16.

MODE16 = "0": 8-bit mode (8 bits × 2 channels)
MODE16 = "1": 16-bit mode (16 bits × 1 channel)

Figure 11.4.2 shows the timer configuration in each operation mode.

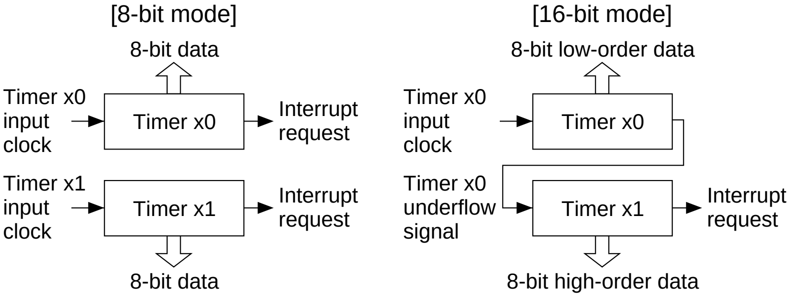


Figure 11.4.2 Timer Configuration in 8- and 16-Bit Modes

In the 8-bit mode, timers x0 and x1 can be controlled independently of each other.

In the 16-bit mode, the underflow signal of timer x0 is fed to timer x1 as its input clock, enabling them to operate as consecutive 16-bit down-counters.
Timers in the 16-bit mode are controlled using the timer x0 control register, except for clock output.

11.4.5 Control and Operation of 16-Bit Programmable Timer

The following settings must first be made before the 16-bit programmable timer starts counting:
1. Setting pins for input/output (only when necessary)
2. Setting operation mode
3. Setting input clock
4. Setting preset data (initial counter value)
5. Setting comparison data
6. Setting interrupt/IDMA

For details on how to set interrupts and IDMA, refer to Section 11.4.7, "16-Bit Programmable Timer Interrupts and DMA".

Setting pin for input/output

The pin must be set for output for the output clock of the 16-bit programmable timer to be fed to external devices.
Clocks can be output from the following timers:

16-bit mode:	Timer 0 (01), timer 1 (11), timer 2 (21), timer 3 (31), timer 4 (41), timer 5 (51)
8-bit mode:	Timer 00, timer 01, timer 10, timer 11, timer 21, timer 31, timer 41, timer 51

The pin for input must be set for the 16-bit programmable timer to be used as an event counter that counts external clock pulses.
External clocks can be input to the following timers:

16-bit mode:	Timer 0 (00), timer 1 (10), timer 2 (20), timer 3 (30), timer 4 (40), timer 5 (50)
8-bit mode:	Timer 00, timer 01, timer 10, timer 20, timer 30, timer 40, timer 50

For details on how to set the pin, refer to Section 11.4.2, "I/O Pins of 16-Bit Programmable Timers".

Setting the operation mode

Timers 0 through 5 each can be used as a 16-bit × 1-channel timer or as 8-bit × 2-channel timers, as described earlier. Use each timer's 8/16-bit mode select bit MODE16 to set the operation mode. The 16-bit mode is selected by writing "1" to MODE16 and the 8-bit mode is selected by writing "0".

Timer 0 mode selection: MODE16 (D4) / 16-bit timer 00 control register (0x40180)
Timer 1 mode selection: MODE16 (D4) / 16-bit timer 10 control register (0x40188)
Timer 2 mode selection: MODE16 (D4) / 16-bit timer 20 control register (0x40190)
Timer 3 mode selection: MODE16 (D4) / 16-bit timer 30 control register (0x40198)
Timer 4 mode selection: MODE16 (D4) / 16-bit timer 40 control register (0x401A0)
Timer 5 mode selection: MODE16 (D4) / 16-bit timer 50 control register (0x401A8)

Setting the input clock

The count clock for each timer can be selected from between an internal clock and an external clock. Use the following control bits to select the input clock:

Timer 00 input clock selection: CKSL00 (D3) / 16-bit timer 00 input clock select register (0x40180)
Timer 01 input clock selection: CKSL01 (D3) / 16-bit timer 00 input clock select register (0x40181)
Timer 10 input clock selection: CKSL10 (D3) / 16-bit timer 00 input clock select register (0X40188)
Timer 11 input clock selection: CKSL11 (D3) / 16-bit timer 00 input clock select register (0x40189)*
Timer 20 input clock selection: CKSL20 (D3) / 16-bit timer 00 input clock select register (0x40190)
Timer 21 input clock selection: CKSL21 (D2) / 16-bit timer 00 input clock select register (0x40191)*
Timer 30 input clock selection: CKSL30 (D3) / 16-bit timer 00 input clock select register (0x40198)
Timer 31 input clock selection: CKSL31 (D3) / 16-bit timer 00 input clock select register (0x40199)*
Timer 40 input clock selection: CKSL40 (D3) / 16-bit timer 00 input clock select register (0x401A0)
Timer 41 input clock selection: CKSL41 (D3) / 16-bit timer 00 input clock select register (0x401A1)*
Timer 50 input clock selection: CKSL50 (D3) / 16-bit timer 00 input clock select register (0x401A8)
Timer 51 input clock selection: CKSL51 (D3) / 16-bit timer 00 input clock select register (0x401A9)*

An external clock is selected by writing "1" to CKSLxx, and the internal clock is selected by writing "0".
In the 16-bit mode, select the input clock for timer x0. Input clock selection for timer x1 is invalid.
At initial reset, CKSLxx is set for the internal clock.
An External clock can be used for the timer for which the pin is set for input. Timers x1, except for timer 01, do not have the external clock input function. Therefore, the timers marked with an * cannot be operated when the bit is set to "1" (external clock).

- **Internal clock**

When the internal clock is selected as a timer, the timer is operated by the prescaler output clock. The prescaler division ratio can be selected for each timer.

Table 11.4.2 Setting the Internal Clock

Timer	Control register	Division ratio select bit	Clock control bit
Timer 00	16-bit timer 0x clock control register (0x40147)	P16TS00[2:0] (D2:0])	P16TON00 (D3)
Timer 01		P16TS01[2:0] (D6:4])	P16TON01 (D7)
Timer 10	16-bit timer 1x clock control register (0x40148)	P16TS10[2:0] (D2:0])	P16TON10 (D3)
Timer 11		P16TS11[2:0] (D6:4])	P16TON11 (D7)
Timer 20	16-bit timer 2x clock control register (0x40149)	P16TS20[2:0] (D2:0])	P16TON20 (D3)
Timer 21		P16TS21[2:0] (D6:4])	P16TON21 (D7)
Timer 30	16-bit timer 3x clock control register (0x4014A)	P16TS30[2:0] (D2:0])	P16TON30 (D3)
Timer 31		P16TS31[2:0] (D6:4])	P16TON31 (D7)
Timer 40	16-bit timer 4x clock control register (0x4014B)	P16TS40[2:0] (D2:0])	P16TON40 (D3)
Timer 41		P16TS41[2:0] (D6:4])	P16TON41 (D7)
Timer 50	16-bit timer 5x clock control register (0x4014C)	P16TS50[2:0] (D2:0])	P16TON50 (D3)
Timer 51		P16TS51[2:0] (D6:4])	P16TON51 (D7)

Note that the division ratio differs for each timer.

Table 11.4.3 Input Clock Selection

Timer	P16TS = 7	P16TS = 6	P16TS = 5	P16TS = 4	P16TS = 3	P16TS = 2	P16TS = 1	P16TS = 0
0x, 2x, 4x	f_{OSC3}/4096	f_{OSC3}/1024	f_{OSC3}/256	f_{OSC3}/128	f_{OSC3}/64	f_{OSC3}/32	f_{OSC3}/16	f_{OSC3}/4
1x, 3x, 5x	f_{OSC3}/2048	f_{OSC3}/512	f_{OSC3}/256	f_{OSC3}/128	f_{OSC3}/64	f_{OSC3}/32	f_{OSC3}/8	f_{OSC3}/2

f_{OSC3}: OSC3 oscillation frequency

The selected clock is output from the prescaler to the 16-bit programmable timer by writing "1" to P16TONxx.

Notes: • When the internal clock is used, the 16-bit programmable timer operates only when the prescaler is operating. The prescaler generates a clock for each timer from the OSC3 oscillation clock by dividing it as set using the clock control register. When the CPU is operating using the low-speed (OSC1) clock, the prescaler is inactive, so the 16-bit programmable timer cannot be used (refer to Section 10.2, "Prescaler and Operating Clock for Peripheral Circuits").

• The data set in the prescaler is the division ratio for the OSC3 oscillation frequency. If the CPU operating clock is generated from the OSC3 oscillation clock by dividing it using CLKDT[1:0] (D[7:6]) / Power control register (0x40140), do not use a clock that is faster than the CPU operating clock.

• When setting an input clock, make sure the 16-bit programmable timer is turned off.

- **External clock**

When using the timer as an event counter by supplying clock pulses from an external source, make sure the event cycle is at least twice that of the CPU operating clock period.

Setting preset data (initial counter value)

Set the initial values of the down-counters in two 8-bit reload data registers in each timer before using the timer.

Timer 00 reload data: RR00[7:0] (D[7:0]) / 16-bit timer 00 reload data register (0x40182)
Timer 01 reload data: RR01[7:0] (D[7:0]) / 16-bit timer 01 reload data register (0x40183)
Timer 10 reload data: RR10[7:0] (D[7:0]) / 16-bit timer 10 reload data register (0x4018A)
Timer 11 reload data: RR11[7:0] (D[7:0]) / 16-bit timer 11 reload data register (0x4018B)
Timer 20 reload data: RR20[7:0] (D[7:0]) / 16-bit timer 20 reload data register (0x40192)
Timer 21 reload data: RR21[7:0] (D[7:0]) / 16-bit timer 21 reload data register (0x40193)
Timer 30 reload data: RR30[7:0] (D[7:0]) / 16-bit timer 30 reload data register (0x4019A)
Timer 31 reload data: RR31[7:0] (D[7:0]) / 16-bit timer 31 reload data register (0x4019B)
Timer 40 reload data: RR40[7:0] (D[7:0]) / 16-bit timer 40 reload data register (0x401A2)
Timer 41 reload data: RR41[7:0] (D[7:0]) / 16-bit timer 41 reload data register (0x401A3)
Timer 50 reload data: RR50[7:0] (D[7:0]) / 16-bit timer 50 reload data register (0x401AA)
Timer 51 reload data: RR51[7:0] (D[7:0]) / 16-bit timer 51 reload data register (0x401AB)

In the 8-bit mode, the two registers are used as independent 8-bit registers.
In the 16-bit mode, the register for timer x0 is used to store the 8 low-order bits of reload data, and the register for timer x1 is used to store the 8 high-order bits of reload data.
The reload data registers can be read and written. At initial reset, the reload data registers are not initialized.

The data written to this register is preset in the down-counter, and the counter starts counting down from the preset value.

Data is thus preset in the down-counter in the following two cases:

1. **When preset in the software**
   Presetting in the software is performed using the preset control bit PSETxx. When this bit is set to "1", the content of the reload data register is loaded into the down-counter.
   Timer 00 preset: PSET00 (D1) / 16-bit timer 00 control register (0x40180)
   Timer 01 preset: PSET01 (D1) / 16-bit timer 01 control register (0x40181)
   Timer 10 preset: PSET10 (D1) / 16-bit timer 10 control register (0x40188)
   Timer 11 preset: PSET11 (D1) / 16-bit timer 11 control register (0x40189)
   Timer 20 preset: PSET20 (D1) / 16-bit timer 20 control register (0x40190)
   Timer 21 preset: PSET21 (D1) / 16-bit timer 21 control register (0x40191)
   Timer 30 preset: PSET30 (D1) / 16-bit timer 30 control register (0x40198)
   Timer 31 preset: PSET31 (D1) / 16-bit timer 31 control register (0x40199)
   Timer 40 preset: PSET40 (D1) / 16-bit timer 40 control register (0x401A0)
   Timer 41 preset: PSET41 (D1) / 16-bit timer 41 control register (0x401A1)
   Timer 50 preset: PSET50 (D1) / 16-bit timer 50 control register (0x401A8)
   Timer 51 preset: PSET51 (D1) / 16-bit timer 51 control register (0x401A9)

   In the 16-bit mode, all 16 bits of reload data are loaded in one operation by the PSETx0 bit. In this case, writing to PSETx1 is invalidated.

2. **When the down-counter underflows during counting**
   Since the reload data is preset in the down-counter upon underflow, its underflow cycle is determined by the value that is set in the reload data register. This underflow signal controls the clock output (TMxx signal) to external devices, in addition to generating an interrupt.

   In the 16-bit mode, all 16 bits of reload data are loaded in one operation by an underflow of timer x1.

Before starting the 16-bit programmable timer, set the initial value in the reload data register and use the PSETxx bit to preset the data in the down-counter.
The underflow cycle is determined by the input clock frequency and the reload data. The relationship between these two parameters is expressed by the following equation:

$$\text{Underflow cycle} = \frac{RR + 1}{f_{CLK}} \quad \text{[sec.]}$$

f_{CLK}: Input clock frequency [Hz] (f_{OSC3} × dr for the internal clock)
RR: Set value of the reload data register
f_{OSC3}: High-speed (OSC3) oscillation frequency [Hz]
dr: Division ratio of the prescaler set using P16TSxx

Setting comparison data

The programmable timer contains a data comparator that allows the count data to be compared with a given value. The comparison data register is used to set this value.
Timer 00 comparison data: CR00[7:0] (D[7:0]) / 16-bit timer 00 comarison data register (0x40184)
Timer 01 comparison data: CR01[7:0] (D[7:0]) / 16-bit timer 01 comarison data register (0x40185)
Timer 10 comparison data: CR10[7:0] (D[7:0]) / 16-bit timer 10 comarison data register (0x4018C)
Timer 11 comparison data: CR11[7:0] (D[7:0]) / 16-bit timer 11 comarison data register (0x4018D)
Timer 20 comparison data: CR20[7:0] (D[7:0]) / 16-bit timer 20 comarison data register (0x40194)
Timer 21 comparison data: CR21[7:0] (D[7:0]) / 16-bit timer 21 comarison data register (0x40195)
Timer 30 comparison data: CR30[7:0] (D[7:0]) / 16-bit timer 30 comarison data register (0x4019C)
Timer 31 comparison data: CR31[7:0] (D[7:0]) / 16-bit timer 31 comarison data register (0x4019D)
Timer 40 comparison data: CR40[7:0] (D[7:0]) / 16-bit timer 40 comarison data register (0x401A4)
Timer 41 comparison data: CR41[7:0] (D[7:0]) / 16-bit timer 41 comarison data register (0x401A5)
Timer 50 comparison data: CR50[7:0] (D[7:0]) / 16-bit timer 50 comarison data register (0x401AC)
Timer 51 comparison data: CR51[7:0] (D[7:0]) / 16-bit timer 51 comarison data register (0x401AD)

In the 8-bit mode, these registers are used as independent 8-bit registers.
In the 16-bit mode, the register for timer x0 is used to store the 8 low-order bits of comparison data, and the register for timer x1 is used to store the 8 high-order bits of comparison data.
The comparison data registers can be read and written. At initial reset, the comparison data registers are not initialized.

The programmable timer compares the comparison data register and count data and, when the two values are equal, generates a comparison match signal. This comparison match signal controls the clock output (TMxx signal) to external devices, in addition to generating an interrupt. However, when the comparison data register is set to "0", the comparisom match signal is not generated even if the timer data becomes "0".

Timer RUN/STOP control

Each timer includes the PRUNxx bit to control RUN/STOP.
Timer 00 RUN/STOP control: PRUN00 (D0) / 16-bit timer 00 control register (0x40180)
Timer 01 RUN/STOP control: PRUN01 (D0) / 16-bit timer 01 control register (0x40181)
Timer 10 RUN/STOP control: PRUN10 (D0) / 16-bit timer 10 control register (0x40188)
Timer 11 RUN/STOP control: PRUN11 (D0) / 16-bit timer 11 control register (0x40189)
Timer 20 RUN/STOP control: PRUN20 (D0) / 16-bit timer 20 control register (0x40190)
Timer 21 RUN/STOP control: PRUN21 (D0) / 16-bit timer 21 control register (0x40191)
Timer 30 RUN/STOP control: PRUN30 (D0) / 16-bit timer 30 control register (0x40198)
Timer 31 RUN/STOP control: PRUN11 (D0) / 16-bit timer 31 control register (0x40199)
Timer 40 RUN/STOP control: PRUN40 (D0) / 16-bit timer 40 control register (0x401A0)
Timer 41 RUN/STOP control: PRUN01 (D0) / 16-bit timer 41 control register (0x401A1)
Timer 50 RUN/STOP control: PRUN50 (D0) / 16-bit timer 50 control register (0x401A8)
Timer 51 RUN/STOP control: PRUN11 (D0) / 16-bit timer 51 control register (0x401A9)

The timer starts counting when "1" is written to PRUNxx. The clock input is disabled and the timer stops counting when "0" is written to PRUNx.
In the 8-bit mode, the 8-bit counters can be controlled independently of each other.
In the 16-bit mode, all 16 bits are controlled collectively by PRUNx0. In this case, control of PRUNx1 is invalid.

This RUN/STOP control does not affect the counter data. Even when the timer has stopped counting, the counter retains its count so that the timer can start counting again from that point.

If the count of the counter matches the set value of the comparison data register during count-down, the timer generates a comparison match interrupt.
When the terminal count is reached and the counter underflows, an underflow interrupt is generated while the initial count that is set in the reload data register is reloaded in the counter.
The counter continues counting down regardless of which interrupt has occurred. In the case of an underflow interrupt, the counter starts counting beginning with the reloaded initial count.

When both the timer RUN/STOP control bit (PRUNxx) and the timer preset bit (PSETxx) are set to "1" at the same time, the timer starts counting after presetting the reload register value into the counter.

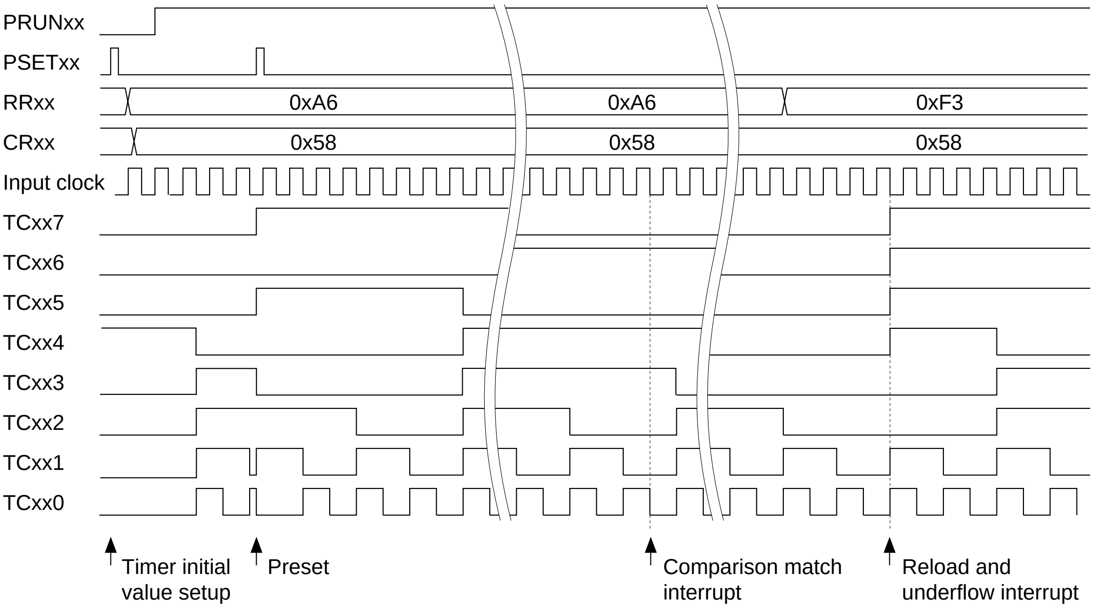


Figure 11.4.3 Basic Operation Timing of Counter (example in 8-bit mode)

Reading counter data

The counter data can be read out from the following addresses shown below at any time:

Timer 00 data: TC00[7:0] (D[7:0]) / 16-bit timer 00 counter data register (0x40186)
Timer 01 data: TC01[7:0] (D[7:0]) / 16-bit timer 01 counter data register (0x40187)
Timer 10 data: TC10[7:0] (D[7:0]) / 16-bit timer 10 counter data register (0x4018E)
Timer 11 data: TC11[7:0] (D[7:0]) / 16-bit timer 11 counter data register (0x4018F)
Timer 20 data: TC20[7:0] (D[7:0]) / 16-bit timer 20 counter data register (0x40196)
Timer 21 data: TC21[7:0] (D[7:0]) / 16-bit timer 21 counter data register (0x40197)
Timer 30 data: TC30[7:0] (D[7:0]) / 16-bit timer 30 counter data register (0x4019E)
Timer 31 data: TC31[7:0] (D[7:0]) / 16-bit timer 31 counter data register (0x4019F)
Timer 40 data: TC40[7:0] (D[7:0]) / 16-bit timer 40 counter data register (0x401A6)
Timer 41 data: TC41[7:0] (D[7:0]) / 16-bit timer 41 counter data register (0x401A7)
Timer 50 data: TC50[7:0] (D[7:0]) / 16-bit timer 50 counter data register (0x401AE)
Timer 51 data: TC51[7:0] (D[7:0]) / 16-bit timer 51 counter data register (0x401AF)

In the 16-bit mode, the 8 low-order bits of counter data are stored at the address for timer x0, and the 8 high-order bits of counter data are stored at the address for timer x1.

In the 16-bit mode, data should be read in half word units. If the counter underflows while reading 8 separate bits once, the data thus obtained may not be correct.

11.4.6 Controlling Clock Output

The timers shown below can generate a TMxx signal using the underflow and comparison match signals from the counter.

16-bit mode: Timer 0 (01), timer 1 (11), timer 2 (21), timer 3 (31), timer 4 (41), timer 5 (51)
8-bit mode: Timer 00, timer 01, timer 10, timer 11, timer 21, timer 31, timer 41, timer 51

The TMxx signal generated here can be output from the clock output pins (see Table 11.4.1), enabling a programmable clock to be supplied to external devices.

To output the TMxx clock, write "1" to the clock output control bit PTMxx. Clock output is fixed at "0" by writing "0" to PTMxx.

Timer 00 clock output control: PTM00 (D2) / 16-bit timer 00 control register (0x40180)
Timer 01 clock output control: PTM01 (D2) / 16-bit timer 01 control register (0x40181)
Timer 10 clock output control: PTM10 (D2) / 16-bit timer 10 control register (0x40188)
Timer 11 clock output control: PTM11 (D2) / 16-bit timer 11 control register (0x40189)
Timer 21 clock output control: PTM21 (D2) / 16-bit timer 21 control register (0x40191)
Timer 31 clock output control: PTM31 (D2) / 16-bit timer 31 control register (0x40199)
Timer 41 clock output control: PTM41 (D2) / 16-bit timer 41 control register (0x401A1)
Timer 51 clock output control: PTM51 (D2) / 16-bit timer 51 control register (0x401A9)

Although timers x0 other than the above include a PTMx0 bit, no clock can be output even if "1" is written to PTMx0 because the timer has no clock output function.

Figure 11.4.4 shows the waveform of the output signal.

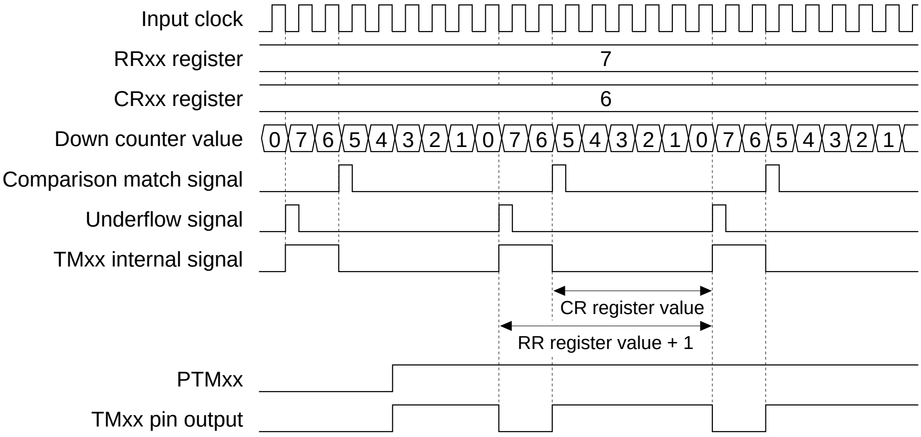


Figure 11.4.4 Waveform of 16-Bit Programmable Timer Output

The TMxx signal falls synchronously with a rising edge of the underflow signal, and rises synchronously with a rising edge of the comparison match signal. Therefore, the TMxx signal's frequency and duty ratio can be changed by setting the reload data register (RRxx) and comparison data register (CRxx) as necessary. However, the conditions RRxx ≥ CRxx and CDR ≠ 0 must be met. If RRxx < CRxx or CRxx = 0, the TMxx signal is fixed at "0".

Note: At cold start, the TMxx output pin outputs a low level.
If the operation mode (8-bit/16-bit) is changed after the pin for clock output is set, an indeterminate value is output.

16-bit timer external output (R70–R77 ports)

1) After an initial reset (cold start), the ports (R70–R77) are set to general-purpose output ports and output "0".
2) The port outputs "0" when it is set to the 16-bit timer output (timer output is off status).
3) The timer output is left as "0" when the timer output is turned on after setting the operating mode, input clock, timer initial value and comparison data.
4) After starting the timer, the output signal rises by a comparison match and falls by an underflow.

11.4.7 16-Bit Programmable Timer Interrupts and DMA

The 16-bit programmable timer has a function for generating an interrupt using the underflow and comparison match states of each timer.

The timing at which an interrupt is generated is shown in Figure 11.4.3 in the preceding section.

Control registers of the interrupt controller

Table 11.4.4 shows the control registers of the interrupt controller provided for each timer.

Table 11.4.4 Control Registers of Interrupt Controller

Interrupt factor	Interrupt factor flag	Interrupt enable register	Interrupt priority register	IDMA request register
Timer 00 underflow	F16TU00 (D0/0x40282)	E16TU00 (D0/0x40272)	P16TM0[2:0] (D[6:4]/0x40261)	–
Timer 00 comparison match	F16TC00 (D1/0x40282)	E16TC00 (D1/0x40272)		–
Timer 01 underflow	F16TU01 (D2/0x40282)	E16TU01 (D2/0x40272)		R16TU01 (D6/0x40290)
Timer 01 comparison match	F16TC01 (D3/0x40282)	E16TC01 (D3/0x40272)		R16TC01 (D7/0x40290)
Timer 10 underflow	F16TU10 (D4/0x40282)	E16TU10 (D4/0x40272)		–
Timer 10 comparison match	F16TC10 (D5/0x40282)	E16TC10 (D5/0x40272)		–
Timer 11 underflow	F16TU11 (D6/0x40282)	E16TU11 (D6/0x40272)		R16TU11 (D0/0x40291)
Timer 11 comparison match	F16TC11 (D7/0x40282)	E16TC11 (D7/0x40272)		R16TC11 (D1/0x40291)
Timer 20 underflow	F16TU20 (D0/0x40283)	E16TU20 (D0/0x40273)	P16TM2[2:0] (D[2:0]/0x40262)	–
Timer 20 comparison match	F16TC20 (D1/0x40283)	E16TC20 (D1/0x40273)		–
Timer 21 underflow	F16TU21 (D2/0x40283)	E16TU21 (D2/0x40273)		R16TU21 (D2/0x40291)
Timer 21 comparison match	F16TC21 (D3/0x40283)	E16TC21 (D3/0x40273)		R16TC21 (D3/0x40291)
Timer 30 underflow	F16TU30 (D4/0x40283)	E16TU30 (D4/0x40273)		–
Timer 30 comparison match	F16TC30 (D5/0x40283)	E16TC30 (D5/0x40273)		–
Timer 31 underflow	F16TU31 (D6/0x40283)	E16TU31 (D6/0x40273)		R16TU31 (D4/0x40291)
Timer 31 comparison match	F16TC31 (D7/0x40283)	E16TC31 (D7/0x40273)		R16TC31 (D5/0x40291)
Timer 40 underflow	F16TU40 (D0/0x40284)	E16TU40 (D0/0x40274)	P16TM4[2:0] (D[6:4]/0x40262)	–
Timer 40 comparison match	F16TC40 (D1/0x40284)	E16TC40 (D1/0x40274)		–
Timer 41 underflow	F16TU41 (D2/0x40284)	E16TU41 (D2/0x40274)		R16TU41 (D6/0x40291)
Timer 41 comparison match	F16TC41 (D3/0x40284)	E16TC41 (D3/0x40274)		R16TC41 (D7/0x40291)
Timer 50 underflow	F16TU50 (D4/0x40284)	E16TU50 (D4/0x40274)		–
Timer 50 comparison match	F16TC50 (D5/0x40284)	E16TC50 (D5/0x40274)		–
Timer 51 underflow	F16TU51 (D6/0x40284)	E16TU51 (D6/0x40274)		R16TU51 (D0/0x40292)
Timer 51 comparison match	F16TC51 (D7/0x40284)	E16TC51 (D7/0x40274)		R16TC51 (D1/0x40292)

When an underflow or comparison match state occurs in the timer, the corresponding interrupt factor flag is set to "1".

In the 8-bit mode, the comparison match interrupt factor and underflow interrupt factor flags are each set independently for timers x0 and x1.

In the 16-bit mode, the interrupt factor flag for timer x1 is set by a 16-bit comparison match or underflow.

If the interrupt enable register bit corresponding to that interrupt factor flag has been set to "1", an interrupt request is generated.

An interrupt caused by a timer can be disabled by leaving the interrupt enable register bit for that timer set to "0". The interrupt factor flag is always set to "1" by the timer's underflow or comparison match state, regardless of how the interrupt enable register is set (even when set to "0").

The interrupt priority register sets an interrupt priority level (0 to 7) for two timer blocks as one interrupt source. Priorities within a timer block are such that timers of smaller numbers have a higher priority. Priorities between interrupt types are such that the underflow interrupt has priority over the comparison match interrupt.

An interrupt request to the CPU is accepted only when no other interrupt request of a higher priority has been generated.

It is only when the PSR's IE bit = "1" (interrupts enabled) and the set value of the IL is smaller than the timer interrupt level set by the interrupt priority register, that a timer interrupt request is actually accepted by the CPU.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, refer to Chapter 8, "Interrupt".

Intelligent DMA

The interrupt factor of timer x1 can also invoke intelligent DMA (IDMA). This allows memory-to-memory DMA transfers to be performed cyclically.

The following shows the IDMA channel numbers set for each interrupt factor of timer:

	IDMA Ch.
Timer 01 underflow:	0x07
Timer 01 comparison match:	0x08
Timer 11 underflow:	0x09
Timer 11 comparison match:	0x0A
Timer 21 underflow:	0x0B
Timer 21 comparison match:	0x0C
Timer 31 underflow:	0x0D
Timer 31 comparison match:	0x0E
Timer 41 underflow:	0x0F
Timer 41 comparison match:	0x10
Timer 51 underflow:	0x11
Timer 51 comparison match:	0x12

For IDMA to be invoked, one of the IDMA request bits shown in Table 11.4.4 must be set to "1" in advance. Transfer conditions, etc. must also be set on the IDMA side in advance.

If the IDMA request bit is set to "1", IDMA is invoked by the generation of an interrupt factor. No interrupt request is generated at that point in time. An interrupt request is generated upon completion of the DMA transfer. The registers can also be set not to generate an interrupt, but only to perform a DMA transfer.

For details on DMA transfers and interrupt control upon completion of DMA transfer, refer to Section 9.2, "Intelligent DMA".

Trap vectors

The trap vector addresses for each default interrupt factor are set as shown below:

	BTA3 = low
Timer 00 underflow interrupt:	0x0C00070
Timer 00 comparison match interrupt:	0x0C00074
Timer 01 underflow interrupt:	0x0C00078
Timer 01 comparison match interrupt:	0x0C0007C
Timer 10 underflow interrupt:	0x0C00080
Timer 10 comparison match interrupt:	0x0C00084
Timer 11 underflow interrupt:	0x0C00088
Timer 11 comparison match interrupt:	0x0C0008C
Timer 20 underflow interrupt:	0x0C00090
Timer 20 comparison match interrupt:	0x0C00094
Timer 21 underflow interrupt:	0x0C00098
Timer 21 comparison match interrupt:	0x0C0009C
Timer 30 underflow interrupt:	0x0C000A0
Timer 30 comparison match interrupt:	0x0C000A4
Timer 31 underflow interrupt:	0x0C000A8
Timer 31 comparison match interrupt:	0x0C000AC
Timer 40 underflow interrupt:	0x0C000B0
Timer 40 comparison match interrupt:	0x0C000B4
Timer 41 underflow interrupt:	0x0C000B8
Timer 41 comparison match interrupt:	0x0C000BC
Timer 50 underflow interrupt:	0x0C000C0
Timer 50 comparison match interrupt:	0x0C000C4
Timer 51 underflow interrupt:	0x0C000C8
Timer 51 comparison match interrupt:	0x0C000CC

The base address of the trap table can be changed using the TTBR register (0x48134 to 0x48137).

11.4.8 I/O Memory of 16-Bit Programmable Timers

Table 11.4.5 shows the control bits of the 16-bit programmable timers.
For details on the I/O memory of the prescaler used to set a clock, refer to Section 10.2,4, " I/O Memory of Prescaler".

Table 11.4.5 Control Bits of 16-Bit Programmable Timer

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 00 control register	**0040180 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**MODE16**	16-bit timer 0 mode selection	1	16 bits	0	8 bits	0	R/W	
		D3	**CKSL00**	16-bit timer 00 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	**PTM00**	16-bit timer 00 clock output control	1	On	0	Off	0	R/W	*1
		D1	**PSET00**	16-bit timer 00 preset	1	Preset	0	Invalid	0	W	0 when being read.
		D0	**PRUN00**	16-bit timer 00 Run/Stop control	1	Run	0	Stop	0	R/W	
16-bit timer 01 control register	**0040181 (B)**	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**CKSL01**	16-bit timer 01 input clock selection	1	External clock	0	Internal clock	0	R/W	*1
		D2	**PTM01**	16-bit timer 01 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET01**	16-bit timer 01 preset	1	Preset	0	Invalid	0	W	0 when being read.*1
		D0	**PRUN01**	16-bit timer 01 Run/Stop control	1	Run	0	Stop	0	R/W	*1
16-bit timer 00 reload data register	**0040182 (B)**	D7	**RR007**	16-bit timer 00 reload data RR007 = MSB RR000 = LSB	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR006**						X		
		D5	**RR005**						X		
		D4	**RR004**						X		
		D3	**RR003**						X		
		D2	**RR002**						X		
		D1	**RR001**						X		
		D0	**RR000**						X		
16-bit timer 01 reload data register	**0040183 (B)**	D7	**RR017**	16-bit timer 01 reload data RR017 = MSB RR010 = LSB	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR016**						X		
		D5	**RR015**						X		
		D4	**RR014**						X		
		D3	**RR013**						X		
		D2	**RR012**						X		
		D1	**RR011**						X		
		D0	**RR010**						X		
16-bit timer 00 comparison data register	**0040184 (B)**	D7	**CR007**	16-bit timer 00 comparison data CR007 = MSB CR000 = LSB	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR006**						X		
		D5	**CR005**						X		
		D4	**CR004**						X		
		D3	**CR003**						X		
		D2	**CR002**						X		
		D1	**CR001**						X		
		D0	**CR000**						X		
16-bit timer 01 comparison data register	**0040185 (B)**	D7	**CR017**	16-bit timer 01 comparison data CR017 = MSB CR010 = LSB	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR016**						X		
		D5	**CR015**						X		
		D4	**CR014**						X		
		D3	**CR013**						X		
		D2	**CR012**						X		
		D1	**CR011**						X		
		D0	**CR010**						X		
16-bit timer 00 counter data register	**0040186 (B)**	D7	**TC007**	16-bit timer 00 counter data TC007 = MSB TC000 = LSB	0 to 255				X	R	Used as the low-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC006**						X		
		D5	**TC005**						X		
		D4	**TC004**						X		
		D3	**TC003**						X		
		D2	**TC002**						X		
		D1	**TC001**						X		
		D0	**TC000**						X		
16-bit timer 01 counter data register	**0040187 (B)**	D7	**TC017**	16-bit timer 01 counter data TC017 = MSB TC010 = LSB	0 to 255				X	R	Used as the high-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC016**						X		
		D5	**TC015**						X		
		D4	**TC014**						X		
		D3	**TC013**						X		
		D2	**TC012**						X		
		D1	**TC011**						X		
		D0	**TC010**						X		

*1: Invalid in 16-bit mode (fixed at 0).

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 10 control register	**0040188 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**MODE16**	16-bit timer 1 mode selection	1	16 bits	0	8 bits	0	R/W	
		D3	**CKSL10**	16-bit timer 10 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	**PTM10**	16-bit timer 10 clock output control	1	On	0	Off	0	R/W	*1
		D1	**PSET10**	16-bit timer 10 preset	1	Preset	0	Invalid	0	W	0 when being read.
		D0	**PRUN10**	16-bit timer 10 Run/Stop control	1	Run	0	Stop	0	R/W	
16-bit timer 11 control register	**0040189 (B)**	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**CKSL11**	16-bit timer 11 input clock selection	1	Invalid	0	Internal clock	0	R/W	*2
		D2	**PTM11**	16-bit timer 11 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET11**	16-bit timer 11 preset	1	Preset	0	Invalid	0	W	0 when being read.*1
		D0	**PRUN11**	16-bit timer 11 Run/Stop control	1	Run	0	Stop	0	R/W	*1
16-bit timer 10 reload data register	**004018A (B)**	D7	**RR107**	16-bit timer 10 reload data RR107 = MSB RR100 = LSB	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR106**						X		
		D5	**RR105**						X		
		D4	**RR104**						X		
		D3	**RR103**						X		
		D2	**RR102**						X		
		D1	**RR101**						X		
		D0	**RR100**						X		
16-bit timer 11 reload data register	**004018B (B)**	D7	**RR117**	16-bit timer 11 reload data RR117 = MSB RR110 = LSB	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR116**						X		
		D5	**RR115**						X		
		D4	**RR114**						X		
		D3	**RR113**						X		
		D2	**RR112**						X		
		D1	**RR111**						X		
		D0	**RR110**						X		
16-bit timer 10 comparison data register	**004018C (B)**	D7	**CR107**	16-bit timer 10 comparison data CR107 = MSB CR100 = LSB	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR106**						X		
		D5	**CR105**						X		
		D4	**CR104**						X		
		D3	**CR103**						X		
		D2	**CR102**						X		
		D1	**CR101**						X		
		D0	**CR100**						X		
16-bit timer 11 comparison data register	**004018D (B)**	D7	**CR117**	16-bit timer 11 comparison data CR117 = MSB CR110 = LSB	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR116**						X		
		D5	**CR115**						X		
		D4	**CR114**						X		
		D3	**CR113**						X		
		D2	**CR112**						X		
		D1	**CR111**						X		
		D0	**CR110**						X		
16-bit timer 10 counter data register	**004018E (B)**	D7	**TC107**	16-bit timer 10 counter data TC107 = MSB TC100 = LSB	0 to 255				X	R	Used as the low-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC106**						X		
		D5	**TC105**						X		
		D4	**TC104**						X		
		D3	**TC103**						X		
		D2	**TC102**						X		
		D1	**TC101**						X		
		D0	**TC100**						X		
16-bit timer 11 counter data register	**004018F (B)**	D7	**TC117**	16-bit timer 11 counter data TC117 = MSB TC110 = LSB	0 to 255				X	R	Used as the high-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC116**						X		
		D5	**TC115**						X		
		D4	**TC114**						X		
		D3	**TC113**						X		
		D2	**TC112**						X		
		D1	**TC111**						X		
		D0	**TC110**						X		

*1: Invalid in 16-bit mode (fixed at 0).

*2: Invalid in 16-bit mode (fixed at 0). The timer does not work if "1" (external clock) is set in 8-bit mode.

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 20 control register	0040190 (B)	D7–5	–	reserved	–				–	–	0 when being read.
		D4	**MODE16**	16-bit timer 2 mode selection	1	16 bits	0	8 bits	0	R/W	
		D3	**CKSL20**	16-bit timer 20 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	**PTM20**	16-bit timer 20 clock output control	1	Invalid	0	Off	0	R/W	*3
		D1	**PSET20**	16-bit timer 20 preset	1	Preset	0	Invalid	0	W	0 when being read.
		D0	**PRUN20**	16-bit timer 20 Run/Stop control	1	Run	0	Stop	0	R/W	
16-bit timer 21 control register	0040191 (B)	D7–4	–	reserved	–				–	–	0 when being read.
		D3	**CKSL21**	16-bit timer 21 input clock selection	1	Invalid	0	Internal clock	0	R/W	*2
		D2	**PTM21**	16-bit timer 21 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET21**	16-bit timer 21 preset	1	Preset	0	Invalid	0	W	0 when being read.*1
		D0	**PRUN21**	16-bit timer 21 Run/Stop control	1	Run	0	Stop	0	R/W	*1
16-bit timer 20 reload data register	**0040192 (B)**	D7	**RR207**	16-bit timer 20 reload data RR207 = MSB RR200 = LSB	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR206**						X		
		D5	**RR205**						X		
		D4	**RR204**						X		
		D3	**RR203**						X		
		D2	**RR202**						X		
		D1	**RR201**						X		
		D0	**RR200**						X		
16-bit timer 21 reload data register	**0040193 (B)**	D7	**RR217**	16-bit timer 21 reload data RR217 = MSB RR210 = LSB	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR216**						X		
		D5	**RR215**						X		
		D4	**RR214**						X		
		D3	**RR213**						X		
		D2	**RR212**						X		
		D1	**RR211**						X		
		D0	**RR210**						X		
16-bit timer 20 comparison data register	**0040194 (B)**	D7	**CR207**	16-bit timer 20 comparison data CR207 = MSB CR200 = LSB	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR206**						X		
		D5	**CR205**						X		
		D4	**CR204**						X		
		D3	**CR203**						X		
		D2	**CR202**						X		
		D1	**CR201**						X		
		D0	**CR200**						X		
16-bit timer 21 comparison data register	**0040195 (B)**	D7	**CR217**	16-bit timer 21 comparison data CR217 = MSB CR210 = LSB	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR216**						X		
		D5	**CR215**						X		
		D4	**CR214**						X		
		D3	**CR213**						X		
		D2	**CR212**						X		
		D1	**CR211**						X		
		D0	**CR210**						X		
16-bit timer 20 counter data register	**0040196 (B)**	D7	**TC207**	16-bit timer 20 counter data TC207 = MSB TC200 = LSB	0 to 255				X	R	Used as the low-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC206**						X		
		D5	**TC205**						X		
		D4	**TC204**						X		
		D3	**TC203**						X		
		D2	**TC202**						X		
		D1	**TC201**						X		
		D0	**TC200**						X		
16-bit timer 21 counter data register	**0040197 (B)**	D7	**TC217**	16-bit timer 21 counter data TC217 = MSB TC210 = LSB	0 to 255				X	R	Used as the high-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC216**						X		
		D5	**TC215**						X		
		D4	**TC214**						X		
		D3	**TC213**						X		
		D2	**TC212**						X		
		D1	**TC211**						X		
		D0	**TC210**						X		

*1: Invalid in 16-bit mode (fixed at 0).

*2: Invalid in 16-bit mode (fixed at 0). The timer does not work if "1" (external clock) is set in 8-bit mode.

*3: This timer has no clock output terminal, so a clock cannot be output even if this bit is set to "1" (output On).

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 30 control register	**0040198 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**MODE16**	16-bit timer 3 mode selection	1	16 bits	0	8 bits	0	R/W	
		D3	**CKSL30**	16-bit timer 30 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	**PTM30**	16-bit timer 30 clock output control	1	Invalid	0	Off	0	R/W	∗3
		D1	**PSET30**	16-bit timer 30 preset	1	Preset	0	Invalid	0	W	0 when being read.
		D0	**PRUN30**	16-bit timer 30 Run/Stop control	1	Run	0	Stop	0	R/W	
16-bit timer 31 control register	**0040199 (B)**	D7–4	–	reserved	–				–	–	0 when being read.
		D3	**CKSL31**	16-bit timer 31 input clock selection	1	Invalid	0	Internal clock	0	R/W	∗2
		D2	**PTM31**	16-bit timer 31 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET31**	16-bit timer 31 preset	1	Preset	0	Invalid	0	W	0 when being read.∗1
		D0	**PRUN31**	16-bit timer 31 Run/Stop control	1	Run	0	Stop	0	R/W	∗1
16-bit timer 30 reload data register	**004019A (B)**	D7	**RR307**	16-bit timer 30 reload data RR307 = MSB RR300 = LSB	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR306**						X		
		D5	**RR305**						X		
		D4	**RR304**						X		
		D3	**RR303**						X		
		D2	**RR302**						X		
		D1	**RR301**						X		
		D0	**RR300**						X		
16-bit timer 31 reload data register	**004019B (B)**	D7	**RR317**	16-bit timer 31 reload data RR317 = MSB RR310 = LSB	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR316**						X		
		D5	**RR315**						X		
		D4	**RR314**						X		
		D3	**RR313**						X		
		D2	**RR312**						X		
		D1	**RR311**						X		
		D0	**RR310**						X		
16-bit timer 30 comparison data register	**004019C (B)**	D7	**CR307**	16-bit timer 30 comparison data CR307 = MSB CR300 = LSB	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR306**						X		
		D5	**CR305**						X		
		D4	**CR304**						X		
		D3	**CR303**						X		
		D2	**CR302**						X		
		D1	**CR301**						X		
		D0	**CR300**						X		
16-bit timer 31 comparison data register	**004019D (B)**	D7	**CR317**	16-bit timer 31 comparison data CR317 = MSB CR310 = LSB	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR316**						X		
		D5	**CR315**						X		
		D4	**CR314**						X		
		D3	**CR313**						X		
		D2	**CR312**						X		
		D1	**CR311**						X		
		D0	**CR310**						X		
16-bit timer 30 counter data register	**004019E (B)**	D7	**TC307**	16-bit timer 30 counter data TC307 = MSB TC300 = LSB	0 to 255				X	R	Used as the low-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC306**						X		
		D5	**TC305**						X		
		D4	**TC304**						X		
		D3	**TC303**						X		
		D2	**TC302**						X		
		D1	**TC301**						X		
		D0	**TC300**						X		
16-bit timer 31 counter data register	**004019F (B)**	D7	**TC317**	16-bit timer 31 counter data TC317 = MSB TC310 = LSB	0 to 255				X	R	Used as the high-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC316**						X		
		D5	**TC315**						X		
		D4	**TC314**						X		
		D3	**TC313**						X		
		D2	**TC312**						X		
		D1	**TC311**						X		
		D0	**TC310**						X		

∗1: Invalid in 16-bit mode (fixed at 0).

∗2: Invalid in 16-bit mode (fixed at 0). The timer does not work if "1" (external clock) is set in 8-bit mode.

∗3: This timer has no clock output terminal, so a clock cannot be output even if this bit is set to "1" (output On).

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 40 control register	**00401A0 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**MODE16**	16-bit timer 4 mode selection	1	16 bits	0	8 bits	0	R/W	
		D3	**CKSL40**	16-bit timer 40 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	**PTM40**	16-bit timer 40 clock output control	1	Invalid	0	Off	0	R/W	*3
		D1	**PSET40**	16-bit timer 40 preset	1	Preset	0	Invalid	0	W	0 when being read.
		D0	**PRUN40**	16-bit timer 40 Run/Stop control	1	Run	0	Stop	0	R/W	
16-bit timer 41 control register	**00401A1 (B)**	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**CKSL41**	16-bit timer 41 input clock selection	1	Invalid	0	Internal clock	0	R/W	*2
		D2	**PTM41**	16-bit timer 41 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET41**	16-bit timer 41 preset	1	Preset	0	Invalid	0	W	0 when being read.*1
		D0	**PRUN41**	16-bit timer 41 Run/Stop control	1	Run	0	Stop	0	R/W	*1
16-bit timer 40 reload data register	**00401A2 (B)**	D7	**RR407**	16-bit timer 40 reload data RR407 = MSB RR400 = LSB	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR406**						X		
		D5	**RR405**						X		
		D4	**RR404**						X		
		D3	**RR403**						X		
		D2	**RR402**						X		
		D1	**RR401**						X		
		D0	**RR400**						X		
16-bit timer 41 reload data register	**00401A3 (B)**	D7	**RR417**	16-bit timer 41 reload data RR417 = MSB RR410 = LSB	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR416**						X		
		D5	**RR415**						X		
		D4	**RR414**						X		
		D3	**RR413**						X		
		D2	**RR412**						X		
		D1	**RR411**						X		
		D0	**RR410**						X		
16-bit timer 40 comparison data register	**00401A4 (B)**	D7	**CR407**	16-bit timer 40 comparison data CR407 = MSB CR400 = LSB	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR406**						X		
		D5	**CR405**						X		
		D4	**CR404**						X		
		D3	**CR403**						X		
		D2	**CR402**						X		
		D1	**CR401**						X		
		D0	**CR400**						X		
16-bit timer 41 comparison data register	**00401A5 (B)**	D7	**CR417**	16-bit timer 41 comparison data CR417 = MSB CR410 = LSB	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR416**						X		
		D5	**CR415**						X		
		D4	**CR414**						X		
		D3	**CR413**						X		
		D2	**CR412**						X		
		D1	**CR411**						X		
		D0	**CR410**						X		
16-bit timer 40 counter data register	**00401A6 (B)**	D7	**TC407**	16-bit timer 40 counter data TC407 = MSB TC400 = LSB	0 to 255				X	R	Used as the low-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC406**						X		
		D5	**TC405**						X		
		D4	**TC404**						X		
		D3	**TC403**						X		
		D2	**TC402**						X		
		D1	**TC401**						X		
		D0	**TC400**						X		
16-bit timer 41 counter data register	**00401A7 (B)**	D7	**TC417**	16-bit timer 41 counter data TC417 = MSB TC410 = LSB	0 to 255				X	R	Used as the high-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC416**						X		
		D5	**TC415**						X		
		D4	**TC414**						X		
		D3	**TC413**						X		
		D2	**TC412**						X		
		D1	**TC411**						X		
		D0	**TC410**						X		

*1: Invalid in 16-bit mode (fixed at 0).

*2: Invalid in 16-bit mode (fixed at 0). The timer does not work if "1" (external clock) is set in 8-bit mode.

*3: This timer has no clock output terminal, so a clock cannot be output even if this bit is set to "1" (output On).

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 50 control register	**00401A8 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**MODE16**	16-bit timer 5 mode selection	1	16 bits	0	8 bits	0	R/W	
		D3	**CKSL50**	16-bit timer 50 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	**PTM50**	16-bit timer 50 clock output control	1	Invalid	0	Off	0	R/W	*3
		D1	**PSET50**	16-bit timer 50 preset	1	Preset	0	Invalid	0	W	0 when being read.
		D0	**PRUN50**	16-bit timer 50 Run/Stop control	1	Run	0	Stop	0	R/W	
16-bit timer 51 control register	**00401A9 (B)**	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**CKSL51**	16-bit timer 51 input clock selection	1	Invalid	0	Internal clock	0	R/W	*2
		D2	**PTM51**	16-bit timer 51 clock output control	1	On	0	Off	0	R/W	
		D1	**PSET51**	16-bit timer 51 preset	1	Preset	0	Invalid	0	W	0 when being read.*1
		D0	**PRUN51**	16-bit timer 51 Run/Stop control	1	Run	0	Stop	0	R/W	*1
16-bit timer 50 reload data register	**00401AA (B)**	D7	**RR507**	16-bit timer 50 reload data RR507 = MSB RR500 = LSB	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR506**						X		
		D5	**RR505**						X		
		D4	**RR504**						X		
		D3	**RR503**						X		
		D2	**RR502**						X		
		D1	**RR501**						X		
		D0	**RR500**						X		
16-bit timer 51 reload data register	**00401AB (B)**	D7	**RR517**	16-bit timer 51 reload data RR517 = MSB RR510 = LSB	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit reload data in 16-bit mode.
		D6	**RR516**						X		
		D5	**RR515**						X		
		D4	**RR514**						X		
		D3	**RR513**						X		
		D2	**RR512**						X		
		D1	**RR511**						X		
		D0	**RR510**						X		
16-bit timer 50 comparison data register	**00401AC (B)**	D7	**CR507**	16-bit timer 50 comparison data CR507 = MSB CR500 = LSB	0 to 255				X	R/W	Used as the low-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR506**						X		
		D5	**CR505**						X		
		D4	**CR504**						X		
		D3	**CR503**						X		
		D2	**CR502**						X		
		D1	**CR501**						X		
		D0	**CR500**						X		
16-bit timer 51 comparison data register	**00401AD (B)**	D7	**CR517**	16-bit timer 51 comparison data CR517 = MSB CR510 = LSB	0 to 255				X	R/W	Used as the high-order 8 bits of 16-bit compare data in 16-bit mode.
		D6	**CR516**						X		
		D5	**CR515**						X		
		D4	**CR514**						X		
		D3	**CR513**						X		
		D2	**CR512**						X		
		D1	**CR511**						X		
		D0	**CR510**						X		
16-bit timer 50 counter data register	**00401AE (B)**	D7	**TC507**	16-bit timer 50 counter data TC507 = MSB TC500 = LSB	0 to 255				X	R	Used as the low-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC506**						X		
		D5	**TC505**						X		
		D4	**TC504**						X		
		D3	**TC503**						X		
		D2	**TC502**						X		
		D1	**TC501**						X		
		D0	**TC500**						X		
16-bit timer 51 counter data register	**00401AF (B)**	D7	**TC517**	16-bit timer 51 counter data TC517 = MSB TC510 = LSB	0 to 255				X	R	Used as the high-order 8 bits of 16-bit data in 16-bit mode.
		D6	**TC516**						X		
		D5	**TC515**						X		
		D4	**TC514**						X		
		D3	**TC513**						X		
		D2	**TC512**						X		
		D1	**TC511**						X		
		D0	**TC510**						X		

*1: Invalid in 16-bit mode (fixed at 0).

*2: Invalid in 16-bit mode (fixed at 0). The timer does not work if "1" (external clock) is set in 8-bit mode.

*3: This timer has no clock output terminal, so a clock cannot be output even if this bit is set to "1" (output On).

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
16-bit timer 0–1, DMA interrupt priority register	**0040261 (B)**	D7	**–**	reserved	–				–	–	0 when being read.
		D6	**P16T02**	16-bit timer 0–1 interrupt level	0 to 7				X	R/W	
		D5	**P16T01**						X		
		D4	**P16T00**						X		
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**PDM2**	IDMA, high-speed DMA interrupt level	0 to 7				X	R/W	
		D1	**PDM1**						X		
		D0	**PDM0**						X		
16-bit timer 2–5 interrupt priority register	**0040262 (B)**	D7	**–**	reserved	–				–	–	0 when being read.
		D6	**P16T42**	16-bit timer 4–5 interrupt level	0 to 7				X	R/W	
		D5	**P16T41**						X		
		D4	**P16T40**						X		
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**P16T22**	16-bit timer 2–3 interrupt level	0 to 7				X	R/W	
		D1	**P16T21**						X		
		D0	**P16T20**						X		
16-bit timer 0–1 interrupt enable register	**0040272 (B)**	D7	**E16TC11**	16-bit timer 11 comparison match	1	Enabled	0	Disabled	0	R/W	
		D6	**E16TU11**	16-bit timer 11 underflow					0	R/W	
		D5	**E16TC10**	16-bit timer 10 comparison match					0	R/W	
		D4	**E16TU10**	16-bit timer 10 underflow					0	R/W	
		D3	**E16TC01**	16-bit timer 01 comparison match					0	R/W	
		D2	**E16TU01**	16-bit timer 01 underflow					0	R/W	
		D1	**E16TC00**	16-bit timer 00 comparison match					0	R/W	
		D0	**E16TU00**	16-bit timer 00 underflow					0	R/W	
16-bit timer 2–3 interrupt enable register	**0040273 (B)**	D7	**E16TC31**	16-bit timer 31 comparison match	1	Enabled	0	Disabled	0	R/W	
		D6	**E16TU31**	16-bit timer 31 underflow					0	R/W	
		D5	**E16TC30**	16-bit timer 30 comparison match					0	R/W	
		D4	**E16TU30**	16-bit timer 30 underflow					0	R/W	
		D3	**E16TC21**	16-bit timer 21 comparison match					0	R/W	
		D2	**E16TU21**	16-bit timer 21 underflow					0	R/W	
		D1	**E16TC20**	16-bit timer 20 comparison match					0	R/W	
		D0	**E16TU20**	16-bit timer 20 underflow					0	R/W	
16-bit timer 4–5 interrupt enable register	**0040274 (B)**	D7	**E16TC51**	16-bit timer 51 comparison match	1	Enabled	0	Disabled	0	R/W	
		D6	**E16TU51**	16-bit timer 51 underflow					0	R/W	
		D5	**E16TC50**	16-bit timer 50 comparison match					0	R/W	
		D4	**E16TU50**	16-bit timer 50 underflow					0	R/W	
		D3	**E16TC41**	16-bit timer 41 comparison match					0	R/W	
		D2	**E16TU41**	16-bit timer 41 underflow					0	R/W	
		D1	**E16TC40**	16-bit timer 40 comparison match					0	R/W	
		D0	**E16TU40**	16-bit timer 40 underflow					0	R/W	
16-bit timer 0–1 interrupt factor flag register	**0040282 (B)**	D7	**F16TC11**	16-bit timer 11 comparison match	1	Factor is generated	0	No factor is generated	X	R/W	
		D6	**F16TU11**	16-bit timer 11 underflow					X	R/W	
		D5	**F16TC10**	16-bit timer 10 comparison match					X	R/W	
		D4	**F16TU10**	16-bit timer 10 underflow					X	R/W	
		D3	**F16TC01**	16-bit timer 01 comparison match					X	R/W	
		D2	**F16TU01**	16-bit timer 01 underflow					X	R/W	
		D1	**F16TC00**	16-bit timer 00 comparison match					X	R/W	
		D0	**F16TU00**	16-bit timer 00 underflow					X	R/W	
16-bit timer 2–3 interrupt factor flag register	**0040283 (B)**	D7	**F16TC31**	16-bit timer 31 comparison match	1	Factor is generated	0	No factor is generated	X	R/W	
		D6	**F16TU31**	16-bit timer 31 underflow					X	R/W	
		D5	**F16TC30**	16-bit timer 30 comparison match					X	R/W	
		D4	**F16TU30**	16-bit timer 30 underflow					X	R/W	
		D3	**F16TC21**	16-bit timer 21 comparison match					X	R/W	
		D2	**F16TU21**	16-bit timer 21 underflow					X	R/W	
		D1	**F16TC20**	16-bit timer 20 comparison match					X	R/W	
		D0	**F16TU20**	16-bit timer 20 underflow					X	R/W	
16-bit timer 4–5 interrupt factor flag register	**0040284 (B)**	D7	**F16TC51**	16-bit timer 51 comparison match	1	Factor is generated	0	No factor is generated	X	R/W	
		D6	**F16TU51**	16-bit timer 51 underflow					X	R/W	
		D5	**F16TC50**	16-bit timer 50 comparison match					X	R/W	
		D4	**F16TU50**	16-bit timer 50 underflow					X	R/W	
		D3	**F16TC41**	16-bit timer 41 comparison match					X	R/W	
		D2	**F16TU41**	16-bit timer 41 underflow					X	R/W	
		D1	**F16TC40**	16-bit timer 40 comparison match					X	R/W	
		D0	**F16TU40**	16-bit timer 40 underflow					X	R/W	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
K6, DMA, 16-bit timer 0 IDMA request register	**0040290 (B)**	D7	**R16TC01**	16-bit timer 01 comparison match	1	IDMA request	0	Interrupt request	X	R/W	
		D6	**R16TU01**	16-bit timer 01 underflow					X	R/W	
		D5	**RHDM1**	High-speed DMA Ch.1					X	R/W	
		D4	**RHDM0**	High-speed DMA Ch.0					X	R/W	
		D3	**RK64**	K64 input					X	R/W	
		D2	**RK65**	K65 input					X	R/W	
		D1	**RK66**	K66 input					X	R/W	
		D0	**RK67**	K67 input					X	R/W	
16-bit timer 1–4 IDMA request register	**0040291 (B)**	D7	**R16TC41**	16-bit timer 41 comparison match	1	IDMA request	0	Interrupt request	X	R/W	
		D6	**R16TU41**	16-bit timer 41 underflow					X	R/W	
		D5	**R16TC31**	16-bit timer 31 comparison match					X	R/W	
		D4	**R16TU31**	16-bit timer 31 underflow					X	R/W	
		D3	**R16TC21**	16-bit timer 21 comparison match					X	R/W	
		D2	**R16TU21**	16-bit timer 21 underflow					X	R/W	
		D1	**R16TC11**	16-bit timer 11 comparison match					X	R/W	
		D0	**R16TU11**	16-bit timer 11 underflow					X	R/W	
16-bit timer 5, 8-bit timer, serial I/F Ch.0 IDMA request register	**0040292 (B)**	D7	**RSTX0**	SIF Ch.0 transmit buffer empty	1	IDMA request	0	Interrupt request	X	R/W	
		D6	**RSRX0**	SIF Ch.0 receive buffer full					X	R/W	
		D5	**R8TU3**	8-bit timer 3 underflow					X	R/W	
		D4	**R8TU2**	8-bit timer 2 underflow					X	R/W	
		D3	**R8TU1**	8-bit timer 1 underflow					X	R/W	
		D2	**R8TU0**	8-bit timer 0 underflow					X	R/W	
		D1	**R16TC51**	16-bit timer 51 comparison match					X	R/W	
		D0	**R16TU51**	16-bit timer 51 underflow					X	R/W	
R7 function select register	**00402CA (B)**	D7	**CFR77**	R77 function selection	1	TM51	0	R77	0	W	Undefined in read.
		D6	**CFR76**	R76 function selection	1	TM41	0	R76	0	W	
		D5	**CFR75**	R75 function selection	1	TM31	0	R75	0	W	
		D4	**CFR74**	R74 function selection	1	TM21	0	R74	0	W	
		D3	**CFR73**	R73 function selection	1	TM11	0	R73	0	W	
		D2	**CFR72**	R72 function selection	1	TM10	0	R72	0	W	
		D1	**CFR71**	R71 function selection	1	TM01	0	R71	0	W	
		D0	**CFR70**	R70 function selection	1	TM00	0	R70	0	W	
P1 function select register	**00402D4 (B)**	D7	**–**	reserved	–				–	–	
		D6	**CFP16**	P16 function selection	1	EXCL50 #DMAEND1	0	P16	0	W	Undefined in read.
		D5	**CFP15**	P15 function selection	1	EXCL40 #DMAEND0	0	P15	0	W	
		D4	**CFP14**	P14 function selection	1	EXCL30 #BUSGET	0	P14	0	W	
		D3	**CFP13**	P13 function selection	1	EXCL20 T8UF3	0	P13	0	W	
		D2	**CFP12**	P12 function selection	1	EXCL10 T8UF2	0	P12	0	W	
		D1	**CFP11**	P11 function selection	1	EXCL01 T8UF1	0	P11	0	W	
		D0	**CFP10**	P10 function selection	1	EXCL00 T8UF0	0	P10	0	W	
P1 I/O control register	**00402D6 (B)**	D7	**–**	reserved	–				–	–	
		D6	**IOC16**	P16 I/O control	1	Output	0	Input	0	W	Undefined in read.
		D5	**IOC15**	P15 I/O control					0	W	
		D4	**IOC14**	P14 I/O control					0	W	
		D3	**IOC13**	P13 I/O control					0	W	
		D2	**IOC12**	P12 I/O control					0	W	
		D1	**IOC11**	P11 I/O control					0	W	
		D0	**IOC10**	P10 I/O control					0	W	
P1 pull-up control register	**00402D7 (B)**	D7	**–**	reserved	–				–	–	
		D6	**IOU16**	P16 pull-up control	1	Pulled up	0	No pull-up	0	W	Undefined in read.
		D5	**IOU15**	P15 pull-up control					0	W	
		D4	**IOU14**	P14 pull-up control					0	W	
		D3	**IOU13**	P13 pull-up control					0	W	
		D2	**IOU12**	P12 pull-up control					0	W	
		D1	**IOU11**	P11 pull-up control					0	W	
		D0	**IOU10**	P10 pull-up control					0	W	

CFR77–CFR70: R7[7:0] pin function selection (D[7:0]) / R7 function select register (0x402CA)

Selects the pin used for clock output.

