MF805-04

EPSON

CMOS 8-BIT SINGLE CHIP MICROCOMPUTER

E0C88112/88104 TECHNICAL MANUAL

E0C88112/88104 Technical Hardware

E0C88112/88104 Technical Software

ENERGY SAVING EPSON

SEIKO EPSON CORPORATION

NOTICE

No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Control Law of Japan and may require an export license from the Ministry of International Trade and Industry or other approval from another government agency. Please note that "E0C" is the new name for the old product "SMC". If "SMC" appears in other manuals understand that it now reads "E0C".

CMOS 8-bit Single Chip Microcomputer

E0C88112/88104 Technical Manual

Introduction

This Manual contains separate descriptions of the hardware and software of the E0C88112/88104 CMOS 8-bit single chip microcomputers.

I. E0C88112/88104 Technical Hardware

This section of the Manual describes the functions, circuit configuration and control system of the E0C88112/88104.

II. E0C88112/88104 Technical Software

This section of the Manual describes the programming of the E0C88112/88104.

I

E0C88112/88104 Technical Hardware

Contents

1 INTRODUCTION I-1
1.1 Configuration I-1
1.2 Features I-2
1.3 Block Diagram I-3
1.4 Pin Layout Diagram I-5
2 POWER SUPPLY I-7
2.1 Operating Voltage I-7
2.2 Internal Power Supply Circuit I-7
2.3 Heavy Load Protection Mode I-7
3 CPU AND BUS CONFIGURATION I-8
3.1 CPU I-8
3.2 Internal Memory I-8
3.2.1 ROM I-8
3.2.2 RAM I-8
3.2.3 I/O memory I-8
3.3 Exception Processing Vectors I-9
3.4 CC (Customized Condition Flag) I-9
3.5 Chip Mode I-10
3.5.1 MCU mode and MPU mode I-10
3.5.2 Bus mode I-11
3.6 External Bus I-14
3.6.1 Data bus I-14
3.6.2 Address bus I-14
3.6.3 Read ($\overline{RD}$)/write ($\overline{WR}$) signals I-15
3.6.4 Chip enable ($\overline{CE}$) signal I-15
3.6.5 WAIT control I-17
3.6.6 Bus authority release state I-18
4 INITIAL RESET I-19
4.1 Initial Reset Factors I-19
4.1.1 $\overline{RESET}$ terminal I-20
4.1.2 Simultaneous LOW level input at input port terminals K00–K03 ..... I-20
4.1.3 Supply voltage detection (SVD) circuit I-20
4.1.4 Initial reset sequence I-20
4.2 Initial Settings After Initial Reset I-21
5 PERIPHERAL CIRCUITS AND THEIR OPERATION I-22
5.1 I/O Memory Map I-23
5.2 System Controller and Bus Control I-35
5.2.1 Bus mode settings I-35
5.2.2 Address decoder ($\overline{CE}$ output) settings I-37
5.2.3 WAIT state settings I-38
5.2.4 Setting the bus authority release request signal I-38
5.2.5 Stack page setting I-38
5.2.6 Control of system controller I-39
5.2.7 Programming notes I-42

5.3 Watchdog Timer I-43
5.3.1 Configuration of watchdog timer I-43
5.3.2 Interrupt function I-43
5.3.3 Control of watchdog timer I-43
5.3.4 Programming note I-43
5.4 Oscillation Circuits and Operating Mode I-44
5.4.1 Configuration of oscillation circuits I-44
5.4.2 Mask option I-44
5.4.3 OSC1 oscillation circuit I-44
5.4.4 OSC3 oscillation circuit I-45
5.4.5 Operating mode I-45
5.4.6 Switching the CPU clocks I-46
5.4.7 Control of oscillation circuit and operating mode I-47
5.4.8 Programming notes I-48
5.5 Input Ports (K ports) I-49
5.5.1 Configuration of input ports I-49
5.5.2 Mask option I-49
5.5.3 Interrupt function and input comparison register I-50
5.5.4 Control of input ports I-52
5.5.5 Programming note I-54
5.6 Output Ports (R ports) I-55
5.6.1 Configuration of output ports I-55
5.6.2 Mask option I-56
5.6.3 High impedance control I-56
5.6.4 DC output I-57
5.6.5 Special output I-57
5.6.6 Control of output ports I-59
5.6.7 Programming notes I-63
5.7 I/O Ports (P ports) I-64
5.7.1 Configuration of I/O ports I-64
5.7.2 Mask option I-64
5.7.3 I/O control registers and I/O mode I-65
5.7.4 Control of I/O ports I-65
5.7.5 Programming notes I-66
5.8 Serial Interface I-67
5.8.1 Configuration of serial interface I-67
5.8.2 Mask option I-68
5.8.3 Transfer modes I-68
5.8.4 Clock source I-69
5.8.5 Transmit-receive control I-70
5.8.6 Operation of clock synchronous transfer I-71
5.8.7 Operation of asynchronous transfer I-75
5.8.8 Interrupt function I-79
5.8.9 Control of serial interface I-81
5.8.10 Programming notes I-85
5.9 Clock Timer I-86
5.9.1 Configuration of clock timer I-86
5.9.2 Interrupt function I-86
5.9.3 Control of clock timer I-88
5.9.4 Programming notes I-90
5.10 Stopwatch Timer I-91
5.10.1 Configuration of stopwatch timer I-91
5.10.2 Count up pattern I-91

5.10.3 Interrupt function I-92
5.10.4 Control of stopwatch timer I-93
5.10.5 Programming notes I-95

5.11 Programmable Timer I-96
5.11.1 Configuration of programmable timer I-96
5.11.2 Count operation and setting basic mode I-96
5.11.3 Setting of input clock I-98
5.11.4 Timer mode I-98
5.11.5 Event counter mode I-99
5.11.6 Pulse width measurement timer mode I-99
5.11.7 Interrupt function I-100
5.11.8 Setting of TOUT output I-100
5.11.9 Transmission rate setting of serial interface I-101
5.11.10 Control of programmable timer I-102
5.11.11 Programming notes I-108

5.12 Sound Generator I-109
5.12.1 Configuration of sound generator I-109
5.12.2 Control of buzzer output I-109
5.12.3 Setting of buzzer frequency and sound level I-110
5.12.4 Digital envelope I-110
5.12.5 One-shot output I-111
5.12.6 Control of sound generator I-112
5.12.7 Programming notes I-114

5.13 Analog Comparator I-115
5.13.1 Configuration of analog comparator I-115
5.13.2 Mask option I-115
5.13.3 Analog comparator operation I-115
5.13.4 Control of analog comparator I-116
5.13.5 Programming notes I-116

5.14 Supply Voltage Detection (SVD) Circuit I-117
5.14.1 Configuration of SVD circuit I-117
5.14.2 Operation of SVD circuit I-117
5.14.3 Control of SVD circuit I-119
5.14.4 Programming notes I-120

5.15 Interrupt and Standby Status I-121
5.15.1 Interrupt generation conditions I-122
5.15.2 Interrupt factor flag I-122
5.15.3 Interrupt enable register I-123
5.15.4 Interrupt priority register and interrupt priority level I-123
5.15.5 Exception processing vectors I-124
5.15.6 Control of interrupt I-125
5.15.7 Programming notes I-126

5.16 Notes for Low Current Consumption I-127

6 BASIC EXTERNAL WIRING DIAGRAM I-128

7 ELECTRICAL CHARACTERISTICS I-129

7.1 Absolute Maximum Rating I-129

7.2 Recommended Operating Conditions I-129

7.3 DC Characteristics I-130

7.4 Analog Circuit Characteristics I-131

7.5 Power Current Consumption I-132

7.6 AC Characteristics I-133
7.7 Oscillation Characteristics I-140
7.8 Characteristics Curves (reference value) I-141

8 PACKAGE I-148
8.1 Plastic Package I-148
8.2 Ceramic Package I-150

9 PAD LAYOUT I-151
9.1 Diagram of Pad Layout I-151
9.2 Pad Coordinates I-153

1 INTRODUCTION

The E0C88112 microcomputer features the E0C88 (Model 3) CMOS 8-bit core CPU along with 12K bytes of ROM, 256 bytes of RAM, three different timers and a serial interface with optional asynchronization or clock synchronization.
Also provided is the E0C88104, which has the same peripheral circuits and is configured with 4K bytes of ROM.

These devices are fully operable over a wide range of voltages, and can perform high speed operations even at low voltage. Like all the equipment in the E0C Family, these microcomputers have low power consumption.

A 19-bit external address bus and 4 bits chip enable signals make it possible for these microcomputers to control up to 512K × 4 bytes of memory, making them ideal for high performance data bank systems.

1.1 Configuration

In this manual, the E0C88112 is associated with E0C88112 and E0C88104. In these models, there is a difference in built-in ROM capacity, but the other peripheral circuits are made with the same configuration.

Table 1.1.1 Configuration

Model	Internal ROM
E0C88112	12K byte
E0C88104	4K byte

1.2 Features

Table 1.2.1 lists the features of the E0C88112 and E0C88104.

Table 1.2.1 Main features

Model		E0C88112	E0C88104
Core CPU		E0C88 (MODEL3) CMOS 8-bit core CPU	
OSC1 Oscillation circuit		Crystal oscillation circuit/CR oscillation circuit/external clock input 32.768 kHz (Typ.)	
OSC3 Oscillation circuit		Crystal oscillation circuit/ceramic oscillation circuit/CR oscillation circuit/external clock input 4.2 MHz/3 V, 8.2 MHz /5 V	
Instruction set		608 types (Usable for multiplication and division instructions)	
Min. instruction execution time		0.48 µsec/4.2 MHz (2 clock)	
Internal ROM capacity		12K byte	4K byte
Internal RAM capacity		256 byte	
Bus line		Address bus: 19 bits (Also usable as a general output port when not used as a bus) Data bus: 8 bits (Also usable as a general I/O port when not used as a bus) $\overline{\text{CE}}$ signal: 4 bits $\overline{\text{WR}}$ signal: 1 bit (Also usable as a general output port when not used as a bus) $\overline{\text{RD}}$ signal: 1 bit	
Input port		10 bits (2 bits can be set for event counter external clock input and bus request signal input terminal)	
Output port		9 bits (4 bits can be set for buzzer output, FOUT, TOUT and bus acknowledge signal output terminal)	
I/O port		8 bits (4 bits each can be set for serial interface input/output and analog comparator input)	
Serial interface		1ch (Optional clock synchronous system or asynchronous system)	
Timer		Programmable timer (8 bits): 2ch (1ch can be set as a an event counter or 2ch as a 16 bits programmable timer for 1ch) Clock timer (8 bits): 1ch Stopwatch timer (8 bits): 1ch	
Sound generator		Envelope function, equipped with volume control	
Watchdog timer		Built-in	
Analog comparator		2ch built-in	
Supply voltage detection (SVD) circuit		Can detect up to 16 different voltage levels	
Interrupt		External interrupt: Input interrupt 2 systems (3 types) Internal interrupt: Timer interrupt 3 systems (9 types) Serial interface interrupt 1 system (3 types)	
Supply voltage		Normal mode: 2.4 V–5.5 V (Max. 4.2 MHz) Low power mode: 1.8 V–3.5 V (Max. 50 kHz) High speed mode: 3.5 V–5.5 V (Max. 8.2 MHz)	
Consumed current	SLEEP	300 nA (Typ./normal mode)	
	HALT (32.768 kHz)	2 µA (Typ./normal mode)	
	In operation (32.768 kHz)	14 µA (Typ./normal mode)	
	In operation (4.2 MHz)	2 mA (Typ./normal mode)	
Supply form		QFP14-80 pin/QFP15-100 pin or chip	

* The number of bits cited for output ports and I/O ports does not include those shared with the bus.

1.3 Block Diagram

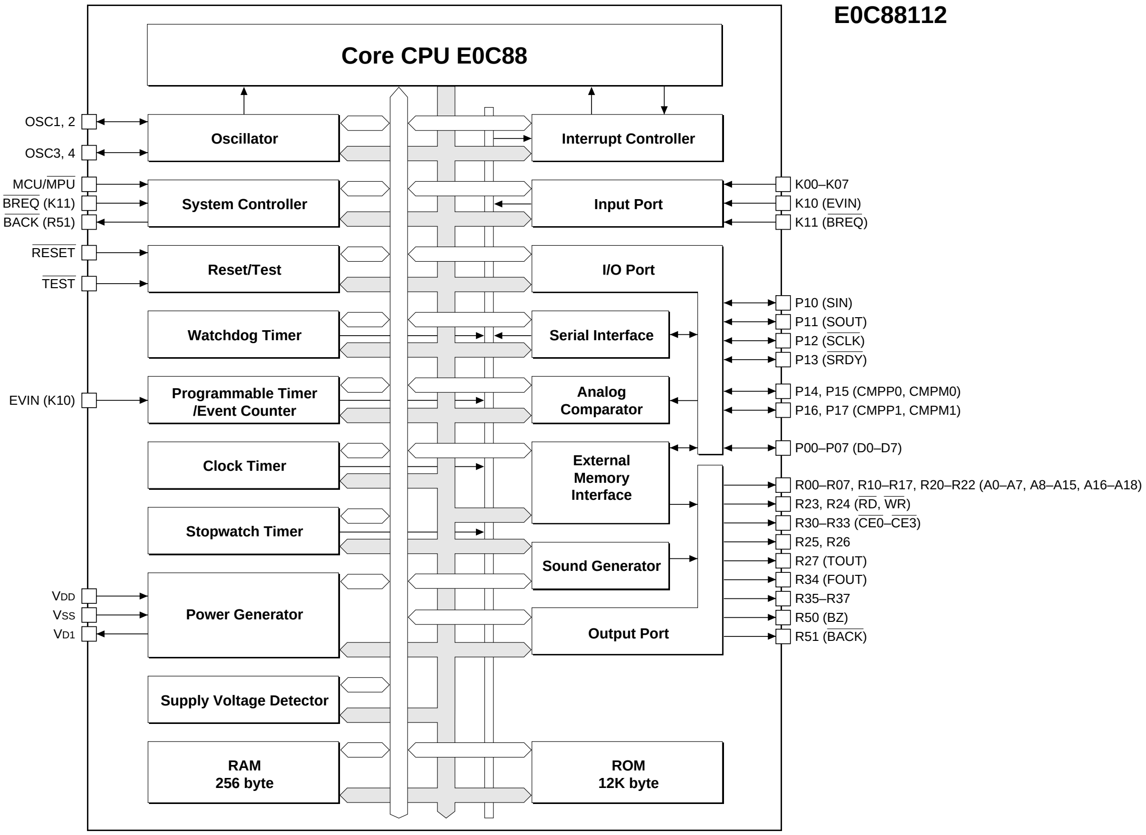


Fig. 1.3.1 E0C88112 block diagram

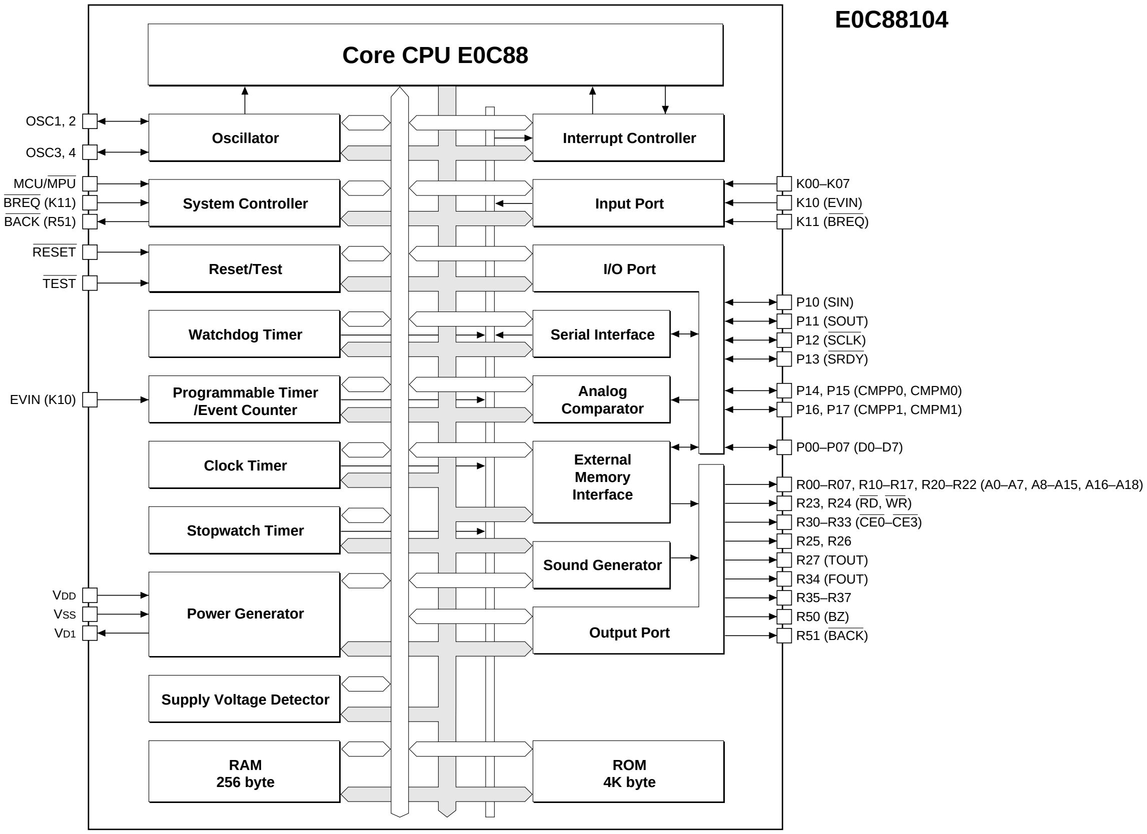
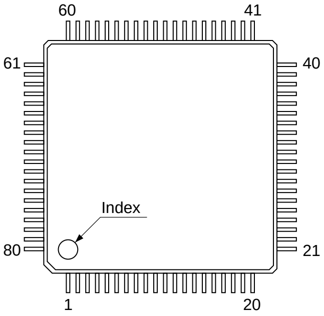


Fig. 1.3.2 E0C88104 block diagram

1.4 Pin Layout Diagram

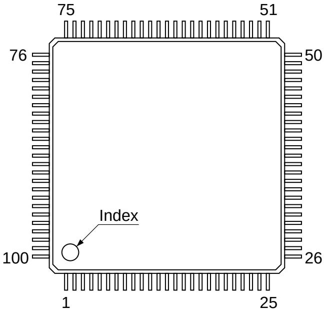
QFP14-80 pin



Pin No.	Pin name	Pin No.	Pin name	Pin No.	Pin name	Pin No.	Pin name
1	R00/A0	21	R24/$\overline{WR}$	41	R50/BZ	61	$\overline{RESET}$
2	R01/A1	22	R25	42	R51/$\overline{BACK}$	62	N.C.
3	R02/A2	23	R26	43	P17/CMPM1	63	K11/$\overline{BREQ}$
4	R03/A3	24	R27/TOUT	44	P16/CMPP1	64	K10/EVIN
5	R04/A4	25	R30/$\overline{CE0}$	45	P15/CMPM0	65	K07
6	R05/A5	26	R31/$\overline{CE1}$	46	P14/CMPP0	66	K06
7	R06/A6	27	R32/$\overline{CE2}$	47	P13/$\overline{SRDY}$	67	K05
8	R07/A7	28	R33/$\overline{CE3}$	48	P12/$\overline{SCLK}$	68	K04
9	R10/A8	29	R34/FOUT	49	P11/SOUT	69	K03
10	R11/A9	30	R35	50	P10/SIN	70	K02
11	R12/A10	31	R36	51	P07/D7	71	K01
12	R13/A11	32	R37	52	P06/D6	72	K00
13	R14/A12	33	*	53	P05/D5	73	MCU/$\overline{MPU}$
14	R15/A13	34	*	54	P04/D4	74	V_{DD}
15	R16/A14	35	*	55	P03/D3	75	OSC4
16	R17/A15	36	*	56	P02/D2	76	OSC3
17	R20/A16	37	*	57	P01/D1	77	V_{D1}
18	R21/A17	38	*	58	P00/D0	78	OSC2
19	R22/A18	39	*	59	N.C.	79	OSC1
20	R23/$\overline{RD}$	40	*	60	$\overline{TEST}$	80	V_{SS}

* Pins No. 33 to 40 are the pads used for outgoing inspection of the IC. Do not connect anything to these pins.

QFP15-100 pin



Pin No.	Pin name	Pin No.	Pin name	Pin No.	Pin name	Pin No.	Pin name
1	N.C.	26	N.C.	51	N.C.	76	N.C.
2	N.C.	27	N.C.	52	N.C.	77	N.C.
3	N.C.	28	R00/A0	53	R24/$\overline{WR}$	78	N.C.
4	$\overline{RESET}$	29	R01/A1	54	R25	79	R50/BZ
5	N.C.	30	R02/A2	55	R26	80	R51/$\overline{BACK}$
6	K11/$\overline{BREQ}$	31	R03/A3	56	R27/TOUT	81	P17/CMPM1
7	K10/EVIN	32	R04/A4	57	R30/$\overline{CE0}$	82	P16/CMPP1
8	K07	33	R05/A5	58	R31/$\overline{CE1}$	83	P15/CMPM0
9	K06	34	R06/A6	59	R32/$\overline{CE2}$	84	P14/CMPP0
10	K05	35	R07/A7	60	R33/$\overline{CE3}$	85	P13/$\overline{SRDY}$
11	K04	36	R10/A8	61	R34/FOUT	86	P12/$\overline{SCLK}$
12	K03	37	R11/A9	62	R35	87	P11/SOUT
13	K02	38	R12/A10	63	R36	88	P10/SIN
14	K01	39	R13/A11	64	R37	89	P07/D7
15	K00	40	R14/A12	65	*	90	P06/D6
16	MCU/$\overline{MPU}$	41	R15/A13	66	*	91	P05/D5
17	V_{DD}	42	R16/A14	67	*	92	P04/D4
18	OSC4	43	R17/A15	68	*	93	P03/D3
19	OSC3	44	R20/A16	69	*	94	P02/D2
20	V_{D1}	45	R21/A17	70	*	95	P01/D1
21	OSC2	46	R22/A18	71	*	96	P00/D0
22	OSC1	47	R23/$\overline{RD}$	72	*	97	N.C.
23	V_{SS}	48	N.C.	73	N.C.	98	$\overline{TEST}$
24	N.C.	49	N.C.	74	N.C.	99	N.C.
25	N.C.	50	N.C.	75	N.C.	100	N.C.

* Pins No. 65 to 72 are the pads used for outgoing inspection of the IC. Do not connect anything to these pins.

Fig. 1.4.1 E0C88112/88104 pin layout

Table 1.4.1 E0C88112/88104 pin description

Pin name	Pin No.		In/out	Function
	QFP14-80	QFP15-100		
VDD	74	17	I	Power supply (+) terminal
VSS	80	23	I	Power supply (GND) terminal
VD1	77	20	–	Regulated voltage output terminal for oscillators
OSC1	79	22	I	OSC1 oscillation input terminal (select crystal oscillation/CR oscillation/external clock input with mask option)
OSC2	78	21	O	OSC1 oscillation output terminal
OSC3	76	19	I	OSC3 oscillation input terminal (select crystal/ceramic/CR oscillation/external clock input with mask option)
OSC4	75	18	O	OSC3 oscillation output terminal
MCU/$\overline{\text{MPU}}$	73	16	I	Terminal for setting MCU or $\overline{\text{MPU}}$ modes
K00–K07	72–65	15–8	I	Input port (K00–K07) terminal
K10/EVIN	64	7	I	Input port (K10) terminal or event counter external clock (EVIN) input terminal
K11/$\overline{\text{BREQ}}$	63	6	I	Input port (K11) terminal or bus request signal ($\overline{\text{BREQ}}$) input terminal
R00–R07/A0–A7	1–8	28–35	O	Output port (R00–R07) terminals or address bus (A0–A7)
R10–R17/A8–A15	9–16	36–43	O	Output port (R10–R17) terminals or address bus (A8–A15)
R20–R22/A16–A18	17–19	44–46	O	Output port (R20–R22) terminals or address bus (A16–A18)
R23/$\overline{\text{RD}}$	20	47	O	Output port (R23) terminal or read signal ($\overline{\text{RD}}$) output terminal
R24/$\overline{\text{WR}}$	21	53	O	Output port (R24) terminal or write signal ($\overline{\text{WR}}$) output terminal
R25	22	54	O	Output port (R25) terminal
R26	23	55	O	Output port (R26) terminal
R27/TOUT	24	56	O	Output port (R27) terminal or programmable timer underflow signal (TOUT) output terminal
R30–R33/$\overline{\text{CE0}}$–$\overline{\text{CE3}}$	25–28	57–60	O	Output port (R30–R33) terminals or chip enable ($\overline{\text{CE0}}$–$\overline{\text{CE3}}$) output terminals
R34/FOUT	29	61	O	Output port (R34) terminal or clock (FOUT) output terminal
R35–R37	30–32	62–64	O	Output port (R35–R37) terminal
R50/BZ	41	79	O	Output port (R50) terminal or buzzer (BZ) output terminal
R51/$\overline{\text{BACK}}$	42	80	O	Output port (R51) terminal or bus acknowledge signal ($\overline{\text{BACK}}$) output terminal
P00–P07/D0–D7	58–51	96–89	I/O	I/O port (P00–P07) terminals or data bus (D0–D7)
P10/SIN	50	88	I/O	I/O port (P10) terminal or serial I/F data input (SIN) terminal
P11/SOUT	49	87	I/O	I/O port (P11) terminal or serial I/F data output (SOUT) terminal
P12/$\overline{\text{SCLK}}$	48	86	I/O	I/O port (P12) terminal or serial I/F clock ($\overline{\text{SCLK}}$) I/O terminal
P13/$\overline{\text{SRDY}}$	47	85	I/O	I/O port (P13) terminal or serial I/F ready signal ($\overline{\text{SRDY}}$) output terminal
P14/CMPP0	46	84	I/O	I/O port (P14) terminal or comparator 0 non-inverted input terminal
P15/CMPM0	45	83	I/O	I/O port (P15) terminal or comparator 0 inverted input terminal
P16/CMPP1	44	82	I/O	I/O port (P16) terminal or comparator 1 non-inverted input terminal
P17/CMPM1	43	81	I/O	I/O port (P17) terminal or comparator 1 inverted input terminal
$\overline{\text{RESET}}$	61	4	I	Initial reset input terminal
$\overline{\text{TEST}}$ *1	60	98	I	Test input terminal

*1 $\overline{\text{TEST}}$ is the terminal used for outgoing inspection of the IC. For normal operation be sure it is connected to VDD.

2 POWER SUPPLY

In this section, we will explain the operating voltage and the configuration of the internal power supply circuit of the E0C88112.

2.1 Operating Voltage

The E0C88112 operating power voltage is as follows:

Normal mode:	2.4 V to 5.5 V
Low power mode:	1.8 V to 3.5 V
High speed mode:	3.5 V to 5.5 V

If supply voltage drops below level 0 (see Chapter 7, "ELECTRICAL CHARACTERISTICS"), the system is automatically reset by a supply voltage detection (SVD) circuit described in the latter. This function can be selected by mask option.

2.2 Internal Power Supply Circuit

The E0C88112 incorporates the power supply circuit shown in Figure 2.2.1. When voltage within the range described above is supplied to V_{DD} (+) and V_{SS} (GND), the voltage needed for the internal circuit is generated internally in the IC.

The oscillation and internal circuits operate on the voltage V_{D1}, output by the oscillation system voltage regulator.
V_{D1} voltage can be selected from among three types: 1.3 V (low-power mode), 2.2 V (normal mode) and 3.3 V (high-speed mode).
It should be selected by a program to switch according to the supply voltage and oscillation frequency.
See Section 5.4, "Oscillation Circuits and Operating Mode", for the switching of operating mode.

Note: Under no circumstances should V_{D1} terminal output be used to drive external circuit.

2.3 Heavy Load Protection Mode

The E0C88112 has a heavy load protection function for stable operation even when the supply voltage fluctuates by driving a heavy load. The heavy load protection mode becomes valid when the peripheral circuits are in the following status:

(1) The OSC3 oscillation circuit is switched ON
(OSCC = "1" and not in SLEEP)

(2) The buzzer output is switched ON
(BZON = "1" or BZSHT = "1")

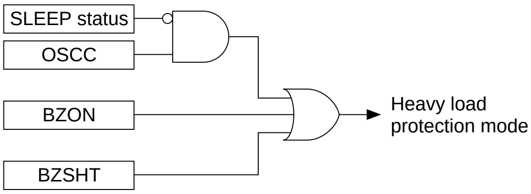


Fig. 2.3.1 Configuration of heavy load protection mode control circuit

For details of the OSC3 oscillation circuit and buzzer output, see "5.4 Oscillation Circuits and Operating Mode" and "5.12 Sound Generator", respectively.

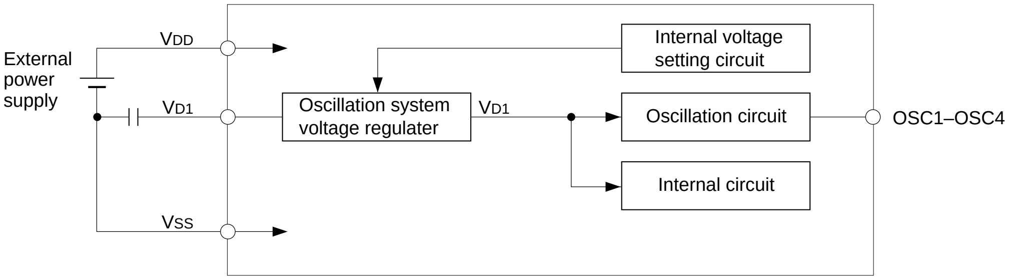


Fig. 2.2.1 Configuration of power supply circuit

3 CPU AND BUS CONFIGURATION

In this section, we will explain the CPU, operating mode and bus configuration.

3.1 CPU

The E0C88112 utilize the E0C88 8-bit core CPU whose resistor configuration, command set, etc. are virtually identical to other units in the family of processors incorporating the E0C88.

See the "E0C88 Core CPU Manual" for the E0C88.

Specifically, the E0C88112 employ the Model 3 E0C88 CPU which has a maximum address space of 512K bytes × 4.

3.2 Internal Memory

The E0C88112 is equipped with internal ROM and RAM as shown in Figure 3.2.1. Small scale applications can be handled by one chip. It is also possible to utilize internal memory in combination with external memory.
Furthermore, internal ROM can be disconnected from the bus and the resulting space released for external applications.

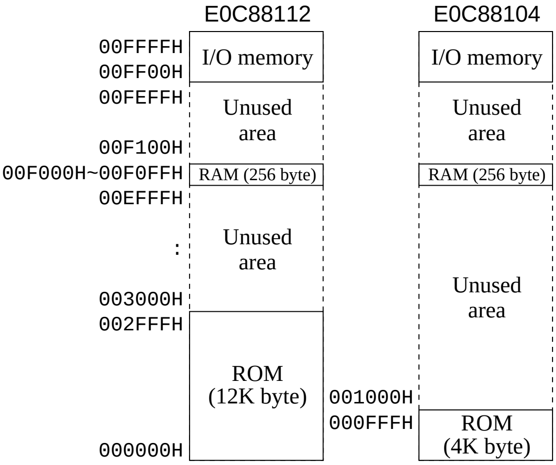


Fig. 3.2.1 Internal memory map

3.2.1 ROM

The internal ROM capacity is shown in Table 3.2.1.1.

Table 3.2.1.1 Internal ROM capacity

Model	ROM capacity	Address
E0C88112	12K bytes	000000H–002FFFH
E0C88104	4K bytes	000000H–000FFFH

The ROM area shown above can be released to external memory depending on the setting of the MCU/$\overline{\text{MPU}}$ terminal. (See "3.5 Chip Mode".)

3.2.2 RAM

The internal RAM capacity is shown in Table 3.2.2.1.

Table 3.2.2.1 Internal RAM capacity

Model	RAM capacity	Address
E0C88112	256 bytes	00F000H–00F0FFH
E0C88104	256 bytes	00F000H–00F0FFH

Even when external memory which overlaps the internal RAM area is expanded, the RAM area is not released to external memory. Access to this area is via internal RAM.

3.2.3 I/O memory

A memory mapped I/O method is employed in the E0C88112 for interfacing with internal peripheral circuit. Peripheral circuit control bits and data register are arranged in data memory space. Control and data exchange are conducted via normal memory access. I/O memory is arranged in page 0: 00FF00H–00FFFFH area.
See Section 5.1, "I/O Memory Map", for details of the I/O memory.
Even when external memory which overlaps the I/O memory area is expanded, the I/O memory area is not released to external memory. Access to this area is via I/O memory.

3.3 Exception Processing Vectors

000000H–000023H in the program area of the E0C88112 is assigned as exception processing vectors. Furthermore, from 000026H to 0000FFH, software interrupt vectors are assignable to any two bytes which begin with an even address.
Table 3.3.1 lists the vector addresses and the exception processing factors to which they correspond.

Table 3.3.1 Vector addresses and the corresponding exception processing factors

Vector address	Exception processing factor	Priority
000000H	Reset	High
000002H	Zero division	↑
000004H	Watchdog timer ($\overline{\text{NMI}}$)	
000006H	Programmable timer 1 interrupt	
000008H	Programmable timer 0 interrupt	
00000AH	K10, K11 input interrupt	
00000CH	K04–K07 input interrupt	
00000EH	K00–K03 input interrupt	
000010H	Serial I/F error interrupt	
000012H	Serial I/F receiving complete interrupt	
000014H	Serial I/F transmitting complete interrupt	
000016H	Stopwatch timer 100 Hz interrupt	
000018H	Stopwatch timer 10 Hz interrupt	
00001AH	Stopwatch timer 1 Hz interrupt	
00001CH	Clock timer 32 Hz interrupt	
00001EH	Clock timer 8 Hz interrupt	
000020H	Clock timer 2 Hz interrupt	↓
000022H	Clock timer 1 Hz interrupt	Low
000024H	System reserved (cannot be used)	No priority rating
000026H : 0000FEH	Software interrupt	

For each vector address and the address after it, the start address of the exception processing routine is written into the subordinate and super ordinate sequence. When an exception processing factor is generated, the exception processing routine is executed starting from the recorded address.

When multiple exception processing factors are generated at the same time, execution starts with the highest priority item.
The priority sequence shown in Table 3.3.1 assumes that the interrupt priority levels are all the same. The interrupt priority levels can be set by software in each system. (See Section 5.15 "Interrupt and Standby Status".)

Note: For exception processing other than reset, SC (system condition flag) and PC (program counter) are evacuated to the stack and branches to the exception processing routines. Consequently, when returning to the main routine from exception processing routines, please use the RETE instruction.

See the "E0C88 Core CPU Manual" for information on CPU operations when an exception processing factor is generated.

3.4 CC (Customized Condition Flag)

The E0C88112 does not use the customized condition flag (CC) in the core CPU. Accordingly, it cannot be used as a branching condition for the conditional branching instruction (JRS, CARS).

3.5 Chip Mode

3.5.1 MCU mode and MPU mode

The chip operating mode can be set to one of two settings using the MCU/$\overline{\text{MPU}}$ terminal.

■ **MCU mode...Set the MCU/$\overline{\text{MPU}}$ terminal to HIGH**

Switch to this setting when using internal ROM. With respect to areas other than internal memory, external memory can even be expanded. See Section 3.5.2, "Bus mode", for the memory map.

In the MCU mode, during initial reset, only systems in internal memory are activated. Internal ROM is normally fixed as the top portion of the program memory common area (logical space 0000H–7FFFH). Exception processing vectors are assigned in internal ROM. Furthermore, the application initialization routines that start with reset exception processing must likewise be written to internal ROM. Since bus and other settings which correlate with external expanded memory can be executed in software, this processing is executed in the initialization routine written to internal ROM. Once these bus mode settings are made, external memory can be accessed.

When accessing internal memory in this mode, the chip enable ($\overline{\text{CE}}$) and read ($\overline{\text{RD}}$)/write ($\overline{\text{WR}}$) signals are not output to external memory, and the data bus (D0–D7) changed to high impedance status (pull-up status when the "pull-up resistors for P00–P07 enabled" have been selected by the mask option).
Consequently, in cases where addresses overlap in external and internal memory, the areas in external memory will be unavailable.

■ **MPU mode...Set the MCU/$\overline{\text{MPU}}$ terminal to LOW**

Internal ROM area is released to an external device source. Internal ROM then becomes unusable and when this area is accessed, chip enable ($\overline{\text{CE}}$) and read ($\overline{\text{RD}}$)/write ($\overline{\text{WR}}$) signals are output to external memory and the data bus (D0–D7) become active. These signals are not output to an external source when other areas of internal memory are accessed.

In the MPU mode, the system is activated by external memory.
For this reason, in order to adjust bus settings to conform to the configuration of external memory during initial reset, the user can select the applicable system configuration using the mask option. (See "3.5.2 Bus mode")
When employing this mode, the exception processing vectors and initialization routine must be assigned within the common area (000000H–007FFFH).
You can select whether to use the built-in pull-up resistor of the MCU/$\overline{\text{MPU}}$ terminal by the mask option.

Note: The MCU/$\overline{\text{MPU}}$ terminal must be fixed at either the HIGH or LOW setting. Do not readjust this setting while the system is in operation.

3.5.2 Bus mode

In order to set bus specifications to match the configuration of external expanded memory, four different bus modes described below are selectable in software.

■ Single chip mode

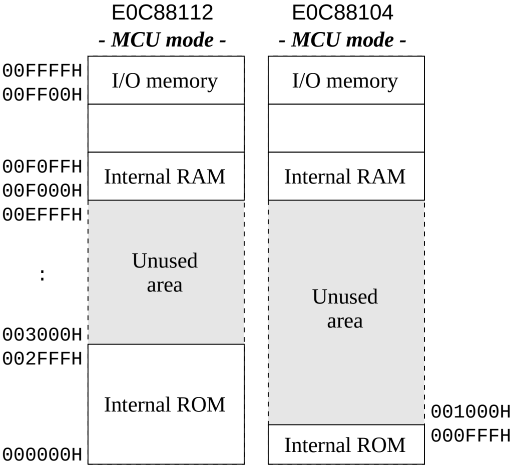


Fig. 3.5.2.1 Memory map for the single chip mode

The single chip mode setting applies when the E0C88112 is used as a single chip microcomputer without external expanded memory. Since this mode employs internal ROM, the system can only be operated in the MCU mode discussed in Section 3.5.1. In the MPU mode, the system cannot be set to the single chip mode.

Since there is no need for an external bus line in this mode, terminals normally set for bus use can be used as general purpose output ports or I/O ports.
Accordingly, the output ports are in a 34-bit configuration in the E0C88112. The I/O ports are in a 16-bit configuration.

CPU operation in this mode is equivalent to the E0C88 core CPU Model 3 minimum mode. Addresses assigned to internal memory within physical space 000000H to 00FFFFH are only effective as a target for accessing.

■ Expanded 64K mode

The expanded 64K mode setting applies when the E0C88112 is used with 64K bytes or less of external expanded memory. This mode is usable regardless of the MCU/MPU mode setting.

Because internal ROM is being used in the MCU mode, external memory in that area is not accessible.
External memory can be assigned to the area from 004000H to 00EFFFH in the E0C88112. Since the internal ROM area is released in the MPU mode, external memory in this model can be assigned to the area from 000000H to 00EFFFH. The area from 00F000H to 00FFFFH is assigned to internal memory (RAM, etc.) and cannot be used to access an external device.

This mode setting is suitable for small- to mid-scale systems. The address range of the chip enable ($\overline{CE}$) signal, adapted to memory chips with a capacity of from 8 to 64K bytes, can be selected in software to any one of four settings. See Section 3.6.4, "Chip enable ($\overline{CE}$) signal", for the $\overline{CE}$ signal.

CPU operation in this mode is equivalent to the E0C88 core CPU Model 3 minimum mode. The area within physical space 000000H to 00FFFFH is only effective as a target for accessing.

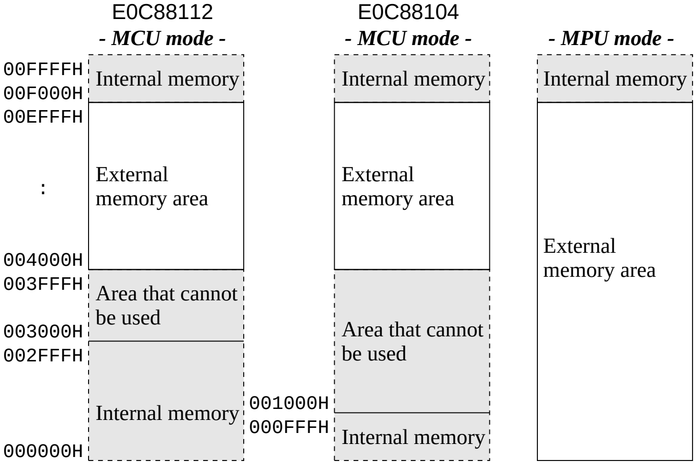


Fig. 3.5.2.2
Memory map for the expanded 64K mode

See Figure 3.2.1 for the internal memory

■ Expanded 512K minimum mode

The expanded 512K minimum mode setting applies when the E0C88112 is used with over 64K bytes and less than 512K bytes × 4 of external expanded memory. This mode is usable regardless of the MCU/MPU mode setting.

Because internal ROM is being used in the MCU mode, external memory can be assigned to the area from 080000H to 27FFFFH.
Since the internal ROM area is released in the MPU mode, external memory can be assigned to the area from 000000H to 1FFFFFH. However, the area from 00F000H to 00FFFFH is assigned to internal memory and cannot be used to access an external device.

CPU operation in this mode is equivalent to the E0C88 core CPU Model3 minimum mode. The area within physical space 000000H to 1FFFFFH in the MPU mode or physical space 080000H to 27FFFFH + internal memory in the MCU mode is effective as a target for accessing.
Furthermore, since program memory expansion is limited to less than 64K bytes configured with the common area (000000H to 007FFFH) and one optional bank area (internal ROM + 32K in the MCU mode), this mode is suitable for small- to mid-scale program memory and large-scale data memory systems.

The address range of chip enable ($\overline{CE}$) signals in this mode is fixed at 512K bytes.

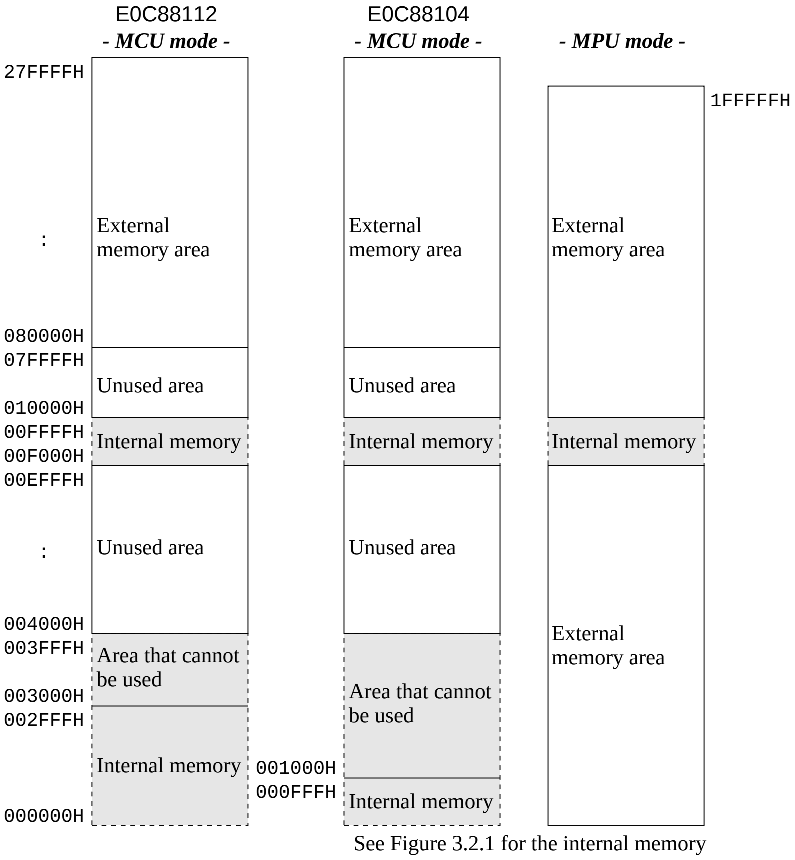


Fig. 3.5.2.3 Memory map for the expanded 512K minimum mode

■ Expanded 512K maximum mode

The expanded 512K maximum mode setting applies when the E0C88112 is used with over 64K bytes and less than 512K bytes × 4 of external expanded memory. This mode is usable regardless of the MCU/MPU mode setting.

Because internal ROM is being used in the MCU mode, external memory can be assigned to the area from 080000H to 27FFFFH.
Since the internal ROM area is released in the MPU mode, external memory can be assigned to the area from 000000H to 1FFFFFH.
The area from 00F000H to 00FFFFH is assigned to internal memory and cannot be used to access an external device.