Write "1": Clock output pin
Write "0": Output port pin
Read: Invalid

Select the pin to be used to output a timer-generated clock to external devices from among R70 through R77, by writing "1" to CFR70–CFR77. For the relationship between each pin and timer, refer to Table 11.4.1. The pin is set for an output port by writing "0" to CFR7x.
Since the function select register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite the register. Use ordinary storage instructions.
At cold start, CFR7x is set to "0" (output port). At hot start, CFR7x retains its status from prior to the reset.

CFP16–CFP10: P1[6:0] pin function selection (D[6:0]) / P1 function select register (0x402D4)

Selects the pin to be used for input of an external count clock to the timer.

Write "1": Clock input pin
Write "0": I/O port pin
Read: Invalid

Select clock input pins for the timers that are used as an event counter from among P10 through P16, by writing "1" to CFP10–CFP16. For the relationship between each pin and timer, refer to Table 11.4.1. The pin is set for an I/O port by writing "0" to CFP1x.
In addition to pin selection here, the pin to be used for clock input to the 16-bit programmable timer must be set to input mode using the I/O control register.
Since the pin function select register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite the register. Use ordinary storage instructions.
At cold start, CFP1x is set to "0" (I/O port). At hot start, CFP1x retains its status from prior to the reset.

IOC16–IOC10: P1[6:0] port I/O control (D[6:0]) / P1 I/O control register (0x402D6)

Directs P10 through P13 for input or output.

Write "1": Output mode
Write "0": Input mode
Read: Invalid

For the pin selected from among P10 through P16 for use for external clock input, write "0" to the corresponding I/O control bit to set it to input mode . If the pin is set to output mode, even though its CFP1x may be set to "1", it functions as the output pin of an 8-bit programmable timer and cannot be used to receive an external clock.
Since the IOC register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite the register.
At cold start, all IOCs are set to "0" (input mode). At hot start, they retain their state from prior to the an initial reset.

IOU16–IOU10: P1[6:0] pull-up control (D[6:0]) / P1 pull-up control register (0x402D7)

Controls the pull-up of the clock input pin.

Write "1": Pulled up
Write "0": Not pulled up
Read: Invalid

When a bit of the IOU register is set to "1", the internal pull-up resistor of the corresponding pin is enabled. The pins for which bits are set to "0" are not pulled up.
Since the IOU register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite the register. Use ordinary storage instructions.
At cold start, IOU1x is set to "0" (not pulled up). At hot start, they retain their state from prior to the initial reset.

MODE16: Timer 0 8/16-bit mode selection (D4) / 16-bit timer 00 control register (0x40180)
MODE16: Timer 1 8/16 bit mode selection (D4) / 16-bit timer 10 control register (0x40188)
MODE16: Timer 2 8/16 bit mode selection (D4) / 16-bit timer 20 control register (0x40190)
MODE16: Timer 3 8/16 bit mode selection (D4) / 16-bit timer 30 control register (0x40198)
MODE16: Timer 4 8/16 bit mode selection (D4) / 16-bit timer 40 control register (0x401A0)
MODE16: Timer 5 8/16 bit mode selection (D4) / 16-bit timer 50 control register (0x401A8)

Selects the operation mode of each timer.

Write "1": 16-bit mode
Write "0": 8-bit mode
Read: Valid

Select timers x0 and x1 for use as independent 2-channel 8-bit timers (8-bit mode), or use them as a combined 1-channel 16-bit timer (16-bit mode). The 8-bit mode is selected by writing "0" to MODE16, and the 16-bit mode is selected by writing "1".
At initial reset, MODE16 is set to "0" (8-bit mode).

CKSL00: Timer 00 input clock selection (D3) / 16-bit timer 00 control register (0x40180)
CKSL01: Timer 01 input clock selection (D3) / 16-bit timer 01 control register (0x40181)
CKSL10: Timer 10 input clock selection (D3) / 16-bit timer 10 control register (0x40188)
CKSL20: Timer 20 input clock selection (D3) / 16-bit timer 20 control register (0x40190)
CKSL30: Timer 30 input clock selection (D3) / 16-bit timer 30 control register (0x40198)
CKSL40: Timer 40 input clock selection (D3) / 16-bit timer 40 control register (0x401A0)
CKSL50: Timer 50 input clock selection (D3) / 16-bit timer 50 control register (0x401A8)

Selects the input clock of each timer.

Write "1": External clock
Write "0": Internal clock
Read: Valid

The internal clock (prescaler output) is selected for the input clock of each timer by writing "0" to CKSLxx. An external clock (one that is fed from the clock input pin) is selected by writing "1", and the timer functions as an event counter. In this case, the clock input pin must be set using CFP1x before an external clock is selected here.
Although timers x1 other than the above include a CKSLx1 bit, this bit should be fixed at "0" (internal clock) because it has no external-clock input function. These timer does not operate when CKSLx1 is set to "1" (external clock).
When timer 0 is used in the 16-bit mode, the settings of CKSL01 are ignored.
At initial reset, CKSLxx is set to "0" (internal clock).

PTM00: Timer 00 clock output control (D2) / 16-bit timer 00 control register (0x40180)
PTM01: Timer 01 clock output control (D2) / 16-bit timer 01 control register (0x40181)
PTM10: Timer 10 clock output control (D2) / 16-bit timer 10 control register (0x40188)
PTM11: Timer 11 clock output control (D2) / 16-bit timer 11 control register (0x40189)
PTM21: Timer 21 clock output control (D2) / 16-bit timer 21 control register (0x40191)
PTM31: Timer 31 clock output control (D2) / 16-bit timer 31 control register (0x40199)
PTM41: Timer 41 clock output control (D2) / 16-bit timer 41 control register (0x401A1)
PTM51: Timer 51 clock output control (D2) / 16-bit timer 51 control register (0x401A9)

Controls the output of the TMxx signal (timer output clock).

Write "1": On
Write "0": Off
Read: Valid

The TMxx signal is output from the clock output pin by writing "1" to PTMxx. The clock output is turned off by writing "0", and the output pin is on a low level. In this case, the clock output pin must be set using CFR7x before outputting the TMxx signal here.

Although timers x0 other than the above include a PTMx0 bit, no clock is output even if "1" is set to PTMx0 because the timers have no the clock output function.
When timers 0 and 1 are used in the 16-bit mode, settings of PTM00 and PTM10 are ignored.
At initial reset, PTMxx is set to "0" (off).

PSET00: Timer 00 preset (D1) / 16-bit timer 00 control register (0x40180)
PSET01: Timer 01 preset (D1) / 16-bit timer 01 control register (0x40181)
PSET10: Timer 10 preset (D1) / 16-bit timer 10 control register (0x40188)
PSET11: Timer 11 preset (D1) / 16-bit timer 11 control register (0x40189)
PSET20: Timer 20 preset (D1) / 16-bit timer 20 control register (0x40190)
PSET21: Timer 21 preset (D1) / 16-bit timer 21 control register (0x40191)
PSET30: Timer 30 preset (D1) / 16-bit timer 30 control register (0x40198)
PSET31: Timer 31 preset (D1) / 16-bit timer 31 control register (0x40199)
PSET40: Timer 40 preset (D1) / 16-bit timer 40 control register (0x401A0)
PSET41: Timer 41 preset (D1) / 16-bit timer 41 control register (0x401A1)
PSET50: Timer 50 preset (D1) / 16-bit timer 50 control register (0x401A8)
PSET51: Timer 51 preset (D1) / 16-bit timer 51 control register (0x401A9)

Presets reload data in the counter.

Write "1":	Preset
Write "0":	Invalid
Read:	Always "0"

The reload data in RRxx is preset in the counter of timer xx by writing "1" to PSETxx. If the counter is preset when in a RUN state, the counter starts counting immediately after the reload data is preset.
If preset in a STOP state, the reload data that has been preset is retained intact.
Writing "0" results in No Operation.
If the 16-bit mode is selected, writing "1" to PSETx1 is ignored because 16 bits of data are preset using PSETx0.
Since PSETxx is a write-only bit, its content when read is always "0".

PRUN00: Timer 00 RUN/STOP control (D0) / 16-bit timer 00 control register (0x40180)
PRUN01: Timer 01 RUN/STOP control (D0) / 16-bit timer 01 control register (0x40181)
PRUN10: Timer 10 RUN/STOP control (D0) / 16-bit timer 10 control register (0x40188)
PRUN11: Timer 11 RUN/STOP control (D0) / 16-bit timer 11 control register (0x40189)
PRUN20: Timer 20 RUN/STOP control (D0) / 16-bit timer 20 control register (0x40190)
PRUN21: Timer 21 RUN/STOP control (D0) / 16-bit timer 21 control register (0x40191)
PRUN30: Timer 30 RUN/STOP control (D0) / 16-bit timer 30 control register (0x40198)
PRUN31: Timer 31 RUN/STOP control (D0) / 16-bit timer 31 control register (0x40199)
PRUN40: Timer 40 RUN/STOP control (D0) / 16-bit timer 40 control register (0x401A0)
PRUN41: Timer 41 RUN/STOP control (D0) / 16-bit timer 41 control register (0x401A1)
PRUN50: Timer 50 RUN/STOP control (D0) / 16-bit timer 50 control register (0x401A8)
PRUN51: Timer 51 RUN/STOP control (D0) / 16-bit timer 51 control register (0x401A9)

Controls the timer's RUN/STOP state.

Write "1":	RUN
Write "0":	STOP
Read:	Valid

Each timer is made to start counting down by writing "1" to PRUNxx and made to stop counting by writing "0".
In the STOP state, the counter data is retained until the timer is preset with reload data or placed in a RUN state. By changing states from STOP to RUN, the timer can restart counting beginning at the retained count.
If the 16-bit mode is selected, PRUNx0 is used to control RUN/STOP. In this case, PRUNx1 is fixed at "0".
At initial reset, PRUN is set to "0" (STOP).

RR007–RR000: Timer 00 reload data (D[7:0]) / 16-bit timer 00 reload data register (0x40182)
RR017–RR010: Timer 01 reload data (D[7:0]) / 16-bit timer 01 reload data register (0x40183)
RR107–RR100: Timer 10 reload data (D[7:0]) / 16-bit timer 10 reload data register (0x4018A)
RR117–RR110: Timer 11 reload data (D[7:0]) / 16-bit timer 11 reload data register (0x4018B)
RR207–RR200: Timer 20 reload data (D[7:0]) / 16-bit timer 20 reload data register (0x40192)
RR217–RR210: Timer 21 reload data (D[7:0]) / 16-bit timer 21 reload data register (0x40193)
RR307–RR300: Timer 30 reload data (D[7:0]) / 16-bit timer 30 reload data register (0x4019A)
RR317–RR310: Timer 31 reload data (D[7:0]) / 16-bit timer 31 reload data register (0x4019B)
RR407–RR400: Timer 40 reload data (D[7:0]) / 16-bit timer 40 reload data register (0x401A2)
RR417–RR410: Timer 41 reload data (D[7:0]) / 16-bit timer 41 reload data register (0x401A3)
RR507–RR500: Timer 50 reload data (D[7:0]) / 16-bit timer 50 reload data register (0x401AA)
RR517–RR510: Timer 51 reload data (D[7:0]) / 16-bit timer 51 reload data register (0x401AB)

Sets the initial value of each timer.

The reload data set in this register is loaded into each corresponding counter, and the counter starts counting down beginning with this data, which is used as the initial count.

There are two cases in which the reload data is loaded into the counter: when presetting data after writing "1" to PSETxx, or when data is automatically reloaded in the event of counter underflow.

In the 16-bit mode, the 8 low-order bits of 16-bit reload data are stored in RRx0[7:0], and the 8 high-order bits are stored in RRx1[7:0].

This register is readable.

At initial reset, RRxx is not initialized.

CR007–CR000: Timer 00 comparison data (D[7:0]) / 16-bit timer 00 comparison data register (0x40184)
CR017–CR010: Timer 01 comparison data (D[7:0]) / 16-bit timer 01 comparison data register (0x40185)
CR107–CR100: Timer 10 comparison data (D[7:0]) / 16-bit timer 10 comparison data register (0x4018C)
CR117–CR110: Timer 11 comparison data (D[7:0]) / 16-bit timer 11 comparison data register (0x4018D)
CR207–CR200: Timer 20 comparison data (D[7:0]) / 16-bit timer 20 comparison data register (0x40194)
CR217–CR210: Timer 21 comparison data (D[7:0]) / 16-bit timer 21 comparison data register (0x40195)
CR307–CR300: Timer 30 comparison data (D[7:0]) / 16-bit timer 30 comparison data register (0x4019C)
CR317–CR310: Timer 31 comparison data (D[7:0]) / 16-bit timer 31 comparison data register (0x4019D)
CR407–CR400: Timer 40 comparison data (D[7:0]) / 16-bit timer 40 comparison data register (0x401A4)
CR417–CR410: Timer 41 comparison data (D[7:0]) / 16-bit timer 41 comparison data register (0x401A5)
CR507–CR500: Timer 50 comparison data (D[7:0]) / 16-bit timer 50 comparison data register (0x401AC)
CR517–CR510: Timer 51 comparison data (D[7:0]) / 16-bit timer 51 comparison data register (0x401AD)

Sets the comparison data of each timer.

The data set in this register is compared with each corresponding counter data. When the contents match, the timer outputs a comparison match signal. This signal controls interrupts and TMxx output waveforms.

In the 16-bit mode, the 8 low-order bits of 16-bit comparison data are stored in CRx0[7:0], and the 8 high-order bits are stored in CRx1[7:0].

This register is readable.

At initial reset, CRxx is not initialized.

TC007–TC000: Timer 00 counter data (D[7:0]) / 16-bit timer 00 counter data register (0x40186)
TC017–TC010: Timer 01 counter data (D[7:0]) / 16-bit timer 01 counter data register (0x40187)
TC107–TC100: Timer 10 counter data (D[7:0]) / 16-bit timer 10 counter data register (0x4018E)
TC117–TC110: Timer 11 counter data (D[7:0]) / 16-bit timer 11 counter data register (0x4018F)
TC207–TC200: Timer 20 counter data (D[7:0]) / 16-bit timer 20 counter data register (0x40196)
TC217–TC210: Timer 21 counter data (D[7:0]) / 16-bit timer 21 counter data register (0x40197)
TC307–TC300: Timer 30 counter data (D[7:0]) / 16-bit timer 30 counter data register (0x4019E)
TC317–TC310: Timer 31 counter data (D[7:0]) / 16-bit timer 31 counter data register (0x4019F)
TC407–TC400: Timer 40 counter data (D[7:0]) / 16-bit timer 40 counter data register (0x401A6)
TC417–TC410: Timer 41 counter data (D[7:0]) / 16-bit timer 41 counter data register (0x401A7)
TC507–TC500: Timer 50 counter data (D[7:0]) / 16-bit timer 50 counter data register (0x401AE)
TC517–TC510: Timer 51 counter data (D[7:0]) / 16-bit timer 51 counter data register (0x401AF)

The counter data of each timer can be read from this register.
The data can be read out at any time.
In the 16-bit mode, the 8 low-order bits of 16-bit data are stored in TCx0[7:0], and the 8 high-order bits are stored in TCx1[7:0]. In the 16-bit mode, data must be read out in half-word.
Since TCxx is a read-only register, writing to this register is ignored.
At initial reset, TCxx is not initialized.

P16T02–P16T00: Timer 0–1 interrupt level (D[6:4]) / 16-bit timer 0–1, DMA interrupt priority register (0x40261)
P16T22–P16T20: Timer 2–3 interrupt level (D[2:0]) / 16-bit timer 2–5 interrupt priority register (0x40262)
P16T42–P16T40: Timer 4–5 interrupt level (D[6:4]) / 16-bit timer 2–5 interrupt priority register (0x40262)

Sets the priority levels of 16-bit programmable timer interrupts.
The priority level can be set for two timer blocks each in the range of 0 to 7.
At initial reset, P16Txx becomes indeterminate.

E16TU00, E16TC00: Timer 00 interrupt enable (D0, D1) / 16-bit timer 0–1 interrupt enable register (0x40272)
E16TU01, E16TC01: Timer 01 interrupt enable (D2, D3) / 16-bit timer 0–1 interrupt enable register (0x40272)
E16TU10, E16TC10: Timer 10 interrupt enable (D4, D5) / 16-bit timer 0–1 interrupt enable register (0x40272)
E16TU11, E16TC11: Timer 11 interrupt enable (D6, D7) / 16-bit timer 0–1 interrupt enable register (0x40272)
E16TU20, E16TC20: Timer 20 interrupt enable (D0, D1) / 16-bit timer 2–3 interrupt enable register (0x40273)
E16TU21, E16TC21: Timer 21 interrupt enable (D2, D3) / 16-bit timer 2–3 interrupt enable register (0x40273)
E16TU30, E16TC30: Timer 30 interrupt enable (D4, D5) / 16-bit timer 2–3 interrupt enable register (0x40273)
E16TU31, E16TC31: Timer 31 interrupt enable (D6, D7) / 16-bit timer 2–3 interrupt enable register (0x40273)
E16TU40, E16TC40: Timer 40 interrupt enable (D0, D1) / 16-bit timer 4–5 interrupt enable register (0x40274)
E16TU41, E16TC41: Timer 41 interrupt enable (D2, D3) / 16-bit timer 4–5 interrupt enable register (0x40274)
E16TU50, E16TC50: Timer 50 interrupt enable (D4, D5) / 16-bit timer 4–5 interrupt enable register (0x40274)
E16TU51, E16TC51: Timer 51 interrupt enable (D6, D7) / 16-bit timer 4–5 interrupt enable register (0x40274)

Enables or disables the generation of an interrupt to the CPU.

Write "1":	Interrupt enabled
Write "0":	Interrupt disabled
Read:	Valid

The E16TUxx and E16TCxx are provided for the counter underflow and comparison match interrupt factors, respectively. The interrupt for which the bit is set to "1" is enabled, and the interrupt for which the bit is set to "0" is disabled.
At initial reset, these bits are set to "0" (interrupt disabled).

F16TU00, F16TC00: Timer 00 interrupt factor flag (D0, D1) / 16-bit timer 0–1 interrupt factor flag register (0x40282)
F16TU01, F16TC01: Timer 01 interrupt factor flag (D2, D3) / 16-bit timer 0–1 interrupt factor flag register (0x40282)
F16TU10, F16TC10: Timer 10 interrupt factor flag (D4, D5) / 16-bit timer 0–1 interrupt factor flag register (0x40282)
F16TU11, F16TC11: Timer 11 interrupt factor flag (D6, D7) / 16-bit timer 0–1 interrupt factor flag register (0x40282)
F16TU20, F16TC20: Timer 20 interrupt factor flag (D0, D1) / 16-bit timer 2–3 interrupt factor flag register (0x40283)
F16TU21, F16TC21: Timer 21 interrupt factor flag (D2, D3) / 16-bit timer 2–3 interrupt factor flag register (0x40283)
F16TU30, F16TC30: Timer 30 interrupt factor flag (D4, D5) / 16-bit timer 2–3 interrupt factor flag register (0x40283)
F16TU31, F16TC31: Timer 31 interrupt factor flag (D6, D7) / 16-bit timer 2–3 interrupt factor flag register (0x40283)
F16TU40, F16TC40: Timer 40 interrupt factor flag (D0, D1) / 16-bit timer 4–5 interrupt factor flag register (0x40284)
F16TU41, F16TC41: Timer 41 interrupt factor flag (D2, D3) / 16-bit timer 4–5 interrupt factor flag register (0x40284)
F16TU50, F16TC50: Timer 50 interrupt factor flag (D4, D5) / 16-bit timer 4–5 interrupt factor flag register (0x40284)
F16TU51, F16TC51: Timer 51 interrupt factor flag (D6, D7) / 16-bit timer 4–5 interrupt factor flag register (0x40284)

Indicates the status of 16-bit programmable timer interrupt generation.

When read

Read "1":	Interrupt factor has occurred
Read "0":	No interrupt factor has occurred

When written using the reset-only method (default)

Write "1":	Interrupt factor flag is reset
Write "0":	Invalid

When written using the read/write method

Write "1":	Interrupt flag is set
Write "0":	Interrupt flag is reset

F16TUxx and F16TCxx are the interrupt factor flags corresponding to counter underflow and comparison match interrupts, respectively. The flag is set to "1" when each interrupt factor occurs. In the 16-bit mode, only the flag for timer x1 is set.
At this time, if the following conditions are met, an interrupt to the CPU is generated:
1. The corresponding interrupt enable register bit is set to "1".
2. No other interrupt request of a higher priority has been generated.
3. The PSR's IE bit is set to "1" (interrupts enabled).
4. The value set in the corresponding interrupt priority register is higher than the CPU's interrupt level (IL).
When using the interrupt factor of the 8-bit programmable timer to request IDMA, note that even when the above conditions are met, no interrupt request to the CPU is generated for the interrupt factor that has occurred. If interrupts are enabled at the setting of IDMA, an interrupt is generated under the above conditions after the data transfer by IDMA is completed.

The interrupt factor flag is set to "1" whenever interrupt generation conditions are met, regardless of how the interrupt enable and interrupt priority registers are set.
If the next interrupt is to be accepted after an interrupt has occurred, it is necessary that the interrupt factor flag be reset, and that the PSR be set again (by setting the IE bit to "1" after setting the IL to a value lower than the level indicated by the interrupt priority register, or by executing the reti instruction).
The interrupt factor flag can be reset only by writing to it in the software. Note that if the PSR is set again to accept interrupts generated (or if the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt occurs again. Note also that the value to be written to reset the flag is "1" when the reset-only method (RSTONLY = "1") is used, and "0" when the read/write method (RSTONLY = "0") is used.
At initial reset, all these flags become indeterminate, so be sure to reset them in the software.

R16TU01, R16TC01: Timer 01 IDMA request (D6, D7) /
K6, DMA, 16-bit timer 0 IDMA request register (0x40290)
R16TU11, R16TC11: Timer 11 IDMA request (D0, D1) / 16-bit timer 1–4 IDMA request register (0x40291)
R16TU21, R16TC21: Timer 21 IDMA request (D2, D3) / 16-bit timer 1–4 IDMA request register (0x40291)
R16TU31, R16TC31: Timer 31 IDMA request (D4, D5) / 16-bit timer 1–4 IDMA request register (0x40291)
R16TU41, R16TC41: Timer 41 IDMA request (D6, D7) / 16-bit timer 1–4 IDMA request register (0x40291)
R16TU51, R16TC51: Timer 51 IDMA request (D0, D1) /
16-bit timer 5, 8-bit timer, serial I/F Ch.0 IDMA request register (0x40292)

Specifies whether to invoke IDMA when an interrupt factor occurs.

Write "1":	IDMA request
Write "0":	Interrupt request
Read:	Valid

R16TUx1 and R16TCx1 are IDMA request bits corresponding to timer x1's counter underflow and comparison match interrupt factors, respectively. When the bit is set to "1", IDMA is invoked when an interrupt factor occurs, thereby performing programmed data transfers. When the register is set to "0", normal interrupt processing is performed and IDMA is not invoked. For details on IDMA, refer to Section 9.2, "Intelligent DMA".
At initial reset, these bits become indeterminate, so be sure to initialize them in the software.

11.4.9 Programming Notes

(1) The 16-bit programmable timers clocked by the internal clock operate only when the prescaler is operating. The prescaler generates a clock for each timer from the OSC3 oscillation clock by dividing it as set using the register. When the CPU is operating using the low-speed (OSC1) clock, the prescaler is inactive, so the 16-bit programmable timer also cannot be used.

(2) The data set in the prescaler is the division ratio for the OSC3 oscillation frequency. If the CPU operating clock is generated from the OSC3 oscillation clock by dividing it using CLKDT[1:0] (D[7:6]) / Power control register (0x40140), do not use a clock that is faster than the CPU operating clock.

(3) When using the timer as an event counter, make sure the event cycle is at least twice the CPU operating-clock period.

(4) When setting the input clock or operation mode, make sure the 16-bit programmable timer is turned off.

(5) Since the interrupt signal and the timer output status are undefined after an initial reset, the counter initial value should be set to the 16-bit timer before resetting the interrupt factor flag or turning the timer output on. However, the timer output is fixed at "0" when the function is turned off.

(6) When the comparison data register value is "0" (CRxx = "0") or larger than the reload data register value (RRxx < CRxx), the comparison match interrupt will not be generated. In this case, the timer output is left as "0".

(7) Because the function select, I/O control, and pull-up control registers used to set the input/output pins of the 16-bit programmable timer are write-only registers (read data is indeterminate), bit operation instructions (bset, bclr, and bnot) cannot be used to rewrite these registers. Use ordinary storage instructions for this purpose.

(8) After an initial reset, the interrupt factor flag and IDMA request register become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset this flag and register in the software.

(9) To prevent another interrupt from being generated by the same factor after an interrupt has occurred, be sure to reset the interrupt factor flag before setting the PSR again or executing the reti instruction.

12 Serial Interface

The E0C33A104 contains two channels of serial interfaces. This chapter describes the functions of these serial interfaces and how to control them.

12.1 Configuration of Serial Interfaces

12.1.1 Features of Serial Interfaces

The E0C33A104 contains two channels (Ch.0, Ch.1) of serial interfaces, the features of which are described below. The functions of these two serial interfaces are the same.

- A clock-synchronized or asynchronous mode can be selected for the transfer method.
  Clock-synchronized mode
  Data length: 8 bits, fixed (No start, stop, and parity bits)
  Receive error: An overrun error can been detected.
  Asynchronous mode
  Data length: 7 or 8 bits, selectable
  Receive error: Overrun, framing, or parity errors can been detected.
  Start bit: 1 bit, fixed
  Stop bit: 1 or 2 bits, selectable
  Parity bit: Even, odd, or none; selectable
  Since the transmit and receive units are independent, full-duplex communication is possible.
- Baud-rate setting: Any desired baud rate can be set by selecting the prescaler's division ratio, setting the 8-bit programmable timer, or using external clock input (asynchronous mode only).
- The receive and transmit units are constructed with a double-buffer structure, allowing for successive receive and transmit operations.
- Data transfers using IDMA are possible.
- Three types of interrupts (transmit data empty, receive data full, and receive error) can be generated.

Figure 12.1.1 shows the configuration of the serial interface (one channel).

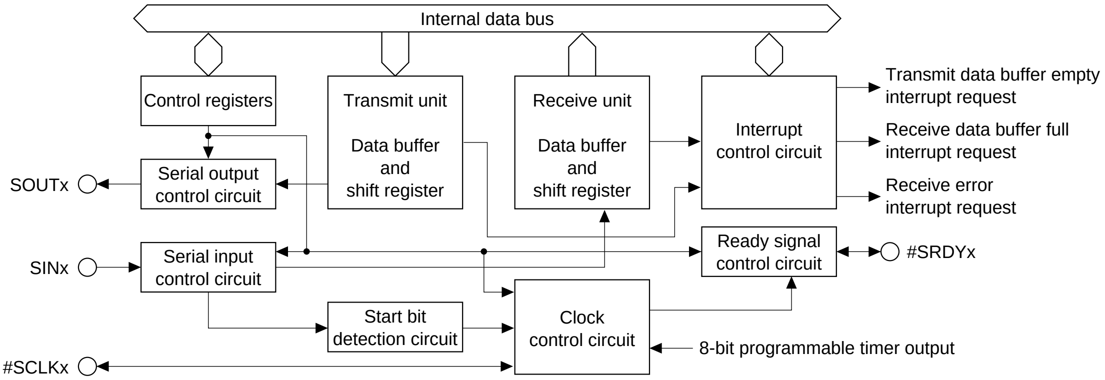

Internal data bus
Control registers
Transmit unit
Data buffer and shift register
Receive unit
Data buffer and shift register
Interrupt control circuit
Transmit data buffer empty interrupt request
Receive data buffer full interrupt request
Receive error interrupt request
SOUTx
Serial output control circuit
SINx
Serial input control circuit
Ready signal control circuit
#SRDYx
Start bit detection circuit
Clock control circuit
8-bit programmable timer output
#SCLKx

Figure 12.1.1 Configuration of Serial Interface

Note: Ch.0 and Ch.1 have the same configuration and the same function. The signal and control bit names are suffixed by a 0 or a 1 to indicate the channel number, enabling discrimination between channels 0 and 1. In this manual, however, channel numbers 0 and 1 are replaced with "x" unless discrimination is necessary, because explanations are common to both channels.

12.1.2 I/O Pins of Serial Interface

Table 12.1.1 lists the I/O pins used by the serial interface.

Table 12.1.1 Serial-Interface Pin Configuration

Pin name	Pin No.		I/O	Pull-up	Function	Function select bit
	QFP5-128	QFP15-128				
P00/SIN0	60	57	I/O	Built-in	I/O port / Serial IF Ch.0 data input	CFP00(D0)/P0 function select register(0x402D0)
P01/SOUT0	63	60	I/O	Built-in	I/O port / Serial IF Ch.0 data output	CFP01(D1)/P0 function select register(0x402D0)
P02/#SCLK0	64	61	I/O	Built-in	I/O port / Serial IF Ch.0 clock input/output	CFP02(D2)/P0 function select register(0x402D0)
P03/#SRDY0	65	62	I/O	Built-in	I/O port / Serial IF Ch.0 ready signal input/output	CFP03(D3)/P0 function select register(0x402D0)
P04/SIN1	66	63	I/O	Built-in	I/O port / Serial IF Ch.1 data input	CFP04(D4)/P0 function select register(0x402D0)
P05/SOUT1	67	64	I/O	Built-in	I/O port / Serial IF Ch.1 data output	CFP05(D5)/P0 function select register(0x402D0)
P06/#SCLK1	68	65	I/O	Built-in	I/O port / Serial IF Ch.1 clock input/output	CFP06(D6)/P0 function select register(0x402D0)
P07/#SRDY1	69	66	I/O	Built-in	I/O port / Serial IF Ch.1 ready signal input/output	CFP07(D7)/P0 function select register(0x402D0)

SINx (serial-data input pin)

This pin is used to input serial data to the device, regardless of the transfer mode.

SOUTx (serial-data output pin)

This pin is used to output serial data from the device, regardless of the transfer mode.

#SCLKx (clock input/output pin)

This pin is used to input or output a clock.

In the clock-synchronized slave mode, it is used as a clock input pin; in the clock-synchronized master mode, it is used as a clock output pin.

In the asynchronous mode, this pin is used as clock input when an external clock is used. This pin is not used when the internal clock is used, so it can be used as an I/O port.

#SRDYx (ready-signal input/output pin)

This pin is used to input or output the ready signal that is used in the clock-synchronized mode.

In the clock-synchronized slave mode, it is used as a ready-signal output pin; in the clock-synchronized master mode, it is used as a ready-signal input pin.

This pin is not used in the asynchronous mode, so it can be used as an I/O port.

Method for setting the serial-interface input/output pins

All of the pins used in the serial interface are shared with I/O ports. At cold start, they are all set for I/O port pins P0x (function select bit CFP0x = "0"). When using the serial interface, write "1" to CFP0x for the pin to be used in accordance with the channel and transfer mode used. Since the function select register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) that accompany a read-modify-write operation cannot be used to rewrite it. Use ordinary storage instructions for this purpose.

At hot start, the pins retain their status from prior to the reset.

Pull-up resistors of input pins

Since the pins used for the serial interface are shared with I/O ports, they contain a pull-up resistor. Whether this pull-up resistor should be used can be specified for each pin through the use of the pull-up control register.

Pull-up control is effective only for the pins used for input, and has no effect on the pins used for output.

Pull-up control for P07–P00: IOU0[7:0] (D[7:0]) / P0 pull-up control register (0x402D3)

At cold start, the pull-up control register is set to "0", with the pull-up resistors disconnected from the input line. If it is necessary for an input pin to be pulled up, write "1" to the IOU0x bit for that pin. Since the pull-up control register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.

At hot start, the pins retain their status from prior to the reset.

12.1.3 Setting Transfer Mode

The transfer mode of the serial interface can be set using SMDx[1:0] individually for each channel as shown in Table 12.1.2 below.

Table 12.1.2 Transfer Mode

SMDx1	SMDx0	Transfer mode
1	1	8-bit asynchronous mode
1	0	7-bit asynchronous mode
0	1	Clock-synchronized slave mode
0	0	Clock-synchronized master mode

At initial reset, SMDx becomes indeterminate, so be sure to initialize it in the software.

When using the IrDA interface, set the transfer mode for the asynchronous 7-bit or asynchronous 8-bit mode.

The input/output pins are configured differently, depending on the transfer mode. The pin configuration in each mode is shown in Table 12.1.3.

Table 12.1.3 Pin Configuration by Transfer Mode

Transfer mode	SINx (P00/P04)	SOUTx (P01/P05)	#SCLKx (P02/P06)	#SRDYx (P03/P07)
8-bit asynchronous	Data input	Data output	Clock input/P port	P port
7-bit asynchronous	Data input	Data output	Clock input/P port	P port
Clock-synchronized slave	Data input	Data output	Clock input	Ready output
Clock-synchronized master	Data input	Data output	Clock output	Ready input

All four pins are used in the clock-synchronized mode.

In the asynchronous mode, since #SRDYx is unused, P03 (or P07) can be used as an I/O (P) port. In addition, when an external clock is not used, P02 (or P06) can also be used as an I/O port.

The I/O control and data registers for the I/O ports used in the serial interface can be used as general-purpose read/write registers.

Note: To enable the IrDA interface to be set, IRMDx[1:0] (D[1:0]) / Serial I/F Ch.0 IrDA register (0x401E4) or Serial I/F Ch.1 IrDA register (0x401E9) is provided. Since these bits become indeterminate at initial reset, be sure to initialize them by writing "00" when using as the normal interface or "10" when using as the IrDA interface.

12.2 Clock-Synchronized Interface

12.2.1 Outline of Clock-Synchronized Interface

In the clock-synchronized transfer mode, 8 bits of data are synchronized to the common clock on both the transmit and receive sides when the data is transferred. Since the transmit and receive units both have a double-buffer structure, successive transmit and receive operations are possible. Since the clock line is shared between the transmit and receive units, the communication mode is half-duplex.

Master and slave modes

Either the clock-synchronized master mode or the clock-synchronized slave mode can be selected using SMDx[1:0].

Clock-synchronized master mode (SMDx[1:0] = "00")

In this mode, clock-synchronized 8-bit serial transfers, in which the serial interface functions as the master, can be performed using the internal clock to synchronize the operation of the internal shift registers.

The synchronizing clock is output from the #SCLKx pin, enabling an external (slave side) serial input/output device to be controlled. The #SRDYx pin is also used to input a signal that indicates whether the external serial input/output device is ready to transmit or receive (when ready in a low level).

Clock-synchronized slave mode (SMDx[1:0] = "01")

In this mode, clock-synchronized 8-bit serial transfers, in which the serial interface functions as a slave, can be performed using the synchronizing clock that is supplied by an external (master side) serial input/output device.

The synchronizing clock is input from the #SCLKx pin for use as the synchronizing clock of the serial interface. In addition, a #SRDYx signal indicating whether the serial interface is ready to transmit or receive (when ready in a low level) is output from the #SRDYx pin.

Figure 12.2.1 shows an example of how the input/output pins are connected in the clock-synchronized mode.

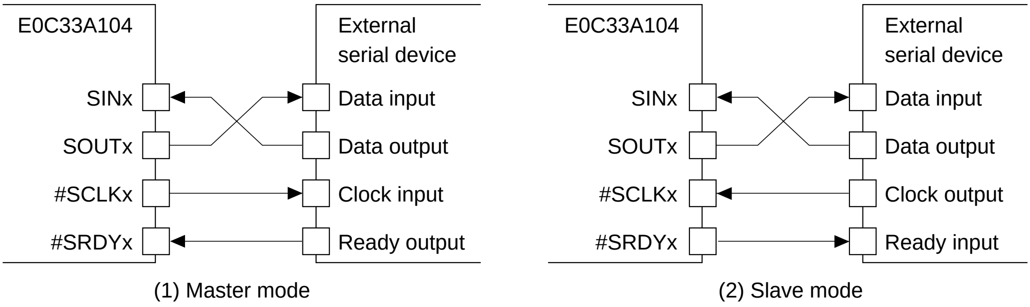


Figure 12.2.1 Example of Connection in Clock-Synchronized Mode

Clock-synchronized transfer data format

In clock-synchronized transfers, the data format is fixed as shown below.

Data length:	8 bits
Start bit:	None
Stop bit:	None
Parity bit:	None

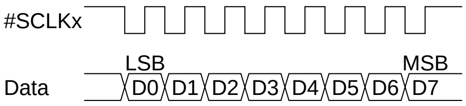


Figure 12.2.2 Clock-Synchronized Transfer Data Format

Serial data is transmitted and received starting with the LSB.

12.2.2 Setting Clock-Synchronized Interface

When performing clock-synchronized transfers via the serial interface, the following settings must be made before data transfer is actually begun:

1. Setting input/output pins
2. Setting the interface mode
3. Setting the transfer mode
4. Setting the input clock
5. Setting interrupts and IDMA

The following explains the content of each setting. For details on interrupt/IDMA settings, refer to Section 12.5, "Serial Interface Interrupts and DMA".

Note: Always make sure the serial interface is inactive (TXENx and RXENx = "0") before these settings are made. A change of settings during operation may cause a malfunction.

Setting input/output pins

All four pins—SINx, SOUTx, #SCLKx, and #SRDYx—are used in the clock-synchronized mode. When using Ch.0, set CFP0[3:0] (D[3:0]) / P0 function select register (0x402D0) to "1111" and when using Ch.1, set CFP0[7:4] (D[7:4]) / P0 function select register (0x402D0) to "1111". (It is possible to use both channels.)

Setting the interface mode

IRMDx[1:0] (D[1:0]) / Serial I/F Ch.0 IrDA register (0x401E4) or Serial I/F Ch.1 IrDA register (0x401E9) is used to set the interface mode (normal or IrDA interface). Write "00" to IRMDx[1:0] to choose the ordinary interface. Since IRMDx[1:0] becomes indeterminate at initial reset, it must be initialized.

Setting the transfer mode

Use SMDx to set the transfer mode of the serial interface as described earlier. When using the serial interface as the master for clock-synchronized transfer, set SMDx[1:0] to "00"; when using the serial interface as a slave, set SMDx[1:0] to "01".

Setting the input clock

- **Clock-synchronized master mode**

This mode operates using an internally derived clock. The clock source for each channel is as follows:
Ch.0: A clock output by 8-bit programmable timer 2
Ch.1: A clock output by 8-bit programmable timer 3

Therefore, in order for the serial interface to be used in the clock-synchronized master mode, the following conditions must be met:

1. The CPU is operating using the high-speed (OSC3) oscillation clock or a clock derived from the OSC3 clock by dividing it.
2. The prescaler is feeding a clock to 8-bit programmable timer 2 (3).
3. The 8-bit programmable timer 2 (3) is generating a clock.

Any desired clock frequency can be selected by setting the division ratio of the prescaler and the reload data of the 8-bit programmable timer as necessary. The relationship between the contents of these settings and the transfer rate is expressed by Eq. 1 below.
To ensure that the duty ratio of the clock to be fed to the serial interface is 50%, the 8-bit programmable timer further divides the underflow signal frequency by 2 internally. This 1/2 frequency division is factored into Eq. 1.

$$\mathrm{RLD} = \frac{f_{OSC3} \times dr}{2 \times bps} - 1 \qquad \text{(Eq. 1)}$$

RLD: Reload data register setup value of the 8-bit programmable timer
f_{OSC3}: OSC3 oscillation frequency (Hz)
bps: Transfer rate (bits/second)
dr: Division ratio of the prescaler

Note: The division ratios selected by the prescaler differ between 8-bit programmable timers 2 and 3, so be careful when setting the ratio.
8-bit programmable timer 2: 1/2, 1/4, 1/8, 1/64, 1/2048, 1/4096
8-bit programmable timer 3: 1/2, 1/4, 1/8, 1/64, 1/128, 1/256

For details on how to control the prescaler and 8-bit programmable timers, refer to Section 10.2, "Prescaler and Operating Clock for Peripheral Circuits", and Section 11.3, "8-Bit Programmable Timers".

The serial-interface control register contains an SSCKx bit to select the clock source used for the asynchronous mode. Although this bit does not affect the clock in the clock-synchronized mode, its content becomes indeterminate at initial reset. Therefore, be sure to initialize this bit by writing "0" (OSC3), even when using the serial interface in the clock-synchronized master mode.

- **Clock-synchronized slave mode**

This mode operates using the clock that is output by the external master. This clock is input from the #SCLK pin.
Therefore, there is no need to control the prescaler or 8-bit programmable timer.

Initialize SSCKx by writing "1" (#SCLKx).

12.2.3 Control and Operation of Clock-Synchronized Transfer

Transmit control

(1) Enabling transmit operation

Use the transmit-enable bit TXENx for transmit control.
Ch.0 transmit-enable: TXEN0 (D7) / Serial I/F Ch.0 control register (0x401E3)
Ch.1 transmit-enable: TXEN1 (D7) / Serial I/F Ch.1 control register (0x401E8)
When transmit is enabled by writing "1" to this bit, the clock input to the shift register is enabled (ready for input), thus allowing for data to be transmitted. The synchronizing clock input/output of the #SCLKx pin is also enabled (ready for input/output).
Transmit is disabled by writing "0" to TXENx.

Note: In clock-synchronized transfers, the clock line is shared between the transmit and receive units, so the communication mode is half-duplex. Therefore, TXENx and receive-enable bit RXENx cannot be enabled simultaneously. When transmitting data, fix RXENx at "0" and do not change it during a transmit operation.
In addition, make sure TXENx is not set to "0" during a transmit operation.

(2) Transmit procedure

The serial interface contains a transmit shift register and a transmit data register (transmit data buffer), which are provided independently of those used for a receive operation.
Ch.0 transmit data: TXD0[7:0] (D[7:0]) / Serial I/F Ch.0 transmit data register (0x401E0)
Ch.1 transmit data: TXD1[7:0] (D[7:0]) / Serial I/F Ch.1 transmit data register (0x401E5)
The serial interface contains a status bit to indicate the status of the transmit data register.
Ch.0 transmit data buffer empty: TDBE0(D1) / Serial I/F Ch.0 status register (0x401E2)
Ch.1 transmit data buffer empty: TDBE1(D1) / Serial I/F Ch.1 status register (0x401E7)
This bit is reset to "0" by writing data to the transmit-data register, and set to "1" again (buffer empty) when the data is transferred to the shift register.
The serial interface starts transmitting when data is written to the transmit data register.
After the P0 function select register is set for the serial interface, the I/O direction of the #SRDY and #SCLK pins are changed at follows:
#SRDY: When transmission is enabled in slave mode, P03 (P07) enters output mode.
Otherwise, P03 (P07) stays in input mode.
#SCLK: When transmission is enabled in master mode, P02 (P06) enters output mode.
Otherwise, P02 (P06) stays in input mode.

Following explains transmit operation in both the master and slave modes.

- Clock-synchronized master mode

The timing at which the device starts transmitting in the master mode is as follows:
When #SRDY is on a low level while TDBEx = "0" (the transmit-data register contains data written to it) or when TDBEx is set to "0" (data has been written to the transmit-data register) while #SRDY is on a low level.
Figure 12.2.3 shows a transmit timing chart in the clock-synchronized master mode.

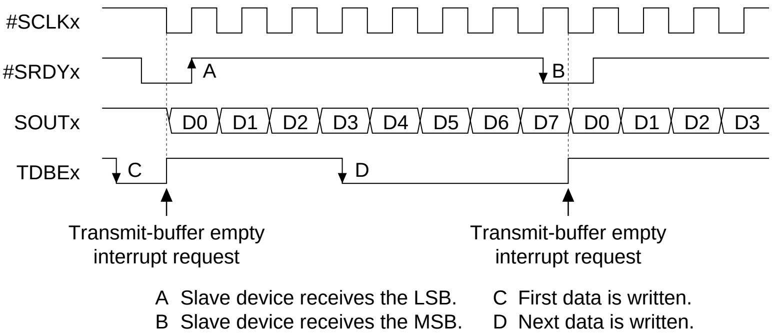

#SCLKx
#SRDYx
A
B
SOUTx
D0
D1
D2
D3
D4
D5
D6
D7
D0
D1
D2
D3
TDBEx
C
D
Transmit-buffer empty interrupt request
Transmit-buffer empty interrupt request
A Slave device receives the LSB.
B Slave device receives the MSB.
C First data is written.
D Next data is written.

Figure 12.2.3 Transmit Timing Chart in Clock-Synchronized Master Mode

1. If the #SRDYx signal from the slave is on a high level, the master waits until it is on a low level (ready to receive).
2. If #SRDYx is on a low level, the synchronizing clock input to the serial interface begins. The synchronizing clock is also output from the #SCLKx pin to the slave device.
3. The content of the data register is transferred to the shift register synchronously with the first falling edge of the clock. At the same time, the LSB of the data transferred to the shift register is output from the SOUTx pin.
4. The data in the shift register is shifted 1 bit by the next falling edge of the clock, and the bit following the LSB is output from SOUTx. This operation is repeated until all 8 bits of data are transmitted.

The slave device must take in each bit synchronously with the rising edges of the synchronizing clock.

- Clock-synchronized slave mode

Figure 12.2.4 shows a transmit timing chart in the clock-synchronized slave mode.

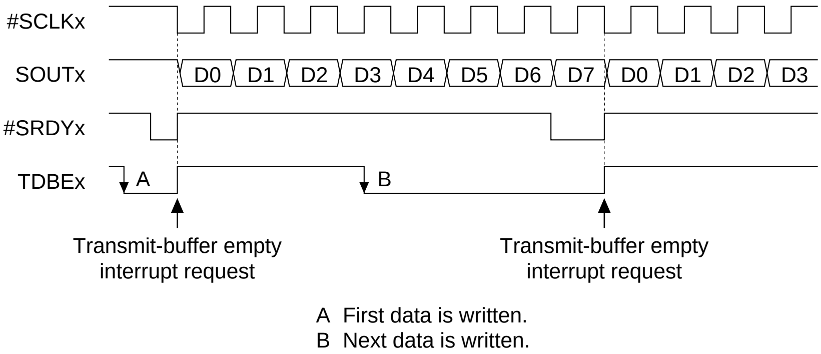


Figure 12.2.4 Transmit Timing Chart in Clock-Synchronized Slave Mode

1. After setting the #SRDYx signal to a low level (ready to transmit), the slave waits for clock input from the master.
2. When the synchronizing clock is input from the #SCLKx pin, the content of the data register is transferred to the shift register synchronously with the first falling edge of the clock. At the same time, the LSB of the data transferred to the shift register is output from the SOUTx pin.
   The #SRDYx signal is returned to a high level at this point.
3. The data in the shift register is shifted 1 bit by the next falling edge of the clock, and the bit following the LSB is output from SOUTx. This operation is repeated until all 8 bits of data are transmitted.
4. The #SRDYx signal is set to a low level when the last bit (8th bit) is output from the SOUTx pin.

The master device must take in each bit synchronously with the rising edges of the synchronizing clock.

- Successive transmit operations

When the data in the transmit data register is transferred to the shift register, TDBEx is reset to "1" (buffer empty). Once this occurs, the next transmit data can be written to the transmit data register, even during data transmission.

This allows data to be transmitted successively. The transmit procedure is described above.

When TDBEx is set to "1", a transmit-data empty interrupt factor occurs. Since an interrupt can be generated as set by the interrupt controller, the next piece of transmit data can be written using an interrupt processing routine. In addition, since this interrupt factor can be used to invoke IDMA, the data prepared in memory can be transmitted successively to the transmit-data register through DMA transfers.

For details on how to control interrupts and IDMA requests, refer to Section 12.5, "Serial Interface Interrupts and DMA".

(3) Terminating transmit operation

Upon completion of data transmission, write "0" to the transmit-enable bit TXENx to disable transmit operation.

Receive control

(1) Enabling receive operation

Use the receive-enable bit RXENx for receive control.
Ch.0 receive-enable: RXEN0 (D6) / Serial I/F Ch.0 receive-enable register (0x401E3)
Ch.1 receive-enable: RXEN1 (D6) / Serial I/F Ch.1 receive-enable register (0x401E8)
When receive operations are enabled by writing "1" to this bit, clock input to the shift register is enabled (ready for input), thereby starting a data-receive operation. The synchronizing clock input/output on the #SCLKx pin also is enabled (ready for input/output). Receive operations are disabled by writing "0" to RXENx.

Note: In clock-synchronized transfers, the clock line is shared between the transmit and receive units, so the communication mode is half-duplex. Therefore, RXENx and transmit-enable bit TXENx cannot be enabled simultaneously. When receiving data, fix TXENx at "0" and do not change it during a receive operation. In addition, make sure RXENx is not set to "0" during a receive operation.

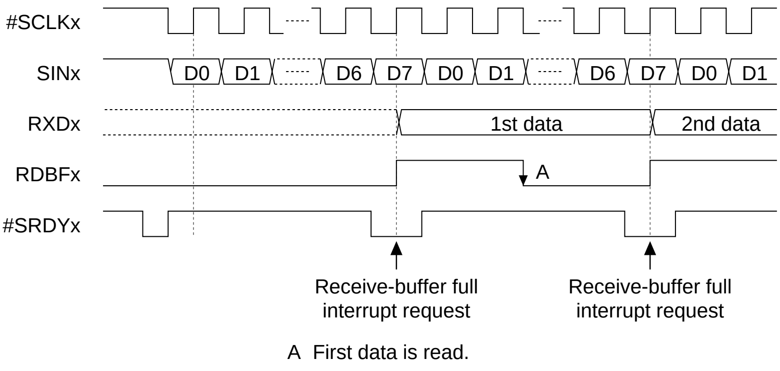
(2) Receive procedure

This serial interface has a receive shift register and a receive data register (receive data buffer) that are provided independently of those used for transmit operations.
Ch.0 receive data: RXD0[7:0] (D[7:0]) / Serial I/F Ch.0 receive data register (0x401E1)
Ch.1 receive data: RXD1[7:0] (D[7:0]) / Serial I/F Ch.1 receive data register (0x401E6)
The receive data can be read out from this register.
A status bit is also provided that indicates the status of the receive data register.
Ch.0 receive data buffer full: RDBF0 (D0) / Serial I/F Ch.0 status register (0x401E2)
Ch.1 receive data buffer full: RDBF1 (D0) / Serial I/F Ch.1 status register (0x401E7)
This bit is set to "1" (buffer full) when the MSB of serial data is received and the data in the shift register is transferred to the receive data register, indicating that the received data can be read out. When the data is read out, the bit is reset to "0".
After the P0 function select register is set for the serial interface, the I/O direction of the #SRDY and #SCLK pins are changed at follows:
#SRDY: When receive operation is enabled in slave mode, P03 (P07) enters output mode.
Otherwise, P03 (P07) stays in input mode.
#SCLK: When receive operation is enabled in master mode, P02 (P06) enters output mode.
Otherwise, P02 (P06) stays in input mode.

The following describes a receive operation in the master and slave modes.

- Clock-synchronized master mode

Figure 12.2.5 shows a receive timing chart in the clock-synchronized master mode.


#SCLKx
SINx
D0
D1
D6
D7
D0
D1
D6
D7
D0
D1
RXDx
1st data
2nd data
RDBFx
A
#SRDYx
Receive-buffer full interrupt request
Receive-buffer full interrupt request

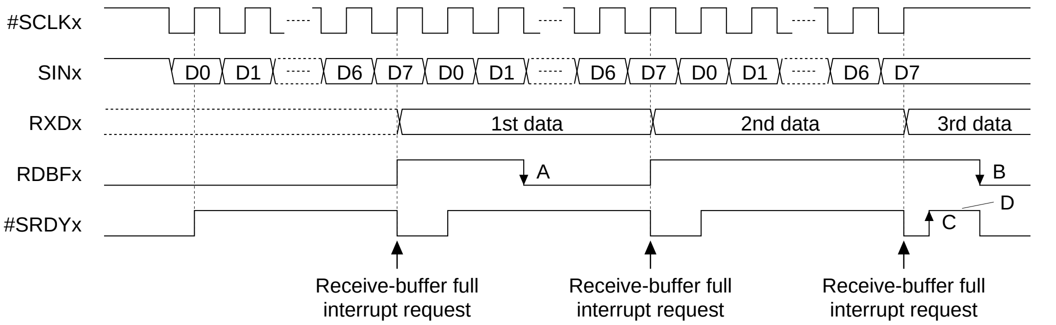
A First data is read.

Figure 12.2.5 Receive Timing Chart in Clock-Synchronized Master Mode

1. If the #SRDYx signal from the slave is on a high level, the master waits until it turns to a low level (ready to receive).
2. If #SRDYx is on a low level, synchronizing clock input to the serial interface begins. The synchronizing clock is also output from the #SCLKx pin to the slave device.
3. The slave device outputs each bit of data synchronously with the falling edges of the clock. The LSB is output first.
4. This serial interface takes the SIN input into the shift register at the rising edges of the clock. The data in the shift register is sequentially shifted as bits are taken in. This operation is repeated until the MSB of data is received.
5. When the MSB is taken in, the data in the shift register is transferred to the receive data register, enabling the data to be read out.

- Clock-synchronized slave mode

Figure 12.2.6 shows a receive timing chart in the clock-synchronized slave mode.


#SCLKx
SINx
D0
D1
D6
D7
D0
D1
D6
D7
D0
D1
D6
D7
RXDx
1st data
2nd data
3rd data
RDBFx
A
B
#SRDYx
C
D
Receive-buffer full interrupt request
Receive-buffer full interrupt request
Receive-buffer full interrupt request

A First data is read.
B 3rd data is read.
C An overrun error occurs because the receive operation has completed when RDBFx = "1".
D Send the busy signal to the master device to stop the clock.

Figure 12.2.6 Receive Timing Chart in Clock-Synchronized Slave Mode

1. After setting the #SRDYx signal to a low level (ready to receive), the slave waits for clock input from the master.
2. The master device outputs each bit of data synchronously with the falling edges of the clock. The LSB is output first.
3. This serial interface takes the SIN input into the shift register at the rising edges of the clock that is input from #SCLKx. The data in the shift register is sequentially shifted as bits are taken in. This operation is repeated until the MSB of data is received.
4. When the MSB is taken in, the data in the shift register is transferred to the receive data register, enabling the data to be read out.

- Successive receive operations

When the data received in the shift register is transferred to the receive data register, RDBFx is set to "1" (buffer full), indicating that the received data can be read out.

Since the receive data register can be read out while receiving the next data, data can be received successively. The procedure for receiving is described above.

When RDBFx is set to "1", a receive-data full interrupt factor occurs. Since an interrupt can be generated as set by the interrupt controller, the received data can be read by an interrupt processing routine. In addition, since this interrupt factor can be used to invoke IDMA, the received data can be received successively in locations prepared in memory through DMA transfers.

For details on how to control interrupts/IDMA, refer to Section 12.5, "Serial Interface Interrupts and DMA".

(3) Overrun error

If, during successive receive operation, a receive operation for the next data is completed before the receive data register is read out, the receive data register is overwritten with the new data. Therefore, the receive data register must always be read out before a receive operation for the next data is completed.

When the receive data register is overwritten, an overrun error is generated and the overrun error flag is set to "1".

Ch.0 overrun error flag: OER0 (D2) / Serial I/F Ch.0 status register (0x401E2)
Ch.1 overrun error flag: OER1 (D2) / Serial I/F Ch.1 status register (0x401E7)

Once the overrun error flag is set to "1", it remains set until it is reset by writing "0" to it in the software.

The overrun error is one of the receive-error interrupt factors in the serial interface. An interrupt can be generated for this error by setting the interrupt controller as necessary, so that the error can be processed by an interrupt processing routine.

(4) #SRDYx in slave mode

When receive operations are enabled by writing "1" to RXENx, the #SRDYx signal is turned to a low level, thereby indicating to the master device that the slave is ready to receive. When the LSB of serial data is received, #SRDYx is turned to a high level; when the MSB is received, #SRDYx is returned to a low level, in preparation for the next receive operation.

If an overrun error occurs, #SRDYx is turned to a high level (unable to receive) at that point, with receive operations for the following data thus suspended. In this case, #SRDYx is returned to a low by reading out the data overwritten in the receive data register, and if any receive data follows, the slave restarts receiving data.

(5) Terminating receive operation

Upon completion of a data receive operation, write "0" to the receive-enable bit RXENx to disable receive operations.

12.3 Asynchronous Interface

12.3.1 Outline of Asynchronous Interface

Asynchronous transfers are performed by adding a start bit and a stop bit to the start and end points of each serial-converted data. With this method, there is no need to use a clock that is fully synchronized on the transmit and receive sides; instead, transfer operations are timed by the start and stop bits added to the start and end points of each data.

In the 8-bit asynchronous mode (SMDx[1:0] = "11"), 8 bits of data can be transferred; in the 7-bit asynchronous mode (SMDx[1:0] = "10"), 7 bits of data can be transferred.

In either mode, it is possible to select the stop-bit length, add a parity bit, and choose between even and odd parity. The start bit is fixed at "1".

The operating clock can be selected between an internal clock generated by an 8-bit programmable timer or an external clock that is input from the #SCLKx pin.

Since the transmit and receive units are both constructed with a double-buffer structure, successive transmit and receive operations are possible. Furthermore, since the transmit and receive units are independent, full-duplex communication in which transmit and receive operations are performed simultaneously is also possible.

Figure 12.3.1 shows an example of how input/output pins are connected for transfers in the asynchronous mode.

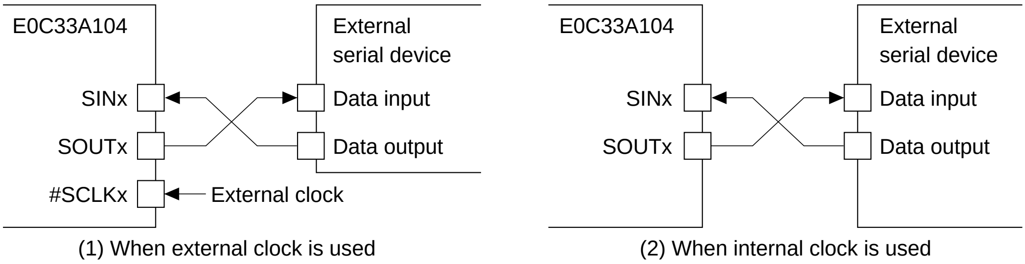


Figure 12.3.1 Example of Connection in Asynchronous Mode

When the asynchronous mode is selected, it is possible to use the IrDA interface function.

Asynchronous-transfer data format

The data format for asynchronous transfer is shown below.

Data length:	7 or 8 bits (determined by the selected transfer mode)
Start bit:	1 bit, fixed
Stop bit:	1 or 2 bits
Parity bit:	Even or odd parity, or none

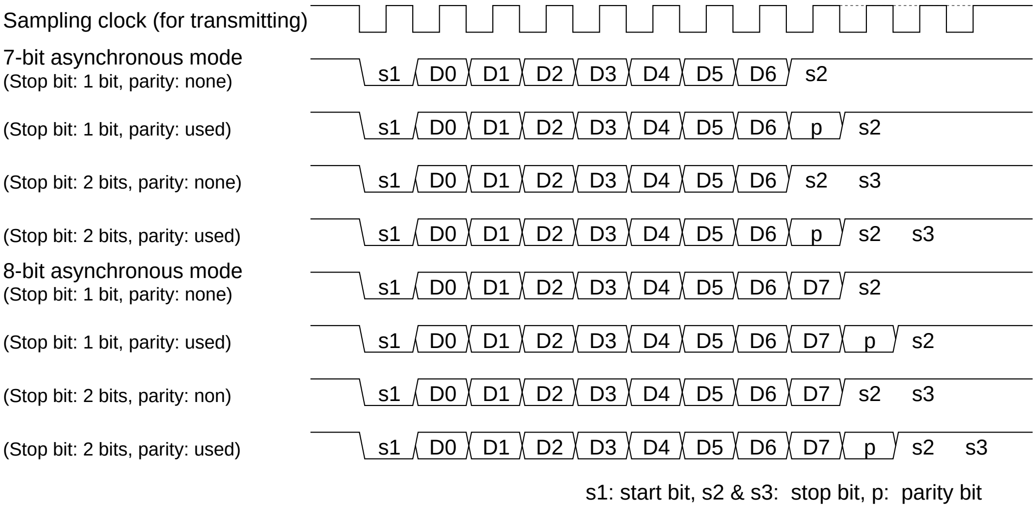

Sampling clock (for transmitting)
7-bit asynchronous mode
(Stop bit: 1 bit, parity: none)
s1 D0 D1 D2 D3 D4 D5 D6 s2
(Stop bit: 1 bit, parity: used)
s1 D0 D1 D2 D3 D4 D5 D6 p s2
(Stop bit: 2 bits, parity: none)
s1 D0 D1 D2 D3 D4 D5 D6 s2 s3
(Stop bit: 2 bits, parity: used)
s1 D0 D1 D2 D3 D4 D5 D6 p s2 s3
8-bit asynchronous mode
(Stop bit: 1 bit, parity: none)
s1 D0 D1 D2 D3 D4 D5 D6 D7 s2
(Stop bit: 1 bit, parity: used)
s1 D0 D1 D2 D3 D4 D5 D6 D7 p s2
(Stop bit: 2 bits, parity: non)
s1 D0 D1 D2 D3 D4 D5 D6 D7 s2 s3
(Stop bit: 2 bits, parity: used)
s1 D0 D1 D2 D3 D4 D5 D6 D7 p s2 s3
s1: start bit, s2 & s3: stop bit, p: parity bit

Figure 12.3.2 Data Format for Asynchronous Transfer

Serial data is transmitted and received, starting with the LSB.

12.3.2 Setting Asynchronous Interface

When performing asynchronous transfer via the serial interface, the following must be done before data transfer can be started:

1. Setting input/output pins
2. Setting the interface mode
3. Setting the transfer mode
4. Setting the input clock
5. Setting the data format
6. Setting interrupt/IDMA

The following describes how to set each of the above. For details on interrupt/IDMA settings, refer to Section 12.5, "Serial Interface Interrupts and DMA".

Note: Always make sure the serial interface is inactive (TXENx and RXENx = "0") before making these settings. A change in settings during operation may result in a malfunction.

Setting input/output pins

In the asynchronous mode, two pins–SINx and SOUTx–are used. When external clock input is used, one more pin, #SCLKx, is also used.
Set CFP0[7:0] (D[7:0]) / P0 function select register (0x402D0) according to the pins used. (Both channels can be used, if necessary.) Since the #SRDYx pin is not used, P03 or P07 can be used as an I/O port. During operation using the internal clock, P03 or P06 can also be used as an I/O port.

Setting the interface mode

IRMDx[1:0] (D[1:0]) / Serial I/F Ch.0 IrDA register (0x401E4) is used to set the IrDA interface. Since IRMDx[1:0] becomes indeterminate at initial reset, initialize it by writing "00" when using the serial interface as a normal interface, or "10" when using the serial interface as an IrDA interface. This setting must be made before a transfer mode is set.

Setting the transfer mode

Use SMDx to set the transfer mode of the serial interface as described earlier. When using the serial interface in the 8-bit asynchronous mode, set SMDx[1:0] to "11", when using the serial interface in the 7-bit asynchronous mode, set SMDx[1:0] to "10".

Setting the input clock

In the asynchronous mode, the operating clock can be selected between the internal clock and an external clock.
Ch.0 input clock selection: SSCK0 (D2) / Serial I/F Ch.0 control register (0x401E3)
Ch.1 input clock selection: SSCK1 (D2) / Serial I/F Ch.1 control register (0x401E8)
The external clock is selected (input from the #SCLKx pin) by writing "1" to SSCKx, and an internal clock is selected by writing "0".

Note: SSCKx becomes indeterminate at initial reset, so be sure to reset it in the software.

- **Internal clock**

  When the internal clock is selected, the serial interface is clocked by a clock generated using an 8-bit programmable timer. The clock source for each channel is as follows:
  Ch.0: Clock output by 8-bit programmable timer 2
  Ch.1: Clock output by 8-bit programmable timer 3

  Therefore, before the internal clock can be used, the following conditions must be met:
  1. The CPU is operating using the high-speed (OSC3) oscillation clock or a clock derived from it.
  2. The prescaler is outputting a clock to the 8-bit programmable timer 2 (or 3).
  3. The 8-bit programmable timer 2 (or 3) is outputting a clock.

Any desired clock frequency can be obtained by setting the prescaler division ratio and the reload data of the 8-bit programmable timer as necessary. The relationship between the contents of these setting and the transfer rate is expressed by Eq. 2.

The 8-bit programmable timer has its underflow signal further divided by 2 internally, in order to ensure that the duty ratio of the clock supplied to the serial interface is 50%.

Furthermore, the clock output by the 8-bit programmable timer is divided by 16 or 8 internally in the serial interface, in order to create a sampling clock (refer to "Sampling clock"). This division ratio must also be considered when setting the transfer rate.

These division ratios are taken into account in Eq. 2.

$$\text{RLD} = \frac{f_{\text{OSC3}} \times \text{pdr} \times \text{sdr}}{2 \times \text{bps}} - 1 \qquad \text{(Eq. 2)}$$

RLD: Set value of the 8-bit programmable timer's reload data register
f_{OSC3}: OSC3 oscillation frequency (Hz)
bps: Transfer rate (bits/second)
pdr: Division ratio of the prescaler
sdr: Internal division ratio of the serial interface (1/16 or 1/8)

Note: The division ratio selected using the prescaler differs between 8-bit programmable timers 2 and 3. Take this into account when setting a division ratio.
8-bit programmable timer 2: 1/2, 1/4, 1/8, 1/32, 1/64, 1/2048, 1/4096
8-bit programmable timer 3: 1/2, 1/4, 1/8, 1/32, 1/64, 1/128, 1/256

Table 12.3.1 shows examples of prescaler division ratios and the reload data settings of the programmable timer, in cases in which the internal division ratio of the serial interface is set to 1/16.

Table 12.3.1 Example of Transfer Rate Settings

Transfer rate	f_{OSC3} = 20 MHz			f_{OSC3} = 25 MHz			f_{OSC3} = 33 MHz		
(bps)	RLD	dr	Error (%)	RLD	dr	Error (%)	RLD	dr	Error (%)
300	129	1/16	0.16025	162	1/16	-0.14698	216	1/16	0.00640
1200	129	1/4	0.16025	162	1/4	-0.14698	216	1/4	0.00640
2400	129	1/2	0.16025	162	1/2	-0.14698	216	1/2	0.00640
4800	64	1/2	0.16025	80	1/2	-0.46939	108	1/2	-0.45234
9600	32	1/2	-1.35732	40	1/2	-0.75584	53	1/2	0.46939
14400	21	1/2	-1.35732	13	1/4	-3.11880	35	1/2	0.46939
28800	10	1/2	-1.35732	13	1/2	-3.11880	17	1/2	0.46939

Make sure the error is within 1%. Calculate the error using the following equation:

$$\text{Error} = \left\{ \frac{f_{\text{OSC3}} \times \text{dr}}{(\text{RLD} + 1) \times 32 \times \text{bps}} - 1 \right\} \times 100\ [\%]$$

For details on how to control the prescaler and 8-bit programmable timers, refer to Section 10.2, "Prescaler and Operating Clock for Peripheral Circuits", and Section 11.3, "8-Bit Programmable Timers".

- **External clock**

When an external clock is selected, the serial interface is clocked by a clock input from the #SCLKx pin. Therefore, there is no need to control the prescaler and 8-bit programmable timers.

Any desired clock frequency can be set. The clock input from the #SCLKx pin is internally divided by 16 or 8 in the serial interface, in order to create a sampling clock (refer to "Sampling clock"). This division ratio must also be considered when setting the transfer rate.

- **Sampling clock**

In the asynchronous mode, TCLK (the clock output by the 8-bit programmable timer or input from the #SCLKx pin) is internally divided in the serial interface, in order to create a sampling clock.
A 1/16 division ratio is selected by writing "1" to DIVMDx , and a 1/8 ratio is selected by writing "0".
Ch.0 clock division ratio selection: DIVMD0 (D4) / Serial I/F Ch.0 IrDA register (0x401E4)
Ch.1 clock division ratio selection: DIVMD1 (D4) / Serial I/F Ch.1 IrDA register (0x401E9)

Note: The DIVMDx bit becomes indeterminate at initial reset, so be sure to reset it in the software. Settings of this bit are valid only in the asynchronous mode (and when using the IrDA interface).

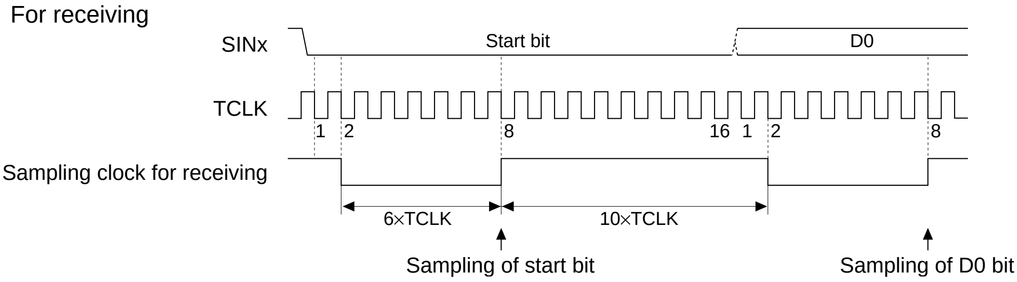


Figure 12.3.3 Sampling Clock for Asynchronous Receive Operation (when 1/16 division is selected)

As shown in Figure 12.3.3, the sampling clock is created by dividing TCLK by 16 (or 8). Its duty ratio (low: high ratio) is 6:10 (or 2:6 when divided by 8), and not 50%. Since the receive data is sampled in the middle point of each bit, the sampling clock recognizes the start bit first, and then changes the level from high to low at the second falling edge of TCLK. And at the 8th (4th for 1/8) falling edge of TCLK, it changes the level from low to high. This change in levels is repeated for the following bits of data:
Each bit of data is sampled at each rising edge of this sampling clock. When the stop bit is sampled, the sampling clock is fixed at high level until the next start bit is sampled.
If the SINx pin is returned to high level at the second falling edge of TCLK when it recognize the start bit, the data is assumed to be noise, and generation of the sampling clock is stopped.
If the SINx pin is not on a low level when the start bit is sampled at the 8th (4th for 1/8) clock, such as when the baud rate is not matched between the transmit and receive units, the serial interface stops sampling the following data and returns to a start-bit detection mode. In this case, no error is generated.

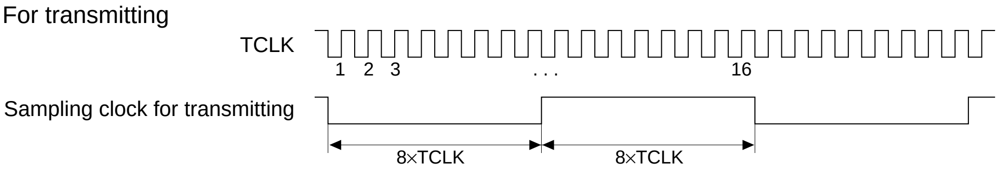


Figure 12.3.4 Sampling Clock for Asynchronous Transmit Operation (when 1/16 division is selected)

When transmitting data, a sampling clock of a 50% duty cycle is generated from TCLK by dividing it by 16 (or 8), and each bit of data is output synchronously with this clock.

Setting the data format

In the asynchronous mode, the data length is 7 or 8 bits as determined by the transfer mode set. The start bit is fixed at 1.

The stop and parity bits can be set as shown in the Table 12.3.2 using the following control bits:

Stop-bit selection Ch.0: STPB0 (D3) / Serial I/F Ch.0 control register (0x401E3)
Ch.1: STPB1 (D3) / Serial I/F Ch.1 control register (0x401E8)
Parity enable Ch.0: EPR0 (D5) / Serial I/F Ch.0 control register (0x401E3)
Ch.1: EPR1 (D5) / Serial I/F Ch.1 control register (0x401E8)
Parity-mode selection Ch.0: PMD0 (D4) / Serial I/F Ch.0 control register (0x401E3)
Ch.1: PMD1 (D4) / Serial I/F Ch.1 control register (0x401E8)

Table 12.3.2 Stop Bit and Parity Bit Settings

STPBx	EPRx	PMDx	Stop bit	Parity bit
1	1	1	2 bits	Odd
		0	2 bits	Even
	0	*	2 bits	None
0	1	1	1 bit	Odd
		0	1 bit	Even
	0	*	1 bit	Non

* Setting PMDx is invalid when EPRx = "0".

Note: These bits become indeterminate at initial reset, so be sure to initialize them in the software.

12.3.3 Control and Operation of Asynchronous Transfer

Transmit control

(1) Enabling transmit operation

Use the transmit-enable bit TXENx for transmit control.
Ch.0 transmit-enable: TXEN0 (D7) / Serial I/F Ch.0 control register (0x401E3)
Ch.1 transmit-enable: TXEN1 (D7) / Serial I/F Ch.1 control register (0x401E8)
When transmit is enabled by writing "1" to this bit, the clock input to the shift register is enabled (ready for input), thus allowing data to be transmitted.
Transmit is disabled by writing "0" to TXENx.

Note: Do not set TXENx to "0" during a transmit operation.

(2) Transmit procedure

The serial interface has a transmit shift register and a transmit data register (transmit data buffer) that are provided independently of those used for receive operations.
Ch.0 transmit data: TXD0[7:0] (D[7:0]) / Serial I/F Ch.0 transmit data register (0x401E0)
Ch.1 transmit data: TXD1[7:0] (D[7:0]) / Serial I/F Ch.1 transmit data register (0x401E5)
The serial interface starts a transmit operation by writing data to this register. In the 7-bit asynchronous mode, bit 7 (MSB) in each register is ignored.
The serial interface also contains a status bit to indicate the status of the transmit data register.
Ch.0 transmit data buffer empty: TDBE0 (D1) / Serial I/F Ch.0 status register (0x401E2)
Ch.1 transmit data buffer empty: TDBE1 (D1) / Serial I/F Ch.1 status register (0x401E7)
This bit is reset to "0" by writing data to the transmit data register, and set back to "1" (buffer empty) when the data is transferred to the shift register. The transfer begins when the serial interface starts sending the start bit.

Figure 12.3.5 shows a transmit timing chart in the asynchronous mode.

Example:	Data length	8 bits
	Stop bit	1 bit
	Parity bit	Included

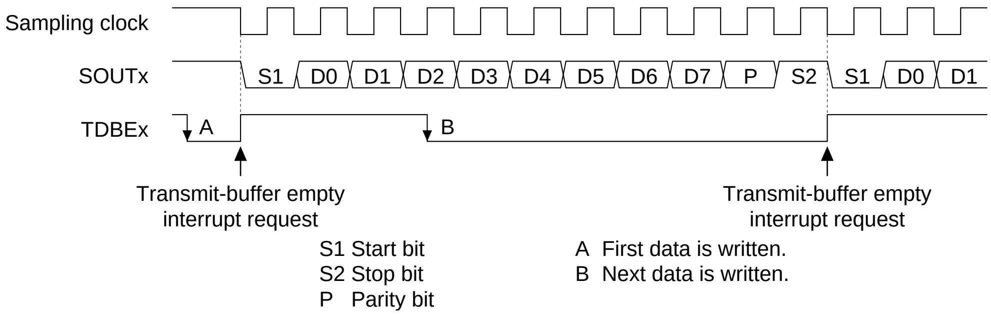


Figure 12.3.5 Transmit Timing Chart in Asynchronous Mode

1. The contents of the data register are transferred to the shift register synchronously with the first falling edge of the sampling clock. At the same time, the SOUTx pin is setting to a low level to send the start bit.
2. Each bit of data in the shift register is transmitted beginning with the LSB at each falling edge of the subsequent sampling clock. This operation is repeated until all 8 (or 7) bits of data are transmitted.
3. After sending the MSB, the parity bit (if EPRx = "1") and the stop bit are transmitted insuccession.

- Successive transmit operation

When the data in the transmit data register is transferred to the shift register, TDBEx is reset to "1" (buffer empty). Once this occurs, the next transmit data can be written to the transmit data register, even during data transmission.

This allows data to be transmitted successively. The transmit procedure is described above.

When TDBEx is set to "1", a transmit-data empty interrupt factor simultaneously occurs. Since an interrupt can be generated as set by the interrupt controller, the next transmit data can be written using an interrupt processing routine. In addition, since this interrupt factor can be used to invoke IDMA, the data prepared in memory can be transmitted successively to the transmit data register through DMA transfers.

For details on how to control interrupts and IDMA requests, refer to Section 12.5, "Serial Interface Interrupts and DMA".

(3) Terminating transmit operations

When data transmission is completed, write "0" to the transmit-enable bit TXENx to disable transmit operations.

Receive control

(1) Enabling receive operations

Use the receive-enable bit RXENx for receive control.

Ch.0 receive-enable: RXEN0 (D6) / Serial I/F Ch.0 control register (0x401E3)

Ch.1 receive-enable: RXEN1 (D6) / Serial I/F Ch.1 control register (0x401E8)

When receiving enabled by writing "1" to this bit, clock input to the shift register is enabled (ready for input), meaning that it is ready to receive data.

Receive operations are disabled by writing "0" to RXENx.

Note: Do not set RXENx to "0" during a receive operation.

(2) Receive procedure

This serial interface has a receive shift register and a receive data register (receive data buffer) that are provided independently of those used for transmit operations.

Ch.0 receive data: RXD0[7:0] (D[7:0]) / Serial I/F Ch.0 receive data register (0x401E1)

Ch.1 receive data: RXD1[7:0] (D[7:0]) / Serial I/F Ch.1 receive data register (0x401E6)

Receive data can be read out from this register.

A status bit is also provided to indicate the status of the receive data register.
Ch.0 receive data buffer full: RDBF0 (D0) / Serial I/F Ch.0 status register (0x401E2)
Ch.1 receive data buffer full: RDBF1 (D0) / Serial I/F Ch.1 status register (0x401E7)
This bit is set to "1" (buffer full) when data is transferred from the shift register to the receive data register after the stop bit is sampled (the second bit if two stop bits are used), indicating that the received data can be read out. When the data is read out, the bit is reset to "0".

Figure 12.3.6 shows a receive timing chart in the asynchronous mode.

Example:	Data length	8 bits
	Stop bit	1 bit
	Parity bit	Included

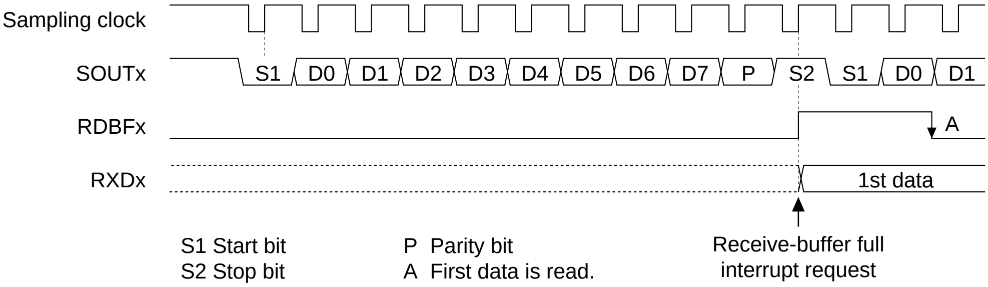


Figure 12.3.6 Receive Timing Chart in Asynchronous Mode

1. The serial interface starts sampling when the start bit is input (SINx = low).
2. When the start bit is sampled at the first rising edge of the sampling clock, each bit of receive data is taken into the shift register, beginning with the LSB at each rising edge of the subsequent clock. This operation is repeated until the MSB of data is received.
3. When the MSB is taken in, the parity bit that follows is also taken in (if EPRx = "1").
4. When the stop bit (or the second bit if two stop bits are used) is sampled, the data in the shift register is transferred to the receive data register, enabling the data to be read out.
   The parity is checked when data is transferred to the receive data register (if EPRx = "1").

- Successive receive operations

When the data received in the shift register is transferred to the receive data register, RDBFx is set to "1" (buffer full), indicating that the received data can be read out. Thereafter, data can be received successively because the receive data register can be read out while the next data is received. The procedure for receiving is described above.
When RDBFx is set to "1", a receive-data full interrupt factor occurs. Since an interrupt can be generated as set by the interrupt controller, the received data can be read using an interrupt processing routine. In addition, since this interrupt factor can be used to invoke IDMA, the received data can be received successively in locations prepared in memory through DMA transfers.
For details on how to control interrupts and IDMA requests, refer to Section 12.5, "Serial Interface Interrupts and DMA".

(3) Receive errors

Three types of receive errors can be detected when receiving data in the asynchronous mode.
Since an interrupt can be generated by setting the interrupt controller, the error can be processed using an interrupt processing routine. For details on receive error interrupts, refer to Section 12.5, "Serial Interface Interrupts and DMA".

- Parity error

If EPRx is set to "1" (parity added), the parity is checked when data is received.
This parity check is performed when the data received in the shift register is transferred to the receive data register in order to check conformity with PMDx settings (odd or even parity). If any nonconformity is found in this check, a parity error is assumed and the parity error flag is set to "1".

Ch.0 parity error flag: PER0 (D3) / Serial I/F Ch.0 status register (0x401E2)
Ch.1 parity error flag: PER1 (D3) / Serial I/F Ch.1 status register (0x401E7)
Even when this error occurs, the received data in error is transferred to the receive data register and the receive operation is continued. However, the content of the received data for which a parity error is flagged cannot be guaranteed.
The PERx flag is reset to "0" by writing "0".

- Framing error

If data with a stop bit = "0" is received, the serial interface assumes that the data is out of synchronization and generates a framing error.
If two stop bits are used, both bits are checked.
When this error occurs, the framing-error flag is set to "1".
Ch.0 framing-error flag: FER0 (D4) / Serial I/F Ch.0 status register (0x401E2)
Ch.1 framing-error flag: FER1 (D4) / Serial I/F Ch.1 status register (0x401E7)
Even when this error occurs, the received data in error is transferred to the receive data register and the receive operation is continued. However, the content of the received data for which a framing error is flagged cannot be guaranteed, even if no framing error is found in the following data received.
The FERx flag is reset to "0" by writing "0".

- Overrun error

If during successive receive operations, a receive operation for the next data is completed before the receive data register is read out, the receive data register is overwritten with the new data. Therefore, the receive data register must always be read out before a receive operation for the next data is completed.
When the receive data register is overwritten, an overrun error is generated and the overrun-error flag is set to "1".
Ch.0 overrun-error flag: OER0 (D2) / Serial I/F Ch.0 status register (0x401E2)
Ch.1 overrun-error flag: OER1 (D2) / Serial I/F Ch.1 status register (0x401E7)
Even when this error occurs, the received data in error is transferred to the receive data register and the receive operation is continued.
The OERx flag is reset to "0" by writing "0".

(4) Terminating receive operation

When a data receive operation is completed, write "0" to the receive-enable bit RXENx to disable receive operations.

12.4 IrDA Interface

12.4.1 Outline of IrDA Interface

Each channel of the serial interface contains a PPM modulator circuit, allowing an infrared-ray communication circuit to be configured based on IrDA 1.0 simply by adding a simple external circuit.

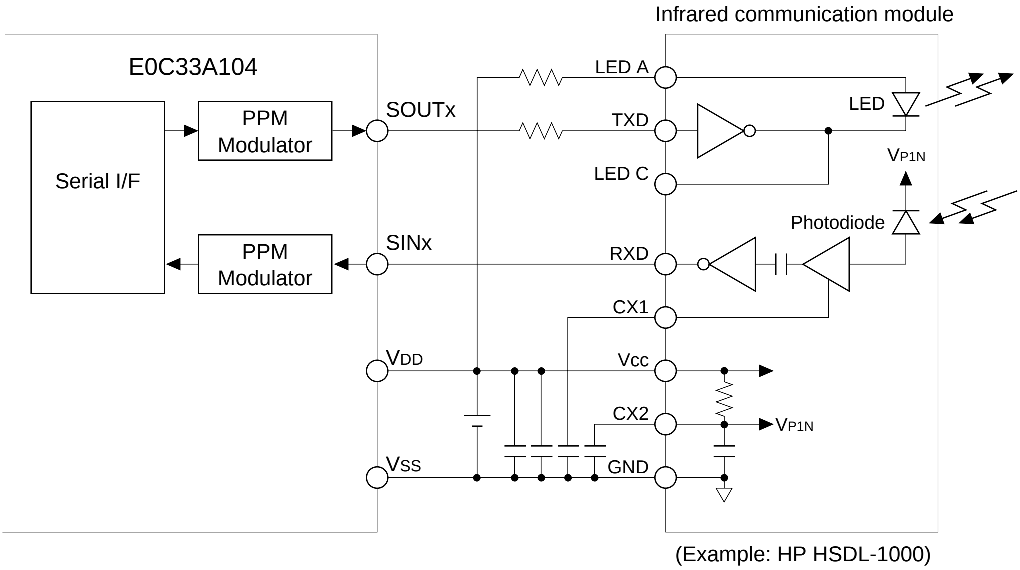


Figure 12.4.1 Configuration Example of IrDA Interface

This IrDA interface function can be used only when the selected transfer mode is an asynchronous mode.
Since the contents of the asynchronous mode are applied directly for the serial-interface functions other than the IrDA interface unit, refer to Section 12.3, "Asynchronous Interface", for details on how to set and control the data formats and data transfers.

12.4.2 Setting IrDA Interface

When performing infrared-ray communication, the following settings must be made before communication can be started:

1. Setting input/output pins
2. Selecting the interface mode (IrDA interface function)
3. Setting the transfer mode
4. Setting the input clock
5. Setting the data format
6. Setting the interrupt/IDMA
7. Setting the input/output logic

The contents for items 1 through 5 have been explained in connection with the asynchronous interface. For details, refer to Section 12.3, "Asynchronous Interface". For details on item 6, refer to Section 12.5, "Serial Interface Interrupts and DMA".

Note: Before making these settings, always make sure the serial interface is inactive (TXENx and RXENx are both set to "0"), as a change in settings during operation could cause a malfunction. In addition, be sure to set the transfer mode in (3) and the following items before selecting the IrDA interface function in (2).

Selecting the IrDA interface function

To use the IrDA interface function, select it using the control bits shown below and then set the 8-bit (or 7-bit) asynchronous mode as the transfer mode.

Ch.0 IrDA interface-function selection: IRMD0[1:0] (D[1:0]) / Serial I/F Ch.0 IrDA register (0x401E4)
Ch.1 IrDA interface-function selection: IRMD1[1:0] (D[1:0]) / Serial I/F Ch.1 IrDA register (0x401E9)

Table 12.4.1 Setting of IrDA Interface

IRMDx1	IRMDx0	Interface mode
1	1	Do not set. (reserved)
1	0	IrDA 1.0 interface
0	1	Do not set. (reserved)
0	0	Normal interface

Note: The IRMDx bit becomes indeterminate when initially reset, so be sure to initialize it in the software.

Setting the input/output logic

When using the IrDA interface, the logic of the input/output signals of the PPM modulator circuit can be changed in accordance with the infrared-ray communication module or the circuit connected externally to the chip. The logic of the internal serial interface is "active-low". If the input/output signals are active-high, the logic of these signals must be inverted before they can be used. The input SINx and output SOUTx logic can be set individually through the use of the IRRLx and IRTLx bits, respectively.

IrDA input logic inversion Ch.0: IRRL0 (D2) / Serial I/F Ch.0 IrDA register (0x401E4)
Ch.1: IRRL1 (D2) / Serial I/F Ch.1 IrDA register (0x401E9)
IrDA output logic inversion Ch.0: IRTL0 (D3) / Serial I/F Ch.0 IrDA register (0x401E4)
Ch.1: IRTL1 (D3) / Serial I/F Ch.1 IrDA register (0x401E9)

The logic of the input/output signal is inverted by writing "1" to each corresponding bit. Logic is not inverted if the bit is set to "0".

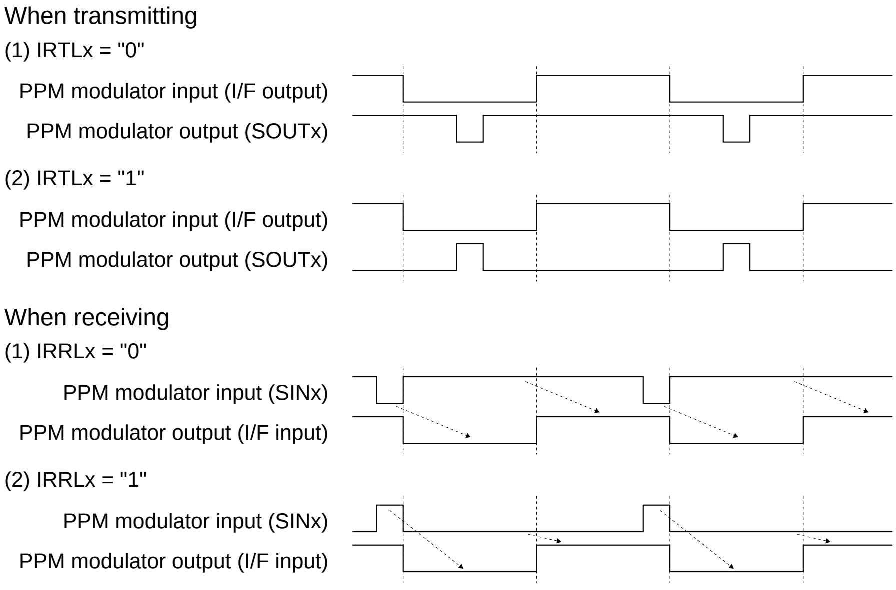

When transmitting
(1) IRTLx = "0"
PPM modulator input (I/F output)
PPM modulator output (SOUTx)
(2) IRTLx = "1"
PPM modulator input (I/F output)
PPM modulator output (SOUTx)
When receiving
(1) IRRLx = "0"
PPM modulator input (SINx)
PPM modulator output (I/F input)
(2) IRRLx = "1"
PPM modulator input (SINx)
PPM modulator output (I/F input)

Figure 12.4.2 IRRLx and IRTLx Settings

Note: The IRRLx and IRTLx bits become indeterminate at initial reset, so be sure to initialize them in the software.

12.4.3 Control and Operation of IrDA Interface

The transmit/receive procedures have been explained in the section on the asynchronous interface, so refer to Section 12.3.3, "Control and Operation of Asynchronous Transfer".

The following describes the data modulation and demodulation performed using the PPM modulator circuit:

When transmitting

During data transmission, the pulse width of the serial interface output signal is set to 3/16 before the signal is output from the SOUTx pin.

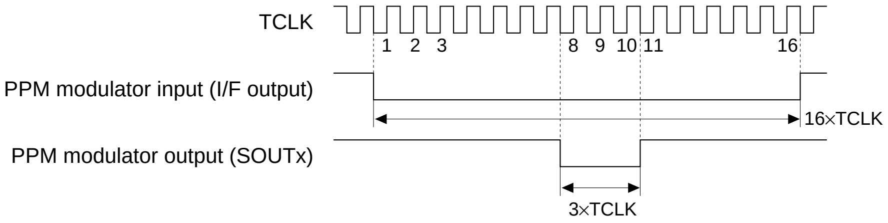

TCLK
1 2 3
8 9 10 11
16
PPM modulator input (I/F output)
16×TCLK
PPM modulator output (SOUTx)
3×TCLK

Figure 12.4.3 Data Modulation by PPM Circuit

When receiving

During data reception, the pulse width of the input signal from SINx is set to 16/3 before the signal is transferred to the serial interface.

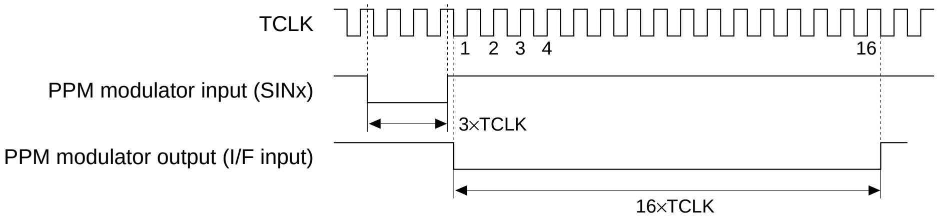


Figure 12.4.4 Demodulation by PPM Circuit

Note: When using the IrDA interface, set the internal division ratio of the serial interface 1/16 (DIVMDx = "1"), rather than 1/8 (DIVMDx = "0").

12.5 Serial Interface Interrupts and DMA

The serial interface can generate the following three types of interrupts in each channel:

- Transmit-buffer empty interrupt
- Receive-buffer full interrupt
- Receive-error interrupt

Transmit-buffer empty interrupt factor

This interrupt factor occurs when the transmit data set in the transmit data register is transferred to the shift register, in which case the interrupt factor flag FSTXx is set to "1". At this time, if the interrupt conditions set using the interrupt control register are met, an interrupt to the CPU is generated.

Occurrence of this interrupt factor indicates that the next transmit data can be written to the transmit data register.

This interrupt factor can also be used to invoke IDMA, enabling transmit data to be written to the register by means of a DMA transfer.

Receive-completion interrupt

This interrupt factor occurs when a receive operation is completed and the receive data taken into the shift register is transferred to the receive data register, in which case the interrupt factor flag FSRXx is set to "1". At this time, if the interrupt conditions set using the interrupt control register are met, an interrupt to the CPU is generated. Occurrence of this interrupt factor indicates that the received data can be read out.

This interrupt factor can also be used to invoke IDMA, enabling the received data to be written into specified memory locations by means of a DMA transfer.

Receive-error interrupt

This interrupt factor occurs when a parity, framing, or overrun error is detected during data reception, in which case the interrupt factor flag FSERRx is set to "1". At this time, if the interrupt conditions set using the interrupt control register are met, an interrupt to the CPU is generated.

Since all three types of errors generate the same interrupt factor, check the error flags PERx (parity error), OERx (overrun error), and FERx (framing error) to identify the type of error that has occurred. In the clock-synchronized mode, parity and framing errors do not occur.

Note: If a receive error (parity or framing error) occurs, the receive-error interrupt and receive-buffer full interrupt factors occur simultaneously. However, since the receive-error interrupt has priority over the receive-buffer full interrupt, the receive-error interrupt is processed first. It is therefore necessary for the receive-buffer full interrupt factor flag be cleared through the use of the receive-error interrupt processing routine.

Control registers of the interrupt controller

Table 12.5.1 shows the interrupt controller's control registers provided for each interrupt source (channel).

Table 12.5.1 Control Register of Interrupt Controller

Channel	Interrupt factor flag	Interrupt enable register	Interrupt priority register	IDMA request register
Ch.0	FSERR0(D0/0x40286)	ESERR0(D0/0x40276)	PSIO0[2:0](D[6:4]/0x40263)	–
	FSRX0(D1/0x40286)	ESRX0(D1/0x40276)		RSRX0(D6/0x40292)
	FSTX0(D2/0x40286)	ESTX0(D2/0x40276)		RSTX0(D7/0x40292)
Ch.1	FSERR1(D3/0x40286)	ESERR1(D3/0x40276)	PSI01[2:0](D[2:0]/0x40264)	–
	FSRX1(D4/0x40286)	ESRX1(D4/0x40276)		RSRX1(D0/0x40293)
	FSTX1(D5/0x40286)	ESTX1(D5/0x40276)		RSTX1(D1/0x40293)

When the interrupt factor described above occurs, the corresponding interrupt factor flag is set to "1". If the interrupt enable register bit for that interrupt factor has been set to "1", an interrupt request is generated. Interrupts caused by an interrupt factor can be disabled by leaving the interrupt enable register bit for that factor set to "0". The interrupt factor flag is set to "1" whenever interrupt conditions are met, regardless of the setting of the interrupt enable register (even if it is set to "0").

The interrupt priority register sets the interrupt priority level of each interrupt source in a range between 0 and 7. An interrupt request to the CPU is accepted only when no other interrupt request of a higher priority has been generated.

In addition, only when the PSR's IE bit = "1" (interrupts enabled) and the set value of the IL is smaller than the input interrupt level set by the interrupt priority register, will the input interrupt request actually be accepted by the CPU.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, refer to Chapter 8, "Interrupt".

Intelligent DMA

The receive-buffer full interrupt and transmit-buffer empty interrupt factors can be used to invoke intelligent DMA (IDMA). This enables successive transmit/receive operations between memory and the transmit/receive-buffer to be performed by means of a DAM transfer.

The following shows the IDMA channel numbers set for each interrupt factor:

	IDMA Ch.
Ch.0 receive-buffer full interrupt:	0x17
Ch.0 transmit-buffer empty interrupt:	0x18
Ch.1 receive-buffer full interrupt:	0x19
Ch.1 transmit-buffer empty interrupt:	0x1A

The IDMA request bit shown in Table 12.5.1 must be set to "1" for IDMA to be invoked. Transfer conditions, etc. on the IDMA side must also be set in advance.

If an interrupt factor occurs when the IDMA request bit is set to "1", IDMA is invoked. No interrupt request is generated at that point. An interrupt request is generated upon completion of the DMA transfer. The bits can also be set so as not to generate an interrupt, with only a DAM transfer performed.

For details on DMA transfer and how to control interrupts upon completion of DMA transfer, refer to Section 9.2, "Intelligent DMA".

Trap vectors

The trap-vector address of each default interrupt factor is set as follows:

	BTA3 = low
Ch.0 receive-error interrupt:	0x0C000E0
Ch.0 receive-buffer full interrupt:	0x0C000E4
Ch.0 transmit-buffer empty interrupt:	0x0C000E8
Ch.1 receive-error interrupt:	0x0C000EC
Ch.1 receive-buffer full interrupt:	0x0C000F0
Ch.1 transmit-buffer empty interrupt:	0x0C000F4

The base address of the trap table can be changed using the TTBR register (0x48134 to 0x48137).

12.6 I/O Memory of Serial Interface

Table 12.6.1 shows the control bits of the serial interface.

For details on the I/O memory of the prescaler that is used to set clocks, as well of that of 8-bit programmable timers, refer to Section 10.2.4, "I/O Memory of Prescaler", and Section 11.3.7, "I/O Memory of 8-Bit Programmable Timers", respectively.

Table 12.6.1 Control Bits of Serial Interface

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Serial I/F Ch.0 transmit data register	**00401E0 (B)**	D7	**TXD07**	Serial I/F Ch.0 transmit data	0x0 to 0xFF(0x7F)				X	R/W	7-bit asynchronous mode does not use TXD07.
		D6	**TXD06**	TXD07(06) = MSB					X		
		D5	**TXD05**	TXD00 = LSB					X		
		D4	**TXD04**						X		
		D3	**TXD03**						X		
		D2	**TXD02**						X		
		D1	**TXD01**						X		
		D0	**TXD00**						X		
Serial I/F Ch.0 receive data register	**00401E1 (B)**	D7	**RXD07**	Serial I/F Ch.0 receive data	0x0 to 0xFF(0x7F)				X	R	7-bit asynchronous mode does not use RXD07 (fixed at 0).
		D6	**RXD06**	RXD07(06) = MSB					X		
		D5	**RXD05**	RXD00 = LSB					X		
		D4	**RXD04**						X		
		D3	**RXD03**						X		
		D2	**RXD02**						X		
		D1	**RXD01**						X		
		D0	**RXD00**						X		
Serial I/F Ch.0 status register	**00401E2 (B)**	D7–5	–	–	–				–	–	0 when being read.
		D4	**FER0**	Ch.0 flaming error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D3	**PER0**	Ch.0 parity error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D2	**OER0**	Ch.0 overrun error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D1	**TDBE0**	Ch.0 transmit data buffer empty	1	Empty	0	Buffer full	1	R	
		D0	**RDBF0**	Ch.0 receive data buffer full	1	Buffer full	0	Empty	0	R	
Serial I/F Ch.0 control register	**00401E3 (B)**	D7	**TXEN0**	Ch.0 transmit enable	1	Enabled	0	Disabled	0	R/W	
		D6	**RXEN0**	Ch.0 receive enable	1	Enabled	0	Disabled	0	R/W	
		D5	**EPR0**	Ch.0 parity enable	1	With parity	0	No parity	X	R/W	Valid only in asynchronous mode.
		D4	**PMD0**	Ch.0 parity mode selection	1	Odd	0	Even	X	R/W	
		D3	**STPB0**	Ch.0 stop bit selection	1	2 bits	0	1 bit	X	R/W	
		D2	**SSCK0**	Ch.0 input clock selection	1	#SCLK0	0	Internal clock	X	R/W	
		D1	**SMD01**	Ch.0 transfer mode selection	SMD0[1:0]		Transfer mode		X	R/W	
		D0	**SMD00**		1	1	8-bit asynchronous		X		
					1	0	7-bit asynchronous				
					0	1	Clock sync. Slave				
					0	0	Clock sync. Master				
Serial I/F Ch.0 IrDA register	**00401E4 (B)**	D7–5	–	–	–				–	–	0 when being read.
		D4	**DIVMD0**	Ch.0 async. clock division ratio	1	1/8	0	1/16	X	R/W	
		D3	**IRTL0**	Ch.0 IrDA I/F output logic inversion	1	Inverted	0	Direct	X	R/W	Valid only in asynchronous mode.
		D2	**IRRL0**	Ch.0 IrDA I/F input logic inversion	1	Inverted	0	Direct	X	R/W	
		D1	**IRMD01**	Ch.0 interface mode selection	IRMD0[1:0]		I/F mode		X	R/W	
		D0	**IRMD00**		1	1	reserved		X		
					1	0	IrDA 1.0				
					0	1	reserved				
					0	0	General I/F				
Serial I/F Ch.1 transmit data register	**00401E5 (B)**	D7	**TXD17**	Serial I/F Ch.1 transmit data	0x0 to 0xFF(0x7F)				X	R/W	7-bit asynchronous mode does not use TXD17.
		D6	**TXD16**	TXD17(16) = MSB					X		
		D5	**TXD15**	TXD10 = LSB					X		
		D4	**TXD14**						X		
		D3	**TXD13**						X		
		D2	**TXD12**						X		
		D1	**TXD11**						X		
		D0	**TXD10**						X		
Serial I/F Ch.1 receive data register	**00401E6 (B)**	D7	**RXD17**	Serial I/F Ch.1 receive data	0x0 to 0xFF(0x7F)				X	R	7-bit asynchronous mode does not use RXD17 (fixed at 0).
		D6	**RXD16**	RXD17(16) = MSB					X		
		D5	**RXD15**	RXD10 = LSB					X		
		D4	**RXD14**						X		
		D3	**RXD13**						X		
		D2	**RXD12**						X		
		D1	**RXD11**						X		
		D0	**RXD10**						X		
Serial I/F Ch.1 status register	**00401E7 (B)**	D7–5	–	–	–				–	–	0 when being read.
		D4	**FER1**	Ch.1 flaming error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D3	**PER1**	Ch.1 parity error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D2	**OER1**	Ch.1 overrun error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D1	**TDBE1**	Ch.1 transmit data buffer empty	1	Empty	0	Buffer full	1	R	
		D0	**RDBF1**	Ch.1 receive data buffer full	1	Buffer full	0	Empty	0	R	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Serial I/F Ch.1 control register	**00401E8 (B)**	D7	**TXEN1**	Ch.1 transmit enable	1	Enabled	0	Disabled	0	R/W	
		D6	**RXEN1**	Ch.1 receive enable	1	Enabled	0	Disabled	0	R/W	
		D5	**EPR1**	Ch.1 parity enable	1	With parity	0	No parity	X	R/W	Valid only in asynchronous mode.
		D4	**PMD1**	Ch.1 parity mode selection	1	Odd	0	Even	X	R/W	
		D3	**STPB1**	Ch.1 stop bit selection	1	2 bits	0	1 bit	X	R/W	
		D2	**SSCK1**	Ch.1 input clock selection	1	#SCLK1	0	Internal clock	X	R/W	
		D1	**SMD11**	Ch.1 transfer mode selection	SMD1[1:0]		Transfer mode		X	R/W	
		D0	**SMD10**		1	1	8-bit asynchronous		X		
					1	0	7-bit asynchronous				
					0	1	Clock sync. Slave				
					0	0	Clock sync. Master				
Serial I/F Ch.1 IrDA register	**00401E9 (B)**	D7–5	–	–	–				–	–	0 when being read.
		D4	**DIVMD1**	Ch.1 async. clock division ratio	1	1/8	0	1/16	X	R/W	
		D3	**IRTL1**	Ch.1 IrDA I/F output logic inversion	1	Inverted	0	Direct	X	R/W	Valid only in asynchronous mode.
		D2	**IRRL1**	Ch.1 IrDA I/F input logic inversion	1	Inverted	0	Direct	X	R/W	
		D1	**IRMD11**	Ch.1 interface mode selection	IRMD1[1:0]		I/F mode		X	R/W	
		D0	**IRMD10**		1	1	reserved		X		
					1	0	IrDA 1.0				
					0	1	reserved				
					0	0	General I/F				
8-bit timer, serial I/F Ch.0 interrupt priority register	**0040263 (B)**	D7	–	reserved	–				–	–	0 when being read.
		D6	**PSI002**	Serial interface Ch.0 interrupt level	0 to 7				X	R/W	
		D5	**PSI001**						X		
		D4	**PSI000**						X		
		D3	–	reserved	–				–	–	0 when being read.
		D2	**P8TM2**	8-bit timer 0–3 interrupt level	0 to 7				X	R/W	
		D1	**P8TM1**						X		
		D0	**P8TM0**						X		
Serial I/F Ch.1, A/D interrupt priority register	**0040264 (B)**	D7	–	reserved	–				–	–	0 when being read.
		D6	**PAD2**	A/D converter interrupt level	0 to 7				X	R/W	
		D5	**PAD1**						X		
		D4	**PAD0**						X		
		D3	–	reserved	–				–	–	0 when being read.
		D2	**PSI012**	Serial interface Ch.1 interrupt level	0 to 7				X	R/W	
		D1	**PSI011**						X		
		D0	**PSI010**						X		
Serial I/F interrupt enable register	**0040276 (B)**	D7–6	–	reserved	–				–	–	0 when being read.
		D5	**ESTX1**	SIF Ch.1 transmit buffer empty	1	Enabled	0	Disabled	0	R/W	
		D4	**ESRX1**	SIF Ch.1 receive buffer full					0	R/W	
		D3	**ESERR1**	SIF Ch.1 receive error					0	R/W	
		D2	**ESTX0**	SIF Ch.0 transmit buffer empty					0	R/W	
		D1	**ESRX0**	SIF Ch.0 receive buffer full					0	R/W	
		D0	**ESERR0**	SIF Ch.0 receive error					0	R/W	
Serial I/F interrupt factor flag register	**0040286 (B)**	D7–6	–	reserved	–				–	–	0 when being read.
		D5	**FSTX1**	SIF Ch.1 transmit buffer empty	1	Factor is generated	0	No factor is generated	X	R/W	
		D4	**FSRX1**	SIF Ch.1 receive buffer full					X	R/W	
		D3	**FSERR1**	SIF Ch.1 receive error					X	R/W	
		D2	**FSTX0**	SIF Ch.0 transmit buffer empty					X	R/W	
		D1	**FSRX0**	SIF Ch.0 receive buffer full					X	R/W	
		D0	**FSERR0**	SIF Ch.0 receive error					X	R/W	
16-bit timer 5, 8-bit timer, serial I/F Ch.0 IDMA request register	**0040292 (B)**	D7	**RSTX0**	SIF Ch.0 transmit buffer empty	1	IDMA request	0	Interrupt request	X	R/W	
		D6	**RSRX0**	SIF Ch.0 receive buffer full					X	R/W	
		D5	**R8TU3**	8-bit timer 3 underflow					X	R/W	
		D4	**R8TU2**	8-bit timer 2 underflow					X	R/W	
		D3	**R8TU1**	8-bit timer 1 underflow					X	R/W	
		D2	**R8TU0**	8-bit timer 0 underflow					X	R/W	
		D1	**R16TC51**	16-bit timer 51 comparison match					X	R/W	
		D0	**R16TU51**	16-bit timer 51 underflow					X	R/W	
Serial I/F Ch.1, A/D IDMA request register	**0040293 (B)**	D7–3	–	reserved	–				–	–	
		D2	**RADE**	A/D converter	1	IDMA request	0	Interrupt request	X	R/W	
		D1	**RSTX1**	SIF Ch.1 transmit buffer empty					X	R/W	
		D0	**RSRX1**	SIF Ch.1 receive buffer full					X	R/W	
P0 function select register	**00402D0 (B)**	D7	**CFP07**	P07 function selection	1	#SRDY1	0	P07	0	W	Undefined in read.
		D6	**CFP06**	P06 function selection	1	#SCLK1	0	P06	0	W	
		D5	**CFP05**	P05 function selection	1	SOUT1	0	P05	0	W	
		D4	**CFP04**	P04 function selection	1	SIN1	0	P04	0	W	
		D3	**CFP03**	P03 function selection	1	#SRDY0	0	P03	0	W	
		D2	**CFP02**	P02 function selection	1	#SCLK0	0	P02	0	W	
		D1	**CFP01**	P01 function selection	1	SOUT0	0	P01	0	W	
		D0	**CFP00**	P00 function selection	1	SIN0	0	P00	0	W	

CFP07–CFP00: P0[7:0] pin function selection (D[7:0]) / P0 function select register (0x402D0)

Selects the pins used for the serial interface.

Write "1":	Serial-interface input/output pin
Write "0":	I/O port pin
Read:	Invalid

Select the pins used for the serial interface from among P00 through P07 by writing "1" to CFP00 through CFP07. P00–P03 (SIN0, SOUT0, #SCLK0, #SRDY0) are used for channel 0; P04–P07 (SIN1, SOUT1, #SCLK1, #SRDY1) are used for channel 1. If the bit for a pin is set to "0", the pin functions as an I/O port.
The necessary input/output pins differ depending on the transfer mode set (see Table 12.1.3).
At cold start, CFP is set to "0" (I/O port). At hot start, CFP retains its state from prior to the initial reset.

TXD07–TXD00: Ch.0 transmit data (D[7:0]) / Serial I/F Ch.0 transmit data register (0x401E0)
TXD17–TXD10: Ch.1 transmit data (D[7:0]) / Serial I/F Ch.1 transmit data register (0x401E5)

Sets transmit data.
When data is written to this register (transmit buffer) after "1" is written to TXENx, a transmit operation is begun. TDBEx is set to "1" (transmit-buffer empty) when the data is transferred to the shift register. A transmit-buffer empty interrupt factor is simultaneously generated. The next transmit data can be written to the buffer at any time thereafter, even when the serial interface is sending data.
In the 7-bit asynchronous mode, TXDx7 (MSB) is ignored.
The serial-converted data is output from the SOUT pin beginning with the LSB, in which the bits set to "1" are output as high-level signals and those set to "0" output as low-level signals.
This register can be read as well as written.
At initial reset, the content of TXDx becomes indeterminate.

RXD07–RXD00: Ch.0 receive data (D[7:0]) / Serial I/F Ch.0 receive data register (0x401E1)
RXD17–RXD10: Ch.1 receive data (D[7:0]) / Serial I/F Ch.1 receive data register (0x401E6)

Stores received data.
When a receive operation is completed and the data received in the shift register is transferred to this register (receive buffer), RDBFx is set to "1" (receive buffer full). At the same time, a receive-buffer full interrupt factor is generated. Thereafter, the data can be read out at any time before a receive operation for the next data is completed. If the next data receive operation is completed before this register is read out, the data in it is overwritten with the newly received data, causing an overrun error to occur.
In the 7-bit asynchronous mode, "0" is stored in RXDx7.
The serial data input from the SINx pin is converted into parallel data beginning with the LSB, with the high-level signals changed to "1"s and the low-level signals changed to "0"s. The resulting data is stored in this buffer.
This register is a read-only register, so no data can be written to it.
At initial reset, the content of RXDx becomes indeterminate.

FER0: Ch.0 framing-error flag (D4) / Serial I/F Ch.0 status register (0x401E2)
FER1: Ch.1 framing-error flag (D4) / Serial I/F Ch.1 status register (0x401E7)

Indicates whether a framing error occurred.

Read "1":	An error occurred
Read "0":	No error occurred
Write "1":	Invalid
Write "0":	Reset to "0"

The FERx flag is an error flag indicating whether a framing error occurred. When an error has occurred, it is set to "1". A framing error occurs when data with a stop bit = "0" is received in the asynchronous mode.
The FERx flag is reset by writing "0".
At initial reset, as well as when RXENx and TXENx both are set to "0", the FERx flag is set to "0" (no error).

PER0: Ch.0 parity-error flag (D3) / Serial I/F Ch.0 status register (0x401E2)
PER1: Ch.1 parity-error flag (D3) / Serial I/F Ch.1 status register (0x401E7)

Indicates whether a parity error occurred.

Read "1":	An error occurred
Read "0":	No error occurred
Write "1":	Invalid
Write "0":	Reset to "0"

The PERx flag is an error flag indicating whether a parity error occurred. When an error has occurred, it is set to "1". Parity checks are valid only in the asynchronous mode with EPRx set to "1" (parity added). This check is performed when the received data is transferred from the shift register to the receive data register.
The PERx flag is reset by writing "0".
At initial reset, as well as when RXENx and TXENx both are set to "0", PERx is set to "0" (no error).

OER0: Ch.0 overrun-error flag (D2) / Serial I/F Ch.0 status register (0x401E2)
OER1: Ch.1 overrun-error flag (D2) / Serial I/F Ch.1 status register (0x401E7)

Indicates whether an overrun error occurred.

Read "1":	An error occurred
Read "0":	No error occurred
Write "1":	Invalid
Write "0":	Reset to "0"

The OERx flag is an error flag indicating whether an overrun error occurred. When an error has occurred, it is set to "1". An overrun error occurs when the next receive operation is completed before the receive data register is read out, resulting in the receive data register being overwritten.
The OERx flag is reset by writing "0".
At initial reset, as well as when RXENx and TXENx both are set to "0", OERx is set to "0" (no error).

TDBE0: Ch.0 transmit data buffer empty (D1) / Serial I/F Ch.0 status register (0x401E2)
TDBE1: Ch.1 transmit data buffer empty (D1) / Serial I/F Ch.1 status register (0x401E7)

Indicates the status of the transmit data register (buffer).

Read "1":	Buffer empty
Read "0":	Buffer full
Write:	Invalid

TDBEx is set to "0" when transmit data is written to the transmit data register, and is set to "1" when this data is transferred to the shift register (transmit operation started).
Transmit data is written to the transmit data register when this bit = "1".
At initial reset, TDBEx is set to "1" (buffer empty).

RDBF0: Ch.0 receive data buffer full (D0) / Serial I/F Ch.0 status register (0x401E2)
RDBF1: Ch.1 receive data buffer full (D0) / Serial I/F Ch.1 status register (0x401E7)

Indicates the status of the receive data register (buffer).

Read "1":	Buffer full
Read "0":	Buffer empty
Write:	Invalid

RDBFx is set to "1" when the data received in the shift register is transferred to the receive data register (receive operation completed), indicating that the received data can be read out. This bit is reset to "0" when the data is read out.
At initial reset, RDBFx is set to "0" (buffer empty).

TXEN0: Ch.0 transmit enable (D7) / Serial I/F Ch.0 control register (0x401E3)
TXEN1: Ch.1 transmit enable (D7) / Serial I/F Ch.1 control register (0x401E8)

Enables each channel for transmit operations.

Write "1":	Transmit enabled
Write "0":	Transmit disabled
Read:	Valid

When TXENx for a channel is set to "1", the channel is enabled for transmit operations. When TXENx is set to "0", the channel is disabled for transmit operations.
Always make sure the TXENx = "0" before setting the transfer mode and other conditions.
At initial reset, TXENx is set to "0" (transmit disabled).

RXEN0: Ch.0 receive enable (D6) / Serial I/F Ch.0 control register (0x401E3)
RXEN1: Ch.1 receive enable (D6) / Serial I/F Ch.1 control register (0x401E8)

Enables each channel for receive operations.

Write "1":	Receive enabled
Write "0":	Receive disabled
Read:	Valid

When RXENx for a channel is set to "1", the channel is enabled for receive operations. When RXENx is set to "0", the channel is disabled for receive operations.
Always make sure the RXENx = "0" before setting the transfer mode and other conditions.
At initial reset, RXENx is set to "0" (receive disabled).

EPR0: Ch.0 parity enable (D5) / Serial I/F Ch.0 control register (0x401E3)
EPR1: Ch.1 parity enable (D5) / Serial I/F Ch.1 control register (0x401E8)

Selects a parity function.

Write "1":	Parity added
Write "0":	No parity added
Read:	Valid

EPRx is used to select whether receive data is to be checked for parity, and whether a parity bit is to be added to transmit data. When EPRx is set to "1", the receive data is checked for parity. A parity bit is automatically added to the transmit data. When EPRx is set to "0", parity is not checked and no parity bit is added.
The parity function is only valid in the asynchronous mode. Settings of EPRx have no effect in the clock-synchronized mode.
At initial reset, EPRx becomes indeterminate.

PMD0: Ch.0 parity mode selection (D4) / Serial I/F Ch.0 control register (0x401E3)
PMD1: Ch.1 parity mode selection (D4) / Serial I/F Ch.1 control register (0x401E8)

Selects an odd or even parity.

Write "1":	Odd parity
Write "0":	Even parity
Read:	Valid

Odd parity is selected by writing "1" to PMDx, and even parity is selected by writing "0". Parity check and the addition of a parity bit are only effective in asynchronous transfers in which EPRx is set to "1". If EPRx = "0", settings of PMDx do not have any effect.
At initial reset, PMDx becomes indeterminate.

STPB0: Ch.0 stop bit selection (D3) / Serial I/F Ch.0 control register (0x401E3)
STPB1: Ch.1 stop bit selection (D3) / Serial I/F Ch.1 control register (0x401E8)

Selects a stop-bit length during the performance of an asynchronous transfer.

Write "1":	2 bits
Write "0":	1 bit
Read:	Valid

STPBx is only valid in an asynchronous transfer. Two stop bits are selected by writing "1" to STPBx , and one stop bit is selected by writing "0". The start bit is fixed at 1 bit.
Settings of STPBx are ignored during the performance of a clock-synchronized transfer.
At initial reset, STPBx becomes indeterminate.

SSCK0: Ch.0 input clock selection (D2) / Serial I/F Ch.0 control register (0x401E3)
SSCK1: Ch.1 input clock selection (D2) / Serial I/F Ch.1 control register (0x401E8)

Selects the clock source for an asynchronous transfer.

Write "1":	#SCLK (external clock)
Write "0":	OSC3 (internal clock)
Read:	Valid

During operation in the asynchronous mode, this bit is used to select the clock source between an internal clock (output by an 8-bit programmable timer) and an external clock (input from the #SCLKx pin). An external clock is selected by writing "1" to this bit, and an internal clock is selected by writing "0".
At initial reset, SSCKx becomes indeterminate.

SMD01–SMD00: Ch.0 transfer mode selection (D[1:0]) / Serial I/F Ch.0 control register (0x401E3)
SMD11–SMD10: Ch.1 transfer mode selection (D[1:0]) / Serial I/F Ch.1 control register (0x401E8)

Sets the transfer mode of the serial interface as shown in Table 12.6.2 below.

Table 12.6.2 Setting of Transfer Mode

SMDx1	SMDx0	Transfer mode
1	1	8-bit asynchronous mode
1	0	7-bit asynchronous mode
0	1	Clock-synchronized slave mode
0	0	Clock-synchronized master mode

The SMDx bit can be read as well as written.
When using the IrDA interface, always be sure to set an asynchronous mode for the transfer mode.
At initial reset, SMDx becomes indeterminate.

DIVMD0: Sampling clock division ratio (D4) / Serial I/F Ch.0 IrDA register (0x401E4)
DIVMD1: Sampling clock division ratio (D4) / Serial I/F Ch.1 IrDA register (0x401E9)

Selects the division ratio of the sampling clock.

Write "1":	1/8
Write "0":	1/16
Read:	Valid

Select the division ratio necessary to generate the sampling clock for asynchronous transfers. When DIVMDx is set to "1", the sampling clock is generated from the input clock of the serial interface (output by an 8-bit programmable timer or input from #SCLKx) by dividing it by 8. When DIVMDx is set to "0", the input clock is divided by 16.
At initial reset, DIVMDx becomes indeterminate.

IRTL0: Ch.0 IrDA output logic inversion (D3) / Serial I/F Ch.0 IrDA register (0x401E4)
IRTL1: Ch.1 IrDA output logic inversion (D3) / Serial I/F Ch.1 IrDA register (0x401E9)

Inverts the logic of the IrDA output signal.