CPU operation in this mode is equivalent to the E0C88 core CPU Model 3 maximum mode, the area within physical space 000000H to 1FFFFFH in the MPU mode or physical space 080000H to 27FFFFH + internal memory in the MCU mode is effective as a target for accessing. In the above mentioned physical space, since program memory and data memory can be secured with an optional (maximum 512K bytes × 4 program + data) size, this mode is suitable for systems with large-scale program and data capacity.

The address range of chip enable ($\overline{\text{CE}}$) signals in this mode is fixed at 512K bytes.

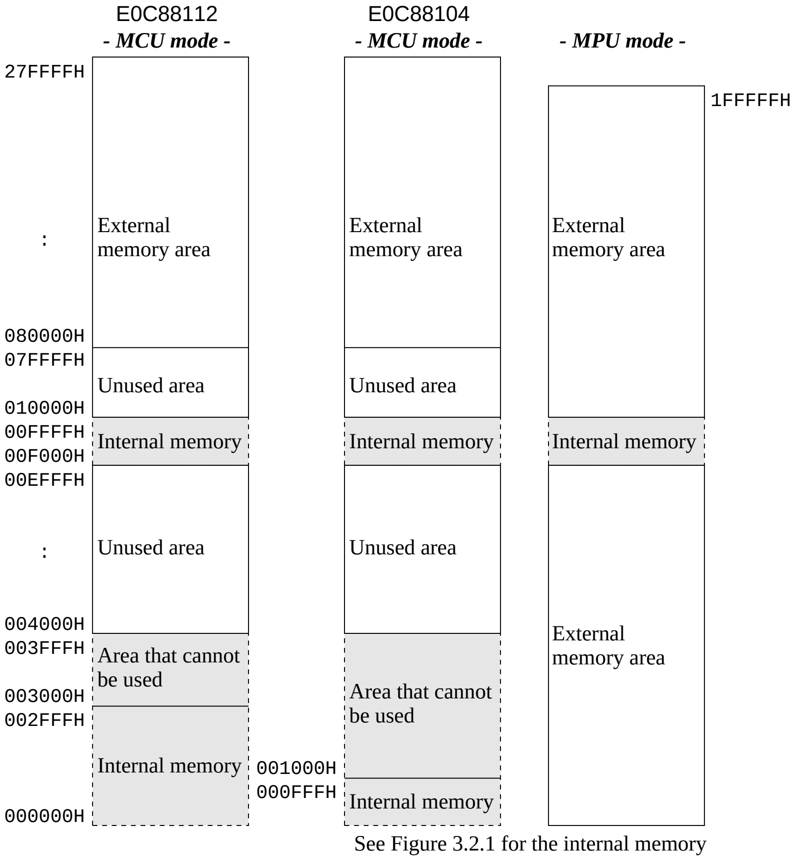


Fig. 3.5.2.4 Memory map for the expanded 512K maximum mode

There is an explanation on how all these settings are actually made in "5.2 System Controller and Bus Control" of this Manual.

3.6 External Bus

The E0C88112 has bus terminals that can address a maximum of 512K × 4 bytes and memory (and other) devices can be externally expanded according to the range of each bus mode described in the previous section.

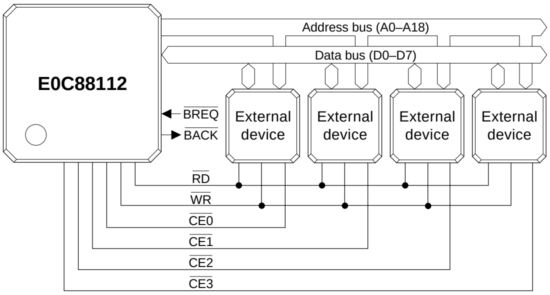
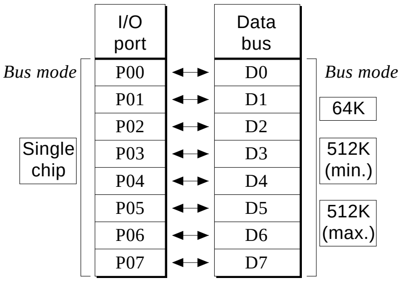


Fig. 3.6.1 External bus lines

Below is an explanation of external bus terminals. For information on control methods, see Section 5.2, "System Controller and Bus Control".

3.6.1 Data bus

The E0C88112 possess an 8-bit external data bus (D0–D7). The terminals and I/O circuits of data bus D0–D7 are shared with I/O ports P00–P07, switching between these functions being determined by the bus mode setting.
In the single chip mode, the 8-bit terminals are all set as I/O ports P00–P07 and in the other expanded modes, they are set as data bus (D0–D7).
When set as data bus, the data register and I/O control register of each I/O port are detached from the I/O circuits and usable as a general purpose data register with read/write capabilities.



I/O port		Data bus
P00	◄─►	D0
P01	◄─►	D1
P02	◄─►	D2
P03	◄─►	D3
P04	◄─►	D4
P05	◄─►	D5
P06	◄─►	D6
P07	◄─►	D7

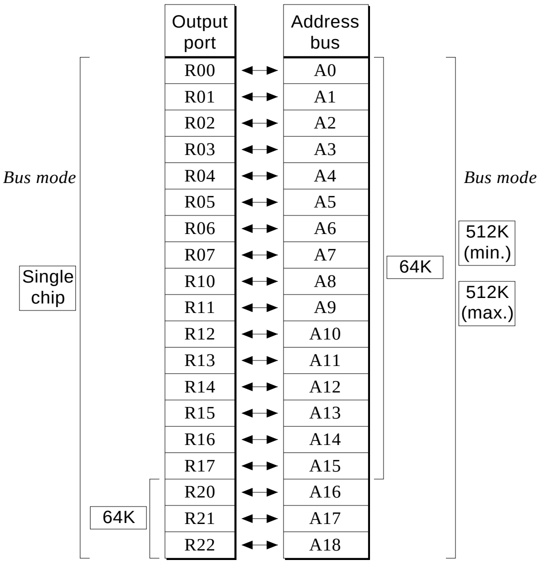
Fig. 3.6.1.1 Correspondence between data bus and I/O ports

With regard to the pull-up resistors that go ON only in input mode, the mask option can be used to select whether or not to use the pull-up resistor for each data bus line. (The same holds true when the terminals are used as I/O ports.)

3.6.2 Address bus

The E0C88112 possess a 19-bit external address bus A0–A18. The terminals and output circuits of address bus A0–A18 are shared with output ports R00–R07 (=A0–A7), R10–R17 (=A8–A15) and R20–R22 (=A16–A18), switching between these functions being determined by the bus mode setting.
In the single chip mode, the 19-bit terminals are all set as output ports R00–R07, R10–R17 and R20–R22.
In the expanded 64K mode, 16 of the 19-bit terminals, A0–A15, are set as the address bus, while the remaining 3 bits, A16–A18, are set as output ports R20–R22.
In the expanded 512K minimum and maximum modes, all of the 19-bit terminals are set as the address bus (A0–A18).

When set as an address bus, the data register and high impedance control register of each output port are detached from the output circuit and used as a general purpose data register with read/write capabilities.



Output port		Address bus
R00	◄─►	A0
R01	◄─►	A1
R02	◄─►	A2
R03	◄─►	A3
R04	◄─►	A4
R05	◄─►	A5
R06	◄─►	A6
R07	◄─►	A7
R10	◄─►	A8
R11	◄─►	A9
R12	◄─►	A10
R13	◄─►	A11
R14	◄─►	A12
R15	◄─►	A13
R16	◄─►	A14
R17	◄─►	A15
R20	◄─►	A16
R21	◄─►	A17
R22	◄─►	A18

Fig. 3.6.2.1 Correspondence between address bus and output ports

3.6.3 Read ($\overline{\text{RD}}$)/write ($\overline{\text{WR}}$) signals

The output terminals and output circuits for the read ($\overline{\text{RD}}$)/write ($\overline{\text{WR}}$) signals directed to external devices are shared respectively with output ports R23 and R24, switching between these functions being determined by the bus mode setting.
In the single chip mode, both of these terminals are set as output port terminals and in the other expanded modes, they are set as read ($\overline{\text{RD}}$)/write ($\overline{\text{WR}}$) signal output terminals. When set as read ($\overline{\text{RD}}$)/write ($\overline{\text{WR}}$) signal output terminal, the data register and high impedance control register for each output port (R23, R24) are detached from the output circuit and is usable as a general purpose data register with read/write capabilities.

These two signals are only output when the memory area of the external device is being accessed. They are not output when internal memory is accessed.
See Section 3.6.5, "WAIT control", for the output timing of the signal.

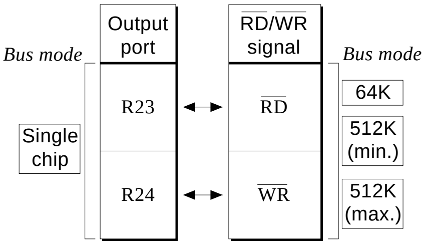


Fig. 3.6.3.1 Correspondence between read ($\overline{\text{RD}}$)/write ($\overline{\text{WR}}$) signal and output ports

3.6.4 Chip enable ($\overline{\text{CE}}$) signal

The E0C88112 is equipped with address decoders which can output four different chip enable ($\overline{\text{CE}}$) signals.
Consequently, four devices equipped with a chip enable ($\overline{\text{CE}}$) or chip select ($\overline{\text{CS}}$) terminal can be directly connected without setting the address decoder to an external device.

The four chip enable ($\overline{\text{CE0}}$–$\overline{\text{CE3}}$) signal output terminals and output circuits are shared with output ports R30–R33 and in modes other than the single chip mode, the selection of chip enable ($\overline{\text{CE}}$) or output port can be set in software for each of the four bits. When set for chip enable ($\overline{\text{CE}}$) output, the data register and high impedance control register for each output port are detached from the output circuit and is usable as general purpose data register with read/write capabilities.
In the single chip mode, these terminals are set as output ports R30–R33.

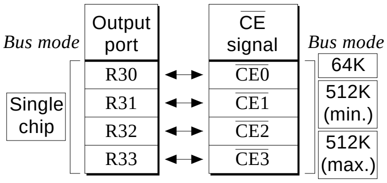


Fig. 3.6.4.1 Correspondence between $\overline{\text{CE}}$ signals and output ports

The address range assigned to the four chip enable ($\overline{\text{CE}}$) signals is determined by the bus mode setting. In the expanded 64K mode, the four different address ranges which match the amount of memory in use can be selected in software.
Table 3.6.4.1 shows the address ranges which are assigned to the chip enable ($\overline{\text{CE}}$) signal in each mode. When accessing the internal memory area, the $\overline{\text{CE}}$ signal is not output. Care should be taken here because the address range for these portions of memory involves irregular settings.
The arrangement of memory space for external devices does not necessarily have to be continuous from a subordinate address and any of the chip enable signals can be used to assign areas in memory.

Each of these signals is only output when the memory area of the external device is being accessed. They are not output when internal memory is accessed.

See Section 3.6.5, "WAIT control", for the output timing of signal.

Table 3.6.4.1 $\overline{CE0}$–$\overline{CE3}$ address settings

(1) Expanded 64K mode + MCU mode (E0C88112/88104)

$\overline{CE}$ signal	Address range (selected in software)			
	8K bytes	16K bytes	32K bytes	64K bytes
$\overline{CE0}$	008000H–009FFFH	–	004000H–007FFFH	004000H–00EFFFH
$\overline{CE1}$	00A000H–00BFFFH	004000H–007FFFH	008000H–00EFFFH	–
$\overline{CE2}$	004000H–005FFFH	008000H–00BFFFH	–	–
$\overline{CE3}$	006000H–007FFFH	00C000H–00EFFFH	–	–

(2) Expanded 64K mode + MPU mode (E0C88112/88104)

$\overline{CE}$ signal	Address range (selected in software)			
	8K bytes	16K bytes	32K bytes	64K bytes
$\overline{CE0}$	000000H–001FFFH	000000H–003FFFH	000000H–007FFFH	000000H–00EFFFH
$\overline{CE1}$	002000H–003FFFH	004000H–007FFFH	008000H–00EFFFH	–
$\overline{CE2}$	004000H–005FFFH	008000H–00BFFFH	–	–
$\overline{CE3}$	006000H–007FFFH	00C000H–00EFFFH	–	–

(3) Expanded 512K minimum/maximum modes (E0C88112/88104)

$\overline{CE}$ signal	Address range	
	MCU mode	MPU mode
$\overline{CE0}$	200000H–27FFFFH	000000H–00EFFFH, 010000H–07FFFFH
$\overline{CE1}$	080000H–0FFFFFH	080000H–0FFFFFH
$\overline{CE2}$	100000H–17FFFFH	100000H–17FFFFH
$\overline{CE3}$	180000H–1FFFFFH	180000H–1FFFFFH

3.6.5 WAIT control

In order to insure accessing of external low speed devices during high speed operations, the E0C88112 is equipped with a WAIT function which prolongs access time. (See the "E0C88 Core CPU Manual" for details of the WAIT function.)

The WAIT state numbers to be inserted can be selected in software from a series of 8 as shown in Table 3.6.5.1.

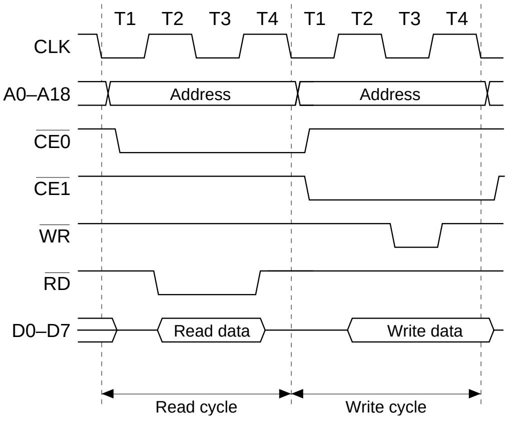
Table 3.6.5.1 Selectable WAIT state numbers

Selection No.	1	2	3	4	5	6	7	8
Insert states	0	2	4	6	8	10	12	14

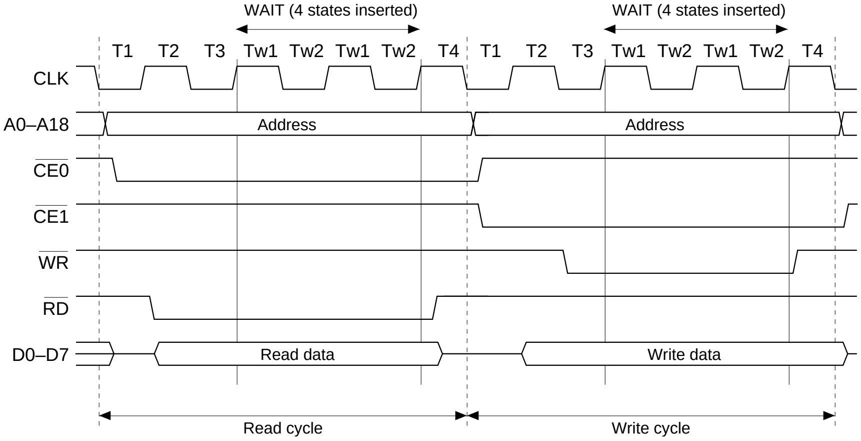
* One state is a 1/2 cycle of the clock in length.

The WAIT states set in software are inserted between bus cycle states T3–T4.
Note, however, that WAIT states cannot be inserted when an internal register and internal memory are being accessed and when operating with the OSC1 oscillation circuit (see "5.4 Oscillation Circuits and Operating Mode").
Consequently, WAIT state settings are meaningless in the single chip mode.

Figure 3.6.5.1 shows the memory read/write timing charts.



(1) No WAIT



(2) WAIT state insertion

Fig. 3.6.5.1 Memory read/write cycle

3.6.6 Bus authority release state

The E0C88112 is equipped with a bus authority release function on request from an external device so that DMA (Direct Memory Access) transfer can be conducted between external devices. The internal memory cannot be accessed by this function.

There are two terminals used for this function: the bus authority release request signal (BREQ) input terminal and the bus authority release acknowledge signal (BACK) output terminal.
The BREQ input terminal is shared with input port terminal K11 and the BACK output terminal with output port terminal R51, use with setting to BREQ/BACK terminals done in software. In the single chip mode, or when using a system which does not require bus authority release, set respective terminals as input and output ports.

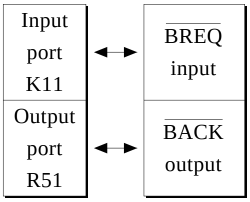


Fig. 3.6.6.1 BREQ/BACK terminals

When the bus authority release request (BREQ = LOW) is received from an external device, the E0C88112 switchs the address bus, data bus, RD/WR signal, and CE signal lines to a high impedance state, output a LOW level from the BACK terminal and release bus authority.
As soon as a LOW level is output from the BACK terminal, the external device can use the external bus. When DMA is completed, the external device returns the BREQ terminal to HIGH and releases bus authority.
Figure 3.6.6.2 shows the bus authority release sequence.

During bus authority release state, internal memory cannot be accessed from the external device. In cases where external memory has areas which overlap areas in internal memory, the external memory areas can be accessed accordance with the CE signal output by the external device.

Note: Be careful with the system, such that an external device does not become the bus master, other than during the bus release status.
After setting the BREQ terminal to LOW level, hold the BREQ terminal at LOW level until the BACK terminal becomes LOW level. If the BREQ terminal is returned to HIGH level, before the BACK terminal becomes LOW level, the shift to the bus authorization release status will become indefinite.

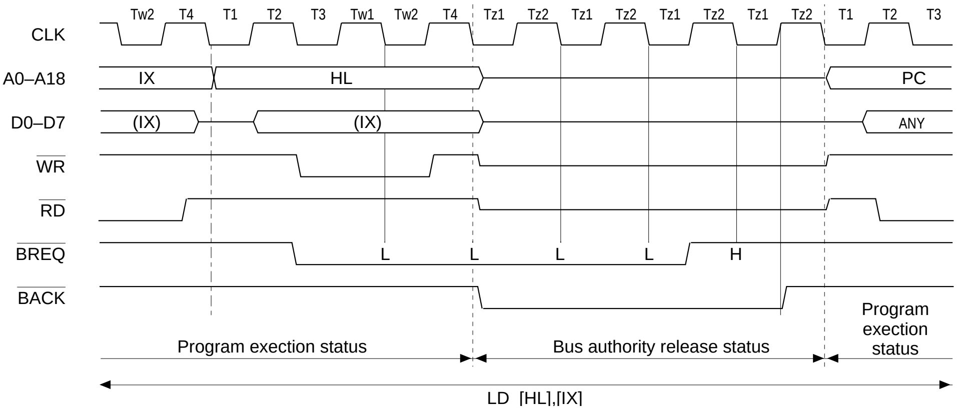


Fig. 3.6.6.2 Bus authority release sequence

4 INITIAL RESET

Initial reset in the E0C88112 is required in order to initialize circuits. This section of the Manual contains a description of initial reset factors and the initial settings for internal registers, etc.

4.1 Initial Reset Factors

There are three initial reset factors for the E0C88112 as shown below.

(1) $\overline{\text{RESET}}$ terminal
(2) Simultaneous LOW level input at input port terminals K00–K03.
(3) Supply voltage detection (SVD) circuit

Figure 4.1.1 shows the configuration of the initial reset circuit.

The CPU and peripheral circuits are initialized by means of initial reset factors. When the factor is canceled, the CPU commences reset exception processing. (See "E0C88 Core CPU Manual".)

When this occurs, reset exception processing vectors, Bank 0, 000000H–000001H from program memory are read out and the program (initialization routine) which begins at the readout address is executed.

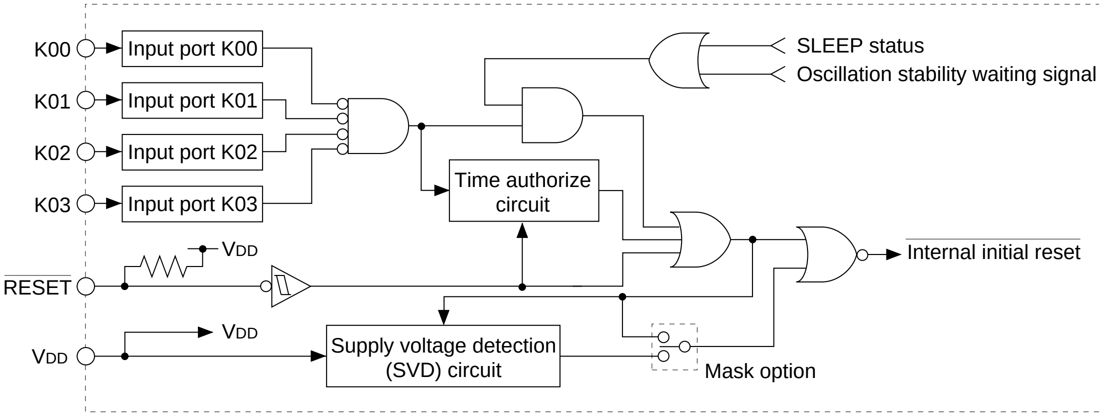


Fig. 4.1.1 Configuration of initial reset circuit

4.1.1 RESET terminal

Initial reset can be done by executed externally inputting a LOW level to the RESET terminal.
Be sure to maintain the RESET terminal at LOW level for the regulation time after the power on to assure the initial reset.
In addition, be sure to use the RESET terminal for the first initial reset after the power is turned on.
The RESET terminal is equipped with a pull-up resistor. You can select whether or not to use by mask option.

4.1.2 Simultaneous LOW level input at input port terminals K00–K03

Another way of executing initial reset externally is to input a LOW level simultaneously to the input ports (K00–K03) selected by mask option.
Since there is a built-in time authorize circuit, be sure to maintain the designated input port terminal at LOW level for two seconds (when the oscillation frequency is f_{OSC1} = 32.768 kHz) or more to perform the initial reset by means of this function.
However, the time authorize circuit is bypassed during the SLEEP (standby) status and oscillation stabilization waiting period, and initial reset is executed immediately after the simultaneous LOW level input to the designated input ports.
The combination of input ports (K00–K03) that can be selected by mask option are as follows:
(1) Not use
(2) K00 & K01
(3) K00 & K01 & K02
(4) K00 & K01 & K02 & K03

For instance, let's say that mask option (4) "K00 & K01 & K02 & K03" is selected.
When the input level at input ports K00–K03 is simultaneously LOW, initial reset will take place.

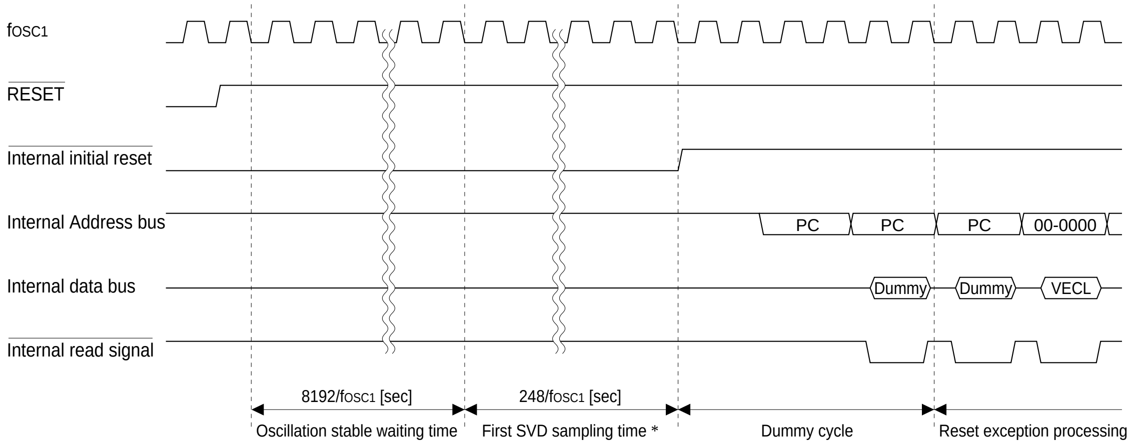
When using this function, make sure that the designated input ports do not simultaneously switch to LOW level while the system is in normal operation.

4.1.3 Supply voltage detection (SVD) circuit

When the SVD circuit detects that supply voltage has dropped below level 0 four successive times (see Chapter 7, "ELECTRICAL CHARACTERISTICS"), it outputs an initial reset signal until the supply voltage has been restored to level 2.
You can select whether or not to use the initial reset according to the SVD circuit by mask option. If you use it, the supply voltage must be at least level 2 for the first sampling of the SVD circuit, when the power is turned on. At this time, if the power voltage level is less than level 2, the initial reset status will not be canceled and instead the SVD circuit will continue sampling until the supply voltage reaches level 2 or more.
For more information, see "5.14 Supply Voltage Detection (SVD) Circuit" in this Manual.

4.1.4 Initial reset sequence

After cancellation of the LOW level input to the RESET terminal, when the power is turned on, the start-up of the CPU is held back until the oscillation stabilization waiting time (8,192/f_{OSC1} sec.) has elapsed. When the initial reset by the SVD circuit has been used, an initial sampling time (248/f_{OSC1} sec.) is added as additional waiting time.
Figure 4.1.4.1 shows the operating sequence following initial reset release.



* When the initial reset by the SVD circuit with the mask option has been used, this cycle is inserted as the waiting time.

Fig. 4.1.4.1 Initial reset sequence

Also, when using the initial reset by simultaneous LOW level input into the input port, you should be careful of the following points.

(1) During SLEEP status, since the time authorization circuit is bypassed, an initial reset is triggered immediately after a LOW level simultaneous input value. In this case, the CPU starts after waiting the oscillation stabilization time and the SVD circuit initial sampling time (when used with the mask option), following cancellation of the LOW level simultaneous input.

(2) Other than during SLEEP status, an initial reset will be triggered 1–2 seconds after a LOW level simultaneous input. In this case, since a reset differential pulse (64/fOSC1 sec.) is generated within the E0C88112, the CPU will start even if the LOW level simultaneous input status is not canceled.

4.2 Initial Settings After Initial Reset

The CPU internal registers are initialized as follows during initial reset.

Table 4.2.1 Initial settings

Register name	Code	Bit length	Setting value
Data register A	A	8	Undefined
Data register B	B	8	Undefined
Index (data) register L	L	8	Undefined
Index (data) register H	H	8	Undefined
Index register IX	IX	16	Undefined
Index register IY	IY	16	Undefined
Program counter	PC	16	Undefined*
Stack pointer	SP	16	Undefined
Base register	BR	8	Undefined
Zero flag	Z	1	0
Carry flag	C	1	0
Overflow flag	V	1	0
Negative flag	N	1	0
Decimal flag	D	1	0
Unpack flag	U	1	0
Interrupt flag 0	I0	1	1
Interrupt flag 1	I1	1	1
New code bank register	NB	8	01H
Code bank register	CB	8	Undefined*
Expand page register	EP	8	00H
Expand page register for IX	XP	8	00H
Expand page register for IY	YP	8	00H

* Reset exception processing loads the preset values stored in 0 bank, 0000H–0001H into the PC. At the same time, 01H of the NB initial value is loaded into CB.

Initialize the registers which are not initialized at initial reset using software.

Since the internal RAM and display memory are not initialized at initial reset, be sure to initialize using software.

The respectively stipulated initializations are done for internal peripheral circuits. If necessary, the initialization should be done using software.
For initial value at initial reset, see the sections on the I/O memory map and peripheral circuit descriptions in the following chapter of this Manual.

5 PERIPHERAL CIRCUITS AND THEIR OPERATION

The peripheral circuits of the E0C88112 is interfaced with the CPU by means of the memory mapped I/O method. For this reason, just as with other memory access operations, peripheral circuits can be controlled by manipulating I/O memory. Below is a description of the operation and control method for each individual peripheral circuit.

5.1 I/O Memory Map

Table 5.1.1(a) I/O Memory map (00FF00H–00FF02H, MCU mode)

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF00 (MCU)	D7	BSMD1	Bus mode (CPU mode) BSMD1 BSMD0 Mode 1 1 512K (Maximum) 1 0 512K (Minimum) 0 1 64K 0 0 Single chip			0	R/W	
	D6	BSMD0				0	R/W	
	D5	CEMD1	Chip enable mode CEMD1 CEMD0 Mode 1 1 64K ($\overline{CE0}$) 1 0 32K ($\overline{CE0}$, $\overline{CE1}$) 0 1 16K ($\overline{CE1}$–$\overline{CE3}$) 0 0 8K ($\overline{CE0}$–$\overline{CE3}$)			1	R/W	Only for 64K bus mode
	D4	CEMD0				1	R/W	
	D3	CE3	$\overline{CE3}$ (R33) $\overline{CE}$ signal output Enable/Disable Enable: $\overline{CE}$ signal output Disable: DC (R3x) output	$\overline{CE3}$ enable	$\overline{CE3}$ disable	0	R/W	In the Single chip mode, these setting are fixed at DC output.
	D2	CE2	$\overline{CE2}$ (R32)	$\overline{CE2}$ enable	$\overline{CE2}$ disable	0	R/W	
	D1	CE1	$\overline{CE1}$ (R31)	$\overline{CE1}$ enable	$\overline{CE1}$ disable	0	R/W	
	D0	CE0	$\overline{CE0}$ (R30)	$\overline{CE0}$ enable	$\overline{CE0}$ disable	0	R/W	
00FF01	D7	SPP7	Stack pointer page address (MSB)	1	0	0	R/W	
	D6	SPP6		1	0	0	R/W	
	D5	SPP5	< SP page allocatable address >	1	0	0	R/W	
	D4	SPP4	• Single chip mode: only 0 page	1	0	0	R/W	
	D3	SPP3	• 64K mode: only 0 page	1	0	0	R/W	
	D2	SPP2	• 512K (min) mode: 0–27H page	1	0	0	R/W	
	D1	SPP1	• 512K (max) mode: 0–27H page	1	0	0	R/W	
	D0	SPP0	(LSB)	1	0	0	R/W	
00FF02	D7	EBR	Bus release enable register K11 (K11 and R51 terminal specification) R51	$\overline{BREQ}$ $\overline{BACK}$	Input port Output port	0	R/W	
	D6	WT2	Wait control register WT2 WT1 WT0 Number of state 1 1 1 14 1 1 0 12 1 0 1 10 1 0 0 8 0 1 1 6 0 1 0 4 0 0 1 2 0 0 0 No wait			0	R/W	
	D5	WT1						
	D4	WT0						
	D3	CLKCHG	CPU operating clock switch	OSC3	OSC1	0	R/W	
	D2	OSCC	OSC3 oscillation On/Off control	On	Off	0	R/W	
	D1	VDC1	Operating mode selection VDC1 VDC0 Operating mode 1 × High speed (V_{D1}=3.3V) 0 1 Low power (V_{D1}=1.3V) 0 0 Normal (V_{D1}=2.2V)			0	R/W	
	D0	VDC0						

Note: All the interrupts including $\overline{NMI}$ are disabled, until you write the optional value into both the "00FF00H" and "00FF01H" addresses.

Table 5.1.1(b) I/O Memory map (00FF00H–00FF02H, MPU mode)

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF00 (MPU)	D7	BSMD1	Bus mode (CPU mode) BSMD1 BSMD0 Mode 1 1 512K (Maximum) 1 0 512K (Minimum) 0 1 64K 0 0 * Option selection ◀			*	R/W	* Initial setting can be selected among 3 types (64K, 512K min and 512K max) by mask option setting.
	D6	BSMD0				*	R/W	
	D5	CEMD1	Chip enable mode CEMD1 CEMD0 Mode 1 1 64K ($\overline{CE0}$) 1 0 32K ($\overline{CE0}$, $\overline{CE1}$) 0 1 16K ($\overline{CE0}$–$\overline{CE3}$) 0 0 8K ($\overline{CE0}$–$\overline{CE3}$)			1	R/W	Only for 64K bus mode
	D4	CEMD0				1	R/W	
	D3	CE3	$\overline{CE3}$ (R33) $\overline{CE}$ signal output Enable/Disable Enable: $\overline{CE}$ signal output Disable: DC (R3x) output	$\overline{CE3}$ enable	$\overline{CE3}$ disable	0	R/W	
	D2	CE2	$\overline{CE2}$ (R32)	$\overline{CE2}$ enable	$\overline{CE2}$ disable	0	R/W	
	D1	CE1	$\overline{CE1}$ (R31)	$\overline{CE1}$ enable	$\overline{CE1}$ disable	0	R/W	
	D0	CE0	$\overline{CE0}$ (R30)	$\overline{CE0}$ enable	$\overline{CE0}$ disable	1	R/W	
00FF01	D7	SPP7	Stack pointer page address (MSB)	1	0	0	R/W	
	D6	SPP6		1	0	0	R/W	
	D5	SPP5	< SP page allocatable address >	1	0	0	R/W	
	D4	SPP4	• Single chip mode: only 0 page	1	0	0	R/W	
	D3	SPP3	• 64K mode: only 0 page	1	0	0	R/W	
	D2	SPP2	• 512K (min) mode: 0–27H page	1	0	0	R/W	
	D1	SPP1	• 512K (max) mode: 0–27H page	1	0	0	R/W	
	D0	SPP0	(LSB)	1	0	0	R/W	
00FF02	D7	EBR	Bus release enable register K11 (K11 and R51 terminal specification) R51	$\overline{BREQ}$ $\overline{BACK}$	Input port Output port	0	R/W	
	D6	WT2	Wait control register WT2 WT1 WT0 Number of state 1 1 1 14 1 1 0 12 1 0 1 10 1 0 0 8 0 1 1 6 0 1 0 4 0 0 1 2 0 0 0 No wait			0	R/W	
	D5	WT1						
	D4	WT0						
	D3	CLKCHG	CPU operating clock switch	OSC3	OSC1	0	R/W	
	D2	OSCC	OSC3 oscillation On/Off control	On	Off	0	R/W	
	D1	VDC1	Operating mode selection VDC1 VDC0 Operating mode 1 × High speed (V_{D1}=3.3V) 0 1 Low power (V_{D1}=1.3V) 0 0 Normal (V_{D1}=2.2V)			0	R/W	
	D0	VDC0						

Note: All the interrupts including $\overline{NMI}$ are disabled, until you write the optional value into both the "00FF00H" and "00FF01H" addresses.

Table 5.1.1(c) I/O Memory map (00FF12H–00FF13H)

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF12	D7	–	–	–	–	–		Constantry "0" when being read
	D6	–	–	–	–	–		
	D5	SVDSP	SVD auto-sampling control	On	Off	0	R/W	These registers are reset to "0" when SLP instruction is executed.
	D4	SVDON	SVD continuous sampling control/status R W	Busy On	Ready Off	1→0*1 0	R/W	
	D3	SVD3	SVD detection level SVD3 SVD2 SVD1 SVD0 Detection level 1 1 1 1 Level 15 1 1 1 0 Level 14 : : : : : 0 0 0 0 Level 0			X	R	*2
	D2	SVD2				X	R	
	D1	SVD1				X	R	
	D0	SVD0				X	R	
00FF13	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	CMP1ON	Comparator 1 On/Off control	On	Off	0	R/W	
	D2	CMP0ON	Comparator 0 On/Off control	On	Off	0	R/W	
	D1	CMP1DT	Comparator 1 data	+ > -	+ < -	0	R	
	D0	CMP0DT	Comparator 0 data	+ > -	+ < -	0	R	

*1 After initial reset, this status is set "1" until conclusion of hardware first sampling.

*2 Initial values are set according to the supply voltage detected at first sampling by hardware.
Until conclusion of first sampling, SVD0–SVD3 data are undefined.

Table 5.1.1(d) I/O Memory map (00FF20H–00FF25H)

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF20	D7	PK01	K00–K07 interrupt priority register	PK01 PK00 PSIF1 PSIF0 PSW1 PSW0 Priority PTM1 PTM0 level 1 1 Level 3 1 0 Level 2 0 1 Level 1 0 0 Level 0		0	R/W	
	D6	PK00						
	D5	PSIF1	Serial interface interrupt priority register			0	R/W	
	D4	PSIF0						
	D3	PSW1	Stopwatch timer interrupt priority register			0	R/W	
	D2	PSW0						
	D1	PTM1	Clock timer interrupt priority register			0	R/W	
	D0	PTM0						
00FF21	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	PPT1	Programmable timer interrupt priority register	PPT1 PPT0 Priority PK11 PK10 level 1 1 Level 3 1 0 Level 2 0 1 Level 1 0 0 Level 0		0	R/W	
	D2	PPT0						
	D1	PK11	K10 and K11 interrupt priority register			0	R/W	
	D0	PK10						
00FF22	D7	–	–	–	–	–		"0" when being read
	D6	ESW100	Stopwatch timer 100 Hz interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D5	ESW10	Stopwatch timer 10 Hz interrupt enable register					
	D4	ESW1	Stopwatch timer 1 Hz interrupt enable register					
	D3	ETM32	Clock timer 32 Hz interrupt enable register					
	D2	ETM8	Clock timer 8 Hz interrupt enable register					
	D1	ETM2	Clock timer 2 Hz interrupt enable register					
	D0	ETM1	Clock timer 1 Hz interrupt enable register					
00FF23	D7	EPT1	Programmable timer 1 interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D6	EPT0	Programmable timer 0 interrupt enable register					
	D5	EK1	K10 and K11 interrupt enable register					
	D4	EK0H	K04–K07 interrupt enable register					
	D3	EK0L	K00–K03 interrupt enable register					
	D2	ESERR	Serial I/F (error) interrupt enable register					
	D1	ESREC	Serial I/F (receiving) interrupt enable register					
	D0	ESTRA	Serial I/F (transmitting) interrupt enable register					
00FF24	D7	–	–	–	–	–		"0" when being read
	D6	FSW100	Stopwatch timer 100 Hz interrupt factor flag	(R) Interrupt factor is generated	(R) No interrupt factor is generated	0	R/W	
	D5	FSW10	Stopwatch timer 10 Hz interrupt factor flag					
	D4	FSW1	Stopwatch timer 1 Hz interrupt factor flag					
	D3	FTM32	Clock timer 32 Hz interrupt factor flag					
	D2	FTM8	Clock timer 8 Hz interrupt factor flag	(W) Reset	(W) No operation			
	D1	FTM2	Clock timer 2 Hz interrupt factor flag					
	D0	FTM1	Clock timer 1 Hz interrupt factor flag					
00FF25	D7	FPT1	Programmable timer 1 interrupt factor flag	(R) Interrupt factor is generated	(R) No interrupt factor is generated	0	R/W	
	D6	FPT0	Programmable timer 0 interrupt factor flag					
	D5	FK1	K10 and K11 interrupt factor flag					
	D4	FK0H	K04–K07 interrupt factor flag					
	D3	FK0L	K00–K03 interrupt factor flag					
	D2	FSERR	Serial I/F (error) interrupt factor flag	(W) Reset	(W) No operation			
	D1	FSREC	Serial I/F (receiving) interrupt factor flag					
	D0	FSTRA	Serial I/F (transmitting) interrupt factor flag					

Table 5.1.1(e) I/O Memory map (00FF30H–00FF33H)

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF30	D7	–	–	–	–	–		Constantry "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	MODE16	8/16-bit mode selection	16-bit x 1	8-bit x 2	0	R/W	
	D3	CHSEL	TOUT output channel selection	Timer 1	Timer 0	0	R/W	
	D2	PTOUT	TOUT output control	On	Off	0	R/W	
	D1	CKSEL1	Prescaler 1 source clock selection	f_{OSC3}	f_{OSC1}	0	R/W	
	D0	CKSEL0	Prescaler 0 source clock selection	f_{OSC3}	f_{OSC1}	0	R/W	
00FF31	D7	EVCNT	Timer 0 counter mode selection	Event counter	Timer	0	R/W	
	D6	FCSEL	Timer 0 function selection: In timer mode In event counter mode	Pulse width measurement With noise rejector	Normal mode Without noise rejector	0	R/W	
	D5	PLPOL	Timer 0 pulse polarity selection: Down count timing in event counter mode In pulse width measurement mode	Rising edge of K10 input High level measurement for K10 input	Falling edge of K10 input Low level measurement for K10 input	0	R/W	
	D4	PSC01	Timer 0 prescaler dividing ratio selection PSC01 PSC00 Prescaler dividing ratio 1 1 Source clock / 64 1 0 Source clock / 16 0 1 Source clock / 4 0 0 Source clock / 1			0	R/W	
	D3	PSC00				0	R/W	
	D2	CONT0	Timer 0 continuous/one-shot mode selection	Continuous	One-shot	0	R/W	
	D1	PSET0	Timer 0 preset	Preset	No operation	–	W	"0" when being read
	D0	PRUN0	Timer 0 Run/Stop control	Run	Stop	0	R/W	
00FF32	D7	–	–	–	–	–		Constantry "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	PSC11	Timer 1 prescaler dividing ratio selection PSC11 PSC10 Prescaler dividing ratio 1 1 Source clock / 64 1 0 Source clock / 16 0 1 Source clock / 4 0 0 Source clock / 1			0	R/W	
	D3	PSC10				0	R/W	
	D2	CONT1	Timer 1 continuous/one-shot mode selection	Continuous	One-shot	0	R/W	
	D1	PSET1	Timer 1 preset	Preset	No operation	–	W	"0" when being read
	D0	PRUN1	Timer 1 Run/Stop control	Run	Stop	0	R/W	
00FF33	D7	RLD07	Timer 0 reload data D7 (MSB)	High	Low	1	R/W	
	D6	RLD06	Timer 0 reload data D6					
	D5	RLD05	Timer 0 reload data D5					
	D4	RLD04	Timer 0 reload data D4					
	D3	RLD03	Timer 0 reload data D3					
	D2	RLD02	Timer 0 reload data D2					
	D1	RLD01	Timer 0 reload data D1					
	D0	RLD00	Timer 0 reload data D0 (LSB)					

Table 5.1.1(f) I/O Memory map (00FF34H–00FF36H)

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF34	D7	RLD17	Timer 1 reload data D7 (MSB)	High	Low	1	R/W	
	D6	RLD16	Timer 1 reload data D6					
	D5	RLD15	Timer 1 reload data D5					
	D4	RLD14	Timer 1 reload data D4					
	D3	RLD13	Timer 1 reload data D3					
	D2	RLD12	Timer 1 reload data D2					
	D1	RLD11	Timer 1 reload data D1					
	D0	RLD10	Timer 1 reload data D0 (LSB)					
00FF35	D7	PTD07	Timer 0 counter data D7 (MSB)	High	Low	1	R	
	D6	PTD06	Timer 0 counter data D6					
	D5	PTD05	Timer 0 counter data D5					
	D4	PTD04	Timer 0 counter data D4					
	D3	PTD03	Timer 0 counter data D3					
	D2	PTD02	Timer 0 counter data D2					
	D1	PTD01	Timer 0 counter data D1					
	D0	PTD00	Timer 0 counter data D0 (LSB)					
00FF36	D7	PTD17	Timer 1 counter data D7 (MSB)	High	Low	1	R	
	D6	PTD16	Timer 1 counter data D6					
	D5	PTD15	Timer 1 counter data D5					
	D4	PTD14	Timer 1 counter data D4					
	D3	PTD13	Timer 1 counter data D3					
	D2	PTD12	Timer 1 counter data D2					
	D1	PTD11	Timer 1 counter data D1					
	D0	PTD10	Timer 1 counter data D0 (LSB)					

Table 5.1.1(g) I/O Memory map (00FF40H–00FF41H)

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF40	D7	–	–	–	–	–		"0" when being read
	D6	FOUT2	FOUT frequency selection FOUT2 FOUT1 FOUT0 Frequency 0 0 0 f_{OSC1} / 1 0 0 1 f_{OSC1} / 2 0 1 0 f_{OSC1} / 4 0 1 1 f_{OSC1} / 8 1 0 0 f_{OSC3} / 1 1 0 1 f_{OSC3} / 2 1 1 0 f_{OSC3} / 4 1 1 1 f_{OSC3} / 8			0	R/W	
	D5	FOUT1				0	R/W	
	D4	FOUT0				0	R/W	
	D3	FOUTON	FOUT output control	On	Off	0	R/W	
	D2	WDRST	Watchdog timer reset	Reset	No operation	–	W	Constantly "0" when being read
	D1	TMRST	Clock timer reset	Reset	No operation	–	W	
	D0	TMRUN	Clock timer Run/Stop control	Run	Stop	0	R/W	
00FF41	D7	TMD7	Clock timer data 1 Hz	High	Low	0	R	
	D6	TMD6	Clock timer data 2 Hz					
	D5	TMD5	Clock timer data 4 Hz					
	D4	TMD4	Clock timer data 8 Hz					
	D3	TMD3	Clock timer data 16 Hz					
	D2	TMD2	Clock timer data 32 Hz					
	D1	TMD1	Clock timer data 64 Hz					
	D0	TMD0	Clock timer data 128 Hz					

Table 5.1.1(h) I/O Memory map (00FF42H–00FF45H)

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF42	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	–	–	–	–	–		
	D2	–	–	–	–	–		
	D1	SWRST	Stopwatch timer reset	Reset	No operation	–	W	
	D0	SWRUN	Stopwatch timer Run/Stop control	Run	Stop	0	R/W	
00FF43	D7	SWD7	Stopwatch timer data			0	R	
	D6	SWD6						
	D5	SWD5	BCD (1/10 sec)					
	D4	SWD4						
	D3	SWD3	Stopwatch timer data					
	D2	SWD2						
	D1	SWD1	BCD (1/100 sec)					
	D0	SWD0						
00FF44	D7	–	–	–	–	–		Constantry "0" when being read
	D6	BZSTP	One-shot buzzer forcibly stop	Forcibly stop	No operation	–	W	
	D5	BZSHT	One-shot buzzer trigger/status R W	Busy Trigger	Ready No operation	0	R/W	
	D4	SHTPW	One-shot buzzer duration width selection	125 msec	31.25 msec	0	R/W	
	D3	ENRTM	Envelope attenuation time	1 sec	0.5 sec	0	R/W	
	D2	ENRST	Envelope reset	Reset	No operation	–	W	"0" when being read
	D1	ENON	Envelope On/Off control	On	Off	0	R/W	*1
	D0	BZON	Buzzer output control	On	Off	0	R/W	
00FF45	D7	–	–	–	–	–		"0" when being read
	D6	DUTY2	Buzzer signal duty ratio selection DUTY2–1 / Buzzer frequency (Hz) 2 1 0 / 4096.0 2048.0 / 3276.8 1638.4 / 2730.7 1365.3 / 2340.6 1170.3			0	R/W	
	D5	DUTY1	0 0 0 8/16 8/20 12/24 12/28 0 0 1 7/16 7/20 11/24 11/28 0 1 0 6/16 6/20 10/24 10/28 0 1 1 5/16 5/20 9/24 9/28			0	R/W	
	D4	DUTY0	1 0 0 4/16 4/20 8/24 8/28 1 0 1 3/16 3/20 7/24 7/28 1 1 0 2/16 2/20 6/24 6/28 1 1 1 1/16 1/20 5/24 5/28			0	R/W	
	D3	–	–			–		"0" when being read
	D2	BZFQ2	Buzzer frequency selection BZFQ2 BZFQ1 BZFQ0 Frequency (Hz) 0 0 0 4096.0			0	R/W	
	D1	BZFQ1	0 0 1 3276.8 0 1 0 2730.7 0 1 1 2340.6 1 0 0 2048.0			0	R/W	
	D0	BZFQ0	1 0 1 1638.4 1 1 0 1365.3 1 1 1 1170.3			0	R/W	

*1 Reset to "0" during one-shot output.