Write "1": Inverted
Write "0": Not inverted
Read: Valid

When using the IrDA interface, set the logic of the SOUTx output signal to suit the infrared-ray communication circuit that is connected external to the chip. If IRTLx is set to "1", a high pulse is output when the output data = "0" (held low-level when the output data = "1"). If IRTLx is set to "0", a low pulse is output when the output data = "0" (held high-level when the output data = "1").
At initial reset, IRTLx becomes indeterminate.

IRRL0: Ch.0 IrDA input logic inversion (D2) / Serial I/F Ch.0 IrDA register (0x401E4)
IRRL1: Ch.1 IrDA input logic inversion (D2) / Serial I/F Ch.1 IrDA register (0x401E9)

Inverts the logic of the IrDA input signal.

Write "1": Inverted
Write "0": Not inverted
Read: Valid

When using the IrDA interface, set the logic of the signal that is input from an external infrared-ray communication circuit to the chip to suit the serial interface. If IRRLx is set to "1", a high pulse is input as a logic "0". If IRRLx is set to "0", a low pulse is input as a logic "0".
At initial reset, IRRLx becomes indeterminate.

IRMD01–IRMD00: Ch.0 IrDA interface mode selection (D[1:0]) / Serial I/F Ch.0 IrDA register (0x401E4)
IRMD11–IRMD10: Ch.1 IrDA interface mode selection (D[1:0]) / Serial I/F Ch.1 IrDA register (0x401E9)

Selects the IrDA interface function.

Table 12.6.3 IrDA Interface Setting

IRMDx1	IRMDx0	Interface mode
1	1	Do not set. (reserved)
1	0	IrDA 1.0 interface
0	1	Do not set. (reserved)
0	0	Normal interface

When using the IrDA interface function, write "10" to IRMDx while setting to an asynchronous mode for the transfer mode. If the IrDA interface function is not to be used, write "00" to IRMDx.
At initial reset, IRMDx becomes indeterminate.

Note: This selection must always be performed before the transfer mode and other conditions are set.

PSIO02–PSIO00: Ch.0 interrupt level (D[6:4]) / 8-bit timer, serial I/F Ch.0 interrupt priority register (0x40263)
PSIO12–PSIO10: Ch.1 interrupt level (D[2:0]) / Serial I/F Ch.1, A/D interrupt priority register (0x40264)

Sets the priority level of the serial-interface interrupt.
The interrupt priority level can be set for each channel in the range of 0 to 7.
At initial reset, PSIOx becomes indeterminate.

ESERR0, ESRX0, ESTX0: Ch.0 interrupt enable (D0,D1,D2) / Serial I/F interrupt enable register (0x40276)
ESERR1, ESRX1, ESTX1: Ch.1 interrupt enable (D3,D4,D5) / Serial I/F interrupt enable register (0x40276)

Enable or disable interrupt generation to the CPU.

Write "1":	Interrupt enabled
Write "0":	Interrupt disabled
Read:	Valid

The ESERRx, ESRXx, and ESTXx bits are interrupt enable bits corresponding to receive-error, receive-buffer full, and transmit-buffer empty interrupt factors, respectively, in each channel. The interrupts for which this bit is set to "1" are enabled, and the interrupts for which this bit is set to "0" are disabled.
At initial reset, all these bits are set to "0" (interrupts disabled).

FSERR0, FSRX0, FSTX0: Ch.0 interrupt factor flags (D0,D1,D2) / Serial I/F interrupt factor flag register (0x40286)
FSERR1, FSRX1, FSTX1: Ch.1 interrupt factor flags (D3,D4,D5) / Serial I/F interrupt factor flag register (0x40286)

Indicate the status of serial-interface interrupt generation.

When read

Read "1":	An interrupt factor occurred
Read "0":	No interrupt factor occurred

When written using the reset-only method (default)

Write "1":	Flag is reset
Write "0":	Invalid

When written using the read/write method

Write "1":	Flag is set
Write "0":	Flag is reset

The FSERRx, FSRXx, and FSTXx flags are interrupt factor flags corresponding to receive-error, receive-buffer full, and transmit-buffer empty interrupts, respectively, in each channel. The flag is set to "1" when each interrupt factor occurs.
A transmit-buffer empty interrupt factor occurs when transmit data is transferred from the transmit data register to the shift register.
A receive-buffer full interrupt factor occurs when receive data is transferred from the shift register to the receive data register.
A receive-error interrupt factor occurs when a parity, framing, or overrun error is detected during reception of data.
At this time, if the following conditions are met, an interrupt to the CPU is generated:
1. The corresponding interrupt enable register bit is set to "1".
2. No other interrupt request of a higher priority has been generated.
3. The PSR's IE bit is set to "1" (interrupts enabled).
4. The set value of the corresponding interrupt priority register is higher than the CPU interrupt level (IL).
When using the receive-buffer full or transmit-buffer empty interrupt factor as an IDMA request, the fact that the above conditions are met does not necessarily mean that an interrupt request to the CPU has been output simultaneously when an interrupt factor occurs. An interrupt is generated under the above conditions upon completion of the data transfer by IDMA, provided that interrupts are enabled by settings on the IDMA side.
The interrupt factor flag is set to "1" whenever an interrupt factor occurs, regardless of the settings of the interrupt-enable and interrupt priority registers.
If the next interrupt is to be accepted following the occurrence of an interrupt, it is necessary that the interrupt factor flag be reset, and that the PSR be set up again (by setting the IE bit to "1" after setting the IL to a value lower than the level indicated by the interrupt priority register, or by executing the reti instruction).
The interrupt factor flag can only be reset by writing to it in the software. Note that if the PSR is set up again to accept interrupts generated (or if the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt occurs again. Note also that the value to be written to reset the flag is "1" when the reset-only method (RSTONLY = "1") is used, and "0" when the read/write method (RSTONLY = "0") is used.
At initial reset, all of these flags become indeterminate, so be sure to reset them in the software.

RSRX0, RSTX0: Ch.0 IDMA request (D6, D7) /
16-bit timer 5, 8-bit timer, serial I/F Ch.0 IDMA request register (0x40292)
RSRX1, RSTX1: Ch.1 IDMA request (D0, D1) / Serial I/F Ch.1, A/D IDMA request register (0x40293)

Specifies whether to invoke IDMA when an interrupt factor occurs.

Write "1": IDMA request
Write "0": Interrupt request
Read: Valid

The RSRXx and RSTXx bits are IDMA request bits corresponding to receive-buffer full and transmit-buffer empty interrupt factors, respectively. If the bit is set to "1", IDMA is invoked when an interrupt factor occurs, thus performing a programmed data transfer. If this bit is set to "0", normal interrupt processing is performed, without invoking IDMA.
For details on IDMA, refer to Section 9.2, "Intelligent DMA".
At initial reset, these bits become indeterminate, so be sure to initialize them in the software.

12.7 Programming Notes

(1) Before setting various serial-interface parameters, make sure the transmit and receive operations are disabled (TXENx = RXENx = "0").

(2) When the serial interface is transmitting or receiving data, do not set TXENx or RXENx to "0", and do not execute the slp instruction.

(3) In clock-synchronized transfers, the mode of communication is half-duplex, in which the clock line is shared between the transmit and receive units. Therefore, RXENx and TXENx cannot be enabled simultaneously.

(4) Since the pin function select register (CFP) is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.

(5) After an initial reset, the interrupt factor flag and IDMA request register become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, reset this flag and this register in the program.

(6) If a receive error occurs, the receive-error interrupt and receive-buffer full interrupt factors occur simultaneously. However, since the receive-error interrupt has priority over the receive-buffer full interrupt, the receive-error interrupt is processed first. Therefore, it is necessary to reset the receive-buffer full interrupt factor flag through the use of the receive-error interrupt processing routine

(7) To prevent the regeneration of interrupts due to the same factor following the occurrence of an interrupt, always be sure to reset the interrupt factor flag before setting the PSR again or executing the reti instruction.

(8) Follow the procedure described below to initialize the serial interface.

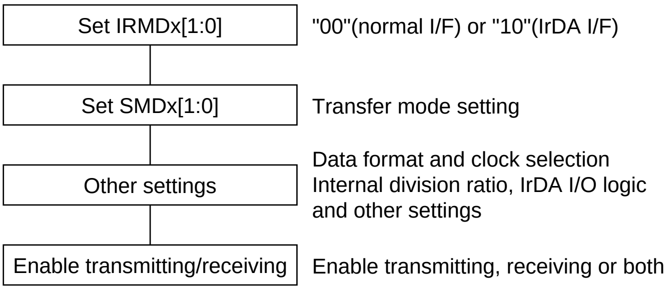

Set IRMDx[1:0]
"00"(normal I/F) or "10"(IrDA I/F)
Set SMDx[1:0]
Transfer mode setting
Other settings
Data format and clock selection
Internal division ratio, IrDA I/O logic
and other settings
Enable transmitting/receiving
Enable transmitting, receiving or both

Figure 12.7.1 Serial Interface Initialize Procedure

(9) The maximum transfer rate in the clock-synchronized mode is limited to 1 Mbps.
The maximum transfer clock (TCLK) frequency in the asynchronous mode is limited to 1 MHz.

(10) When sending data in clock-synchronized master mode, follow the steps below.

1) Turn on the 8-bit timer output to the serial interface.
2) Wait until at least one underflow pulse is provided to the serial interface from the 8-bit timer.
3) Write sending data to the TXD data register of the serial interface.

If the above steps are not followed, the serial interface may send 0xFF data first before the correct data is transferred.

(11) When using the serial interface in the clock-synchronized mode, be aware that the #SRDYx and #SCLKx pins does not change their I/O direction even if only the transfer mode is changed.
The I/O direction of the #SRDYx and #SCLKx pins are changed as follows (when the P0x port is set for the serial interface):

#SRDYx (P03/P07): When transmit/receive operation is enabled in slave mode, the #SRDYx pin enters output mode.
Otherwise, the #SRDYx pin stays in input mode.

#SCLKx (P02/P06): When transmit/receive operation is enabled in master mode, the #SRDYx pin enters output mode.
Otherwise, the #SRDYx pin stays in input mode.

The SINx (P00/P04) and SOUTx (P01/P05) pins are set to input mode and output mode, respectively, immediately after the P0x port is set for the serial interface.

13 A/D and D/A Converters

The E0C33A104 contains an A/D and a D/A converter. This chapter describes the functions of each converter and how to control them.

13.1 A/D Converter

13.1.1 Features and Structure of A/D Converter

The E0C33A104 contains an A/D converter with the following features:

• Conversion method:	Successive comparison
• Resolution:	10 bits
• Input channels:	Maximum of 8
• Conversion time:	Maximum of 10 µs (when a 2-MHz input clock is selected)

- The analog-conversion voltage range (2 V to the power supply voltage) can be set using the reference voltage pins (AVRH and AVRL).
- Two conversion modes can be selected:

  Normal mode: Conversion is completed in one operation.

  Continuous mode: Conversion is continuous and terminated through software control.

  Continuous conversion of multiple channels can be performed in each mode.
- Four types of A/D-conversion start triggers can be selected:

  Triggered by the external pin (#ADTRG)

  Triggered by the underflow of the 16-bit programmable timer 0

  Triggered by the underflow of the 8-bit programmable timer 0

  Triggered by the software
- A/D conversion results can be read out from a 10-bit data register.
- An interrupt is generated upon completion of A/D conversion.

Figure 13.1.1 shows the structure of the A/D converter.

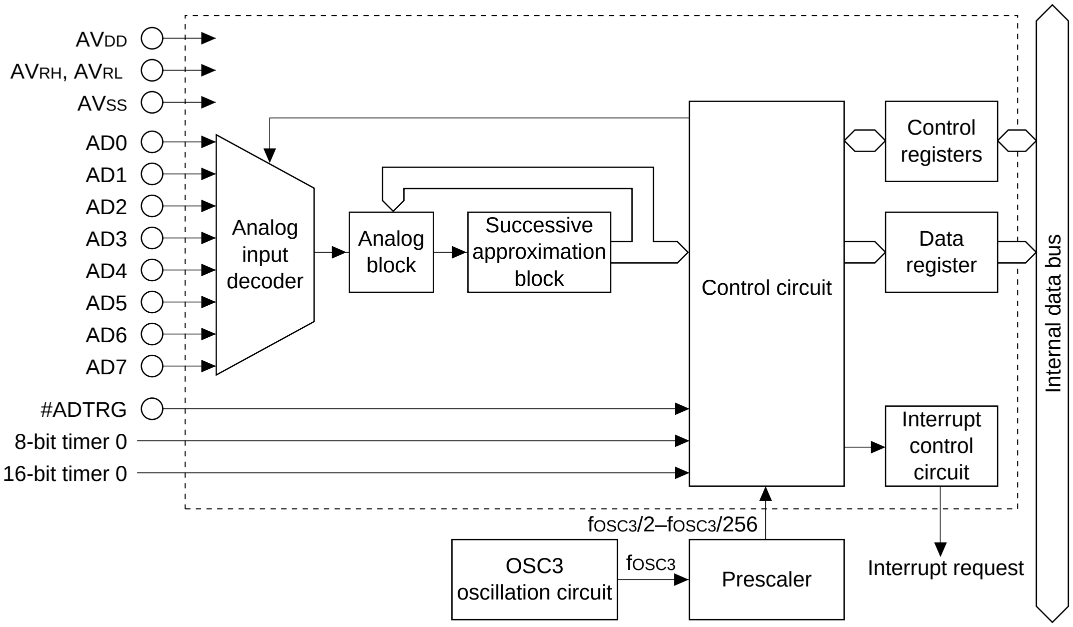


Figure 13.1.1 Structure of A/D Converter

13.1.2 I/O Pins of A/D Converter

Table 13.1.1 shows the pins used by the A/D converter.

Table 13.1.1 I/O Pins of A/D Converter

Pin name	Pin No.		I/O	Pull-up	Function	Function select bit
	QFP5-128	QFP15-128				
K52/#ADTRG	35	32	I	Built-in	Input port / AD trigger	CFK52(D2)/K5 function select register(0x402C0)
K60/AD0	26	23	I	Built-in	Input port / AD converter input 0	CFK60(D0)/K6 function select register(0x402C5)
K61/AD1	25	22	I	Built-in	Input port / AD converter input 1	CFK61(D1)/K6 function select register(0x402C5)
K62/AD2	24	21	I	Built-in	Input port / AD converter input 2	CFK62(D2)/K6 function select register(0x402C5)
K63/AD3	23	20	I	Built-in	Input port / AD converter input 3	CFK63(D3)/K6 function select register(0x402C5)
K64/AD4	22	19	I	Built-in	Input port / AD converter input 4	CFK64(D4)/K6 function select register(0x402C5)
K65/AD5	21	18	I	Built-in	Input port / AD converter input 5	CFK65(D5)/K6 function select register(0x402C5)
K66/AD6	20	17	I	Built-in	Input port / AD converter input 6	CFK66(D6)/K6 function select register(0x402C5)
K67/AD7	19	16	I	Built-in	Input port / AD converter input 7	CFK67(D7)/K6 function select register(0x402C5)
AVDD	29	26	–	–	Power supply for analog system (+)	–
AVSS	18	15	–	–	Power supply for analog system (-)	–
AVRH	16	13	–	–	Analog reference voltage (+)	–
AVRL	17	14	–	–	Analog reference voltage (-)	–

AVDD, AVSS (analog power-supply pins)

AVDD and AVSS are the power-supply pins for the analog circuit. The voltage levels supplied to these pins must be AVDD = VDD and AVSS = VSS. These power-supply pins are shared with the D/A converter.

AVRH, AVRL (reference-voltage input pins)

AVRH and AVRL are the reference-voltage input pins for the A/D converter. The input voltage range of the A/D converter is determined by the reference voltages of these pins (AVRL to AVRH). The voltages input to these pins must be AVSS ≤ AVRL ≤ AVRH ≤ AVDD and (AVRH - AVRL) ≤ 2 V.

Note: These pins should be set as AVRH = AVRL = GND if the A/D converter is not used.

AD[7:0] (analog-signal input pins)

The analog input pins AD7 (Ch.7) through AD0 (Ch.0) are shared with input port pins K67 through K60. Therefore, when these pins are used for analog input, they must be set for use with the A/D converter in the software. This setting can be made individually for each pin. At cold start, all these pins are set for input ports. The analog input voltage AVIN can be input in the range of AVRL ≤ AVIN ≤ AVRH.

#ADTRG (external-trigger input pin)

This pin is used to input a trigger signal to start A/D conversion from an external source. Since this pin is shared with input port K52, it must be set for use with the A/D converter in the software before an external trigger can be applied to the pin. At cold start, this pin is set for an input port.

Method for setting A/D-converter input pins

At cold start, the #ADTRG and AD[7:0] pins all are set for input ports Kxx (function select bit CFKxx = "0"). When using these pins for the A/D converter, write "1" to the function select bit CFKxx. Since the function select register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) that accompany a read-modify-write operation cannot be used to rewrite it. Use ordinary storage instructions for this purpose. At hot start, these pins retain their state from prior to the reset.

Pull-up resistors of input pins

Since the #ADTRG and AD[7:0] pins are shared with input ports, they contain a pull-up resistor. Whether this pull-up resistor is to be used can be set for each pin through the use of the pull-up control register.

K52 (#ADTRG) pull-up control: KPU52 (D2) / K5 pull-up control register (0x402C4)
K67–K60 (AD7–AD0) pull-up control:KPU6[7:0] (D[7:0]) / K5 pull-up control register (0x402C4)

At cold start, the pull-up control register is set to "0", so the pull-up resistor is disconnected from the input line. If the #ADTRG pin is to be pulled up, write "1" to KPU52. Since the pull-up control register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose. At hot start, this register retain its state from prior to the reset.

Disconnect the internal pull-up resistors of the analog input pins (KPU6x = "0"), as the presence of pull-up resistors affects conversion accuracy.

13.1.3 Setting A/D Converter

When the A/D converter is used, the following settings must be made before an A/D conversion can be performed:

1. Setting analog input pins
2. Setting the input clock
3. Selecting the analog-conversion start and end channels
4. Setting the A/D conversion mode
5. Selecting a trigger
6. Setting the sampling time
7. Setting interrupt/IDMA

The following describes how to set each item. For details on how to set the analog input pins, refer to the preceding section. For details on how to set interrupt/IDMA, refer to Section 13.1.5, "A/D Converter Interrupt and DMA".

Note: Before making these settings, make sure the A/D converter is disabled (ADE (D2) / A/D enable register (0x40244) = "0"). Changing the settings while the A/D converter is enabled could cause a malfunction.

Setting the input clock

As explained in Section 10.2, "Prescaler and Operating Clock for Peripheral Circuits", the A/D conversion clock can be selected from among the eight types shown in Table 13.1.2 below. Use PSAD[2:0] (D[2:0]) / A/D clock control register (0x4014F) for this selection.

Table 13.1.2 Input Clock Selection

PSAD2	PSAD1	PSAD0	Division ratio
1	1	1	f_{OSC3}/256
1	1	0	f_{OSC3}/128
1	0	1	f_{OSC3}/64
1	0	0	f_{OSC3}/32
0	1	1	f_{OSC3}/16
0	1	0	f_{OSC3}/8
0	0	1	f_{OSC3}/4
0	0	0	f_{OSC3}/2

f_{OSC3}: OSC3 oscillation frequency

The selected clock is output from the prescaler to the A/D converter by writing "1" to PSONAD (D3) / A/D clock control register (0x4014F).

Notes: • The A/D converter operates only when the prescaler is operating. The prescaler generates a clock for the A/D converter from the OSC3 oscillation clock by dividing it. If the CPU is operating using the low-speed (OSC1) clock, the prescaler is inactive, so the A/D converter cannot be used. (Refer to Section 10.2, "Prescaler and Operating Clock for Peripheral Circuits".)

- The value set in the prescaler is the division ratio of the OSC3 oscillation frequency. If the CPU operating clock is generated from the OSC3 oscillation clock by dividing it using CLKDT[1:0] (D[7:6]) / Power control register (0x40140), do not use a clock that is faster than the CPU operating clock.
- The recommended input clock frequency is a maximum of 2 MHz.
- Do not start an A/D conversion when the clock output from the prescaler to the A/D converter is turned off, and do not turn off the prescaler's clock output when an A/D conversion is underway. This could cause the A/D converter to operate erratically.

Selecting analog-conversion start and end channels

Select the channel in which the A/D conversion is to be performed from among the pins (channels) that have been set for analog input. To enable A/D conversions in multiple channels to be performed successively through one convert operation, specify the conversion start and conversion end channels.

Conversion start channel: CS[2:0] (D[2:0]) / A/D channel register (0x40243)
Conversion end channel: CE[2:0] (D[5:3]) / A/D channel register (0x40243)

Table 13.1.3 Relationship between CS/CE and Input Channel

CS2/CE2	CS1/CE1	CS0/CE0	Channel selected
1	1	1	AD7
1	1	0	AD6
1	0	1	AD5
1	0	0	AD4
0	1	1	AD3
0	1	0	AD2
0	0	1	AD1
0	0	0	AD0

Example: Operation of one A/D conversion
CS[2:0] = "0", CE[2:0] = "0": Converted only in AD0
CS[2:0] = "0", CE[2:0] = "3": Converted in the following order: AD0→AD1→AD2→AD3
CS[2:0] = "5", CE[2:0] = "1": Converted in the following order: AD5→AD6→AD7→AD0→AD1

Note: Only conversion-channel input pins that have been set for use with the A/D converter can be set using the CS and CE bits.

Setting the A/D conversion mode

The A/D converter can operate in one of the following two modes. This operation mode is selected using MS (D5) / A/D trigger register (0x40242).

1. Normal mode (MS = "0")

   All inputs in the range of channels set using the CS and CE bits are A/D converted once and then stopped.

2. Continuous mode (MS = "1")

   A/D conversions in the range of channels set using the CS and CE bits are executed successively until stopped by the software.

At initial reset, the normal mode is selected.

Selecting a trigger

Use TS[1:0] (D[4:3]) / A/D trigger register (0x40242) to select a trigger to start A/D conversion from among the four types shown in Table 13.1.4.

Table 13.1.4 Trigger Selection

TS1	TS0	Trigger
1	1	External trigger (K52/#ADTRG)
1	0	8-bit programmable timer 0
0	1	16-bit programmable timer 0
0	0	Software

1. External trigger

The signal input to the #ADTRG pin is used as a trigger.

When this trigger is used, the K52 pin must be set for #ADTRG in advance by writing "1" to CFK52 (D2) / K5 function select register (0x402C0).

A/D conversion is started at a falling edge of the #ADTRG signal.

2. Programmable timer

The underflow signal of 8-bit programmable timer 0 or 16-bit programmable timer 0 (16-bit mode) is used as a trigger. Since the underflow cycle can be programmed using each timer, this trigger is effective when cyclic A/D conversions are required.

For details on how to set a timer, refer to the explanation of each programmable timer in this manual.

3. Software trigger

Writing "1" to ADST (D1) / A/D enable register (0x40244) in the software serves as a trigger to start A/D conversion.

Setting the sampling time

The A/D converter used in the E0C33A104 contains ST[1:0] (D[1:0]) / A/D sampling register (0x402454) that allows the analog-signal input sampling time to be set in four steps (3, 5, 7, or 9 times the input clock period).

However, this register should be used as set by default (ST = "11"; x9 clock periods).

13.1.4 Control and Operation of A/D Conversion

Figure 13.1.2 shows the operation of the A/D converter.

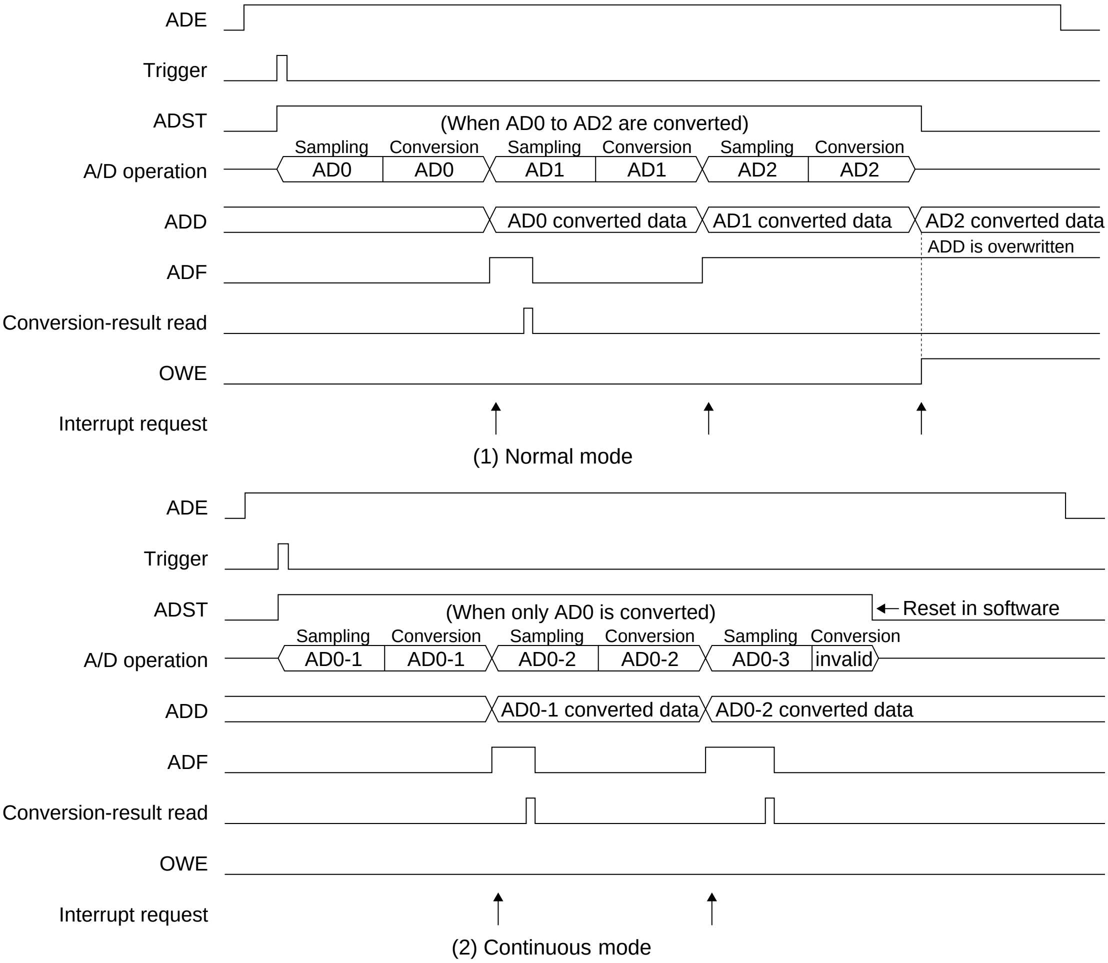


Figure 13.1.2 Operation of A/D Converter

Starting up the A/D converter circuit

After the settings specified in the preceding section have been made, write "1" to ADE (D2) / A/D enable register (0x40244) to enable the A/D converter. The A/D converter is thereby readied to accept a trigger to start A/D conversion. To set the A/D converter again, or if it is not be used, set ADE to "0".

Starting A/D conversion

When a trigger is input while ADE = "1", A/D conversion is started. If a software trigger has been selected, A/D conversion is started by writing "1" to ADST (D1) / A/D enable register (0x40244).
Only the trigger selected using TS[1:0] (D[4:3]) / A/D trigger register (0x40242) are valid; no other trigger is accepted.

When a trigger is input, the A/D converter samples and A/D-converts the analog input signal, beginning with the conversion start channel selected by CS[2:0].
Upon completion of the A/D conversion in that channel, the A/D converter stores the conversion result, in 10-bit data registers ADD[9:0] (ADD[9:8] = D[1:0]/0x40241, ADD[7:0] = D[7:0]/0x40240), and sets the conversion-complete flag ADF (D3) / A/D enable register (0x40244) and interrupt factor flag FADE (D0) / Clock timer, A/D interrupt factor flag register (0x40287). If multiple channels are specified using CS[2:0] and CE[2:0], A/D conversions in the subsequent channels are performed in succession.

The ADST used for the software trigger is set to "1" during A/D conversion, even when it is started by some other trigger, so it can be used as an A/D-conversion status bit.
The channel in which conversion is underway can be identified by reading CH[2:0] (D[2:0]) / A/D trigger register (0x40242).

Reading out A/D conversion results

As explained earlier, the results of A/D conversion are stored in the ADD[9:0] register each time conversion in one channel is completed. Since an interrupt can be generated simultaneously, this interrupt is normally used to read out the converted data. In addition, be sure to reset the interrupt factor flag (by writing "0") to prepare the A/D converter for the next operation.
Since the interrupt factor of the A/D converter can also be used to invoke IDMA, the conversion results can automatically be transferred to a specified memory location.

If multiple A/D conversion channels are specified, the conversion results in one channel must be read out prior to completion of conversion in the next channel. If the A/D conversion currently under way is completed before the previous conversion results are read out, the ADD[9:0] register is overwritten with the new conversion results.
If ADD[9:0] is updated when the conversion-complete flag ADF = "1" (before the converted data is read out), the overwrite-error flag OWE (D0) / A/D enable register (0x40244) is set to "1". The conversion-complete flag ADF is reset to "0" when the converted data is read out. If ADD[9:0] is updated when ADF = "0", OWE remains at "0", indicating that the operation has been completed normally. When reading out data, also read the OWE flag also to make sure the data is valid. Once OWE is set, it remains set until it is reset to "0" in the software. Note also that if OWE is set, ADF also is set. In this case, read out the converted data and reset ADF.

Terminating A/D conversion

- **For normal mode (MS = "1")**
  In the normal mode, A/D conversion is performed successively from the conversion start channel specified using CS[2:0] to the conversion end channel specified using CE[2:0], and is completed after these conversions are executed in one operation. ADST is reset to "0" upon completion of the conversion.

- **For continuous mode (MS = "0")**
  In the continuous mode, A/D conversion from the conversion-start to the conversion-end channels is executed repeatedly, without being stopped in the hardware. To terminate conversion, therefore, ADST must be reset to "0" in the software. The A/D conversion being executed when ADST is reset to "0" is forcibly stopped. Therefore, the results of this conversion cannot be obtained.

- **Forced termination**
  In either normal or continuous mode, A/D conversion is immediately terminated by writing "0" to ADST. The results of the conversion then under-way cannot be obtained.
  In addition, ADST is reset to "0" by writing "0" to ADE, so the conversion under-way is terminated.

13.1.5 A/D Converter Interrupt and DMA

Upon completion of A/D conversion in each channel, the A/D converter generates an interrupt and invokes the IDMA if necessary.

Control registers of the interrupt controller

The following shows the interrupt control registers available for the A/D converter:

Interrupt factor flag:	FADE (D0) / Clock timer, A/D interrupt factor flag (0x40287)
Interrupt enable:	EADE (D0) / Clock timer, A/D interrupt enable register (0x40277)
Interrupt level:	PADE[2:0] (D[6:4]) / Serial I/F Ch.1, A/D interrupt priority register (0x40264)
IDMA request:	RADE (D2)/ Serial I/F Ch.1, A/D IDMA request register (0x40293)

The A/D converter sets the interrupt factor flag to "1" when A/D conversion in one channel is completed, and the conversion results are stored in the ADD register. At this time, if the interrupt enable register bit has been set to "1", an interrupt request is generated.

Interrupts can be disabled by leaving the interrupt enable register bit set to "0". The interrupt factor flag is set to "1" upon completion of A/D conversion in each channel, regardless of the setting of the interrupt enable register (even when it is set to "0").

The interrupt priority register sets the priority level (0 to 7) of an interrupt. An interrupt request to the CPU is accepted no other interrupt request of a higher priority has been generated.

In addition, it is only when the PSR's IE bit = "1" (interrupts enabled) and the set value of the IL is smaller than the A/D-converter interrupt level set by the interrupt priority register, that the A/D converter's interrupt request is actually accepted by the CPU.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, refer to Chapter 8, "Interrupt".

Intelligent DMA

The A/D converter can invoke the intelligent DMA (IDMA) through the use of its interrupt factor. This allows the conversion results to be transferred to a specified memory location with no need to execute an interrupt processing routine.

The IDMA channel number assigned to the A/D converter is 0x1B.

Before IDMA can be invoked, the IDMA request bit must be set to "1". Transfer conditions on the IDMA side must also be set in advance.

If an interrupt factor occurs when the IDMA request bit is set to "1", IDMA is invoked. No interrupt request is generated at that point. An interrupt request is generated upon completion of the DMA transfer. Otherwise, the bits can be set so as not to generate an interrupt, with only a DMA transfer performed.

For details on DMA transfers and how to control interrupts upon completion of a DMA transfer, refer to Section 9.2, "Intelligent DMA".

Trap vector

The A/D converter's interrupt trap-vector default address is set as follows:

BTA3 = low: 0x0C00100

The base address of the trap table can be changed using the TTBR register (0x48134 to 0x48137).

13.1.6 I/O Memory of A/D Converter

Table 13.1.5 shows the control bits of the A/D converter.

For details on the I/O memory of the prescaler used to set clocks, refer to Section 10.2.4, "I/O Memory of Prescaler". For details on the I/O memory of the programmable timers used for a trigger, refer to Section 11.3.7, "I/O Memory of 8-Bit Programmable Timers" or Section 11.4.8, "I/O Memory of 16-Bit Programmable Timers".

Table 13.1.5 Control Bits of A/D Converter

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
A/D conversion result (low-order) register	**0040240 (B)**	D7	**ADD7**	A/D converted data	0x0 to 0x3FF				0	R	
		D6	**ADD6**	(low-order 8 bits)					0		
		D5	**ADD5**	ADD0 = LSB					0		
		D4	**ADD4**						0		
		D3	**ADD3**						0		
		D2	**ADD2**						0		
		D1	**ADD1**						0		
		D0	**ADD0**						0		
A/D conversion result (high-order) register	**0040241 (B)**	D7–2	**–**	–					–	–	0 when being read.
		D1	**ADD9**	A/D converted data					0	R	
		D0	**ADD8**	(high-order 2 bits) ADD9 = MSB					0		
A/D trigger register	**0040242 (B)**	D7–6	**–**	–	–				–	–	0 when being read.
		D5	**MS**	A/D conversion mode selection	1	Continuous	0	Normal	0	R/W	
		D4	**TS1**	A/D conversion trigger selection	TS[1:0]			Trigger	0	R/W	
		D3	**TS0**		1	1		#ADTRG pin	0		
					1	0		8-bit timer 0			
					0	1		16-bit timer 0			
					0	0		Software			
		D2	**CH2**	A/D conversion channel status	CH[2:0]			Channel	0	R	
		D1	**CH1**		1	1	1	AD7	0		
		D0	**CH0**		1	1	0	AD6	0		
					1	0	1	AD5			
					1	0	0	AD4			
					0	1	1	AD3			
					0	1	0	AD2			
					0	0	1	AD1			
					0	0	0	AD0			
A/D channel register	**0040243 (B)**	D7–6	**–**	–	–				–	–	0 when being read.
		D5	**CE2**	A/D converter	CE[2:0]			End channel	0	R/W	
		D4	**CE1**	end channel selection	1	1	1	AD7	0		
		D3	**CE0**		1	1	0	AD6	0		
					1	0	1	AD5			
					1	0	0	AD4			
					0	1	1	AD3			
					0	1	0	AD2			
					0	0	1	AD1			
					0	0	0	AD0			
		D2	**CS2**	A/D converter	CS[2:0]			Start channel	0	R/W	
		D1	**CS1**	start channel selection	1	1	1	AD7	0		
		D0	**CS0**		1	1	0	AD6	0		
					1	0	1	AD5			
					1	0	0	AD4			
					0	1	1	AD3			
					0	1	0	AD2			
					0	0	1	AD1			
					0	0	0	AD0			
A/D enable register	**0040244 (B)**	D7–4	**–**	–	–				–	–	0 when being read.
		D3	**ADF**	Conversion-complete flag	1	Completed	0	Run/Standby	0	R	Reset when ADD is read.
		D2	**ADE**	A/D enable	1	Enabled	0	Disabled	0	R/W	
		D1	**ADST**	A/D conversion control/status	1	Start/Run	0	Stop	0	R/W	
		D0	**OWE**	Overwrite error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
A/D sampling register	**0040245 (B)**	D7–2	**–**	–	–				–	–	0 when being read.
		D1	**ST1**	Input signal sampling time setup	ST[1:0]			Sampring time	1	R/W	Use with 9 clocks.
		D0	**ST0**		1	1		9 clocks	1		
					1	0		7 clocks			
					0	1		5 clocks			
					0	0		3 clocks			

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Serial I/F Ch.1, A/D interrupt priority register	**0040264 (B)**	D7	**–**	reserved	–				–	–	0 when being read.
		D6	**PAD2**	A/D converter interrupt level	0 to 7				X	R/W	
		D5	**PAD1**						X		
		D4	**PAD0**						X		
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**PSI012**	Serial interface Ch.1 interrupt level	0 to 7				X	R/W	
		D1	**PSI011**						X		
		D0	**PSI010**						X		
Clock timer, A/D interrupt enable register	**0040277 (B)**	D7–2	**–**	reserved	–				–	–	Writing 0 not allowed.
		D1	**ECTM**	Clock timer	1	Enabled	0	Disabled	0	R/W	
		D0	**EADE**	A/D converter					0	R/W	
Clock timer, A/D interrupt factor flag register	**0040287 (B)**	D7–2	**–**	reserved	–				–	–	0 when being read.
		D1	**FCTM**	Clock timer	1	Factor is generated	0	No factor is generated	X	R/W	
		D0	**FADE**	A/D converter					X	R/W	
K5 function select register	**00402C0 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**CFK54**	K54 function selection	1	DA1	0	K54	0	W	Undefined in read.
		D3	**CFK53**	K53 function selection	1	DA0	0	K53	0	W	
		D2	**CFK52**	K52 function selection	1	#ADTRG	0	K52	0	W	
		D1	**CFK51**	K51 function selection	1	#DMAREQ1	0	K51	0	W	
		D0	**CFK50**	K50 function selection	1	#DMAREQ0	0	K50	0	W	
K5 pull-up control register	**00402C4 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**KPU54**	K54 pull-up control	1	Pulled up	0	No pull-up	0	W	Undefined in read.
		D3	**KPU53**	K53 pull-up control					0	W	
		D2	**KPU52**	K52 pull-up control					0	W	
		D1	**KPU51**	K51 pull-up control					0	W	
		D0	**KPU50**	K50 pull-up control					0	W	
K6 function select register	**00402C5 (B)**	D7	**CFK67**	K67 function selection	1	AD7	0	K67	0	W	Undefined in read.
		D6	**CFK66**	K66 function selection	1	AD6	0	K66	0	W	
		D5	**CFK65**	K65 function selection	1	AD5	0	K65	0	W	
		D4	**CFK64**	K64 function selection	1	AD4	0	K64	0	W	
		D3	**CFK63**	K63 function selection	1	AD3	0	K63	0	W	
		D2	**CFK62**	K62 function selection	1	AD2	0	K62	0	W	
		D1	**CFK61**	K61 function selection	1	AD1	0	K61	0	W	
		D0	**CFK60**	K60 function selection	1	AD0	0	K60	0	W	
K6 pull-up control register	**00402C9 (B)**	D7	**KPU67**	K67 pull-up control	1	Pulled up	0	No pull-up	0	W	Undefined in read.
		D6	**KPU66**	K66 pull-up control					0	W	
		D5	**KPU65**	K65 pull-up control					0	W	
		D4	**KPU64**	K64 pull-up control					0	W	
		D3	**KPU63**	K63 pull-up control					0	W	
		D2	**KPU62**	K62 pull-up control					0	W	
		D1	**KPU61**	K61 pull-up control					0	W	
		D0	**KPU60**	K60 pull-up control					0	W	

CFK52: K52 pin function selection (D2) / K5 function select register (0x402C0)
CFK67–CFK60: K6[7:0] pin function selection (D[7:0]) / K6 function select register (0x402C5)

Selects the pins used by the A/D converter.

Write "1": A/D converter
Write "0": Input port
Read: Invalid

When an external trigger is used, write "1" to CFK52 to set the K52 pin for external trigger input #ADTRG. Select the pin used for analog input from among K60 (AD0) through K67 (AD7) by writing "1" to CFK60 through CFK67. If the function select bit for a pin is set to "0", the pin is set for an input port.

At cold start, CFK is set to "0" (input port). At hot start, CFK retains its state from prior to the initial reset.

KPU52: K52 pull-up control (D2) / K5 pull-up control register (0x402C4)
KPU67–KPU60: K6[7:0] pull-up control (D[7:0]) / K6 pull-up control register (0x402C9)

Controls the pull-up of each input pin.

Write "1": Pulled up
Write "0": Not pulled up
Read: Invalid

When a bit of the KPU register is set to "1", the internal pull-up resistor of the corresponding pin is enabled, so the pin is pulled up for VDD. The pins for which register bits are set to "0" are not pulled up.
Regarding the pins used for analog input, set the corresponding bits to "0" (no pull-up), as the presence of pull-up resistors affects conversion accuracy.
Since the KPU register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.
At cold start, KPU is set to "0" (no pull-up). At hot start, KPU retains its state from prior to the initial reset.

ADD9–ADD0: A/D converted data (D[1:0]) / A/D conversion result (high-order) register (0x40241) (D[7:0]) / A/D conversion result (low-order) register (0x40240)

Stores the results of A/D conversion.
The LSB is stored in ADD0, and the MSB is stored in ADD9. ADD0 and ADD1 are mapped to bits D0 and D1 at the address 0x40241, but bits D2 through D7 are always 0 when read.
This is a read-only register, so writing to this register is ignored.
At initial reset, the data in this register is cleared to "0".

MS: A/D conversion mode selection (D5) / A/D trigger register (0x40242)

Selects an A/D conversion mode.

Write "1": Continuous mode
Write "0": Normal mode
Read: Valid

The A/D converter is set for the continuous mode by writing "1" to MS. In this mode, A/D conversions in the range of the channels selected using CS and CE are executed continuously until stopped in the software.
When MS = "0", the A/D converter operates in the normal mode. In this mode, A/D conversion is completed after all inputs in the range of the channels selected by CS and CE are converted in one operation.
At initial reset, MS is set to "0" (normal mode).

TS1–TS0: Trigger selection (D[4:3]) / A/D trigger register (0X40242)

Selects a trigger to start A/D conversion.

Table 13.1.6 Trigger Selection

TS1	TS0	Trigger
1	1	External trigger (K52/#ADTRG)
1	0	8-bit programmable timer 0
0	1	16-bit programmable timer 0
0	0	Software

When an external trigger is used, use the CFK52 bit to set the K52 pin for #ADTRG.
When a programmable timer is used, since its underflow signal serves as a trigger, set the underflow cycle and other parameters for the programmable timer.
At initial reset, TS is set to "0" (software trigger).

CH2–CH0: Conversion channel status (D[2:0]) / A/D trigger register (0X40242)

Indicates the channel number (0 to 7) currently being A/D-converted.
When A/D conversion is performed in multiple channels, read this bit to identify the channel in which conversion is underway.
At initial reset, CH is set to "0" (AD0).

CE2–CE0: Conversion end-channel setup (D[5:3]) / A/D channel register (0x40243)

Sets the conversion end channel by selecting a channel number from 0 to 7.
Analog inputs can be A/D-converted successively from the channel set using CS to the channel set using this bit in one operation. If only one channel is to be A/D converted, set the same channel number in both the CS and CE bits.
At initial reset, CE is set to "0" (AD0).

CS2–CS0: Conversion start-channel setup (D[2:0]) / A/D channel register (0x40243)

Sets the conversion start channel by selecting a channel number from 0 to 7.
Analog inputs can be A/D-converted successively from the channel set using this bit to the channel set using CE in one operation. If only one channel is to be A/D converted, set the same channel number in both the CS and CE bits.
At initial reset, CS is set to "0" (AD0).

ADF: Conversion-complete flag (D3) / A/D enable register (0x40244)

Indicates that A/D conversion has been completed.

Read "1":	Conversion completed
Read "0":	Being converted or standing by
Write:	Invalid

This flag is set to "1" when A/D conversion is completed, and the converted data is stored in the data register and is reset to "0" when the converted data is read out. When A/D conversion is performed in multiple channels, if the next A/D conversion is completed while ADF = "1" (before the converted data is read out), the data register is overwritten with the new conversion results, causing an overrun error to occur. Therefore, ADF must be reset by reading out the converted data before the next A/D conversion is completed.
At initial reset, ADF is set to "0" (being converted or standing by).

ADE: A/D enable (D2) / A/D enable register (0x40244)

Enables the A/D converter (readied for conversion).

Write "1":	Enabled
Write "0":	Disabled
Read:	Valid

When ADE is set to "1", the A/D converter is enabled, meaning it is ready to start A/D conversion (i.e., ready to accept a trigger). When ADE = "0", the A/D converter is disabled, meaning it is unable to accept a trigger.
Before setting the conversion mode, start/end channels, etc. for the A/D converter, be sure to reset ADE to "0". This helps to prevent the A/D converter from operating erratically.
At initial reset, ADE is set to "0" (disabled).

ADST: A/D conversion control/status (D1) / A/D enable register (0x40244)

Controls A/D conversion.

Write "1":	Software trigger
Write "0":	A/D conversion is stopped
Read:	Valid

If A/D conversion is to be started by a software trigger, set ADST to "1". If any other trigger is used, ADST is automatically set to "1" by the hardware.
ADST remains set while A/D conversion is underway.
In normal mode, upon completion of A/D conversion in selected channels, ADST is reset to "0" and the A/D conversion circuit is turned off. To stop A/D conversion during operation in continuous mode or forcibly terminate A/D conversion, reset ADST by writing "0".
When ADE = "0" (A/D conversion disabled), ADST is fixed to "0", with no trigger accepted. In addition, ADST is reset to "0" when ADE is reset by writing "0" during A/D conversion.
At initial reset, ADST is set to "0" (A/D conversion stopped).

OWE: Overwrite-error flag (D0) / A/D enable register (0x40244)

Indicates that the converted data has been overwritten.

Read "1":	Overwritten
Read "0":	Normal
Write "1":	Invalid
Write "0":	Flag is set

During A/D conversion in multiple channels, if the conversion results for the next channel are written to the converted-data register (overwritten) before the converted data is read out to reset the conversion-complete flag ADF that has been set through conversion of the preceding channel, OWE is set to "1". When ADF is reset, because this means that the converted data has been read out, OWE is not set.
Once OWE is set to "1", it remains set until it is reset by writing "0" in the software.
At initial reset, OWE is set to "0" (normal).

ST1–ST0: Sampling-time setup (D[1:0]) / A/D sampling register (0x40245)

Sets the analog input sampling time.

Table 13.1.7 Sampling Time

ST1	ST0	Sampling Time
1	1	9-clock period
1	0	7-clock period
0	1	5-clock period
0	0	3-clock period

The A/D converter input clock is used for counting.

At initial reset, ST is set to "11" (9-clock period).
To maintain the conversion accuracy, use ST as set by default (9-clock period).

PAD2–PAD0: A/D converter interrupt level (D[6:4]) / Serial I/F Ch.1, A/D interrupt priority register (0x40264)

Sets the priority level of the A/D-converter interrupt in the range of 0 to 7.
At initial reset, PAD becomes indeterminate.

EADE: A/D converter interrupt enable (D0) / Clock timer, A/D interrupt enable register (0x40277)

Enables or disables an interrupt to the CPU generated by the A/D converter.

Write "1":	Interrupt enabled
Write "0":	Interrupt disabled
Read:	Valid

EADE is an interrupt enable bit to control the A/D converter interrupt.
When EADE is set to "1", the A/D converter interrupt is enabled. When EADE is set to "0", the A/D-converter interrupt is disabled.
At initial reset, EADE is set to "0" (interrupt disabled).

FADE: A/D converter interrupt factor flag (D0) / Clock timer, A/D interrupt factor flag register (0x40287)

Indicates the status of an A/D-converter interrupt factor generated.

When read

Read "1":	Interrupt factor has occurred
Read "0":	No interrupt factor has occurred

When written using the reset-only method (default)

Write "1":	Interrupt factor flag is reset
Write "0":	Invalid

When written using the read/write method

Write "1":	Interrupt flag is set
Write "0":	Interrupt flag is reset

FADE is the interrupt factor flag of the A/D converter. It is set to "1" upon completion of A/D conversion in one channel (i.e., when the conversion results are written into the ADD register).
At this time, if the following conditions are met, an interrupt to the CPU is generated:
1. The corresponding interrupt enable register bit is set to "1".
2. No other interrupt request of a higher priority has been generated.
3. The IE bit of the PSR is set to "1" (interrupts enabled).
4. The value set in the corresponding interrupt priority register is higher than the interrupt level (IL) of the CPU.
When using the interrupt factor of the A/D converter to request IDMA, note that even when the above conditions are met, no interrupt request to the CPU is generated for the interrupt factor that has occurred. If interrupts are enabled at the setting of IDMA, an interrupt is generated under the above conditions after the data transfer by IDMA is completed.

The interrupt factor flag is set to "1" whenever interrupt generation conditions are met, regardless of how the interrupt enable and interrupt priority registers are set.
If the next interrupt is to be accepted after an interrupt has occurred, it is necessary that the interrupt factor flag be reset, and that the PSR be set again (by setting the IE bit to "1" after setting the IL to a value lower than the level indicated by the interrupt priority register, or by executing the reti instruction).
The interrupt factor flag can be reset only by writing to it in the software. Note that if the PSR is set again to accept interrupts generated (or if the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt occurs again. Note also that the value to be written to reset the flag is "1" when the reset-only method (RSTONLY = "1") is used, and "0" when the read/write method (RSTONLY = "0") is used.
At initial reset, the content of FADE becomes indeterminate, so be sure to reset it in the software.

RADE: A/D converter IDMA request (D2) / Serial I/F Ch.1, A/D IDMA request register (0x40293)

Specifies whether to invoke IDMA when an interrupt factor occurs.

Write "1": IDMA request
Write "0": Interrupt request
Read: Valid

When RADE is set to "1", IDMA is invoked when an interrupt factor occurs, thereby performing a programmed data transfer. If RADE is set to "0", normal interrupt processing is performed, without invoking IDMA.
For details on IDMA, refer to Section 9.2, "Intelligent DMA".
At initial reset, RADE becomes indeterminate, so be sure to initialize it in the software.

13.1.7 Programming Notes

(1) Before setting the conversion mode, start/end channels, etc. for the A/D converter, be sure to disable the A/D converter (ADE (D2) / A/D enable register (0x40244) = "0"). A change in settings while the A/D converter is enabled could cause it to operate erratically.

(2) The A/D converter operates only when the prescaler is operating. The prescaler generates a clock for the A/D converter from the OSC3 oscillation clock by dividing it. If the CPU is operating using the low-speed (OSC1) clock, the prescaler is inactive, so the A/D converter cannot be used. (Refer to Section 10.2, "Prescaler and Operating Clock for Peripheral Circuits".)
When the A/D converter registers are set up, the prescaler must be operating. Therefore, start the prescaler first and make sure the A/D converter is supplied with its operating clock before setting up the A/D converter registers.
The value set in the prescaler is a division ratio for the OSC3 oscillation frequency. If the CPU operating clock is generated from the OSC3 oscillation clock by dividing it using CLKDT[1:0] (D[7:6]) / Power control register (0x40140), do not use a clock that is faster than the CPU operating clock.
In consideration of the conversion accuracy, we recommends that the A/D converter operating clock be 2 MHz (max.).

(3) Do not start an A/D conversion when the clock supplied from the prescaler to the A/D converter is turned off, and do not turn off the prescaler's clock output when an A/D conversion is underway, as doing so could cause the A/D converter to operate erratically.

(4) Since the pin function select register (CFK) is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.

(5) After an initial reset, the interrupt factor flag (FADE) and IDMA request register (RADE) become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset this flag and register in a program.

(6) To prevent the regeneration of interrupts due to the same factor following the occurrence an interrupt, always be sure to reset the interrupt factor flag before setting the PSR again or executing the reti instruction.

13.2 D/A Converter

13.2.1 Features and Structure of D/A Converter

The E0C33A104 contains two channels of D/A converters that have the following features:

• Conversion method:	R-2R ladder resistor
• Resolution:	8 bits
• Output channel:	2 channels
• Settling time:	Maximum of 10 µs (when the load capacitance = 20 pF, load resistance = 2 kΩ; AVDD = 5 V + 0.5 V; AVSS = 0 V)
• Output voltage range:	0 V to AVDD (AVDD/256 steps)

Figure 13.2.1 shows the structure of the D/A converter.

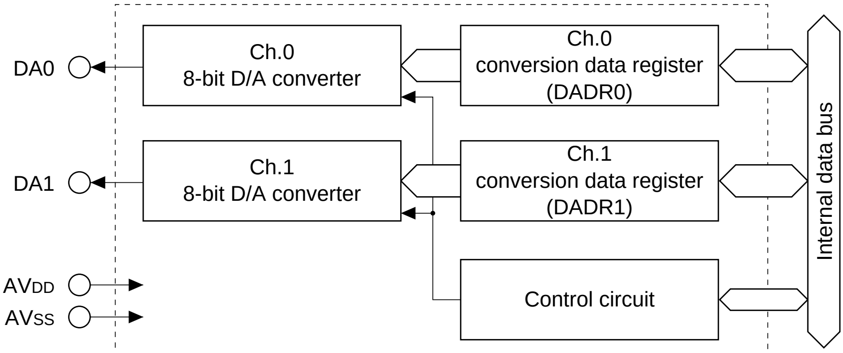


Figure 13.2.1 Structure of D/A Converter

13.2.2 Pin Configuration of D/A Converter

Table 13.2.1 shows the pins used by the D/A converter.

Table 13.2.1 Pin Configuration of D/A Converter

Pin name	Pin No.		I/O	Pull-up	Function	Function select bit
	QFP5-128	QFP15-128				
K53/DA0	27	24	I/O	Built-in	Input port / DA converter output 0	CFK53(D3)/K5 function select register(0x402C0)
K54/DA1	28	25	I/O	Built-in	Input port / DA converter output 1	CFK54(D4)/K5 function select register(0x402C0)
AVDD	29	26	–	–	Power supply for analog system (+)	–
AVSS	18	15	–	–	Power supply for analog system (-)	–

AVDD, AVSS (analog power-supply pins)

AVDD and AVSS are the power-supply pins for the analog circuit. The voltage levels supplied to these pins must be AVDD = VDD and AVSS = VSS. These power-supply pins are shared with the A/D converter.

DA0, DA1 (analog output pins)

The analog output pins DA0 (Ch.0) and DA1 (Ch.1) are shared with input-port pins K53 and K54. Therefore, when these pins are used for analog output, they must be set for use with the D/A converter through the software. This setting can be made for each pin individually. At cold start, these pins are both set for input ports.

Method for setting D/A-converter output pins

At cold start, the DA0 and DA1 pins are all set for input ports Kxx (function select bit CFKxx = "0"). When using these pins for the D/A converter, write "1" to the function select bit CFKxx. Only one or both of these pins can be used for the D/A converter. Since the function select register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) that accompany a read-modify-write operation cannot be used to rewrite it. Use ordinary storage instructions for this purpose.
At hot start, these pins retain their state from prior to the reset.

Pull-up resistor for input pins

Since the DA0 and DA1 pins are shared with input ports, they contain a pull-up resistor. Whether this pull-up resistor is to be used can be set for each pin through the use of a pull-up control register.
K53 (DA0) pull-up control: KPU53 (D3) / K5 pull-up control register (0x402C4)
K54 (DA1) pull-up control: KPU54 (D4) / K5 pull-up control register (0x402C4)

At cold start, the pull-up control register is set to "0", so the pull-up resistor is disconnected from the input line. Disconnect the internal pull-up resistors (KPU5x = "0") for the pins for analog output, as the presence of pull-up resistors affects conversion accuracy.

13.2.3 Control and Operation of D/A Conversion

Setting the D/A converter

When using the D/A converter, the following settings must be made before starting D/A conversion:

1. Setting analog output pins
2. Setting the operating voltage

Note: Before making these settings, be sure to disable the D/A converter (DAE0 (D1) and DAE1 (D0) / D/A output control register (0x40246) = "0"). Changing the settings while the D/A converter is enabled could cause it to operate erratically.

For details on how to set the analog output pins, refer to the preceding section of this manual. Note that the analog output pins are placed in the high-impedance state until D/A output begins.

The D/A converter must have its operating voltage set according to the power-supply voltage. Use SELV (D0) / D/A operating clock switch register (0x40247) to set this voltage.
When operating with 5 V: Write "1" to SELV.
When operating with 3.3 V: Write "0" to SELV.

At initial reset, SELV is set to "0" (3.3 V). This setting is common to both DA0 and DA1.

D/A-conversion data register

An 8-bit data register is provided for each channel to set the data to be D/A-converted. The conversion data must be written to the data register corresponding to each output channel.
DA0: DADR0[7:0] (D[7:0]) / DA0 output data register (0x40249)
DA1: DADR1[7:0] (D[7:0]) / DA1 output data register (0x40248)

At initial reset, both these data registers are set to "0x0".

Starting D/A-conversion output

To start D/A conversion for DA0, set DAE0 (D1) / D/A output control register (0x40246) to "1" after setting the D/A converter and writing data into the register as described above. Similarly, to start D/A conversion for DA1, set DAE1 (D0) / D/A output control register (0x40246) to "1". When DAEx is set in this way, the data set in the DADR register is D/A-converted and the conversion results are presented in the analog output pin. Note that after a D/A conversion is started by setting DAE, the analog output pin requires a finite settling time before its output stabilizes.
If DAEx is set to "0", the corresponding analog output pin is placed in the high-impedance state.

The voltage output by the D/A converter is expressed by the following equation:

$$\text{Output voltage} = \frac{\text{Content of DADR}}{256} \times AV_{DD}\ [V]$$

If new data is written to the DADR register while DAE remains set, the data thus written to the register is D/A-converted and the results are presented in the analog output pin. This output also requires a finite settling time before it stabilizes after data is written to the register.

Note: If D/A conversion for DA1(0) is performed while the D/A conversion results for DA0(1) are being output, the output level of DA0(1) may fluctuate.

Figure 13.2.2 shows a timing chart of D/A conversion.

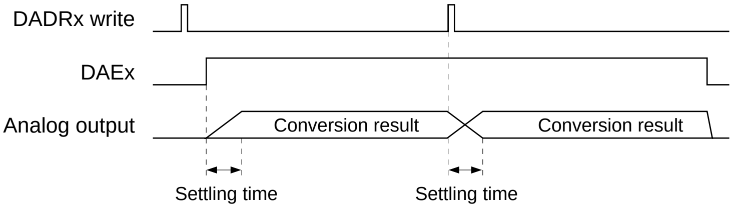


Figure 13.2.2 Timing chart of D/A Conversion

13.2.4 I/O Memory of D/A Converter

Table 13.2.2 shows the control bits of the D/A converter.

Table 13.2.2 Control Bits of D/A Converter

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
D/A output control register	**0040246 (B)**	D7–2	**–**	–	–				–	–	0 when being read.
		D1	**DAE0**	DA0 output control	1	Output	0	High-Z	0	R/W	
		D0	**DAE1**	DA1 output control	1	Output	0	High-Z	0	R/W	
D/A operating voltage switch register	**0040247 (B)**	D7–2	**–**	–	–				–	–	0 when being read.
		D1	**–**	–	–				X	–	Undefined in read.
		D0	**SELV**	D/A operating voltage switch	1	5 V	0	3.3 V	0	R/W	
DA1 output data register	**0040248 (B)**	D7	**DADR17**	DA1 output data	0 to 255				0	R/W	
		D6	**DADR16**	DADR17 = MSB					0		
		D5	**DADR15**	DADR10 = LSB					0		
		D4	**DADR14**						0		
		D3	**DADR13**						0		
		D2	**DADR12**						0		
		D1	**DADR11**						0		
		D0	**DADR10**						0		
DA0 output data register	**0040249 (B)**	D7	**DADR07**	DA0 output data	0 to 255				0	R/W	
		D6	**DADR06**	DADR07 = MSB					0		
		D5	**DADR05**	DADR00 = LSB					0		
		D4	**DADR04**						0		
		D3	**DADR03**						0		
		D2	**DADR02**						0		
		D1	**DADR01**						0		
		D0	**DADR00**						0		
K5 function select register	**00402C0 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**CFK54**	K54 function selection	1	DA1	0	K54	0	W	Undefined in read.
		D3	**CFK53**	K53 function selection	1	DA0	0	K53	0	W	
		D2	**CFK52**	K52 function selection	1	#ADTRG	0	K52	0	W	
		D1	**CFK51**	K51 function selection	1	#DMAREQ1	0	K51	0	W	
		D0	**CFK50**	K50 function selection	1	#DMAREQ0	0	K50	0	W	
K5 pull-up control register	**00402C4 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**KPU54**	K54 pull-up control	1	Pulled up	0	No pull-up	0	W	Undefined in read.
		D3	**KPU53**	K53 pull-up control					0	W	
		D2	**KPU52**	K52 pull-up control					0	W	
		D1	**KPU51**	K51 pull-up control					0	W	
		D0	**KPU50**	K50 pull-up control					0	W	

CFK53: K53 pin function selection (D3) / K5 function select register (0x402C0)
CFK54: K54 pin function selection (D4) / K5 function select register (0x402C0)

Selects the pins used by the D/A converter.

Write "1": D/A converter
Write "0": Input port
Read: Invalid

To use D/A converter Ch.0, write "1" to CFK53 to set the K53 pin for D/A conversion output, DA0 (K53). To use D/A converter Ch.1, write "1" to CFK54 to set the K54 pin for D/A conversion output, DA1 (K54). Only one or both of these pins can be set for D/A output.

If this bit is set to "0", the corresponding pin is set for an input port.

Before selecting the output pin here, make sure the DAE0 and DAE1 bits are set to "0".

At cold start, CFK is set to "0" (input port). At hot start, CFK retains its state from prior to the initial reset.

KPU53: K53 pull-up control (D3) / K5 pull-up control register (0x402C4)
KPU54: K54 pull-up control (D4) / K5 pull-up control register (0x402C4)

Controls the pull-up of each pin.

Write "1":	Pulled up
Write "0":	Not pulled up
Read:	Invalid

When a bit of the KPU register is set to "1", the internal pull-up resistor of the corresponding pin is enabled, so the pin is pulled up to VDD. The pins for which register bits are set to "0" are not pulled up.
Regarding the pins used for analog output, se the corresponding bits to "0" (no pull-up), as the presence of pull-up resistors affects conversion accuracy.
Since the KPU register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.
At cold start, KPU is set to "0" (no pull-up). At hot start, KPU retains its state from prior to the initial reset.

DAE0: DA0 output control (D1) / D/A output control register (0x40246)
DAE1: DA1 output control (D0) / D/A output control register (0x40246)

Controls the D/A conversion output.

Write "1":	D/A conversion output
Write "0":	High-impedance state (D/A conversion output disabled)
Read:	Valid

When DAE0(1) is set to "1", DA0(1) starts D/A-converting the data set in the DADR0(1) register and the conversion result is output from the DA0(1) pin. The conversion output requires a finite settling time until it stabilizes after the DAE bit is written to.
If DAE is set to "0", D/A conversion is disabled, with the corresponding pin placed in the high-impedance state.
At initial reset, DAE is set to "0" (D/A conversion output disabled).

SELV: D/A operating voltage switch (D0) / D/A operating voltage switch register (0x40247)

Sets the operating voltage of the D/A converter.

Write "1":	5 V
Write "0":	3.3 V
Read:	Valid

Set SELV to "1" (5 V) or "0" (3.3 V) according to the supply voltage used.
Before selecting this voltage, set the DAE0 and DAE1 bits to "0".
At initial reset, SELV is set to "0" (3.3 V).

DADR07–DADR00: DA0 conversion output data (D[7/0]) / DA0 conversion output data register (0x40249)
DADR17–DADR10: DA1 conversion output data (D[7/0]) / DA1 conversion output data register (0x40248)

Sets the data to be D/A-converted in each channel.
Use the DADR0 register to store the conversion data for DA0, and the DADR1 register to store the conversion data for DA1.
The data written to this register is converted into an analog value when DAE for that channel is set to "1", and the converted data is output from the corresponding analog output pin.
The analog voltage thus output is equal to AVDD × [set value of the DADR register]/256.
If the data in this register is rewritten when DAE = "1", the new data is also D/A-converted, with the result presented to the analog output pin. In this case too, the output requires a finite settling time until it stabilizes after data has been written to the register.
At initial reset, DADR is set to "0x00".

13.2.5 Programming Notes

(1) Before setting the D/A converter pins and operating voltage, be sure to disable the D/A converter (DAE0 (D1) and DAE1 (D0) / D/A output control register (0x40246) = "0"). A change in settings while the D/A converter is enabled could cause it to operate erratically.

(2) Since the pin function select register (CFK) is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.

14 Input/Output Ports

The E0C33A104 has a total of 39 input/output pins. Although each pin is used for input/output from/to the internal peripheral circuits, some pins can be used as general-purpose input/output ports unless they are used for the peripheral circuits. This chapter describes the functions of the pins used as general-purpose input/output ports and the means of controlling them.

14.1 Input Ports (K Ports)

14.1.1 Structure of Input Port

The E0C33A104 contains 13 bits of input ports (K50 to K54, K60 to K67).
Figure 14.1.1 shows the structure of a typical input port.

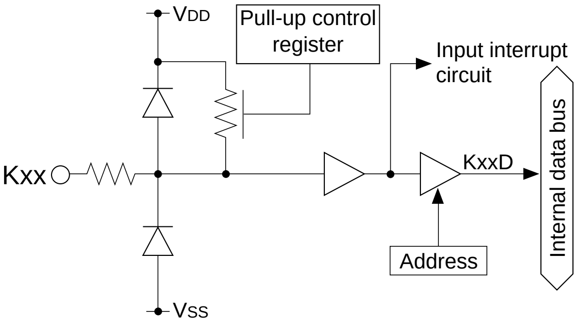


Figure 14.1.1 Structure of Input Port

Each input-port pin is connected directly to the data bus via a three-state buffer. The state of the input signal when read at an input port is directly taken into the internal circuit as data.

14.1.2 Input-Port Pins

The input pins concurrently serve as the input/output pins for peripheral circuits, as shown in Table 14.1.1. Whether they are used as input ports or for peripheral circuits can be set bit-for-bit using a function select register. All pins not used for peripheral circuits can be used as general-purpose input ports that have an interrupt function.

Table 14.1.1 Input Pins

Pin name	Pin No.		I/O	Pull-up	Function	Function select bit
	QFP5-128	QFP15-128				
K50/#DMAREQ0	58	55	I	Built-in	Input port / High-speed DMA request 0	CFK50(D0)/K5 function select register(0x402C0)
K51/#DMAREQ1	54	51	I	Built-in	Input port / High-speed DMA request 1	CFK51(D1)/K5 function select register(0x402C0)
K52/#ADTRG	35	32	I	Built-in	Input port / AD converter trigger	CFK52(D2)/K5 function select register(0x402C0)
K53/DA0	27	24	I(O)	Built-in	Input port / DA converter output 0	CFK53(D3)/K5 function select register(0x402C0)
K54/DA1	28	25	I(O)	Built-in	Input port / DA converter output 1	CFK54(D4)/K5 function select register(0x402C0)
K60/AD0	26	23	I	Built-in	Input port / AD converter input 0	CFK60(D0)/K6 function select register(0x402C5)
K61/AD1	25	22	I	Built-in	Input port / AD converter input 1	CFK61(D1)/K6 function select register(0x402C5)
K62/AD2	24	21	I	Built-in	Input port / AD converter input 2	CFK62(D2)/K6 function select register(0x402C5)
K63/AD3	23	20	I	Built-in	Input port / AD converter input 3	CFK63(D3)/K6 function select register(0x402C5)
K64/AD4	22	19	I	Built-in	Input port / AD converter input 4	CFK64(D4)/K6 function select register(0x402C5)

Pin name	Pin No.		I/O	Pull-up	Function	Function select bit
	QFP5-128	QFP15-128				
K65/AD5	21	18	I	Built-in	Input port / AD converter input 5	CFK65(D5)/K6 function select register(0x402C5)
K66/AD6	20	17	I	Built-in	Input port / AD converter input 6	CFK66(D6)/K6 function select register(0x402C5)
K67/AD7	19	16	I	Built-in	Input port / AD converter input 7	CFK67(D7)/K6 function select register(0x402C5)

At cold start, all pins are set for input ports Kxx (function select register CFKxx = "0"). When these pins are used for the internal peripheral circuits, write "1" to CFKxx. For details on pin functions in this case, refer to the description of each peripheral circuit in this manual. Since the function select register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) that accompany a read-modify-write operation cannot be used to rewrite it. Use ordinary storage instructions for this purpose.
At hot start, the pins retain their state from prior to the reset.
When the ports set for A/D converter input or D/A converter output are read, the value obtained is always "0".

14.1.3 Pull-Up Resistors

The input-port pins contain a pull-up resistor. Whether this pull-up resistor is to be used can be selected for each pin. Use a pull-up control register for this selection.
K54–K50 pull-up control: KPU5[4:0] (D[4:0]) / K5 pull-up control register (0x402C4)
K67–K60 pull-up control: KPU6[7:0] (D[7:0]) / K6 pull-up control register (0x402C9)

At cold start, the pull-up control register is set to "0", so the pull-up resistor is disconnected from the input line. If a pin is to be pulled up, set the bit (KPUxx) corresponding to the pin to "1". Since the pull-up control register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.
At hot start, the pins retain their state from prior to the reset.

The pins with a pull-up resistor (KPUxx = "1") are effective for inputs to push switches and the key matrix. However, when the state of the input pin changes from low to high, a time constant comprised of the pull-up resistor and the pin's load capacitance causes a delay in the rising transition of the waveform. For this reason, an appropriate wait time must be provided prior to taking the input port into the internal logic. This is especially important for key scanning in a key-matrix configuration. The wait time set here must be equal to or greater than the value calculated from the following equation:

Wait time = $R_{IN} \times (C_{IN}$ + load capacitance on the board) $\times$ 1.6 [sec]
R_{IN} = pull-up resistance (max.)
C_{IN} = pin capacitance (max.)

If the pull-up resistor is disconnected (KPUxx = "0"), the pins can be used effectively for slide-switch input or interfacing with other LSI. In this case, be careful not to cause a floating state in the inputs.

The unused input-port pins must have their pull-up resistors connected.

Note: The pull-up control register can also be used when the input pins are set for use with other peripheral circuits.
Regarding the pins set for A/D input or D/A output, leave their internal pull-up resistors disconnected while in use, as the pull-up resistors affect conversion accuracy.

14.1.4 Interrupt Function and Setting of Interrupt Conditions

All input ports have an interrupt function.

Input-port interrupts are classified into three groups: K5[4:0] and K6[3:0] (in units of four bits, with two interrupt factors), and K6[7:4] (in units of one bit, with four interrupt factors). Interrupts can be controlled for each group through the software.

Conditions for input-interrupt generation

(1) K5[4:0] and K6[3:0] input interrupts

Figure 14.1.2 shows the structure of K5[4:0] and K6[3:0] input-interrupt circuits.

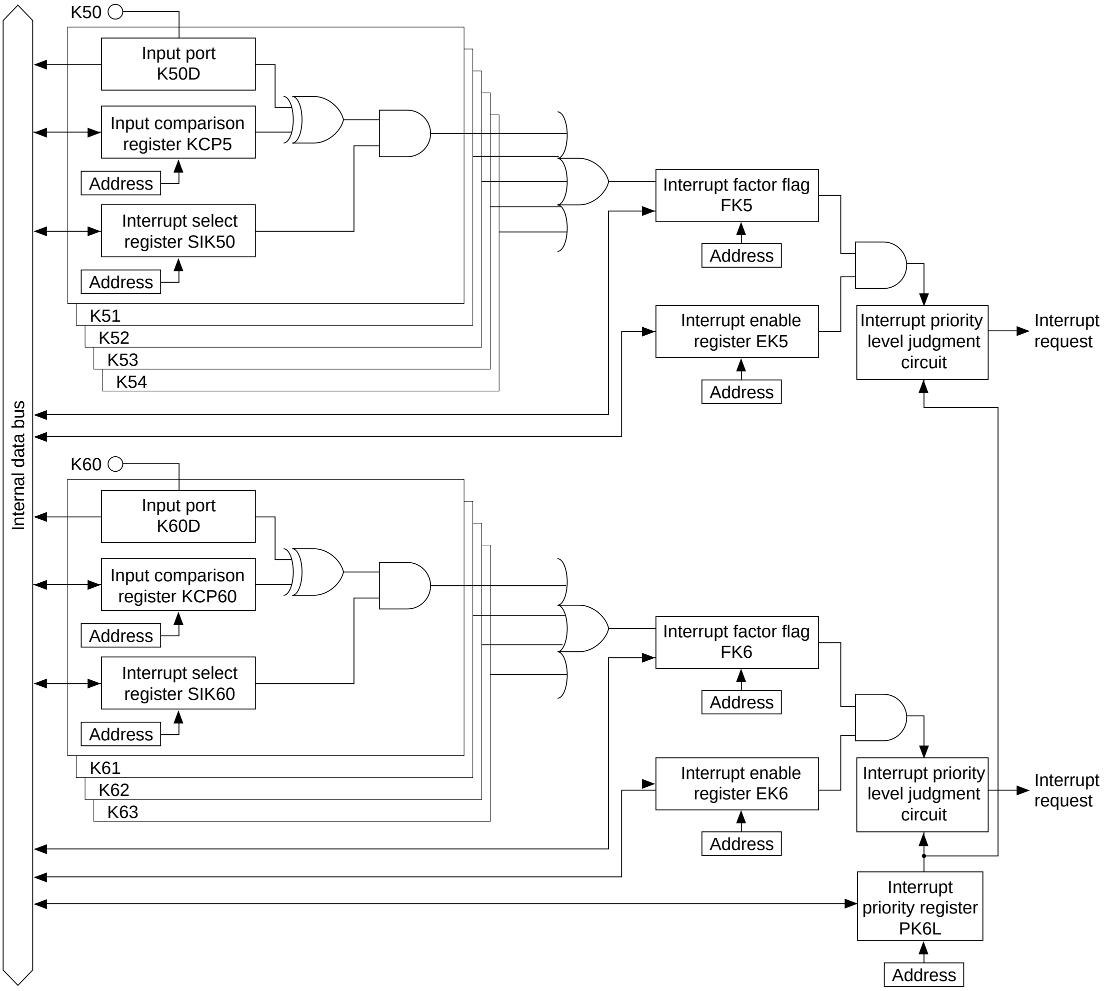

K50
Input port K50D
Input comparison register KCP5
Address
Interrupt select register SIK50
Address
K51
K52
K53
K54
Interrupt factor flag FK5
Address
Interrupt enable register EK5
Address
Interrupt priority level judgment circuit
Interrupt request
Internal data bus
K60
Input port K60D
Input comparison register KCP60
Address
Interrupt select register SIK60
Address
K61
K62
K63
Interrupt factor flag FK6
Address
Interrupt enable register EK6
Address
Interrupt priority level judgment circuit
Interrupt request
Interrupt priority register PK6L
Address

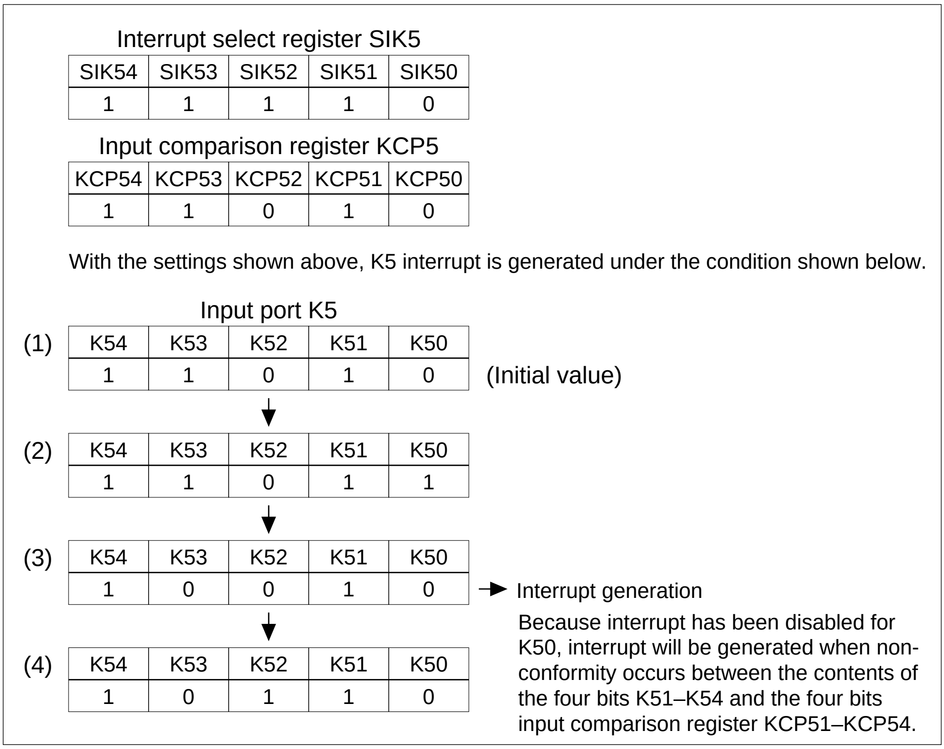
Figure 14.1.2 Structure of K5[4:0] and K6[3:0] Input-Interrupt Circuits

The K5[4:0] and K6[3:0] input-interrupt circuits have interrupt select registers SIK5[4:0] and SIK6[3:0] to set input-interrupt conditions, as well as input comparison registers KCP5[4:0] and KCP6[3:0].

The interrupt select register SIK is used to select the input pin to be used for an interrupt. This register masks each input pin, whereas the interrupt enable register of the interrupt controller masks the interrupt factor for each interrupt group.

The input comparison register KCP is used to select whether an interrupt for each input port is to be generated at the rising or falling edge of the input.

A change in state occurs so that the input pin enabled for interrupt by the interrupt select register SIK and the content of the input comparison register KCP become unmatched after being matched, the interrupt factor flag FK is set to "1" and, if other interrupt conditions are met, an interrupt is generated.
Figure 14.1.3 shows cases in which a K5[4:0] interrupt is generated. Here, it is assumed that the control register of the interrupt controller is set so as to enable generation of a K5[4:0] interrupt.



Interrupt select register SIK5

SIK54	SIK53	SIK52	SIK51	SIK50
1	1	1	1	0

Input comparison register KCP5

KCP54	KCP53	KCP52	KCP51	KCP50
1	1	0	1	0

With the settings shown above, K5 interrupt is generated under the condition shown below.

Input port K5

(1)

K54	K53	K52	K51	K50
1	1	0	1	0

(Initial value)

(2)

K54	K53	K52	K51	K50
1	1	0	1	1

(3)

K54	K53	K52	K51	K50
1	0	0	1	0

Interrupt generation

(4)

K54	K53	K52	K51	K50
1	0	1	1	0

Because interrupt has been disabled for K50, interrupt will be generated when non-conformity occurs between the contents of the four bits K51–K54 and the four bits input comparison register KCP51–KCP54.

Figure 14.1.3 K5[4:0] Interrupt Generation Example

Since K50 is disabled from interrupt by SIK50, no interrupt occurs at that point (2) above.
Next, because K53 becomes "0" at (3), an interrupt is generated due to the lack of a match between the data of the input pin K5[4:1] that is enabled for interrupt and that of the input comparison register KCP5[4:1].
Since only a change in states in which the input data and the content of the input comparison register KCP become unmatched after being matched constitutes an interrupt generation condition as described above, no interrupt is generated when a change in states from one unmatched state to another, as in (4), occurs.
Consequently, if another interrupt is to be generated again following the occurrence of an interrupt, the state of the input pin must be temporarily restored to the same content as that of the input comparison register KCP, or the input comparison register KCP must be set again. Note that the input pins disabled from interrupt by the SIK register do not affect interrupt generation conditions.

An interrupt is generated for K6[3:0] in the same way as described above.

(2) K64 to K67 input interrupts

Figure 14.1.4 shows the structure of K64 to K67 input-interrupt circuits.

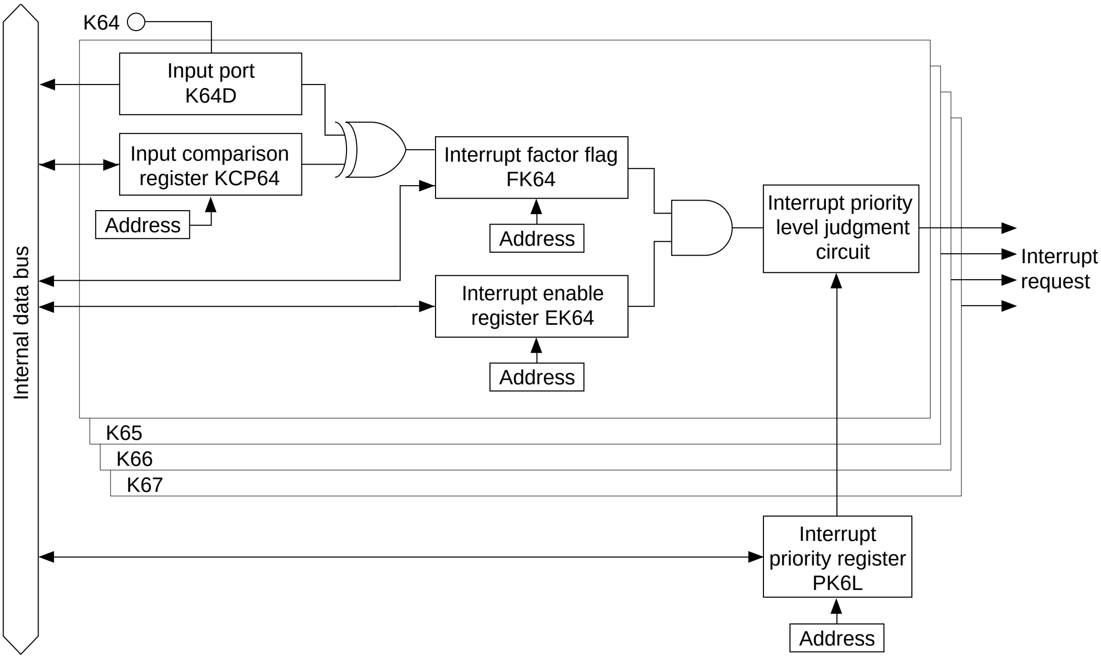

K64
Input port K64D
Input comparison register KCP64
Address
Interrupt factor flag FK64
Address
Interrupt priority level judgment circuit
Interrupt enable register EK64
Address
Interrupt request
Internal data bus
K65
K66
K67
Interrupt priority register PK6L
Address

Figure 14.1.4 Structure of K64 to K67 Input-Interrupt Circuits

Since the K64 to K67 input ports can generate an interrupt independently for each bit, no interrupt select register is provided. The interrupt enable register EK is used to enable and disable the interrupts.

The input comparison register KCP is used to select whether an interrupt for each input port is to be generated at the rising or falling edge of the input.

Each individual interrupt for the K64 to K67 input ports is generated, if the interrupt has been enabled by the interrupt-enable register EK, at the rising or falling edge of the input (as set by the input comparison register KCP).

Control registers of the interrupt controller

Table 14.1.2 shows the control registers of the interrupt controller that are provided for each input-interrupt group.

Table 14.1.2 Control Registers of Interrupt Controller

Group	Interrupt factor flag	Interrupt enable register	Interrupt priority register	IDMA request register
K5[4:0]	FK5(D5/0x40280)	EK5(D5/0x40270)	PK6L[2:0](D[6:4]/0x40260)	–
K6[3:0]	FK6(D4/0x40280)	EK6(D4/0x40270)		–
K64	FK64(D3/0x40280)	EK64(D3/0x40270)	PK6H[2:0](D[2:0]/0x40260)	RK64(D3/0x40290)
K65	FK65(D2/0x40280)	EK65(D2/0x40270)		RK65(D2/0x40290)
K66	FK66(D1/0x40280)	EK66(D1/0x40270)		RK66(D1/0x40290)
K67	FK67(D0/0x40280)	EK67(D0/0x40270)		RK67(D0/0x40290)

When the interrupt generation condition described above is met, the corresponding interrupt factor flag is set to "1". If the interrupt enable register bit for that interrupt factor has been set to "1", an interrupt request is generated.

Interrupts due to an interrupt factor can be disabled by leaving the interrupt enable register bit for that factor set to "0". The interrupt factor flag is set to "1" whenever interrupt generation conditions are met, regardless of the setting of the interrupt enable register.

The interrupt priority register sets the interrupt priority level (0 to 7) for each interrupt group. An interrupt request to the CPU is accepted only when no other interrupt request of a higher priority has been generated.

In addition, only when the PSR's IE bit = "1" (interrupts enabled) and the set value of the IL is smaller than the input interrupt level set using the interrupt priority register will the input interrupt request actually be accepted by the CPU.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, refer to Chapter 8, "Interrupt".

Intelligent DMA

The interrupt group K6[7:4] can invoke an intelligent DMA (IDMA) through the use of its interrupt factor. This enables the K67 to K64 inputs to be used as a trigger to perform DMA transfer.

The following shows the IDMA channel numbers assigned to each interrupt factor:

	IDMA Ch.
K67 input interrupt:	1
K66 input interrupt:	2
K65 input interrupt:	3
K64 input interrupt:	4

Before IDMA can be invoked, the IDMA request bit shown in Table 14.1.2 must be set to "1". Transfer conditions, etc. on the IDMA side must also be set in advance.

If an interrupt factor occurs when the IDMA request bit is set to "1", IDMA is invoked. No interrupt request is generated at that point. An interrupt request is generated upon completion of the DMA transfer. Alternately, the bits can be set so that they do not generate an interrupt, with only a DMA transfer performed.

For details on DMA transfer and means of controlling interrupts upon completion of DMA transfer, refer to Section 9.2, "Intelligent DMA".

Trap vectors

The trap-vector address of each input default interrupt factor is set as follows:

	BTA3 = low
K67 input interrupt:	0x0C00040
K66 input interrupt:	0x0C00044
K65 input interrupt:	0x0C00048
K64 input interrupt:	0x0C0004C
K6[3:0] input interrupt:	0x0C00050
K5[4:0] input interrupt:	0x0C00054

The base address of the trap table can be changed using the TTBR register (0x48134 to 0x48137).

14.1.5 I/O Memory of Input Ports

Table 14.1.3 shows the control bits of the input ports.

Table 14.1.3 Control Bits of Input Ports

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
K6/K5 interrupt priority register	**0040260 (B)**	D7	**–**	reserved	–				–	–	0 when being read.
		D6	**PK6L2**	K6[3:0], K5[4:0] interrupt level	0 to 7				X	R/W	
		D5	**PK6L1**						X		
		D4	**PK6L0**						X		
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**PK6H2**	K6[7:4] interrupt level	0 to 7				X	R/W	
		D1	**PK6H1**						X		
		D0	**PK6H0**						X		
K6/K5 interrupt enable register	**0040270 (B)**	D7–6	**–**	reserved	–				–	–	0 when being read.
		D5	**EK5**	K5[4:0] input	1	Enabled	0	Disabled	0	R/W	
		D4	**EK6**	K6[3:0] input					0	R/W	
		D3	**EK64**	K64 input					0	R/W	
		D2	**EK65**	K65 input					0	R/W	
		D1	**EK66**	K66 input					0	R/W	
		D0	**EK67**	K67 input					0	R/W	
K6/K5 interrupt factor flag register	**0040280 (B)**	D7–6	**–**	reserved	–				–	–	0 when being read.
		D5	**FK5**	K5[4:0] input	1	Factor is generated	0	No factor is generated	X	R/W	
		D4	**FK6**	K6[3:0] input					X	R/W	
		D3	**FK64**	K64 input					X	R/W	
		D2	**FK65**	K65 input					X	R/W	
		D1	**FK66**	K66 input					X	R/W	
		D0	**FK67**	K67 input					X	R/W	
K6, DMA, 16-bit timer 0 IDMA request register	**0040290 (B)**	D7	**R16TC01**	16-bit timer 01 comparison match	1	IDMA request	0	Interrupt request	X	R/W	
		D6	**R16TU01**	16-bit timer 01 underflow					X	R/W	
		D5	**RHDM1**	High-speed DMA Ch.1					X	R/W	
		D4	**RHDM0**	High-speed DMA Ch.0					X	R/W	
		D3	**RK64**	K64 input					X	R/W	
		D2	**RK65**	K65 input					X	R/W	
		D1	**RK66**	K66 input					X	R/W	
		D0	**RK67**	K67 input					X	R/W	
K5 function select register	**00402C0 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**CFK54**	K54 function selection	1	DA1	0	K54	0	W	Undefined in read.
		D3	**CFK53**	K53 function selection	1	DA0	0	K53	0	W	
		D2	**CFK52**	K52 function selection	1	#ADTRG	0	K52	0	W	
		D1	**CFK51**	K51 function selection	1	#DMAREQ1	0	K51	0	W	
		D0	**CFK50**	K50 function selection	1	#DMAREQ0	0	K50	0	W	
K5 input port data register	**00402C1 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**K54D**	K54 input port data	1	High	0	Low	–	R	
		D3	**K53D**	K53 input port data					–	R	
		D2	**K52D**	K52 input port data					–	R	
		D1	**K51D**	K51 input port data					–	R	
		D0	**K50D**	K50 input port data					–	R	
K5 interrupt select register	**00402C2 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**SIK54**	K54 interrupt selection	1	Interrupt enabled	0	Interrupt disabled	0	R/W	
		D3	**SIK53**	K53 interrupt selection					0	R/W	
		D2	**SIK52**	K52 interrupt selection					0	R/W	
		D1	**SIK51**	K51 interrupt selection					0	R/W	
		D0	**SIK50**	K50 interrupt selection					0	R/W	
K5 input comparison register	**00402C3 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**KCP54**	K54 input comparison	1	Interrupt is generated at falling edge.	0	Interrupt is generated at rising edge.	0	R/W	
		D3	**KCP53**	K53 input comparison					0	R/W	
		D2	**KCP52**	K52 input comparison					0	R/W	
		D1	**KCP51**	K51 input comparison					0	R/W	
		D0	**KCP50**	K50 input comparison					0	R/W	
K5 pull-up control register	**00402C4 (B)**	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**KPU54**	K54 pull-up control	1	Pulled up	0	No pull-up	0	W	Undefined in read.
		D3	**KPU53**	K53 pull-up control					0	W	
		D2	**KPU52**	K52 pull-up control					0	W	
		D1	**KPU51**	K51 pull-up control					0	W	
		D0	**KPU50**	K50 pull-up control					0	W	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
K6 function select register	**00402C5 (B)**	D7	**CFK67**	K67 function selection	1	AD7	0	K67	0	W	Undefined in read.
		D6	**CFK66**	K66 function selection	1	AD6	0	K66	0	W	
		D5	**CFK65**	K65 function selection	1	AD5	0	K65	0	W	
		D4	**CFK64**	K64 function selection	1	AD4	0	K64	0	W	
		D3	**CFK63**	K63 function selection	1	AD3	0	K63	0	W	
		D2	**CFK62**	K62 function selection	1	AD2	0	K62	0	W	
		D1	**CFK61**	K61 function selection	1	AD1	0	K61	0	W	
		D0	**CFK60**	K60 function selection	1	AD0	0	K60	0	W	
K6 input port data register	**00402C6 (B)**	D7	**K67D**	K67 input port data	1	High	0	Low	–	R	
		D6	**K66D**	K66 input port data					–	R	
		D5	**K65D**	K65 input port data					–	R	
		D4	**K64D**	K64 input port data					–	R	
		D3	**K63D**	K63 input port data					–	R	
		D2	**K62D**	K62 input port data					–	R	
		D1	**K61D**	K61 input port data					–	R	
		D0	**K60D**	K60 input port data					–	R	
K6 interrupt select register	**00402C7 (B)**	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**SIK63**	K63 interrupt selection	1	Interrupt enabled	0	Interrupt disabled	0	R/W	
		D2	**SIK62**	K62 interrupt selection					0	R/W	
		D1	**SIK61**	K61 interrupt selection					0	R/W	
		D0	**SIK60**	K60 interrupt selection					0	R/W	
K6 input comparison register	**00402C8 (B)**	D7	**KCP67**	K67 input comparison	1	Interrupt is generated at falling edge.	0	Interrupt is generated at rising edge.	0	R/W	
		D6	**KCP66**	K66 input comparison					0	R/W	
		D5	**KCP65**	K65 input comparison					0	R/W	
		D4	**KCP64**	K64 input comparison					0	R/W	
		D3	**KCP63**	K63 input comparison					0	R/W	
		D2	**KCP62**	K62 input comparison					0	R/W	
		D1	**KCP61**	K61 input comparison					0	R/W	
		D0	**KCP60**	K60 input comparison					0	R/W	
K6 pull-up control register	**00402C9 (B)**	D7	**KPU67**	K67 pull-up control	1	Pulled up	0	No pull-up	0	W	Undefined in read.
		D6	**KPU66**	K66 pull-up control					0	W	
		D5	**KPU65**	K65 pull-up control					0	W	
		D4	**KPU64**	K64 pull-up control					0	W	
		D3	**KPU63**	K63 pull-up control					0	W	
		D2	**KPU62**	K62 pull-up control					0	W	
		D1	**KPU61**	K61 pull-up control					0	W	
		D0	**KPU60**	K60 pull-up control					0	W	

CFK54–CFK50: K5[4:0] function selection (D[4:0]) / K5 function select register (0x402C0)
CFK67–CFK60: K6[7:0] function selection (D[4:0]) / K6 function select register (0x402C0)

Selects the function of each input-port pin.

Write "1": Used for peripheral circuit
Write "0": Input port pin
Read: Invalid

When a bit of the CFK register is set to "1", the corresponding pin is set for use with the peripheral circuit (see Table 14.1.1). The pins for which register bits are set to "0" can be used as general-purpose input ports.
Since the CFK register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.
At cold start, CFK is set to "0" (input port). At hot start, CFK retains its state from prior to the initial reset.

K54D–K50D: K5[4:0] input port data (D[4:0]) / K5 input port data register (0x402C1)
K67D–K60D: K6[7:0] input port data (D[7:0]) / K6 input port data register (0x402C6)

The input data on each input port pin can be read from this register.

Read "1": High level
Read "0": Low level
Write: Invalid

The pin voltage of each input port can be read out "1" directly when the voltage is high (V_{DD}) or "0" when the voltage is low (V_{SS}) respectively.
Since this register is a read-only register, writing to the register is ignored.
When the ports set for A/D converter input or D/A converter output are read, the value obtained is always "0".

SIK54–SIK50: K5[4:0] port interrupt selection (D[4:0]) / K5 interrupt select register (0x402C2)
SIK63–SIK60: K6[3:0] port interrupt selection (D[3:0]) / K6 interrupt select register (0x402C6)

Sets conditions for input-port interrupt generation (interrupt enabled/disabled).

Write "1":	Interrupt enabled
Write "0":	Interrupt disabled
Read:	Valid

SIK is an interrupt select register for each bit of input ports. Interrupts for bits set to "1" are enabled, and interrupts for bits set to "0" are disabled. A change in the state of an input pin that is disabled from interrupt does not affect interrupt generation.
At cold start, SIK is set to "0" (interrupt disabled). At hot start, SIK retains its state from prior to the initial reset.
No interrupt select register is provided for the K6[7:4] input ports, as interrupts for these ports can be controlled bit-for-bit.

KCP54–KCP50: K5[4:0] port input comparison (D[4:0]) / K5 input comparison register (0x402C3)
KCP67–KCP60: K6[7:0] port input comparison (D[7:0]) / K6 input comparison register (0x402C8)

Sets the conditions for input-port interrupt generation (timing of interrupt generation).

Write "1":	Generated at falling edge
Write "0":	Generated at rising edge
Read:	Valid

KCP5[4:0] is compared with the input state of five bits of K5[4:0], and KCP6[3:0] is compared with the input state of four bits of K6[3:0], and when a change in states from a matched to an unmatched state occurs in either, an interrupt is generated (except for the inputs disabled from interrupt by the SIK register).
KCP67 to KCP64 are compared individually with each bit, K67 to K64. An interrupt for which bits are set to "1" occurs at the falling edge of the input, and an interrupt for which bits are set to "0" occurs at the rising edge of the input.
At cold start, KCP is set to "0" (rising edge). At hot start, KCP retains its state from prior to the initial reset.

KPU54–KPU50: K5[4:0] pull-up control (D[4:0]) / K5 pull-up control register (0x402C4)
KPU67–KPU60: K6[7:0] pull-up control (D[7:0]) / K6 pull-up control register (0x402C9)

Controls the pull-up of each input pin.

Write "1":	Pulled up
Write "0":	Not pulled up
Read:	Invalid

When a bit of the KPU register is set to "1", the internal pull-up resistor of the corresponding pin is enabled, so the pin is pulled up. The pins for which register bits are set to "0" are not pulled up.
Since the KPU register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.
At cold start, KPU is set to "0" (no pull-up). At hot start, KPU retains its state from prior to the initial reset.

PK6L2–PK6L0: K6[3:0]&K5[4:0] input interrupt level (D[6:4]) / K6/K5 interrupt priority register (0x40260)
PK6H2–PK6H0: K6[7:4] input interrupt level (D[2:0]) / K6/K5 interrupt priority register (0x40260)

Sets the priority level of the input interrupt.
PK6L[2:0] and PK6H[2:0] are interrupt priority registers corresponding to the K6[3:0] and K5[4:0] input interrupts and the K6[7:4] input interrupt, respectively.
The priority level can be set for each interrupt group in the range of 0 to 7.
At initial reset, PK becomes indeterminate.

EK5: K5[4:0] input interrupt enable (D5) / K6/K5 interrupt enable register (0x40270)
EK6: K6[3:0] input interrupt enable (D4) / K6/K5 interrupt enable register (0x40270)
EK64–EK67: K64–K67 input interrupt enable (D[3:0]) / K6/K5 interrupt enable register (0x40270)

Enables or disables the generation of an interrupt to the CPU.

Write "1": Interrupt enabled
Write "0": Interrupt disabled
Read: Valid

EK5, EK6, and EK67–EK64 are interrupt enable bits corresponding to five bits of K5[4:0], four bits of K6[3:0], and each bit from K67 to K64, respectively. Interrupts for input ports set to "1" are enabled, and interrupts for input ports set to "0" are disabled.
At initial reset, all EK bits are set to "0" (interrupt disabled).

FK5: K5[4:0] input interrupt factor flag (D5) / K6/K5 interrupt factor flag register (0x40280)
FK6: K6[3:0] input interrupt factor flag (D4) / K6/K5 interrupt factor flag register (0x40280)
FK64–FK67: K64–K67 input interrupt factor flag (D[3:0]) / K6/K5 interrupt factor flag register (0x40280)

Indicates the status of an input interrupt factor generated.

When read
Read "1": Interrupt factor has occurred
Read "0": No interrupt factor has occurred

When written using the reset-only method (default)
Write "1": Interrupt factor flag is reset
Write "0": Invalid

When written using the read/write method
Write "1": Interrupt flag is set
Write "0": Interrupt flag is reset

FK5, FK6, and FK64–FK67 are an interrupt factor flags corresponding to five bits of K5[4:0], four bits of K6[3:0], and each bit from K64 to K67, respectively. The flag is set to "1" when interrupt generation conditions are met.
At this time, if the following conditions are met, an interrupt to the CPU is generated:
1. The corresponding interrupt enable register bit is set to "1".
2. No other interrupt request of a higher priority has been generated.
3. The IE bit of the PSR is set to "1" (interrupts enabled).
4. The value set in the corresponding interrupt priority register is higher than the interrupt level (IL) of the CPU.
When using the interrupt factor of the K67–K64 to request IDMA, note that even when the above conditions are met, no interrupt request to the CPU is generated for the interrupt factor that has occurred. If interrupts are enabled at the setting of IDMA, an interrupt is generated under the above conditions after the data transfer by IDMA is completed.

The interrupt factor flag is set to "1" whenever interrupt generation conditions are met, regardless of how the interrupt enable and interrupt priority registers are set.
If the next interrupt is to be accepted after an interrupt has occurred, it is necessary that the interrupt factor flag be reset, and that the PSR be set again (by setting the IE bit to "1" after setting the IL to a value lower than the level indicated by the interrupt priority register, or by executing the reti instruction).
The interrupt factor flag can be reset only by writing to it in the software. Note that if the PSR is set again to accept interrupts generated (or if the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt occurs again. Note also that the value to be written to reset the flag is "1" when the reset-only method (RSTONLY = "1") is used, and "0" when the read/write method (RSTONLY = "0") is used.
At initial reset, all FK flags become indeterminate, so be sure to reset them in the software.

RK64–RK67: K64–K67 IDMA request (D[3:0]) / K6, DMA, 16-bit timer 0 IDMA request register (0x40290)

Specifies whether to invoke IDMA when an interrupt factor occurs.

Write "1":	IDMA request
Write "0":	Interrupt request
Read:	Valid

RK64 to RK67 are IDMA request bits corresponding to bits K67 to K64, respectively. If the bit is set to "1", IDMA is invoked when an interrupt factor occurs, thereby performing a programmed data transfer. If the bit is set to "0", normal interrupt processing is performed, without invoking IDMA.
For details on IDMA, refer to Section 9.2, "Intelligent DMA".
At initial reset, RK becomes indeterminate, so be sure to reset it in the software.

14.1.6 Programming Notes

(1) When using the internal pull-up resistor, note that when the state of the input pin changes from low to high, a time constant comprised of the pull-up resistor and the pin's load capacitance causes a delay in the rising transition of the waveform. For this reason, an appropriate wait time must be provided prior to taking the input port into the internal logic. This is especially important for key scanning in a key-matrix configuration. The wait time set here must be equal to or greater than the value calculated from the following equation:

Wait time = $R_{IN} \times (C_{IN}$ + load capacitance on the board) $\times$ 1.6 [sec]
R_{IN} = pull-up resistance (max.)
C_{IN} = pin capacitance (max.)

(2) Since the pin function select register (CFK) and the pull-up control register (KPU) are write-only registers (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite them. Use ordinary storage instructions for this purpose.

(3) After an initial reset, the interrupt factor flag (FK) and IDMA request register (RK) become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset this flag and register in a program.

(4) To prevent regeneration of interrupts due to the same factor following the occurrence of an interrupt, always be sure to reset the interrupt factor flag (FK) before resetting the PSR or executing the reti instruction.

14.2 Output Ports (R Ports)

14.2.1 Structure of Output Port

The E0C33A104 contains 11 bits of output ports (R70 to R77, R81 to R83).

Figure 14.2.1 shows the structure of a typical output port.

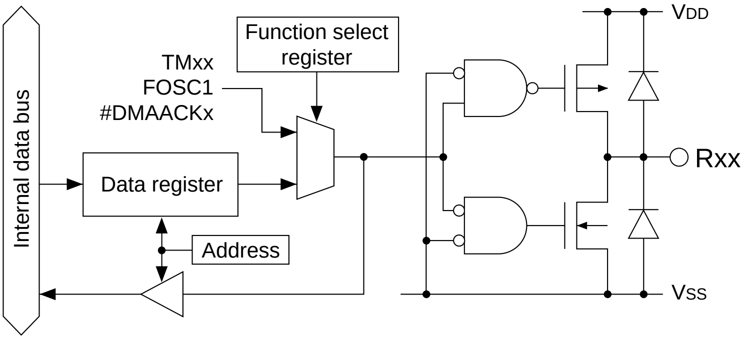


Figure 14.2.1 Structure of Output Port

Output specifications are fixed at a complimentary output.

As shown in Figure 14.2.1, the state of the output pin is high (VDD level) when "1" is written to the data register of the output port, and changes to low (VSS level) when "0" is written.

14.2.2 Output-Port Pins

The output pins concurrently serve as the output pins for peripheral circuits, as shown in Table 14.2.1. Whether they are used as output ports or for peripheral circuits can be set bit-for-bit using a function select register. All pins not used for peripheral circuits can be used as general-purpose output ports.

Table 14.2.1 Output Pins

Pin name	Pin No.		I/O	Pull-up	Function	Function select bit
	QFP5-128	QFP15-128				
R70/TM00	36	33	O	–	Output port / 16-bit timer (timer 00) output	CFR70(D0)/R7 function select register(0x402CA)
R71/TM01	37	34	O	–	Output port / 16-bit timer (timer 01) output	CFR71(D1)/R7 function select register(0x402CA)
R72/TM10	38	35	O	–	Output port / 16-bit timer (timer 10) output	CFR72(D2)/R7 function select register(0x402CA)
R73/TM11	39	36	O	–	Output port / 16-bit timer (timer 11) output	CFR73(D3)/R7 function select register(0x402CA)
R74/TM21	40	37	O	–	Output port / 16-bit timer (timer 21) output	CFR74(D4)/R7 function select register(0x402CA)
R75/TM31	41	38	O	–	Output port / 16-bit timer (timer 31) output	CFR75(D5)/R7 function select register(0x402CA)
R76/TM41	42	39	O	–	Output port / 16-bit timer (timer 41) output	CFR76(D6)/R7 function select register(0x402CA)
R77/TM51	43	40	O	–	Output port / 16-bit timer (timer 51) output	CFR77(D7)/R7 function select register(0x402CA)
R81/FOSC1/ #DRD	57	54	O	–	Output port / Low-speed (OSC1) oscillation clock output / #DRD output	CFR81(D1)/R8 function select register(0x402CC) CFEX2(D2)/Port function extension register(0x402D8)
R82/#DMAACK0	59	56	O	–	Output port / High-speed DMA acknowledge 0	CFR82(D2)/R8 function select register(0x402CC)
R83/#DMAACK1	55	52	O	–	Output port / High-speed DMA acknowledge 1	CFR83(D3)/R8 function select register(0x402CC)

At cold start, all pins are set for output ports Rxx (function select register CFRxx = "0"). When these pins are used for the internal peripheral circuits, write "1" to CFRxx. For details on pin functions in this case, refer to the description of each peripheral circuit in this manual. Since the function select register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) that accompany a read-modify-write operation cannot be used to rewrite it. Use ordinary storage instructions for this purpose.
At hot start, the pins retain their state from prior to the reset.
The DRAM read signal #DRD shares a pin with R81/FOSC1. When outputting #DRD, set the R81 function extension bit CFEX2 (D2) / Port function extension register (0x402B8) to "1". The settings of CFEX2 have priority over that of CFR81.

14.2.3 I/O Memory of Output Ports

Table 14.2.2 shows the control bits of the output ports.

Table 14.2.2 Control Bits of Output Ports

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
R7 function select register	00402CA (B)	D7	**CFR77**	R77 function selection	1	TM51	0	R77	0	W	Undefined in read.
		D6	**CFR76**	R76 function selection	1	TM41	0	R76	0	W	
		D5	**CFR75**	R75 function selection	1	TM31	0	R75	0	W	
		D4	**CFR74**	R74 function selection	1	TM21	0	R74	0	W	
		D3	**CFR73**	R73 function selection	1	TM11	0	R73	0	W	
		D2	**CFR72**	R72 function selection	1	TM10	0	R72	0	W	
		D1	**CFR71**	R71 function selection	1	TM01	0	R71	0	W	
		D0	**CFR70**	R70 function selection	1	TM00	0	R70	0	W	
R7 output port data register	00402CB (B)	D7	**R77D**	R77 output port data	1	High	0	Low	0	R/W	
		D6	**R76D**	R76 output port data					0	R/W	
		D5	**R75D**	R75 output port data					0	R/W	
		D4	**R74D**	R74 output port data					0	R/W	
		D3	**R73D**	R73 output port data					0	R/W	
		D2	**R72D**	R72 output port data					0	R/W	
		D1	**R71D**	R71 output port data					0	R/W	
		D0	**R70D**	R70 output port data					0	R/W	
R8 function select register	00402CC (B)	D7–4	–	reserved	–				–	–	0 when being read.
		D3	**CFR83**	R83 function selection	1	#DMAACK1	0	R83	0	W	Undefined in read.
		D2	**CFR82**	R82 function selection	1	#DMAACK0	0	R82	0	W	
		D1	**CFR81**	R81 function selection	1	FOSC1	0	R81	0	W	
		D0	–	–	–				–	–	0 when being read.
R8 output port data register	00402CD (B)	D7–4	–	reserved	–				–	–	0 when being read.
		D3	**R83D**	R83 output port data	1	High	0	Low	1	R/W	
		D2	**R82D**	R82 output port data					1	R/W	
		D1	**R81D**	R81 output port data					1	R/W	
		D0	–	–	–				–	–	0 when being read.
Port function extension register	00402D8 (B)	D7–3	–	reserved	–				–	–	Undefined in read.
		D2	**CFEX2**	R81 port extended function	1	#DRD	0	R81/FOSC1	0	W	Undefined in read.
		D1	**CFEX1**	P14 port extended function	1	#DWE	0	P14/#BUSGET	0	W	
		D0	**CFEX0**	P1[3:0] port extended function	1	DST/DPCO	0	P1x/T8UFx	0	W	

CFR77–CFR70: R7[7:0] function selection (D[7:0]) / R7 function select register (0x402CA)
CFR83–CFR81: R8[3:1] function selection (D[3:1]) / R8 function select register (0x402CC)

Selects the function of each output-port pin.

Write "1": Used for peripheral circuit
Write "0": Output port pin
Read: Invalid

When a bit of the CFR register is set to "1", the corresponding pin is set for use with peripheral circuits (see Table 14.2.1). The pins for which register bits are set to "0" can be used as general-purpose output ports.
Since the CFR register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.
At cold start, CFR is set to "0" (output port). At hot start, CFR retains its state from prior to the initial reset.

R77D–R70D: R7[7:0] output port data (D[7:0]) / R7 output port data register (0x402CB)
R83D–R81D: R8[3:1] output port data (D[3:1]) / R8 output port data register (0x402CD)

Sets output data for each output port.

Write "1":	High-level output
Write "0":	Low-level output
Read:	Valid

When RxxD is set to "1", the corresponding output port pin goes high level (VDD level); when set to "0", the corresponding output port pin goes low level (VSS level).
At initial reset, all R7xD registers are set to "0" (low) and all R8xD registers are set to "1" (high level).

CFEX2: R81 function extension (D2) / Port function extension register (0x402D8)

Sets whether the function of the R81 pin is to be extended.

Write "1":	#DRD output pin
Write "0":	R81/FOSC1 output pin
Read:	Invalid

When CFEX2 is set to "1", the R81 pin functions as a DRAM read-signal #DRD output pin. When CFEX2 = "0", the CFR81 bit becomes effective, so the settings of this bit determine whether the R81 pin functions as an R81 output port or a FOUT1 output pin.
Since the CFEX2 bit is a write-only bit (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.
At cold start, CFEX2 is set to "0" (R81/FOSC1 output pin). At hot start, CFEX2 retains its state from prior to the initial reset.

14.2.4 Programming Note

Since the pin function select register (CFR) and R81 function extension bit (CFEX2) are write-only register/bit (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite the registers. Use ordinary storage instructions for this purpose.

14.3 I/O Ports (P Ports)

14.3.1 Structure of I/O Port

The E0C33A104 contains 15 bits of I/O ports (P00 to P07, P10 to p16) that can be directed for input or output through the use of a program.

Figure 14.3.1 shows the structure of a typical I/O port.

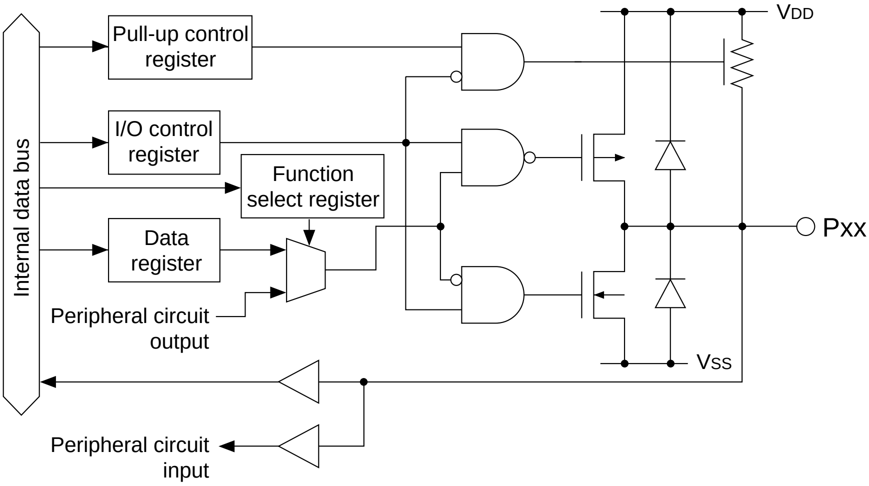


Figure 14.3.1 Structure of I/O Port

14.3.2 I/O Port Pins

The I/O ports concurrently serve as the input/output pins for peripheral circuits, as shown in Table 14.3.1. Whether they are used as I/O ports or for peripheral circuits can be set bit-for-bit using a function select register. All pins not used for peripheral circuits can be used as general-purpose I/O ports.

Table 14.3.1 I/O Pins

Pin name	Pin No.		I/O	Pull-up	Function	Function select bit
	QFP5-128	QFP15-128				
P00/SIN0	60	57	I/O	Built-in	I/O port / Serial IF Ch0 data input	CFP00(D0)/P0 function select register(0x402D0)
P01/SOUT0	63	60	I/O	Built-in	I/O port / Serial IF Ch0 data output	CFP01(D1)/P0 function select register(0x402D0)
P02/#SCLK0	64	61	I/O	Built-in	I/O port / Serial IF Ch0 clock input/output	CFP02(D2)/P0 function select register(0x402D0)
P03/#SRDY0	65	62	I/O	Built-in	I/O port / Serial IF Ch0 ready input/output	CFP03(D3)/P0 function select register(0x402D0)
P04/SIN1	66	63	I/O	Built-in	I/O port / Serial IF Ch1 data input	CFP04(D4)/P0 function select register(0x402D0)
P05/SOUT1	67	64	I/O	Built-in	I/O port / Serial IF Ch1 data output	CFP05(D5)/P0 function select register(0x402D0)
P06/#SCLK1	68	65	I/O	Built-in	I/O port / Serial IF Ch1 clock input/output	CFP06(D6)/P0 function select register(0x402D0)
P07/#SRDY1	69	66	I/O	Built-in	I/O port / Serial IF Ch1 ready input/output	CFP07(D7)/P0 function select register(0x402D0)
P10/EXCL00/ T8UF0/DST0	46	43	I/O	Built-in	I/O port / Timer 00 event counter input (I) / 8-bit timer 0 output (O) / DST0 output (O)	CFP10(D0)/P1 function select register(0x402D4) CFEX0(D0)/Port function extension register(0x402D8)
P11/EXCL01/ T8UF1/DST1	47	44	I/O	Built-in	I/O port / Timer 01 event counter input (I) / 8-bit timer 1 output (O) / DST1 output (O)	CFP11(D1)/P1 function select register(0x402D4) CFEX0(D0)/Port function extension register(0x402D8)

Pin name	Pin No.		I/O	Pull-up	Function	Function select bit
	QFP5-128	QFP15-128				
P12/EXCL10/ T8UF2/DST2	48	45	I/O	Built-in	I/O port / Timer 10 event counter input (I) / 8-bit timer 2 output (O) / DST2 output (O)	CFP12(D2)/P1 function select register(0x402D4) CFEX0(D0)/Port function extension register(0x402D8)
P13/EXCL20/ T8UF3/DPCO	49	46	I/O	Built-in	I/O port / Timer 20 event counter input (I) / 8-bit timer 3 output (O) / DPCO output (O)	CFP13(D3)/P1 function select register(0x402D4) CFEX0(D0)/Port function extension register(0x402D8)
P14/EXCL30 /#BUSGET/ #DWE	50	47	I/O	Built-in	I/O port / Timer 30 event counter input (I) / #BUSGET output (O) / #DWE output (O)	CFP14(D4)/P1 function select register(0x402D4) CFEX1(D1)/Port function extension register(0x402D8)
P15/EXCL40 /#DMAEND0	51	48	I/O	Built-in	I/O port / Timer 40 event counter input (I) / High-speed DMA Ch0 end signal output (O)	CFP15(D5)/P1 function select register(0x402D4)
P16/EXCL50 /#DMAEND1	52	49	I/O	Built-in	I/O port / Timer 50 event counter input (I) / High-speed DMA Ch1 end signal output (O)	CFP16(D6)/P1 function select register(0x402D4)

(I): Input mode, (O): Output mode

At cold start, all pins are set for I/O ports Pxx (function select register CFPxx = "0"). When these pins are used for the internal peripheral circuits, write "1" to CFPxx. For details on pin functions in this case, refer to the description of each peripheral circuit in this manual. Since the function select register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) that accompany a read-modify-write operation cannot be used to rewrite it. Use ordinary storage instructions for this purpose.

At hot start, the pins retain their state from prior to the reset.

In addition to being an I/O port, the P10–P16 pins are shared with two types (three types for P10–P14) of peripheral circuits. The type of peripheral circuit for which these pins are used is determined by the direction (input or output) in which the pin is set using an I/O control register, as will be described later.

The debug signals (DST[2:0], DPCO) and DRAM write signal #DWE share pins with P10 to P14. When outputting #DWE, set the P14 function extension bit CFEX1 (D1) / Port function extension register (0x402D8) to "1".

P10 to P13 can be set for DST[2:0] and DPCO outputs by setting the P1[3:0] function extension bit CFEX0 (D0) / Port function extension register (0x402D8) to "1". Since DST[2:0] and DPCO are test signals for debugging purposes, leave CFEX0 set to "0" while it is in use.

The settings of CFEX1 and CFEX0 have priority over those of CFP14 and CFP1[3:0].

14.3.3 I/O Control Register and I/O Modes

The I/O ports are directed for input or output modes by writing data to an I/O control register corresponding to each port bit.

P07–P00 I/O control: IOC0[7:0] (D[7:0]) / P0 I/O control register (0x402D2)

P16–P10 I/O control: IOC1[6:0] (D[6:0]) / P1 I/O control register (0x402D6)

To set an I/O port for input, write "0" to the I/O control bit. I/O ports set for input mode are placed in the high-impedance state, and thus function as input ports.

In the input mode, the state of the input pin is read directly, so the data is "1" when the pin state is high (VDD level) or "0" when the pin state is low (VSS level).

Even in the input mode, data can be written to the data register without affecting the pin state.

The pins used for input can have an internal pull-up resistor added, as will be described in the next section.

To set an I/O port for output, write "1" to the I/O control bit. I/O port set for output function as output ports. When the port output data is "1", the port outputs a high level (VDD level); when the data is "0", the port outputs a low level (VSS level).

At cold start, the I/O control register is set to "0" (input mode).
Since the I/O control register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) that accompany a read-modify-write operation cannot be used to rewrite it. Use ordinary storage instructions for this purpose.
At hot start, the pins retain their state from prior to the reset.

Note: If pins P10 to P16 are set for use with peripheral circuits, their pin functions vary depending on the input/output direction control by the IOC1x register.
If pins P00 to P07 are set for use with the serial interface, their input/output directions are controlled by the hardware.

14.3.4 Pull-Up Resistors

The I/O port pins contain a pull-up resistor. Whether this pull-up resistor is to be used can be selected for each pin. Use a pull-up control register for this selection.
P07–P00 pull-up control: IOU0[7:0] (D[7:0]) / P0 pull-up control register (0x402D3)
P16–P10 pull-up control: IOU1[6:0] (D[6:0]) / P1 pull-up control register (0x402D7)

At cold start, the pull-up control registers are all set to "0", so the pull-up resistors are disconnected the pins. If the pins are to be pulled up, set the bit (IOUxx) corresponding to the pin (Pxx) to "1". Since the pull-up control registers are write-only registers (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite the registers. Use ordinary storage instructions for this purpose.
At hot start, the pins retain their state from prior to the reset.

The pins with a pull-up resistor (IOUxx = "1") are effective for inputs to push switches and the key matrix. However, when the state of the input pin changes from low to high, a time constant comprised of the pull-up resistor and the pin's load capacitance causes a delay in the rising transition of the waveform. For this reason, an appropriate wait time must be provided before taking the input data into the internal logic. This is especially important for key scanning in a key-matrix configuration. The wait time set here must be equal to or greater than the value calculated from the following equation:

Wait time = $R_{IN} \times (C_{IN}$ + load capacitance on the board) × 1.6 [sec]
R_{IN} = pull-up resistance (max.)
C_{IN} = pin capacitance (max.)

If the pull-up resistor is disconnected (IOUxx = "0"), the pins can be used effectively for slide-switch input or interfacing with other LSIs. In this case, be careful not to cause a floating state in the inputs.

The unused I/O port pins must have their pull-up resistors connected.
The pull-up resistors set here are only valid in the input mode; in the output mode, they are disconnected from the pin.

Note: The pull-up control register can also be used when the input pins are set for use with other peripheral circuits.

14.3.5 I/O Memory of I/O Ports

Table 14.3.2 shows the control bits of the I/O ports.