Table 5.1.1(i) I/O Memory map (00FF48H–00FF4AH)

Address	Bit	Name	Function		1	0	SR	R/W	Comment
00FF48	D7	–	–		–	–	–		"0" when being read
	D6	EPR	Parity enable register		With parity	Non parity	0	R/W	Only for asynchronous mode
	D5	PMD	Parity mode selection		Odd	Even	0	R/W	
	D4	SCS1	Clock source selection SCS1 SCS0 Clock source 1 1 Programmable timer 1 0 f_{OSC3} / 4 0 1 f_{OSC3} / 8 0 0 f_{OSC3} / 16				0	R/W	In the clock synchronous slave mode, external clock is selected.
	D3	SCS0					0	R/W	
	D2	SMD1	Serial I/F mode selection SMD1 SMD0 Mode 1 1 Asynchronous 8-bit 1 0 Asynchronous 7-bit 0 1 Clock synchronous slave 0 0 Clock synchronous master				0	R/W	
	D1	SMD0					0	R/W	
	D0	ESIF	Serial I/F enable register		Serial I/F	I/O port	0	R/W	
00FF49	D7	–	–		–	–	–		"0" when being read
	D6	FER	Framing error flag	R W	Error Reset (0)	No error No operation	0	R/W	Only for asynchronous mode
	D5	PER	Parity error flag	R W	Error Reset (0)	No error No operation	0	R/W	
	D4	OER	Overrun error flag	R W	Error Reset (0)	No error No operation	0	R/W	
	D3	RXTRG	Receive trigger/status	R W	Run Trigger	Stop No operation	0	R/W	
	D2	RXEN	Receive enable		Enable	Disable	0	R/W	
	D1	TXTRG	Transmit trigger/status	R W	Run Trigger	Stop No operation	0	R/W	
	D0	TXEN	Transmit enable		Enable	Disable	0	R/W	
00FF4A	D7	TRXD7	Transmit/Receive data D7 (MSB)		High	Low	X	R/W	
	D6	TRXD6	Transmit/Receive data D6						
	D5	TRXD5	Transmit/Receive data D5						
	D4	TRXD4	Transmit/Receive data D4						
	D3	TRXD3	Transmit/Receive data D3						
	D2	TRXD2	Transmit/Receive data D2						
	D1	TRXD1	Transmit/Receive data D1						
	D0	TRXD0	Transmit/Receive data D0 (LSB)						

Table 5.1.1(j) I/O Memory map (00FF50H–00FF55H)

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF50	D7	SIK07	K07 interrupt selection register	Interrupt enable	Interrupt disable	0	R/W	
	D6	SIK06	K06 interrupt selection register					
	D5	SIK05	K05 interrupt selection register					
	D4	SIK04	K04 interrupt selection register					
	D3	SIK03	K03 interrupt selection register					
	D2	SIK02	K02 interrupt selection register					
	D1	SIK01	K01 interrupt selection register					
	D0	SIK00	K00 interrupt selection register					
00FF51	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	–	–	–	–	–		
	D2	–	–	–	–	–		
	D1	SIK11	K11 interrupt selection register	Interrupt enable	Interrupt disable	0	R/W	
	D0	SIK10	K10 interrupt selection register					
00FF52	D7	KCP07	K07 interrupt comparison register	Interrupt generated at falling edge	Interrupt generated at rising edge	1	R/W	
	D6	KCP06	K06 interrupt comparison register					
	D5	KCP05	K05 interrupt comparison register					
	D4	KCP04	K04 interrupt comparison register					
	D3	KCP03	K03 interrupt comparison register					
	D2	KCP02	K02 interrupt comparison register					
	D1	KCP01	K01 interrupt comparison register					
	D0	KCP00	K00 interrupt comparison register					
00FF53	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	–	–	–	–	–		
	D2	–	–	–	–	–		
	D1	KCP11	K11 interrupt comparison register	Falling edge	Rising edge	1	R/W	
	D0	KCP10	K10 interrupt comparison register					
00FF54	D7	K07D	K07 input port data	High level input	Low level input	–	R	
	D6	K06D	K06 input port data					
	D5	K05D	K05 input port data					
	D4	K04D	K04 input port data					
	D3	K03D	K03 input port data					
	D2	K02D	K02 input port data					
	D1	K01D	K01 input port data					
	D0	K00D	K00 input port data					
00FF55	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	_		
	D4	–	–	–	–	_		
	D3	–	–	–	–	_		
	D2	–	–	–	–	_		
	D1	K11D	K11 input port data	High level input	Low level input	–	R	
	D0	K10D	K10 input port data					

Table 5.1.1(k) I/O Memory map (00FF60H–00FF63H)

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF60	D7	IOC07	P07 I/O control register	Output	Input	0	R/W	
	D6	IOC06	P06 I/O control register					
	D5	IOC05	P05 I/O control register					
	D4	IOC04	P04 I/O control register					
	D3	IOC03	P03 I/O control register					
	D2	IOC02	P02 I/O control register					
	D1	IOC01	P01 I/O control register					
	D0	IOC00	P00 I/O control register					
00FF61	D7	IOC17	P17 I/O control register	Output	Input	0	R/W	
	D6	IOC16	P16 I/O control register					
	D5	IOC15	P15 I/O control register					
	D4	IOC14	P14 I/O control register					
	D3	IOC13	P13 I/O control register					
	D2	IOC12	P12 I/O control register					
	D1	IOC11	P11 I/O control register					
	D0	IOC10	P10 I/O control register					
00FF62	D7	P07D	P07 I/O port data	High	Low	1	R/W	
	D6	P06D	P06 I/O port data					
	D5	P05D	P05 I/O port data					
	D4	P04D	P04 I/O port data					
	D3	P03D	P03 I/O port data					
	D2	P02D	P02 I/O port data					
	D1	P01D	P01 I/O port data					
	D0	P00D	P00 I/O port data					
00FF63	D7	P17D	P17 I/O port data	High	Low	1	R/W	
	D6	P16D	P16 I/O port data					
	D5	P15D	P15 I/O port data					
	D4	P14D	P14 I/O port data					
	D3	P13D	P13 I/O port data					
	D2	P12D	P12 I/O port data					
	D1	P11D	P11 I/O port data					
	D0	P10D	P10 I/O port data					

Table 5.1.1(l) I/O Memory map (00FF70H–00FF75H)

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF70	D7	HZR51	R51 high impedance control	High impedance	Complementary	0	R/W	
	D6	HZR50	R50 high impedance control					
	D5	HZR4H	R/W register	1	0	0	R/W	Reserved register
	D4	HZR4L	R/W register					
	D3	HZR1H	R14–R17 high impedance control	High impedance	Complementary	0	R/W	
	D2	HZR1L	R10–R13 high impedance control					
	D1	HZR0H	R04–R07 high impedance control					
	D0	HZR0L	R00–R03 high impedance control					
00FF71	D7	HZR27	R27 high impedance control	High impedance	Complementary	0	R/W	
	D6	HZR26	R26 high impedance control					
	D5	HZR25	R25 high impedance control					
	D4	HZR24	R24 high impedance control					
	D3	HZR23	R23 high impedance control					
	D2	HZR22	R22 high impedance control					
	D1	HZR21	R21 high impedance control					
	D0	HZR20	R20 high impedance control					
00FF72	D7	HZR37	R37 high impedance control	High impedance	Complementary	0	R/W	
	D6	HZR36	R36 high impedance control					
	D5	HZR35	R35 high impedance control					
	D4	HZR34	R34 high impedance control					
	D3	HZR33	R33 high impedance control					
	D2	HZR32	R32 high impedance control					
	D1	HZR31	R31 high impedance control					
	D0	HZR30	R30 high impedance control					
00FF73	D7	R07D	R07 output port data	High	Low	1	R/W	
	D6	R06D	R06 output port data					
	D5	R05D	R05 output port data					
	D4	R04D	R04 output port data					
	D3	R03D	R03 output port data					
	D2	R02D	R02 output port data					
	D1	R01D	R01 output port data					
	D0	R00D	R00 output port data					
00FF74	D7	R17D	R17 output port data	High	Low	1	R/W	
	D6	R16D	R16 output port data					
	D5	R15D	R15 output port data					
	D4	R14D	R14 output port data					
	D3	R13D	R13 output port data					
	D2	R12D	R12 output port data					
	D1	R11D	R11 output port data					
	D0	R10D	R10 output port data					
00FF75	D7	R27D	R27 output port data	High	Low	1	R/W	
	D6	R26D	R26 output port data					
	D5	R25D	R25 output port data					
	D4	R24D	R24 output port data					
	D3	R23D	R23 output port data					
	D2	R22D	R22 output port data					
	D1	R21D	R21 output port data					
	D0	R20D	R20 output port data					

Table 5.1.1(m) I/O Memory map (00FF76H–00FF78H)

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF76	D7	R37D	R37 output port data	High	Low	1	R/W	
	D6	R36D	R36 output port data					
	D5	R35D	R35 output port data					
	D4	R34D	R34 output port data					
	D3	R33D	R33 output port data					
	D2	R32D	R32 output port data					
	D1	R31D	R31 output port data					
	D0	R30D	R30 output port data					
00FF77	D7	R47D	R/W register	1	0	1	R/W	Reserved register
	D6	R46D	R/W register					
	D5	R45D	R/W register					
	D4	R44D	R/W register					
	D3	R43D	R/W register					
	D2	R42D	R/W register					
	D1	R41D	R/W register					
	D0	R40D	R/W register					
00FF78	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	–	–	–	–	–		
	D2	–	–	–	–	–		
	D1	R51D	R51 output port data	High	Low	1	R/W	
	D0	R50D	R50 output port data	High	Low	0	R/W	

5.2 System Controller and Bus Control

The system controller is a management unit which sets such items as the bus mode in accordance with memory system configuration factors.
For the purposes of controlling the system, the following settings can be performed in software:

(1) Bus mode (CPU mode) settings
(2) Chip enable ($\overline{\text{CE}}$) signal output settings
(3) WAIT state settings for external memory
(4) Bus authority release request / acknowledge signal ($\overline{\text{BREQ}}$/$\overline{\text{BACK}}$) settings
(5) Page address setting of the stack pointer

Below is a description of the how these settings are to be made.

5.2.1 Bus mode settings

As explained in "3.5.2 Bus mode", the E0C88112 has four bus modes. Settings for bus modes must be made in software and must match the capacity of the external memory.

As shown in Table 5.2.1.1, bus mode settings are performed on the basis of the preset values for each mode written to the registers BSMD0 and BSMD1.

Table 5.2.1.1 Bus mode settings

Setting value		Bus mode	Configuration of external memory
BSMD1	BSMD0		
1	1	Expanded 512K maximum mode	ROM+RAM>64K bytes (Program>64K bytes)
1	0	Expanded 512K minimum mode	ROM+RAM>64K bytes (Program≤64K bytes)
0	1	Expanded 64K mode	ROM+RAM≤64K bytes
0	0	Single chip mode (MCU)	None
		Optional setting of one of the expanded modes (MPU)	See above

* The single chip mode setting is only possible when this IC is used in the MCU mode. The single chip mode setting is incompatible with the MPU mode, since this mode does not utilize internal ROM.
When using in the MPU mode, it is necessary to select the bus mode at the time of the initial resetting and at the time of the <BSMD1 = "0" and BSMD0 = "0"> setting from among the three types of expanded modes (expanded 64K mode, expanded 512K minimum mode and expanded 512K maximum mode) by mask option.
Select the expanded 512K maximum mode for this option, when the MPU mode is not used at all.

The function of I/O terminals is set as shown in Table 5.2.1.2 in accordance with mode selection.

Table 5.2.1.2 I/O terminal settings

Terminal	Bus mode		
	Single chip	Expanded 64K mode	Expanded 512K mode
R00	Output port R00	Address bus A0	
R01	Output port R01	Address bus A1	
R02	Output port R02	Address bus A2	
R03	Output port R03	Address bus A3	
R04	Output port R04	Address bus A4	
R05	Output port R05	Address bus A5	
R06	Output port R06	Address bus A6	
R07	Output port R07	Address bus A7	
R10	Output port R10	Address bus A8	
R11	Output port R11	Address bus A9	
R12	Output port R12	Address bus A10	
R13	Output port R13	Address bus A11	
R14	Output port R14	Address bus A12	
R15	Output port R15	Address bus A13	
R16	Output port R16	Address bus A14	
R17	Output port R17	Address bus A15	
R20	Output port R20		Address bus A16
R21	Output port R21		Address bus A17
R22	Output port R22		Address bus A18
R23	Output port R23	$\overline{\text{RD}}$ signal	
R24	Output port R24	$\overline{\text{WR}}$ signal	
P00	I/O port P00	Data bus D0	
P01	I/O port P01	Data bus D1	
P02	I/O port P02	Data bus D2	
P03	I/O port P03	Data bus D3	
P04	I/O port P04	Data bus D4	
P05	I/O port P05	Data bus D5	
P06	I/O port P06	Data bus D6	
P07	I/O port P07	Data bus D7	

At initial reset, the bus mode is set as explained below.

- **In MCU mode:**
  At initial reset, the E0C88112 is set in single chip mode.
  Accordingly, in MCU mode, even if a memory has been externally expanded, the system is activated by the program written to internal ROM.
  In systems with externally expanded memory, perform the applicable bus mode settings during the initialization routine originating in internal ROM.

- **In MPU mode:**
  When the MPU mode is used, the expanded mode (expanded 64K mode, expanded 512K minimum mode or expanded 512K maximum mode) set during initial reset must be preselected by mask option.
  You should set it to conform properly to system configuration.

5.2.2 Address decoder ($\overline{\text{CE}}$ output) settings

As explained in Section 3.6.4, the E0C88112 is equipped with address decoders that can output a maximum of four chip enable signals ($\overline{\text{CE0}}$–$\overline{\text{CE3}}$) to external devices.
The output terminals and output circuits for $\overline{\text{CE0}}$–$\overline{\text{CE3}}$ are shared with output ports R30–R33. At initial reset, they are set as output port terminals. For this reason, when operating in a mode other than single chip mode, the ports to be used as $\overline{\text{CE}}$ signal output terminals must be set as such.
This setting is performed through software which writes "1" to registers CE0–CE3 corresponding the $\overline{\text{CE}}$ signals to be used.
Table 5.2.2.1 shows the address range assigned to the four chip enable ($\overline{\text{CE}}$) signals.

The arrangement of memory space for external devices does not necessarily have to be continuous from a subordinate address and any of the chip enable signals can be used to assign areas in memory. However, in the MPU mode, program memory must be assigned to $\overline{\text{CE0}}$.
In the expanded 512K mode, the address range of each of the $\overline{\text{CE}}$ signals is fixed. In the expanded 64K mode, the four address ranges, which match the amount of memory in use, are selected with registers CEMD0 and CEMD1.

These signals are only output when the appointed external memory area is accessed and are not output when internal memory is accessed.

Table 5.2.2.1 Address settings of $\overline{\text{CE0}}$–$\overline{\text{CE3}}$

(1) Expanded 64K mode + MCU mode

CEMD1	CEMD0	Chip size	$\overline{\text{CE0}}$	$\overline{\text{CE1}}$	$\overline{\text{CE2}}$	$\overline{\text{CE3}}$
1	1	64K bytes	004000H–00EFFFH	–	–	–
1	0	32K bytes	004000H–007FFFH	008000H–00EFFFH	–	–
0	1	16K bytes	–	004000H–007FFFH	008000H-00BFFFH	00C000H–00EFFFH
0	0	8K bytes	008000H–009FFFH	00A000H–00BFFFH	004000H–005FFFH	006000H–007FFFH

(2) Expanded 64K mode + MPU mode

CEMD1	CEMD0	Chip size	$\overline{\text{CE0}}$	$\overline{\text{CE1}}$	$\overline{\text{CE2}}$	$\overline{\text{CE3}}$
1	1	64K bytes	000000H–00EFFFH	–	–	–
1	0	32K bytes	000000H–007FFFH	008000H–00EFFFH	–	–
0	1	16K bytes	000000H–003FFFH	004000H–007FFFH	008000H–00BFFFH	00C000H–00EFFFH
0	0	8K bytes	000000H–001FFFH	002000H–003FFFH	004000H–005FFFH	006000H–007FFFH

(3) Expanded 512K minimum/maximum modes

$\overline{\text{CE}}$ signal	Address range	
	MCU mode	MPU mode
$\overline{\text{CE0}}$	200000H–27FFFFH	000000H–00EFFFH, 010000H–07FFFFH
$\overline{\text{CE1}}$	080000H–0FFFFFH	080000H–0FFFFFH
$\overline{\text{CE2}}$	100000H–17FFFFH	100000H–17FFFFH
$\overline{\text{CE3}}$	180000H–1FFFFFH	180000H–1FFFFFH

5.2.3 WAIT state settings

In order to insure accessing of external low speed devices during high speed operations, the E0C88112 is equipped with a WAIT function which prolongs access time.
The number of wait states inserted can be selected from a choice of eight as shown in Table 5.2.3.1 by means of registers WT0–WT2.

Table 5.2.3.1 Setting the number of WAIT states

WT2	WT1	WT0	Number of inserted states
1	1	1	14
1	1	0	12
1	0	1	10
1	0	0	8
0	1	1	6
0	1	0	4
0	0	1	2
0	0	0	No wait

* A state is 1/2 cycles of the clock in length.

WAIT states set in software are inserted between bus cycle states T3–T4.
Note, however, that WAIT states cannot be inserted when an internal register and internal memory are being accessed and when operating with the OSC1 oscillation circuit (see "5.4 Oscillation Circuits and Operating Mode").
Consequently, WAIT state settings in single chip mode are meaningless.
With regard to WAIT insertion timing, see Section 3.6.5, "WAIT control".

5.2.4 Setting the bus authority release request signal

With systems performing DMA transfer, the bus authority release request signal ($\overline{\text{BREQ}}$) input terminal and acknowledge signal ($\overline{\text{BACK}}$) output terminal have to be set.
The $\overline{\text{BREQ}}$ input terminal is shared with input port terminal K11 and the $\overline{\text{BACK}}$ output terminal with output port terminal R51. At initial reset, these terminal facilities are set as input port terminal and output port terminal, respectively. The terminals can be altered to function as $\overline{\text{BREQ}}$/$\overline{\text{BACK}}$ terminals by writing a "1" to register EBR.

For details on bus authority release, see "3.6.6 Bus authority release state" and "E0C88 Core CPU Manual".

5.2.5 Stack page setting

Although the stack area used to evacuate registers during subroutine calls can be arbitrarily moved to any area in data RAM using the stack pointer SP, its page address is set in registers SPP0–SPP7 in I/O memory.
At initial reset, SPP0–SPP7 are set to "00H" (page 0).

Since the internal RAM is arranged on page 0 (00F000H–00F0FFH), the stack area in single chip mode is inevitably located in page 0.
In expanded 64K mode where RAM is externally expanded, stack page is likewise limited to page 0. In order to place the stack area at the final address in internal RAM, the stack pointer SP is placed at an initial setting of "F100H". (SP is pre-decremented.)

In the expanded 512K mode, to place the stack in external expanded RAM, set a corresponding page to SPP0–SPP7. The page addresses to which SPP0–SPP7 can be set are 00H–27H and must be within a RAM area.

* A page is each recurrent 64K division of data memory beginning at address zero.

5.2.6 Control of system controller

Table 5.2.6.1 shows the control bits for the system controller.

Table 5.2.6.1(a) System controller control bits (MCU mode)

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF00 (MCU)	D7	BSMD1	Bus mode (CPU mode) BSMD1 BSMD0 Mode 1 1 512K (Maximum) 1 0 512K (Minimum) 0 1 64K 0 0 Single chip			0	R/W	
	D6	BSMD0				0	R/W	
	D5	CEMD1	Chip enable mode CEMD1 CEMD0 Mode 1 1 64K ($\overline{CE0}$) 1 0 32K ($\overline{CE0}$, $\overline{CE1}$) 0 1 16K ($\overline{CE1}$–$\overline{CE3}$) 0 0 8K ($\overline{CE0}$–$\overline{CE3}$)			1	R/W	Only for 64K bus mode
	D4	CEMD0				1	R/W	
	D3	CE3	$\overline{CE3}$ (R33) $\overline{CE}$ signal output Enable/Disable Enable: $\overline{CE}$ signal output Disable: DC (R3x) output	$\overline{CE3}$ enable	$\overline{CE3}$ disable	0	R/W	In the Single chip mode, these setting are fixed at DC output.
	D2	CE2	$\overline{CE2}$ (R32)	$\overline{CE2}$ enable	$\overline{CE2}$ disable	0	R/W	
	D1	CE1	$\overline{CE1}$ (R31)	$\overline{CE1}$ enable	$\overline{CE1}$ disable	0	R/W	
	D0	CE0	$\overline{CE0}$ (R30)	$\overline{CE0}$ enable	$\overline{CE0}$ disable	0	R/W	
00FF01	D7	SPP7	Stack pointer page address (MSB)	1	0	0	R/W	
	D6	SPP6		1	0	0	R/W	
	D5	SPP5	< SP page allocatable address >	1	0	0	R/W	
	D4	SPP4	• Single chip mode: only 0 page	1	0	0	R/W	
	D3	SPP3	• 64K mode: only 0 page	1	0	0	R/W	
	D2	SPP2	• 512K (min) mode: 0–27H page	1	0	0	R/W	
	D1	SPP1	• 512K (max) mode: 0–27H page	1	0	0	R/W	
	D0	SPP0	(LSB)	1	0	0	R/W	
00FF02	D7	EBR	Bus release enable register K11 (K11 and R51 terminal specification) R51	$\overline{BREQ}$ $\overline{BACK}$	Input port Output port	0	R/W	
	D6	WT2	Wait control register WT2 WT1 WT0 Number of state 1 1 1 14 1 1 0 12 1 0 1 10 1 0 0 8 0 1 1 6 0 1 0 4 0 0 1 2 0 0 0 No wait			0	R/W	
	D5	WT1						
	D4	WT0						
	D3	CLKCHG	CPU operating clock switch	OSC3	OSC1	0	R/W	
	D2	OSCC	OSC3 oscillation On/Off control	On	Off	0	R/W	
	D1	VDC1	Operating mode selection VDC1 VDC0 Operating mode 1 × High speed (V_{D1}=3.3V) 0 1 Low power (V_{D1}=1.3V) 0 0 Normal (V_{D1}=2.2V)			0	R/W	
	D0	VDC0						

Note: *All the interrupts including $\overline{NMI}$ are disabled, until you write the optional value into both the "00FF00H" and "00FF01H" addresses.*

Table 5.2.6.1(b) System controller control bits (MPU mode)

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF00 (MPU)	D7	BSMD1	Bus mode (CPU mode) BSMD1 BSMD0 Mode 1 1 512K (Maximum) 1 0 512K (Minimum) 0 1 64K 0 0 * Option selection ◀			*	R/W	* Initial setting can be selected among 3 types (64K, 512K min and 512K max) by mask option setting.
	D6	BSMD0				*	R/W	
	D5	CEMD1	Chip enable mode CEMD1 CEMD0 Mode 1 1 64K ($\overline{\text{CE0}}$) 1 0 32K ($\overline{\text{CE0}}$, $\overline{\text{CE1}}$) 0 1 16K ($\overline{\text{CE0}}$–$\overline{\text{CE3}}$) 0 0 8K ($\overline{\text{CE0}}$–$\overline{\text{CE3}}$)			1	R/W	Only for 64K bus mode
	D4	CEMD0				1	R/W	
	D3	CE3	$\overline{\text{CE3}}$ (R33) $\overline{\text{CE}}$ signal output Enable/Disable Enable: $\overline{\text{CE}}$ signal output Disable: DC (R3x) output	$\overline{\text{CE3}}$ enable	$\overline{\text{CE3}}$ disable	0	R/W	
	D2	CE2	$\overline{\text{CE2}}$ (R32)	$\overline{\text{CE2}}$ enable	$\overline{\text{CE2}}$ disable	0	R/W	
	D1	CE1	$\overline{\text{CE1}}$ (R31)	$\overline{\text{CE1}}$ enable	$\overline{\text{CE1}}$ disable	0	R/W	
	D0	CE0	$\overline{\text{CE0}}$ (R30)	$\overline{\text{CE0}}$ enable	$\overline{\text{CE0}}$ disable	1	R/W	
00FF01	D7	SPP7	Stack pointer page address (MSB)	1	0	0	R/W	
	D6	SPP6		1	0	0	R/W	
	D5	SPP5	< SP page allocatable address >	1	0	0	R/W	
	D4	SPP4	• Single chip mode: only 0 page	1	0	0	R/W	
	D3	SPP3	• 64K mode: only 0 page	1	0	0	R/W	
	D2	SPP2	• 512K (min) mode: 0–27H page	1	0	0	R/W	
	D1	SPP1	• 512K (max) mode:0–27H page	1	0	0	R/W	
	D0	SPP0	(LSB)	1	0	0	R/W	
00FF02	D7	EBR	Bus release enable register K11 (K11 and R51 terminal specification) R51	$\overline{\text{BREQ}}$ $\overline{\text{BACK}}$	Input port Output port	0	R/W	
	D6	WT2	Wait control register WT2 WT1 WT0 Number of state 1 1 1 14 1 1 0 12 1 0 1 10 1 0 0 8 0 1 1 6 0 1 0 4 0 0 1 2 0 0 0 No wait			0	R/W	
	D5	WT1						
	D4	WT0						
	D3	CLKCHG	CPU operating clock switch	OSC3	OSC1	0	R/W	
	D2	OSCC	OSC3 oscillation On/Off control	On	Off	0	R/W	
	D1	VDC1	Operating mode selection VDC1 VDC0 Operating mode 1 × High speed (V_{D1}=3.3V) 0 1 Low power (V_{D1}=1.3V) 0 0 Normal (V_{D1}=2.2V)			0	R/W	
	D0	VDC0						

Note: All the interrupts including $\overline{\text{NMI}}$ are disabled, until you write the optional value into both the "00FF00H" and "00FF01H" addresses.

BSMD0, BSMD1: 00FF00H•D6, D7

Bus modes are set as shown in Table 5.2.6.2.

Table 5.2.6.2 Bus mode settings

Setting values		Bus mode
BSMD1	BSMD0	
1	1	Expanded 512K maximum mode
1	0	Expanded 512K minimum mode
0	1	Expanded 64K mode
0	0	Single chip mode (MCU)
		Optional setting of one of the expanded modes (MPU)

The single chip mode setting is only possible when this IC is used in the MCU mode. The single chip mode setting is incompatible with the MPU mode, since this mode does not utilize internal ROM.
When using in the MPU mode, it is necessary to select the bus mode at the time of the initial resetting and at the time of the <BSMD1 = "0" and BSMD0 = "0"> setting from among the three types of expanded modes (expanded 64K mode, expanded 512K minimum mode and expanded 512K maximum mode) by mask option.
Select the expanded 512K maximum mode for this option, when the MPU mode is not used at all.
At initial reset, in the MCU mode the unit is set to single chip mode and in the MPU mode the mask option is used to select the applicable mode.

CEMD0, CEMD1: 00FF00H•D4, D5

Sets the $\overline{\text{CE}}$ signal address range (valid only in the expanded 64K mode).
Settings are made according to external memory chip size as shown in Table 5.2.6.3.

Table 5.2.6.3 $\overline{\text{CE}}$ signal settings

CEMD1	CEMD0	Address range	Usable terminals
1	1	64K bytes	$\overline{\text{CE0}}$
1	0	32K bytes	$\overline{\text{CE0}}$, $\overline{\text{CE1}}$
0	1	16K bytes	$\overline{\text{CE0}}$–$\overline{\text{CE3}}$..MPU mode $\overline{\text{CE1}}$–$\overline{\text{CE3}}$..MCU mode
0	0	8K bytes	$\overline{\text{CE0}}$–$\overline{\text{CE3}}$

These settings are invalid for any mode other than expanded 64K mode.
At initial reset, each register is set to "1" (64K bytes).

CE0–CE3: 00FF00H•D0–D3

Sets the $\overline{\text{CE}}$ output terminals being used.

When "1" is written: $\overline{\text{CE}}$ output enable
When "0" is written: $\overline{\text{CE}}$ output disable
Reading: Valid

$\overline{\text{CE}}$ output is enabled when a "1" is written to registers CE0–CE3 which correspond to the $\overline{\text{CE}}$ output being used. A "0" written to any of the registers disables $\overline{\text{CE}}$ signal output from that terminal and it reverts to its alternate function as an output port terminal (R30–R33).
At initial reset, register CE0 is set to "0" in the MCU mode and in the MPU mode, "1" is set in the register. Registers CE1–CE3 are always set to "0" regardless of the MCU/MPU mode setting.

Note: *To avoid a malfunction from an interrupt generated before the bus configuration is initialized, all interrupts including $\overline{\text{NMI}}$ are masked until you write an optional value into address "00FF00H".*

SPP0–SPP7: 00FF01H

Sets the page address of stack area. In single chip mode and expanded 64K mode, set page address to "00H".
In expanded 512K mode, it can be set to any value within the range "00H"–"27H".
Since a carry and borrow from/to the stack pointer SP is not reflected in register SPP, the upper limit on continuous use of the stack area is 64K bytes.
At initial reset, this register is set to "00H" (page 0).

Note: *To avoid a malfunction from an interrupt generated before the bus configuration is initialized, all interrupts including $\overline{\text{NMI}}$ are disabled, until you write an optional value into "00FF01H" address. Furthermore, to avoid generating an interrupt while the stack area is being set, all interrupts including $\overline{\text{NMI}}$ are disabled in one instruction execution period after writing to address "00FF01H".*

WT0–WT2: 00FF02H•D4–D6

How WAIT state settings are performed.
The number of WAIT states to be inserted based on register settings is as shown in Table 5.2.6.4.

Table 5.2.6.4 Setting WAIT states

WT2	WT1	WT0	No. of inserted states
1	1	1	14
1	1	0	12
1	0	1	10
1	0	0	8
0	1	1	6
0	1	0	4
0	0	1	2
0	0	0	No wait

* A state is 1/2 cycles of the clock in length.

At initial reset, this register is set to "0" (NO WAIT).

EBR: 00FF02H•D7

Sets the $\overline{\text{BREQ}}$/$\overline{\text{BACK}}$ terminals function.

When "1" is written: $\overline{\text{BREQ}}$/$\overline{\text{BACK}}$ enabled
When "0" is written: $\overline{\text{BREQ}}$/$\overline{\text{BACK}}$ disabled
Reading: Valid

How $\overline{\text{BREQ}}$ and $\overline{\text{BACK}}$ terminal functions are set.
Writing "1" to EBR enables $\overline{\text{BREQ}}$/$\overline{\text{BACK}}$ input/output. Writing "0" sets the $\overline{\text{BREQ}}$ terminal as input port terminal K11 and the $\overline{\text{BACK}}$ terminal as output port terminal R51.
At initial reset, EBR is set to "0" ($\overline{\text{BREQ}}$/$\overline{\text{BACK}}$ disabled).

5.2.7 Programming notes

(1) All the interrupts including $\overline{\text{NMI}}$ are masked, until you write the optional value into both the "00FF00H" and "00FF01H" addresses. Consequently, even if you do not change the content of this address (You use the initial value, as is.), you should still be sure to perform the writing operation using the initialization routine.

(2) When setting stack fields, including page addresses as well, you should write them in the order of the register SPP ("00FF01H") and the stack pointer SP.

Example: When setting the "178000H" address

```
LD    EP, #00H
LD    HL, #0FF01H
LD    [HL], #17H    ┐ During this period the
LD    SP, #8000H    ┘ interrupts (including
                      NMI) are masked.
```

5.3 Watchdog Timer

5.3.1 Configuration of watchdog timer

The E0C88112 is equipped with a watchdog timer driven by OSC1 as source oscillation. The watchdog timer must be reset periodically in software, and if reset of more than 3–4 seconds (where f_{OSC1} = 32.768 kHz) does not take place, a non-maskable interrupt signal is generated and output to the CPU.

Figure 5.3.1.1 is a block diagram of the watchdog timer.

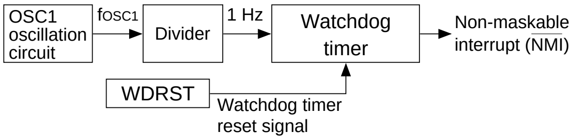


Fig. 5.3.1.1 Block diagram of watchdog timer

By running watchdog timer reset during the main routine of the program, it is possible to detect program runaway as if watchdog timer processing had not been applied. Normally, this routine is integrated at points that are regularly being processed.

The watchdog timer continues to operate during HALT and when a HALT state is continuous for longer than 3–4 seconds, the CPU shifts to exception processing.
During SLEEP, the watchdog timer is stopped.

5.3.2 Interrupt function

In cases where the watchdog timer is not periodically reset in software, the watchdog timer outputs an interrupt signal to the CPU's $\overline{\text{NMI}}$ (level 4) input. Unmaskable and taking priority over other interrupts, this interrupt triggers the generation of exception processing. See the "E0C88 Core CPU Manual" for more details on $\overline{\text{NMI}}$ exception processing.
This exception processing vector is set at 000004H.

5.3.3 Control of watchdog timer

Table 5.3.3.1 shows the control bits for the watchdog timer.

WDRST: 00FF40H•D2

Resets the watchdog timer.

When "1" is written:	Watchdog timer is reset
When "0" is written:	No operation
Reading:	Constantly "0"

By writing "1" to WDRST, the watchdog timer is reset, after which it is immediately restarted. Writing "0" will mean no operation.
Since WDRST is for writing only, it is constantly set to "0" during readout.

5.3.4 Programming note

The watchdog timer must reset within 3-second cycles by software.

Table 5.3.3.1 Watchdog timer control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF40	D7	–	–	–	–	–		"0" when being read
	D6	FOUT2	FOUT frequency selection FOUT2 FOUT1 FOUT0 Frequency 0 0 0 f_{OSC1} / 1			0	R/W	
	D5	FOUT1	0 0 1 f_{OSC1} / 2 0 1 0 f_{OSC1} / 4 0 1 1 f_{OSC1} / 8 1 0 0 f_{OSC3} / 1			0	R/W	
	D4	FOUT0	1 0 1 f_{OSC3} / 2 1 1 0 f_{OSC3} / 4 1 1 1 f_{OSC3} / 8			0	R/W	
	D3	FOUTON	FOUT output control	On	Off	0	R/W	
	D2	WDRST	Watchdog timer reset	Reset	No operation	–	W	Constantly "0" when being read
	D1	TMRST	Clock timer reset	Reset	No operation	–	W	
	D0	TMRUN	Clock timer Run/Stop control	Run	Stop	0	R/W	

5.4 Oscillation Circuits and Operating Mode

5.4.1 Configuration of oscillation circuits

The E0C88112 is twin clock system with two internal oscillation circuits (OSC1 and OSC3). OSC1 oscillation circuit generates the 32.768 kHz (Typ.) main clock and OSC3 oscillation circuit the sub-clock when the CPU and some peripheral circuits (output port, serial interface and programmable timer) are in high speed operation.
Figure 5.4.1.1 shows the configuration of the oscillation circuit.

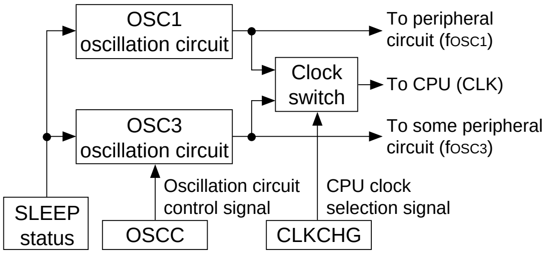


Fig. 5.4.1.1 Configuration of oscillation circuits

At initial reset, OSC1 oscillation circuit is selected for the CPU operating clock and OSC3 oscillation circuit is in a stopped state. ON/OFF switching of the OSC3 oscillation circuit and switching of the system clock between OSC1 and OSC3 are controlled in software. OSC3 circuit is utilized when high speed operation of the CPU and some peripheral circuits become necessary. Otherwise, OSC1 should be used to generate the operating clock and OSC3 circuit placed in a stopped state in order to reduce current consumption.

5.4.2 Mask option

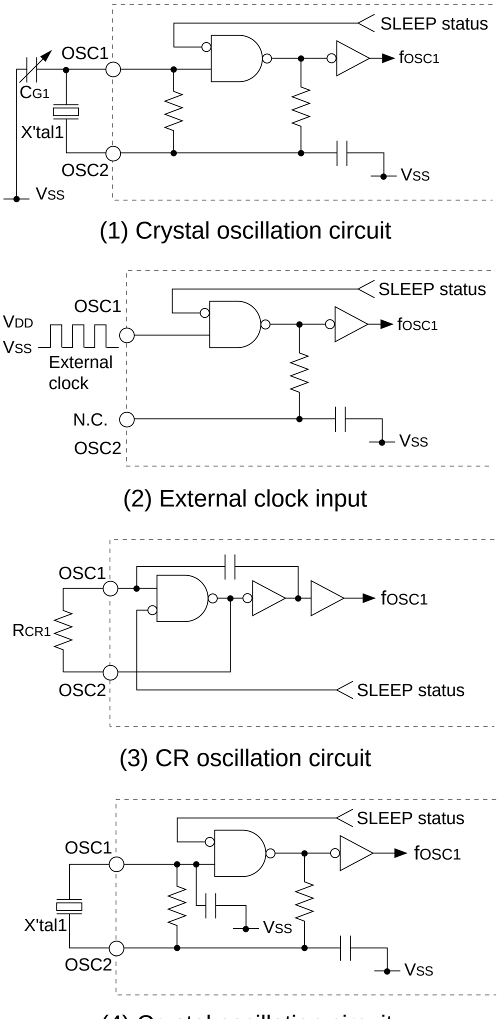
OSC1 oscillation circuit
- □ Crystal oscillation circuit
- □ External clock input
- □ CR oscillation circuit
- □ Crystal oscillation circuit (gate capacitor built-in)

OSC3 oscillation circuit
- □ Crystal oscillation circuit
- □ Ceramic oscillation circuit
- □ CR oscillation circuit
- □ External clock input

In terms of the oscillation circuit types for OSC1, either crystal oscillation, CR oscillation, crystal oscillation (gate capacitor built-in) or external clock input can be selected with the mask option.
In terms of oscillation circuit types for OSC3, either crystal oscillation, ceramic oscillation, CR oscillation or external clock input can be selected with the mask option, in the same way as OSC1.

5.4.3 OSC1 oscillation circuit

The OSC1 oscillation circuit generates the 32.768 kHz (Typ.) system clock which is utilized during low speed operation (low power mode) of the CPU and peripheral circuits. Furthermore, even when OSC3 is utilized as the system clock, OSC1 continues to generate the source clock for the clock timer and stopwatch timer.
This oscillation circuit stops when the SLP instruction is executed. However, in case the SVD circuit is executing an SLP instruction, oscillation is stopped in synchronization with the completion of sampling.
In terms of the oscillation circuit types, either crystal oscillation, CR oscillation, crystal oscillation (gate capacitor built-in) or external clock input can be selected with the mask option.
Figure 5.4.3.1 shows the configuration of the OSC1 oscillation circuit.



(1) Crystal oscillation circuit

(2) External clock input

(3) CR oscillation circuit

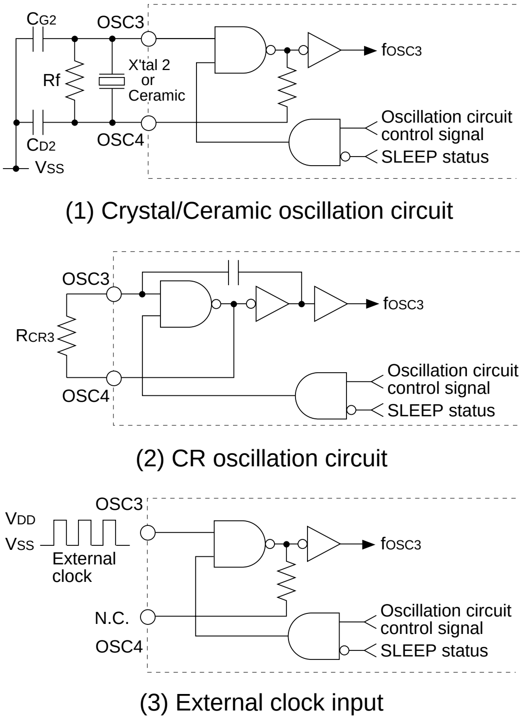
(4) Crystal oscillation circuit

Fig. 5.4.3.1 OSC1 oscillation circuit (gate capacitor built-in)

When crystal oscillation is selected, a crystal oscillation circuit can be easily formed by connecting a crystal oscillator X'tal1 (Typ. 32.768 kHz) between the OSC1 and OSC2 terminals along with a trimmer capacitor C_{G1} (5–25pF) between the OSC1 terminal and V_{SS}.
In addition, the gate capacitor C_{G1} (5 pF) can be built into the circuit by the mask option.
When CR oscillation is selected, connect a resistor (R_{CR1}) between the OSC1 and OSC2 terminals.
When external input is selected, release the OSC2 terminal and input the rectangular wave clock into the OSC1 terminal.

5.4.4 OSC3 oscillation circuit

The OSC3 oscillation circuit generates the system clock when the CPU and some peripheral circuits (output port, serial interface and programmable timer) are in high speed operation.
This oscillation circuit stops when the SLP instruction is executed, or the OSCC register is set to "0".
In terms of oscillation circuit types, any one of crystal oscillation, ceramic oscillation, CR oscillation or external clock input can be selected with the mask option.
Figure 5.4.4.1 shows the configuration of the OSC3 oscillation circuit.



(1) Crystal/Ceramic oscillation circuit

(2) CR oscillation circuit

(3) External clock input

Fig. 5.4.4.1 OSC3 oscillation circuit

When crystal or ceramic oscillation circuit is selected, the crystal or ceramic oscillation circuit are formed by connecting either a crystal oscillator (X'tal2) or a combination of ceramic oscillator (Ceramic) and feedback resistor (Rf) between OSC3 and OSC4 terminals and connecting two capacitors (C_{G2}, C_{D2}) between the OSC3 terminal and V_{SS}, and between the OSC4 terminal and V_{SS}, respectively.
When CR oscillation is selected, the CR oscillation circuit is formed merely by connecting a resistor (R_{CR3}) between OSC3 and OSC4 terminals.
When external input is selected, release the OSC4 terminal and input the rectangular wave clock into the OSC3 terminal.

5.4.5 Operating mode

You can select three types of operating modes using software, to obtain a stable operation and good characteristics (operating frequency and current consumption) over a broad operation voltage. Here below are indicated the features of the respective modes.

- **Normal mode (V_{DD} = 2.4 V–5.5 V)**
  This mode is set following the initial reset. It permits the OSC3 oscillation circuit (Max. 4.2 MHz) to be used and also permits relative low power operation.