Table 14.3.2 Control Bits of I/O Ports

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 function select register	**00402D0 (B)**	D7	**CFP07**	P07 function selection	1	#SRDY1	0	P07	0	W	Undefined in read.
		D6	**CFP06**	P06 function selection	1	#SCLK1	0	P06	0	W	
		D5	**CFP05**	P05 function selection	1	SOUT1	0	P05	0	W	
		D4	**CFP04**	P04 function selection	1	SIN1	0	P04	0	W	
		D3	**CFP03**	P03 function selection	1	#SRDY0	0	P03	0	W	
		D2	**CFP02**	P02 function selection	1	#SCLK0	0	P02	0	W	
		D1	**CFP01**	P01 function selection	1	SOUT0	0	P01	0	W	
		D0	**CFP00**	P00 function selection	1	SIN0	0	P00	0	W	
P0 I/O port data register	**00402D1 (B)**	D7	**P07D**	P07 I/O port data	1	High	0	Low	0	R/W	
		D6	**P06D**	P06 I/O port data					0	R/W	
		D5	**P05D**	P05 I/O port data					0	R/W	
		D4	**P04D**	P04 I/O port data					0	R/W	
		D3	**P03D**	P03 I/O port data					0	R/W	
		D2	**P02D**	P02 I/O port data					0	R/W	
		D1	**P01D**	P01 I/O port data					0	R/W	
		D0	**P00D**	P00 I/O port data					0	R/W	
P0 I/O control register	**00402D2 (B)**	D7	**IOC07**	P07 I/O control	1	Output	0	Input	0	W	Undefined in read.
		D6	**IOC06**	P06 I/O control					0	W	
		D5	**IOC05**	P05 I/O control					0	W	
		D4	**IOC04**	P04 I/O control					0	W	
		D3	**IOC03**	P03 I/O control					0	W	
		D2	**IOC02**	P02 I/O control					0	W	
		D1	**IOC01**	P01 I/O control					0	W	
		D0	**IOC00**	P00 I/O control					0	W	
P0 pull-up control register	**00402D3 (B)**	D7	**IOU07**	P07 pull-up control	1	Pulled up	0	No pull-up	0	W	Undefined in read.
		D6	**IOU06**	P06 pull-up control					0	W	
		D5	**IOU05**	P05 pull-up control					0	W	
		D4	**IOU04**	P04 pull-up control					0	W	
		D3	**IOU03**	P03 pull-up control					0	W	
		D2	**IOU02**	P02 pull-up control					0	W	
		D1	**IOU01**	P01 pull-up control					0	W	
		D0	**IOU00**	P00 pull-up control					0	W	
P1 function select register	**00402D4 (B)**	D7	**–**	reserved	–				–	–	
		D6	**CFP16**	P16 function selection	1	EXCL50 #DMAEND1	0	P16	0	W	Undefined in read.
		D5	**CFP15**	P15 function selection	1	EXCL40 #DMAEND0	0	P15	0	W	
		D4	**CFP14**	P14 function selection	1	EXCL30 #BUSGET	0	P14	0	W	
		D3	**CFP13**	P13 function selection	1	EXCL20 T8UF3	0	P13	0	W	
		D2	**CFP12**	P12 function selection	1	EXCL10 T8UF2	0	P12	0	W	
		D1	**CFP11**	P11 function selection	1	EXCL01 T8UF1	0	P11	0	W	
		D0	**CFP10**	P10 function selection	1	EXCL00 T8UF0	0	P10	0	W	
P1 I/O port data register	**00402D5 (B)**	D7	**–**	reserved	–				–	–	Undefined in read.
		D6	**P16D**	P16 I/O port data	1	High	0	Low	0	R/W	
		D5	**P15D**	P15 I/O port data					0	R/W	
		D4	**P14D**	P14 I/O port data					0	R/W	
		D3	**P13D**	P13 I/O port data					0	R/W	
		D2	**P12D**	P12 I/O port data					0	R/W	
		D1	**P11D**	P11 I/O port data					0	R/W	
		D0	**P10D**	P10 I/O port data					0	R/W	
P1 I/O control register	**00402D6 (B)**	D7	**–**	reserved	–				–	–	
		D6	**IOC16**	P16 I/O control	1	Output	0	Input	0	W	Undefined in read.
		D5	**IOC15**	P15 I/O control					0	W	
		D4	**IOC14**	P14 I/O control					0	W	
		D3	**IOC13**	P13 I/O control					0	W	
		D2	**IOC12**	P12 I/O control					0	W	
		D1	**IOC11**	P11 I/O control					0	W	
		D0	**IOC10**	P10 I/O control					0	W	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P1 pull-up control register	**00402D7 (B)**	D7	–	reserved	–				–	–	
		D6	**IOU16**	P16 pull-up control	1	Pulled up	0	No pull-up	0	W	Undefined in read.
		D5	**IOU15**	P15 pull-up control					0	W	
		D4	**IOU14**	P14 pull-up control					0	W	
		D3	**IOU13**	P13 pull-up control					0	W	
		D2	**IOU12**	P12 pull-up control					0	W	
		D1	**IOU11**	P11 pull-up control					0	W	
		D0	**IOU10**	P10 pull-up control					0	W	
Port function extension register	**00402D8 (B)**	D7–3	–	reserved	–				–	–	Undefined in read.
		D2	**CFEX2**	R81 port extended function	1	#DRD	0	R81/FOSC1	0	W	Undefined in read.
		D1	**CFEX1**	P14 port extended function	1	#DWE	0	P14/#BUSGET	0	W	
		D0	**CFEX0**	P1[3:0] port extended function	1	DST/DPCO	0	P1x/T8UFx	0	W	

CFP07–CFP00: P0[7:0] function selection (D[7:0]) / P0 function select register (0x402D0)
CFP16–CFP10: P1[6:0] function selection (D[6:0]) / P1 function select register (0x402D4)

Selects the function of each I/O port pin.

Write "1": Used for peripheral circuit
Write "0": I/O port pin
Read: Invalid

When a bit of the CFP register is set to "1", the corresponding pin is set for use with peripheral circuits (see Table 14.3.1). The pins for which register bits are set to "0" can be used as general-purpose I/O ports.
Since the CFP register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.
At cold start, CFP is set to "0" (I/O port). At hot start, CFP retains its state from prior to the initial reset.

P07D–P00D: P0[7:0] I/O port data (D[7:0]) / P0 I/O port data register (0x402D1)
P16D–P10D: P1[6:0] I/O port data (D[6:0]) / P1 I/O port data register (0x402D5)

This register reads data from I/O-port pins or sets output data.

When writing data

Write "1": High level
Write "0": Low level

When an I/O port is set for output, the data written to it is directly output to the I/O port pin. If the data written to the port is "1", the port pin is set high (VDD level); if the data is "0", the port pin is set low (VSS level).
Even in the input mode, data can be written to the port data register.

When reading data

Read "1": High level
Read "0": Low level

The voltage level on the port pin is read out regardless of whether an I/O port is set for input or output mode. If the pin voltage is high (VDD level), "1" is read out as input data; if the pin voltage is low (VSS level), "0" is read out as input data.
At cold start, all data bits are set to "0". At hot start, they retain their state from prior to the initial reset.

IOC07–IOC00: P0[7:0] port I/O control (D[7:0]) / P0 port I/O control register (0x402D2)
IOC16–IOC10: P1[6:0] port I/O control (D[6:0]) / P1 port I/O control register (0x402D6)

Directs an I/O port for input or output.

Write "1": Output mode
Write "0": Input mode
Read: Invalid

This I/O control register corresponds bit-for-bit to each I/O port. When an IOC bit is set to "1", the corresponding I/O port is directed for output; if it is set to "0", the I/O port is directed for input.
Since the IOC register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.
At cold start, all IOC bits are set to "0" (input). At hot start, IOC retains its state from prior to the initial reset.

If pins P10 to P16 are set for use with peripheral circuits, their pin functions vary depending on the input/output direction control by the IOCx register.
If pins P00 to P07 are set for use with the serial interface, their input/output directions are controlled by the hardware.

IOU07–IOU00: P0[7:0] pull-up control (D[7:0]) / P0 pull-up control register (0x402D3)
IOU16–IOU10: P1[6:0] pull-up control (D[6:0]) / P1 pull-up control register (0x402D7)

Controls the pull-up of each I/O port pin.

Write "1":	Pulled up
Write "0":	Not pulled up
Read:	Invalid

When a bit of the IOU register is set to "1", the internal pull-up resistor of the corresponding pin is enabled, so the pin is pulled up when it is in the input mode. The pins for which register bits are set to "0" and those set for the output mode are not pulled up.
Since the IOU register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.
At cold start, the IOU registers are set to "0" (no pull-up) . At hot start, IOUxx retains its state from prior to the initial reset.
The pull-up control is effective when the I/O port pins are set for use with other peripheral circuits.

CFEX0: P1[3:0] function extension (D0) / Port function extension register (0x402D8)
CFEX1: P14 function extension (D1) / Port function extension register (0x402D8)

Sets whether the function of an I/O-port pin is to be extended.

Write "1":	Function-extended pin
Write "0":	I/O-port/peripheral-circuit pin
Read:	Invalid.

When CFEX1 is set to "1", pin P14 functions as a DRAM write-signal #DWE output pin. When CFEX1 = "0", the CFP14 bit becomes effective, so the settings of this bit determine whether pin P14 functions as an I/O port or a #BUSGET output pin.
When CFEX0 is set to "1", P10 to P13 are set for DST[2:0] and DPCO outputs. However, since DST[2:0] and DPCO are test signals for debugging purposes, leave CFEX0 set to "0" while in use.
Since the CFEX register is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.
At cold start, CFEX is set to "0" (I/O-port/peripheral-circuit pin). At hot start, CFEX retains its state from prior to the initial reset.

14.3.6 Programming Notes

(1) When the internal pull-up resistor is used, note that when the state of the input pin changes from low to high, a time constant comprised of the pull-up resistor and the pin's load capacitance causes a delay in the rising transition of the waveform. For this reason, an appropriate wait time must be provided before taking the input data into the internal logic. This is especially important for key scanning in a key-matrix configuration. The wait time set here must be equal to or greater than the value calculated from the following equation:

$$\text{Wait time} = R_{IN} \times (C_{IN} + \text{load capacitance on the board}) \times 1.6 \text{ [sec]}$$

R_{IN} = pull-up resistance (max.)
C_{IN} = pin capacitance (max.)

(2) Since the pin function select register (CFP), function extension register (CFEX), I/O control register (IOC), and pull-up control register (IOU) are all write-only registers (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite the registers. Use ordinary storage instructions for this purpose.

15 Power-Down Control

This chapter describes the controls used to reduce power consumption of the device.

Points on power saving

The current consumption of the device varies greatly with the CPU's operation mode, the system clocks used, and the peripheral circuits operated.

Current consumption	low←					→high
CPU/BCU	SLEEP	HALT2	Operating	HALT2	HALT(basic)	Operating
System clock	–	OSC1	OSC1	OSC3	OSC3	OSC3
OSC3 oscillation circuit	OFF	OFF	OFF	ON	ON	ON
Prescaler/peripheral circuit	STOP	STOP	STOP	RUN	RUN	RUN

To reduce power consumption of the device, it is important that as many unnecessary circuits as possible be turned off. In particular, peripheral circuits operating at a fast-clock rate consume a large amount of current, so design the program so that these circuits are turned off whenever unnecessary.

Power-saving in standby modes

When CPU processing is unnecessary, such as when waiting for an interrupt from key entries or peripheral circuits, place the device in standby mode to reduce current consumption.

Standby mode	Method to enter the mode	Circuits/functions stopped
Basic HALT mode	Execute the halt instruction after setting HLT2OP to "0". When the #BUSREQ signal is asserted from an external bus master while SEPD = "1".	CPU only
HALT2 mode	Execute the halt instruction after setting HLT2OP to "1".	CPU, BCU, bus clock, and DMA
SLEEP mode	Execute the slp instruction.	CPU, BCU, bus clock, DMA, high-speed (OSC3) oscillation circuit, prescaler, and peripheral circuits that use the prescaler output clocks

HLT2OP (D3/0x40150) that is used to select a HALT mode is set to "0" (basic HALT mode) at initial reset.

Notes:
- In systems in which DRAM is connected directly to the device, the refresh function is turned off during HALT2 and SLEEP modes.
- The standby mode is cleared by interrupt generation (except for the basic HALT mode, which is set using an external bus master). Therefore, before entering standby mode, set the related registers to allow an interrupt to be used to clear the standby mode to be generated.

The low-speed (OSC1) oscillation circuit and clock timer continue operating even during SLEEP mode. If they are unnecessary, these circuits can also be turned off.

Function	Control bit	"1"	"0"	Default
Low-speed (OSC1) oscillation ON/OFF control	SOSC1(D0)/ Power control register(0x40140)	ON	OFF	ON

Switching over the system clocks

Normally, the system is clocked by the high-speed (OSC3) oscillation clock. If high-speed operation is unnecessary, switch the system clock to the low-speed (OSC1) oscillation clock and turn off the high-speed (OSC3) oscillation circuit. This helps to reduce current consumption. However, if DRAM is connected directly to the device, note that the refresh function is also turned off.

Even during operation using the high-speed (OSC3) oscillation clock, power reduction can also be achieved through the use of a system clock derived from the OSC3 clock by dividing it (1/1, 1/2, 1/4, or 1/8).

Function	Control bit	"1"	"0"	Default
System clock switch over	CLKCHG(D2)/ Power control register(0x40140)	OSC3	OSC1	OSC3
High-speed (OSC3) oscillation ON/OFF control	SOSC3(D1)/ Power control register(0x40140)	ON	OFF	ON
System clock division ratio selection	CLKDT(D[7:6])/ Power control register(0x40140)	"11" = 1/8 "10" = 1/4 "01" = 1/2 "00" = 1/1		1/1

Turning off the prescaler and peripheral circuits

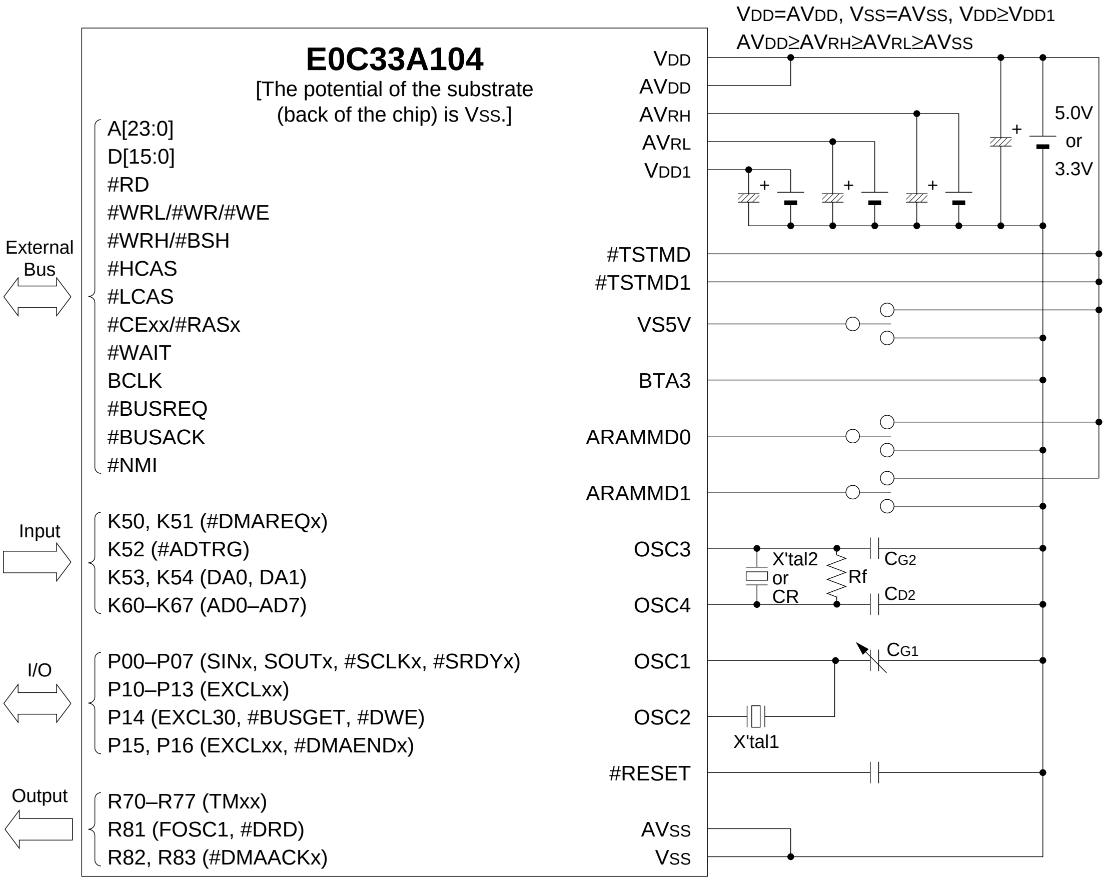
Current consumption can be reduced by turning off the peripheral circuits operating at high speed as much as possible. The circuits listed below are operated using a clock generated by a prescaler that is clocked by a high-speed (OSC3) clock:

- 16-bit programmable timers 0 to 5 (watchdog timer)
- 8-bit programmable timers 0 to 3 (DRAM refresh, serial interface)
- A/D converter

If none of these circuits need to be used, turn off the prescaler. If some of these circuits need to be used, turn off all other unnecessary circuits and stop clock supply from the prescaler to those circuits.

Function	Control bit	"1"	"0"	Default
Prescaler ON/OFF	PSCON(D5)/Power control register(0x40140)	ON	OFF	ON
16-bit timer 00 clock control	P16TON00(D3)/16-bit timer 0x clock control register(0x40147)	ON	OFF	OFF
16-bit timer 00 Run/Stop	PRUN00(D0)/16-bit timer 00 control register(0x40180)	RUN	STOP	STOP
16-bit timer 01 clock control	P16TON01(D7)/16-bit timer 0x clock control register(0x40147)	ON	OFF	OFF
16-bit timer 01 Run/Stop	PRUN01(D0)/16-bit timer 01 control register(0x40181)	RUN	STOP	STOP
16-bit timer 10 clock control	P16TON10(D3)/16-bit timer 1x clock control register(/0x40148)	ON	OFF	OFF
16-bit timer 10 Run/Stop	PRUN10(D0)/16-bit timer 10 control register(0x40188)	RUN	STOP	STOP
16-bit timer 11 clock control	P16TON11(D7)/16-bit timer 1x clock control register(0x40148)	ON	OFF	OFF
16-bit timer 11 Run/Stop	PRUN11(D0)/16-bit timer 11 control register(0x40189)	RUN	STOP	STOP
16-bit timer 20 clock control	P16TON20(D3)/16-bit timer 2x clock control register(0x40149)	ON	OFF	OFF
16-bit timer 20 Run/Stop	PRUN20(D0)/16-bit timer 20 control register(0x40190)	RUN	STOP	STOP
16-bit timer 21 clock control	P16TON21(D7)/16-bit timer 2x clock control register(0x40149)	ON	OFF	OFF
16-bit timer 21 Run/Stop	PRUN21(D0)/16-bit timer 21 control register(0x40191)	RUN	STOP	STOP
16-bit timer 30 clock control	P16TON30(D3)/16-bit timer 3x clock control register(0x4014A)	ON	OFF	OFF
16-bit timer 30 Run/Stop	PRUN30(D0)/16-bit timer 30 control register(0x40198)	RUN	STOP	STOP
16-bit timer 31 clock control	P16TON31(D7)/16-bit timer 3x clock control register(0x4014A)	ON	OFF	OFF
16-bit timer 31 Run/Stop	PRUN31(D0)/16-bit timer 31 control register(0x40199)	RUN	STOP	STOP
16-bit timer 40 clock control	P16TON40(D3)/16-bit timer 4x clock control register(0x4014B)	ON	OFF	OFF
16-bit timer 40 Run/Stop	PRUN40(D0)/16-bit timer 40 control register(0x401A0)	RUN	STOP	STOP
16-bit timer 41 clock control	P16TON41(D7)/16-bit timer 4x clock control register(0x4014B)	ON	OFF	OFF
16-bit timer 41 Run/Stop	PRUN41(D0)/16-bit timer 41 control register(0x401A1)	RUN	STOP	STOP
16-bit timer 50 clock control	P16TON50(D3)/16-bit timer 5x clock control register(0x4014C)	ON	OFF	OFF
16-bit timer 50 Run/Stop	PRUN50(D0)/16-bit timer 50 control register(0x401A8)	RUN	STOP	STOP
16-bit timer 51 clock control	P16TON51(D7)/16-bit timer 5x clock control register(0x4014C)	ON	OFF	OFF
16-bit timer 51 Run/Stop	PRUN51(D0)/16-bit timer 51 control register(0x401A9)	RUN	STOP	STOP
8-bit timer 0 clock control	P8TON0(D3)/8-bit timer 0/1 clock control register(0x4014D)	ON	OFF	OFF
8-bit timer 0 Run/Stop	PTRUN0(D0)/8-bit timer 0 control register(0x40160)	RUN	STOP	STOP
8-bit timer 1 clock control	P8TON1(D7)/8-bit timer 0/1 clock control register(0x4014D)	ON	OFF	OFF
8-bit timer 1 Run/Stop	PTRUN1(D0)/8-bit timer 1 control register(0x40164)	RUN	STOP	STOP
8-bit timer 2 clock control	P8TON2(D3)/8-bit timer 2/3 clock control register(0x4014E)	ON	OFF	OFF
8-bit timer 2 Run/Stop	PTRUN2(D0)/8-bit timer 2 control register(0x40168)	RUN	STOP	STOP
8-bit timer 3 clock control	P8TON3(D7)/8-bit timer 2/3 clock control register(0x4014E)	ON	OFF	OFF
8-bit timer 3 Run/Stop	PTRUN3(D0)/8-bit timer 3 control register(0x4016C)	RUN	STOP	STOP
A/D converter clock control	PSONAD(D3)/A/D clock control register(0x4014F)	ON	OFF	OFF
A/D conversion enable	ADE(D2)/A/D enable register(0x40244)	RUN	STOP	STOP

16 Basic External Wiring Diagram



X'tal1	Crystal oscillator	32.768 kHz, CI(Max.) = 34 kΩ
CG1	Trimmer capacitor	5–25 pF
X'tal2	Crystal oscillator	20 MHz (3.3 V) / 33 MHz (5.0 V)
CR	Ceramic oscillator	20 MHz (3.3 V) / 33 MHz (5.0 V)
CG2	Gate capacitor	5 pF
CD2	Drain capacitor	5 pF
Rf	Feedback resistor	1 MΩ

Note: The above table is simply an example, and is not guaranteed to work.

17 Precautions on Mounting

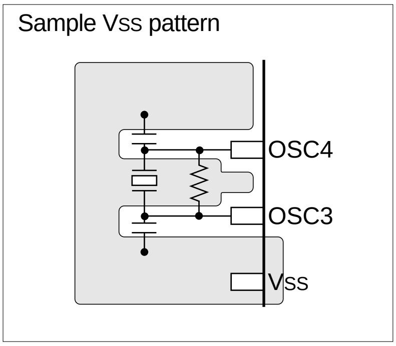
The following shows the precautions when designing the board and mounting the IC.

Oscillation Circuit

- Oscillation characteristics change depending on conditions (board pattern, components used, etc.). In particular, when a ceramic oscillator or crystal oscillator is used, use the oscillator manufacturer's recommended values for constants such as capacitance and resistance.
- Disturbances of the oscillation clock due to noise may cause a malfunction. Consider the following points to prevent this:

  (1) Components which are connected to the OSC3 (OSC1) and OSC4 (OSC2) pins, such as oscillators, resistors and capacitors, should be connected in the shortest line.

  (2) As shown in the figure below, make a VSS pattern as large as possible at circumscription of the OSC3 (OSC1) and OSC4 (OSC2) pins and the components connected to these pins.
  Furthermore, do not use this VSS pattern to connect other components than the oscillation system.



  (3) When supplying an external clock to the OSC3 (OSC1) pin, the clock source should be connected to the OSC3 (OSC1) pin in the shortest line.
  Furthermore, do not connect anything else to the OSC4 (OSC2) pin.

- In order to prevent unstable operation of the oscillation circuit due to current leak between OSC3 (OSC1) and VDD, please keep enough distance between OSC3 (OSC1) and VDD or other signals on the board pattern.

Reset Circuit

- The power-on reset signal which is input to the #RESET pin changes depending on conditions (power rise time, components used, board pattern, etc.). Decide the time constant of the capacitor and resistor after enough tests have been completed with the application product.
  When using the built-in pull-up resistor of the #RESET pin to configure a power-on reset circuit, take into consideration dispersion of the resistance for setting the constant.
- In order to prevent any occurrences of unnecessary resetting caused by noise during operating, components such as capacitors and resistors should be connected to the #RESET pin in the shortest line.

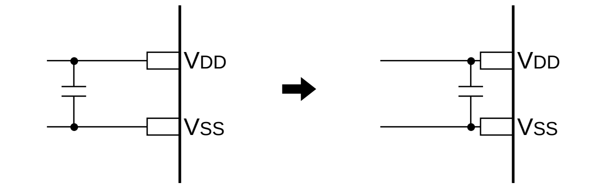
Power Supply Circuit

- Sudden power supply variation due to noise may cause malfunction. Consider the following points to prevent this:

  (1) The power supply should be connected to the VDD, VDD1, VSS, AVDD, AVSS, AVRH and AVRL pins with patterns as short and large as possible.
  In particular, the power supply for AVDD, AVSS, AVRH and AVRL affects A/D conversion precision.

(2) When connecting between the VDD and VSS pins with a bypass capacitor, the pins should be connected as short as possible.

Bypass capacitor connection example



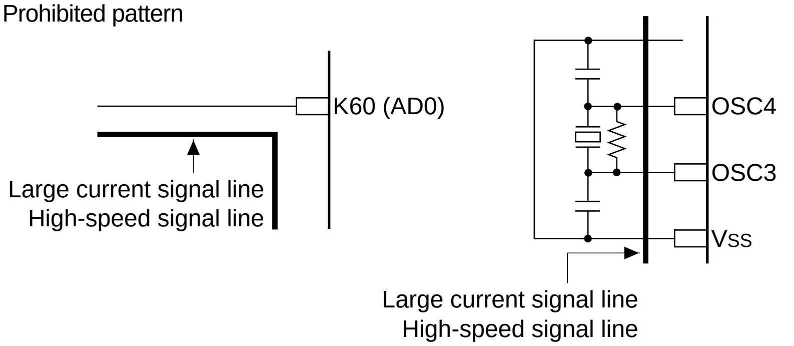
A/D Converter

- When the A/D converter is not used, the power supply pins for the analog system should be connected as shown below.
  AVDD → VDD
  AVSS, AVRH, AVRL → VSS

Arrangement of Signal Lines

- In order to prevent generation of electromagnetic induction noise caused by mutual inductance, do not arrange a large current signal line near the circuits that are sensitive to noise such as the oscillation unit and analog input unit.
- When a signal line is parallel with a high-speed line in long distance or intersects a high-speed line, noise may generated by mutual interference between the signals and it may cause a malfunction.
  Do not arrange a high-speed signal line especially near circuits that are sensitive to noise such as the oscillation unit and analog input unit.

Prohibited pattern



Precautions for Visible Radiation (when bare chip is mounted)

- Visible radiation causes semiconductor devices to change the electrical characteristics. It may cause this IC to malfunction. When developing products which use this IC, consider the following precautions to prevent malfunctions caused by visible radiation.

(1) Design the product and implement the IC on the board so that it is shielded from visible radiation in actual use.

(2) The inspection process of the product needs an environment that shields the IC from visible radiation.

(3) As well as the face of the IC, shield the back and side too.

18 Electrical Characteristics

18.1 Absolute Maximum Rating

(V_{SS}=0V)

Item	Symbol	Condition	Rated value	Unit	∗
Supply voltage	V_{DD}		-0.3 to +7.0	V	1
Input voltage	V_I		-0.3 to V_{DD}+0.3	V	
High-level output current	I_{OH}	1 pin	-10	mA	
		Total of all pins	-40	mA	
Low-level output current	I_{OL}	1 pin	10	mA	
		Total of all pins	40	mA	
Analog power voltage	AV_{DD}		-0.3 to +7.0	V	
High-level analog reference voltage	AV_{RH}		-0.3 to AV_{DD}+0.3	V	2
Low-level analog reference voltage	AV_{RL}		-0.3 to AV_{DD}+0.3	V	2
Analog input voltage	AV_{IN}		-0.3 to AV_{DD}+0.3	V	
Low-speed oscillation power voltage	V_{DD1}		-0.3 to +7.0	V	1
Low-speed oscillation input voltage	V_{I1}		-0.3 to V_{DD1}+0.3	V	
Low-speed oscillation output voltage	V_{O1}		-0.3 to V_{DD1}+0.3	V	
Operating temperature	T_{OPR}		-20 to +70	°C	
Storage temperature	T_{STG}		-65 to +150	°C	

∗ Note) 1. $V_{DD} \geq V_{DD1}$; $V_{DD} < V_{DD1}$ is prohibited

2. $AV_{RH} \geq AV_{RL}$; $AV_{RH} < AV_{RL}$ is prohibited

18.2 Recommended Operating Conditions

(V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	∗
Operating power voltage	V_{DD}	5V system	4.5	5.0	5.5	V	
		3.3V system	3.0	3.3	3.6	V	
Low-speed oscillation operating power voltage	V_{DD1}	5V system	4.5	5.0	5.5	V	
		3.3V system	3.0	3.3	3.6	V	
Operating clock frequency	f_{OSC3}	V_{DD}=5V±10%	>0		33	MHz	
		V_{DD}=3.3V±0.3V	>0		20	MHz	
Low-speed oscillation frequency	f_{OSC1}	V_{DD}=5V±10%		32.768		kHz	
		V_{DD}=3.3V±0.3V		32.768		kHz	

18.3 DC Characteristics

(Unless otherwise specified: V_{DD}=5.0V±10%, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
High-level input voltage 1	V_{IH1}	OSC1, OSC3 pins	3.6			V	
High-level input voltage 2	V_{IH2}	Other input pins (VS5V pin=L)	2.8			V	
		Other input pins (VS5V pin=H)	2.0			V	
Low-level input voltage 1	V_{IL1}	OSC1, OSC3 pins			1.4	V	
Low-level input voltage 2	V_{IH2}	Other input pins (VS5V pin=L)			0.8	V	
		Other input pins (VS5V pin=H)			0.6	V	
High-level schmitt input voltage 1	V_{T1+}	#RESET pin	4.0			V	
Low-level schmitt input voltage 1	V_{T1-}	#RESET pin			0.8	V	
High-level schmitt input voltage 2	V_{T2+}	#NMI pin	3.6			V	
Low-level schmitt input voltage 2	V_{T2-}	#NMI pin			0.8	V	
Schmitt input voltage difference 1	ΔV_{T1}	(V_{T1+})-(V_{T1-}): #RESET pin	0.4			V	
Schmitt input voltage difference 2	ΔV_{T2}	(V_{T2+})-(V_{T2-}): #NMI pin	0.4			V	
High-level output voltage	V_{OH}	eBUS_OUT[#1], Rxx, Pxx, BCLK, #BUSACK, I_{OH}=-0.3mA	V_{DD} - 0.4			V	
Low-level output voltage	V_{OL}	eBUS_OUT[#1], Rxx, Pxx, BCLK, #BUSACK, I_{OL}=2mA			0.4	V	
Input leakage current	I_{LI}	eBUS_IN[#2], Kxx, Pxx	-1		1	μA	
Output leakage current	I_{LO}	eBUS_OUT[#1], Rxx, Pxx	-1		1	μA	
Input pull-up resistor	R_{IN}	Kxx, Pxx, #RESET, Ta=25°C	50	100	200	kΩ	
Input pin capacitance	C_{IN}	Kxx, Pxx, V_{IN}=0V, f=1MHz, Ta=25°C			15	pF	

#1 eBUS_OUT: [23:0], #RD, #WRL, #WRH, #HCAS, #LCAS, #CE[17:4], D[15:0]
#2 eBUS_IN: BTA3, #NMI, #WAIT, #BUSREQ

(Unless otherwise specified: V_{DD}=3.3±0.3V, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
High-level input voltage 1	V_{IH1}	OSC1, OSC3 pins	2.4			V	
High-level input voltage 2	V_{IH2}	Other input pins (VS5V pin=L)	1.8			V	
Low-level input voltage 1	V_{IL1}	OSC1, OSC3 pins			0.6	V	
Low-level input voltage 2	V_{IL2}	Other input pins (VS5V pin=L)			0.6	V	
High-level schmitt input voltage 1	V_{T1+}	#RESET pin	2.4			V	
Low-level schmitt input voltage 1	V_{T1-}	#RESETpin			0.6	V	
High-level schmitt input voltage 2	V_{T2+}	#NMI pin	2.2			V	
Low-level schmitt input voltage 2	V_{T2-}	#NMI pin			0.6	V	
Schmitt input voltage difference 1	ΔV_{T1}	(V_{T1+})-(V_{T1-}): #RESET pin	0.2			V	
Schmitt input voltage difference 2	ΔV_{T2}	(V_{T2+})-(V_{T2-}): #NMI pin	0.2			V	
High-level output voltage	V_{OH}	eBUS_OUT[#1], Rxx, Pxx, BCLK, #BUSACK, I_{OH}=-0.2mA	V_{DD} - 0.4			V	
Low-level output voltage	V_{OL}	eBUS_OUT[#1], Rxx, Pxx, BCLK, #BUSACK, I_{OL}=1.6mA			0.4	V	
Input leakage current	I_{LI}	eBUS_IN[#2], Kxx, Pxx	-1		1	μA	
Output leakage current	I_{LO}	eBUS_OUT[#1], Rxx, Pxx	-1		1	μA	
Input pull-up resistor	R_{IN}	Kxx, Pxx, #RESET, Ta=25°C	80	160	320	kΩ	
Input pin capacitance	C_{IN}	Kxx, Pxx, V_{IN}=0V, f=1MHz, Ta=25°C			15	pF	

#1 eBUS_OUT: [23:0], #RD, #WRL, #WRH, #HCAS, #LCAS, #CE[17:4], D[15:0]
#2 eBUS_IN: BTA3, #NMI, #WAIT, #BUSREQ

18.4 Current Consumption

(Unless otherwise specified: V_{DD}=5.0V, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	∗
Operating current	I_{DD1}	When CPU is operating (32.768kHz)		90	250	µA	1
	I_{DD2}	When CPU is operating (33MHz)		80	110	mA	2
	I_{DD3}	HALT mode (33MHz)		40	60	mA	3
	I_{DD4}	SLEEP mode		1	3	µA	4
	I_{DD5}	Power-down mode (33MHz)		40	60	mA	5
Analog power voltage	AI_{DD1}	During A/D conversion (2MHz)		650	1500	µA	6
	AI_{DD2}	During D/A conversion		1.5	3	mA	7
Analog reference power current	AI_{DR1}	During A/D conversion		170	250	µA	6
Clock timer operation current	I_{DDCT}	When clock timer only is operating OSC1 oscillation: 32kHz		12	30	µA	

(Unless otherwise specified: V_{DD}=3.3V, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	∗
Operating current	I_{DD1}	When CPU is operating (32.768kHz)		50	150	µA	1
	I_{DD2}	When CPU is operating (20MHz)		30	40	mA	2
	I_{DD3}	HALT mode (20MHz)		12	20	mA	3
	I_{DD4}	SLEEP mode		1	3	µA	4
	I_{DD5}	Power-down mode (20MHz)		12	20	mA	5
Analog power voltage	AI_{DD1}	During A/D conversion (2MHz)		200	500	µA	6
	AI_{DD2}	During D/A conversion		300	800	µA	7
Analog reference power current	AI_{DR1}	During A/D conversion		100	160	µA	6
Clock timer operation current	I_{DDCT}	When clock timer only is operating OSC1 oscillation: 32kHz		3	10	µA	

Current consumption measurement condition:

V_{IH}=V_{DD}-0.5V, V_{IL}=V_{SS}+0.5V, output pins are open, no pull-up resistor
OSC1: 32kHz external clock input

∗ Note)

No.	OSC3	OSC1	CPU	Clock timer	Other peripheral circuits
1	Off	On	32.768kHz operation ∗1	Run	Stop
2	On	Off	20MHz operation ∗1	Stop	Stop
3	On	Off	HALT mode	Stop	Stop
4	Off	Off	SLEEP mode	Stop	Stop
5	On	Off	Power-down mode by #BUSREQ	Stop	Stop
6	On	Off	HALT mode	Stop	A/D converter only operated, conversion clock frequency=2MHz (AV_{RH}=V_{DD}, AV_{RL}=V_{SS})
7	On	Off	HALT mode	Stop	1 channel of D/A converter only operated

∗1: The values of current consumption during execution were measured when a test program that consists of 55% load instructions, 23% arithmetic operation instructions, 1% mac instruction, 12% branch instructions and 9% ext instruction is being executed continuously.

18.5 A/D Converter Characteristics

(Unless otherwise specified: V_{DD}=AV_{DD}=5.0V±10%, V_{SS}=AV_{SS}=0V, Ta=-20 to +70°C,
A/D clock f=2MHz, ST[1:0]=11)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	∗
Resolution	–		–	10	–	bit	
Conversion time	t_{ADC}	ST[1:0]=00(Min.), 11(Max.)	7	–	10	µS	
Zero scale error	E_{ZS}		2	3	4	LSB	1
Full scale error	E_{FS}		-3	-1	0	LSB	1
Integral lenearity error	E_{IL}	Best straight line method	-3	–	3	LSB	1
Differential lenearity error	E_{DL}		-3	–	3	LSB	1
Permissible signal source impedance	A_{IMP}		–	–	5	kΩ	
Analog input capacitance	AC_{IN}		–	–	45	pF	

∗ Note) 1. The above characteristics indicate the accuracy when AV_{RH}=AV_{DD} and AV_{RL}=AV_{SS}.
When AV_{RH} and AV_{RL} are set as (AV_{DD}–AV_{SS})>(AV_{RH}–AV_{RL}), conversion accuracy may be degraded.

(Unless otherwise specified: V_{DD}=AV_{DD}=3.3±0.3V, V_{SS}=AV_{SS}=0V, Ta=-20 to +70°C, A/D clock f=2MHz, ST[1:0]=11)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	∗
Resolution	–		–	10	–	bit	
Conversion time	t_{ADC}	ST[1:0]=00(Min.), 11(Max.)	7	–	10	µS	
Zero scale error	E_{ZS}		1.5	2.5	3.5	LSB	1
Full scale error	E_{FS}		-3	-1	0	LSB	1
Integral lenearity error	E_{IL}	Best straight line method	-3	–	3	LSB	1
Differential lenearity error	E_{DL}		-3	–	3	LSB	1
Permissible signal source impedance	A_{IMP}		–	–	5	kΩ	
Analog input capacitance	AC_{IN}		–	–	45	pF	

∗ Note) 1. The above characteristics indicate accuracy when AV_{RH}=AV_{DD} and AV_{RL}=AV_{SS}.
When AV_{RH} and AV_{RL} are set as (AV_{DD}–AV_{SS})>(AV_{RH}–AV_{RL}), conversion accuracy may be degraded.

18.6 D/A Converter Characteristics

(Unless otherwise specified: V_{DD}=AV_{DD}=5.0V±10%, V_{SS}=AV_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	∗
Resolution	–		–	8	–	bit	
Zero scale error	E_{ZS}		-2	–	2	LSB	1
Full scale error	E_{FS}		-2	–	2	LSB	1
Integral lenearity error	E_{IL}	Best straight line method	-2	–	2	LSB	1
Differential lenearity error	E_{DL}		-2	–	2	LSB	1
Settling time	t_{SET}		–	–	10	µS	
Permissible load condition	R_L	Load resistance	2	–	–	kΩ	

∗ Note) 1. The ideal characteristic (DC best straight line) of this D/A converter is a line that starts at 1-LSB output voltage (input code = 00) and ends at 255-LSB output voltage (input code = FF).
Accuracy is guaranteed within the output voltage range from AV_{SS}+AV_{DD}∗10% to AV_{DD}-AV_{DD}∗10%.
The characteristics are measured using a load resistance of 10 kΩ.

(Unless otherwise specified: V_{DD}=AV_{DD}=3.3±0.3V, V_{SS}=AV_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	∗
Resolution	–		–	8	–	bit	
Zero scale error	E_{ZS}		-2	–	2	LSB	1
Full scale error	E_{FS}		-2	–	2	LSB	1
Integral lenearity error	E_{IL}	Best straight line method	-2	–	2	LSB	1
Differential lenearity error	E_{DL}		-2	–	2	LSB	1
Settling time	t_{SET}		–	–	50	µS	
Permissible load condition	R_L	Load resistance	2	–	–	kΩ	

∗ Note) 1. The ideal characteristic (DC best straight line) of this D/A converter is a line that starts at 1-LSB output voltage (input code = 00) and ends at 255-LSB output voltage (input code = FF).
Accuracy is guaranteed within the output voltage range from AV_{SS}+AV_{DD}∗10% to AV_{DD}-AV_{DD}∗10%.
The characteristics are measured using a load resistance of 10 kΩ.

18.7 AC Characteristics

18.7.1 Symbol Description

t_{CYC}: Clock cycle time

50 nS when the device is operated with a 20-MHz clock, or 30 nS when operated with a 33-MHz clock.

WC: Number of wait cycles

Up to 7 cycles can be set for the number of cycles using the BCU control register. Furthermore, it can be extended to a desired number of cycles by setting the #WAIT pin from outside of the IC.

The minimum number of read cycles with no wait (0) inserted is 1 cycle.

The minimum number of write cycles with no wait cycle (0) inserted is 2 cycles. It does not change even if 1-wait cycle is set. The write cycle is actually extended when 2 or more wait cycles are set.

When inserting wait cycles by controlling the #WAIT pin from outside of the IC, pay attention to the timing of the #WAIT signal sampling. Read cycles are terminated at the cycle in which the #WAIT signal is negated. Write cycles are terminated at the following cycle after the #WAIT signal is negated.

C1, C2, C3, Cn: Cycle number

C1 indicates the first cycle when the BCU transfers data from/to an external memory or another device. Similarly, C2 and Cn indicate the second cycle and nth cycle, respectively.

Cw: Wait cycle

Indicates that the cycle is wait cycle inserted.

18.7.2 AC Characteristics Measurement Condition

Signal detection level:	Input signal	High level	V_{IH} = 2.4 V
		Low level	V_{IL} = 0.4 V
	Output signal	High level	V_{OH} = 2.0 V
		Low level	V_{OL} = 0.8 V
	The following applies to clock duty:		
	Input signal	High level	V_{IH} = 1.6 V
		Low level	V_{IL} = 1.6 V
	Output signal	High level	V_{OH} = 1.6 V
		Low level	V_{OL} = 1.6 V
Input signal waveform:	Rise time (10% → 90% V_{DD})	5 nS	
	Fall time (90% → 10% V_{DD})	5 nS	
Output load capacitance:	C_L = 50 pF		

18.7.3 AC Characteristic Tables

External clock input characteristics

(Note) These AC characteristics apply to input signals from outside the IC.

(Unless otherwise specified: V_{DD}=5.0V±10%, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
High-speed clock cycle time	t_{C3}	30		nS	
OSC3 clock input duty	t_{C3ED}	45	55	%	
OSC3 clock input rise time	t_{IF}		5	nS	
OSC3 clock input fall time	t_{IR}		5	nS	
BCLK high level output delay time	t_{CD1}		35	nS	
BCLK low-level output delay time	t_{CD2}		35	nS	
OSC1 clock input cycle time	t_{C1}	1000		nS	
OSC1 clock input duty	t_{C1ED}	35	65	%	
Minimum reset pulse width	t_{RST}	$6 \cdot t_{CYC}$		nS	

(Unless otherwise specified: V_{DD}=3.3±0.3V, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
High-speed clock cycle time	t_{C3}	50		nS	
OSC3 clock input duty	t_{C3ED}	45	55	%	
OSC3 clock input rise time	t_{IF}		5	nS	
OSC3 clock input fall time	t_{IR}		5	nS	
BCLK high level output delay time	t_{CD1}		40	nS	
BCLK low-level output delay time	t_{CD2}		40	nS	
OSC1 clock input cycle time	t_{C1}	1000		nS	
OSC1 clock input duty	t_{C1ED}	35	65	%	
Minimum reset pulse width	t_{RST}	$6 \cdot t_{CYC}$		nS	

BCLK clock output characteristics

(Note) These AC characteristic values are applied only when the high-speed oscillation circuit is used.

(Unless otherwise specified: V_{DD}=5.0V±10%, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
BCLK clock output duty	t_{CBD}	40	60	%	

(Unless otherwise specified: V_{DD}=3.3±0.3V, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
BCLK clock output duty	t_{CBD}	40	60	%	

Common characteristics

(Unless otherwise specified: V_{DD}=5.0V±10%, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
Address delay time	t_{AD}	–	8	nS	
#CEx delay time (1)	t_{CE1}	–	8	nS	
#CEx delay time (2)	t_{CE2}	–	8	nS	
Wait setup time	t_{WTS}	15	–	nS	
Wait hold time	t_{WTH}	0	–	nS	
Read signal delay time (1)	t_{RDD1}		8	nS	
Read data setup time	t_{RDS}	12		nS	
Read data hold time	t_{RDH}	0		nS	
Write signal delay time (1)	t_{WRD1}		8	nS	
Write data delay time (1)	t_{WDD1}		10	nS	
Write data delay time (2)	t_{WDD2}	0	10	nS	
Write data hold time	t_{WDH}	0		nS	

(Unless otherwise specified: V_{DD}=3.3±0.3V, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
Address delay time	t_{AD}	–	10	nS	
#CEx delay time (1)	t_{CE1}	–	10	nS	
#CEx delay time (2)	t_{CE2}	–	10	nS	
Wait setup time	t_{WTS}	15	–	nS	
Wait hold time	t_{WTH}	0	–	nS	
Read signal delay time (1)	t_{RDD1}		10	nS	
Read data setup time	t_{RDS}	15		nS	
Read data hold time	t_{RDH}	0		nS	
Write signal delay time (1)	t_{WRD1}		10	nS	
Write data delay time (1)	t_{WDD1}		10	nS	
Write data delay time (2)	t_{WDD2}	0	10	nS	
Write data hold time	t_{WDH}	0		nS	

SRAM read cycle

(Unless otherwise specified: V_{DD}=5.0V±10%, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
Read signal delay time (2)	t_{RDD2}		8	nS	
Read signal pulse width	t_{RDW}	t_{CYC}(0.5+WC)-10		nS	
Read address access time (1)	t_{ACC1}		t_{CYC}(1+WC)-20	nS	
Chip enable access time (1)	t_{CEAC1}		t_{CYC}(1+WC)-20	nS	
Read signal access time (1)	t_{RDAC1}		t_{CYC}(0.5+WC)-20	nS	

(Unless otherwise specified: V_{DD}=3.3±0.3V, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
Read signal delay time (2)	t_{RDD2}		10	nS	
Read signal pulse width	t_{RDW}	t_{CYC}(0.5+WC)-10		nS	
Read address access time (1)	t_{ACC1}		t_{CYC}(1+WC)-25	nS	
Chip enable access time (1)	t_{CEAC1}		t_{CYC}(1+WC)-25	nS	
Read signal access time (1)	t_{RDAC1}		t_{CYC}(0.5+WC)-25	nS	

SRAM write cycle

(Unless otherwise specified: V_{DD}=5.0V±10%, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
Write signal delay time (2)	t_{WRD2}		8	nS	
Write signal pulse width	t_{WRW}	t_{CYC}(1+WC)-10		nS	

(Unless otherwise specified: V_{DD}=3.3±0.3V, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
Write signal delay time (2)	t_{WRD2}		10	nS	
Write signal pulse width	t_{WRW}	t_{CYC}(1+WC)-10		nS	

DRAM access cycle common characteristics

(Unless otherwise specified: V_{DD}=5.0V±10%, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
#RAS signal delay time (1)	t_{RASD1}		10	nS	
#RAS signal delay time (2)	t_{RASD2}		10	nS	
#RAS signal pulse width	t_{RASW}	t_{CYC}(2+WC)-10		nS	
#CAS signal delay time (1)	t_{CASD1}		10	nS	
#CAS signal delay time (2)	t_{CASD2}		10	nS	
#CAS signal pulse width	t_{CASW}	t_{CYC}(0.5+WC)-5		nS	
Read signal delay time (3)	t_{RDD3}		10	nS	
Read signal pulse width (2)	t_{RDW2}	t_{CYC}(2+WC)-10		nS	
Write signal delay time (3)	t_{WRD3}		10	nS	
Write signal pulse width (2)	t_{WRW2}	t_{CYC}(2+WC)-10		nS	

(Unless otherwise specified: V_{DD}=3.3±0.3V, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
#RAS signal delay time (1)	t_{RASD1}		10	nS	
#RAS signal delay time (2)	t_{RASD2}		10	nS	
#RAS signal pulse width	t_{RASW}	t_{CYC}(2+WC)-10		nS	
#CAS signal delay time (1)	t_{CASD1}		10	nS	
#CAS signal delay time (2)	t_{CASD2}		10	nS	
#CAS signal pulse width	t_{CASW}	t_{CYC}(0.5+WC)-10		nS	
Read signal delay time (3)	t_{RDD3}		10	nS	
Read signal pulse width (2)	t_{RDW2}	t_{CYC}(2+WC)-10		nS	
Write signal delay time (3)	t_{WRD3}		10	nS	
Write signal pulse width (2)	t_{WRW2}	t_{CYC}(2+WC)-10		nS	

DRAM random access cycle and DRAM fast-page cycle

(Unless otherwise specified: V_{DD}=5.0V±10%, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
Column address access time	t_{ACCF}		t_{CYC}(1+WC)-25	nS	
#RAS access time	t_{RACF}		t_{CYC}(1.5+WC)-25	nS	
#CAS access time	t_{CACF}		t_{CYC}(0.5+WC)-25	nS	

(Unless otherwise specified: V_{DD}=3.3±0.3V, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
Column address access time	t_{ACCF}		t_{CYC}(1+WC)-25	nS	
#RAS access time	t_{RACF}		t_{CYC}(1.5+WC)-25	nS	
#CAS access time	t_{CACF}		t_{CYC}(0.5+WC)-25	nS	

EDO DRAM random access cycle and EDO DRAM page cycle

(Unless otherwise specified: V_{DD}=5.0V±10%, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
Column address access time	t_{ACCE}		t_{CYC}(1.5+WC)-25	nS	
#RAS access time	t_{RACE}		t_{CYC}(2+WC)-25	nS	
#CAS access time	t_{CACE}		t_{CYC}(1+WC)-15	nS	
Read data setup time	t_{RDS2}	20		nS	

(Unless otherwise specified: V_{DD}=3.3±0.3V, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
Column address access time	t_{ACCE}		t_{CYC}(1.5+WC)-25	nS	
#RAS access time	t_{RACE}		t_{CYC}(2+WC)-25	nS	
#CAS access time	t_{CACE}		t_{CYC}(1+WC)-20	nS	
Read data setup time	t_{RDS2}	20		nS	

Burst ROM read cycle

(Unless otherwise specified: V_{DD}=5.0V±10%, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
Read address access time (2)	t_{ACC2}		t_{CYC}(1+WC)-20	nS	
Chip enable access time (2)	t_{CEAC2}		t_{CYC}(1+WC)-20	nS	
Read signal access time (2)	t_{RDAC2}		t_{CYC}(0.5+WC)-20	nS	
Burst address access time	t_{ACCB}		t_{CYC}(1+WC)-20	nS	

(Unless otherwise specified: V_{DD}=3.3±0.3V, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
Read address access time (2)	t_{ACC2}		t_{CYC}(1+WC)-25	nS	
Chip enable access time (2)	t_{CEAC2}		t_{CYC}(1+WC)-25	nS	
Read signal access time (2)	t_{RDAC2}		t_{CYC}(0.5+WC)-25	nS	
Burst address access time	t_{ACCB}		t_{CYC}(1+WC)-25	nS	

External bus master and NMI

(Unless otherwise specified: V_{DD}=5.0V±10%, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
#BUSREQ signal setup time	t_{BRQS}	15		nS	
#BUSREQ signal hold time	t_{BRQH}	0		nS	
#BUSACK signal output delay time	t_{BAKD}		10	nS	
High-impedance → output delay time	t_{Z2E}		10	nS	
Output → high-impedance delay time	t_{B2Z}		10	nS	
#NMI pulse width	t_{NMIW}	30		nS	

(Unless otherwise specified: V_{DD}=3.3±0.3V, V_{SS}=0V, Ta=-20 to +70°C)

Item	Symbol	Min.	Max.	Unit	*
#BUSREQ signal setup time	t_{BRQS}	15		nS	
#BUSREQ signal hold time	t_{BRQH}	0		nS	
#BUSACK signal output delay time	t_{BAKD}		10	nS	
High-impedance → output delay time	t_{Z2E}		10	nS	
Output → high-impedance delay time	t_{B2Z}		10	nS	
#NMI pulse width	t_{NMIW}	30		nS	

Input, Output and I/O port

(Unless otherwise specified: V_{DD}=5.0V±10%, V_{SS}=0V, Ta=-20 to +70°C)

Item		Symbol	Min.	Max.	Unit	*
Input data setup time		t_{INPS}	20		nS	
Input data hold time		t_{INPH}	10		nS	
Output data delay time		t_{OUTD}		20	nS	
K-port interrupt input pulse width	SLEEP, HALT2 mode	t_{KINW}	30		nS	
	Others		2 × t_{CYC}		nS	

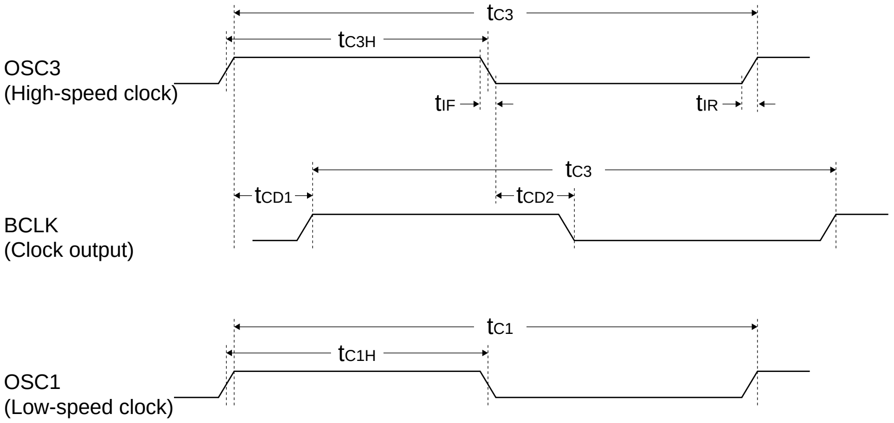
(Unless otherwise specified: V_{DD}=3.3±0.3V, V_{SS}=0V, Ta=-20 to +70°C)

Item		Symbol	Min.	Max.	Unit	*
Input data setup time		t_{INPS}	20		nS	
Input data hold time		t_{INPH}	10		nS	
Output data delay time		t_{OUTD}		20	nS	
K-port interrupt input pulse width		t_{KINW}	30		nS	
K-port interrupt input pulse width	SLEEP, HALT2 mode	t_{KINW}	30		nS	
	Others		2 × t_{CYC}		nS	

18.7.4 AC Characteristic Timing Charts

Clock

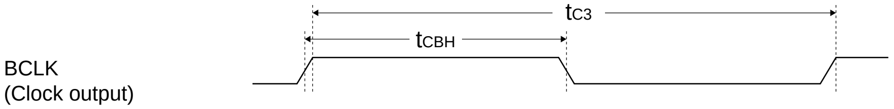
(1) When an external clock is input:



$t_{C3ED} = t_{C3H}/t_{C3}$

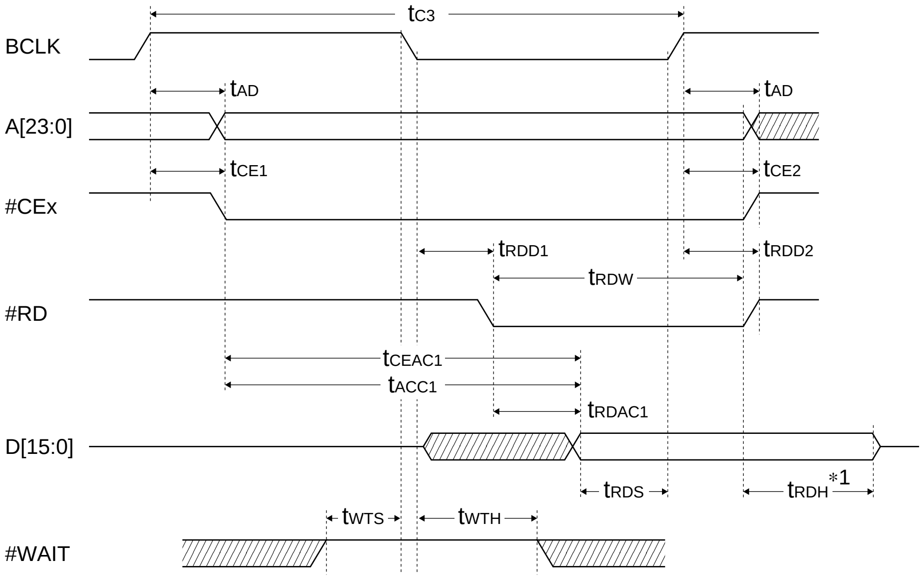
$t_{C1ED} = t_{C1H}/t_{C1}$

(2) When the high-speed oscillation circuit is used for the operating clock:



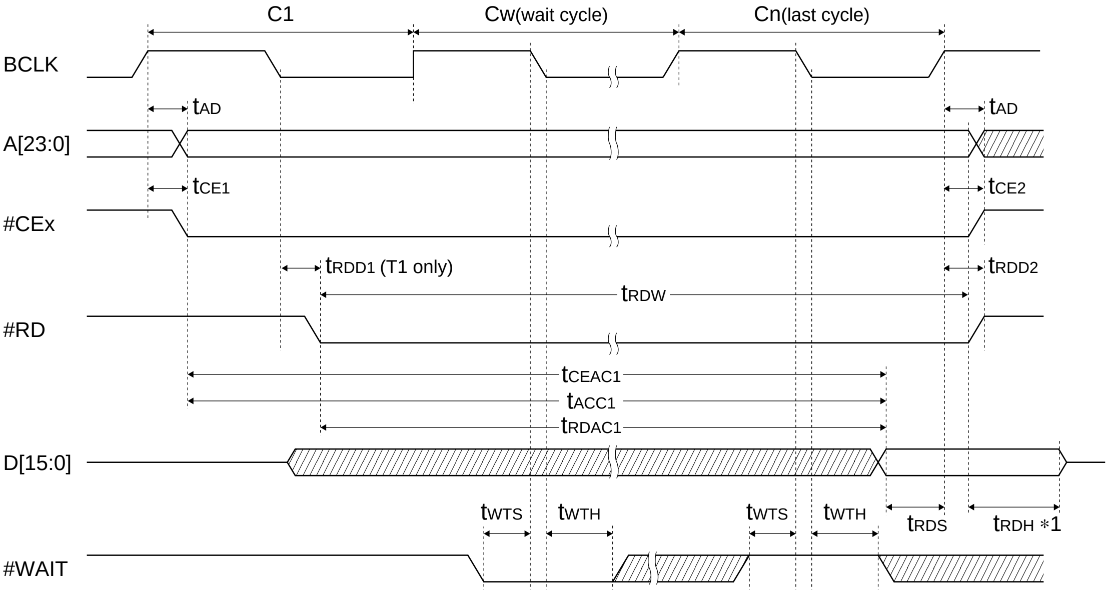
$t_{CBD} = t_{CBH}/t_{C3}$

SRAM read cycle (basic cycle: 1 cycle)



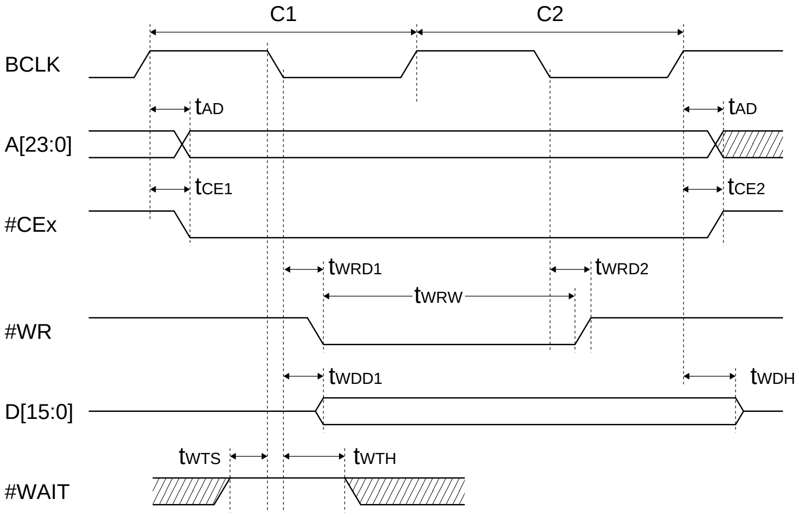
∗1 tRDH is measured with respect to the first signal change (negation) from among the #RD, #CEx and A[23:0] signals.

SRAM read cycle (when a wait cycle is inserted)

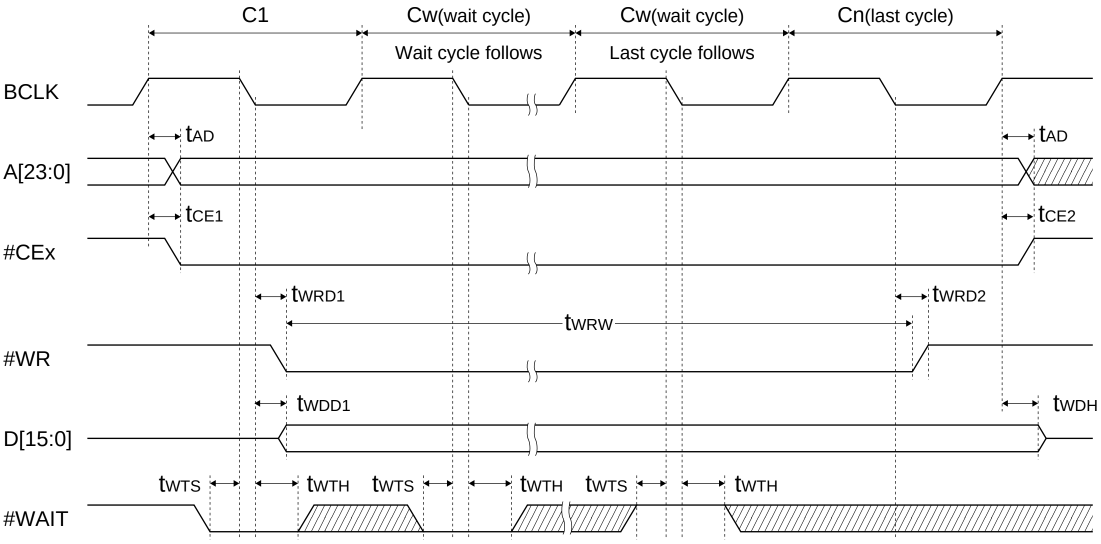


∗1 tRDH is measured with respect to the first signal change (negation) from among the #RD, #CEx and A[23:0] signals.

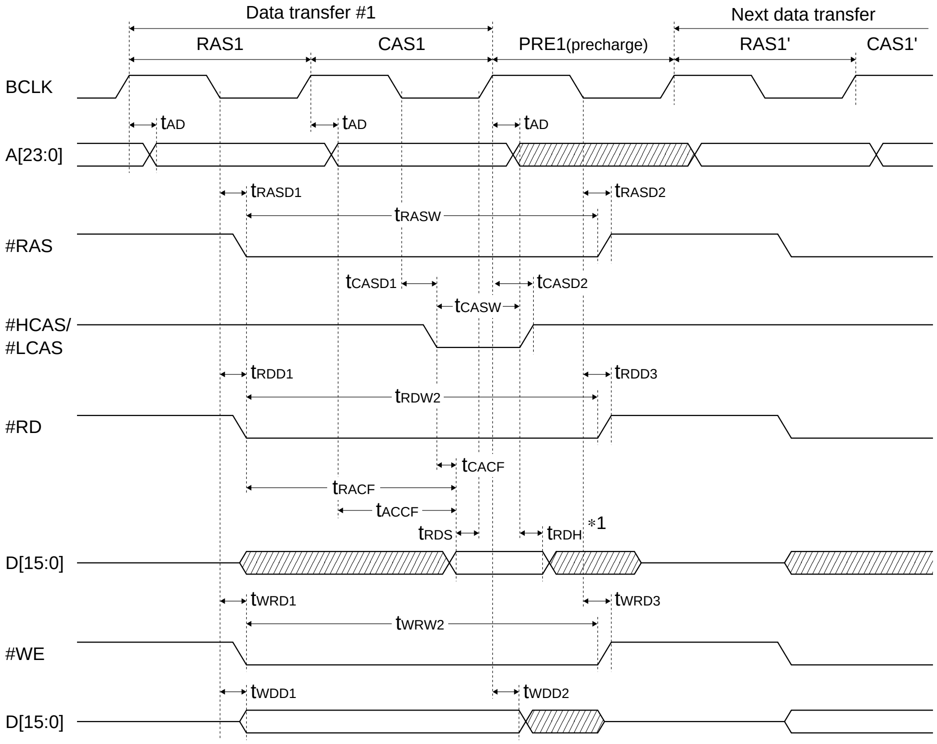
SRAM write cycle (basic cycle: 2 cycles)



SRAM write cycle (when wait cycles are inserted)

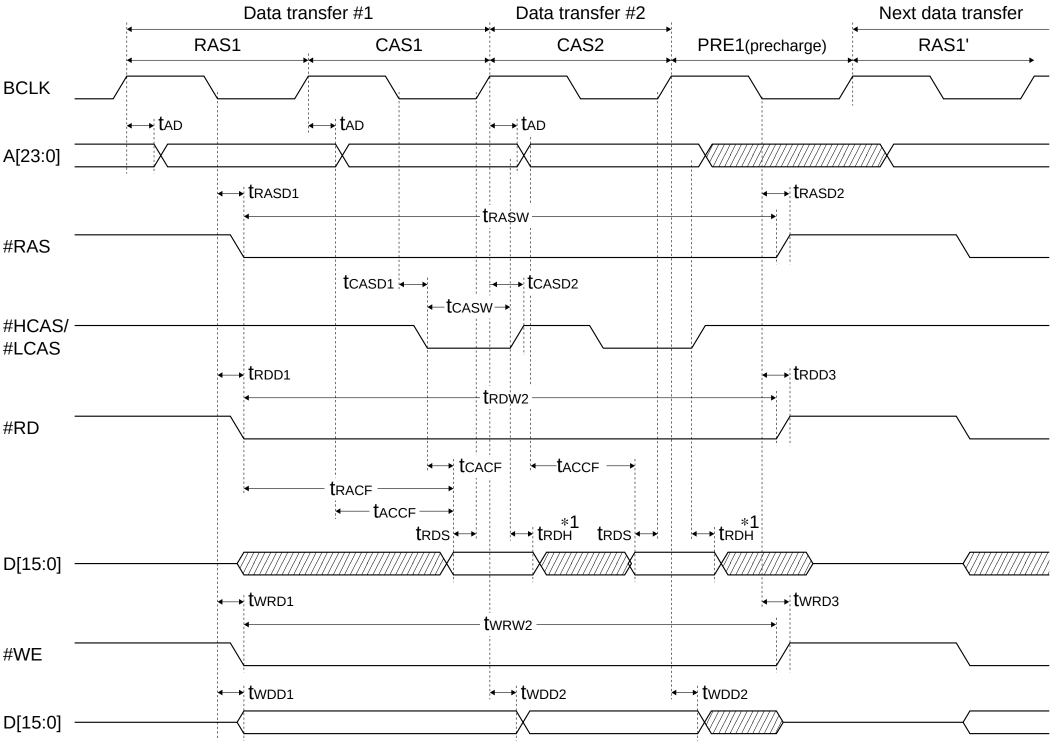


DRAM random access cycle (basic cycle)



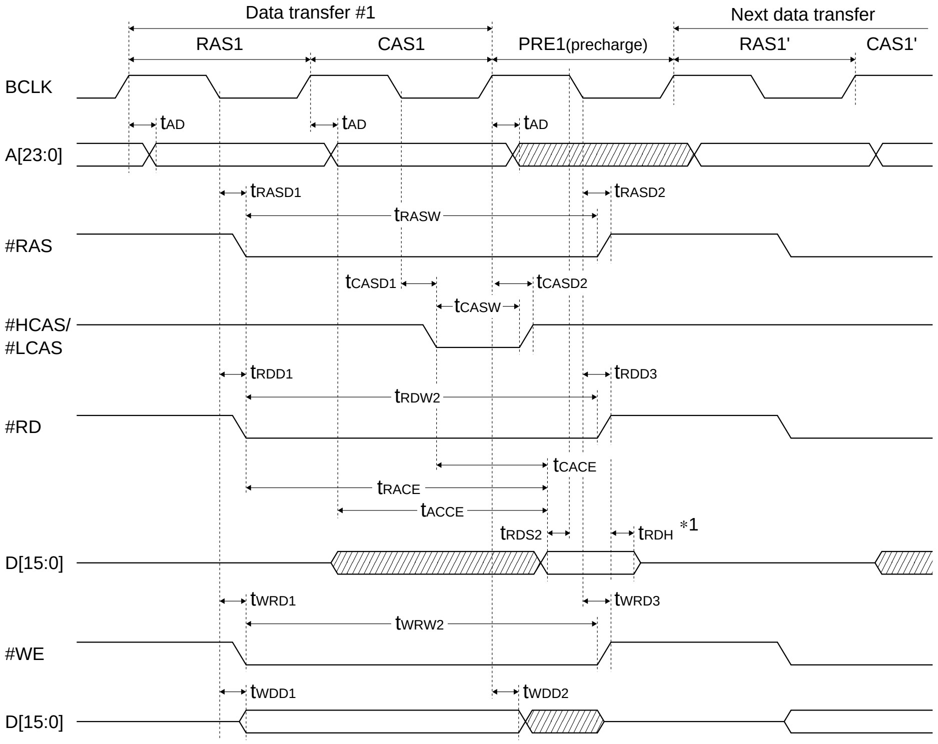
∗1 t_{RDH} is measured with respect to the first signal change (negation) of either the #RD or the A[23:0] signals.

DRAM fast-page access cycle



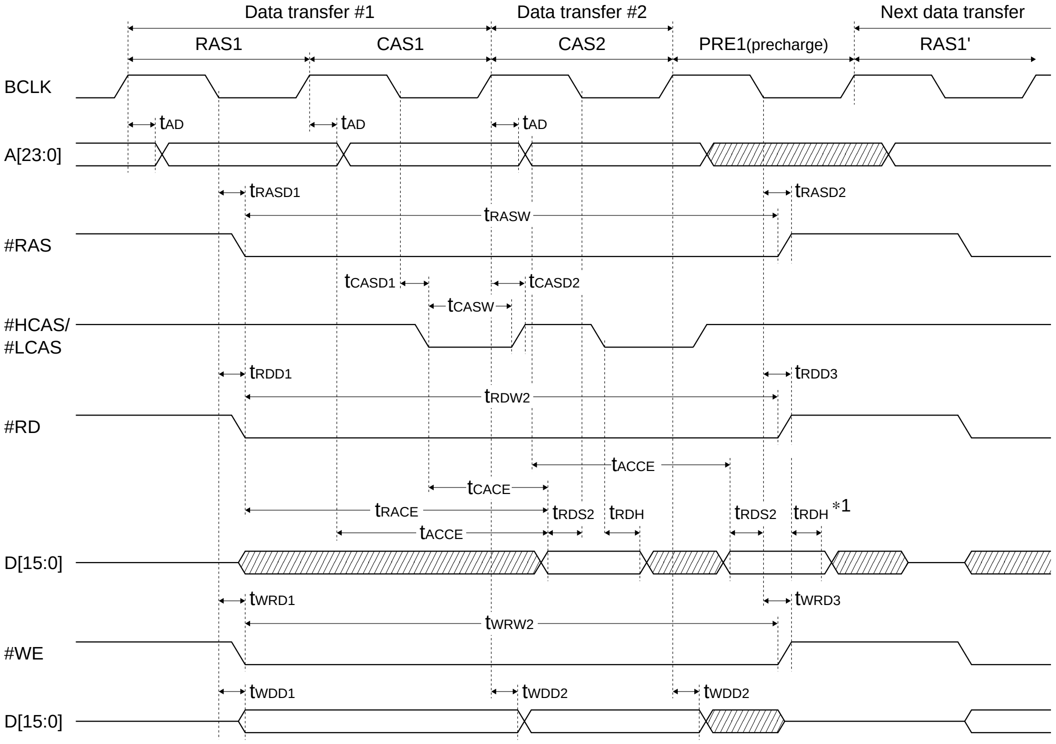
∗1 t_{RDH} is measured with respect to the first signal change (negation) of either the #RD or the A[23:0] signals.

EDO DRAM random access cycle (basic cycle)



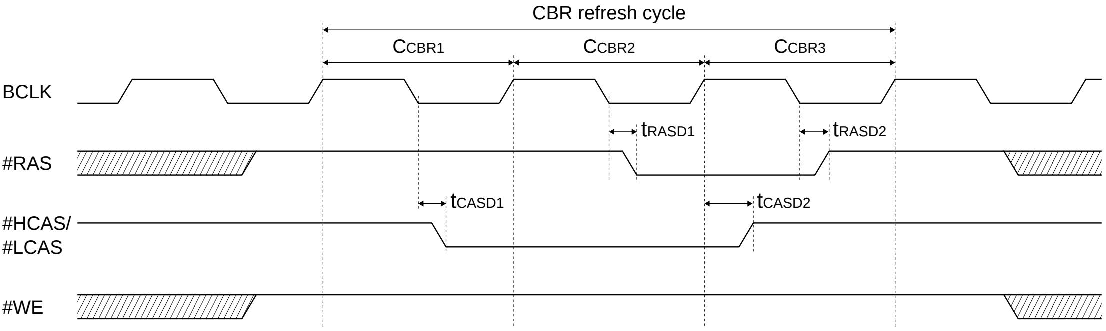
*1 tRDH is measured with respect to the first signal change (negation) of either the #RD or the #RASx signals.

EDO DRAM page access cycle

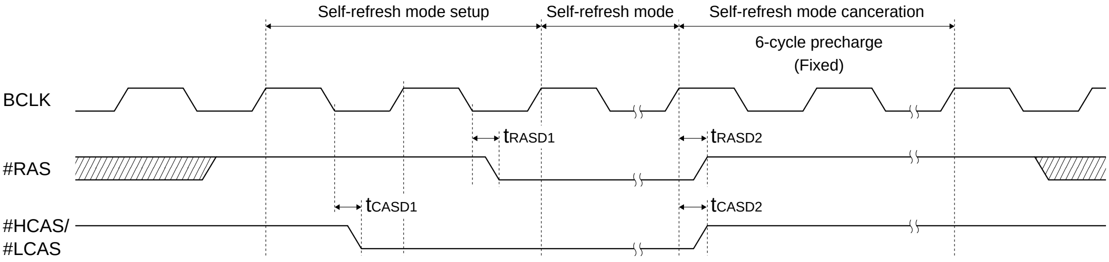


*1 tRDH is measured with respect to the first signal change from among the #RD (negation), #RASx (negation) and #CAS (rise) signals.

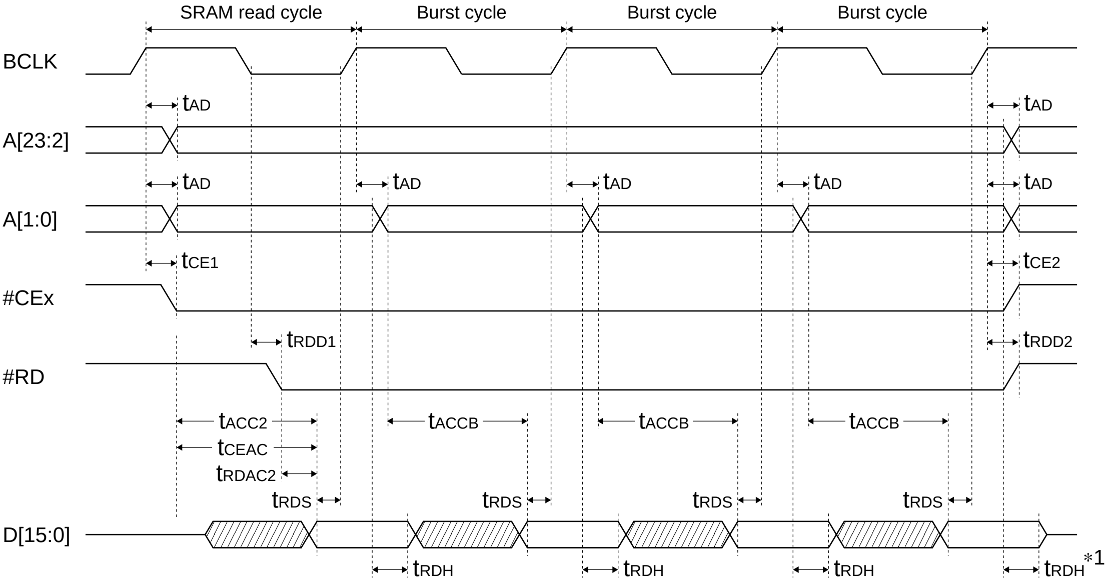
DRAM CAS-before-RAS refresh cycle



DRAM self-refresh cycle

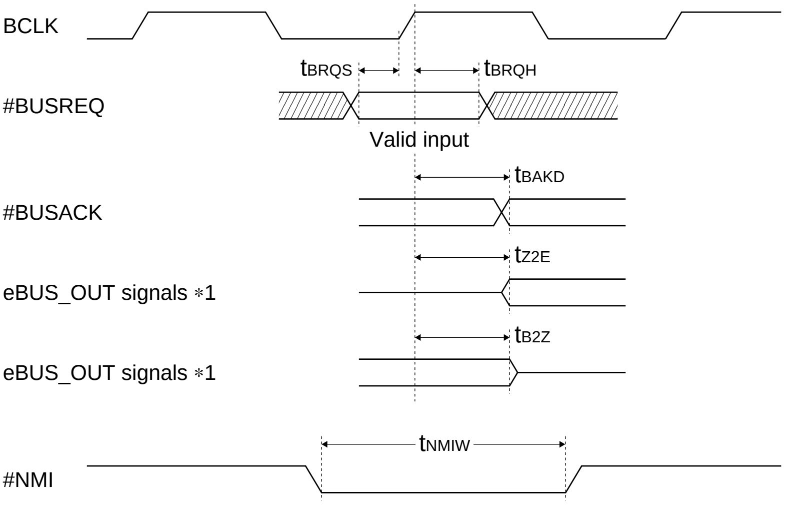


Burst ROM read cycle



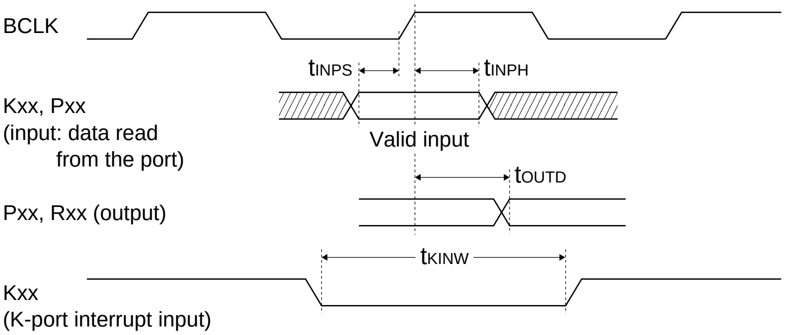
∗1 tRDH is measured with respect to the first signal change (negation) from among the #RD, #CEx and A[23:0] signals.

#BUSREQ, #BUSACK and #NMI timing



∗1 eBUS_OUT indicates the following pins:
A[23:0], #RD, #WRL, #WRH, #HCAS, #LCAS, #CE[17:4], D[15:0]

Input, output and I/O port timing



18.8 Oscillation Characteristics

Oscillation characteristics change depending on conditions (board pattern, components used, etc.). Use the following characteristics as reference values. In particular, when a ceramic or crystal oscillator is used, use the oscillator manufacturer recommended values for constants such as capacitance and resistance.

OSC1 crystal oscillation

(Unless otherwise specified: V_{DD}=5.0V, V_{SS}=0V, crystal=C-002RX[#1], C_{G1}=15pF[#2], Ta=25°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Oscillation start time	t_{STA1}				3	Sec	
External gate capacitance	C_{G1}	Including board capacitance	5		25	pF	
Frequency/IC deviation	$\Delta f/\Delta IC$		-10		10	ppm	
Frequency/power voltage deviation	$\Delta f/\Delta V$	C_G=15pF	-10		10	ppm/V	
Frequency adjustment range	$\Delta f/\Delta C_G$	C_G=5 to 25pF	50			ppm	

#1 C-002RX: Crystal resonator made by Seiko Epson
#2 C_{G1}=15pF: Board capacitance is included.

(Unless otherwise specified: V_{DD}=3.3V, V_{SS}=0V, crystal=C-002RX[#1], C_{G1}=15pF[#2], Ta=25°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Oscillation start time	t_{STA1}				3	Sec	
External gate capacitance	C_{G1}	Including board capacitance	5		25	pF	
Frequency/IC deviation	$\Delta f/\Delta IC$		-10		10	ppm	
Frequency/power voltage deviation	$\Delta f/\Delta V$	C_G=15pF	-10		10	ppm/V	
Frequency adjustment range	$\Delta f/\Delta C_G$	C_G=5 to 25pF	50			ppm	

#1 C-002RX: Crystal resonator made by Seiko Epson
#2 C_{G1}=15pF: Board capacitance is included.

OSC3 crystal oscillation

Note: A "crystal resonator that uses a fundamental" should be used for the OSC3 crystal oscillation circuit.

(Unless otherwise specified: V_{DD}=5.0V, V_{SS}=0V, crystal=20MHz: MA-306[#1]/33MHz: MA-306[#1], C_{G2}=C_{D2}=5pF, Rf=1MΩ, Ta=25°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Oscillation start time	t_{STA3}	20MHz crystal oscillator			5	mS	
		33MHz crystal oscillator			5	mS	

#1 MA-306: Crystal resonator made by Seiko Epson

(Unless otherwise specified: V_{DD}=3.3V, V_{SS}=0V, crystal=MA-306[#1], C_{G2}=C_{D2}=5pF, Rf=1MΩ, Ta=25°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Oscillation start time	t_{STA3}	20MHz crystal oscillator			10	mS	

#1 MA-306: Crystal resonator made by Seiko Epson

OSC3 ceramic oscillation

(Unless otherwise specified: V_{DD}=5.0V, V_{SS}=0V, ceramic=20MHz: CSA20.00MXZ040[#1]/33MHz: CSA33.00MXZ040[#2], C_{G2}=C_{D2}=20MHz: 5pF/33MHz: open, Rf=1MΩ, Ta=25°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Oscillation start time	t_{STA3}	20MHz ceramic oscillator			3	mS	
		33MHz ceramic oscillator			3	mS	

#1 CSA20.00MXZ040: Ceramic oscillator made by Murata Mfg. corporation
#2 CSA33.00MXZ040: Ceramic oscillator made by Murata Mfg. corporation

(Unless otherwise specified: V_{DD}=3.3V, V_{SS}=0V, ceramic=CSA20.00MXZ040[#1], C_{G2}=C_{D2}=5pF, Rf=1MΩ, Ta=25°C)

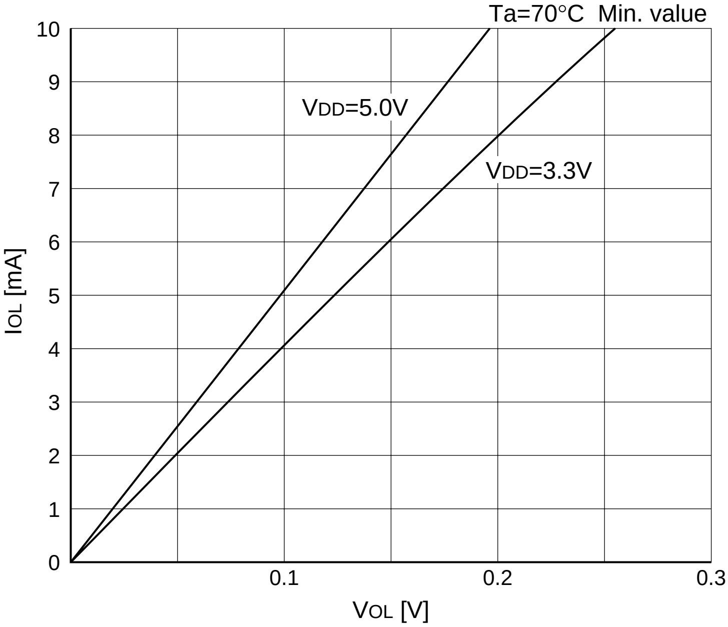
Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Oscillation start time	t_{STA3}	20MHz ceramic oscillator			5	mS	

#1 CSA20.00MXZ040: Ceramic oscillator made by Murata Mfg. corporation

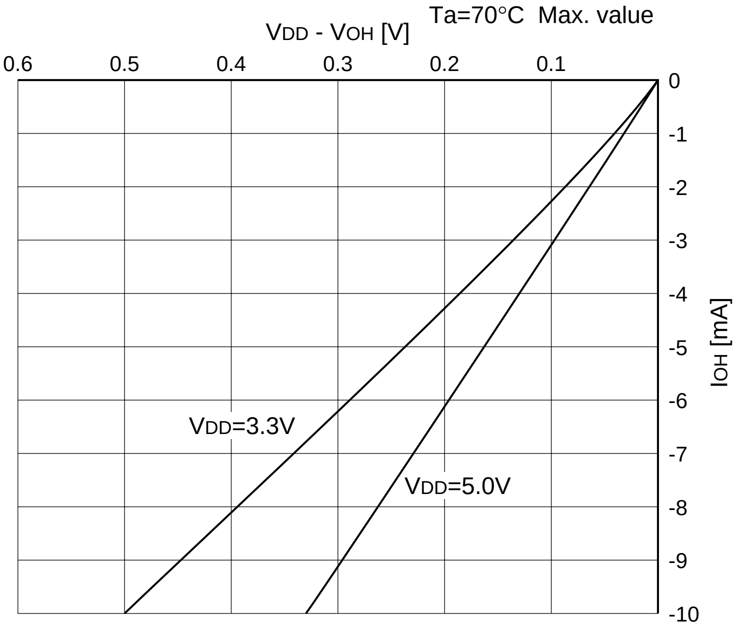
18.9 Characteristic Curves

Use the following characteristics as reference values.

Low-level output current



High-level output current

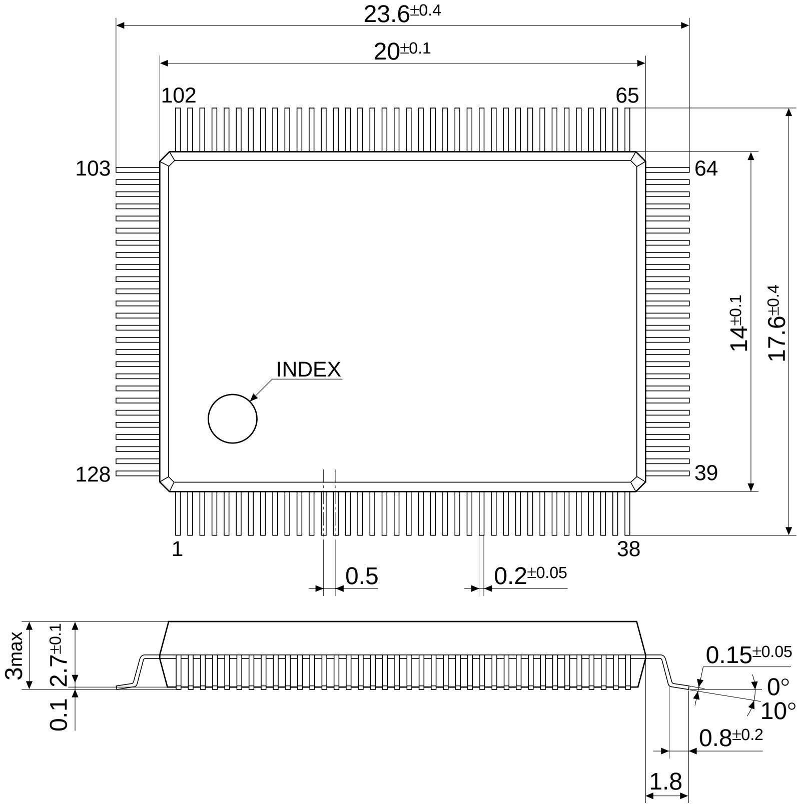


19 Package

19.1 Plastic Package

QFP5-128pin

(Unit: mm)



Limit of power consumption

The chip temperature of an LSI rises according to power consumption. The chip temperature can be calculated from environment temperature (Ta), thermal resistance (θ) and power consumption (PD)

Chip temperature (Tj) = Ta + (PD × θ) (°C)

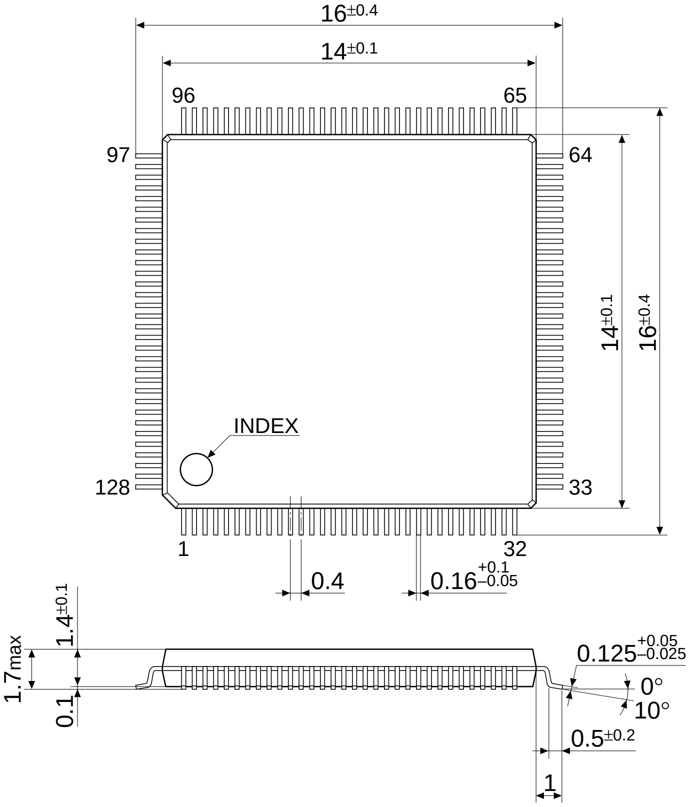
As a guide, normally keep the chip temperature (Tj) lower than 85°C.
The thermal resistance of the QFP5-128pin package is as follows:

Thermal resistance (°C/W) = 105 to 115°C (75 to 85°C for Cu lead frame)

This thermal resistance is a value under the condition that the measured device is hanging in the air and has no air-cooling. Thermal resistance greatly varies according to the mounting condition on the board and air-cooling condition.

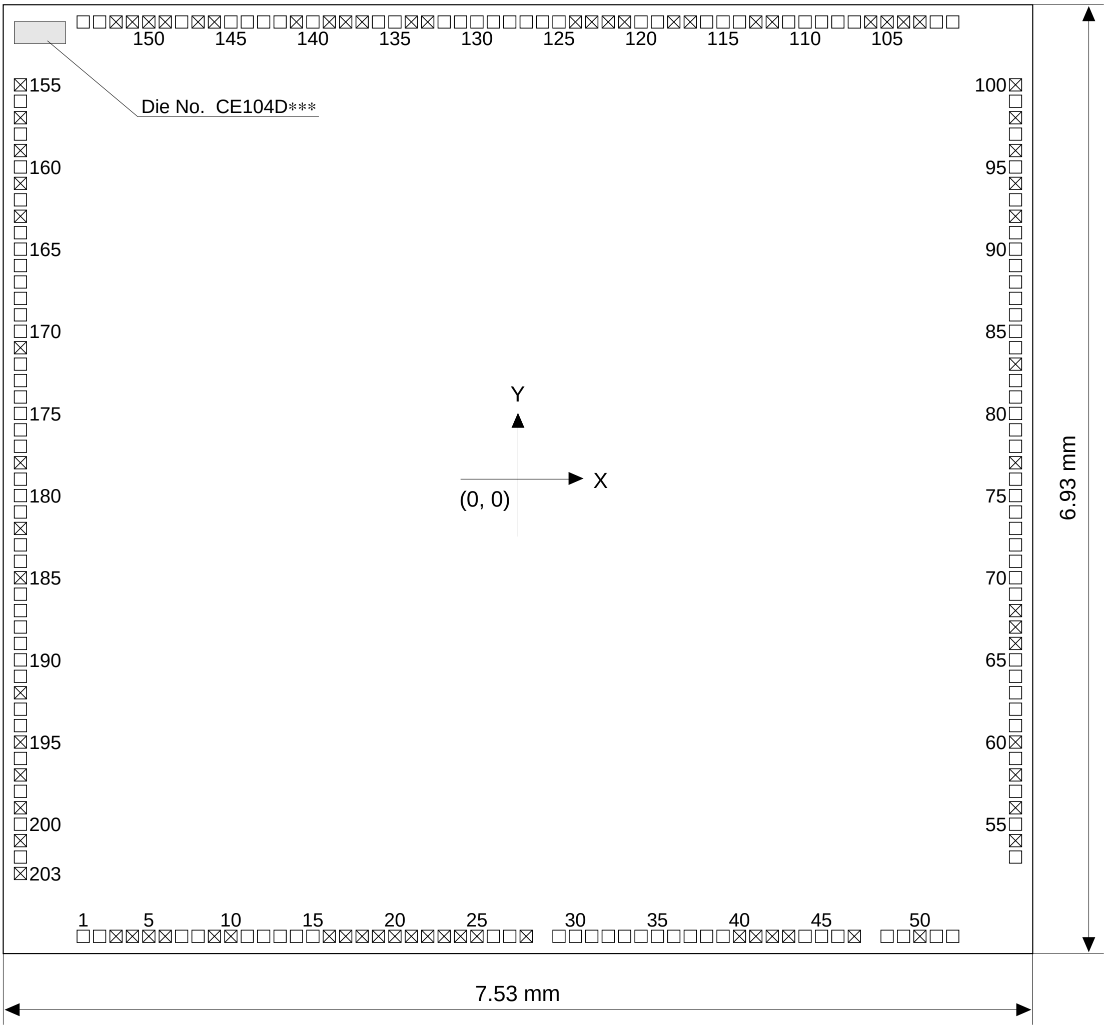
QFP15-128pin

(Unit: mm)



20 Pad Layout

20.1 Pad Layout Diagram



20.2 Pad Coordinate

(Unit: µm)

No.	Pad name	X	Y	No.	Pad name	X	Y
1	#CE9/#CE17	-3180.00	-3339.38	53	K52/#ADTRG	3639.38	-2760.00
2	#CE10	-3060.00	-3339.38	54	N.C.	3639.38	-2640.00
3	N.C.	-2940.00	-3339.38	55	R70/TM00	3639.38	-2520.00
4	N.C.	-2820.00	-3339.38	56	N.C.	3639.38	-2400.00
5	N.C.	-2700.00	-3339.38	57	R71/TM01	3639.38	-2280.00
6	N.C.	-2580.00	-3339.38	58	N.C.	3639.38	-2160.00
7	VS5V	-2460.00	-3339.38	59	R72/TM10	3639.38	-2040.00
8	#NMI	-2340.00	-3339.38	60	N.C.	3639.38	-1920.00
9	N.C.	-2220.00	-3339.38	61	R73/TM11	3639.38	-1800.00
10	N.C.	-2100.00	-3339.38	62	R74/TM21	3639.38	-1680.00
11	V_{SS}	-1980.00	-3339.38	63	R75/TM31	3639.38	-1560.00
12	#TSTMD	-1860.00	-3339.38	64	R76/TM41	3639.38	-1440.00
13	#RESET	-1740.00	-3339.38	65	R77/TM51	3639.38	-1320.00
14	V_{DD}	-1620.00	-3339.38	66	N.C.	3639.38	-1200.00
15	V_{SS}	-1500.00	-3339.38	67	N.C.	3639.38	-1080.00
16	N.C.	-1380.00	-3339.38	68	N.C.	3639.38	-960.00
17	N.C.	-1260.00	-3339.38	69	V_{SS}	3639.38	-840.00
18	N.C.	-1140.00	-3339.38	70	V_{DD}	3639.38	-720.00
19	N.C.	-1020.00	-3339.38	71	P10/EXCL00/T8UF0/DST0	3639.38	-600.00
20	N.C.	-900.00	-3339.38	72	P11/EXCL01/T8UF1/DST1	3639.38	-480.00
21	N.C.	-780.00	-3339.38	73	P12/EXCL10/T8UF2/DST2	3639.38	-360.00
22	N.C.	-660.00	-3339.38	74	P13/EXCL20/T8UF3/DPCO	3639.38	-240.00
23	N.C.	-540.00	-3339.38	75	P14/EXCL30/#BUSGET/#DWE	3639.38	-120.00
24	N.C.	-420.00	-3339.38	76	P15/EXCL40/#DMAEND0	3639.38	0.00
25	N.C.	-300.00	-3339.38	77	N.C.	3639.38	120.00
26	OSC4	-180.00	-3339.38	78	P16/EXCL50/#DMAEND1	3639.38	240.00
27	OSC3	-60.00	-3339.38	79	BTA3	3639.38	360.00
28	N.C.	60.00	-3339.38	80	K51/#DMAREQ1	3639.38	480.00
29	AV_{RH}	300.00	-3339.38	81	R83/#DMAACK1	3639.38	600.00
30	AV_{RL}	420.00	-3339.38	82	#TSTMD1	3639.38	720.00
31	AV_{SS}	540.00	-3339.38	83	N.C.	3639.38	840.00
32	K67/AD7	660.00	-3339.38	84	R81/FOSC1/#DRD	3639.38	960.00
33	K66/AD6	780.00	-3339.38	85	K50/#DMAREQ0	3639.38	1080.00
34	K65/AD5	900.00	-3339.38	86	R82/#DMAACK0	3639.38	1200.00
35	K64/AD4	1020.00	-3339.38	87	P00/SIN0	3639.38	1320.00
36	K63/AD3	1140.00	-3339.38	88	V_{DD}	3639.38	1440.00
37	K62/AD2	1260.00	-3339.38	89	V_{SS}	3639.38	1560.00
38	K61/AD1	1380.00	-3339.38	90	P01/SOUT0	3639.38	1680.00
39	K60/AD0	1500.00	-3339.38	91	P02/#SCLK0	3639.38	1800.00
40	N.C.	1620.00	-3339.38	92	N.C.	3639.38	1920.00
41	N.C.	1740.00	-3339.38	93	P03/#SRDY0	3639.38	2040.00
42	N.C.	1860.00	-3339.38	94	N.C.	3639.38	2160.00
43	N.C.	1980.00	-3339.38	95	P04/SIN1	3639.38	2280.00
44	K53/DA0	2100.00	-3339.38	96	N.C.	3639.38	2400.00
45	K54/DA1	2220.00	-3339.38	97	P05/SOUT1	3639.38	2520.00
46	AV_{DD}	2340.00	-3339.38	98	N.C.	3639.38	2640.00
47	N.C.	2460.00	-3339.38	99	P06/#SCLK1	3639.38	2760.00
48	V_{SS}	2700.00	-3339.38	100	N.C.	3639.38	2880.00
49	OSC2	2820.00	-3339.38	–	–	–	–
50	N.C.	2940.00	-3339.38	–	–	–	–
51	OSC1	3060.00	-3339.38	–	–	–	–
52	V_{DD1}	3180.00	-3339.38	–	–	–	–

No.	Pad name	X	Y	No.	Pad name	X	Y
101	P07/#SRDY1	3180.00	3339.38	155	N.C.	-3639.38	2880.00
102	A7	3060.00	3339.38	156	#WAIT	-3639.38	2760.00
103	N.C.	2940.00	3339.38	157	N.C.	-3639.38	2640.00
104	N.C.	2820.00	3339.38	158	ARAMMD1	-3639.38	2520.00
105	N.C.	2700.00	3339.38	159	N.C.	-3639.38	2400.00
106	N.C.	2580.00	3339.38	160	ARAMMD0	-3639.38	2280.00
107	A6	2460.00	3339.38	161	N.C.	-3639.38	2160.00
108	A5	2340.00	3339.38	162	#BUSACK	-3639.38	2040.00
109	A4	2220.00	3339.38	163	N.C.	-3639.38	1920.00
110	A3	2100.00	3339.38	164	#BUSREQ	-3639.38	1800.00
111	A2	1980.00	3339.38	165	BCLK	-3639.38	1680.00
112	N.C.	1860.00	3339.38	166	#HCAS	-3639.38	1560.00
113	N.C.	1740.00	3339.38	167	#LCAS	-3639.38	1440.00
114	A1	1620.00	3339.38	168	#WRH/#BSH	-3639.38	1320.00
115	A0/#BSL	1500.00	3339.38	169	#WRL/#WR/#WE	-3639.38	1200.00
116	D15	1380.00	3339.38	170	#RD	-3639.38	1080.00
117	N.C.	1260.00	3339.38	171	N.C.	-3639.38	960.00
118	N.C.	1140.00	3339.38	172	VDD	-3639.38	840.00
119	VSS	1020.00	3339.38	173	A8	-3639.38	720.00
120	D14	900.00	3339.38	174	A9	-3639.38	600.00
121	N.C.	780.00	3339.38	175	A10	-3639.38	480.00
122	N.C.	660.00	3339.38	176	A11	-3639.38	360.00
123	N.C.	540.00	3339.38	177	A12	-3639.38	240.00
124	N.C.	420.00	3339.38	178	N.C.	-3639.38	120.00
125	D13	300.00	3339.38	179	A13	-3639.38	0.00
126	D12	180.00	3339.38	180	A14	-3639.38	-120.00
127	D11	60.00	3339.38	181	A15	-3639.38	-240.00
128	D10	-60.00	3339.38	182	N.C.	-3639.38	-360.00
129	D9	-180.00	3339.38	183	A16	-3639.38	-480.00
130	D8	-300.00	3339.38	184	A17	-3639.38	-600.00
131	D7	-420.00	3339.38	185	N.C.	-3639.38	-720.00
132	VDD	-540.00	3339.38	186	A18	-3639.38	-840.00
133	N.C.	-660.00	3339.38	187	A19	-3639.38	-960.00
134	N.C.	-780.00	3339.38	188	A20	-3639.38	-1080.00
135	VSS	-900.00	3339.38	189	A21	-3639.38	-1200.00
136	D6	-1020.00	3339.38	190	A22	-3639.38	-1320.00
137	N.C.	-1140.00	3339.38	191	A23	-3639.38	-1440.00
138	N.C.	-1260.00	3339.38	192	N.C.	-3639.38	-1560.00
139	N.C.	-1380.00	3339.38	193	VSS	-3639.38	-1680.00
140	VDD	-1500.00	3339.38	194	#CE4/#CE11	-3639.38	-1800.00
141	N.C.	-1620.00	3339.38	195	N.C.	-3639.38	-1920.00
142	VSS	-1740.00	3339.38	196	#CE5/#CE15	-3639.38	-2040.00
143	D5	-1860.00	3339.38	197	N.C.	-3639.38	-2160.00
144	D4	-1980.00	3339.38	198	#CE6	-3639.38	-2280.00
145	D3	-2100.00	3339.38	199	N.C.	-3639.38	-2400.00
146	N.C.	-2220.00	3339.38	200	#CE7/#CE13/#RAS0/#RAS2	-3639.38	-2520.00
147	N.C.	-2340.00	3339.38	201	N.C.	-3639.38	-2640.00
148	D2	-2460.00	3339.38	202	#CE8/#CE14/#RAS1/#RAS3	-3639.38	-2760.00
149	N.C.	-2580.00	3339.38	203	N.C.	-3639.38	-2880.00
150	N.C.	-2700.00	3339.38	–	–	–	–
151	N.C.	-2820.00	3339.38	–	–	–	–
152	N.C.	-2940.00	3339.38	–	–	–	–
153	D1	-3060.00	3339.38	–	–	–	–
154	D0	-3180.00	3339.38	–	–	–	–

Appendix A <Reference> External Device Interface Timings

This section shows setup examples for setting timing conditions of the external system interface as a reference material used when configuring a system with external devices.
Pay attention to the following precautions when using this material.

- The described AC characteristic values of external devices are standard values. They may differ from those of the devices actually used, so the actual setup values (number of cycles) should be determined by referring the manual or specification of the device to be used.
- It is necessary to set the timing values allowing ample margin according to the load capacitance of the bus and signal lines, number of devices to be connected, operating temperature range, I/O levels and other conditions. The number of cycles described in this section is an example and the conditions are not considered.
- The values described in "Time" column of the tables are simply calculated by multiplying the number of cycles by the cycle time. Conditions such as the output delay time of the device, delay due to wiring and load capacitance, and input setup time are not considered.
- The described contents are reference data and cannot be guaranteed to work.

A.1 Interface Timing Examples for 5V Operation

A.11 DRAM (70nS)

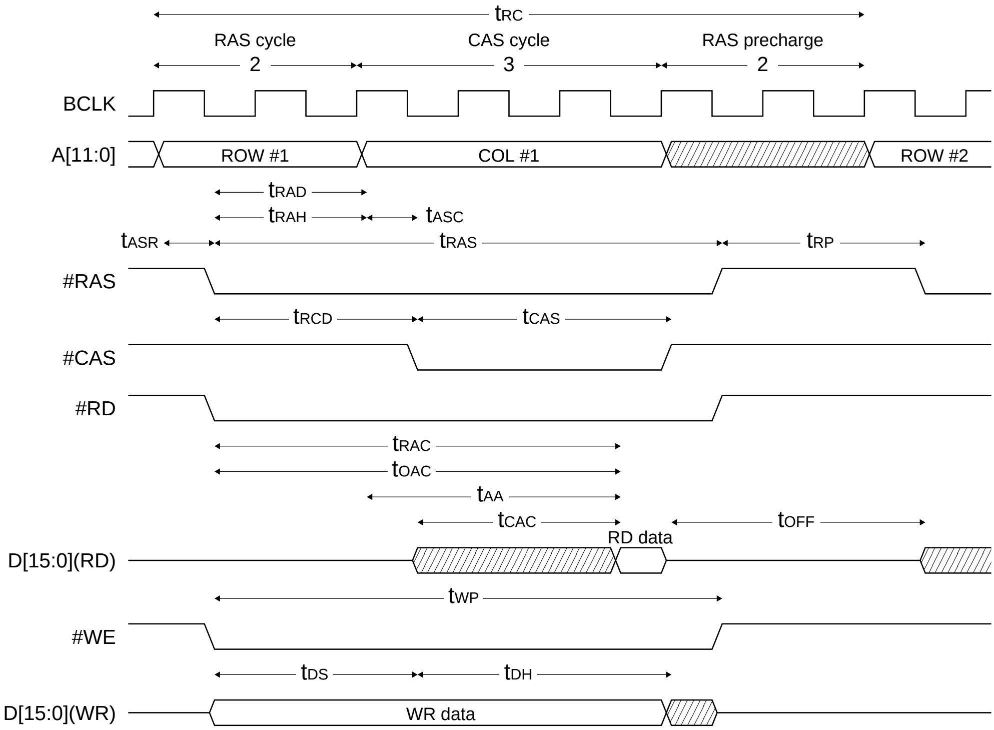
DRAM interface setup examples – 70nS

Operating frequency	RAS precharge cycle	RAS cycle	CAS cycle	Refresh RAS pulse width	Refresh RPC delay
20MHz	2	1	2	2	1
25MHz	2	1	2	2	1
33MHz	2	2	3	3	1

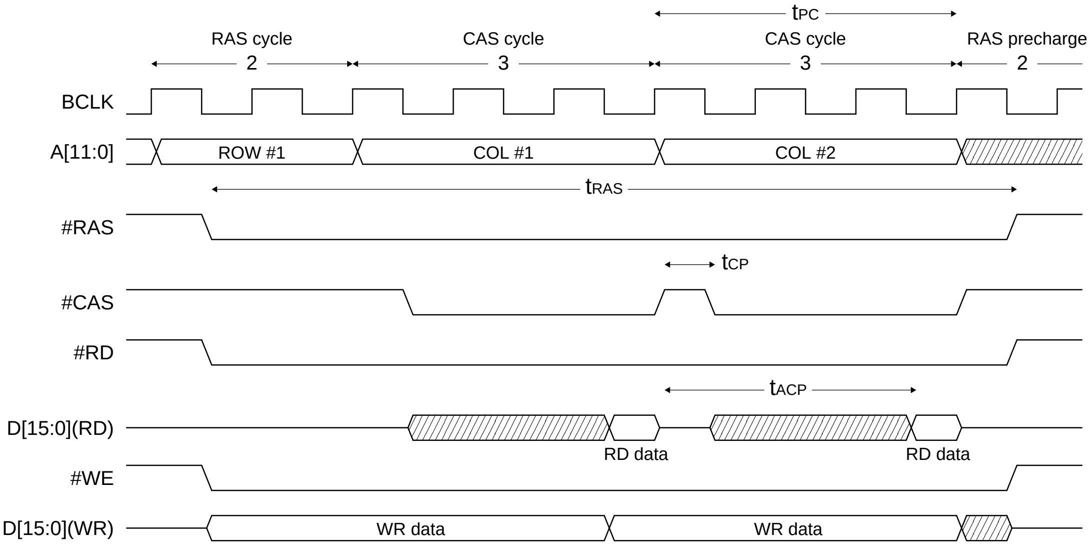
DRAM interface timing – 70nS

DRAM interface		Unit: nS		33MHz		25MHz		20MHz	
Parameter	Symbol	Min.	Max.	Cycle	Time	Cycle	Time	Cycle	Time
<Common parameters>									
Random read/random write cycle time	t_{RC}	130	–	7	210	5	200	5	250
#RAS precharge time	t_{RP}	50	–	2	60	2	80	2	100
#RAS pulse width	t_{RAS}	70	10000	5	150	3	120	3	150
#CAS pulse width	t_{CAS}	20	10000	2.5	75	1.5	60	1.5	75
Row address setup time	t_{ASR}	0	–	0.5	15	0.5	20	0.5	25
Row address hold time	t_{RAH}	10	–	1.5	45	0.5	20	0.5	25
Column address setup time	t_{ASC}	0	–	0.5	15	0.5	20	0.5	25
#RAS→#CAS delay time	t_{RCD}	20	–	2.0	60	1.0	40	1.0	50
#RAS→column address delay time	t_{RAD}	15	–	1.5	45	0.5	20	0.5	25
<Read-cycle parameters>									
#RAS access time	t_{RAC}	–	70	4.5	135	2.5	100	2.5	125
#CAS access time	t_{CAC}	–	20	2.5	75	1.5	60	1.5	75
Address access time	t_{AA}	–	35	3.0	90	2.0	80	2.0	100
#OE access time	t_{OAC}	–	20	4.5	135	2.5	100	2.5	125
Output buffer turn-off time	t_{OFF}	0	20	2	60	2	80	2	100
<Write-cycle parameters>									
Data input hold time	t_{DH}	15	–	2.5	75	1.5	60	1.5	75
<Fast-page mode>									
Fast-page mode cycle time	t_{PC}	45	–	3.0	90	2.0	80	2.0	100
Fast-page mode #CAS precharge time	t_{CP}	10	–	0.5	15	0.5	20	0.5	25
Access time after #CAS precharge	t_{ACP}	–	40	3.0	90	2.0	80	2.0	100
<Refresh cycle>									
#CAS setup time	t_{CSR}	10	–	1.0	30	1.0	40	1.0	50
#CAS hold time	t_{CHR}	10	–	2.5	75	1.5	60	1.5	75
#RAS precharge→#CAS hold time	t_{PPC}	10	–	1.0	30	1.0	40	1.0	50
#RAS pulse width (only in refresh cycle)	t_{RAS}	70	10000	3.0	90	2.0	80	2.0	100

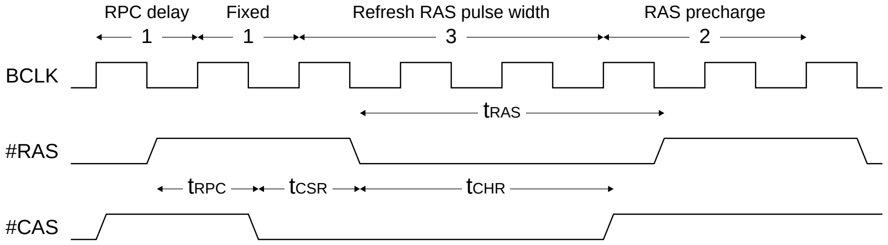
DRAM: 70nS, CPU: 33MHz, random read/write cycle



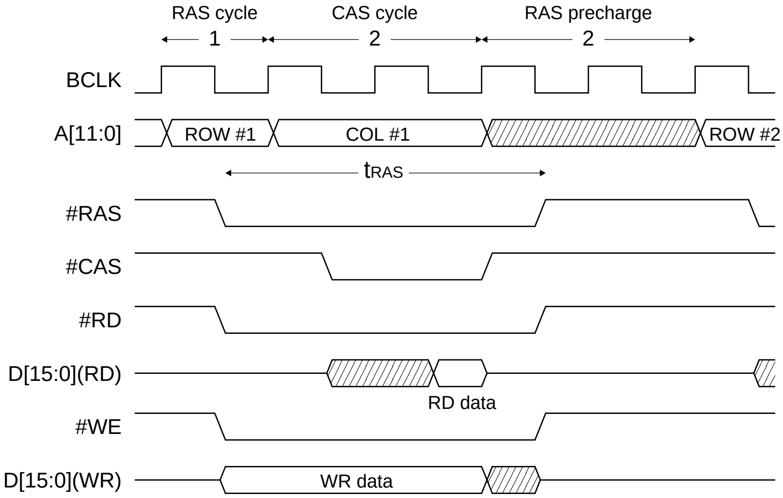
DRAM: 70nS, CPU: 33MHz, page-mode read/write cycle



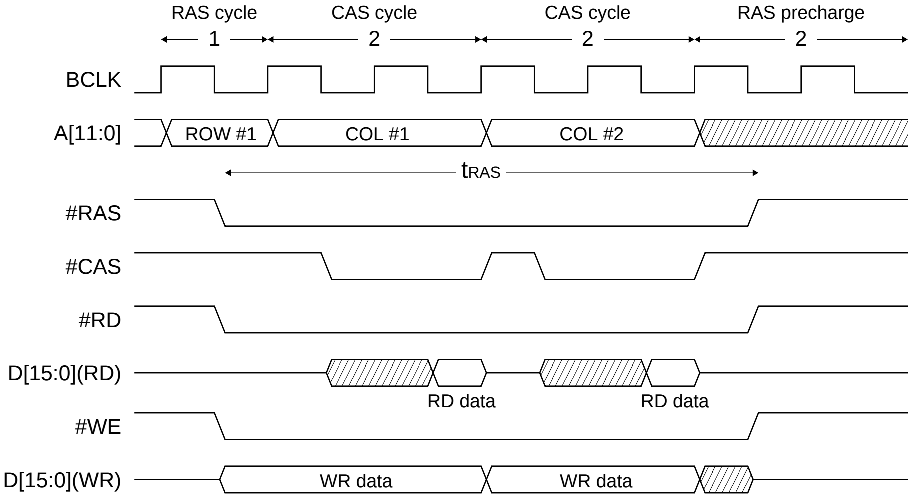
DRAM: 70nS, CPU: 33MHz, CAS-before-RAS refresh cycle



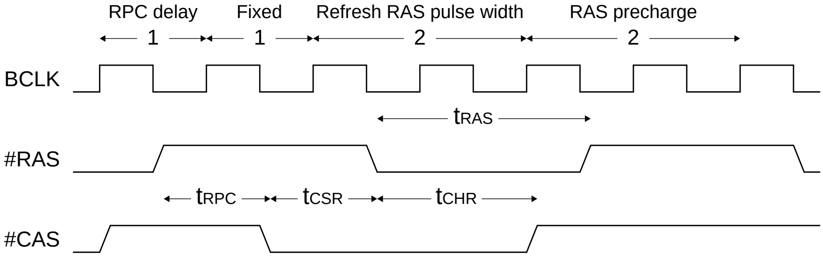
DRAM: 70nS, CPU: 25/20MHz, random read/write cycle



DRAM: 70nS, CPU: 25/20MHz, page-mode read/write cycle



DRAM: 70nS, CPU: 25/20MHz, CAS-before-RAS refresh cycle



A.12 DRAM (60nS)

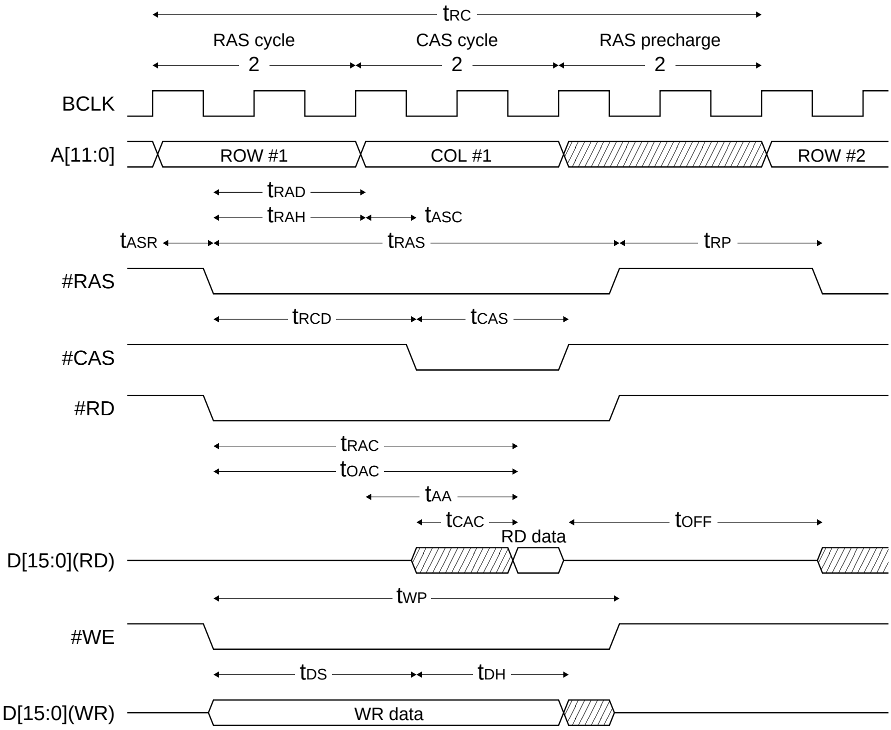
DRAM interface setup examples – 60nS

Operating frequency	RAS precharge cycle	RAS cycle	CAS cycle	Refresh RAS pulse width	Refresh RPC delay
20MHz	1	1	2	2	1
25MHz	2	1	2	2	1
33MHz	2	2	2	3	1

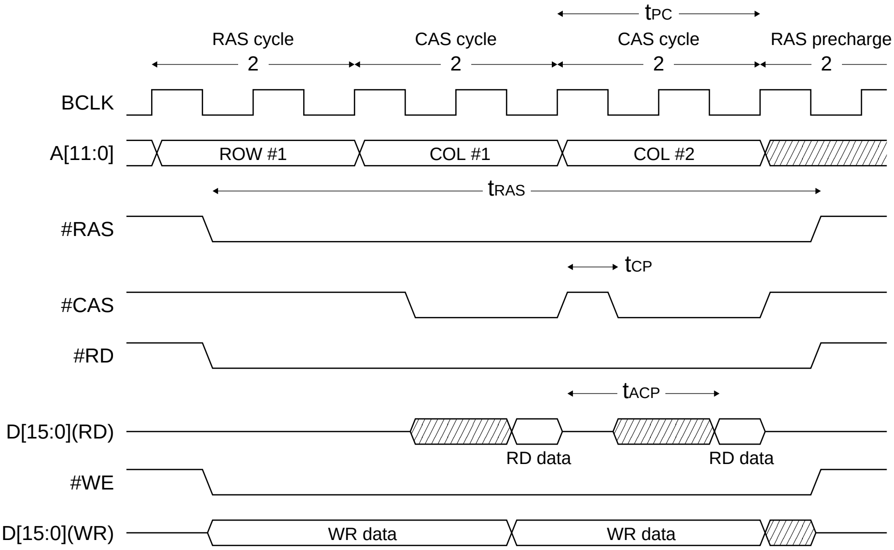
DRAM interface timing – 60nS

DRAM interface		Unit: nS		33MHz		25MHz		20MHz	
Parameter	**Symbol**	**Min.**	**Max.**	**Cycle**	**Time**	**Cycle**	**Time**	**Cycle**	**Time**
<Common parameters>									
Random read/random write cycle time	t_{RC}	110	–	6	180	5	200	4	200
#RAS precharge time	t_{RP}	40	–	2	60	2	80	1	50
#RAS pulse width	t_{RAS}	60	10000	4	120	3	120	3	150
#CAS pulse width	t_{CAS}	15	10000	1.5	45	1.5	60	1.5	75
Row address setup time	t_{ASR}	0	–	0.5	15	0.5	20	0.5	25
Row address hold time	t_{RAH}	10	–	1.5	45	0.5	20	0.5	25
Column address setup time	t_{ASC}	0	–	0.5	15	0.5	20	0.5	25
#RAS→#CAS delay time	t_{RCD}	20	–	2.0	60	1.0	40	1.0	50
#RAS→column address delay time	t_{RAD}	15	–	1.5	45	0.5	20	0.5	25
<Read-cycle parameters>									
#RAS access time	t_{RAC}	–	60	3.5	105	2.5	100	2.5	125
#CAS access time	t_{CAC}	–	15	1.5	45	1.5	60	1.5	75
Address access time	t_{AA}	–	30	2.0	60	2.0	80	2.0	100
#OE access time	t_{OAC}	–	15	3.5	105	2.5	100	2.5	125
Output buffer turn-off time	t_{OFF}	0	15	2	60	2	80	1	50
<Write-cycle parameters>									
Data input hold time	t_{DH}	10	–	1.5	45	1.5	60	1.5	75
<Fast-page mode>									
Fast-page mode cycle time	t_{PC}	40	–	2.0	60	2.0	80	2.0	100
Fast-page mode #CAS precharge time	t_{CP}	10	–	0.5	15	0.5	20	0.5	25
Access time after #CAS precharge	t_{ACP}	–	35	2.0	60	2.0	80	2.0	100
<Refresh cycle>									
#CAS setup time	t_{CSR}	10	–	1.0	30	1.0	40	1.0	50
#CAS hold time	t_{CHR}	10	–	2.5	75	1.5	60	1.5	75
#RAS precharge→#CAS hold time	t_{PPC}	10	–	1.0	30	1.0	40	1.0	50
#RAS pulse width (only in refresh cycle)	t_{RAS}	60	10000	3.0	90	2.0	80	2.0	100

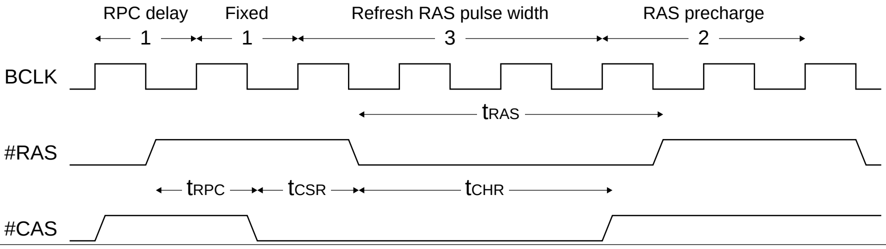
DRAM: 60nS, CPU: 33MHz, random read/write cycle



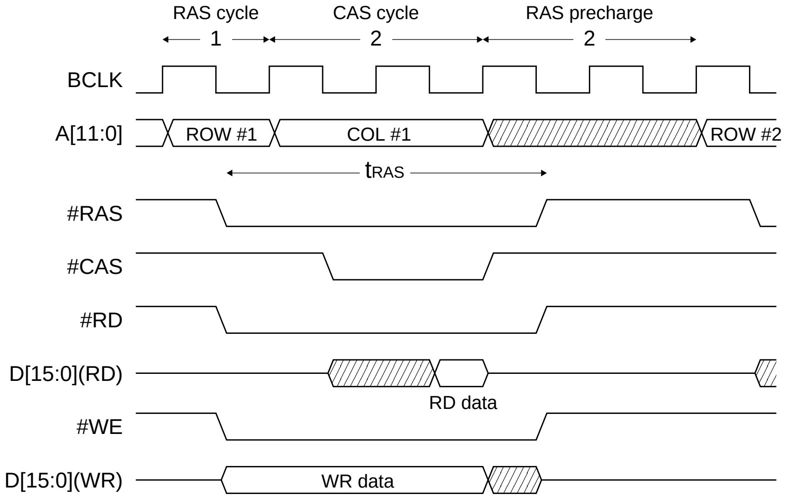
DRAM: 60nS, CPU: 33MHz, page-mode read/write cycle