- **Low power mode (V_{DD} = 1.8 V–3.5 V)**
  This is a lower power mode than the normal mode. It makes ultra-low power consumption possible by operation on the OSC1 oscillation circuit, although the OSC3 circuit cannot be used.

- **High speed mode (V_{DD} = 3.5 V–5.5 V)**
  This mode permits higher speed operation than the normal mode. Since the OSC3 oscillation circuit (Max. 8.2 MHz) can be used, you should use this mode, when you require operation at 4.2 MHz or more. However, the current consumption will increase relative to the normal mode.

Using software to switch over among the above three modes to meet your actual usage circumstances will make possible a low power system. For example, you will be able to reduce current consumption by switching over to the normal mode when using the OSC3 as the CPU clock and, conversely, changing over to the low power mode when using the OSC1 as the CPU clock (OSC3 oscillation circuit is OFF).

Note: *Do not turn the OSC3 oscillation circuit ON in the low power mode.*
Do not switch over the operating mode (normal mode ↔ high speed mode) in the OSC3 oscillation circuit ON status, as this will cause faulty operation.
You can not use two modes, the low power mode and the high speed mode on one application, with respect to the operating voltages.
When CR oscillation is selected for the OSC1 oscillation circuit, the operating mode is fixed in the normal mode to stabilize the oscillation frequency. Consequently, settings of the mode setting registers VDC0 and VDC1 become invalid.

5.4.6 Switching the CPU clocks

You can use either OSC1 or OSC3 as the system clock for the CPU and you can switch over by means of software.
You can save power by turning the OSC3 oscillation circuit off while the CPU is operating in OSC1. When you must operate on OSC3, you can change to high speed operation by turning the OSC3 oscillation circuit ON and switching over the system clock. In this case, since several msec to several tens of msec are necessary for the oscillation to stabilize after turning the OSC3 oscillation circuit ON, you should switch over the clock after stabilization time has elapsed. (The oscillation start time will vary somewhat depending on the oscillator and on the externally attached parts. Refer to the oscillation start time example indicated in Chapter 7, "ELECTRICAL CHARACTERISTICS".)

When switching over from the OSC3 to the OSC1, turn the OSC3 oscillation circuit OFF immediately following the clock changeover. The basic clock switching procedure is as described above, however, you must also combine it with the changeover of the operating mode to permit low current consumption and high speed operation.
Figure 5.4.6.1 indicates the status transition diagram for the operation mode and clock changeover.

Note: *When turning ON the OSC3 oscillation circuit after switching the operating mode, you should allow a minimum waiting time of 5 msec.*

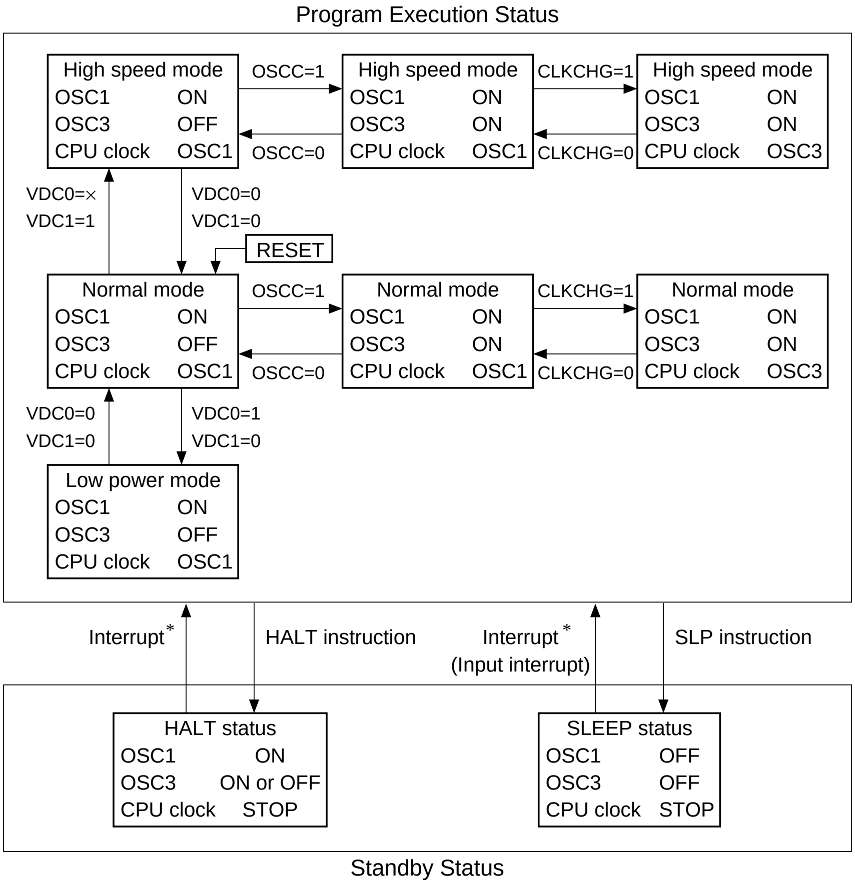


Fig. 5.4.6.1
Status transition diagram for the operation mode and clock changeover

* The return destination from the standby status becomes the program execution status prior to shifting to the standby status

5.4.7 Control of oscillation circuit and operating mode

Table 5.4.7.1 shows the control bits for the oscillation circuits and operating modes.

Table 5.4.7.1 Oscillation circuit and operating mode control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF02	D7	EBR	Bus release enable register K11 (K11 and R51 terminal specification) R51	$\overline{\text{BREQ}}$ $\overline{\text{BACK}}$	Input port Output port	0	R/W	
	D6	WT2	Wait control register WT2 WT1 WT0 Number of state 1 1 1 14 1 1 0 12 1 0 1 10 1 0 0 8 0 1 1 6 0 1 0 4 0 0 1 2 0 0 0 No wait			0	R/W	
	D5	WT1						
	D4	WT0						
	D3	CLKCHG	CPU operating clock switch	OSC3	OSC1	0	R/W	
	D2	OSCC	OSC3 oscillation On/Off control	On	Off	0	R/W	
	D1	VDC1	Operating mode selection VDC1 VDC0 Operating mode 1 × High speed (V_{D1}=3.3V) 0 1 Low power (V_{D1}=1.3V) 0 0 Normal (V_{D1}=2.2V)			0	R/W	
	D0	VDC0						

VDC1, VDC0: 00FF02H•D1, D0

Selects the operating mode according to supply voltage and operating frequency.
Table 5.4.7.2 shows the correspondence between register preset values and operating modes.

Table 5.4.7.2 Correspondence between register preset values and operating modes

Operating mode	VDC1	VDC0	V_{D1}	Power voltage	Operating frequency
Normal mode	0	0	2.2 V	2.4–5.5 V	4.2 MHz (Max.)
Low power mode	0	1	1.3 V	1.8–3.5 V	50 kHz (Max.)
High speed mode	1	×	3.3 V	3.5–5.5 V	8.2 MHz (Max.)

* The V_{D1} voltage is the value where V_{SS} has been made the standard (GND).

At initial reset, this register is set to "0" (normal mode).
When CR oscillation is selected for the OSC1 oscillation circuit, the operating mode is fixed in the normal mode to stabilize the oscillation frequency. Consequently, settings of the mode setting registers VDC0 and VDC1 become invalid.

OSCC: 00FF02H•D2

Controls the ON and OFF settings of the OSC3 oscillation circuit.

When "1" is written: OSC3 oscillation ON
When "0" is written: OSC3 oscillation OFF
Reading: Valid

When the CPU and some peripheral circuits (output port, serial interface and programmable timer) are to be operated at high speed, OSCC is to be set to "1". At all other times, it should be set to "0" in order to reduce current consumption.
At initial reset, OSCC is set to "0" (OSC3 oscillation OFF).

CLKCHG: 00FF02H•D3

Selects the operating clock for the CPU.

When "1" is written:	OSC3 clock
When "0" is written:	OSC1 clock
Reading:	Valid

When the operating clock for the CPU is switched to OSC3, CLKCHG should be set to "1" and when the clock is switched to OSC1, CLKCHG should be set to "0".
At initial reset, CLKCHG is set to "0" (OSC1 clock).

5.4.8 Programming notes

(1) When the high speed CPU operation is not necessary, you should operate the peripheral circuits according to the setting outline indicate below.

- CPU operating clock
  OSC1
- OSC3 oscillation circuit
  OFF (When the OSC3 clock is not necessary for some peripheral circuits.)
- Operating mode
  Low power mode (When V_{DD}–V_{SS} is 3.5 V or less)
  or Normal mode (When V_{DD}–V_{SS} is 3.5 V or more)

(2) Do not turn the OSC3 oscillation circuit ON in the low power mode.
Do not switch over the operating mode (normal mode ↔ high speed mode) in the OSC3 oscillation circuit ON status, as this will cause faulty operation.

(3) When turning ON the OSC3 oscillation circuit after switching the operating mode, you should allow a minimum waiting time of 5 msec.

(4) Since several msec to several tens of msec are necessary for the oscillation to stabilize after turning the OSC3 oscillation circuit ON. Consequently, you should switch the CPU operating clock (OSC1 → OSC3) after allowing for a sufficient waiting time once the OSC3 oscillation goes ON. (The oscillation start time will vary somewhat depending on the oscillator and on the externally attached parts. Refer to the oscillation start time example indicated in Chapter 7, "ELECTRICAL CHARACTERISTICS".)

(5) When switching the clock from OSC3 to OSC1, be sure to switch OSC3 oscillation OFF with separate instructions. Using a single instruction to process simultaneously can cause a malfunction of the CPU.

5.5 Input Ports (K ports)

5.5.1 Configuration of input ports

The E0C88112 is equipped with 10 input port bits (K00–K07, K10 and K11) all of which are usable as general purpose input port terminals with interrupt function.

K10 terminal doubles as the external clock (EVIN) input terminal of the programmable timer (event counter) with input port functions sharing the input signal as is. (See "5.11 Programmable Timer")

Furthermore, it should be noted, however, that K11 terminal is shared with the bus authority release request signal ($\overline{\text{BREQ}}$) input terminal. Function assignment of this terminal can be selected in software. When this terminal is selected for $\overline{\text{BREQ}}$ signal, K11 cannot be used as an input port. (See "5.2 System Controller and Bus Control")
In the explanation below, it is assumed that K11 is set as an input port.

Each input port is equipped with a pull-up resistor. The mask option can be used to select either "With resistor" or "Gate direct" for each input port.
Figure 5.5.1.1 shows the structure of the input port.

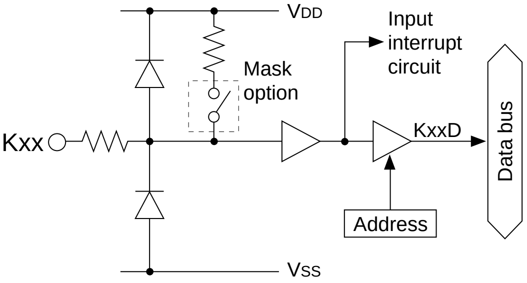


Fig. 5.5.1.1 Structure of input port

Each input port terminal is directly connected via a three-state buffer to the data bus. Furthermore, the input signal state at the instant of input port readout is read in that form as data.

5.5.2 Mask option

Input port pull-up resistors
K00 .... □ With resistor □ Gate direct
K01 .... □ With resistor □ Gate direct
K02 .... □ With resistor □ Gate direct
K03 .... □ With resistor □ Gate direct
K04 .... □ With resistor □ Gate direct
K05 .... □ With resistor □ Gate direct
K06 .... □ With resistor □ Gate direct
K07 .... □ With resistor □ Gate direct

K10 .... □ With resistor □ Gate direct
K11 .... □ With resistor □ Gate direct

Input ports K00–K07, K10 and K11 are all equipped with pull-up resistors. The mask option can be used to select 'With resistor' or 'Gate direct' for each port (bit).

The 'With resistor' option is rendered suitable for purposes such as push switch or key matrix input. When changing the input terminal from LOW level to HIGH with the built-in pull-up resistor, a delay in the waveform rise time will occur depending on the time constant of the pull-up resistor and the load capacitance of the terminal. It is necessary to set an appropriate wait time for introduction of an input port. In particular, special attention should be paid to key scan for key matrix formation. Make this wait time the amount of time or more calculated by the following expression.

Wait time = R_{IN} x (C_{IN} + load capacitance on the board) x 1.6 [sec]

R_{IN}: Pull up resistance Max. value
C_{IN}: Terminal capacitance Max. value

When 'Gate direct' is selected, the pull-up resistor is detached and the port is rendered suitable for purposes such as slide switch input and interfacing with other LSIs.
In this case, take care that a floating state does not occur in input.

For unused input ports, select the default setting of "With resistor".

5.5.3 Interrupt function and input comparison register

Input port K00–K07, K10 and K11 are all equipped with an interrupt function. These input ports are divided into three groupings: K00–K03 (K0L), K04–K07 (K0H) and K10–K11 (K1). Furthermore, the interrupt generation condition for each series of terminals can be set by software.

When the interrupt generation condition set for each series of terminals is met, the interrupt factor flag FK0L, FK0H or FK1 corresponding to the applicable series is set at "1" and an interrupt is generated.

Interrupt can be prohibited by setting the interrupt enable registers EK0L, EK0H and EK1 for the corresponding interrupt factor flags.

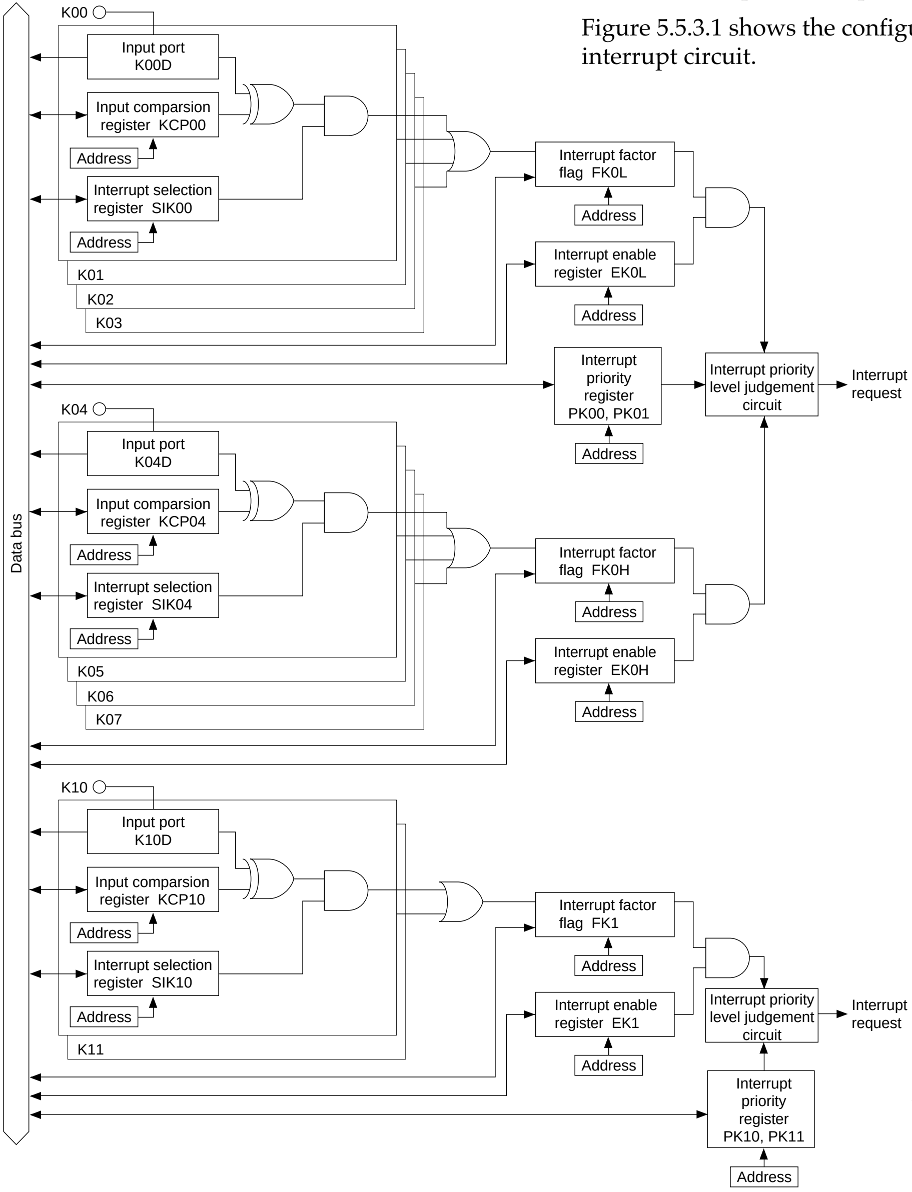
Furthermore, the priority level for input interrupt can be set at the desired level (0–3) using the interrupt priority registers PK00–PK01 and PK10–PK11 corresponding to each of two groups K0x (K00–K07) and K1x (K10–K11).

For details on the interrupt control registers for the above and on operations subsequent to interrupt generation, see "5.15 Interrupt and Standby Status".

The exception processing vectors for each interrupt factor are set as follows:

K10 and K11 input interrupt:	00000AH
K04–K07 input interrupt:	00000CH
K00–K03 input interrupt:	00000EH

Figure 5.5.3.1 shows the configuration of the input interrupt circuit.



K00
Input port K00D
Input comparsion register KCP00
Address
Interrupt selection register SIK00
Address
K01
K02
K03
Interrupt factor flag FK0L
Address
Interrupt enable register EK0L
Address
Interrupt priority register PK00, PK01
Address
Interrupt priority level judgement circuit
Interrupt request
K04
Input port K04D
Input comparsion register KCP04
Address
Interrupt selection register SIK04
Address
K05
K06
K07
Interrupt factor flag FK0H
Address
Interrupt enable register EK0H
Address
Data bus
K10
Input port K10D
Input comparsion register KCP10
Address
Interrupt selection register SIK10
Address
K11
Interrupt factor flag FK1
Address
Interrupt enable register EK1
Address
Interrupt priority level judgement circuit
Interrupt request
Interrupt priority register PK10, PK11
Address

Fig. 5.5.3.1
Configuration of input interrupt circuit

The interrupt selection registers SIK00–SIK03, SIK04–SIK07 and SIK10–SIK11 and input comparison registers KCP00–KCP03, KCP04–KCP07 and KCP10–KCP11 for each port are used to set the interrupt generation condition described above.

Input port interrupt can be permitted or prohibited by the setting of the interrupt selection register SIK. In contrast to the interrupt enable register EK which masks the interrupt factor for each series of terminals, the interrupt selection register SIK is masks the bit units.

The input comparison register KCP selects whether the interrupt for each input port will be generated on the rising edge or the falling edge of input.

When the data content of the input terminals in which interrupt has been permitted by the interrupt selection register SIK and the data content of the input comparison register KCP change from a conformity state to a non-conformity state, the interrupt factor flag FK should be set to "1" and an interrupt is generated.

Figure 5.5.3.2 shows an example of interrupt generation in the series of terminals K0L (K00–K03).

Because interrupt has been prohibited for K00 by the interrupt selection register SIK00, with the settings as shown in (2), an interrupt will not be generated.
Since K03 is "0" in the next settings (3) in the figure, the non-conformity between the input terminal data K01–K03 where interrupt is permitted and the data from the input comparison registers KCP01–KCP03 generates an interrupt.
In line with the explanation above, since the change in the contents of input data and input comparison registers KCP from a conformity state to a non-conformity state introduces an interrupt generation condition, switching from one non-conformity state to another, as is the case in (4) in the figure, will not generate an interrupt. Consequently, in order to be able to generate a second interrupt, either the input terminal must be returned to a state where its content is once again in conformity with that of the input comparison register KCP, or the input comparison register KCP must be reset.
Input terminals for which interrupt is prohibited will not influence an interrupt generation condition.

Interrupt is generated in exactly the same way in the other two series of terminals K0H (K04–K07) and K1 (K10 and K11).

Interrupt selection register

SIK03	SIK02	SIK01	SIK00
1	1	1	0

Input comparison register

KCP03	KCP02	KCP01	KCP00
1	0	1	0

With the settings shown above, interrupt of K0L (K00–K03) is generated under the condition shown below.

Input port

(1)

K03	K02	K01	K00
1	0	1	0

(Initial values)

↓

(2)

K03	K02	K01	K00
1	0	1	1

↓

(3)

K03	K02	K01	K00
0	0	1	1

→ Interrupt generation
Because interrupt has been prohibited for K00, interrupt will be generated when non-conformity occurs between the contents of the three bits K01–K03 and the three bits input comparison register KCP01–KCP03.

↓

(4)

K03	K02	K01	K00
0	1	1	1

Fig. 5.5.3.2 Interrupt generation example in K0L (K00–K03)

5.5.4 Control of input ports

Table 5.5.4.1 shows the input port control bits.

Table 5.5.4.1(a) Input port control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF50	D7	SIK07	K07 interrupt selection register	Interrupt enable	Interrupt disable	0	R/W	
	D6	SIK06	K06 interrupt selection register					
	D5	SIK05	K05 interrupt selection register					
	D4	SIK04	K04 interrupt selection register					
	D3	SIK03	K03 interrupt selection register					
	D2	SIK02	K02 interrupt selection register					
	D1	SIK01	K01 interrupt selection register					
	D0	SIK00	K00 interrupt selection register					
00FF51	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	–	–	–	–	–		
	D2	–	–	–	–	–		
	D1	SIK11	K11 interrupt selection register	Interrupt enable	Interrupt disable	0	R/W	
	D0	SIK10	K10 interrupt selection register					
00FF52	D7	KCP07	K07 interrupt comparison register	Interrupt generated at falling edge	Interrupt generated at rising edge	1	R/W	
	D6	KCP06	K06 interrupt comparison register					
	D5	KCP05	K05 interrupt comparison register					
	D4	KCP04	K04 interrupt comparison register					
	D3	KCP03	K03 interrupt comparison register					
	D2	KCP02	K02 interrupt comparison register					
	D1	KCP01	K01 interrupt comparison register					
	D0	KCP00	K00 interrupt comparison register					
00FF53	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	–	–	–	–	–		
	D2	–	–	–	–	–		
	D1	KCP11	K11 interrupt comparison register	Falling edge	Rising edge	1	R/W	
	D0	KCP10	K10 interrupt comparison register					
00FF54	D7	K07D	K07 input port data	High level input	Low level input	–	R	
	D6	K06D	K06 input port data					
	D5	K05D	K05 input port data					
	D4	K04D	K04 input port data					
	D3	K03D	K03 input port data					
	D2	K02D	K02 input port data					
	D1	K01D	K01 input port data					
	D0	K00D	K00 input port data					
00FF55	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	_		
	D4	–	–	–	–	_		
	D3	–	–	–	–	_		
	D2	–	–	–	–	_		
	D1	K11D	K11 input port data	High level input	Low level input	–	R	
	D0	K10D	K10 input port data					

Table 5.5.4.1(b) Input port control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF20	D7	PK01	K00–K07 interrupt priority register	PK01 PK00 PSIF1 PSIF0 PSW1 PSW0 Priority PTM1 PTM0 level 1 1 Level 3 1 0 Level 2 0 1 Level 1 0 0 Level 0		0	R/W	
	D6	PK00						
	D5	PSIF1	Serial interface interrupt priority register			0	R/W	
	D4	PSIF0						
	D3	PSW1	Stopwatch timer interrupt priority register			0	R/W	
	D2	PSW0						
	D1	PTM1	Clock timer interrupt priority register			0	R/W	
	D0	PTM0						
00FF21	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	PPT1	Programmable timer interrupt priority register	PPT1 PPT0 Priority PK11 PK10 level 1 1 Level 3 1 0 Level 2 0 1 Level 1 0 0 Level 0		0	R/W	
	D2	PPT0						
	D1	PK11	K10 and K11 interrupt priority register			0	R/W	
	D0	PK10						
00FF23	D7	EPT1	Programmable timer 1 interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D6	EPT0	Programmable timer 0 interrupt enable register					
	D5	EK1	K10 and K11 interrupt enable register					
	D4	EK0H	K04–K07 interrupt enable register					
	D3	EK0L	K00–K03 interrupt enable register					
	D2	ESERR	Serial I/F (error) interrupt enable register					
	D1	ESREC	Serial I/F (receiving) interrupt enable register					
	D0	ESTRA	Serial I/F (transmitting) interrupt enable register					
00FF25	D7	FPT1	Programmable timer 1 interrupt factor flag	(R) Interrupt factor is generated (W) Reset	(R) No interrupt factor is generated (W) No operation	0	R/W	
	D6	FPT0	Programmable timer 0 interrupt factor flag					
	D5	FK1	K10 and K11 interrupt factor flag					
	D4	FK0H	K04–K07 interrupt factor flag					
	D3	FK0L	K00–K03 interrupt factor flag					
	D2	FSERR	Serial I/F (error) interrupt factor flag					
	D1	FSREC	Serial I/F (receiving) interrupt factor flag					
	D0	FSTRA	Serial I/F (transmitting) interrupt factor flag					

K00D–K07D: 00FF54H
K10D, K11D: 00FF55H•D0, D1

Input data of input port terminal Kxx can be read out.

When "1" is read:	HIGH level
When "0" is read:	LOW level
Writing:	Invalid

The terminal voltage of each of the input port K00–K07, K10 and K11 can be directly read out as either a "1" for HIGH (VDD) level or a "0" for LOW (VSS) level.
This bit is exclusively for readout and are not usable for write operations.

SIK00–SIK07: 00FF50H
SIK10, SIK11: 00FF51H•D0, D1

Sets the interrupt generation condition (interrupt permission/prohibition) for input port terminals K00–K07, K10 and K11.

When "1" is written:	Interrupt permitted
When "0" is written:	Interrupt prohibited
Reading:	Valid

SIKxx is the interrupt selection register which correspond to the input port Kxx. A "1" setting permits interrupt in that input port and a "0" prohibits it. Changes of state in an input terminal in which interrupt is prohibited, will not influence interrupt generation.
At initial reset, this register is set to "0" (interrupt prohibited).

KCP00–KCP07: 00FF52H
KCP10, KCP11: 00FF53H•D0, D1

Sets the interrupt generation condition (interrupt generation timing) for input port terminals K00–K07, K10 and K11.

When "1" is written: Falling edge
When "0" is written: Rising edge
Reading: Valid

KCPxx is the input comparison register which correspond to the input port Kxx. Interrupt in those ports which have been set to "1" is generated on the falling edge of the input and in those set to "0" on the rising edge.
At initial reset, this register is set to "1" (falling edge).

PK00, PK01: 00FF20H•D6, D7
PK10, PK11: 00FF21H•D0, D1

Sets the input interrupt priority level. The two bits PK00 and PK01 are the interrupt priority registers corresponding to the interrupts for K00–K07 (K0L and K0H). Corresponding to K10–K11 (K1), the two bits PK10 and PK11 perform the same function.
Table 5.5.4.2 shows the interrupt priority level which can be set by this register.

Table 5.5.4.2 Interrupt priority level settings

PK11 PK01	PK10 PK00	Interrupt priority level
1	1	Level 3 ($\overline{\text{IRQ3}}$)
1	0	Level 2 ($\overline{\text{IRQ2}}$)
0	1	Level 1 ($\overline{\text{IRQ1}}$)
0	0	Level 0 (None)

At initial reset, this register is set to "0" (level 0).

EK0L, EK0H, EK1: 00FF23H•D3, D4, D5

How interrupt generation to the CPU is permitted or prohibited.

When "1" is written: Interrupt permitted
When "0" is written: Interrupt prohibited
Reading: Valid

The interrupt enable register EK0L corresponds to K00–K03, EK0H to K04–K07, and EK1 to K10–K11. Interrupt is permitted in those series of terminals set to "1" and prohibited in those set to "0".
At initial reset, this register is set to "0" (interrupt prohibited).

FK0L, FK0H, FK1: 00FF25H•D3, D4, D5

Indicates the generation state for an input interrupt.

When "1" is read: Interrupt factor present
When "0" is read: Interrupt factor not present

When "1" is written: Reset factor flag
When "0" is written: Invalid

The interrupt factor flag FK0L corresponds to K00–K03, FK0H to K04–K07, and FK1 to K10–K11 and they are set to "1" by the occurrence of an interrupt generation condition.
When set in this manner, if the corresponding interrupt enable register is set to "1" and the corresponding interrupt priority register is set to a higher level than the setting of interrupt flags (I0 and I1), an interrupt will be generated to the CPU.
Regardless of the interrupt enable register and interrupt priority register settings, the interrupt factor flag will be set to "1" by the occurrence of an interrupt generation condition.
To accept the subsequent interrupt after interrupt generation, re-setting of the interrupt flags (set interrupt flag to lower level than the level indicated by the interrupt priority registers, or execute the RETE instruction) and interrupt factor flag reset are necessary. The interrupt factor flag is reset to "0" by writing "1".
At initial reset, this flag is all reset to "0".

5.5.5 Programming note

When changing the input terminal from LOW level to HIGH with the built-in pull-up resistor, a delay in the waveform rise time will occur depending on the time constant of the pull-up resistor and the load capacitance of the terminal. It is necessary to set an appropriate wait time for introduction of an input port. In particular, special attention should be paid to key scan for key matrix formation. Make this wait time the amount of time or more calculated by the following expression.

Wait time = RIN x (CIN + load capacitance on the board) x 1.6 [sec]

RIN: Pull up resistance Max. value
CIN: Terminal capacitance Max. value

5.6 Output Ports (R ports)

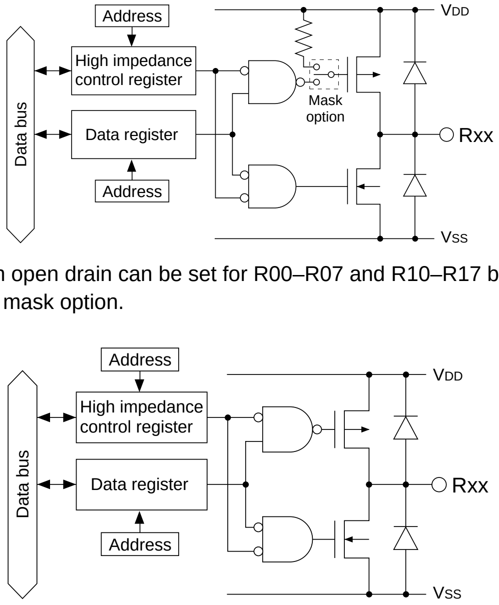
5.6.1 Configuration of output ports

The E0C88112 is equipped with a 34-bit output port (R00–R07, R10–R17, R20–R27, R30–R37, R50, R51). Depending on the bus mode setting, the configuration of the output ports may vary as shown in the table below.

Table 5.6.1.1 Configuration of output ports

Terminal	Bus mode		
	Single chip	Expanded 64K	Expanded 512K
R00	Output port R00	Address A0	
R01	Output port R01	Address A1	
R02	Output port R02	Address A2	
R03	Output port R03	Address A3	
R04	Output port R04	Address A4	
R05	Output port R05	Address A5	
R06	Output port R06	Address A6	
R07	Output port R07	Address A7	
R10	Output port R10	Address A8	
R11	Output port R11	Address A9	
R12	Output port R12	Address A10	
R13	Output port R13	Address A11	
R14	Output port R14	Address A12	
R15	Output port R15	Address A13	
R16	Output port R16	Address A14	
R17	Output port R17	Address A15	
R20	Output port R20		Address A16
R21	Output port R21		Address A17
R22	Output port R22		Address A18
R23	Output port R23	$\overline{\text{RD}}$ signal	
R24	Output port R24	$\overline{\text{WR}}$ signal	
R25	Output port R25		
R26	Output port R26		
R27	Output port R27		
R30	Output port R30	Output port R30/$\overline{\text{CE0}}$ signal	
R31	Output port R31	Output port R31/$\overline{\text{CE1}}$ signal	
R32	Output port R32	Output port R32/$\overline{\text{CE2}}$ signal	
R33	Output port R33	Output port R33/$\overline{\text{CE3}}$ signal	
R34	Output port R34		
R35	Output port R35		
R36	Output port R36		
R37	Output port R37		
R50	Output port R50		
R51	Output port R51	Output port R51/$\overline{\text{BACK}}$ signal	

Only the configuration of the output ports in single chip mode will be discussed here. With respect to bus control, see "5.2 System Controller and Bus Control".
Figure 5.6.1.1 shows the basic structure (excluding special output circuits) of the output ports.



Nch open drain can be set for R00–R07 and R10–R17 by the mask option.

R20–R27, R30–R37, R50, R51

Fig. 5.6.1.1 Structure of output ports

In modes other than single chip mode, the data registers and high impedance control registers of the output ports used for bus function can be used as general purpose registers with read/write capabilities. This will not in any way affect bus signal output.

The output specification of each output port is as complementary output with high impedance control in software possible.
Besides normal DC output, output ports R27, R34, and R50 have a special output function, which can be selected by software.

5.6.2 Mask option

Output ports R00–R07 and R10–R17 output specifications

R00 ..... □ Complementary □ Nch open drain
R01 ..... □ Complementary □ Nch open drain
R02 ..... □ Complementary □ Nch open drain
R03 ..... □ Complementary □ Nch open drain
R04 ..... □ Complementary □ Nch open drain
R05 ..... □ Complementary □ Nch open drain
R06 ..... □ Complementary □ Nch open drain
R07 ..... □ Complementary □ Nch open drain

R10 ..... □ Complementary □ Nch open drain
R11 ..... □ Complementary □ Nch open drain
R12 ..... □ Complementary □ Nch open drain
R13 ..... □ Complementary □ Nch open drain
R14 ..... □ Complementary □ Nch open drain
R15 ..... □ Complementary □ Nch open drain
R16 ..... □ Complementary □ Nch open drain
R17 ..... □ Complementary □ Nch open drain

Output ports R00–R07 and R10–R17 can be used to select output specification for each port (1 bit) by mask option.

The output specification can be selected for either complementary output or Nch open drain output.

Nch open drain output is rendered suitable for purposes as key matrix common output.

For unused input ports, select the default setting of "Complementary".

Note: *When Nch open drain has been selected, voltage in excess of the supply voltage range must not applied to the output port terminal.*

5.6.3 High impedance control

The output port can be high impedance controlled in software.
This makes it possible to share output signal lines with an other external device.

A high impedance control register is set for each series of output port terminals as shown below. Either complementary output and high impedance state can be selected with this register.

Table 5.6.3.1 Correspondence between output ports and high impedance control registers

Register	Output port terminal
HZR0L	R00–R03
HZR0H	R04–R07
HZR1L	R10–R13
HZR1H	R14–R17
HZR20	R20
HZR21	R21
HZR22	R22
HZR23	R23
HZR24	R24
HZR25	R25
HZR26	R26
HZR27	R27
HZR30	R30
HZR31	R31
HZR32	R32
HZR33	R33
HZR34	R34
HZR35	R35
HZR36	R36
HZR37	R37
HZR4L *1	–
HZR4H *1	–
HZR50	R50
HZR51	R51

*1 This is a 2-bit reserved register, it can be used as a general purpose register with read/write capabilities.

When a high impedance control register HZRxx is set to "1", the corresponding output port terminal becomes high impedance state and when set to "0", it becomes complementary output.

5.6.4 DC output

As Figure 5.6.1.1 shows, when "1" is written to the output port data register, the output terminal switches to HIGH (VDD) level and when "0" is written it switches to LOW (VSS) level. When output is in a high impedance state, the data written to the data register is output from the terminal at the instant when output is switched to complementary.

5.6.5 Special output

Besides normal DC output, output ports R27, R34 and R50 can also be assigned special output functions in software as shown in Table 5.6.5.1.

Table 5.6.5.1 Special output ports

Output port	Special output
R27	TOUT output
R34	FOUT output
R50	BZ output

■ TOUT output (R27)

In order for the E0C88112 to provide clock signal to an external device, the output port terminal R27 can be used to output a TOUT signal (clock output by the programmable timer). The configuration of output port R27 is shown in Figure 5.6.5.1.

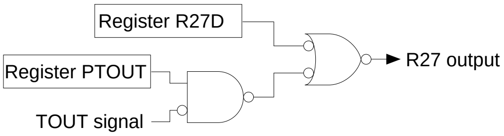


Fig. 5.6.5.1 Configuration of R27

The output control for the TOUT signal is done by the register PTOUT. When you set "1" for the PTOUT, the TOUT signal is output from the output port terminal R27, when "0" is set, the HIGH (VDD) level is output. At this time, "1" must always be set for the data register R27D.

The TOUT signal is the programmable timer underflow divided by 1/2.

With respect to frequency control, see "5.11 Programmable Timer".

Since the TOUT signal is generated asynchronously from the register PTOUT, when the signal is turned ON or OFF by the register settings, a hazard of a 1/2 cycle or less is generated.

Figure 5.6.5.2 shows the output waveform of the TOUT signal.

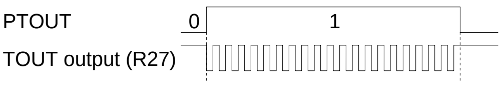


Fig. 5.6.5.2 Output waveform of TOUT signal

■ FOUT output (R34)

In order for the E0C88112 to provide clock signal to an external device, a FOUT signal (oscillation clock fOSC1 or fOSC3 dividing clock) can be output from the output port terminal R34.

Figure 5.6.5.3 shows the configuration of output port R34.

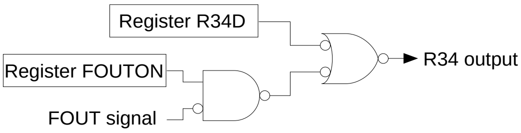


Fig. 5.6.5.3 Configuration of R34

The output control for the FOUT signal is done by the register FOUTON. When you set "1" for the FOUTON, the FOUT signal is output from the output port terminal R34, when "0" is set, the HIGH (VDD) level is output. At this time, "1" must always be set for the data register R34D.

The frequency of the FOUT signal can be selected in software by setting the registers FOUT0–FOUT2. The frequency is selected any one from among eight settings as shown in Table 5.6.5.2.

Table 5.6.5.2 FOUT frequency setting

FOUT2	FOUT1	FOUT0	FOUT frequency
0	0	0	fOSC1 / 1
0	0	1	fOSC1 / 2
0	1	0	fOSC1 / 4
0	1	1	fOSC1 / 8
1	0	0	fOSC3 / 1
1	0	1	fOSC3 / 2
1	1	0	fOSC3 / 4
1	1	1	fOSC3 / 8

fOSC1: OSC1 oscillation frequency
fOSC3: OSC3 oscillation frequency

When the FOUT frequency is made "fOSC3/n", you must turn on the OSC3 oscillation circuit before outputting FOUT. A time interval of several msec to several 10 msec, from the turning ON of the OSC3 oscillation circuit to until the oscillation stabilizes, is necessary, due to the oscillation element that is used. Consequently, if an abnormality occurs as the result of an unstable FOUT signal being output externally, you should allow an adequate waiting time after turning ON of the OSC3 oscillation, before turning outputting FOUT. (The oscillation start time will vary somewhat depending on the oscillator and on the externally attached parts. Refer to the oscillation start time example indicated in Chapter 7, "ELECTRICAL CHARACTERISTICS".)

At initial reset, OSC3 oscillation circuit is set to OFF state.

Since the FOUT signal is generated asynchronously from the register FOUTON, when the signal is turned ON or OFF by the register settings, a hazard of a 1/2 cycle or less is generated.
Figure 5.6.5.4 shows the output waveform of the FOUT signal.

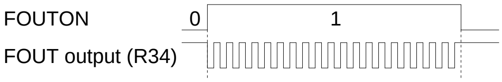


Fig. 5.6.5.4 Output waveform of FOUT signal

■ BZ output (R50)

In order for the E0C88112 to drive an external buzzer, a BZ signal (sound generator output) can be output from the output port terminal R50.
The configuration of the output port R50 is shown in Figure 5.6.5.5.

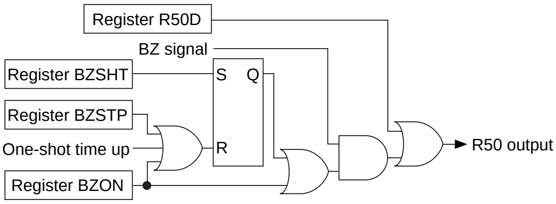


Fig. 5.6.5.5 Configuration of R50

The output control for the BZ signal is done by the registers BZON, BZSHT and BZSTP. When you set "1" for the BZON or BZSHT, the BZ signal is output from the output port terminal R50, when "0" is set for the BZON or "1" is set for the BZSTP, the LOW (VSS) level is output. At this time, "0" must always be set for the data register R50D.

The BZ signal which is output makes use of the output of the sound generator. With respect to control of frequency and envelope, see "5.13 Sound Generator".
Since the BZ signal is generated asynchronously from the registers BZONB, BZSHT and BZSTP, when the signal is turned ON or OFF by the register settings, a hazard of a 1/2 cycle or less is generated.
Figure 5.6.5.6 shows the output waveform of the BZ signal.

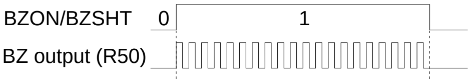


Fig. 5.6.5.6 Output waveform of BZ signal

5.6.6 Control of output ports

Table 5.6.6.1 shows the output port control bits.

Table 5.6.6.1(a) Output port control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF70	D7	HZR51	R51 high impedance control	High impedance	Comple-mentary	0	R/W	
	D6	HZR50	R50 high impedance control					
	D5	HZR4H	R/W register	1	0	0	R/W	Reserved register
	D4	HZR4L	R/W register					
	D3	HZR1H	R14–R17 high impedance control	High impedance	Comple-mentary	0	R/W	
	D2	HZR1L	R10–R13 high impedance control					
	D1	HZR0H	R04–R07 high impedance control					
	D0	HZR0L	R00–R03 high impedance control					
00FF71	D7	HZR27	R27 high impedance control	High impedance	Comple-mentary	0	R/W	
	D6	HZR26	R26 high impedance control					
	D5	HZR25	R25 high impedance control					
	D4	HZR24	R24 high impedance control					
	D3	HZR23	R23 high impedance control					
	D2	HZR22	R22 high impedance control					
	D1	HZR21	R21 high impedance control					
	D0	HZR20	R20 high impedance control					
00FF72	D7	HZR37	R37 high impedance control	High impedance	Comple-mentary	0	R/W	
	D6	HZR36	R36 high impedance control					
	D5	HZR35	R35 high impedance control					
	D4	HZR34	R34 high impedance control					
	D3	HZR33	R33 high impedance control					
	D2	HZR32	R32 high impedance control					
	D1	HZR31	R31 high impedance control					
	D0	HZR30	R30 high impedance control					
00FF73	D7	R07D	R07 output port data	High	Low	1	R/W	
	D6	R06D	R06 output port data					
	D5	R05D	R05 output port data					
	D4	R04D	R04 output port data					
	D3	R03D	R03 output port data					
	D2	R02D	R02 output port data					
	D1	R01D	R01 output port data					
	D0	R00D	R00 output port data					
00FF74	D7	R17D	R17 output port data	High	Low	1	R/W	
	D6	R16D	R16 output port data					
	D5	R15D	R15 output port data					
	D4	R14D	R14 output port data					
	D3	R13D	R13 output port data					
	D2	R12D	R12 output port data					
	D1	R11D	R11 output port data					
	D0	R10D	R10 output port data					
00FF75	D7	R27D	R27 output port data	High	Low	1	R/W	
	D6	R26D	R26 output port data					
	D5	R25D	R25 output port data					
	D4	R24D	R24 output port data					
	D3	R23D	R23 output port data					
	D2	R22D	R22 output port data					
	D1	R21D	R21 output port data					
	D0	R20D	R20 output port data					

Table 5.6.6.1(b) Output port control bits

Address	Bit	Name	Function		1	0	SR	R/W	Comment
00FF76	D7	R37D	R37 output port data		High	Low	1	R/W	
	D6	R36D	R36 output port data						
	D5	R35D	R35 output port data						
	D4	R34D	R34 output port data						
	D3	R33D	R33 output port data						
	D2	R32D	R32 output port data						
	D1	R31D	R31 output port data						
	D0	R30D	R30 output port data						
00FF77	D7	R47D	R/W register		1	0	1	R/W	Reserved register
	D6	R46D	R/W register						
	D5	R45D	R/W register						
	D4	R44D	R/W register						
	D3	R43D	R/W register						
	D2	R42D	R/W register						
	D1	R41D	R/W register						
	D0	R40D	R/W register						
00FF78	D7	–	–		–	–	–		Constantly "0" when being read
	D6	–	–		–	–	–		
	D5	–	–		–	–	–		
	D4	–	–		–	–	–		
	D3	–	–		–	–	–		
	D2	–	–		–	–	–		
	D1	R51D	R51 output port data		High	Low	1	R/W	
	D0	R50D	R50 output port data		High	Low	0	R/W	
00FF30	D7	–	–		–	–	–		Constantry "0" when being read
	D6	–	–		–	–	–		
	D5	–	–		–	–	–		
	D4	MODE16	8/16-bit mode selection		16-bit x 1	8-bit x 2	0	R/W	
	D3	CHSEL	TOUT output channel selection		Timer 1	Timer 0	0	R/W	
	D2	PTOUT	TOUT output control		On	Off	0	R/W	
	D1	CKSEL1	Prescaler 1 source clock selection		f_{OSC3}	f_{OSC1}	0	R/W	
	D0	CKSEL0	Prescaler 0 source clock selection		f_{OSC3}	f_{OSC1}	0	R/W	
00FF44	D7	–	–		–	–	–		Constantry "0" when being read
	D6	BZSTP	One-shot buzzer forcibly stop		Forcibly stop	No operation	–	W	
	D5	BZSHT	One-shot buzzer trigger/status	R	Busy	Ready	0	R/W	
				W	Trigger	No operation			
	D4	SHTPW	One-shot buzzer duration width selection		125 msec	31.25 msec	0	R/W	
	D3	ENRTM	Envelope attenuation time		1 sec	0.5 sec	0	R/W	
	D2	ENRST	Envelope reset		Reset	No operation	–	W	"0" when being read
	D1	ENON	Envelope On/Off control		On	Off	0	R/W	*1
	D0	BZON	Buzzer output control		On	Off	0	R/W	

*1 Reset to "0" during one-shot output.

Table 5.6.6.1(c) Output port control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF40	D7	–	–	–	–	–		"0" when being read
	D6	FOUT2	FOUT frequency selection FOUT2 FOUT1 FOUT0 Frequency 0 0 0 f_{OSC1} / 1 0 0 1 f_{OSC1} / 2 0 1 0 f_{OSC1} / 4 0 1 1 f_{OSC1} / 8 1 0 0 f_{OSC3} / 1 1 0 1 f_{OSC3} / 2 1 1 0 f_{OSC3} / 4 1 1 1 f_{OSC3} / 8			0	R/W	
	D5	FOUT1				0	R/W	
	D4	FOUT0				0	R/W	
	D3	FOUTON	FOUT output control	On	Off	0	R/W	
	D2	WDRST	Watchdog timer reset	Reset	No operation	–	W	Constantly "0" when being read
	D1	TMRST	Clock timer reset	Reset	No operation	–	W	
	D0	TMRUN	Clock timer Run/Stop control	Run	Stop	0	R/W	

■ High impedance control

HZR0L, HZR0H: 00FF70H•D0, D1
HZR1L, HZR1H: 00FF70H•D2, D3
HZR20–HZR27: 00FF71H
HZR30–HZR37: 00FF72H
HZR4L, HZR4H: 00FF70H•D4, D5 *1
HZR50, HZR51: 00FF70H•D6, D7

Sets the output terminals to a high impedance state.

When "1" is written:	High impedance
When "0" is written:	Complementary
Reading:	Valid

HZRxx is the high impedance control register which correspond as shown in Table 5.6.3.1 to the various output port terminals.
When "1" is set to the HZRxx register, the corresponding output port terminal becomes high impedance state and when "0" is set, it becomes complementary output.
At initial reset, this register is set to "0" (complimentary).

*1 HZR4L and HZR4H is 2-bit reserved register, it can be used as a general purpose register with read / write capabilities.

■ DC output control

R00D–R07D: 00FF73H
R10D–R17D: 00FF74H
R20D–R27D: 00FF75H
R30D–R37D: 00FF76H
R40D–R47D: 00FF77H *1
R50D, R51D: 00FF78H•D0, D1

Sets the data output from the output port terminal Rxx.

When "1" is written:	HIGH level output
When "0" is written:	LOW level output
Reading:	Valid

RxxD is the data register for each output port.
When "1" is set, the corresponding output port terminal switches to HIGH (V_{DD}) level, and when "0" is set, it switches to LOW (V_{SS}) level.
At initial reset, R50D is set to "0" (LOW level output), all other registers are set to "1" (HIGH level output).
The output data registers set for bus signal output can be used as general purpose registers with read / write capabilities which do not affect the output terminals.

*1 R40D–R47D is 8-bit reserved register, it can be used as a general purpose register with read / write capabilities.

■ Special output control

PTOUT: 00FF30H•D2

Controls the TOUT (programmable timer output clock) signal output.

When "1" is written: TOUT signal output
When "0" is written: HIGH level (DC) output
Reading: Valid

PTOUT is the output control register for TOUT signal. When "1" is set, the TOUT signal is output from the output port terminal R27 and when "0" is set, HIGH (V_{DD}) level is output. At this time, "1" must always be set for the data register R27D.
At initial reset, PTOUT is set to "0" (HIGH level output).

FOUTON: 00FF40H•D3

Controls the FOUT (f_{OSC1}/f_{OSC3} dividing clock) signal output.

When "1" is written: FOUT signal output
When "0" is written: HIGH level (DC) output
Reading: Valid

FOUTON is the output control register for FOUT signal. When "1" is set, the FOUT signal is output from the output port terminal R34 and when "0" is set, HIGH (V_{DD}) level is output. At this time, "1" must always be set for the data register R34D.
At initial reset, FOUTON is set to "0" (HIGH level output).

FOUT0, FOUT1, FOUT2: 00FF40H•D4, D5, D6

FOUT signal frequency is set as shown in Table 5.6.6.2.

Table 5.6.6.2 FOUT frequency settings

FOUT2	FOUT1	FOUT0	FOUT frequency
0	0	0	f_{OSC1} / 1
0	0	1	f_{OSC1} / 2
0	1	0	f_{OSC1} / 4
0	1	1	f_{OSC1} / 8
1	0	0	f_{OSC3} / 1
1	0	1	f_{OSC3} / 2
1	1	0	f_{OSC3} / 4
1	1	1	f_{OSC3} / 8

f_{OSC1}: OSC1 oscillation frequency
f_{OSC3}: OSC3 oscillation frequency

At initial reset, this register is set to "0" (f_{OSC1}/1).

BZON: 00FF44H•D0

Controls the BZ (buzzer) signal output.

When "1" is written: BZ signal output
When "0" is written: LOW level (DC) output
Reading: Valid

BZON is the output control register for BZ signal. When "1" is set, the BZ signal is output from the output port terminal R50 and when "0" is set, LOW (V_{SS}) level is output. At this time, "0" must always be set for the data register R50D.
At initial reset, BZON is set to "0" (LOW level output).

BZSHT: 00FF45H•D5

Controls the one-shot buzzer output.

When "1" is written: Trigger
When "0" is written: No operation

When "1" is read: Busy
When "0" is read: Ready

Writing "1" into BZSHT causes the one-shot output circuit to operate and the BZ signal to be output. The buzzer output is automatically turned OFF after the time set by SHTPW has elapsed. At this time, "0" must always be set for the data register R50D.
The one-shot output is only valid when the normal buzzer output is OFF (BZON = "0") state. The trigger is invalid during ON (BZON = "1") state.
When a re-trigger is assigned during a one-shot output, the one-shot output time set with SHTPW is measured again from that point. (time extension)
The operation status of the one-shot output circuit can be confirmed by reading BZSHT, when the one-shot output is ON, BZSHT reads "1" and when the output is OFF, it reads "0".
At initial reset, BZSHT is set to "0" (Ready).

BZSTP: 00FF45H•D6

Forcibly stops the one-shot buzzer output.

When "1" is written: Forcibly stop
When "0" is written: No operation
Reading: Constantly "0"

By writing "1" into BZSTP, the one-shot buzzer output can be stopped prior to the elapsing of the time set with SHTPW.
Writing "0" is invalid and writing "1" except during one-shot output is also invalid.
When "1" is written to BZSHT and BZSTP simultaneously, BZSTP takes precedence and one-shot output becomes stop status.
Since BZSTP is for writing only, during readout it is constantly set to "0".

5.6.7 Programming notes

(1) Since the special output signals (TOUT, FOUT and BZ) are generated asynchronously from the output control registers (PTOUT, FOUTON, BZON, BZSHT and BZSTP), when the signals is turned ON or OFF by the output control register settings, a hazard of a 1/2 cycle or less is generated.

(2) When the FOUT frequency is made "f_{OSC3}/n", you must turn on the OSC3 oscillation circuit before outputting FOUT. A time interval of several msec to several 10 msec, from the turning ON of the OSC3 oscillation circuit to until the oscillation stabilizes, is necessary, due to the oscillation element that is used. Consequently, if an abnormality occurs as the result of an unstable FOUT signal being output externally, you should allow an adequate waiting time after turning ON of the OSC3 oscillation, before turning outputting FOUT. (The oscillation start time will vary somewhat depending on the oscillator and on the externally attached parts. Refer to the oscillation start time example indicated in Chapter 7, "ELECTRICAL CHARACTERISTICS".)
At initial reset, OSC3 oscillation circuit is set to OFF state.

(3) The SLP instruction has executed when the special output signals (TOUT, FOUT and BZ) are in the enable status, an unstable clock is output for the special output at the time of return from the SLEEP state. Consequently, when shifting to the SLEEP state, you should set the special output signal to the disable status prior to executing the SLP instruction.

5.7 I/O Ports (P ports)

5.7.1 Configuration of I/O ports

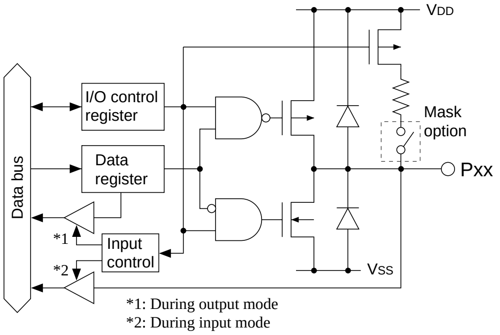
The E0C88112 is equipped with 16 bits of I/O ports (P00–P07, P10–P17). The configuration of these I/O ports will vary according to the bus mode as shown below.

Table 5.7.1.1 Configuration of I/O ports

Terminal	Bus mode		
	Single chip	Expanded 64K	Expanded 512K
P00	I/O port P00	Data bus D0	
P01	I/O port P01	Data bus D1	
P02	I/O port P02	Data bus D2	
P03	I/O port P03	Data bus D3	
P04	I/O port P04	Data bus D4	
P05	I/O port P05	Data bus D5	
P06	I/O port P06	Data bus D6	
P07	I/O port P07	Data bus D7	
P10	I/O port P10 (SIN)		
P11	I/O port P11 (SOUT)		
P12	I/O port P12 ($\overline{\text{SCLK}}$)		
P13	I/O port P13 ($\overline{\text{SRDY}}$)		
P14	I/O port P14 (CMPP0)		
P15	I/O port P15 (CMPM0)		
P16	I/O port P16 (CMPP1)		
P17	I/O port P17 (CMPM1)		

With respect to the data bus, see "5.2 System Controller and Bus Control".

Figure 5.7.1.1 shows the structure of an I/O port.



*1: During output mode
*2: During input mode

Fig. 5.7.1.1 Structure of I/O port

I/O port can be set for input or output mode in one bit unit. These settings are performed by writing data to the I/O control registers.

I/O port terminals P10–P13 and P14–P17 are shared with serial interface input/output terminal and analog comparator input terminals, respectively. The function of each terminals is switchable in software. With respect to serial interface and analog comparator, see "5.8 Serial Interface" and "5.13 Analog Comparator", respectively.

The data registers and I/O control registers of I/O ports set for data bus and serial interface output terminals use are usable as general purpose registers with read/write capabilities which do not affect I/O activities of the terminal.

The same as above, the I/O control register of I/O port set for serial interface input terminal use is usable as general purpose register.

5.7.2 Mask option

I/O port pull-up resistors

P00	□ With resistor	□ Gate direct
P01	□ With resistor	□ Gate direct
P02	□ With resistor	□ Gate direct
P03	□ With resistor	□ Gate direct
P04	□ With resistor	□ Gate direct
P05	□ With resistor	□ Gate direct
P06	□ With resistor	□ Gate direct
P07	□ With resistor	□ Gate direct
P10	□ With resistor	□ Gate direct
P11	□ With resistor	□ Gate direct
P12	□ With resistor	□ Gate direct
P13	□ With resistor	□ Gate direct
P14	□ With resistor	□ Gate direct
P15	□ With resistor	□ Gate direct
P16	□ With resistor	□ Gate direct
P17	□ With resistor	□ Gate direct

I/O ports P00–P07 and P10–P17 are equipped with a pull-up resistor which goes ON in the input mode. Whether this resistor is used or not can be selected for each port (one bit unit).

In cases where the 'With resistor' option is selected, the pull-up resistor goes ON when the port is in input mode.

When changing the port terminal from LOW level to HIGH with the built-in pull-up resistor, a delay in the waveform rise time will occur depending on the time constant of the pull-up resistor and the load capacitance of the terminal. It is necessary to set an appropriate wait time for introduction of an I/O port. Make this wait time the amount of time or more calculated by the following expression.

Wait time = RIN x (CIN + load capacitance on the board) x 1.6 [sec]

RIN: Pull up resistance Max. value
CIN: Terminal capacitance Max. value

When the analog comparator is used, select "Gate direct" for I/O ports (P14–P15 or P16–P17, or both) which then become input terminals.

For unused I/O ports, select the default setting of "With resistor".

5.7.3 I/O control registers and I/O mode

I/O ports P00–P07 and P10–P17 are set either to input or output modes by writing data to the I/O control registers IOC00–IOC07 and IOC10–IOC17 which correspond to each bit.

To set an I/O port to input mode, write "0" to the I/O control register.

An I/O port which is set to input mode will shift to a high impedance state and functions as an input port. Readout in input mode consists simply of a direct readout of the input terminal state: the data being "1" when the input terminal is at HIGH (VDD) level and "0" when it is at LOW (VSS) level.

When the "With resistor" option is selected using the mask option, the resistor is pulled up onto the port terminal in input mode.

Even in input mode, data can be written to the data registers without affecting the terminal state.

To set an I/O port to output mode, write "1" to the I/O control register. An I/O port which is set to output mode functions as an output port.

When port output data is "1", a HIGH (VDD) level is output and when it is "0", a LOW (VSS) level is output. Readout in output mode consists of the contents of the data register.

At initial reset, I/O control registers are set to "0" (I/O ports are set to input mode).

5.7.4 Control of I/O ports

Table 5.7.4.1 shows the I/O port control bits.

Table 5.7.4.1 I/O port control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF60	D7	IOC07	P07 I/O control register	Output	Input	0	R/W	
	D6	IOC06	P06 I/O control register					
	D5	IOC05	P05 I/O control register					
	D4	IOC04	P04 I/O control register					
	D3	IOC03	P03 I/O control register					
	D2	IOC02	P02 I/O control register					
	D1	IOC01	P01 I/O control register					
	D0	IOC00	P00 I/O control register					
00FF61	D7	IOC17	P17 I/O control register	Output	Input	0	R/W	
	D6	IOC16	P16 I/O control register					
	D5	IOC15	P15 I/O control register					
	D4	IOC14	P14 I/O control register					
	D3	IOC13	P13 I/O control register					
	D2	IOC12	P12 I/O control register					
	D1	IOC11	P11 I/O control register					
	D0	IOC10	P10 I/O control register					
00FF62	D7	P07D	P07 I/O port data	High	Low	1	R/W	
	D6	P06D	P06 I/O port data					
	D5	P05D	P05 I/O port data					
	D4	P04D	P04 I/O port data					
	D3	P03D	P03 I/O port data					
	D2	P02D	P02 I/O port data					
	D1	P01D	P01 I/O port data					
	D0	P00D	P00 I/O port data					
00FF63	D7	P17D	P17 I/O port data	High	Low	1	R/W	
	D6	P16D	P16 I/O port data					
	D5	P15D	P15 I/O port data					
	D4	P14D	P14 I/O port data					
	D3	P13D	P13 I/O port data					
	D2	P12D	P12 I/O port data					
	D1	P11D	P11 I/O port data					
	D0	P10D	P10 I/O port data					

P00D–P07D, P10D–P17D: 00FF62H, 00FF63H

How I/O port terminal Pxx data readout and output data settings are performed.

When writing data:

When "1" is written:	HIGH level
When "0" is written:	LOW level

When the I/O port is set to output mode, the data written is output as is to the I/O port terminal. In terms of port data, when "1" is written, the port terminal goes to HIGH (VDD) level and when "0" is written to a LOW (VSS) level.
Even when the port is in input mode, data can still be written in.

When reading out data:

When "1" is read:	HIGH level ("1")
When "0" is read:	LOW level ("0")

When an I/O port is in input mode, the voltage level being input to the port terminal is read out. When terminal voltage is HIGH (VDD), it is read as a "1", and when it is LOW (VSS), it is read as a "0".
Furthermore, in output mode, the contents of the data register are read out.
At initial reset, this register is set to "1" (HIGH level).

Note: The data registers of I/O ports set for the data bus and output terminal of serial interface can be used as general purpose registers with read/write capabilities which do not affect I/O activities of the terminals.

IOC00–IOC07: 00FF60H
IOC10–IOC17: 00FF61H

Sets the I/O ports to input or output mode.

When "1" is written:	Output mode
When "0" is written:	Input mode
Reading:	Valid

IOCxx is the I/O control register which correspond to each I/O port in a bit unit.
Writing "1" to the IOCxx register will switch the corresponding I/O port Pxx to output mode, and writing "0" will switch it to input mode.
When the analog comparator is used, "0" must always be set for the I/O control registers (IOC14–IOC15 or IOC16–IOC17, or both) of I/O ports which will become input terminals.
At initial reset, this register is set to "0" (input mode).

Note: The data registers of I/O ports set for the data bus and input terminal of serial interface can be used as general purpose registers with read/write capabilities which do not affect I/O activities of the terminals.

5.7.5 Programming notes

(1) When changing the port terminal from LOW level to HIGH with the built-in pull-up resistor, a delay in the waveform rise time will occur depending on the time constant of the pull-up resistor and the load capacitance of the terminal. It is necessary to set an appropriate wait time for introduction of an I/O port. Make this wait time the amount of time or more calculated by the following expression.

Wait time = RIN x (CIN + load capacitance on the board) x 1.6 [sec]

RIN: Pull up resistance Max. value
CIN: Terminal capacitance Max. value

(2) When the analog comparator is used, "0" must always be set for the I/O control registers (IOC14–IOC15 or IOC16–IOC17, or both) of I/O ports which will become input terminals.

5.8 Serial Interface

5.8.1 Configuration of serial interface

The E0C88112 incorporates a full duplex serial interface (when asynchronous system is selected) that allows the user to select either clock synchronous system or asynchronous system.
The data transfer method can be selected in software.
When the clock synchronous system is selected, 8-bit data transfer is possible.
When the asynchronous system is selected, either 7-bit or 8-bit data transfer is possible, and a parity check of received data and the addition of a parity bit for transmitting data can automatically be done by selecting in software.
Figure 5.8.1.1 shows the configuration of the serial interface.

Serial interface input/output terminals, SIN, SOUT, $\overline{\text{SCLK}}$ and $\overline{\text{SRDY}}$ are shared with I/O ports P10–P13. In order to utilize these terminals for the serial interface input/output terminals, proper settings have to be made with registers ESIF, SMD0 and SMD1. (At initial reset, these terminals are set as I/O port terminals.)
The direction of I/O port terminals set for serial interface input/output terminals are determined by the signal and transfer mode for each terminal. Furthermore, the settings for the corresponding I/O control registers for the I/O ports become invalid.

Table 5.8.1.1 Configuration of input/output terminals

Terminal	When serial interface is selected
P10	SIN
P11	SOUT
P12	$\overline{\text{SCLK}}$
P13	$\overline{\text{SRDY}}$

* The terminals used may vary depending on the transfer mode.

SIN and SOUT are serial data input and output terminals which function identically in clock synchronous system and asynchronous system. $\overline{\text{SCLK}}$ is exclusively for use with clock synchronous system and functions as a synchronous clock input/output terminal. $\overline{\text{SRDY}}$ is exclusively for use in clock synchronous slave mode and functions as a send-receive ready signal output terminal.
When asynchronous system is selected, since $\overline{\text{SCLK}}$ and $\overline{\text{SRDY}}$ are superfluous, the I/O port terminals P12 and P13 can be used as I/O ports.
In the same way, when clock synchronous master mode is selected, since $\overline{\text{SRDY}}$ is superfluous, the I/O port terminal P13 can be used as I/O port.

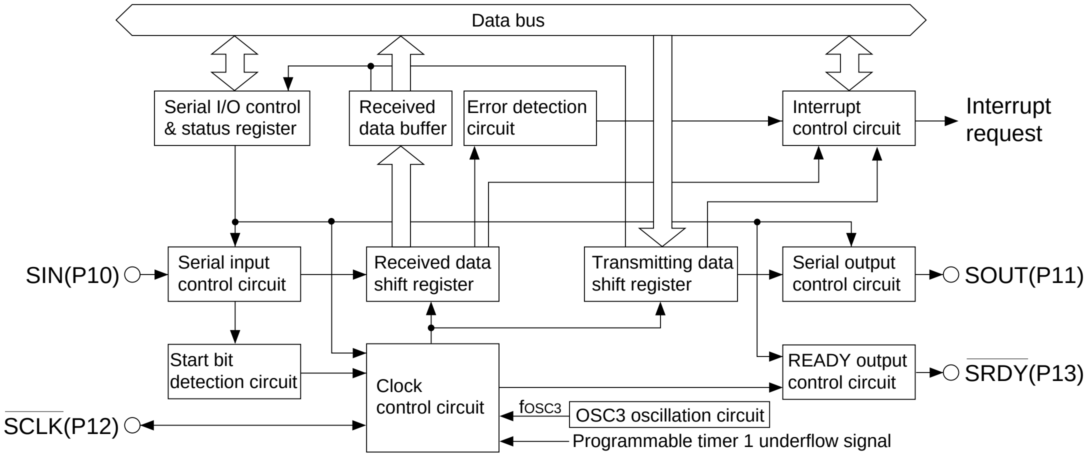

Data bus
Serial I/O control & status register
Received data buffer
Error detection circuit
Interrupt control circuit
Interrupt request
SIN(P10)
Serial input control circuit
Received data shift register
Transmitting data shift register
Serial output control circuit
SOUT(P11)
Start bit detection circuit
Clock control circuit
READY output control circuit
SRDY(P13)
SCLK(P12)
fOSC3
OSC3 oscillation circuit
Programmable timer 1 underflow signal

Fig. 5.8.1.1 Configuration of serial interface

5.8.2 Mask option

Since serial interface input/output terminals are shared with the I/O ports, serial interface terminal specifications have necessarily been selected with the mask option for I/O ports.

I/O port pull-up resistors
P10 (SIN) .... □ With resistor □ Gate direct
P12 (SCLK) .... □ With resistor □ Gate direct

Each I/O port terminal is equipped with a pull-up resistor which goes ON in input mode. A selection can be made for each port (one bit unit) as to whether or not the resistor will be used.
Specifications (whether the pull-up will be used or not) of P10 (SIN) and P12 (SCLK) which will become input terminals when using the serial interface are decided by settings the options for the I/O port.

When "Gate direct" is selected in the serial I/F mode, be sure that the input terminals do not go into a floating state.

5.8.3 Transfer modes

There are four transfer modes for the serial interface and mode selection is made by setting the two bits of the mode selection registers SMD0 and SMD1 as shown in the table below.

Table 5.8.3.1 Transfer modes

SMD1	SMD0	Mode
1	1	Asynchronous 8-bit
1	0	Asynchronous 7-bit
0	1	Clock synchronous slave
0	0	Clock synchronous master

Table 5.8.3.2 Terminal settings corresponding to each transfer mode

Mode	SIN	SOUT	SCLK	SRDY
Asynchronous 8-bit	Input	Output	P12	P13
Asynchronous 7-bit	Input	Output	P12	P13
Clock synchronous slave	Input	Output	Input	Output
Clock synchronous master	Input	Output	Output	P13

At initial reset, transfer mode is set to clock synchronous master mode.

■ Clock synchronous master mode

In this mode, the internal clock is utilized as a synchronous clock for the built-in shift registers, and clock synchronous 8-bit serial transfers can be performed with this serial interface as the master.
The synchronous clock is also output from the SCLK terminal which enables control of the external (slave side) serial I/O device. Since the SRDY terminal is not utilized in this mode, it can be used as an I/O port.
Figure 5.8.3.1(a) shows the connection example of input/output terminals in the clock synchronous master mode.

■ Clock synchronous slave mode

In this mode, a synchronous clock from the external (master side) serial input/output device is utilized and clock synchronous 8-bit serial transfers can be performed with this serial interface as the slave.
The synchronous clock is input to the SCLK terminal and is utilized by this interface as the synchronous clock.
Furthermore, the SRDY signal indicating the transmit-receive ready status is output from the SRDY terminal in accordance with the serial interface operating status.
In the slave mode, the settings for registers SCS0 and SCS1 used to select the clock source are invalid.
Figure 5.8.3.1(b) shows the connection example of input/output terminals in the clock synchronous slave mode.

■ Asynchronous 7-bit mode

In this mode, asynchronous 7-bit transfer can be performed. Parity check during data reception and addition of parity bit (odd/even/none) during transmitting can be specified and data processed in 7 bits with or without parity. Since this mode employs the internal clock, the SCLK terminal is not used. Furthermore, since the SRDY terminal is not utilized either, both of these terminals can be used as I/O ports.
Figure 5.8.3.1(c) shows the connection example of input/output terminals in the asynchronous mode.

■ Asynchronous 8-bit mode

In this mode, asynchronous 8-bit transfer can be performed. Parity check during data reception and addition of parity bit (odd/even/none) during transmitting can be specified and data processed in 8 bits with or without parity. Since this mode employs the internal clock, the SCLK terminal is not used. Furthermore, since the SRDY terminal is not utilized either, both of these terminals can be used as I/O ports.
Figure 5.8.3.1(c) shows the connection example of input/output terminals in the asynchronous mode.

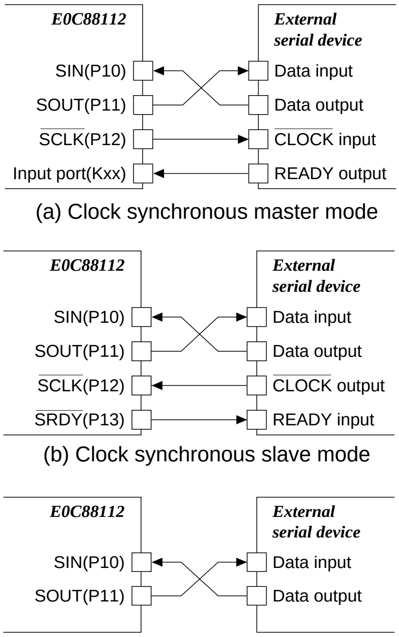

E0C88112
External serial device
SIN(P10)
SOUT(P11)
$\overline{\text{SCLK}}$(P12)
Input port(Kxx)
Data input
Data output
$\overline{\text{CLOCK}}$ input
READY output
(a) Clock synchronous master mode
E0C88112
External serial device
SIN(P10)
SOUT(P11)
$\overline{\text{SCLK}}$(P12)
$\overline{\text{SRDY}}$(P13)
Data input
Data output
$\overline{\text{CLOCK}}$ output
READY input
(b) Clock synchronous slave mode
E0C88112
External serial device
SIN(P10)
SOUT(P11)
Data input
Data output
(c) Asynchronous 7-bit/8-bit mode

Fig. 5.8.3.1 Connection examples of serial interface I/O terminals

5.8.4 Clock source

There are four clock sources and selection is made by setting the two bits of the clock source selection register SCS0 and SCS1 as shown in table below.

Table 5.8.4.1 Clock source

SCS1	SCS0	Clock source
1	1	Programmable timer
1	0	f_{OSC3} / 4
0	1	f_{OSC3} / 8
0	0	f_{OSC3} / 16

This register setting is invalid in clock synchronous slave mode and the external clock input from the $\overline{\text{SCLK}}$ terminal is used.

When the "programmable timer" is selected, the programmable timer 1 underflow signal is divided by 1/2 and this signal used as the clock source. With respect to the transfer rate setting, see "5.11 Programmable Timer".

At initial reset, the synchronous clock is set to "f_{OSC3}/16".

Whichever clock is selected, the signal is further divided by 1/16 and then used as the synchronous clock.

Furthermore, external clock input is used as is for $\overline{\text{SCLK}}$ in clock synchronous slave mode.

Table 5.8.4.2 shows an examples of transfer rates and OSC3 oscillation frequencies when the clock source is set to programmable timer.

When the demultiplied signal of the OSC3 oscillation circuit is made the clock source, it is necessary to turn the OSC3 oscillation ON, prior to using the serial interface.

A time interval of several msec to several 10 msec, from the turning ON of the OSC3 oscillation circuit to until the oscillation stabilizes, is necessary, due to the oscillation element that is used. Consequently, you should allow an adequate waiting time after turning ON of the OSC3 oscillation, before starting transmitting/receiving of serial interface. (The oscillation start time will vary somewhat depending on the oscillator and on the externally attached parts. Refer to the oscillation start time example indicated in Chapter 7, "ELECTRICAL CHARACTERISTICS".)

At initial reset, the OSC3 oscillation circuit is set to OFF status.

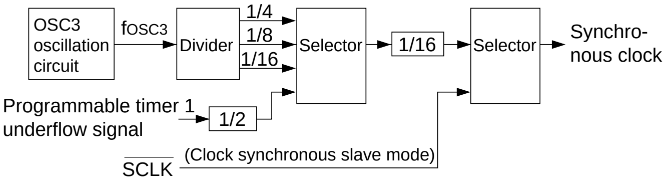


Fig. 5.8.4.1 Division of the synchronous clock

Table 5.8.4.2 OSC3 oscillation frequencies and transfer rates

Transfer rate (bps)	OSC3 oscillation frequency / Programmable timer settings					
	f_{OSC3} = 3.072 MHz		f_{OSC3} = 4.608 MHz		f_{OSC3} = 4.9152 MHz	
	PSC1X	RLD1X	PSC1X	RLD1X	PSC1X	RLD1X
9,600	0 (1/1)	09H	0 (1/1)	0EH	0 (1/1)	0FH
4,800	0 (1/1)	13H	0 (1/1)	1DH	0 (1/1)	1FH
2,400	0 (1/1)	27H	0 (1/1)	3BH	0 (1/1)	3FH
1,200	0 (1/1)	4FH	0 (1/1)	77H	0 (1/1)	7FH
600	0 (1/1)	9FH	0 (1/1)	EFH	0 (1/1)	FFH
300	1 (1/4)	4FH	1 (1/4)	77H	1 (1/4)	7FH
150	1 (1/4)	9FH	1 (1/4)	EFH	1 (1/4)	FFH

5.8.5 Transmit-receive control

Below is a description of the registers which handle transmit-receive control. With respect to transmit-receive control procedures and operations, please refer to the following sections in which these are discussed on a mode by mode basis.

■ Shift register and received data buffer

Exclusive shift registers for transmitting and receiving are installed in this serial interface. Consequently, duplex communication simultaneous transmit and receive is possible when the asynchronous system is selected.

Data being transmitted are written to TRXD0–TRXD7 and converted to serial through the shift register and is output from the SOUT terminal.

In the reception section, a received data buffer is installed separate from the shift register.
Data being received are input to the SIN terminal and is converted to parallel through the shift register and written to the received data buffer.
Since the received data buffer can be read even during serial input operation, the continuous data is received efficiently.
However, since buffer functions are not used in clock synchronous mode, be sure to read out data before the next data reception begins.

■ Transmit enable register and transmit control bit

For transmitting control, use the transmit enable register TXEN and transmit control bit TXTRG.

The transmit enable register TXEN is used to set the transmitting enable/disable status. When "1" is written to this register to set the transmitting enable status, clock input to the shift register is enabled and the system is ready to transmit data. In the clock synchronous mode, synchronous clock input/output from the $\overline{\text{SCLK}}$ terminal is also enabled.

The transmit control bit TXTRG is used as the trigger to start transmitting data.
Data to be transmitted is written to the transmit data shift register, and when transmitting preparations a recomplete, "1" is written to TXTRG whereupon data transmitting begins.
When interrupt has been enabled, an interrupt is generated when the transmission is completed. If there is subsequent data to be transmitted it can be sent using this interrupt.

In addition, TXTRG can be read as the status. When set to "1", it indicates transmitting operation, and "0" indicates transmitting stop.
For details on timing, see the timing chart which gives the timing for each mode.

When not transmitting, set TXEN to "0" to disable transmitting status.

■ Receive enable register, receive control bit

For receiving control, use the receive enable register RXEN and receive control bit RXTRG.
Receive enable register RXEN is used to set receiving enable/disable status. When "1" is written into this register to set the receiving enable status, clock input to the shift register is enabled and the system is ready to receive data. In the clock synchronous mode, synchronous clock input/output from the $\overline{\text{SCLK}}$ terminal is also enabled.
With the above setting, receiving begins and serial data input from the SIN terminal goes to the shift register.
The operation of the receive control bit RXTRG is slightly different depending on whether a clock synchronous system or an asynchronous system is being used.
In the clock synchronous system, the receive control bit TXTRG is used as the trigger to start receiving data.
When received data has been read and the preparation for next data receiving is completed, write "1" into RXTRG to start receiving. (When "1" is written to RXTRG in slave mode, $\overline{\text{SRDY}}$ switches to "0".)
In an asynchronous system, RXTRG is used to prepare for next data receiving. After reading the received data from the received data buffer, write "1" into RXTRG to signify that the received data buffer is empty. If "1" is not written into RXTRG, the overrun error flag OER will be set to "1" when the next receiving operation is completed. (An overrun error will be generated when receiving is completed between reading the received data and the writing of "1" to RXTRG.)
In addition, RXTRG can be read as the status. In either clock synchronous mode or asynchronous mode, when RXTRG is set to "1", it indicates receiving operation and when set to "0", it indicates that receiving has stopped.
For details on timing, see the timing chart which gives the timing for each mode.

When you do not receive, set RXEN to "0" to disable receiving status.

5.8.6 Operation of clock synchronous transfer

Clock synchronous transfer involves the transfer of 8-bit data by synchronizing it to eight clocks. The same synchronous clock is used by both the transmitting and receiving sides.
When the serial interface is used in the master mode, the clock signal selected using SCS0 and SCS1 is further divided by 1/16 and employed as the synchronous clock. This signal is then sent via the $\overline{\text{SCLK}}$ terminal to the slave side (external serial I/O device).
When used in the slave mode, the clock input to the $\overline{\text{SCLK}}$ terminal from the master side (external serial input/output device) is used as the synchronous clock.

In the clock synchronous mode, since one clock line ($\overline{\text{SCLK}}$) is shared for both transmitting and receiving, transmitting and receiving cannot be performed simultaneously. (Half duplex only is possible in clock synchronous mode.)

Transfer data is fixed at 8 bits and both transmitting and receiving are conducted with the LSB (bit 0) coming first.

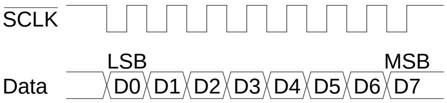


Fig. 5.8.6.1 Transfer data configuration using clock synchronous mode

Below is a description of initialization when performing clock synchronous transfer, transmit-receive control procedures and operations.
With respect to serial interface interrupt, see "5.8.8 Interrupt function".

■ Initialization of serial interface

When performing clock synchronous transfer, the following initial settings must be made.

(1) Setting of transmitting/receiving disable
To set the serial interface into a status in which both transmitting and receiving are disabled, "0" must be written to both the transmit enable register TXEN and the receive enable register RXEN. Fix these two registers to a disable status until data transfer actually begins.

(2) Port selection
Because serial interface input/output ports SIN, SOUT, $\overline{\text{SCLK}}$ and $\overline{\text{SRDY}}$ are set as I/O port terminals P10–P13 at initial reset, "1" must be written to the serial interface enable register ESIF in order to set these terminals for serial interface use.

(3) Setting of transfer mode
Select the clock synchronous mode by writing the data as indicated below to the two bits of the mode selection registers SMD0 and SMD1.

Master mode:	SMD0 = "0", SMD1 = "0"
Slave mode:	SMD0 = "1", SMD1 = "0"

(4) Clock source selection
In the master mode, select the synchronous clock source by writing data to the two bits of the clock source selection registers SCS0 and SCS1. (See Table 5.8.4.1.)
This selection is not necessary in the slave mode.

Since all the registers mentioned in (2)–(4) are assigned to the same address, it's possible to set them all with one instruction. The parity enable register EPR is also assigned to this address, however, since parity is not necessary in the clock synchronous mode, parity check will not take place regardless of how they are set.

(5) Clock source control
When the master mode is selected and programmable timer for the clock source is selected, set transfer rate on the programmable timer side. (See "5.11 Programmable Timer".)
When the divided signal of OSC3 oscillation circuit is selected for the clock source, be sure that the OSC3 oscillation circuit is turned ON prior to commencing data transfer. (See "5.4 Oscillation Circuit and Operating Mode".)

■ Data transmit procedure

The control procedure and operation during transmitting is as follows.

(1) Write "0" in the transmit enable register TXEN and the receive enable register RXEN to reset the serial interface.

(2) Write "1" in the transmit enable register TXEN to set into the transmitting enable status.

(3) Write the transmitting data into TRXD0–TRXD7.

(4) In case of the master mode, confirm the receive ready status on the slave side (external serial input/output device), if necessary. Wait until it reaches the receive ready status.

(5) Write "1" in the transmit control bit TXTRG and start transmitting.

In the master mode, this control causes the synchronous clock to change to enable and to be provided to the shift register for transmitting and output from the $\overline{\text{SCLK}}$ terminal.
In the slave mode, it waits for the synchronous clock to be input from the $\overline{\text{SCLK}}$ terminal.
The transmitting data of the shift register shifts one bit at a time at each falling edge of the synchronous clock and is output from the SOUT terminal. When the final bit (MSB) is output, the SOUT terminal is maintained at that level, until the next transmitting begins.

The transmitting complete interrupt factor flag FSTRA is set to "1" at the point where the data transmitting of the shift register is completed. When interrupt has been enabled, a transmitting complete interrupt is generated at this point.
Set the following transmitting data using this interrupt.

(6) Repeat steps (3) to (5) for the number of bytes of transmitting data, and then set the transmit disable status by writing "0" to the transmit enable register TXEN, when the transmitting is completed.

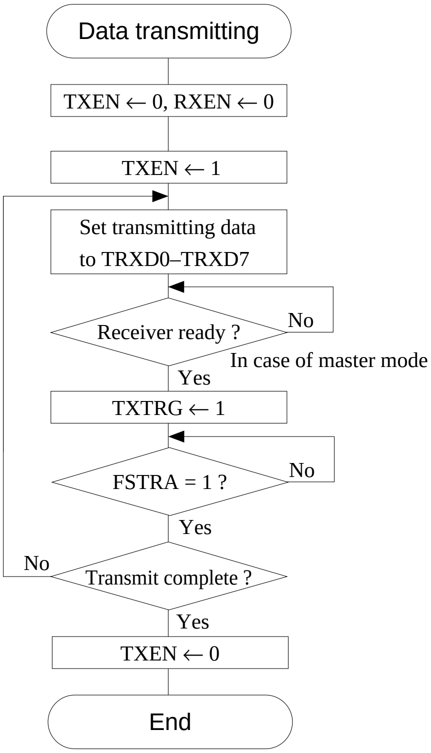


Fig. 5.8.6.2 Transmit procedure in clock synchronous mode

■ Data receive procedure

The control procedure and operation during receiving is as follows.

(1) Write "0" in the receive enable register RXEN and transmit enable register TXEN to reset the serial interface.

(2) Write "1" in the receive enable register RXEN to set into the receiving enable status.

(3) In case of the master mode, confirm the transmit ready status on the slave side (external serial input/output device), if necessary. Wait until it reaches the transmit ready status.

(4) Write "1" in the receive control bit RXTRG and start receiving.

In the master mode, this control causes the synchronous clock to change to enable and is provided to the shift register for receiving and output from the $\overline{\text{SCLK}}$ terminal.
In the slave mode, it waits for the synchronous clock to be input from the $\overline{\text{SCLK}}$ terminal. The received data input from the SIN terminal is successively incorporated into the shift register in synchronization with the rising edge of the synchronous clock.
At the point where the data of the 8th bit has been incorporated at the final (8th) rising edge of the synchronous clock, the content of the shift register is sent to the received data buffer and the receiving complete interrupt factor flag FSREC is set to "1". When interrupt has been enabled, a receiving complete interrupt is generated at this point.

(5) Read the received data from TRXD0–TRXD7 using receiving complete interrupt.

(6) Repeat steps (3) to (5) for the number of bytes of receiving data, and then set the receive disable status by writing "0" to the receive enable register RXEN, when the receiving is completed.

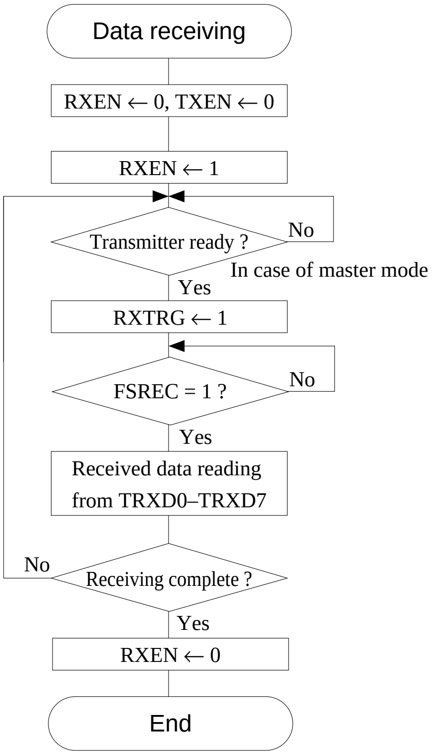


Fig. 5.8.6.3 Receiving procedure in clock synchronous mode

■ Transmit/receive ready ($\overline{\text{SRDY}}$) signal

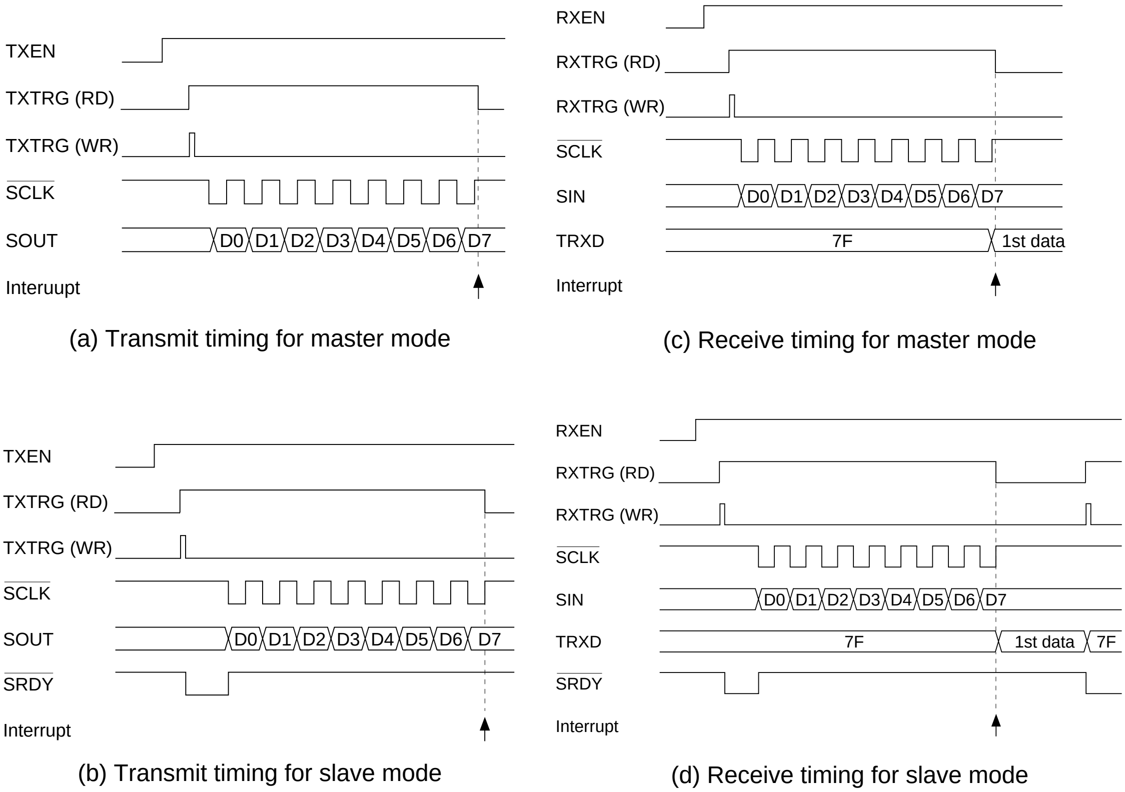
When this serial interface is used in the clock synchronous slave mode (external clock input), an $\overline{\text{SRDY}}$ signal is output to indicate whether or not this serial interface can transmit/receive to the master side (external serial input/output device). This signal is output from the $\overline{\text{SRDY}}$ terminal and when this interface enters the transmit or receive enable (READY) status, it becomes "0" (LOW level) and becomes "1" (HIGH level) when there is a BUSY status, such as during transmit/receive operation.