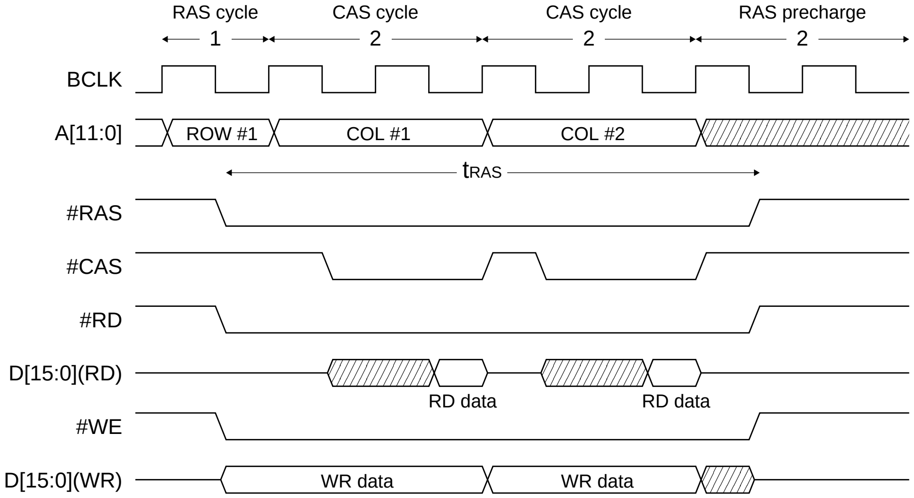
DRAM: 60nS, CPU: 33MHz, CAS-before-RAS refresh cycle



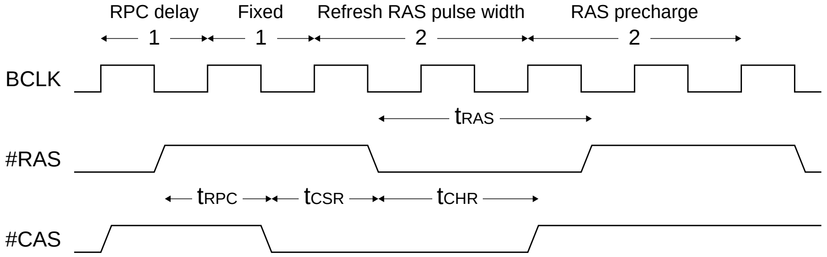
DRAM: 60nS, CPU: 25MHz, random read/write cycle



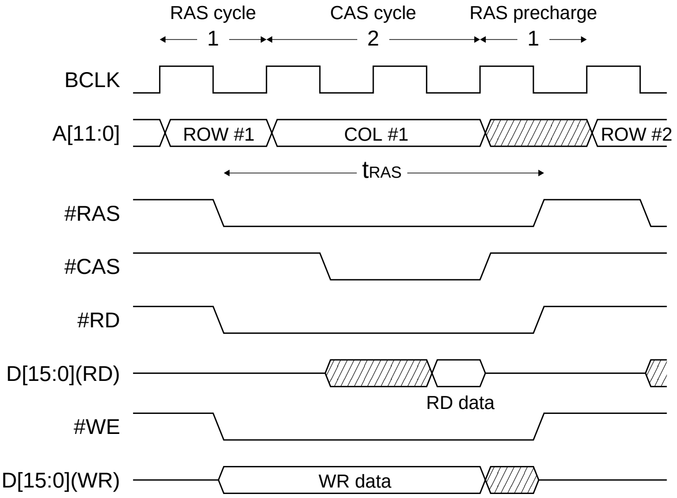
DRAM: 60nS, CPU: 25MHz, page-mode read/write cycle



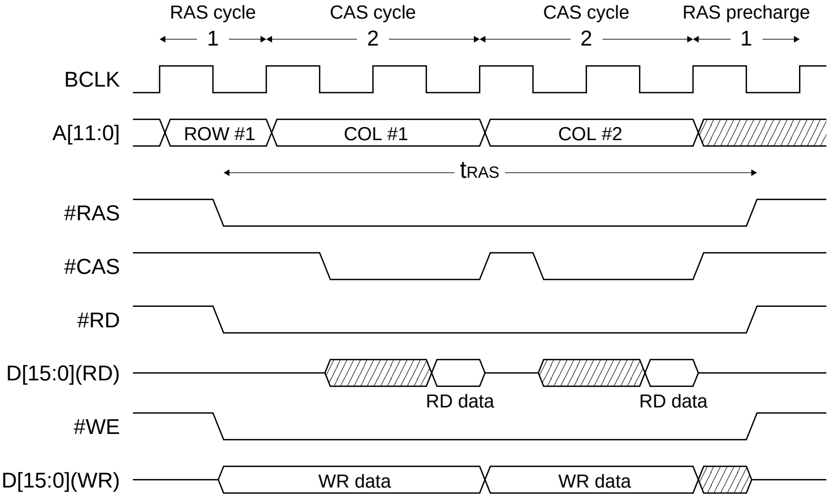
DRAM: 60nS, CPU: 25MHz, CAS-before-RAS refresh cycle



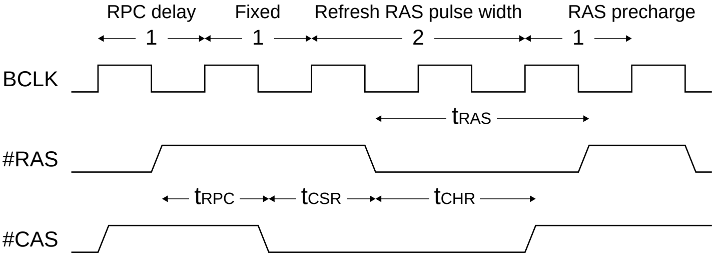
DRAM: 60nS, CPU: 20MHz, random read/write cycle



DRAM: 60nS, CPU: 20MHz, page-mode read/write cycle



DRAM: 60nS, CPU: 20MHz, CAS-before-RAS refresh cycle



A.13 ROM and Burst ROM

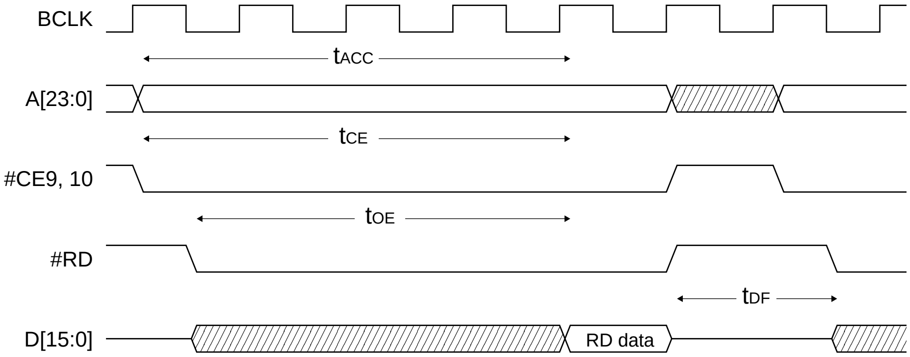
Burst ROM and mask ROM interface setup examples

Operating frequency	Normal read cycle		Burst read cycle		Output disable delay cycle
	Wait cycle	Read cycle	Wait cycle	Read cycle	
20MHz	2	3	1	2	1.5
25MHz	3	4	1	2	1.5
33MHz	4	5	2	3	1.5

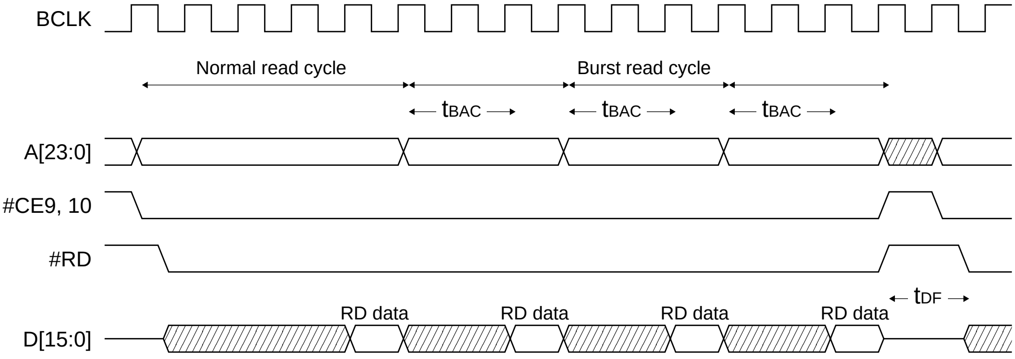
Burst ROM and mask ROM interface timing

Burst ROM and mask ROM interface				33MHz		25MHz		20MHz	
Parameter	Symbol	Min.	Max.	Cycle	Time	Cycle	Time	Cycle	Time
Access time	t_{ACC}	–	100	5	150	4	160	3	150
#CE output delay time	t_{CE}	–	100	5	150	4	160	3	150
#OE output delay time	t_{OE}	–	50	4.5	135	3.5	140	2.5	125
Burst access time	t_{BAC}	–	50	3	90	2	80	2	100
Output disable delay time	t_{DF}	0	40	1.5	45	1.5	60	1.5	75

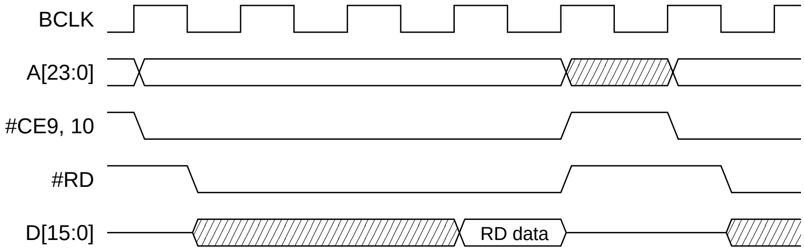
ROM: 100nS, CPU: 33MHz, normal read



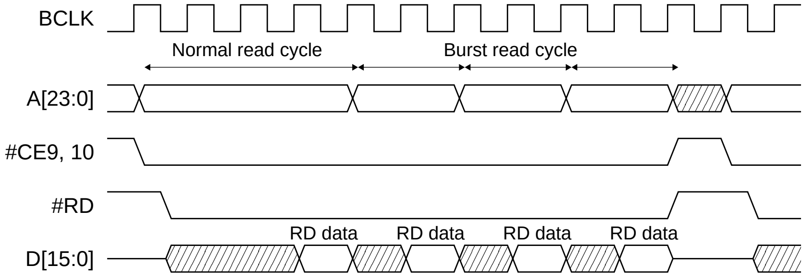
ROM: 100nS, CPU: 33MHz, burst read



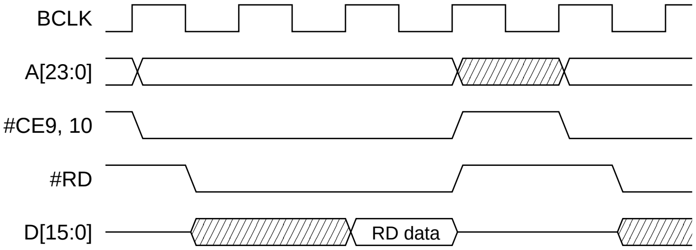
ROM: 100nS, CPU: 25MHz, normal read



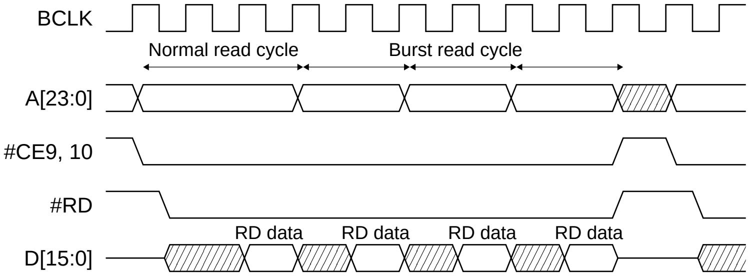
ROM: 100nS, CPU: 25MHz, burst read



ROM: 100nS, CPU: 20MHz, normal read



ROM: 100nS, CPU: 20MHz, burst read



A.14 SRAM (55nS)

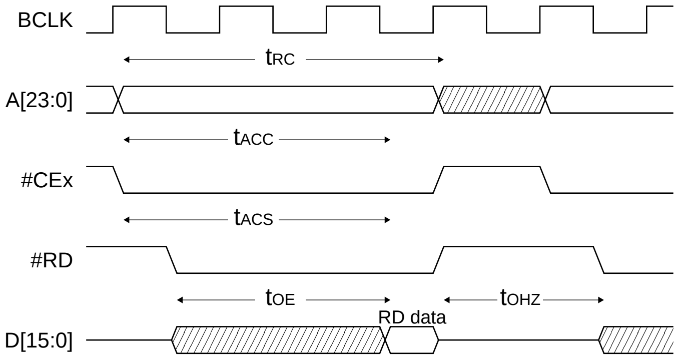
SRAM interface setup examples – 55nS

Operating frequency	Read cycle		Write cycle	Output disable delay time
	Wait cycle	Read cycle		
20MHz	1	2	2	1.5
25MHz	2	3	3	1.5
33MHz	2	3	3	1.5

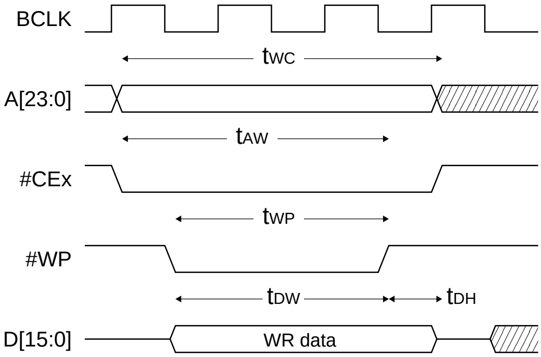
SRAM interface timing – 55nS

SRAM interface		5V		33MHz		25MHz		20MHz	
Parameter	Symbol	Min.	Max.	Cycle	Time	Cycle	Time	Cycle	Time
<Read cycle>									
Read cycle time	t_{RC}	55	–	3	90	3	120	2	100
Address access time	t_{ACC}	–	55	3	90	3	120	2	100
#CE access time	t_{ACS}	–	55	3	90	3	120	2	100
#OE access time	t_{OE}	–	30	2.5	75	2.5	100	1.5	75
Output disable delay time	t_{OHZ}	0	30	1.5	45	1.5	60	1.5	75
<Write cycle>									
Write cycle time	t_{WC}	55	–	3	90	3	120	2	100
Address enable time	t_{AW}	50	–	2.5	75	2.5	100	1.5	75
Write pulse width	t_{WP}	45	–	2	60	2	80	1	50
Input data setup time	t_{DW}	30	–	2	60	2	80	1	50
Input data hold time	t_{DH}	0	–	0.5	15	0.5	20	0.5	25

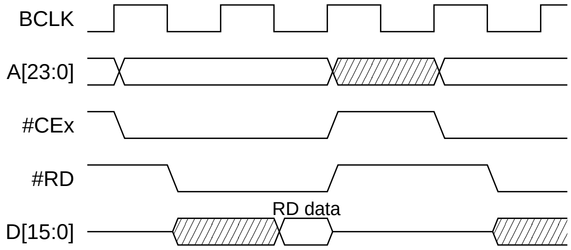
SRAM: 55nS, CPU: 33/25MHz, read cycle



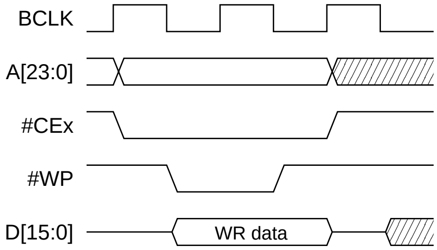
SRAM: 55nS, CPU: 33/25MHz, write cycle



SRAM: 55nS, CPU: 20MHz, read cycle



SRAM: 55nS, CPU: 20MHz, write cycle



A.15 SRAM (70nS)

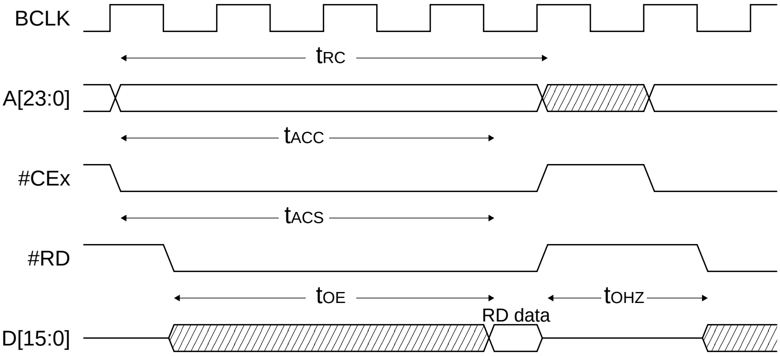
SRAM interface setup examples – 70nS

Operating frequency	Read cycle		Write cycle	Output disable delay time
	Wait cycle	Read cycle		
20MHz	2	3	3	1.5
25MHz	2	3	3	1.5
33MHz	3	4	4	1.5

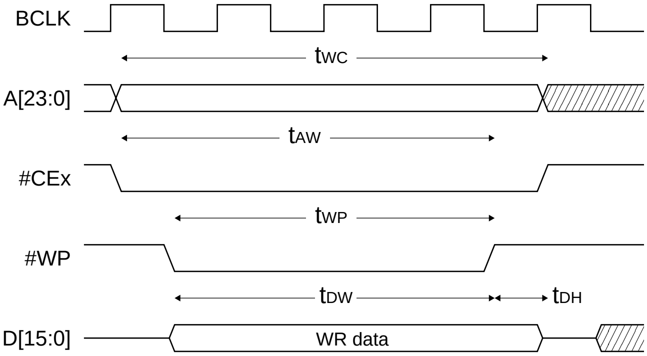
SRAM interface timing – 70nS

SRAM interface		5V		33MHz		25MHz		20MHz	
Parameter	Symbol	Min.	Max.	Cycle	Time	Cycle	Time	Cycle	Time
<Read cycle>									
Read cycle time	t_{RC}	70	–	4	120	3	120	3	150
Address access time	t_{ACC}	–	70	4	120	3	120	3	150
#CE access time	t_{ACS}	–	70	4	120	3	120	3	150
#OE access time	t_{OE}	–	40	3.5	105	2.5	100	2.5	125
Output disable delay time	t_{OHZ}	0	30	1.5	45	1.5	60	1.5	75
<Write cycle>									
Write cycle time	t_{WC}	70	–	4	120	3	120	3	150
Address enable time	t_{AW}	60	–	3.5	105	2.5	100	2.5	125
Write pulse width	t_{WP}	55	–	3	90	2	80	2	100
Input data setup time	t_{DW}	30	–	3	90	2	80	2	100
Input data hold time	t_{DH}	0	–	0.5	15	0.5	20	0.5	25

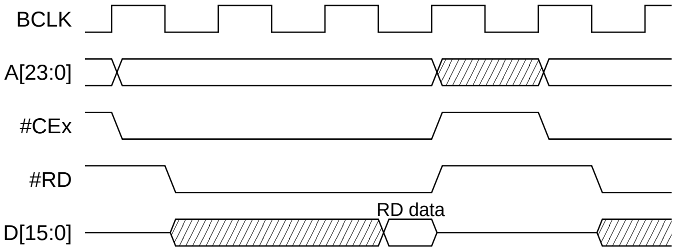
SRAM: 70nS, CPU: 33MHz, read cycle



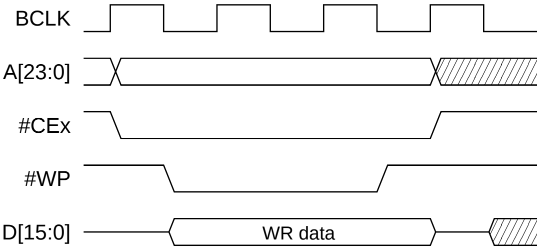
SRAM: 70nS, CPU: 33MHz, write cycle



SRAM: 70nS, CPU: 25/20MHz, read cycle



SRAM: 70nS, CPU: 25/20MHz, write cycle



A.16 8255A

8255A interface setup examples

Operating frequency	Read cycle		Write cycle	Output disable delay time
	Wait cycle	Read cycle		
20MHz	9 ∗1	10	10	3.5
25MHz	11	12	12	3.5
33MHz	14	15	15	3.5 ∗2

8255A interface timing

SRAM interface		5V		33MHz		25MHz		20MHz	
Parameter	Symbol	Min.	Max.	Cycle	Time	Cycle	Time	Cycle	Time
<Read cycle>									
Read cycle time	t_{RC}	300	–	15	450	12	480	10	500
Address access time	t_{ACC}	–	250	15	450	12	480	10	500
#CE access time	t_{ACS}	–	250	15	450	12	480	10	500
#OE access time	t_{OE}	–	250	14.5	435	11.5	460	9.5	475
Output disable delay time	t_{OHZ}	10	150	3.5	105	3.5	140	3.5	175
<Write cycle>									
Write cycle time	t_{WC}	430	–	15	450	12	480	10	500
Address enable time	t_{AW}	400	–	14.5	435	11.5	460	9.5	475
Write pulse width	t_{WP}	400	–	14	420	11	440	9	450
Input data setup time	t_{DW}	100	–	14	420	11	440	9	450
Input data hold time ∗3	t_{DH}	30	–	0.5	15	0.5	20	0.5	25

∗1 The E0C33A104 enables up to 7 cycles of wait-cycle insertion. If a number of wait cycles more than 7 cycles needs to be inserted, input the #WAIT signal from external hardware. Note that the interface must be set for SRAM type devices to insert wait cycles using the #WAIT pin. (Refer to Chapter 7, "External System Interface", for more information.)

∗2 This setting cannot satisfy the 150 nS of output-disable delay time specification required for the 8255A. When implementing such a low-speed device in the system, the external bus must be separated by inserting a 3-state bus buffer at the output side (when viewed from the CPU) of the external system bus.

∗3 If the data hold time that can be set is sufficient for the device, secure it by connecting a bus repeater to the external data bus D[15:0] or by inserting a latch at the output side of the external system interface.

A.2 Interface Timing Examples for 3.3V Operation

A.2.1 ROM and Burst ROM

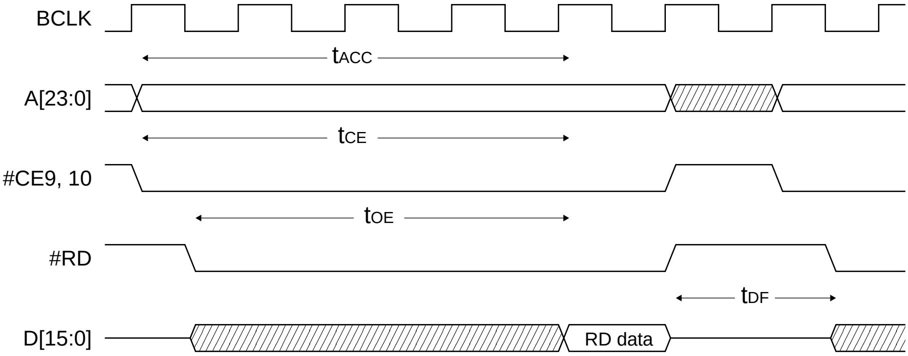
Burst ROM and mask ROM interface setup example

Operating frequency	Normal read cycle		Burst read cycle		Output disable delay cycle
	Wait cycle	Read cycle	Wait cycle	Read cycle	
20MHz	4	5	2	3	1.5

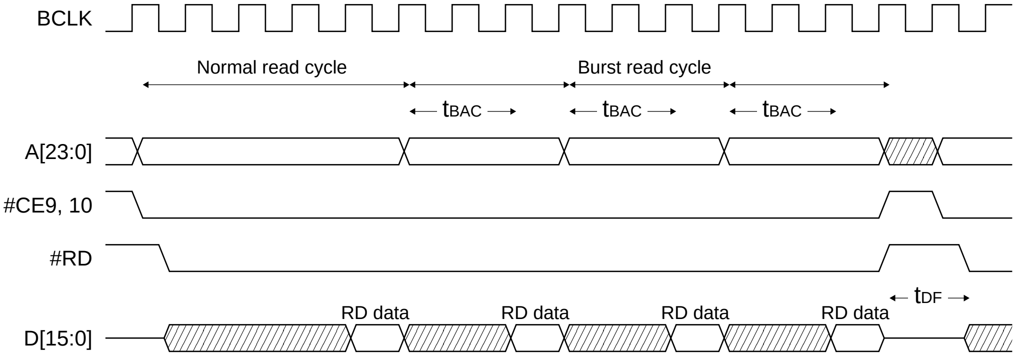
Burst ROM and mask ROM interface timing

Burst ROM and mask ROM interface 3.3V				20MHz	
Parameter	Symbol	Min.	Max.	Cycle	Time
Access time	t_{ACC}	–	200	5	250
#CE output delay time	t_{CE}	–	200	5	250
#OE output delay time	t_{OE}	–	100	4.5	225
Burst access time	t_{BAC}	–	100	3	150
Output disable delay time	t_{DF}	0	60	1.5	75

ROM: 200nS, CPU: 20MHz, normal read



ROM: 200nS, CPU: 20MHz, burst read



A.22 SRAM (150nS)

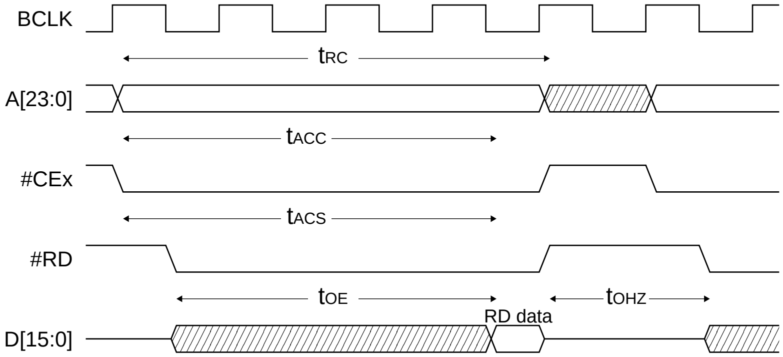
SRAM interface setup example – 150nS

Operating frequency	Read cycle		Write cycle	Output disable delay time
	Wait cycle	Read cycle		
20MHz	3	4	4	1.5

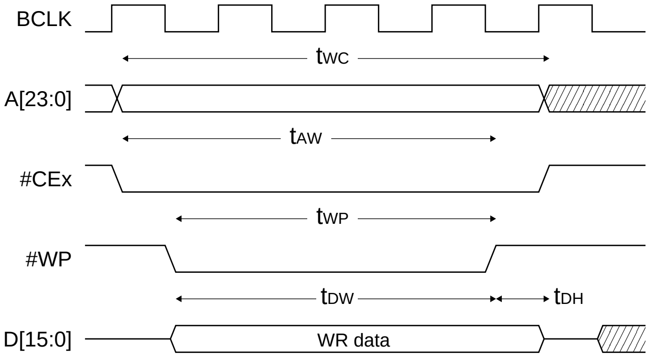
SRAM interface timing – 150nS

SRAM interface		2.7V		20MHz	
Parameter	Symbol	Min.	Max.	Cycle	Time
<Read cycle>					
Read cycle time	t_{RC}	150	–	4	200
Address access time	t_{ACC}	–	150	4	200
#CE access time	t_{ACS}	–	150	4	200
#OE access time	t_{OE}	–	75	3.5	175
Output disable delay time	t_{OHZ}	0	50	1.5	75
<Write cycle>					
Write cycle time	t_{WC}	150	–	4	200
Address enable time	t_{AW}	120	–	3.5	175
Write pulse width	t_{WP}	100	–	3	150
Input data setup time	t_{DW}	60	–	3	150
Input data hold time	t_{DH}	0	–	0.5	25

SRAM: 150nS, CPU: 20MHz, read cycle



SRAM: 150nS, CPU: 20MHz, write cycle



A.23 SRAM (100nS)

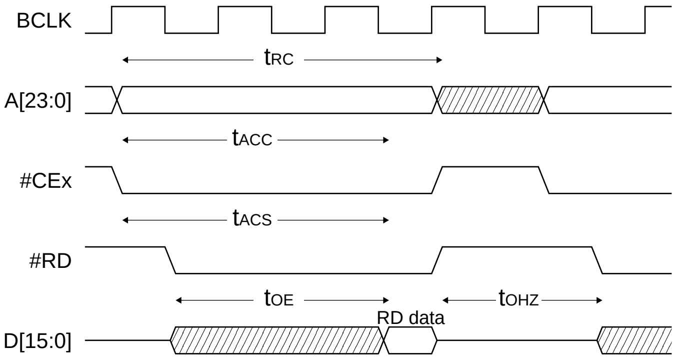
SRAM interface setup example – 100nS

Operating frequency	Read cycle		Write cycle	Output disable delay time
	Wait cycle	Read cycle		
20MHz	2	3	3	1.5

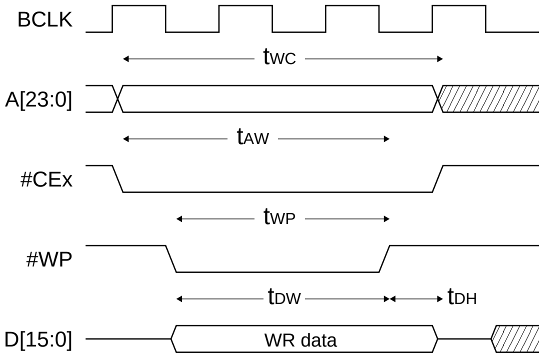
SRAM interface timing – 100nS

SRAM interface		2.7V		20MHz	
Parameter	Symbol	Min.	Max.	Cycle	Time
<Read cycle>					
Read cycle time	tRC	100	–	3	150
Address access time	tACC	–	100	3	150
#CE access time	tACS	–	100	3	150
#OE access time	tOE	–	60	2.5	125
Output disable delay time	tOHZ	0	35	1.5	75
<Write cycle>					
Write cycle time	tWC	100	–	3	150
Address enable time	tAW	90	–	2.5	125
Write pulse width	tWP	80	–	2	100
Input data setup time	tDW	60	–	2	100
Input data hold time	tDH	0	–	0.5	25

SRAM: 100nS, CPU: 20MHz, read cycle



SRAM: 100nS, CPU: 20MHz, write cycle



Appendix B Summary of Notes

CPU

(1) MAC execution error
The result of the MAC instructions may be incorrect when the following instructions appear right before the MAC instruction. The following instructions modify the %ALR and %AHR registers and this causes incorrect MAC results.
"ld.w %ALR, %Rs" "div*" "ld.w %ALR, %Rs" "mlt*"

To avoid this problem, do not place the above instructions right before the MAC instructions.

(2) Load and data-size conversion instruction instruction error
CASE 1: When executing instructions in area 3 or area 0 (internal RAM), the following combination of instruction execution may result in incorrect register values to the destination register of the first instruction.

```
ld.ub  %r12, [%r9]       ld.ub  %r12, [%r9]        ld.h  %r12, [%r4]        ld.w  %r12, [%r7]
                    or                        or                        or
ld.h   %r10, %r12        ld.h   %r10, [%r12]       ld.b  [%r4], %r12        jp    %r12
```

The %r12 may be incorrect.
Notice that the source register of the second instruction is the destination register of the first instruction and the second instruction is a data-size conversion instruction, load/store instruction or branch instruction.

CASE 2: When turning on the CBR refresh cycles for the external DRAM access, the above instruction combination may result in incorrect register values even if the instructions are fetched from an external memory.

CASE 3: When the source register and destination register of the data-size conversion instruction is the same, the result may be incorrect.

CASE 4: When executing a data-size conversion instruction, the results become invalid and a debug environment like ICE will not work properly.

To avoid this problem, a "FILTER" program is provided in the CC33 software development package. Carefully go through the readme file in the CC33\utility\filter directory before using.

BCU

By writing a byte to the BCU register (address 0x0048130), the data will be written to both 0x0048130 and 0x0048131 addresses. When writing a byte to address 0x0048131, the register's data will not be modified. Write in half-word or word units to address 0x0048130.

Interrupt

(1) In cases when an interrupt factor that is used for restarting from the standby mode has been set to invoke IDMA, IDMA is started up by the interrupt at its occurrence. In SLEEP mode, the high-speed (OSC3) oscillation circuit also starts operating. However, if an interrupt to be generated upon completion of IDMA is disabled at the setting of IDMA side, no interrupt request is signaled to the CPU. Therefore, the CPU remains idle until the next interrupt request is generated.

(2) As the E0C33000 Core CPU function, the IL allows interrupt levels to be set in the range of 0 to 15. However, since the interrupt priority register in the E0C33A104 consists of three bits, interrupt levels in each interrupt system can only be set for up to 8.

(3) When the reset-only method is used to reset the interrupt factor flag (by writing "1"), if a read-modify-write instruction (e.g., bset, bclr, or bnot) is executed, the other interrupt factor flags at the same address that have been set to "1" are reset by a write. This requires caution. In cases when the read/write method is used to reset the interrupt factor flag (by writing "0"), all factor flags for which "0" has been written are reset. When a read-modify-write operation is performed, an interrupt factor may occur between reads and writes, so be careful when using this method.

(4) After an initial reset, the interrupt factor flags, IDMA request registers, and interrupt priority registers all become indeterminate. To prevent unwanted interrupts or IDMA requests from being generated inadvertently, be sure to reset these flags and registers in the software application.

(5) To prevent another interrupt from being generated for the same factor again after generation of an interrupt, be sure to reset the interrupt factor flag before enabling interrupts and setting the PSR again or executing the reti instruction.

(6) When an interrupt is disabled using the interrupt enable bit (0x0040270–0x0040277) at the same time the corresponding interrupt factor occurs, the interrupt processing read a wrong vector from [TTBR+284(decimal)] (reserved vector number). Therefore the program does not jump to the correct interrupt service routine.
To avoid this problem, an interrupt enable bit in ITC should be reset by the following procedure:
1) Clear the IE bit in PSR.
2) Execute the NOP instruction
3) Reset the interrupt enable bit in ITC.

Example:

```
ld.w   %r0,%psr
ext    0x3ff
and    %r0,0x2f
ld.w   %psr,%r0   ; clear IE
nop
ld.b   [%r1],%r2  ; %r1=0x00040274, %r2=00000000
                  ; disable interrupt
```

High-speed DMA

(1) When setting the memory address and transfer conditions, always make sure the DMA controller is inactive (DMAEN = "0").

(2) Make sure the memory address register is always accessed for read and write operations in words (32 bits). If written in bytes (8 bits) or half-words (16 bits), the data becomes indeterminate. Similarly, when the register is read, indeterminate data is read out.

(3) No address of the internal memory can be specified for the destination or source of transfer. The address specified here must always be that of an external memory that is connected to the external system interface. If an internal memory address is specified, DMA may not operate properly.

(4) Since the pin function select registers (CFK, CFR, CFP), I/O control register (IOC), and pull-up control register (KPU) are write-only registers (read data is indeterminate), bit operation instructions (e.g., bset, bclr, or bnot) cannot be used to rewrite these registers. Use an ordinary store instruction.

(5) After an initial reset, the interrupt factor flag (FHDMx) and IDMA request register (RHDMx) become indeterminate. Always be sure to reset the flag and register to prevent interrupts or IDMA requests from being generated inadvertently.

(6) To prevent an interrupt from being generated repeatedly for the same cause, be sure to reset the interrupt factor flag before setting up the PSR again or executing the reti instruction.

(7) High-speed DMA is given higher priority over IDMA (intelligent DMA) and the CPU. However, since high-speed DMA and IDMA share the same circuit, high-speed DMA cannot gain the bus ownership while an IDMA transfer is under way. Requests for high-speed DMA invocation that have occurred during an IDMA transfer are kept pending until the IDMA transfer is completed.
A request for IDMA invocation or an interrupt request that has occurred during a high-speed DMA transfer are accepted after completion of the high-speed DMA transfer.

Intelligent DMA

(1) Before setting the IDMA base address, be sure to disable DMA transfers (DMAEN = "0"). Writes to the IDMA base address register during a DMA transfer are invalid. Also, when the register is read during a DMA transfer, the data is indeterminate. When setting or rewriting control information for each channel, make sure that DMA transfers will not occur in any channel.

(2) The address that is set in the IDMA base address register must always be a word (32-bit) boundary address.

(3) The IDMA base address register cannot be read or written in bytes. The register must be accessed in words for read/write to address 0x48230, and in half-words for read/write to addresses 0x48230 and 0x48232. Write operations in half-words must be performed in order of 0x48230 and 0x48232. Read operations in half-words may be performed in any order.

(4) After an initial reset, the interrupt factor flag (FIDMA) becomes indeterminate. To prevent unwanted interrupts from occurring, be sure to reset the flag in a program.

(5) Once an interrupt occurs, be sure to reset the interrupt factor flag (FIDMA) before setting up the PSR again or executing the reti instruction. This ensures that an interrupt will not be generated for the same factor.

(6) During an IDMA transfer, high-speed DMA requests and interrupt requests (even those of higher priority) are kept pending. These requests are accepted after the completion of the transfer.
Since the interrupt level (IL) is rewritten to that of the interrupt being processed and remains so during interrupt processing, an IDMA invocation request triggered by an interrupt factor whose priority is below that level is not accepted. If you want IDMA to be invoked by an interrupt factor of lower priority during interrupt processing, the interrupt factor flag for the interrupt being processed must be reset in the interrupt processing routine and rewrite the IL level.

(7) If an IDMA channel is set so that it generates an interrupt when the IDMA transfer has completed, the IDMA controller clears the IDMA request flag in ITC when the transfer counter becomes "0". This IDMA operation does not clear the interrupt factor flag in ITC, so ITC can generate an interrupt to the CPU.
An IDMA request flag register (byte-size-register) contains several IDMA request flags and the status of the IDMA request flags may be modified by a load/store operation. The CPU performs a read-modify-write operation in byte units for this modification.
Therefore, if an IDMA request flag is cleared due to the end of the transfer while the CPU is setting another IDMA request flag in the same register, the cleared IDMA request flag may be set again by the CPU's read-modify-write operation.
To avoid this problem, do not activate two or more IDMA channels assigned in the same IDMA request register at once. This problem does not occur if only one IDMA channel in an IDMA request flag register is enabled at a time.

(8) When "0" is written to bit 0 of address 0x004029f, interrupt flags in ITC will be modified by the read/write operation of the CPU (the default value is "1", reset-only method). With this setting, an IDMA transfer may not clear the interrupt flag, so the IDMA channel may be activated repeatedly with the same interrupt factor.
To avoid this problem, do not set "0" to bit 0 of address 0x004029f when using IDMA.

(9) If an IDMA trigger which has a higher interrupt level than that of the CPU happens when an interrupt occurs and the program jumps to the interrupt service routine, the interrupt level of the interrupt service routine may be set to the IDMA's interrupt level which is higher than the correct value.
When enabling nested interrupts by setting IE = 1 in PSR, the nested interrupt request from ITC may be masked by this incorrect interrupt level in PSR.
To enable nested interrupts in the interrupt service routine when IDMA is working, set the correct interrupt level to PSR at the same time as setting IE = 1 in the PSR. In other words, over-write the interrupt level by software in the interrupt service routine when enabling nested interrupts.

Oscillation circuit

(1) The high-speed (OSC3) oscillation circuit may not generate the clock signals if a fundamental crystal resonator is NOT being used. Use a fundamental resonator (MA-306 made by Seiko Epson, etc.) for the high-speed (OSC3) oscillation circuit.

(2) At initial reset, the low-speed (OSC1) oscillation circuit is configured for a crystal oscillation. For external clock input, use CLGRON to change the circuit configuration. This setting is required even for hot starts.

(3) Immediately after the oscillation circuit is turned on, a certain period of time is required for oscillation to stabilize (for OSC3 using a 3.3-V crystal resonator, this time is 10 ms max.; for OSC1, this time is 3 sec max. For details, refer to Section 18.8, "Oscillation Characteristics"). To prevent the device from operating erratically, do not use the clock until its oscillation has stabilized.
In particular, if the CPU is set in SLEEP mode during operation using the OSC3 clock, the high-speed (OSC3) oscillation circuit is turned off during in SLEEP mode and starts oscillating again after SLEEP mode is exited. To prevent the CPU from operating erratically at restart due to an unstable OSC3 clock, set a sufficient stabilization waiting time in 8-bit programmable timer 1 to turn on the oscillation stabilization waiting function after SLEEP mode is exited before entering SLEEP mode.

(4) The oscillation circuit used for the CPU operating clock cannot be turned off.

(5) The CPU operating clock can only be switched over when both oscillation circuits are on. Furthermore, when turning off an oscillation circuit that has become unnecessary as a result of the CPU operating clock switchover, be sure to use separate instructions for switchover and oscillation turnoff. If these two operations are processed simultaneously using one instruction, the CPU may operate erratically.

(6) If the high-speed (OSC3) oscillation circuit is turned off or the CPU operating clock is set for the OSC1 clock, all peripheral circuits operated using the OSC3 clock will be inactive.

(7) If the OSC3 clock is unnecessary, use the OSC1 clock to operate the CPU and turn the high-speed (OSC3) oscillation circuit off. This helps reduce current consumption.

Prescaler

(1) The prescaler operates only when the high-speed (OSC3) oscillation circuit is active and the OSC3 clock is set for the CPU operating clock. Note that the 16-bit and 8-bit programmable timers and the A/D converter do not operate when the high-speed (OSC3) oscillation circuit is inactive or when the OSC1 clock is selected for the CPU operating clock.

(2) In the following cases, the prescaler output clock may contain a hazard:
• If, during outputting of a clock, its division ratio is changed
• When the clock output is switched between on and off
• When the high-speed (OSC3) oscillation circuit is turned off or the CPU operating clock is switched over
Before performing these operations, make sure the 16-bit and 8-bit programmable timers and the A/D converter are turned off.

(3) When the 16-bit and 8-bit programmable timers and the A/D converter do not need to be operated, turn off the clock supply to those peripheral circuits. This helps to reduce current consumption.

Clock timer

(1) The low-speed (OSC1) oscillation circuit, which is the clock source for the clock timer, requires a muxmum of three seconds for its oscillation to stabilize after it is started up. Therefore, immediately after power-on, wait until the oscillation stabilizes before starting the clock timer.

(2) At initial reset, the clock timer counter data, the setup contents of alarms, and control bits, including RUN/STOP, are not initialized. (This does not include the CPU core power on/off flag TCHVOF and OSC1 auto-off flag TCAOFF.) Therefore, always initialize the clock timer in the software following power-on.

(3) The clock timer reset bit TCRST and the clock timer RUN/STOP control bit TCRUN are located at the same address (0x40151). However, the clock timer cannot be reset at the same time it is set to RUN by writing "1" to both. In this case, the reset input is ignored and the timer starts counting up from the counter values then in effect. When resetting the timer, always make sure TCRUN = "0" (timer stopped).

(4) When the counters are cleared as the clock timer is reset, an interrupt may be generated depending on the register settings. Therefore, before resetting the clock timer, first disable the clock timer interrupt and, after resetting the clock timer, reset the interrupt factor flag and the interrupt factor generation and alarm factor generation flags.

(5) To prevent generation of an unwanted interrupt, disable the clock timer interrupt before selecting the interrupt and alarm factors. Then, before reenabling the interrupt, reset each factor generation flag and the interrupt factor flag.

(6) The interrupt factor flag (FCTM) becomes indeterminate at initial reset. To prevent generation of an unwanted interrupt, be sure to reset the flag in a program.

(7) To prevent regeneration of interrupts with the same factor after an interrupt has occurred, be sure to reset the interrupt factor flag (FCTM) before setting the PSR again or executing the reti instruction.

(8) When a clock timer control register (address 0x40152, 0x40159, 0x4015A or 0x4015B) is rewritten within 4 msec after the clock timer's alarm or periodic interrupt has occurred, the same interrupt may occur again even though the interrupt factor flag in ITC and the factor generation flag in the clock timer are cleared in the clock timer interrupt service routine.
To avoid this problem, wait at least 4 msec after the clock timer interrupt has occurred before writing data to a clock timer control register (address 0x40152, 0x40159, 0x4015A or 0x4015B).
When using only one of the clock timer interrupt functions(alarm or periodic), this problem may be avoided since it is not necessary to reset the clock timer interrupt factor generation flag in address 0x40152. It is possible to use only the clock timer interrupt factor flag in ITC (bit 1 of 0x40287).

Watchdog timer

(1) If the watchdog timer's NMI is enabled, the watchdog timer must be preset in the software before the 16-bit programmable timer 0 underflows.

(2) Even when EWD is set to "0", the 16-bit programmable timer 0 does not stop counting. Therefore, if the NMI has been temporarily disabled, be sure to preset the 16-bit programmable timer 0 before setting EWD back to "1".

8-bit programmable timer

(1) The 8-bit programmable timer operates only when the prescaler is operating. The prescaler generates a clock for each timer from the OSC3 oscillation clock by dividing it as set using the register. When the CPU is operating using the low-speed (OSC1) clock, the prescaler is inactive, so the 8-bit programmable timer also cannot be used.

(2) The data set in the prescaler is the division ratio for the OSC3 oscillation frequency. If the CPU operating clock is generated from the OSC3 oscillation clock by dividing it using CLKDT[1:0] (D[7:6]) / Power control register (0x40140), do not use a clock that is faster than the CPU operating clock.

(3) When setting an input clock, make sure the 8-bit programmable timer is turned off.

(4) Since the underflow interrupt condition and the timer output status are undefined after an initial reset, the counter initial value should be set to the 8-bit timer before resetting the interrupt factor flag or turning the timer output on.

(5) The function select register (CFP) is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, and bnot) cannot be used to rewrite this register. Use ordinary storage instructions for this purpose.

(6) After an initial reset, the interrupt factor flag (F8TUx) and IDMA request register (R8TUx) become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset this flag and register in the software.

(7) To prevent another interrupt from being generated again by the same factor after an interrupt has occurred, be sure to reset the interrupt factor flag (F8TUx) before setting the PSR again or executing the reti instruction.

16-bit programmable timer

(1) The 16-bit programmable timers clocked by the internal clock operate only when the prescaler is operating. The prescaler generates a clock for each timer from the OSC3 oscillation clock by dividing it as set using the register. When the CPU is operating using the low-speed (OSC1) clock, the prescaler is inactive, so the 16-bit programmable timer also cannot be used.

(2) The data set in the prescaler is the division ratio for the OSC3 oscillation frequency. If the CPU operating clock is generated from the OSC3 oscillation clock by dividing it using CLKDT[1:0] (D[7:6]) / Power control register (0x40140), do not use a clock that is faster than the CPU operating clock.

(3) When using the timer as an event counter, make sure the event cycle is at least twice the CPU operating-clock period.

(4) When setting the input clock or operation mode, make sure the 16-bit programmable timer is turned off.

(5) Since the interrupt signal and the timer output status are undefined after an initial reset, the counter initial value should be set to the 16-bit timer before resetting the interrupt factor flag or turning the timer output on. However, the timer output is fixed at "0" when the function is turned off.

(6) When the comparison data register value is "0" (CRxx = "0") or larger than the reload data register value (RRxx < CRxx), the comparison match interrupt will not be generated. In this case, the timer output is left as "0".

(7) Because the function select, I/O control, and pull-up control registers used to set the input/output pins of the 16-bit programmable timer are write-only registers (read data is indeterminate), bit operation instructions (bset, bclr, and bnot) cannot be used to rewrite these registers. Use ordinary storage instructions for this purpose.

(8) After an initial reset, the interrupt factor flag and IDMA request register become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset this flag and register in the software.

(9) To prevent another interrupt from being generated by the same factor after an interrupt has occurred, be sure to reset the interrupt factor flag before setting the PSR again or executing the reti instruction.

Serial interface

(1) Before setting various serial-interface parameters, make sure the transmit and receive operations are disabled (TXENx = RXENx = "0").

(2) When the serial interface is transmitting or receiving data, do not set TXENx or RXENx to "0", and do not execute the slp instruction.

(3) In clock-synchronized transfers, the mode of communication is half-duplex, in which the clock line is shared between the transmit and receive units. Therefore, RXENx and TXENx cannot be enabled simultaneously.

(4) Since the pin function select register (CFP) is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.

(5) After an initial reset, the interrupt factor flag and IDMA request register become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, reset this flag and this register in the program.

(6) If a receive error occurs, the receive-error interrupt and receive-buffer full interrupt factors occur simultaneously. However, since the receive-error interrupt has priority over the receive-buffer full interrupt, the receive-error interrupt is processed first. Therefore, it is necessary to reset the receive-buffer full interrupt factor flag through the use of the receive-error interrupt processing routine

(7) To prevent the regeneration of interrupts due to the same factor following the occurrence of an interrupt, always be sure to reset the interrupt factor flag before setting the PSR again or executing the reti instruction.

(8) Follow the procedure described below to initialize the serial interface.

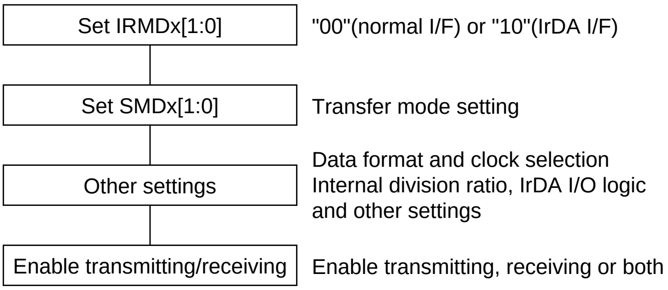

Set IRMDx[1:0]
"00"(normal I/F) or "10"(IrDA I/F)
Set SMDx[1:0]
Transfer mode setting
Other settings
Data format and clock selection
Internal division ratio, IrDA I/O logic
and other settings
Enable transmitting/receiving
Enable transmitting, receiving or both

Figure B.1 Serial Interface Initialize Procedure

(9) The maximum transfer rate in the clock-synchronized mode is limited to 1 Mbps.
The maximum transfer clock (TCLK) frequency in the asynchronous mode is limited to 1 MHz.

(10) When sending data in clock-synchronized master mode, follow the steps below.
1) Turn on the 8-bit timer output to the serial interface.
2) Wait until at least one underflow pulse is provided to the serial interface from the 8-bit timer.
3) Write sending data to the TXD data register of the serial interface.

If the above steps are not followed, the serial interface may send 0xFF data first before the correct data is transferred.

(11) When using the serial interface in the clock-synchronized mode, be aware that the #SRDYx and #SCLKx pins does not change their I/O direction even if only the transfer mode is changed.
The I/O direction of the #SRDYx and #SCLKx pins are changed as follows (when the P0x port is set for the serial interface):
#SRDYx (P03/P07): When transmit/receive operation is enabled in slave mode, the #SRDYx pin enters output mode.
Otherwise, the #SRDYx pin stays in input mode.
#SCLKx (P02/P06): When transmit/receive operation is enabled in master mode, the #SRDYx pin enters output mode.
Otherwise, the #SRDYx pin stays in input mode.

The SINx (P00/P04) and SOUTx (P01/P05) pins are set to input mode and output mode, respectively, immediately after the P0x port is set for the serial interface.

(12) When receiving data in asynchronous mode, a framing error may easily occur if the serial clock source is different between this serial interface and the serial output device.
To avoid this problem, set 2 stop bits for the serial output device and set 1 stop bit for this serial interface, or use the same clock source for this serial interface and the serial output device.

A/D converter

(1) Before setting the conversion mode, start/end channels, etc. for the A/D converter, be sure to disable the A/D converter (ADE (D2) / A/D enable register (0x40244) = "0"). A change in settings while the A/D converter is enabled could cause it to operate erratically.

(2) The A/D converter operates only when the prescaler is operating. The prescaler generates a clock for the A/D converter from the OSC3 oscillation clock by dividing it. If the CPU is operating using the low-speed (OSC1) clock, the prescaler is inactive, so the A/D converter cannot be used. (Refer to Section 10.2, "Prescaler and Operating Clock for Peripheral Circuits".)
When the A/D converter registers are set up, the prescaler must be operating. Therefore, start the prescaler first and make sure the A/D converter is supplied with its operating clock before setting up the A/D converter registers.

The value set in the prescaler is a division ratio for the OSC3 oscillation frequency. If the CPU operating clock is generated from the OSC3 oscillation clock by dividing it using CLKDT[1:0] (D[7:6]) / Power control register (0x40140), do not use a clock that is faster than the CPU operating clock.
In consideration of the conversion accuracy, we recommends that the A/D converter operating clock be 2 MHz (max.).

(3) Do not start an A/D conversion when the clock supplied from the prescaler to the A/D converter is turned off, and do not turn off the prescaler's clock output when an A/D conversion is underway, as doing so could cause the A/D converter to operate erratically.

(4) Since the pin function select register (CFK) is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.

(5) After an initial reset, the interrupt factor flag (FADE) and IDMA request register (RADE) become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset this flag and register in a program.

(6) To prevent the regeneration of interrupts due to the same factor following the occurrence an interrupt, always be sure to reset the interrupt factor flag before setting the PSR again or executing the reti instruction.

D/A converter

(1) Before setting the D/A converter pins and operating voltage, be sure to disable the D/A converter (DAE0 (D1) and DAE1 (D0) / D/A output control register (0x40246) = "0"). A change in settings while the D/A converter is enabled could cause it to operate erratically.

(2) Since the pin function select register (CFK) is a write-only register (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite it. Use ordinary storage instructions for this purpose.

Input port

(1) When using the internal pull-up resistor, note that when the state of the input pin changes from low to high, a time constant comprised of the pull-up resistor and the pin's load capacitance causes a delay in the rising transition of the waveform. For this reason, an appropriate wait time must be provided prior to taking the input port into the internal logic. This is especially important for key scanning in a key-matrix configuration. The wait time set here must be equal to or greater than the value calculated from the following equation:

Wait time = $R_{IN} \times (C_{IN}$ + load capacitance on the board) $\times 1.6$ [sec]
R_{IN} = pull-up resistance (max.)
C_{IN} = pin capacitance (max.)

(2) Since the pin function select register (CFK) and the pull-up control register (KPU) are write-only registers (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite them. Use ordinary storage instructions for this purpose.

(3) After an initial reset, the interrupt factor flag (FK) and IDMA request register (RK) become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset this flag and register in a program.

(4) To prevent regeneration of interrupts due to the same factor following the occurrence of an interrupt, always be sure to reset the interrupt factor flag (FK) before resetting the PSR or executing the reti instruction.

Output port

Since the pin function select register (CFR) and R81 function extension bit (CFEX2) are write-only register/bit (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite the registers. Use ordinary storage instructions for this purpose.

I/O port

(1) When the internal pull-up resistor is used, note that when the state of the input pin changes from low to high, a time constant comprised of the pull-up resistor and the pin's load capacitance causes a delay in the rising transition of the waveform. For this reason, an appropriate wait time must be provided before taking the input data into the internal logic. This is especially important for key scanning in a key-matrix configuration. The wait time set here must be equal to or greater than the value calculated from the following equation:

Wait time = RIN × (CIN + load capacitance on the board) × 1.6 [sec]
RIN = pull-up resistance (max.)
CIN = pin capacitance (max.)

(2) Since the pin function select register (CFP), function extension register (CFEX), I/O control register (IOC), and pull-up control register (IOU) are all write-only registers (read data is indeterminate), bit operation instructions (bset, bclr, or bnot) cannot be used to rewrite the registers. Use ordinary storage instructions for this purpose.

Debugging function

Refer to the "E0C33 Family C Compiler Package Manual" for details of the debugging functions.

(1) Hardware PC break function
There are the following limitations in the hardware PC break function.
Be careful not to generate these conditions during software development.

- When a hardware PC break point is set at the internal memory address following the instruction that issues an address unaligned exception, the correct hardware PC break will not occur at the set address.
Example:

```
0080320   ld.w  [%r0], %r1     ← Issues address unaligned exception
0080322   add   %r2, 0x01      ← Set as a hardware PC break point
```

A hardware PC break actually occurs at address 0x0000320, whereas it should occur at address 0x000322.

- When a hardware PC break point is set at an internal memory address after the instruction that accesses the external memory in which the bus condition is set as 2 wait-cycles, the set PC break point may be ignored.
Example:

```
0080320   ld.ub %r12, [External memory]
0080322   ld.h  %r10, %r12
0080324   ld.w  %r11, %r10     ← Set as hardware PC break point 1
0080326   cmp   %r10, 0x00     ← Set as hardware PC break point 2
```

Both PC breaks 1 and 2 above may not occur (may be ignored).

- When a hardware PC break point is set at an external memory address within the continuous call instructions, the hardware PC break may not occur.
Example:

```
0C00300   call  LABEL01
0C00302   call  LABEL02        ← Set as a hardware PC break point
0C00304   call  LABEL03
```

A hardware PC break will not occur at address 0x00C00302.
To avoid this problem, insert a NOP instruction between the call instructions.

- When a hardware PC break point is set at the first instruction address of a trap service routine, the hardware PC break may not occur.

  Example:

```
0080040   INT                   ← Interrupt vector address

LOOP:
0080300   nop                   ← An interrupt occurs here
0080302   jp     LOOP

INT:
0080500   ld.w   %r0, 0x00      ← Set as a hardware PC break point
0080502   ld.w   %r1, 0x01
            :
```

  A hardware PC break will not occur at address 0x0080500. In this case, set at address 0x0080502.

(2) Data break function

When a data break is set at the bset, bclr or btst instruction, the data break may not occur.

(3) Single step function

There are the following limitations in the single step function.
Be careful not to generate these conditions during software development.

- When an address unaligned exception occurs during single step execution, the exception routine may not be executed in single stepping.

  Example:

```
0080018   ADERR                 ← Vector address of the address unaligned exception

0080300   add    %r0, 0x01
            |
            |    Single step execution
            ↓
0080302   ld.w   [%r1], %r0     ← Address unaligned exception occurs here

ADERR:
0080500   add    %r1, 0x01      ← Address unaligned exception routine starts here
            :
```

  When an address unaligned exception occurs at address 0x0080302 during single step execution, the next PC should be 0x0080500, but it becomes 0x0080302.

- The data break point set in the internal memory does not generate a data break during single step execution. The data break set in the external memory works correctly.
- When a divide-by-zero exception occurs during step execution, the exception processing starts, so the debug function will not be able to work properly.

EPSON International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.

- HEADQUARTERS -
1960 E. Grand Avenue
El Segundo, CA 90245, U.S.A.
Phone: +1-310-955-5300 Fax: +1-310-955-5400

- SALES OFFICES -

West
150 River Oaks Parkway
San Jose, CA 95134, U.S.A.
Phone: +1-408-922-0200 Fax: +1-408-922-0238

Central
101 Virginia Street, Suite 290
Crystal Lake, IL 60014, U.S.A.
Phone: +1-815-455-7630 Fax: +1-815-455-7633

Northeast
301 Edgewater Place, Suite 120
Wakefield, MA 01880, U.S.A.
Phone: +1-781-246-3600 Fax: +1-781-246-5443

Southeast
3010 Royal Blvd. South, Suite 170
Alpharetta, GA 30005, U.S.A.
Phone: +1-877-EEA-0020 Fax: +1-770-777-2637

EUROPE

EPSON EUROPE ELECTRONICS GmbH

- HEADQUARTERS -
Riesstrasse 15
80992 Muenchen, GERMANY
Phone: +49-(0)89-14005-0 Fax: +49-(0)89-14005-110

- GERMANY -
SALES OFFICE
Altstadtstrasse 176
51379 Leverkusen, GERMANY
Phone: +49-(0)217-15045-0 Fax: +49-(0)217-15045-10

- UNITED KINGDOM -
UK BRANCH OFFICE
2.4 Doncastle House, Doncastle Road
Bracknell, Berkshire RG12 8PE, ENGLAND
Phone: +44-(0)1344-381700 Fax: +44-(0)1344-381701

- FRANCE -
FRENCH BRANCH OFFICE
1 Avenue de l' Atlantique, LP 915 Les Conquerants
Z.A. de Courtaboeuf 2, F-91976 Les Ulis Cedex, FRANCE
Phone: +33-(0)1-64862350 Fax: +33-(0)1-64862355

ASIA

- CHINA -
EPSON (CHINA) CO., LTD.
28F, Beijing Silver Tower 2# North RD DongSanHuan
ChaoYang District, Beijing, CHINA
Phone: 64106655 Fax: 64107320

SHANGHAI BRANCH
4F, Bldg., 27, No. 69, Gui Jing Road
Caohejing, Shanghai, CHINA
Phone: 21-6485-5552 Fax: 21-6485-0775

- HONG KONG, CHINA -
EPSON HONG KONG LTD.
20/F., Harbour Centre, 25 Harbour Road
Wanchai, HONG KONG
Phone: +852-2585-4600 Fax: +852-2827-4346
Telex: 65542 EPSCO HX

- TAIWAN, R.O.C. -
EPSON TAIWAN TECHNOLOGY & TRADING LTD.
10F, No. 287, Nanking East Road, Sec. 3
Taipei, TAIWAN, R.O.C.
Phone: 02-2717-7360 Fax: 02-2712-9164
Telex: 24444 EPSONTB

HSINCHU OFFICE
13F-3, No. 295, Kuang-Fu Road, Sec. 2
HsinChu 300, TAIWAN, R.O.C.
Phone: 03-573-9900 Fax: 03-573-9169

- SINGAPORE -
EPSON SINGAPORE PTE., LTD.
No. 1 Temasek Avenue, #36-00
Millenia Tower, SINGAPORE 039192
Phone: +65-337-7911 Fax: +65-334-2716

- KOREA -
SEIKO EPSON CORPORATION KOREA OFFICE
50F, KLI 63 Bldg., 60 Yoido-Dong
Youngdeungpo-Ku, Seoul, 150-010, KOREA
Phone: 02-784-6027 Fax: 02-767-3677

- JAPAN -
SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

Electronic Device Marketing Department
IC Marketing & Engineering Group
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5816 Fax: +81-(0)42-587-5624

ED International Marketing Department I (Europe & U.S.A.)
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5812 Fax: +81-(0)42-587-5564

ED International Marketing Department II (Asia)
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5814 Fax: +81-(0)42-587-5110

In pursuit of **“Saving” Technology**, Epson electronic devices.
Our lineup of semiconductors, liquid crystal displays and quartz devices
assists in creating the products of our customers’ dreams.
Epson IS energy savings.

EPSON

SEIKO EPSON CORPORATION

ELECTRONIC DEVICES MARKETING DIVISION

■ EPSON Electronic Devices Website

http://www.epson.co.jp/device/

First issue JULY 1998, Printed OCTOBER 1999 in Japan Ⓜ B