The $\overline{\text{SRDY}}$ signal changes the "1" to "0," immediately after writing "1" into the transmit control bit TXTRG or the receive control bit RXTRG and returns from "0" to "1", at the point where the first synchronous clock has been input (falling edge).

When you have set in the master mode, control the transfer by inputting the same signal from the slave side using the input port or I/O port. At this time, since the $\overline{\text{SRDY}}$ terminal is not set and instead P13 functions as the I/O port, you can apply this port for said control.

■ Timing chart

The timing chart for the clock synchronous system transmission is shown in Figure 5.8.6.4.



(a) Transmit timing for master mode

(c) Receive timing for master mode

(b) Transmit timing for slave mode

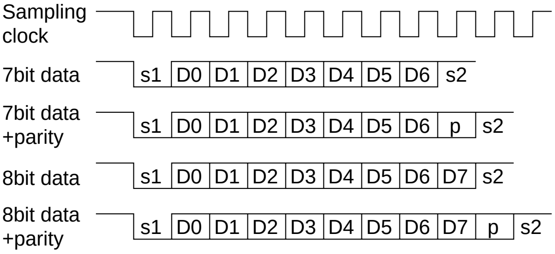
(d) Receive timing for slave mode

Fig. 5.8.6.4 Timing chart (clock synchronous system transmission)

5.8.7 Operation of asynchronous transfer

Asynchronous transfer is a mode that transfers by adding a start bit and a stop bit to the front and the back of each piece of serial converted data. In this mode, there is no need to use a clock that is fully synchronized clock on the transmit side and the receive side, but rather transmission is done while adopting the synchronization at the start/stop bits that have attached before and after each piece of data. The RS-232C interface functions can be easily realized by selecting this transfer mode.
This interface has separate transmit and receive shift registers and is designed to permit full duplex transmission to be done simultaneously for transmitting and receiving.

For transfer data in the asynchronous 7-bit mode, either 7 bits data (no parity) or 7 bits data + parity bit can be selected. In the asynchronous 8-bit mode, either 8 bits data (no parity) or 8 bits data + parity bit can be selected.
Parity can be even or odd, and parity checking of received data and adding a party bit to transmitting data will be done automatically. Thereafter, it is not necessary to be conscious of parity itself in the program.
The start bit and stop bit are respectively fixed at one bit and data is transmitted and received by placing the LSB (bit 0) at the front.



s1 : Start bit (Low level, 1 bit)
s2 : Stop bit (High level, 1 bit)
p : Parity bit

Fig. 5.8.7.1 Transfer data configuration for asynchronous system

Here following, we will explain the control sequence and operation for initialization and transmitting /receiving in case of asynchronous data transfer. See "5.8.8 Interrupt function" for the serial interface interrupts.

■ Initialization of serial interface

The below initialization must be done in cases of asynchronous system transfer.

(1) Setting of transmitting/receiving disable
To set the serial interface into a status in which both transmitting and receiving are disabled, "0" must be written to both the transmit enable register TXEN and the receive enable register RXEN. Fix these two registers to a disable status until data transfer actually begins.

(2) Port selection
Because serial interface input/output terminals SIN and SOUT are set as I/O port terminals P10 and P11 at initial reset, "1" must be written to the serial interface enable register ESIF in order to set these terminals for serial interface use. $\overline{\text{SCLK}}$ and $\overline{\text{SRDY}}$ terminals set in the clock synchronous mode are not used in the asynchronous mode. These terminals function as I/O port terminals P12 and P13.

(3) Setting of transfer mode
Select the asynchronous mode by writing the data as indicated below to the two bits of the mode selection registers SMD0 and SMD1.

7-bit mode: SMD0 = "0", SMD1 = "1"
8-bit mode: SMD0 = "1", SMD1 = "1"

(4) Parity bit selection
When checking and adding parity bits, write "1" into the parity enable register EPR to set to "with parity check". As a result of this setting, in the asynchronous 7-bit mode, it has a 7 bits data + parity bit configuration and in the asynchronous 8-bit mode it has an 8 bits data + parity bit configuration.In this case, parity checking for receiving and adding a party bit for transmitting is done automatically in hardware. Moreover, when "with parity check" has been selected, "odd" or "even" parity must be further selected in the parity mode selection register PMD.
When "0" is written to the PMD register to select "without parity check" in the asynchronous 7-bit mode, data configuration is set to 7 bits data (no parity) and in the asynchronous 8-bit mode (no parity) it is set to 8 bits data (no parity) and parity checking and parity bit adding will not be done.

(5) Clock source selection
Select the clock source by writing data to the two bits of the clock source selection registers SCS0 and SCS1. (See Table 5.8.4.1.)

Since all the registers mentioned in (2)–(5) are assigned to the same address, it's possible to set them all with one instruction.

(6) Clock source control
When the programmable timer is selected for the clock source, set transfer rate on the programmable timer side. (See "5.11 Programmable Timer".)
When the divided signal of OSC3 oscillation circuit is selected for the clock source, be sure that the OSC3 oscillation circuit is turned ON prior to commencing data transfer. (See "5.4 Oscillation Circuit and Operating Mode".)

■ Data transmit procedure

The control procedure and operation during transmitting is as follows.

(1) Write "0" in the transmit enable register TXEN to reset the serial interface.

(2) Write "1" in the transmit enable register TXEN to set into the transmitting enable status.

(3) Write the transmitting data into TRXD0–TRXD7. Also, when 7-bit data is selected, the TRXD7 data becomes invalid.

(4) Write "1" in the transmit control bit TXTRG and start transmitting.
This control causes the shift clock to change to enable and a start bit (LOW) is output to the SOUT terminal in synchronize to its rising edge. The transmitting data set to the shift register is shifted one bit at a time at each rising edge of the clock thereafter and is output from the SOUT terminal. After the data output, it outputs a stop bit (HIGH) and HIGH level is maintained until the next start bit is output.

The transmitting complete interrupt factor flag FSTRA is set to "1" at the point where the data transmitting is completed. When interrupt has been enabled, a transmitting complete interrupt is generated at this point.
Set the following transmitting data using this interrupt.

(5) Repeat steps (3) to (4) for the number of bytes of transmitting data, and then set the transmit disable status by writing "0" to the transmit enable register TXEN, when the transmitting is completed.

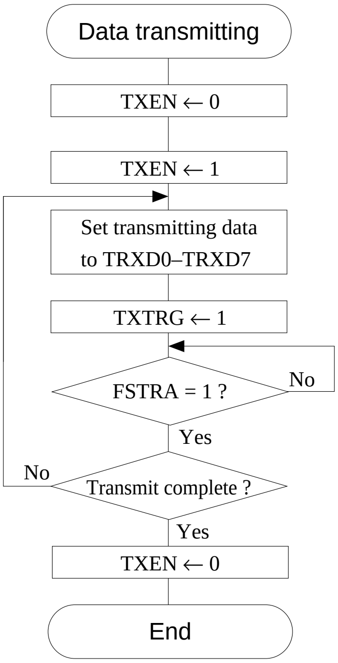


Fig. 5.8.7.2 Transmit procedure in asynchronous mode

■ Data receive procedure

The control procedure and operation during receiving is as follows.

(1) Write "0" in the receive enable register RXEN to set the receiving disable status and to reset the respective PER, OER, FER flags that indicate parity, overrun and framing errors.

(2) Write "1" in the receive enable register RXEN to set into the receiving enable status.

(3) The shift clock will change to enable from the point where the start bit (LOW) has been input from the SIN terminal and the receive data will be synchronized to the rising edge following the second clock, and will thus be successively incorporated into the shift register.
After data bits have been incorporated, the stop bit is checked and, if it is not HIGH, it becomes a framing error and the error interrupt factor flag FSERR is set to "1". When interrupt has been enabled, an error interrupt is generated at this point.
When receiving is completed, data in the shift register is transferred to the received data buffer and the receiving complete interrupt flag FSREC is set to "1". When interrupt has been enabled, a receiving complete interrupt is generated at this point. (When an overrun error is generated, the interrupt factor flag FSREC is not set to "1" and a receiving complete interrupt is not generated.)
If "with parity check" has been selected, a parity check is executed when data is transferred into the received data buffer from the shift register and if a parity error is detected, the error interrupt factor flag is set to "1". When the interrupt has been enabled, an error interrupt is generated at this point just as in the framing error mentioned above.

(4) Read the received data from TRXD0–TRXD7 using receiving complete interrupt.

(5) Write "1" to the receive control bit RXTRG to inform that the receive data has been read out.
When the following data is received prior to writing "1" to RXTRG, it is recognized as an overrun error and the error interrupt factor flag is set to "1". When the interrupt has been enabled, an error interrupt is generated at this point just as in the framing error and parity error mentioned above.

(6) Repeat steps (3) to (5) for the number of bytes of receiving data, and then set the receive disable status by writing "0" to the receive enable register RXEN, when the receiving is completed.

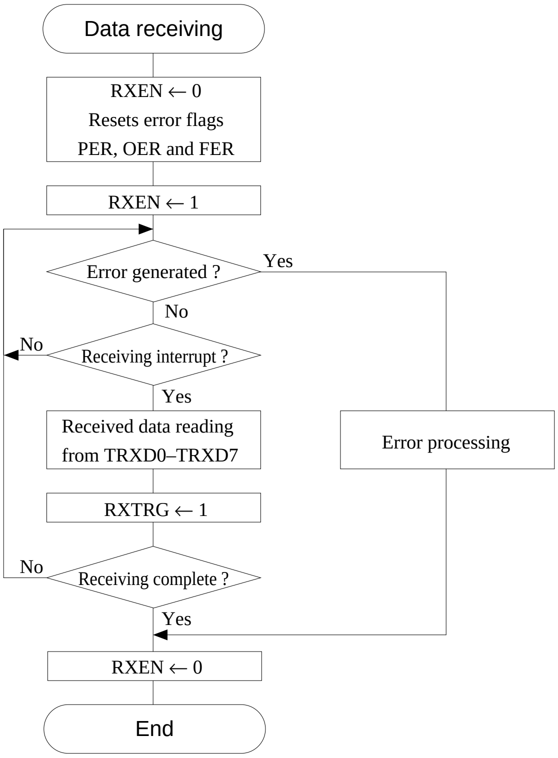


Fig. 5.8.7.3 Receiving procedure in asynchronous mode

■ Receive error

During receiving the following three types of errors can be detected by an interrupt.

(1) Parity error

When writing "1" to the EPR register to select "with parity check", a parity check (vertical parity check) is executed during receiving. After each data bit is sent a parity check bit is sent. The parity check bit is a "0" or a "1". Even parity checking will cause the sum of the parity bit and the other bits to be even. Odd parity causes the sum to be odd. This is checked on the receiving side.

The parity check is performed when data received in the shift register is transferred to the received data buffer. It checks whether the parity check bit is a "1" or a "0" (the sum of the bits including the parity bit) and the parity set in the PMD register match. When it does not match, it is recognized as an parity error and the parity error flag PER and the error interrupt factor flag FSERR is set to "1".

When interrupt has been enabled, an error interrupt is generated at this point.

The PER flag is reset to "0" by writing "1".

Even when this error has been generated, the received data corresponding to the error is transferred in the received data buffer and the receive operation also continues.

The received data at this point cannot assured because of the parity error.

(2) Framing error

In asynchronous transfer, synchronization is adopted for each character at the start bit ("0") and the stop bit ("1"). When receiving has been done with the stop bit set at "0", the serial interface judges the synchronization to be off and a framing error is generated. When this error is generated, the framing error flag FER and the error interrupt factor flag FSERR are set to "1". When interrupt has been enabled, an error interrupt is generated at this point.

The FER flag is reset to "0" by writing "1".

Even when this error has been generated, the received data for it is loaded into the receive data buffer and the receive operation also continues. However, even when it does not become a framing error with the following data receipt, such data cannot be assured.

Even when this error has been generated, the received data corresponding to the error is transferred in the received data buffer and the receive operation also continues. However, even when it does not become a framing error with the following data receiving, such data cannot be assured.

(3) Overrun error

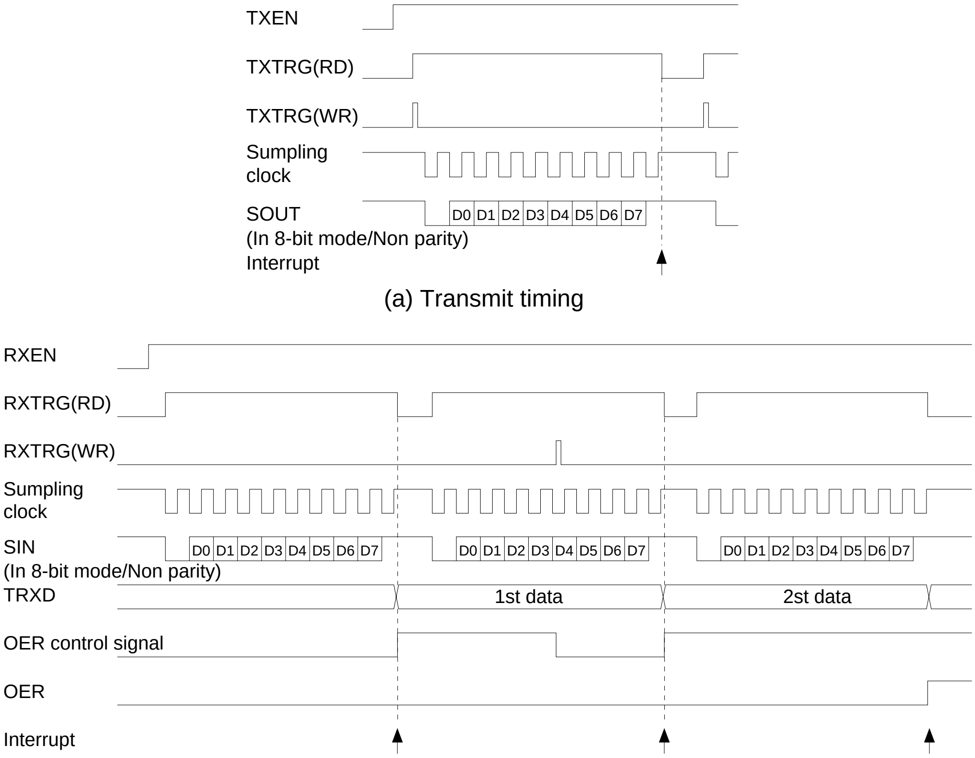
When the next data is received before "1" is written to RXTRG, an overrun error will be generated, because the previous receive data will be overwritten. When this error is generated, the overrun error flag OER and the error interrupt factor flag FSERR are set to "1". When interrupt has been enabled, an error interrupt is generated at this point. The OER flag is reset to "0" by writing "1" into it.

Even when this error has been generated, the received data corresponding to the error is transferred in the received data buffer and the receive operation also continues.

Furthermore, when the timing for writing "1" to RXTRG and the timing for the received data transfer to the received data buffer overlap, it will be recognized as an overrun error.

■ Timing chart

Figure 5.8.7.4 show the asynchronous transfer timing chart.



(a) Transmit timing

(b) Receive timing

Fig. 5.8.7.4 Timing chart (asynchronous transfer)

5.8.8 Interrupt function

This serial interface includes a function that generates the below indicated three types of interrupts.

- **Transmitting complete interrupt**
- **Receiving complete interrupt**
- **Error interrupt**

The interrupt factor flag FSxxx and the interrupt enable register ESxxx for the respective interrupt factors are provided and then the interrupt enable/ disable can be selected by the software. In addition, a priority level of the serial interface interrupt for the CPU can be optionally set at levels 0 to 3 by the interrupt priority registers PSIF0 and PSIF1.
For details on the above mentioned interrupt control register and the operation following generation of an interrupt, see "5.15 Interrupt and Standby Status".
Figure 5.8.8.1 shows the configuration of the serial interface interrupt circuit.

■ Transmitting complete interrupt

This interrupt factor is generated at the point where the sending of the data written into the shift register has been completed and sets the interrupt factor flag FSTRA to "1". When set in this manner, if the corresponding interrupt enable register ESTRA is set to "1" and the corresponding interrupt priority registers PSIF0 and PSIF1 are set to a higher level than the setting of interrupt flags (I0 and I1), an interrupt will be generated to the CPU.
When "0" has been written into the interrupt enable register ESTRA and interrupt has been disabled, an interrupt is not generated to the CPU. Even in this case, the interrupt factor flag FSTRA is set to "1".
The interrupt factor flag FSTRA is reset to "0" by writing "1".
The following transmitting data can be set and the transmitting start (writing "1" to TXTRG) can be controlled by generation of this interrupt factor.
The exception processing vector address for this interrupt factor is set at 000014H.

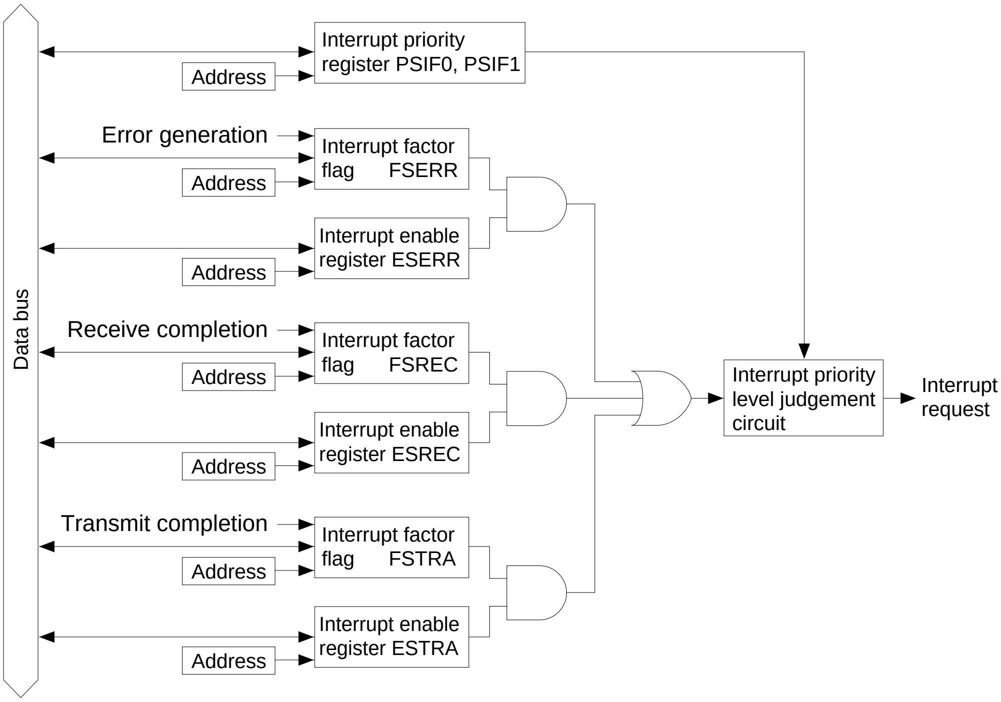

Data bus
Address
Interrupt priority register PSIF0, PSIF1
Error generation
Interrupt factor flag FSERR
Address
Interrupt enable register ESERR
Address
Receive completion
Interrupt factor flag FSREC
Address
Interrupt enable register ESREC
Address
Transmit completion
Interrupt factor flag FSTRA
Address
Interrupt enable register ESTRA
Address
Interrupt priority level judgement circuit
Interrupt request

Fig. 5.8.8.1 Configuration of serial interface interrupt circuit

■ Receiving complete interrupt

This interrupt factor is generated at the point where receiving has been completed and the receive data incorporated into the shift register has been transferred into the received data buffer and it sets the interrupt factor flag FSREC to "1". When set in this manner, if the corresponding interrupt enable register ESREC is set to "1" and the corresponding interrupt priority registers PSIF0 and PSIF1 are set to a higher level than the setting of interrupt flags (I0 and I1), an interrupt will be generated to the CPU.
When "0" has been written into the interrupt enable register ESREC and interrupt has been disabled, an interrupt is not generated to the CPU. Even in this case, the interrupt factor flag FSREC is set to "1".
The interrupt factor flag FSREC is reset to "0" by writing "1".

The generation of this interrupt factor permits the received data to be read.

Also, the interrupt factor flag is set to "1" when a parity error or framing error is generated.

The exception processing vector address for this interrupt factor is set at 000012H.

■ Error interrupt

This interrupt factor is generated at the point where a parity error, framing error or overrun error is detected during receiving and it sets the interrupt factor flag FSERR to "1". When set in this manner, if the corresponding interrupt enable register ESERR is set to "1" and the corresponding interrupt priority registers PSIF0 and PSIF1 are set to a higher level than the setting of interrupt flags (I0 and I1), an interrupt will be generated to the CPU.
When "0" has been written in the interrupt enable register ESERR and interrupt has been disabled, an interrupt is not generated to the CPU. Even in this case, the interrupt factor flag FSERR is set to "1".
The interrupt factor flag FSERR is reset to "0" by writing "1".

Since all three types of errors result in the same interrupt factor, you should identify the error that has been generated by the error flags PER (parity error), OER (overrun error) and FER (framing error).

The exception processing vector address for this interrupt factor is set at 000010H.

5.8.9 Control of serial interface

Table 5.8.9.1 show the serial interface control bits.

Table 5.8.9.1(a) Serial interface control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF48	D7	–	–	–	–	–		"0" when being read
	D6	EPR	Parity enable register	With parity	Non parity	0	R/W	Only for
	D5	PMD	Parity mode selection	Odd	Even	0	R/W	asynchronous mode
	D4	SCS1	Clock source selection			0	R/W	In the clock synchro-
			SCS1 SCS0 Clock source					nous slave mode,
			1 1 Programmable timer					external clock is
	D3	SCS0	1 0 f_{OSC3} / 4			0	R/W	selected.
			0 1 f_{OSC3} / 8					
			0 0 f_{OSC3} / 16					
	D2	SMD1	Serial I/F mode selection			0	R/W	
			SMD1 SMD0 Mode					
			1 1 Asynchronous 8-bit					
	D1	SMD0	1 0 Asynchronous 7-bit			0	R/W	
			0 1 Clock synchronous slave					
			0 0 Clock synchronous master					
	D0	ESIF	Serial I/F enable register	Serial I/F	I/O port	0	R/W	
00FF49	D7	–	–	–	–	–		"0" when being read
	D6	FER	Framing error flag R	Error	No error	0	R/W	Only for
			W	Reset (0)	No operation			asynchronous mode
	D5	PER	Parity error flag R	Error	No error	0	R/W	
			W	Reset (0)	No operation			
	D4	OER	Overrun error flag R	Error	No error	0	R/W	
			W	Reset (0)	No operation			
	D3	RXTRG	Receive trigger/status R	Run	Stop	0	R/W	
			W	Trigger	No operation			
	D2	RXEN	Receive enable	Enable	Disable	0	R/W	
	D1	TXTRG	Transmit trigger/status R	Run	Stop	0	R/W	
			W	Trigger	No operation			
	D0	TXEN	Transmit enable	Enable	Disable	0	R/W	
00FF4A	D7	TRXD7	Transmit/Receive data D7 (MSB)	High	Low	X	R/W	
	D6	TRXD6	Transmit/Receive data D6					
	D5	TRXD5	Transmit/Receive data D5					
	D4	TRXD4	Transmit/Receive data D4					
	D3	TRXD3	Transmit/Receive data D3					
	D2	TRXD2	Transmit/Receive data D2					
	D1	TRXD1	Transmit/Receive data D1					
	D0	TRXD0	Transmit/Receive data D0 (LSB)					
00FF20	D7	PK01	K00–K07 interrupt priority register	PK01 PK00 PSIF1 PSIF0 PSW1 PSW0 Priority PTM1 PTM0 level 1 1 Level 3 1 0 Level 2 0 1 Level 1 0 0 Level 0		0	R/W	
	D6	PK00						
	D5	PSIF1	Serial interface interrupt priority register			0	R/W	
	D4	PSIF0						
	D3	PSW1	Stopwatch timer interrupt priority register			0	R/W	
	D2	PSW0						
	D1	PTM1	Clock timer interrupt priority register			0	R/W	
	D0	PTM0						

Table 5.8.9.1(b) Serial interface control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF23	D7	EPT1	Programmable timer 1 interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D6	EPT0	Programmable timer 0 interrupt enable register					
	D5	EK1	K10 and K11 interrupt enable register					
	D4	EK0H	K04–K07 interrupt enable register					
	D3	EK0L	K00–K03 interrupt enable register					
	D2	ESERR	Serial I/F (error) interrupt enable register					
	D1	ESREC	Serial I/F (receiving) interrupt enable register					
	D0	ESTRA	Serial I/F (transmitting) interrupt enable register					
00FF25	D7	FPT1	Programmable timer 1 interrupt factor flag	(R) Interrupt factor is generated	(R) No interrupt factor is generated	0	R/W	
	D6	FPT0	Programmable timer 0 interrupt factor flag					
	D5	FK1	K10 and K11 interrupt factor flag					
	D4	FK0H	K04–K07 interrupt factor flag					
	D3	FK0L	K00–K03 interrupt factor flag	(W) Reset	(W) No operation			
	D2	FSERR	Serial I/F (error) interrupt factor flag					
	D1	FSREC	Serial I/F (receiving) interrupt factor flag					
	D0	FSTRA	Serial I/F (transmitting) interrupt factor flag					

ESIF: 00FF48H•D0

Sets the serial interface terminals (P10–P13).

When "1" is written:	Serial input/output terminal
When "0" is written:	I/O port terminal
Reading:	Valid

The ESIF is the serial interface enable register and P10–P13 terminals become serial input/output terminals (SIN, SOUT, $\overline{\text{SCLK}}$, $\overline{\text{SRDY}}$) when "1" is written, and they become I/O port terminals when "0" is written.
Also, see Table 5.8.3.2 for the terminal settings according to the transfer modes.
At initial reset, ESIF is set to "0" (I/O port).

SMD0, SMD1: 00FF48H•D1, D2

Set the transfer modes according to Table 5.8.9.2.

Table 5.8.9.2 Transfer mode settings

SMD1	SMD0	Mode
1	1	Asynchronous system 8-bit
1	0	Asynchronous system 7-bit
0	1	Clock synchronous system slave
0	0	Clock synchronous system master

SMD0 and SMD1 can also read out.
At initial reset, this register is set to "0" (clock synchronous master mode).

SCS0, SCS1: 00FF48H•D3, D4

Select the clock source according to Table 5.8.9.3.

Table 5.8.9.3 Clock source selection

SCS1	SCS0	Clock source
1	1	Programmable timer
1	0	f_{OSC3} / 4
0	1	f_{OSC3} / 8
0	0	f_{OSC3} / 16

SCS0 and SCS1 can also be read out.
In the clock synchronous slave mode, setting of this register is invalid.
At initial reset, this register is set to "0" (f_{OSC3}/16).

EPR: 00FF48H•D6

Selects the parity function.

When "1" is written:	With parity
When "0" is written:	Non parity
Reading:	Valid

Selects whether or not to check parity of the received data and to add a parity bit to the transmitting data. When "1" is written to EPR, the most significant bit of the received data is considered to be the parity bit and a parity check is executed. A parity bit is added to the transmitting data. When "0" is written, neither checking is done nor is a parity bit added.

Parity is valid only in asynchronous mode and the EPR setting becomes invalid in the clock synchronous mode.
At initial reset, EPR is set to "0" (non parity).

PMD: 00FF48H•D5

Selects odd parity/even parity.

When "1" is written:	Odd parity
When "0" is written:	Even parity
Reading:	Valid

When "1" is written to PMD, odd parity is selected and even parity is selected when "0" is written. The parity check and addition of a parity bit is only valid when "1" has been written to EPR. When "0" has been written to EPR, the parity setting by PMD becomes invalid.
At initial reset, PMD is set to "0" (even parity).

TXEN: 00FF49H•D0

Sets the serial interface to the transmitting enable status.

When "1" is written:	Transmitting enable
When "0" is written:	Transmitting disable
Reading:	Valid

When "1" is written to TXEN, the serial interface shifts to the transmitting enable status and shifts to the transmitting disable status when "0" is written.
Set TXEN to "0" when making the initial settings of the serial interface and similar operations.
At initial reset, TXEN is set to "0" (transmitting disable).

TXTRG: 00FF49H•D1

Functions as the transmitting start trigger and the operation status indicator (transmitting/stop status).

When "1" is read:	During transmitting
When "0" is read:	During stop
When "1" is written:	Transmitting start
When "0" is written:	Invalid

Starts the transmitting when "1" is written to TXTRG after writing the transmitting data.
TXTRG can be read as the status. When set to "1", it indicates transmitting operation, and "0" indicates transmitting stop.
At initial reset, TXTRG is set to "0" (during stop).

RXEN: 00FF49H•D2

Sets the serial interface to the receiving enable status.

When "1" is written:	Receiving enable
When "0" is written:	Receiving disable
Reading:	Valid

When "1" is written to RXEN, the serial interface shifts to the receiving enable status and shifts to the receiving disable status when "0" is written.
Set RXEN to "0" when making the initial settings of the serial interface and similar operations.
At initial reset, RXEN is set to "0" (receiving disable).

RXTRG: 00FF49H•D3

Functions as the receiving start trigger or preparation for the following data receiving and the operation status indicator (during receiving/during stop).

When "1" is read:	During receiving
When "0" is read:	During stop
When "1" is written:	Receiving start/following data receiving preparation
When "0" is written:	Invalid

RXTRG has a slightly different operation in the clock synchronous system and the asynchronous system.

The RXTRG in the clock synchronous system, is used as the trigger for the receiving start.
Writes "1" into RXTRG to start receiving at the point where the receive data has been read and the following receive preparation has been done. (In the slave mode, $\overline{\text{SRDY}}$ becomes "0" at the point where "1" has been written into into the RXTRG.)

RSTRG is used in the asynchronous system for preparation of the following data receiving. Reads the received data located in the received data buffer and writes "1" into RXTRG to inform that the received data buffer has shifted to empty. When "1" has not been written to RXTRG, the overrun error flag OER is set to "1" at the point where the following receiving has been completed. (When the receiving has been completed between the operation to read the received data and the operation to write "1" into RXTRG, an overrun error occurs.)

In addition, RXTRG can be read as the status. In either clock synchronous mode or asynchronous mode, when RXTRG is set to "1", it indicates receiving operation and when set to "0", it indicates that receiving has stopped.
At initial reset, RXTRG is set to "0" (during stop).

TRXD0–TRXD7: 00FF4AH

During transmitting

Write the transmitting data into the transmit shift register.

When "1" is written: HIGH level
When "0" is written: LOW level

Write the transmitting data prior to starting transmitting.
In the case of continuous transmitting, wait for the transmitting complete interrupt, then write the data.
The TRXD7 becomes invalid for the asynchronous 7-bit mode.
Converted serial data for which the bits set at "1" as HIGH (V_{DD}) level and for which the bits set at "0" as LOW (V_{SS}) level are output from the SOUT terminal.

During receiving

Read the received data.

When "1" is read: HIGH level
When "0" is read: LOW level

The data from the received data buffer can be read out. Since the sift register is provided separately from this buffer, reading can be done during the receive operation in the asynchronous mode. (The buffer function is not used in the clock synchronous mode.)
Read the data after waiting for the receiving complete interrupt.
When performing parity check in the asynchronous 7-bit mode, "0" is loaded into the 8th bit (TRXD7) that corresponds to the parity bit.
The serial data input from the SIN terminal is level converted, making the HIGH (V_{DD}) level bit "1" and the LOW (V_{SS}) level bit "0" and is then loaded into this buffer.
At initial reset, the buffer content is undefined.

OER: 00FF49H•D4

Indicates the generation of an overrun error.

When "1" is read: Error
When "0" is read: No error

When "1" is written: Reset to "0"
When "0" is written: Invalid

OER is an error flag that indicates the generation of an overrun error and becomes "1" when an error has been generated.
An overrun error is generated when the receiving of data has been completed prior to the writing of "1" to RXTRG in the asynchronous mode.
OER is reset to "0" by writing "1".
At initial reset and when RXEN is "0", OER is set to "0" (no error).

PER: 00FF49H•D5

Indicates the generation of a parity error.

When "1" is read: Error
When "0" is read: No error

When "1" is written: Reset to "0"
When "0" is written: Invalid

PER is an error flag that indicates the generation of a parity error and becomes "1" when an error has been generated.
When a parity check is performed in the asynchronous mode, if data that does not match the parity is received, a parity error is generated.
PER is reset to "0" by writing "1".
At initial reset and when RXEN is "0", PER is set to "0" (no error).

FER: 00FF49H•D6

Indicates the generation of a framing error.

When "1" is read: Error
When "0" is read: No error

When "1" is written: Reset to "0"
When "0" is written: Invalid

FER is an error flag that indicates the generation of a framing error and becomes "1" when an error has been generated.
When the stop bit for the receiving of the asynchronous mode has become "0", a framing error is generated.
FER is reset to "0" by writing "1".
At initial reset and when RXEN is "0", FER is set to "0" (no error).

PSIF0, PSIF1: 00FF20H•D4, D5

Sets the priority level of the serial interface interrupt.
The two bits PSIF0 and PSIF1 are the interrupt priority register corresponding to the serial interface interrupt. Table 5.8.9.4 shows the interrupt priority level which can be set by this register.

Table 5.8.9.4 Interrupt priority level settings

PSIF1	PSIF0	Interrupt priority level
1	1	Level 3 ($\overline{\text{IRQ3}}$)
1	0	Level 2 ($\overline{\text{IRQ2}}$)
0	1	Level 1 ($\overline{\text{IRQ1}}$)
0	0	Level 0 (None)

At initial reset, this register is set to "0" (level 0).

ESTRA, ESREC, ESERR: 00FF23H•D0, D1, D2

Enables or disables the generation of an interrupt for the CPU.

When "1" is written:	Interrupt enabled
When "0" is written:	Interrupt disabled
Reading:	Valid

ESTRA, ESREC and ESERR are interrupt enable registers that respectively correspond to the interrupt factors for transmitting complete, receiving complete and receiving error. Interrupts set to "1" are enabled and interrupts set to "0" are disabled.
At initial reset, this register is set to "0" (interrupt disabled).

FSTRA, FSREC, FSERR: 00FF25H•D0, D1, D2

Indicates the serial interface interrupt generation status.

When "1" is read:	Interrupt factor present
When "0" is read:	Interrupt factor not present
When "1" is written:	Resets factor flag
When "0" is written:	Invalid

FSTRA, FSREC and FSERR are interrupt factor flags that respectively correspond to the interrupts for transmitting complete, receiving complete and receiving error and are set to "1" by generation of each factor.
Transmitting complete interrupt factor is generated at the point where the data transmitting of the shift register has been completed.
Receiving complete interrupt factor is generated at the point where the received data has been transferred into the received data buffer.
Receive error interrupt factor is generated when a parity error, framing error or overrun error has been detected during data receiving.
When set in this manner, if the corresponding interrupt enable register is set to "1" and the corresponding interrupt priority register is set to a higher level than the setting of interrupt flags (I0 and I1), an interrupt will be generated to the CPU.
Regardless of the interrupt enable register and interrupt priority register settings, the interrupt factor flag will be set to "1" by the occurrence of an interrupt generation condition.
To accept the subsequent interrupt after interrupt generation, re-setting of the interrupt flags (set interrupt flag to lower level than the level indicated by the interrupt priority registers, or execute the RETE instruction) and interrupt factor flag reset are necessary. The interrupt factor flag is reset to "0" by writing "1".
At initial reset, this flag is reset to "0".

5.8.10 Programming notes

(1) Be sure to initialize the serial interface mode in the transmitting/receiving disable status (TXEN = RXEN = "0").

(2) Do not perform double trigger (writing "1") to TXTRG (RXTRG) when the serial interface is in the transmitting (receiving) operation. Furthermore, do not execute the SLP instruction. (When executing the SLP instruction, set TXEN = RXEN = "0".)

(3) In the clock synchronous mode, since one clock line ($\overline{\text{SCLK}}$) is shared for both transmitting and receiving, transmitting and receiving cannot be performed simultaneously. (Half duplex only is possible in clock synchronous mode.)
Consequently, be sure not to write "1" to RXTRG (TXTRG) when TXTRG (RXTRG) is "1".

(4) When a parity error or flaming error is generated during receiving in the asynchronous mode, the receiving error interrupt factor flag FSERR is set to "1" prior to the receiving complete interrupt factor flag FSREC for the time indicated in Table 5.8.10.1. Consequently, when an error is generated, you should reset the receiving complete interrupt factor flag FSREC to "0" by providing a wait time in error processing routines and similar routines.
When an overrun error is generated, the receiving complete interrupt factor flag FSREC is not set to "1" and a receiving complete interrupt is not generated.

Table 5.8.10.1 Time difference between FSERR and FSREC on error generation

Clock source	Time difference
f_{OSC3} / n	1/2 cycles of f_{OSC3} / n
Programmable timer	1 cycle of timer 1 underflow

(5) When the demultiplied signal of the OSC3 oscillation circuit is made the clock source, it is necessary to turn the OSC3 oscillation ON, prior to using the serial interface.
A time interval of several msec to several 10 msec, from the turning ON of the OSC3 oscillation circuit to until the oscillation stabilizes, is necessary, due to the oscillation element that is used. Consequently, you should allow an adequate waiting time after turning ON of the OSC3 oscillation, before starting transmitting/receiving of serial interface. (The oscillation start time will vary somewhat depending on the oscillator and on the externally attached parts. Refer to the oscillation start time example indicated in Chapter 7, "ELECTRICAL CHARACTERISTICS".)
At initial reset, the OSC3 oscillation circuit is set to OFF status.

5.9 Clock Timer

5.9.1 Configuration of clock timer

The E0C88112 has built in a clock timer that uses the OSC1 oscillation circuit as clock source. The clock timer is composed of an 8-bit binary counter that uses the 256 Hz signal dividing fOSC1 as its input clock and can read the data of each bit (128–1 Hz) by software.
Normally, this clock timer is used for various timing functions such as clocks.
The configuration of the clock timer is shown in Figure 5.9.1.1.

5.9.2 Interrupt function

The clock timer can generate an interrupt by each of the 32 Hz, 8 Hz, 2 Hz and 1 Hz signals.
The configuration of the clock timer interrupt circuit is shown in Figure 5.9.2.1.

Interrupts are generated by respectively setting the corresponding interrupt factor flags FTM32, FTM8, FTM2 and FTM1 at the falling edge of the 32 Hz, 8 Hz, 2 Hz and 1 Hz signals to "1". Interrupt can be prohibited by the setting the interrupt enable registers ETM32, ETM8, ETM2 and ETM1 corresponding to each interrupt factor flag.
In addition, a priority level of the clock timer interrupt for the CPU can be optionally set at levels 0 to 3 by the interrupt priority registers PTM0 and PTM1.
For details on the above mentioned interrupt control register and the operation following generation of an interrupt, see "5.15 Interrupt and Standby Status".

The exception processing vector addresses for each interrupt factor are respectively set as shown below.

32 Hz interrupt:	00001CH
8 Hz interrupt:	00001EH
2 Hz interrupt:	000020H
1 Hz interrupt:	000022H

Figure 5.9.2.2 shows the timing chart for the clock timer.

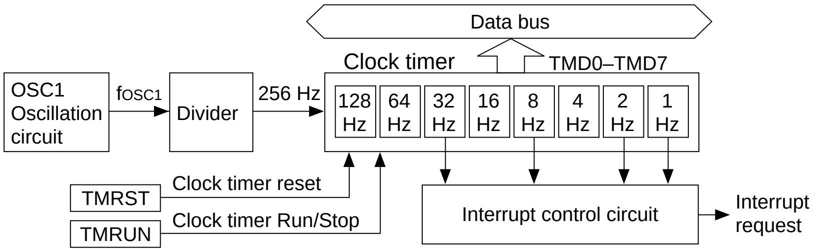


Fig. 5.9.1.1 Configuration of clock timer

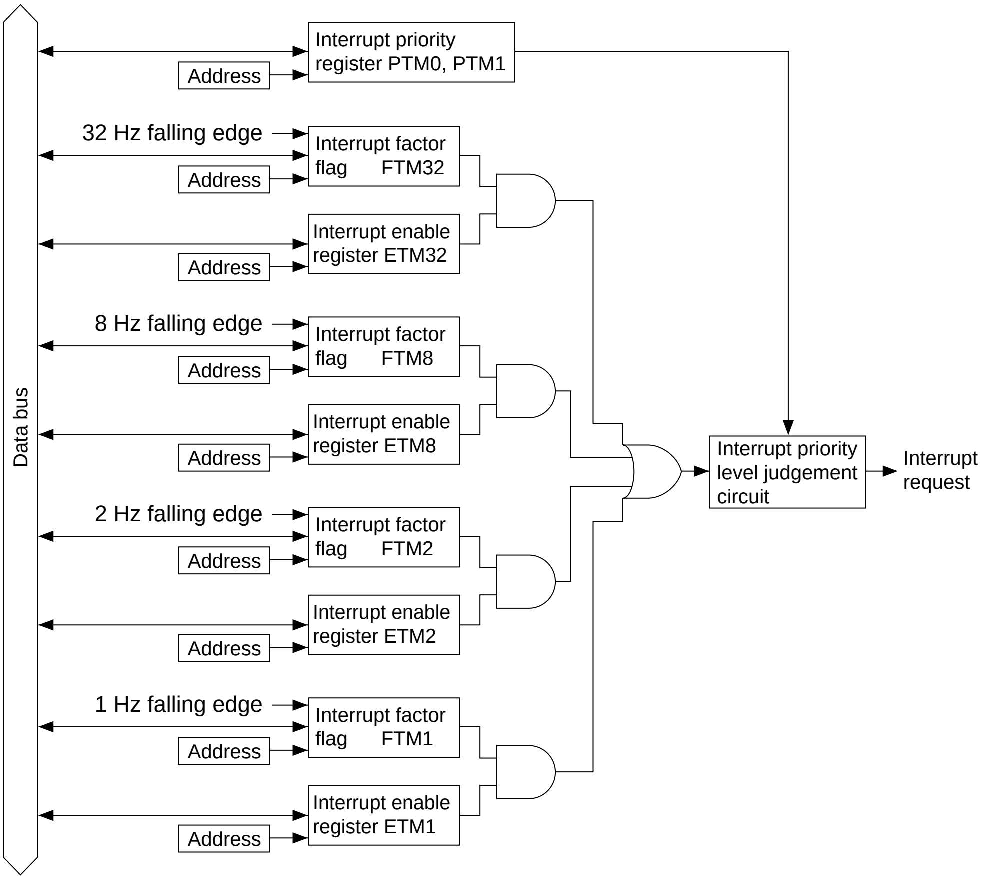

Data bus
Interrupt priority register PTM0, PTM1
Address
32 Hz falling edge
Interrupt factor flag FTM32
Address
Interrupt enable register ETM32
Address
8 Hz falling edge
Interrupt factor flag FTM8
Address
Interrupt enable register ETM8
Address
2 Hz falling edge
Interrupt factor flag FTM2
Address
Interrupt enable register ETM2
Address
1 Hz falling edge
Interrupt factor flag FTM1
Address
Interrupt enable register ETM1
Address
Interrupt priority level judgement circuit
Interrupt request

Fig. 5.9.2.1 Configuration of clock timer interrupt circuit

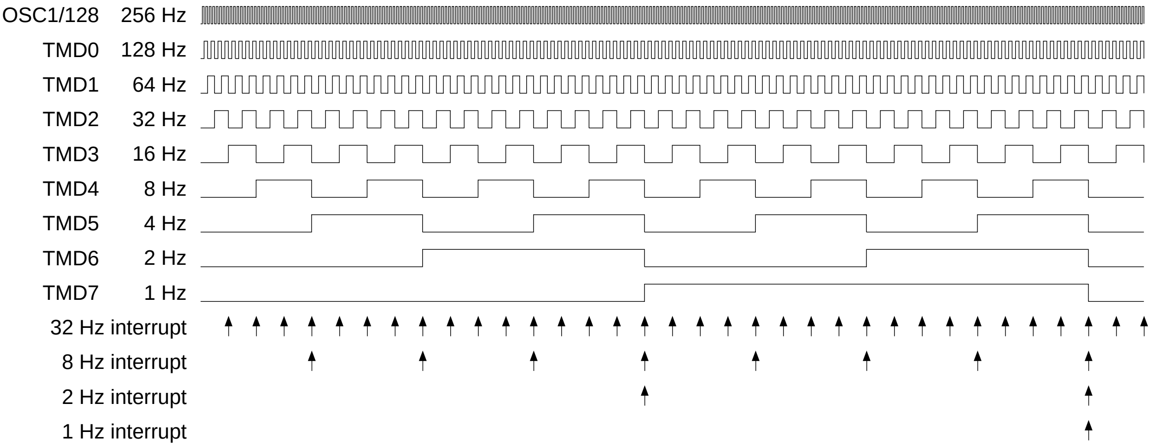


Fig. 5.9.2.2 Timing chart of clock timer

5.9.3 Control of clock timer

Table 5.9.3.1 shows the clock timer control bits.

Table 5.9.3.1 Clock timer control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF40	D7	–	–	–	–	–		"0" when being read
	D6	FOUT2	FOUT frequency selection FOUT2 FOUT1 FOUT0 Frequency 0 0 0 f_{OSC1} / 1 0 0 1 f_{OSC1} / 2 0 1 0 f_{OSC1} / 4 0 1 1 f_{OSC1} / 8 1 0 0 f_{OSC3} / 1 1 0 1 f_{OSC3} / 2 1 1 0 f_{OSC3} / 4 1 1 1 f_{OSC3} / 8			0	R/W	
	D5	FOUT1				0	R/W	
	D4	FOUT0				0	R/W	
	D3	FOUTON	FOUT output control	On	Off	0	R/W	
	D2	WDRST	Watchdog timer reset	Reset	No operation	–	W	Constantly "0" when being read
	D1	TMRST	Clock timer reset	Reset	No operation	–	W	
	D0	TMRUN	Clock timer Run/Stop control	Run	Stop	0	R/W	
00FF41	D7	TMD7	Clock timer data 1 Hz	High	Low	0	R	
	D6	TMD6	Clock timer data 2 Hz					
	D5	TMD5	Clock timer data 4 Hz					
	D4	TMD4	Clock timer data 8 Hz					
	D3	TMD3	Clock timer data 16 Hz					
	D2	TMD2	Clock timer data 32 Hz					
	D1	TMD1	Clock timer data 64 Hz					
	D0	TMD0	Clock timer data 128 Hz					
00FF20	D7	PK01	K00–K07 interrupt priority register	PK01 PK00 PSIF1 PSIF0 PSW1 PSW0 Priority PTM1 PTM0 level 1 1 Level 3 1 0 Level 2 0 1 Level 1 0 0 Level 0		0	R/W	
	D6	PK00						
	D5	PSIF1	Serial interface interrupt priority register			0	R/W	
	D4	PSIF0						
	D3	PSW1	Stopwatch timer interrupt priority register			0	R/W	
	D2	PSW0						
	D1	PTM1	Clock timer interrupt priority register			0	R/W	
	D0	PTM0						
00FF22	D7	–	–	–	–	–		"0" when being read
	D6	ESW100	Stopwatch timer 100 Hz interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D5	ESW10	Stopwatch timer 10 Hz interrupt enable register					
	D4	ESW1	Stopwatch timer 1 Hz interrupt enable register					
	D3	ETM32	Clock timer 32 Hz interrupt enable register					
	D2	ETM8	Clock timer 8 Hz interrupt enable register					
	D1	ETM2	Clock timer 2 Hz interrupt enable register					
	D0	ETM1	Clock timer 1 Hz interrupt enable register					
00FF24	D7	–	–	–	–	–		"0" when being read
	D6	FSW100	Stopwatch timer 100 Hz interrupt factor flag	(R) Interrupt factor is generated	(R) No interrupt factor is generated	0	R/W	
	D5	FSW10	Stopwatch timer 10 Hz interrupt factor flag					
	D4	FSW1	Stopwatch timer 1 Hz interrupt factor flag					
	D3	FTM32	Clock timer 32 Hz interrupt factor flag					
	D2	FTM8	Clock timer 8 Hz interrupt factor flag	(W) Reset	(W) No operation			
	D1	FTM2	Clock timer 2 Hz interrupt factor flag					
	D0	FTM1	Clock timer 1 Hz interrupt factor flag					

TMD0–TMD7: 00FF41H

The clock timer data can be read out.
Each bit of TMD0–TMD7 and frequency correspondence are as follows:

TMD0:	128Hz	TMD4:	8Hz
TMD1:	64Hz	TMD5:	4Hz
TMD2:	32Hz	TMD6:	2Hz
TMD3:	16Hz	TMD7:	1Hz

Since the TMD0–TMD7 is exclusively for reading, the write operation is invalid.
At initial reset, the timer data is set to "00H".

TMRST: 00FF40H•D1

Resets the clock timer.

When "1" is written:	Clock timer reset
When "0" is written:	No operation
Reading:	Always "0"

The clock timer is reset by writing "1" to the TMRST.
When the clock timer is reset in the RUN status, it restarts immediately after resetting. In the case of the STOP status, the reset data "00H" is maintained.
No operation results when "0" is written to the TMRST.
Since the TMRST is exclusively for writing, it always becomes "0" during reading.

TMRUN: 00FF40H•D0

Controls RUN/STOP of the clock timer.

When "1" is written:	RUN
When "0" is written:	STOP
Reading:	Valid

The clock timer starts up-counting by writing "1" to the TMRUN and stops by writing "0".
In the STOP status, the count data is maintained until it is reset or set in the next RUN status. Also, when the STOP status changes to the RUN status, the data that was maintained can be used for resuming the count.
At initial reset, the TMRUN is set to "0" (STOP).

PTM0, PTM1: 00FF20H•D0, D1

Sets the priority level of the clock timer interrupt. The two bits PTM0 and PTM1 are the interrupt priority register corresponding to the clock timer interrupt. Table 5.9.3.2 shows the interrupt priority level which can be set by this register.

Table 5.9.3.2 Interrupt priority level settings

PTM1	PTM0	Interrupt priority level
1	1	Level 3 ($\overline{\text{IRQ3}}$)
1	0	Level 2 ($\overline{\text{IRQ2}}$)
0	1	Level 1 ($\overline{\text{IRQ1}}$)
0	0	Level 0 (None)

At initial reset, this register is set to "0" (level 0).

ETM1, ETM2, ETM8, ETM32: 00FF22H•D0–D3

Enables or disables the generation of an interrupt for the CPU.

When "1" is written:	Interrupt enabled
When "0" is written:	Interrupt disabled
Reading:	Valid

The ETM1, ETM2, ETM8 and ETM32 are interrupt enable registers that respectively correspond to the interrupt factors for 1 Hz, 2 Hz, 8 Hz and 32 Hz. Interrupts set to "1" are enabled and interrupts set to "0" are disabled.
At initial reset, this register is set to "0" (interrupt disabled).

FTM1, FTM2, FTM8, FTM32: 00FF24H•D0–D3

Indicates the clock timer interrupt generation status.

When "1" is read:	Interrupt factor present
When "0" is read:	Interrupt factor not present
When "1" is written:	Resets factor flag
When "0" is written:	Invalid

The FTM1, FTM2, FTM8 and FTM32 are interrupt factor flags that respectively correspond to the interrupts for 1 Hz, 2 Hz, 8 Hz and 32 Hz and are set to "1" at the falling edge of each signal.
When set in this manner, if the corresponding interrupt enable register is set to "1" and the corresponding interrupt priority register is set to a higher level than the setting of interrupt flags (I0 and I1), an interrupt will be generated to the CPU.
Regardless of the interrupt enable register and interrupt priority register settings, the interrupt factor flag will be set to "1" by the occurrence of an interrupt generation condition.
To accept the subsequent interrupt after interrupt generation, re-setting of the interrupt flags (set interrupt flag to lower level than the level indicated by the interrupt priority registers, or execute the RETE instruction) and interrupt factor flag reset are necessary. The interrupt factor flag is reset to "0" by writing "1".
At initial reset, this flag is reset to "0".

5.9.4 Programming notes

(1) The clock timer is actually made to RUN/STOP in synchronization with the falling edge of the 256 Hz signal after writing to the TMRUN register. Consequently, when "0" is written to the TMRUN, the timer shifts to STOP status when the counter is incremented "1". The TMRUN maintains "1" for reading until the timer actually shifts to STOP status.
Figure 5.9.4.1 shows the timing chart of the RUN/STOP control.

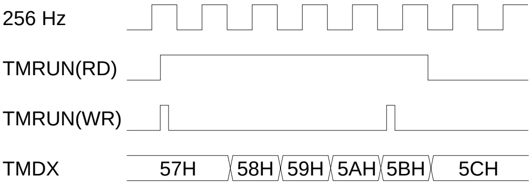


Fig. 5.9.4.1 Timing chart of RUN/STOP control

(2) The SLP instruction is executed when the clock timer is in the RUN status (TMRUN = "1"). The clock timer operation will become unstable when returning from SLEEP status. Therefore, when shifting to SLEEP status, set the clock timer to STOP status (TMRUN = "0") prior to executing the SLP instruction.

5.10 Stopwatch Timer

5.10.1 Configuration of stopwatch timer

The E0C88112 has a built-in 1/100 sec and 1/10 sec stopwatch timer. The stopwatch timer is composed of a 4-bit 2 stage BCD counter (1/100 sec units and 1/10 sec units) that makes the 256 Hz signal that divides the fOSC1 the input clock and it can read the count data by software.
Figure 5.10.1.1 shows the configuration of the stopwatch timer.
The stopwatch timer can be used as a timer different from the clock timer and can easily realize stopwatch and other such functions by software.

5.10.2 Count up pattern

The stopwatch timer is respectively composed of the 4-bit BCD counters SWD0–SWD3 and SWD4–SWD7. Figure 5.10.2.1 shows the count up pattern of the stopwatch timer.

The feedback dividing circuit generates an approximate 100 Hz signal at 2/256 sec and 3/256 sec intervals from a 256 Hz signal divided from fOSC1.

The 1/100 sec counter (SWD0–SWD3) generates an approximate 10 Hz signal at 25/256 sec and 26/256 sec intervals by counting the approximate 100 Hz signal generated by the feedback dividing circuit in 2/256 sec and 3/256 sec intervals. The count-up is made approximately 1/100 sec counting by the 2/256 sec and 3/256 sec intervals.

The 1/10 sec counter (SWD4–SWD7) generates a 1 Hz signal by counting the approximate 10 Hz signal generated by the 1/100 sec counter at 25/256 sec and 26/256 sec intervals in 4:6 ratios.

The count-up is made approximately 1/10 sec counting by 25/256 sec and 26/256 sec intervals.

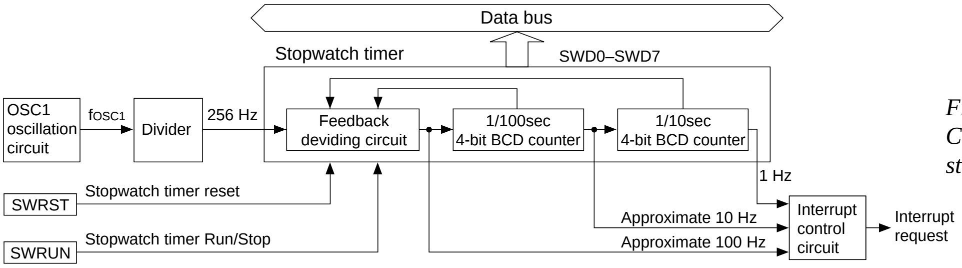


Fig. 5.10.1.1
Configuration of stopwatch timer

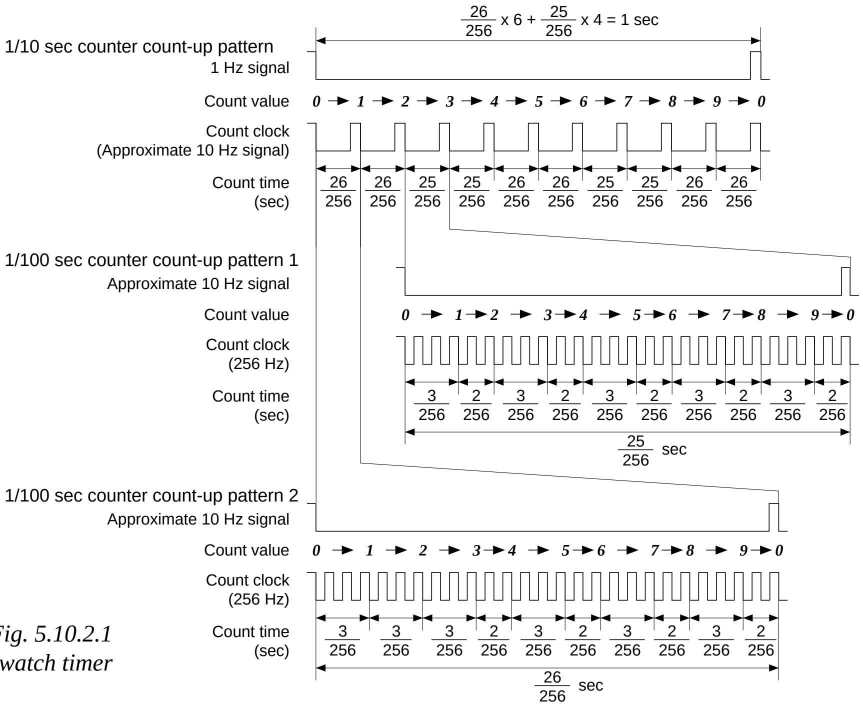


Fig. 5.10.2.1
Count-up pattern of stopwatch timer

5.10.3 Interrupt function

The stopwatch timer can generate an interrupt by each of the 100 Hz (approximately 100 Hz), 10 Hz (approximately 10 Hz) and 1 Hz signals.
Figure 5.10.3.1 shows the configuration of the stopwatch timer interrupt circuit

The corresponding factor flags FSW100, FSW10 and FSW1 are respectively set to "1" at the falling edge of the 100 Hz, 10Hz and 1Hz signal and an interrupt is generated. Interrupt can be prohibited by the setting of the interrupt enable registers ESW100, ESW10 and ESW1 corresponding to each interrupt factor flag.

In addition, a priority level of the stopwatch timer interrupt for the CPU can be optionally set at levels 0 to 3 by the interrupt priority registers PSW0 and PSW1.
For details on the above mentioned interrupt control registers and the operation following generation of an interrupt, see "5.15 Interrupt and Standby Status".
The exception processing vector addresses of each interrupt factor are respectively set as shown below.

100 Hz interrupt:	000016H
10 Hz interrupt:	000018H
1 Hz interrupt:	00001AH

Figure 5.10.3.2 shows the timing chart for the stopwatch timer.

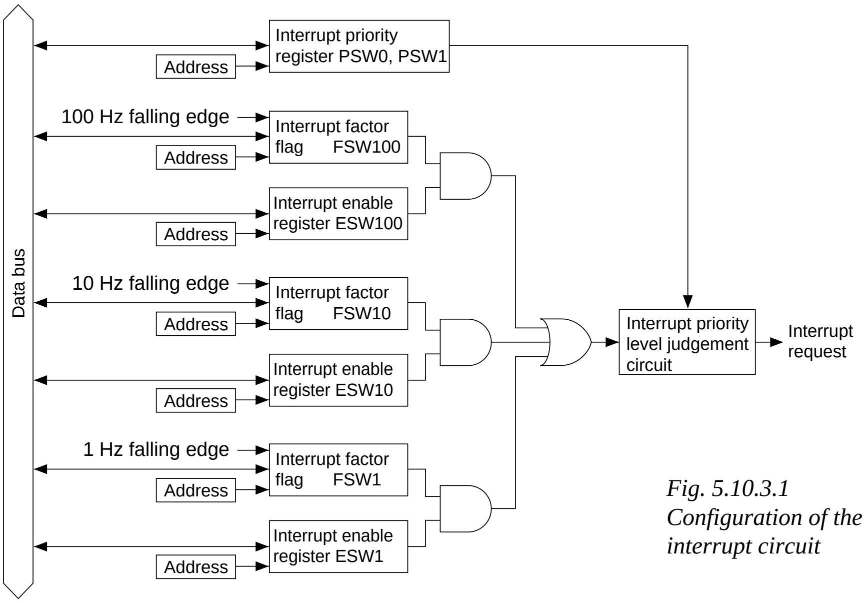

Data bus
Address
Interrupt priority register PSW0, PSW1
100 Hz falling edge
Interrupt factor flag FSW100
Interrupt enable register ESW100
10 Hz falling edge
Interrupt factor flag FSW10
Interrupt enable register ESW10
1 Hz falling edge
Interrupt factor flag FSW1
Interrupt enable register ESW1
Interrupt priority level judgement circuit
Interrupt request

Fig. 5.10.3.1
Configuration of the stopwatch timer interrupt circuit

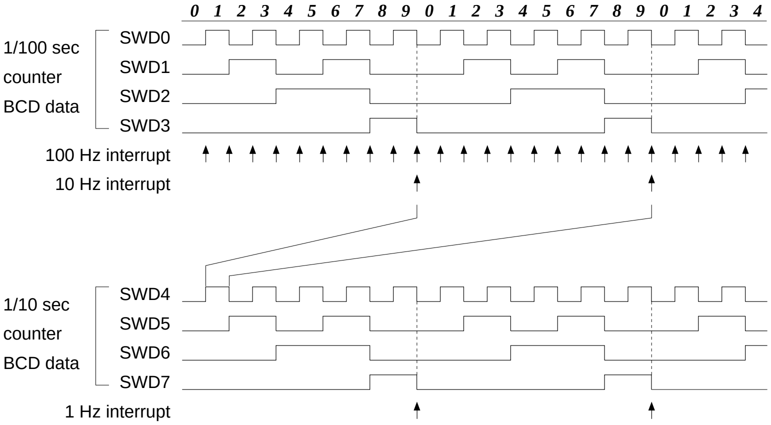


Fig. 5.10.3.2
Stopwatch timer timing chart

5.10.4 Control of stopwatch timer

Table 5.10.4.1 shows the stopwatch timer control bits.

Table 5.10.4.1 Stopwatch timer control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF42	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	–	–	–	–	–		
	D2	–	–	–	–	–		
	D1	SWRST	Stopwatch timer reset	Reset	No operation	–	W	
	D0	SWRUN	Stopwatch timer Run/Stop control	Run	Stop	0	R/W	
00FF43	D7	SWD7	Stopwatch timer data BCD (1/10 sec)			0	R	
	D6	SWD6						
	D5	SWD5						
	D4	SWD4						
	D3	SWD3	Stopwatch timer data BCD (1/100 sec)					
	D2	SWD2						
	D1	SWD1						
	D0	SWD0						
00FF20	D7	PK01	K00–K07 interrupt priority register	PK01 PK00 PSIF1 PSIF0 PSW1 PSW0 PTM1 PTM0 1 1 1 0 0 1 0 0	Priority level Level 3 Level 2 Level 1 Level 0	0	R/W	
	D6	PK00						
	D5	PSIF1	Serial interface interrupt priority register			0	R/W	
	D4	PSIF0						
	D3	PSW1	Stopwatch timer interrupt priority register			0	R/W	
	D2	PSW0						
	D1	PTM1	Clock timer interrupt priority register			0	R/W	
	D0	PTM0						
00FF22	D7	–	–	–	–	–		"0" when being read
	D6	ESW100	Stopwatch timer 100 Hz interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D5	ESW10	Stopwatch timer 10 Hz interrupt enable register					
	D4	ESW1	Stopwatch timer 1 Hz interrupt enable register					
	D3	ETM32	Clock timer 32 Hz interrupt enable register					
	D2	ETM8	Clock timer 8 Hz interrupt enable register					
	D1	ETM2	Clock timer 2 Hz interrupt enable register					
	D0	ETM1	Clock timer 1 Hz interrupt enable register					
00FF24	D7	–	–	–	–	–		"0" when being read
	D6	FSW100	Stopwatch timer 100 Hz interrupt factor flag	(R) Interrupt factor is generated (W) Reset	(R) No interrupt factor is generated (W) No operation	0	R/W	
	D5	FSW10	Stopwatch timer 10 Hz interrupt factor flag					
	D4	FSW1	Stopwatch timer 1 Hz interrupt factor flag					
	D3	FTM32	Clock timer 32 Hz interrupt factor flag					
	D2	FTM8	Clock timer 8 Hz interrupt factor flag					
	D1	FTM2	Clock timer 2 Hz interrupt factor flag					
	D0	FTM1	Clock timer 1 Hz interrupt factor flag					

SWD0–SWD7: 00FF43H

The stopwatch timer data can be read out.
Higher and lower nibbles and BCD digit correspondence are as follows:

SWD0–SWD3:	BCD (1/100sec)
SWD4–SWD7:	BCD (1/10sec)

Since SWD0–SWD7 are exclusively for reading, the write operation is invalid.
At initial reset, the timer data is set to "00H".

SWRST: 00FF42H•D1

Resets the stopwatch timer.

When "1" is written:	Stopwatch timer reset
When "0" is written:	No operation
Reading:	Always "0"

The stopwatch timer is reset by writing "1" to the SWRST. When the stopwatch timer is reset in the RUN status, it restarts immediately after resetting. In the case of the STOP status, the reset data "00H" is maintained.
No operation results when "0" is written to the SWRST.
Since the SWRST is exclusively for writing, it always becomes "0" during reading.

SWRUN: 00FF42H•D0

Controls RUN/STOP of the stopwatch timer.

When "1" is written:	RUN
When "0" is written:	STOP
Reading:	Valid

The stopwatch timer starts up-counting by writing "1" to the SWRUN and stops by writing "0".
In the STOP status, the timer data is maintained until it is reset or set in the next RUN status. Also, when the STOP status changes to the RUN status, the data that was maintained can be used for resuming the count.
At initial reset, the SWRUN is set at "0" (STOP).

PSW0, PSW1: 00FF20H•D2, D3

Sets the priority level of the stopwatch timer interrupt.
The two bits PSW0 and PSW1 are the interrupt priority register corresponding to the stopwatch timer interrupt. Table 5.10.4.2 shows the interrupt priority level which can be set by this register.

Table 5.10.4.2 Interrupt priority level settings

PSW1	PSW0	Interrupt priority level
1	1	Level 3 ($\overline{IRQ3}$)
1	0	Level 2 ($\overline{IRQ2}$)
0	1	Level 1 ($\overline{IRQ1}$)
0	0	Level 0 (None)

At initial reset, this register is set to "0" (level 0).

ESW1, ESW10, ESW100: 00FF22H•D4, D5, D6

Enables or disables the generation of an interrupt for the CPU.

When "1" is written:	Interrupt enabled
When "0" is written:	Interrupt disabled
Reading:	Valid

The ESW1, ESW10 and ESW100 are interrupt enable registers that respectively correspond to the interrupt factors for 1 Hz, 10 Hz and 100 Hz. Interrupts set to "1" are enabled and interrupts set to "0" are disabled.
At initial reset, this register is set to "0" (interrupt disabled).

FSW1, FSW10, FSW100: 00FF24H•D4, D5, D6

Indicates the stopwatch timer interrupt generation status.

When "1" is read:	Interrupt factor present
When "0" is read:	Interrupt factor not present
When "1" is written:	Resets factor flag
When "0" is written:	Invalid

The FSW1, FSW10 and FSW100 are interrupt factor flags that respectively correspond to the interrupts for 1 Hz, 10 Hz and 100 Hz and are set to "1" in synchronization with the falling edge of each signal. When set in this manner, if the corresponding interrupt enable register is set to "1" and the corresponding interrupt priority register is set to a higher level than the setting of interrupt flags (I0 and I1), an interrupt will be generated to the CPU.
Regardless of the interrupt enable register and interrupt priority register settings, the interrupt factor flag will be set to "1" by the occurrence of an interrupt generation condition.
To accept the subsequent interrupt after interrupt generation, re-setting of the interrupt flags (set interrupt flag to lower level than the level indicated by the interrupt priority registers, or execute the RETE instruction) and interrupt factor flag reset are necessary. The interrupt factor flag is reset to "0" by writing "1".
At initial reset, this flag is reset to "0".

5.10.5 Programming notes

(1) The stopwatch timer is actually made to RUN/ STOP in synchronization with the falling edge of the 256 Hz signal after writing to the SWRUN register. Consequently, when "0" is written to the SWRUN, the timer shifts to STOP status when the counter is incremented "1". The SWRUN maintains "1" for reading until the timer actually shifts to STOP status.
Figure 5.10.5.1 shows the timing chart of the RUN/STOP control.

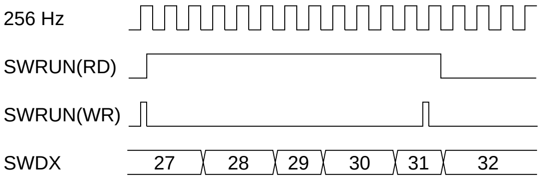


Fig. 5.10.5.1 Timing chart of RUN/STOP control

(2) The SLP instruction is executed when the stopwatch timer is in the RUN status (SWRUN = "1"). The stopwatch timer operation will become unstable when returning from SLEEP status. Therefore, when shifting to SLEEP status, set the clock timer to STOP status (SWRUN = "0") prior to executing the SLP instruction.

5.11 Programmable Timer

5.11.1 Configuration of programmable timer

The E0C88112 has two built-in 8-bit programmable timer systems (timer 0 and timer 1).
Timer 0 and timer 1 are composed of 8-bit presettable down counters and they can be used as 8-bit × 2 channels or 16-bit × 1 channel programmable timer. They also have an event counter function and a pulse width measurement function using the K10 input port terminal.
Figure 5.11.1.1 shows the configuration of the programmable timer.

Programmable setting of the transfer rate is possible, due to the fact that the programmable timer underflow signal can be used as a synchronous clock for the serial interface.
The underflow divided by 1/2 signal can also be output externally from the R27 output port terminal.

5.11.2 Count operation and setting basic mode

Here we will explain the basic operation and setting of the programmable timer.

■ Setting of initial value and counting down

The timers 0 and 1 each have a down counter and reload data register.

The reload data registers RLD00–RLD07 (timer 0) and RLD10–RLD17 (timer 1) are registers that set the initial value of the counter.
By writing "1" to the preset control bit PSET0 (timer 0) or PSET1 (timer 1), the down counter loads the initial value set in the reload register RLD.
Therefore, down-counting is executed from the stored initial value according to the input clock.

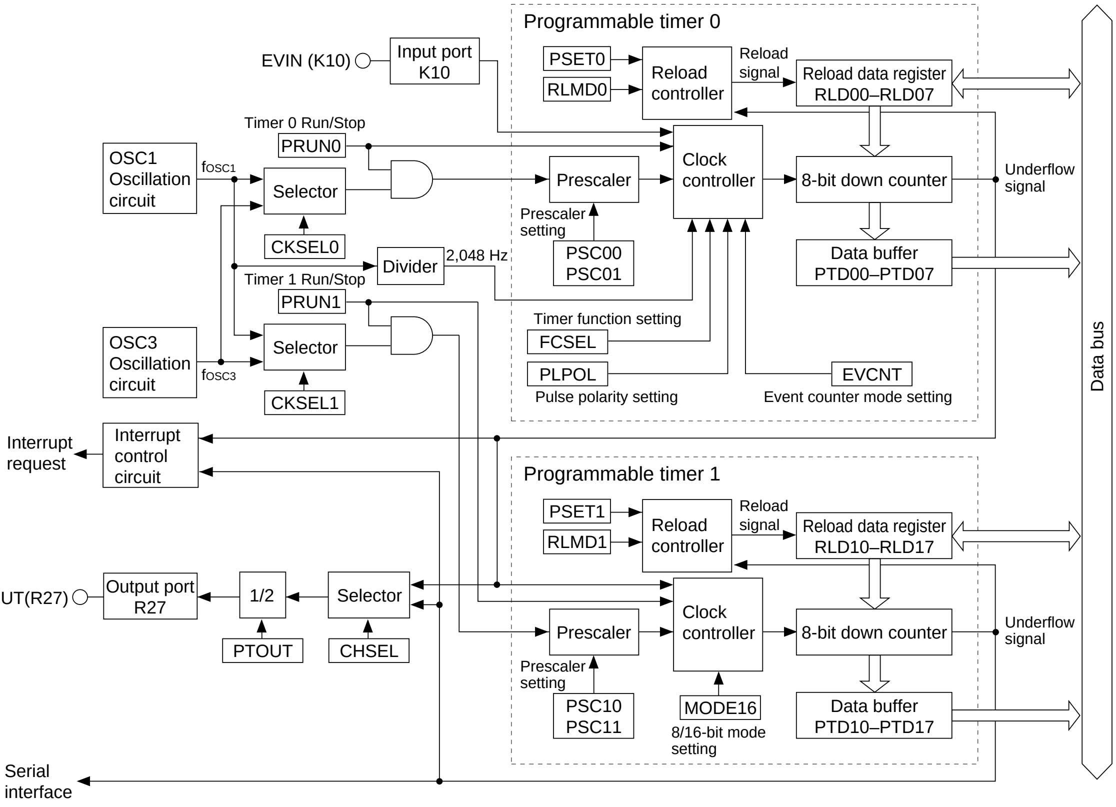


Fig. 5.11.1.1 Configuration of programmable timer

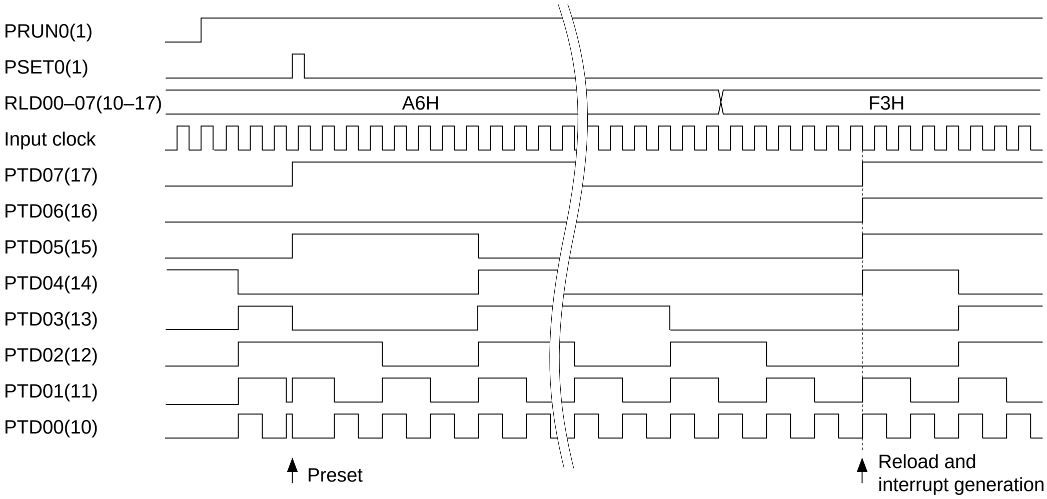


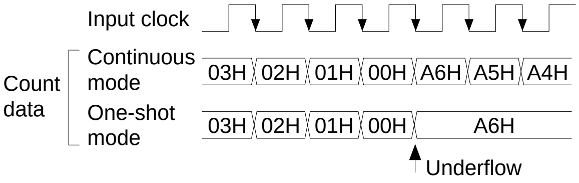
Fig. 5.11.2.1 Basic operation timing of the counter

The registers PRUN0 (timer 0) and PRUN1 (timer 1) are provided to control the RUN/STOP for timers 0 and 1.
After the reload data has been preset into the counter, down-counting is begun by writing "1" to this register. When "0" is written, the clock input is prohibited and the count stops.
The control of this RUN/STOP has no affect on the counter data. The counter data is maintained even during the stoppage of the counter and it can start the count, continuing from that data.

The reading of the counter data can be done through the data buffers PTD00–PTD07 (timer 0) and PTD10–PTD17 (timer 1) with optional timing.
When the down-counting has progressed and an underflow is generated, the counter reloads the initial value set in the reload data register.
This underflow signal controls an interrupt generation, pulse (TOUT signal) output and serial interface clocking, in addition to reloading the counter.

■ Continuous/one-shot mode setting

By writing "1" to the continuous/one-shot mode selection registers CONT0 (timer 0) and CONT1 (timer 1), the programmable timer is set to the continuous mode. In the continuous mode, the initial counter value is automatically loaded when an underflow is generated, and counting is continued. This mode is suitable when programmable intervals are necessary (such as an interrupt and a synchronous clock for the serial interface).
On the other hand, when writing "0" to the registers CONT0 (timer 0) and CONT1 (timer 1), the programmable timer is set to the one-shot mode. The counter loads an initial value and stops when an underflow is generated. At this time, the RUN/STOP control register PRUN0 (timer 0) and PRUN1 (timer 1) are automatically reset to "0". After the counter stops, a one-shot count can be performed once again by writing "1" to registers PRUN0 (timer 0) and PRUN1 (timer 1). This mode is suitable for single time measurement, for example.



When "A6H" is set into reload data register RLD.

Fig. 5.11.2.2 Continuous mode and one-shot mode

■ 8/16-bit mode setting

By writing "0" to the 8/16-bit mode selection register MODE16, timer 0 and timer 1 are set as independent timers in 8-bit × 2 channels. In this mode, timer 0 and timer 1 can be controlled individually and each of them operates independently.
On the other hand, when writing "1" to the register MODE16, timer 0 and timer1 are set as 1 channel 16-bit timer. This is done by setting timer 0 to the lower 8 bits, and timer 1 to the upper 8 bits. The timer is controlled by timer 0's registers. In this case, the control registers for timer 1 are invalid. (PRUN1 is fixed at "0".)

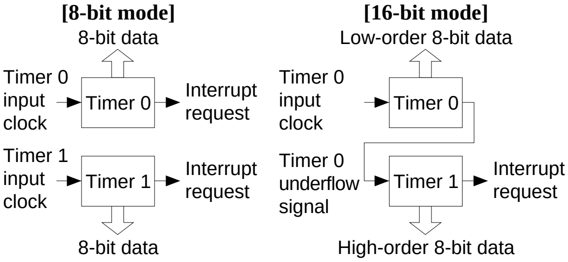


Fig. 5.11.2.3 8/16-bit mode setting and counter configuration

5.11.3 Setting of input clock

Prescalers have been provided for timers 0 and 1. The prescalers generate the input clock for each by dividing the source clock signal from the OSC1 or OSC3 oscillation circuit.
The source clock and the dividing ratio of the prescaler can be selected individually for timer 0 and timer 1 in software.

The input clocks are set by the below sequence.

(1) Selection of source clock

Select the source clock (OSC1 or OSC3) for each prescaler. This is done with the source clock selection registers CKSEL0 (timer 0) and CKSEL1 (timer 1): when "0" is written, OSC1 is selected and when "1" is written, OSC3 is selected. When the 16-bit mode is selected, the source clock is selected by register CKSEL0, and the register CKSEL1 setting becomes invalid.
When the OSC3 oscillation circuit is made the clock source, it is necessary to turn the OSC3 oscillation ON, prior to using the programmable timer.
From the time the OSC3 oscillation circuit is turning ON until oscillation stabilizes, an interval of several msec to several 10 msec is necessary. Consequently, you should allow an adequate waiting time after turning the OSC3 oscillation circuit ON before starting the count of the programmable timer. (The oscillation start time will vary somewhat depending on the oscillator and on external parts. Refer to the oscillation start time example indicated in Chapter 7, "ELECTRICAL CHARACTERISTICS".)
At initial reset, OSC3 oscillation circuit is set to OFF status.

(2) Selection of prescaler dividing ratio

Select the dividing ratio of each prescaler from among 4 types. This selection is done by the prescaler dividing ratio selection registers PSC00/PSC01 (timer 0) and PSC10/PSC11 (timer 1). Setting value and dividing ratio correspondence are shown in Table 5.11.3.1.

Table 5.11.3.1 Selection of prescaler dividing ratio

PSC11 PSC01	PSC10 PSC00	Prescaler dividing ratio
1	1	Source clock / 64
1	0	Source clock / 16
0	1	Source clock / 4
0	0	Source clock / 1

By writing "1" to the register PRUN0 (timer 0) and PRUN1 (timer 1), the source clock is input to the prescaler. Therefore, the clock with selected dividing ratio is input to the timer and the timer starts counting down.
When the 16-bit mode has been selected, the dividing ratio for the source clock is selected by register PSC00/PSC01 and the setting of register PSC10/PSC11 becomes invalid.

5.11.4 Timer mode

The timer mode counts down using the prescaler output as an input clock. In this mode, the programmable timer operates as a timer that obtains fixed cycles using the OSC1 or OSC3 oscillation circuit as a clock source.
See "5.11.2 Count operation and basic mode setting" for basic operation and control, and "5.11.3 Setting input clock" for the clock source and setting of the prescaler.

5.11.5 Event counter mode

Timer 0 includes an even counter function that counts by inputting an external clock (EVIN) to input port K10. This function is selected by writing "1" to the timer 0 counter mode selection register EVCNT.

When the event counter mode is selected, timer 0 operates as an event counter and timer 1 operates as a normal timer in 8-bit mode. In the 16-bit mode, timer 0 and timer 1 operate as 1 channel 16-bit event counter. In the event counter mode, since the timer 0 is clocked externally, the settings of registers PSC00/PSC01 become invalid.

Count down timing can be controlled by either the falling edge or rising edge selected by the timer 0 pulse polarity selection register PLPOL. When "0" is written to the register PLPOL, the falling edge is selected, and when "1" is written, the rising edge is selected. The timing is shown in Figure 5.11.5.1.

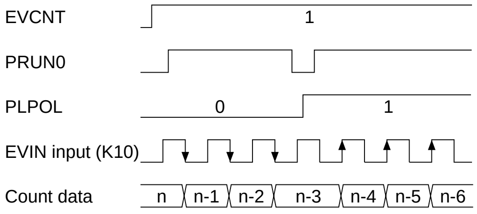
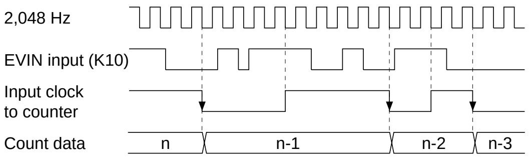


Fig. 5.11.5.1 Timing chart for event counter mode

The event counter also includes a noise rejecter to eliminate noise such as chattering for the external clock (EVIN). This function is selected by writing "1" to the timer 0 function selection register FCSEL. For a reliable count when "with noise rejecter" is selected, you must allow 0.98 msec or more pulse width for both LOW and HIGH levels. (The noise rejecter allows clocking counter at the second falling edge of the internal 2,048 Hz signal after changing the input level of the K10 input port terminal. Consequently, the pulse width that can reliably be rejected is 0.48 msec.)

Figure 5.11.5.2 shows the count down timing with the noise rejecter selected.



When "0" is set into register PLPOL.

Fig. 5.11.5.2 Count down timing with noise rejecter

The event counter mode is the same as the timer mode except that the clock is external (EVIN).

See "5.11.2 Count operation and setting basic mode" for the basic operation and control.

5.11.6 Pulse width measurement timer mode

Timer 0 includes a pulse width measurement function that measures the width of the input signal to the K10 input port terminal. This function is selected by writing "1" to the timer function selection register FCSEL when in the timer mode (EVCNT = "0").

When the pulse width measurement mode is selected, timer 0 operates as an pulse width measurement and timer 1 operates as a normal timer in 8-bit mode. In the 16-bit mode, timer 0 and timer 1 operate as 1 channel 16-bit pulse width measurement.

The level of the input signal (EVIN) for measurement can be changed either a LOW or HIGH level by the timer 0 pulse polarity selection register PLPOL. When "0" is written to register PLPOL, a LOW level width is measured and when "1" is written, a HIGH level width is measured. The timing is shown in Figure 5.11.6.1.

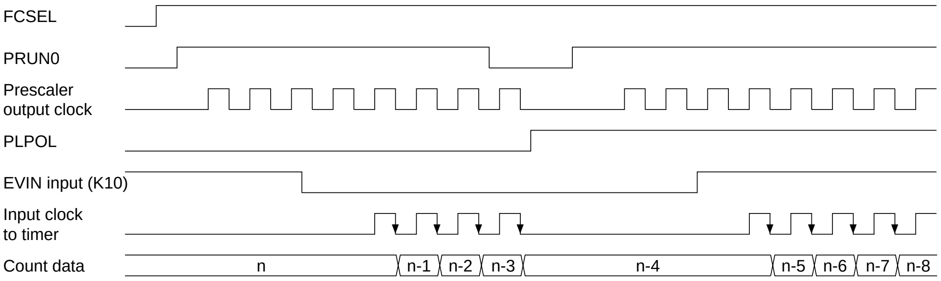


Fig. 5.11.6.1 Timing chart for pulse width measurement timer mode

The pulse width measurement timer mode is the same as the timer mode except that the input clock is controlled by the level of the signal (EVIN) input to the K10 input port terminal.
See "5.11.2 Count operation and setting basic mode" for the basic operation and control.

5.11.7 Interrupt function

The programmable timer can generate an interrupt due to an underflow signal of timer 0 and timer 1. Figure 5.11.7.1 shows the configuration of the programmable timer interrupt circuit.

The respectively corresponding interrupt factor flags FPT0 and FPT1 are set to "1" and an interrupt is generated by an underflow signal of timers 1 and 0. Interrupt can also be prohibited by the setting of the interrupt enable registers EPT0 and EPT1 corresponding to each interrupt flag.

In addition, a priority level of the programmable timer interrupt for the CPU can be optionally set at levels 0 to 3 by the interrupt priority registers PPT0 and PPT1.
For details on the above mentioned interrupt control registers and the operation following generation of an interrupt, see "5.15 Interrupt and Standby Status".

The exception processing vector addresses of each interrupt factor are respectively set as shown below.

Programmable timer 1 interrupt:	000006H
Programmable timer 0 interrupt:	000008H

When the 16-bit mode is selected, the interrupt factor flag FPT0 is not set to "1" and a timer 0 interrupt cannot be generated. (In the 16-bit mode, the interrupt factor flag FPT1 is set to "1" by an underflow of the 16-bit counter.

5.11.8 Setting of TOUT output

The programmable timer can generate a TOUT signal due to an underflow of timer 0 or timer 1. A TOUT signal is the above mentioned underflow divided by 1/2. The timer underflow which is to be used can be selected by the TOUT output channel selection register CHSEL. When writing "0" to register CHSEL, timer 0 is selected and when "1" is written, timer 1 is selected. However, in the 16-bit mode, it is fixed in timer 1 (underflow of the 16-bit timer) and the setting of register CHSEL becomes invalid.
Figure 5.11.8.1 shows the TOUT signal waveform when channel switching.

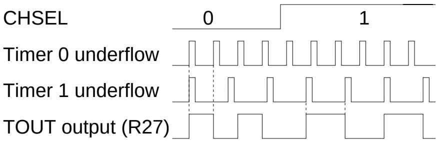


Fig. 5.11.8.1 TOUT signal waveform at channel change

The TOUT signal can be output from the R27 output port terminal and the programmable clock can be supplied to an external device.
The configuration of the output port R27 is shown in Figure 5.11.8.2.

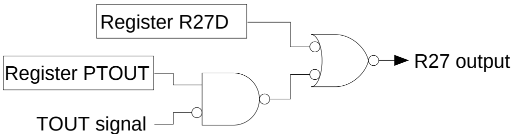


Fig. 5.11.8.2 Configuration of R27

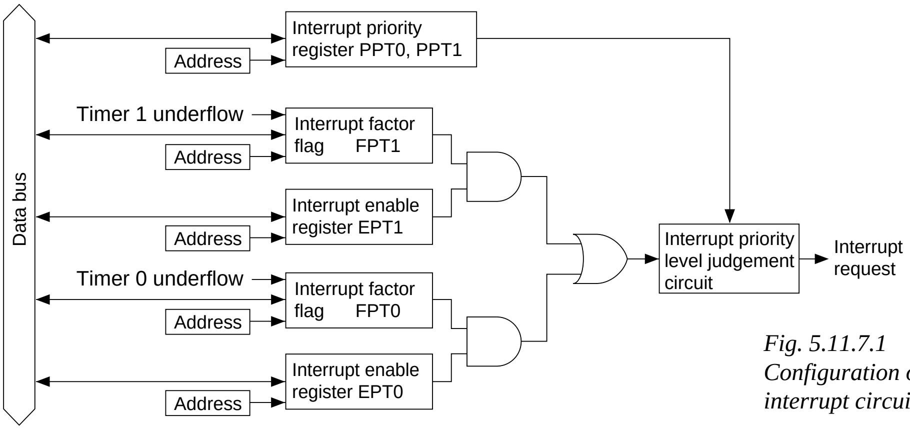

Data bus
Interrupt priority register PPT0, PPT1
Address
Timer 1 underflow
Interrupt factor flag FPT1
Address
Interrupt enable register EPT1
Address
Timer 0 underflow
Interrupt factor flag FPT0
Address
Interrupt enable register EPT0
Address
Interrupt priority level judgement circuit
Interrupt request

Fig. 5.11.7.1
Configuration of programmable timer interrupt circuit

The output control of the TOUT signal is done by register PTOUT. When "1" is set to the PTOUT, the TOUT signal is output from the R27 output port and when "0" is set, HIGH (V_{DD}) level is output. At this time, "1" must always be set in the data register R27D.

Since the TOUT signal is generated asynchronously from the register PTOUT, when the signal is turned ON or OFF by the register setting, a hazard of a 1/2 cycle or less is generated.
Figure 5.11.8.3 shows the output waveform of TOUT signal.

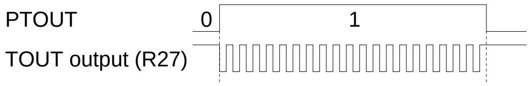


Fig. 5.11.8.3 Output waveform of the TOUT signal

5.11.9 Transmission rate setting of serial interface

The underflow signal of the timer 1 can be used to clock the serial interface.
The transmission rate setting in this case is made in registers PSC1X and PLD1X, and is used to set the count mode to the reload count mode (RLMD1 = "1").
Since the underflow signal of the timer 1 is divided by 1/32 in the serial interface, the value set in register RLD1X which corresponds to the transmission rate is shown in the following expression:

$$\mathbf{RLD1X = fosc / (32*bps*4^{PSC1X}) - 1}$$

fosc: Oscillation frequency (OSC1/OSC3)
bps: Transmission rate
PSC1X: Setting value to the register PSC1X (0–3)

(00H can be set to RLD1X)

Table 5.11.9.1 shows an example of the transmission rate setting when the OSC3 oscillation circuit is used as a clock source.

Table 5.11.9.1 Example of transmission rate setting

Transfer rate (bps)	OSC3 oscillation frequency / Programmable timer settings					
	fosc3 = 3.072 MHz		fosc3 = 4.608 MHz		fosc3 = 4.9152 MHz	
	PSC1X	RLD1X	PSC1X	RLD1X	PSC1X	RLD1X
9,600	0 (1/1)	09H	0 (1/1)	0EH	0 (1/1)	0FH
4,800	0 (1/1)	13H	0 (1/1)	1DH	0 (1/1)	1FH
2,400	0 (1/1)	27H	0 (1/1)	3BH	0 (1/1)	3FH
1,200	0 (1/1)	4FH	0 (1/1)	77H	0 (1/1)	7FH
600	0 (1/1)	9FH	0 (1/1)	EFH	0 (1/1)	FFH
300	1 (1/4)	4FH	1 (1/4)	77H	1 (1/4)	7FH
150	1 (1/4)	9FH	1 (1/4)	EFH	1 (1/4)	FFH

5.11.10 Control of programmable timer

Table 5.11.10.1 shows the programmable timer control bits.

Table 5.11.10.1(a) Programmable timer control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF30	D7	–	–	–	–	–		Constantry "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	MODE16	8/16-bit mode selection	16-bit x 1	8-bit x 2	0	R/W	
	D3	CHSEL	TOUT output channel selection	Timer 1	Timer 0	0	R/W	
	D2	PTOUT	TOUT output control	On	Off	0	R/W	
	D1	CKSEL1	Prescaler 1 source clock selection	f_{OSC3}	f_{OSC1}	0	R/W	
	D0	CKSEL0	Prescaler 0 source clock selection	f_{OSC3}	f_{OSC1}	0	R/W	
00FF31	D7	EVCNT	Timer 0 counter mode selection	Event counter	Timer	0	R/W	
	D6	FCSEL	Timer 0 function selection In timer mode	Pulse width measurement	Normal mode	0	R/W	
			In event counter mode	With noise rejector	Without noise rejector			
	D5	PLPOL	Timer 0 pulse polarity selection Down count timing in event counter mode	Rising edge of K10 input	Falling edge of K10 input	0	R/W	
			In pulse width measurement mode	High level measurement for K10 input	Low level measurement for K10 input			
	D4	PSC01	Timer 0 prescaler dividing ratio selection PSC01 PSC00 Prescaler dividing ratio 1 1 Source clock / 64			0	R/W	
	D3	PSC00	1 0 Source clock / 16 0 1 Source clock / 4 0 0 Source clock / 1			0	R/W	
	D2	CONT0	Timer 0 continuous/one-shot mode selection	Continuous	One-shot	0	R/W	
	D1	PSET0	Timer 0 preset	Preset	No operation	–	W	"0" when being read
	D0	PRUN0	Timer 0 Run/Stop control	Run	Stop	0	R/W	
00FF32	D7	–	–	–	–	–		Constantry "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	PSC11	Timer 1 prescaler dividing ratio selection PSC11 PSC10 Prescaler dividing ratio 1 1 Source clock / 64			0	R/W	
	D3	PSC10	1 0 Source clock / 16 0 1 Source clock / 4 0 0 Source clock / 1			0	R/W	
	D2	CONT1	Timer 1 continuous/one-shot mode selection	Continuous	One-shot	0	R/W	
	D1	PSET1	Timer 1 preset	Preset	No operation	–	W	"0" when being read
	D0	PRUN1	Timer 1 Run/Stop control	Run	Stop	0	R/W	

Table 5.11.10.1(b) Programmable timer control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF33	D7	RLD07	Timer 0 reload data D7 (MSB)	High	Low	1	R/W	
	D6	RLD06	Timer 0 reload data D6					
	D5	RLD05	Timer 0 reload data D5					
	D4	RLD04	Timer 0 reload data D4					
	D3	RLD03	Timer 0 reload data D3					
	D2	RLD02	Timer 0 reload data D2					
	D1	RLD01	Timer 0 reload data D1					
	D0	RLD00	Timer 0 reload data D0 (LSB)					
00FF34	D7	RLD17	Timer 1 reload data D7 (MSB)	High	Low	1	R/W	
	D6	RLD16	Timer 1 reload data D6					
	D5	RLD15	Timer 1 reload data D5					
	D4	RLD14	Timer 1 reload data D4					
	D3	RLD13	Timer 1 reload data D3					
	D2	RLD12	Timer 1 reload data D2					
	D1	RLD11	Timer 1 reload data D1					
	D0	RLD10	Timer 1 reload data D0 (LSB)					
00FF35	D7	PTD07	Timer 0 counter data D7 (MSB)	High	Low	1	R	
	D6	PTD06	Timer 0 counter data D6					
	D5	PTD05	Timer 0 counter data D5					
	D4	PTD04	Timer 0 counter data D4					
	D3	PTD03	Timer 0 counter data D3					
	D2	PTD02	Timer 0 counter data D2					
	D1	PTD01	Timer 0 counter data D1					
	D0	PTD00	Timer 0 counter data D0 (LSB)					
00FF36	D7	PTD17	Timer 1 counter data D7 (MSB)	High	Low	1	R	
	D6	PTD16	Timer 1 counter data D6					
	D5	PTD15	Timer 1 counter data D5					
	D4	PTD14	Timer 1 counter data D4					
	D3	PTD13	Timer 1 counter data D3					
	D2	PTD12	Timer 1 counter data D2					
	D1	PTD11	Timer 1 counter data D1					
	D0	PTD10	Timer 1 counter data D0 (LSB)					

Table 5.11.10.1(c) Programmable timer control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF21	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	PPT1	Programmable timer interrupt priority register	PPT1 PPT0 PK11 PK10 1 1 1 0 0 1 0 0	Priority level Level 3 Level 2 Level 1 Level 0	0	R/W	
	D2	PPT0						
	D1	PK11	K10 and K11 interrupt priority register			0	R/W	
	D0	PK10						
00FF23	D7	EPT1	Programmable timer 1 interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D6	EPT0	Programmable timer 0 interrupt enable register					
	D5	EK1	K10 and K11 interrupt enable register					
	D4	EK0H	K04–K07 interrupt enable register					
	D3	EK0L	K00–K03 interrupt enable register					
	D2	ESERR	Serial I/F (error) interrupt enable register					
	D1	ESREC	Serial I/F (receiving) interrupt enable register					
	D0	ESTRA	Serial I/F (transmitting) interrupt enable register					
00FF25	D7	FPT1	Programmable timer 1 interrupt factor flag	(R) Interrupt factor is generated	(R) No interrupt factor is generated	0	R/W	
	D6	FPT0	Programmable timer 0 interrupt factor flag					
	D5	FK1	K10 and K11 interrupt factor flag					
	D4	FK0H	K04–K07 interrupt factor flag					
	D3	FK0L	K00–K03 interrupt factor flag	(W) Reset	(W) No operation			
	D2	FSERR	Serial I/F (error) interrupt factor flag					
	D1	FSREC	Serial I/F (receiving) interrupt factor flag					
	D0	FSTRA	Serial I/F (transmitting) interrupt factor flag					

MODE16: 00FF30H•D4

Selects the 8/16-bit mode.

When "1" is written:	16 bits × 1 channel
When "0" is written:	8 bits × 2 channels
Reading:	Valid

Select whether timer 0 and timer 1 will be used as 2 channel independent 8-bit timers or as a 1 channel combined 16-bit timer. When "0" is written to MODE16, 8-bit × 2 channels is selected and when "1" is written, 16-bit × 1 channel is selected.
At initial reset, MODE16 is set to "0" (8-bit × 2 channels).

CKSEL0, CKSEL1: 00FF30H•D0, D1

Select the source clock of the prescaler.

When "1" is written:	OSC3 clock
When "0" is written:	OSC1 clock
Reading:	Valid

Select whether the source clock of prescaler 0 will be set to OSC1 or OSC3. When "0" is written to CKSEL0, OSC1 is selected and when "1" is written, OSC3 is selected.
In the same way, the source clock of prescaler 1 is selected by CKSEL1.
When event counter mode has been selected, the setting of the CKSEL0 becomes invalid. In the same way, the CKSEL1 setting becomes invalid when 16-bit mode has been selected.
At initial reset, this register is set to "0" (OSC1 clock).

PSC00, PSC01: 00FF31H•D3, D4
PSC10, PSC11: 00FF32H•D3, D4

Select the dividing ratio of the prescaler.
Two-bit PSC00 and PSC01 is the prescaler dividing ratio selection registers for timer 0, and the two-bit PSC10 and PSC11 correspond to timer 1. The prescaler dividing ratios that can be set by these registers are shown in Table 5.11.10.2.

Table 5.11.10.2 Selection of prescaler dividing ratio

PSC11 PSC01	PSC10 PSC00	Prescaler dividing ratio
1	1	Input clock / 64
1	0	Input clock / 16
0	1	Input clock / 4
0	0	Input clock / 1

When event counter mode has been selected, the setting of the PSC00 and PSC01 becomes invalid. In the same way, the PSC10 and PSC11 setting becomes invalid when 16-bit mode has been selected.
At initial reset, this register is set to "0" (input clock/1).

EVCNT: 00FF31H•D7

Selects the counter mode for the timer 0.

When "1" is written:	Event counter mode
When "0" is written:	Timer mode
Reading:	Valid

Select whether timer 0 will be used as an event counter or a timer. When "1" is written to EVCNT, the event counter mode is selected and when "0" is written, the timer mode is selected.
At initial reset, EVCNT is set to "0" (timer mode).

FCSEL: 00FF31H•D6

Selects the function for each counter mode of timer 0.

• In timer mode

When "1" is written:	Pulse width measurement timer mode
When "0" is written:	Normal mode
Reading:	Valid

In the timer mode, select whether timer 0 will be used as a pulse width measurement timer or a normal timer. When "1" is written to FCSEL, the pulse width measurement mode is selected and the counting is done according to the level of the signal (EVIN) input to the K10 input port terminal. When "0" is written to FCSEL, the normal mode is selected and the counting is not affected by the K10 input port terminal.

• In event counter mode

When "1" is written:	With noise rejecter
When "0" is written:	Without noise rejecter
Reading:	Valid

In the event counter mode, select whether the noise rejecter for the K10 input port terminal will be selected or not.
When "1" is written to FCSEL, the noise rejecter is selected and counting is done by an external clock (EVIN) with 0.98 msec or more pulse width. (The noise rejecter allows clocking counter at the second falling edge of the internal 2,048 Hz signal after changing the input level of the K10 input port terminal. Consequently, the pulse width that can reliably be rejected is 0.48 msec.)
When "0" is written to FCSEL, the noise rejector is not selected and the counting is done directly by an external clock (EVIN) input to the K10 input port terminal.
At initial reset, FCSEL is set to "0".

PLPOL: 00FF31H•D5

Selects the pulse polarity for the K10 input port terminal.

• In event counter mode

When "1" is written:	Rising edge
When "0" is written:	Falling edge
Reading:	Valid

In the event counter mode, select whether the count timing will be set at the falling edge of the external clock (EVIN) input to the K10 input port terminal or at the rising edge. When "0" is written to PLPOL, the falling edge is selected and when "1" is written, the rising edge is selected.

• In pulse width measurement mode

When "1" is written:	High level pulse width measurement
When "0" is written:	LOW level pulse width measurement
Reading:	Valid

In the pulse width measurement mode, select whether the LOW level width of the signal (EVIN) input to the K10 input port terminal will be measured or the HIGH level will be measured. When "0" is written to PLPOL, the LOW level width measurement is selected and when "1" is written, the HIGH level width measurement is selected.
In the normal mode (EVCNT = FCSEL = "0"), the setting of PLPOL becomes invalid.
At initial reset, PLPOL is set to "0".

CONT0, CONT1: 00FF31H•D2, 00FF32H•D2

Select the continuous/one-shot mode.

When "1" is written:	Continuous mode
When "0" is written:	One-shot mode
Reading:	Valid

Select whether timer 0 will be used in the continuous mode or in the one-shot mode.
By writing "1" to CONT0, the programmable timer is set to the continuous mode. In the continuous mode, the initial counter value is automatically loaded when an underflow is generated, and counting is continued. On the other hand, when writing "0" to CONT0, the programmable timer is set to the one-shot mode. The counter loads an initial value and stops when an underflow is generated. At this time, PRUN0 is automatically reset to "0".
In the same way, the continuous/one-shot mode for timer 1 is selected by CONT1. (In the one-shot mode for timer 1, PRUN1 is automatically reset to "0" when the counter underflow is generated.)
At initial reset, this register is set to "0" (one-shot mode).

RLD00–RLD07: 00FF33H
RLD10–RLD17: 00FF34H

Sets the initial value for the counter.

RLD00–RLD07:	Reload data for Timer 0
RLD10–RLD17:	Reload data for Timer 1

The reload data set in this register is loaded into the respective counters and is counted down with that as the initial value.
Reload data is loaded to the counter under two conditions, when "1" is written to PSET0 or PSET1 and when the counter underflow automatically loads.
At initial reset, this register is set to "FFH".

PTD00–PTD07: 00FF35H
PTD10–PTD17: 00FF36H

Data of the programmable timer can be read out.

PTD00–PTD07:	Timer 0 counter data
PTD10–PTD17:	Timer 1 counter data

These bits act as a buffer to maintain the counter data during readout, and the data can be read as optional timing. However, in the 16-bit mode, to avoid a read error, (data error when a borrow from timer 0 to timer 1 is generated in the middle of reading PTD00–PTD07 and PTD10–PTD17), PTD10–PTD17 latches the timer 1 counter data according to the reading of PTD00–PTD07.
The latched status of PTD10–PTD17 is canceled according to the readout of PTD10–PTD17 or when 0.73–1.22 msec (depends on the readout timing) has elapsed. Therefore, in 16-bit mode, be sure to read the counter data of PTD00–PTD07 and PTD10–PTD17 in order.
Since these bits are exclusively for reading, the write operation is invalid.
At initial reset, these bits are set to "FFH".

PSET0, PSET1: 00FF31H•D1, 00FF32H•D1

Presets the reload data to the counter.

When "1" is written:	Preset
When "0" is written:	No operation
Reading:	Always "0"

By writing "1" to PSET0, the reload data in PLD00–PLD07 is preset to the counter of timer 0. When the counter of timer 0 is preset in the RUN status, it restarts immediately after presetting.
In the case of STOP status, the reload data that has been preset is maintained.
No operation results when "0" is written.
In the same way, the reload data in PLD10–PLD17 is preset to the counter of timer 1 by PSET1.
When the 16-bit mode is selected, writing "1" to PSET1 is invalid.
This bit is exclusively for writing, it always becomes "0" during reading.

PRUN0, PRUN1: 00FF31H•D0, 00FF32H•D0

Controls the RUN/STOP of the counter.

When "1" is written:	RUN
When "0" is written:	STOP
Reading:	Valid

The counter of timer 0 starts down-counting by writing "1" to PRUN0 and stops by writing "0".
In the STOP status, the counter data is maintained until it is preset or set in the next RUN status. Also, when the STOP status changes to the RUN status, the data that was maintained can be used for resuming the count.
In the same way, the RUN/STOP of the timer 1 counter is controlled by PRUN1.
When the 16-bit mode is selected, PRUN1 is fixed at "0".
At initial reset and when an underflow is generated in the one-shot mode, this register is set to "0" (STOP).

CHSEL: 00FF30H•D3

Selects the channel of the TOUT signal.

When "1" is written:	Timer 0 underflow
When "0" is written:	Timer 1 underflow
Reading:	Valid

Select whether the timer 0 underflow will be used for the TOUT signal or the timer 1 underflow will be used. When "0" is written to CHSEL, timer 0 is selected and when "1" is written, timer 1 is selected. When the 16-bit mode has been selected, it is fixed to timer 1 (underflow of the 16-bit timer), and setting of CHSEL becomes invalid.
At initial reset, CHSEL is set to "0" (timer 1 underflow).

PTOUT: 00FF30H•D2

Controls the TOUT signal output.

When "1" is written:	TOUT signal output
When "0" is written:	HIGH level (DC) output
Reading:	Valid

PTOUT is the output control register for TOUT signal. When "1" is set, the TOUT signal is output from the output port terminal R27 and when "0" is set, HIGH (VDD) level is output. At this time, "1" must always be set for the data register R27D.
At initial reset, PTOUT is set to "0" (HIGH level output).

PPT0, PPT1: 00FF21H•D2, D3

Sets the priority level of the programmable timer interrupt.
The two bits PPT0 and PPT1 are the interrupt priority register corresponding to the programmable timer interrupt. Table 5.11.10.3 shows the interrupt priority level which can be set by this register.

Table 5.11.10.3 Interrupt priority level settings

PPT1	PPT0	Interrupt priority level
1	1	Level 3 ($\overline{\text{IRQ3}}$)
1	0	Level 2 ($\overline{\text{IRQ2}}$)
0	1	Level 1 ($\overline{\text{IRQ1}}$)
0	0	Level 0 (None)

At initial reset, this register is set to "0" (level 0).

EPT0, EPT1: 00FF23H•D6, D7

Enables or disables the generation of an interrupt for the CPU.

When "1" is written:	Interrupt enabled
When "0" is written:	Interrupt disabled
Reading:	Valid

The EPT0 and EPT1 are interrupt enable registers that respectively correspond to the interrupt factors for timer 0 and timer 1. Interrupts set to "1" are enabled and interrupts set to "0" are disabled.
When the 16-bit mode is selected, setting of EPT0 becomes invalid.
At initial reset, this register is set to "0" (interrupt disabled).

FPT0, FPT1: 00FF25H•D6, D7

Indicates the programmable timer interrupt generation status.

When "1" is read:	Interrupt factor present
When "0" is read:	Interrupt factor not present
When "1" is written:	Resets factor flag
When "0" is written:	Invalid

The FPT0 and FPT1 are interrupt factor flags that respectively correspond to the interrupts for timer 0 and timer 1 and are set to "1" in synchronization with the underflow of each counter.
When set in this manner, if the corresponding interrupt enable register is set to "1" and the corresponding interrupt priority register is set to a higher level than the setting of interrupt flags (I0 and I1), an interrupt will be generated to the CPU.
Regardless of the interrupt enable register and interrupt priority register settings, the interrupt factor flag will be set to "1" by the occurrence of an interrupt generation condition.
To accept the subsequent interrupt after interrupt generation, re-setting of the interrupt flags (set interrupt flag to lower level than the level indicated by the interrupt priority registers, or execute the RETE instruction) and interrupt factor flag reset are necessary. The interrupt factor flag is reset to "0" by writing "1".
When the 16-bit mode is selected, the interrupt factor flag FPT0 is not set to "1" and a timer 0 interrupt cannot be generated. (In the 16-bit mode, the interrupt factor flag FPT1 is set to "1" by an underflow of the 16-bit counter.)
At initial reset, this flag is reset to "0".

5.11.11 Programming notes

(1) The programmable timer is actually made to RUN/STOP in synchronization with the falling edge of the input clock after writing to the PRUN0(1) register. Consequently, when "0" is written to the PRUN0(1), the timer shifts to STOP status when the counter is decremented "1". The PRUN0(1) maintains "1" for reading until the timer actually shifts to STOP status. Figure 5.11.11.1 shows the timing chart of the RUN/STOP control.

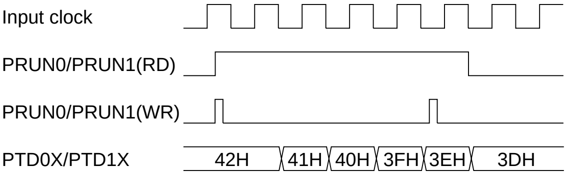


Fig. 5.11.11.1 Timing chart of RUN/STOP control

The event counter mode is excluded from the above note.

(2) The SLP instruction is executed when the programmable timer is in the RUN status (PRUN0(1) = "1"). The programmable timer operation will become unstable when returning from SLEEP status. Therefore, when shifting to SLEEP status, set the clock timer to STOP status (PRUN0(1) = "0") prior to executing the SLP instruction.
In the same way, disable the TOUT signal (PTOUT = "0") to avoid an unstable clock output to the R27 output port terminal.

(3) Since the TOUT signal is generated asynchronously from the register PTOUT, when the signal is turned ON or OFF by the register setting, a hazard of a 1/2 cycle or less is generated.

(4) When the OSC3 oscillation circuit is made the clock source, it is necessary to turn the OSC3 oscillation ON, prior to using the programmable timer.
From the time the OSC3 oscillation circuit is turning ON until oscillation stabilizes, an interval of several msec to several 10 msec is necessary. Consequently, you should allow an adequate waiting time after turning the OSC3 oscillation circuit ON before starting the count of the programmable timer. (The oscillation start time will vary somewhat depending on the oscillator and on external parts. Refer to the oscillation start time example indicated in Chapter 7, "ELECTRICAL CHARACTERISTICS".)
At initial reset, OSC3 oscillation circuit is set to OFF status.

(5) When the 16-bit mode has been selected, be sure to read the counter data in the order of PTD00–PTD07 and PTD10–PTD17. Moreover, the time interval between reading PTD00–PTD07 and PTD10–PTD17 should be 0.73 msec or less.

5.12 Sound Generator

5.12.1 Configuration of sound generator

The E0C88112 has a built-in sound generator for generating BZ (buzzer) signal.
BZ signals generated from the sound generator can be output from the R50 output port terminal.
Aside permitting the respective setting of the buzzer signal frequency and sound level (duty adjustment) to 8 stages, it permits the adding of a digital envelope by means of duty ratio control. It also has a one-shot output function for outputting key operated sounds.
Figure 5.12.1.1 shows the configuration of the sound generator.

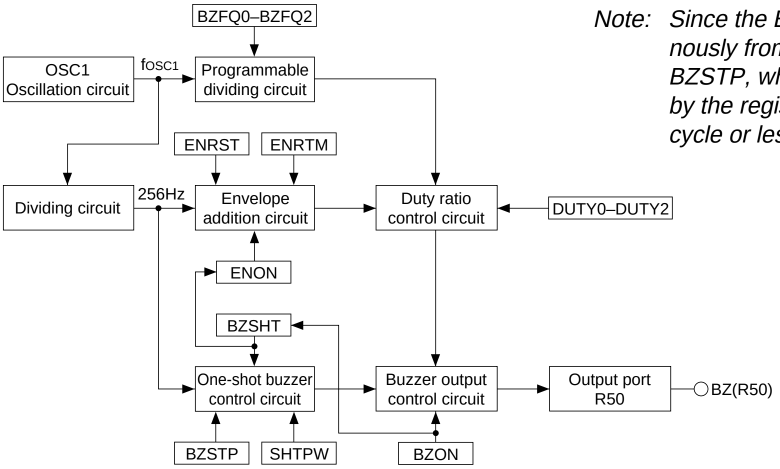

BZFQ0–BZFQ2
OSC1 Oscillation circuit
fOSC1
Programmable dividing circuit
ENRST
ENRTM
Dividing circuit
256Hz
Envelope addition circuit
Duty ratio control circuit
DUTY0–DUTY2
ENON
BZSHT
One-shot buzzer control circuit
Buzzer output control circuit
Output port R50
BZ(R50)
BZSTP
SHTPW
BZON

Fig. 5.12.1.1 Configuration of sound generator

5.12.2 Control of buzzer output

BZ signal can be output from the R50 output port terminal.
The configuration of the output port R50 is shown in Figure 5.12.2.1.

The output control for the BZ signal generated by the sound generator is done by the buzzer output control register BZON, one-shot buzzer trigger bit BZSHT and one-shot buzzer forced stop bit BZSTP. When "1" is set to BZON or BZSHT, the BZ signal is output from the R50 output port terminal and when "0" is set to BZON or "1" is set to BZSTP, the LOW (VSS) level is output. At this time, "0" must always be set for the output data register R50D.
Figure 5.12.2.2 shows the output waveform of the BZ signal.

Note: *Since the BZ signal is generated asynchronously from the registers BZON, BZSHT and BZSTP, when the signal is turned ON or OFF by the register settings, a hazard of a 1/2 cycle or less is generated.*

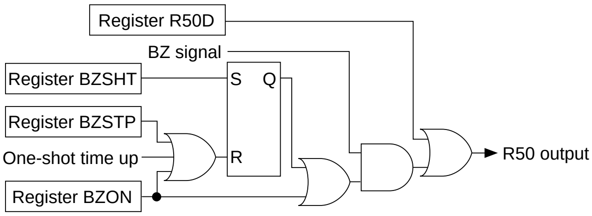


Fig. 5.12.2.1 Configuration of R50

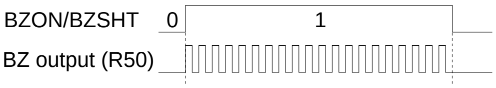


Fig. 5.12.2.2 Output waveform of BZ signal

5.12.3 Setting of buzzer frequency and sound level

The BZ signal is a divided signal using the OSC1 oscillation circuit (32.768 kHz) as the clock source and 8 frequencies can be selected. This selection is done by the buzzer frequency selection register BZFQ0–BZFQ2. The setting value and buzzer frequency correspondence is shown in Table 5.12.3.1.

By selecting the duty ratio of the BZ signal from among 8 types, the buzzer sound level can be adjusted. This selection is made in the duty ratio selection register DUTY0–DUTY2. The setting value and duty ratio correspondence is shown in Table 5.12.3.2.

Table 5.12.3.1 Buzzer signal frequency settings

BZFQ2	BZFQ1	BZFQ0	Buzzer frequency (Hz)
0	0	0	4096.0
0	0	1	3276.8
0	1	0	2730.7
0	1	1	2340.6
1	0	0	2048.0
1	0	1	1638.4
1	1	0	1365.3
1	1	1	1170.3

Table 5.12.3.2 Duty ratio settings

Level	DUTY2	DUTY1	DUTY0	Duty ratio by buzzer frequencies (Hz)			
				4096.0 2048.0	3276.8 1638.4	2730.7 1365.3	2340.6 1170.3
Level 1 (Max)	0	0	0	8/16	8/20	12/24	12/28
Level 2	0	0	1	7/16	7/20	11/24	11/28
Level 3	0	1	0	6/16	6/20	10/24	10/28
Level 4	0	1	1	5/16	5/20	9/24	9/28
Level 5	1	0	0	4/16	4/20	8/24	8/28
Level 6	1	0	1	3/16	3/20	7/24	7/28
Level 7	1	1	0	2/16	2/20	6/24	6/28
Level 8 (Min)	1	1	1	1/16	1/20	5/24	5/28

Duty ratio refers to the ratio of pulse width to the pulse cycle; given that HIGH level output time is TH, and low level output time is TL the BZ signal becomes TH/(TH+TL).

When DUTY0–DUTY2 have all been set to "0", the duty ratio becomes maximum and the sound level also becomes maximum. Conversely, when DUTY0–DUTY2 have all been set to "1", the duty ratio becomes minimum and the sound level also becomes minimum.

Note that the duty ratio setting differ depending on frequency. See Table 5.12.3.2.

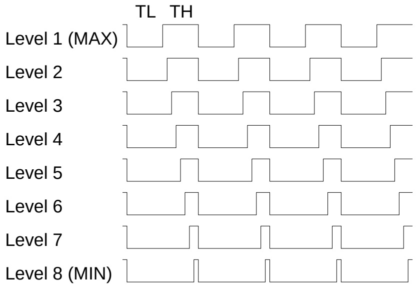


Fig. 5.12.3.1 Duty ratio of buzzer signal waveform

Note: When using the digital envelope, the DUTY0–DUTY2 setting becomes invalid.

5.12.4 Digital envelope

A digital envelope with duty control can be added to the BZ signal.

The envelope can be realized by staged changing of the same duty ratio as detailed in Table 5.12.3.2 in the preceding section from level 1 (maximum) to level 8 (minimum).

The addition of an envelope to the buzzer signal can be done by writing "1" to the envelope control register ENON. When "0" is written, the duty ratio is set at the level selected in DUTY0–DUTY2.

By writing "1" to ENON to turn the buzzer output ON (writing "1" to BZON), a BZ signal with a level 1 duty ratio is output, and then the duty ratio can be attenuated in stages to level 8. The attenuated envelope can be returned to level 1 by writing "1" to the envelope reset bit ENRST. When attenuated to level 8, the duty level remains at level 8 until the buzzer output is turned OFF (writing "0" to BZON) or writing "1" to ENRST.

The stage changing time for the envelope level can be selected either 125 msec or 62.5 msec by the envelope attenuation time selection register ENRTM. Figure 5.12.4.1 shows the timing chart of the digital envelope.

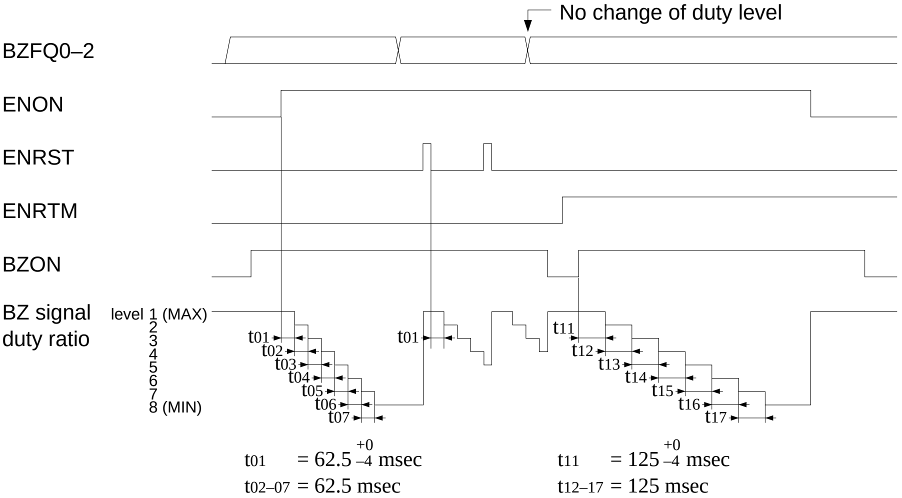


Fig. 5.12.4.1 Timing chart of digital envelope

5.12.5 One-shot output

The sound generator has a built-in one-shot output function for outputting a short duration buzzer signal for key operation sounds and similar effects. Either 125 msec or 31.25 msec can be selected by the one-shot buzzer duration selection register SHTPW for buzzer signal output time.
The output control of the one-shot buzzer is done by writing "1" to the one-shot buzzer trigger BZSHT, then the BZ signal is output in synchronization with the internal 256 Hz signal from the R50 output port terminal. Thereafter, when the set time has elapsed, the BZ signal in synchronization with the 256 Hz signal automatically goes OFF in the same manner.
The BZSHT can be read to determine status. When BZSHT is "1", it indicates a BUSY status (during one-shot output) and when BZSHT is "0", it indicates a READY status (during stop).
When you want to turn the BZ signal OFF prior to the elapse of the set time, the BZ signal can be immediately stopped (goes OFF in asynchonization with 256 Hz signal) by writing "1" to the one-shot forced stop bit BZSTP.

Since the one-shot output has a short duration, an envelope cannot be added. (When "1" is written to BZSHT, ENON is automatically reset to "0".) Consequently, only the frequency and sound level can be set for one-shot output.

The control for the one-shot output is invalid during normal buzzer output.

Figure 5.12.5.1 shows the timing chart of the one-shot output.

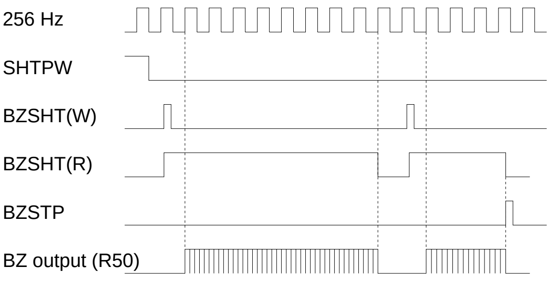


Fig. 5.12.5.1 Timing chart of one-shot output

5.12.6 Control of sound generator

Table 5.12.6.1 shows the sound generator control bits.

Table 5.12.6.1 Sound generator control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF44	D7	–	–	–	–	–		Constantry "0" when being read
	D6	BZSTP	One-shot buzzer forcibly stop	Forcibly stop	No operation	–	W	
	D5	BZSHT	One-shot buzzer trigger/status R W	Busy Trigger	Ready No operation	0	R/W	
	D4	SHTPW	One-shot buzzer duration width selection	125 msec	31.25 msec	0	R/W	
	D3	ENRTM	Envelope attenuation time	1 sec	0.5 sec	0	R/W	
	D2	ENRST	Envelope reset	Reset	No operation	–	W	"0" when being read
	D1	ENON	Envelope On/Off control	On	Off	0	R/W	*1
	D0	BZON	Buzzer output control	On	Off	0	R/W	
00FF45	D7	–	–	–	–	–		"0" when being read
	D6	DUTY2	Buzzer signal duty ratio selection DUTY2–1 Buzzer frequency (Hz) 2 1 0 4096.0 3276.8 2730.7 2340.6 2048.0 1638.4 1365.3 1170.3			0	R/W	
	D5	DUTY1	0 0 0 8/16 8/20 12/24 12/28 0 0 1 7/16 7/20 11/24 11/28 0 1 0 6/16 6/20 10/24 10/28 0 1 1 5/16 5/20 9/24 9/28			0	R/W	
	D4	DUTY0	1 0 0 4/16 4/20 8/24 8/28 1 0 1 3/16 3/20 7/24 7/28 1 1 0 2/16 2/20 6/24 6/28 1 1 1 1/16 1/20 5/24 5/28			0	R/W	
	D3	–	–			–		"0" when being read
	D2	BZFQ2	Buzzer frequency selection BZFQ2 BZFQ1 BZFQ0 Frequency (Hz) 0 0 0 4096.0			0	R/W	
	D1	BZFQ1	0 0 1 3276.8 0 1 0 2730.7 0 1 1 2340.6 1 0 0 2048.0			0	R/W	
	D0	BZFQ0	1 0 1 1638.4 1 1 0 1365.3 1 1 1 1170.3			0	R/W	

*1 Reset to "0" during one-shot output.

BZON: 00FF44H•D0

Controls the BZ signal output.

When "1" is written: BZ signal output
When "0" is written: LOW level (DC) output
Reading: Valid

BZON is the output control register for BZ signal. When "1" is set, the BZ signal is output from the output port terminal R50 and when "0" is set, LOW (VSS) level is output. At this time, "0" must always be set for the data register R50D.
At initial reset, BZON is set to "0" (LOW level output).

BZFQ0–BZFQ2: 00FF45H•D0–D2

Selects the BZ signal frequency.

Table 5.12.6.2 Buzzer frequency settings

BZFQ2	BZFQ1	BZFQ0	Buzzer frequency (Hz)
0	0	0	4096.0
0	0	1	3276.8
0	1	0	2730.7
0	1	1	2340.6
1	0	0	2048.0
1	0	1	1638.4
1	1	0	1365.3
1	1	1	1170.3

The buzzer frequency can be selected from among the above 8 types that have divided the OSC1 clock. At initial reset, this register is set at "0" (4096.0 Hz).

DUTY0–DUTY2: 00FF45H•D4–D6

Selects the duty ratio of the BZ signal.

Table 5.12.6.3 Duty ratio settings

Level	DUTY2	DUTY1	DUTY0	Duty ratio by buzzer frequencies (Hz)			
				4096.0	3276.8	2730.7	2340.6
				2048.0	1638.4	1365.3	1170.3
Level 1 (Max)	0	0	0	8/16	8/20	12/24	12/28
Level 2	0	0	1	7/16	7/20	11/24	11/28
Level 3	0	1	0	6/16	6/20	10/24	10/28
Level 4	0	1	1	5/16	5/20	9/24	9/28
Level 5	1	0	0	4/16	4/20	8/24	8/28
Level 6	1	0	1	3/16	3/20	7/24	7/28
Level 7	1	1	0	2/16	2/20	6/24	6/28
Level 8 (Min)	1	1	1	1/16	1/20	5/24	5/28

The buzzer sound level can be adjusted by selecting the duty ratio from among the above 8 types. However, when the envelope has been set to ON (ENON = "1"), this setting becomes invalid.
At initial reset, this register is set to "0" (level 1).

ENRST: 00FF44H•D2

Resets the envelope.

When "1" is written:	Reset
When "0" is written:	No operation
Reading:	Always "0"

The envelope is reset by writing "1" to ENRST and the duty ratio returns to level 1 (maximum).
Writing "0" to ENRST and writing "1" when an envelope has not been added become invalid. Since ENRST is exclusively for writing, it always becomes "0" during reading.

ENON: 00FF44H•D1

Controls the addition of an envelope to the BZ signal.

When "1" is written:	ON
When "0" is written:	OFF
Reading:	Valid

By writing "1" to ENON, an envelope can be added to BZ signal output. When "0" is written, an envelope is not added and the BZ signal is fixed at the duty ratio selected in DUTY0–DUTY2.
At initial reset and when "1" is written to BZSHT, ENON is set to "0" (OFF).

ENRTM: 00FF44H•D3

Selects the envelope attenuation time that is added to the BZ signal.

When "1" is written:	1.0 sec (125 msec × 7 = 875 msec)
When "0" is written:	0.5 sec (62.5 msec × 7 = 437.5 msec)
Reading:	Valid

The attenuation time of the digital envelope is determined by the time for changing the duty ratio. The duty ratio is changed in 125 msec (8 Hz) units when "1" is written to ENRTM and in 62.5 msec (16 Hz) units, when "0" is written.
This setting becomes invalid when an envelope has been set to OFF (ENON = "0").
At initial reset, ENRTM is set to "0" (0.5 sec).

SHTPW: 00FF44H•D4

Selects the output duration width of the one-shot buzzer.

When "1" is written:	125 msec
When "0" is written:	31.25 msec
Reading:	Valid

The one-shot buzzer output duration width is set to 125 msec when "1" is written to SHTPW and 62.5 msec, when "0" is written.
At initial reset, SHTPW is set to "0" (31.25 msec).

BZSHT: 00FF44H•D5

Controls the one-shot buzzer output.

When "1" is written:	Trigger
When "0" is written:	No operation
When "1" is read:	Busy
When "0" is read:	Ready

Writing "1" into BZSHT causes the one-shot output circuit to operate and the BZ signal to be output. The buzzer output is automatically turned OFF after the time set by SHTPW has elapsed. At this time, "0" must always be set for the data register R50D.
The one-shot output is only valid when the normal buzzer output is OFF (BZON = "0") status. The trigger is invalid during ON (BZON = "1") status.
When a re-trigger is assigned during a one-shot output, the one-shot output time set with SHTPW is measured again from that point. (time extension)
The operation status of the one-shot output circuit can be confirmed by reading BZSHT, when the one-shot output is ON (busy), BZSHT reads "1" and when the output is OFF (ready), it reads "0".
At initial reset, BZSHT is set to "0" (ready).

BZSTP: 00FF44H•D6

Forcibly stops the one-shot buzzer output.

When "1" is written:	Forcibly stop
When "0" is written:	No operation
Reading:	Constantly "0"

By writing "1" into BZSTP, the one-shot buzzer output can be stopped prior to the elapsing of the time set with SHTPW.
Writing "0" is invalid and writing "1" except during one-shot output is also invalid.
When "1" is written to BZSHT and BZSTP simultaneously, BZSTP takes precedence and one-shot output becomes stop status.
Since BZSTP is for writing only, during readout it is constantly set to "0".

5.12.7 Programming notes

(1) Since the BZ signal is generated asynchronously from the register BZON, when the signal is turned ON or OFF by the register setting, a hazard of a 1/2 cycle or less is generated.

(2) The SLP instruction has executed when the BZ signal is in the enable status (BZON = "1" or BZSHT = "1"), an unstable clock is output from the R50 output port terminal at the time of return from the SLEEP status. Consequently, when shifting to the SLEEP status, you should set the BZ signal to the disable status (BZON = BZSHT = "0") prior to executing the SLP instruction.

(3) The one-shot output is only valid when the normal buzzer output is OFF (BZON = "0") status. The trigger is invalid during ON (BZON = "1") status.

5.13 Analog Comparator

5.13.1 Configuration of analog comparator

The E0C88112 has an MOS input analog comparator built into two channels. The respective analog comparators have two differential input terminals (inverted input terminal CMPMx and non-inverted input terminal CMPPx) that are available for general purpose use.

Figure 5.13.1.1 shows the configuration of the analog comparator.

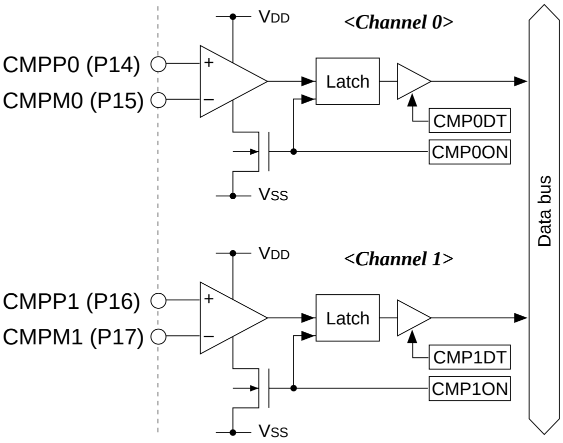


Fig. 5.13.1.1 Configuration of analog comparator

Since the input terminals of the analog comparator CMPP0, CMPM0, CMPP1 and CMPM1 are common to I/O ports P14–P17, when using as the input terminal for the analog comparator, "0" (input mode) must be written to I/O control registers IOC14–IOC17.

Table 5.13.1.1 Input terminal configuration

Terminal	When analog comparator is used
P14	CMPP0
P15	CMPM0
P16	CMPP1
P17	CMPM1

5.13.2 Mask option

Since the input terminals of the analog comparator are common to the I/O ports, the mask option for the I/O port corresponding to the channel to be used must be set to "Gate direct".

I/O ports pull-up resistor
P14 (CMPP0) ... □ With resistor ☑ Gate direct
P15 (CMPM0) .. □ With resistor ☑ Gate direct
P16 (CMPP1) ... □ With resistor ☑ Gate direct
P17 (CMPM1) .. □ With resistor ☑ Gate direct

* "✓" above shows an example of both channels being used.

5.13.3 Analog comparator operation

By writing "1" to the analog comparator control register CMPxON, the analog comparator goes ON, and the analog comparator starts comparing the external voltages that have been input to the two differential input terminals CMPPx and CMPMx. The result can be read from the comparator comparison result detection bit CMPxDT through the latch and when CMPPx (+) > CMPMx (-), it is "1" and when CMPPx (+) < CMPMx (-), it is "0". After the analog comparator has been turned ON, a maximum time of 3 msec is necessary until output stabilizes. Consequently, you should allow an adequate waiting time after turning the analog comparator ON, before reading the comparison result.

When the analog comparator is turned OFF, the comparison result at that point will be latched and the concerned data can be read thereafter, until the analog comparator is turned ON.

You should turn the analog comparator OFF, when it is not necessary, so as to reduce current consumption.

See "7 ELECTRICAL CHARACTERISTICS" for the input voltage range.

Note: Since the input terminals of the analog comparator are common to the I/O ports, the I/O control registers (IOC14–IOC17) corresponding to the channel to be used must be set to the input mode.

5.13.4 Control of analog comparator

Table 5.13.4.1 shows the analog comparator control bits.

Table 5.13.4.1 Analog comparator control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF13	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	CMP1ON	Comparator 1 On/Off control	On	Off	0	R/W	
	D2	CMP0ON	Comparator 0 On/Off control	On	Off	0	R/W	
	D1	CMP1DT	Comparator 1 data	+ > -	+ < -	0	R	
	D0	CMP0DT	Comparator 0 data	+ > -	+ < -	0	R	

CMP0ON, CMP1ON: 00FF13H•D2, D3

Controls the analog comparator ON/OFF.

When "1" is written:	ON
When "0" is written:	OFF
Reading:	Valid

The analog comparator 0 goes ON by writing "1" to CMP0ON and goes OFF, when "0" is written. The analog comparator 1 can be controlled with CMP1ON in the same way.
At initial reset, this register is set "0" (OFF).

CMP0DT, CMP1DT: 00FF13H•D0, D1

The comparison result of the analog comparator can be read out.

When "1" is read:	CMPPx (+) > CMPMx (-)
When "0" is read:	CMPPx (+) < CMPMx (-)
Writing:	Invalid

The result of analog comparator 0 can be read from CMP0DT. When the status of external voltage input to differential input terminals CMPP0 and CMPM0 is CMPP0 (+) > CMPM0 (-), CMP0DT becomes "1" and when it is CMPP0 (+) < CMPM0 (-), CMP0DT becomes "0".
As the same way, the comparison result between CMPP1 and CMPM1 can be read from CMP1DT.
When the analog comparator is turned OFF, the latched result immediately prior to going OFF is read out.
At initial reset, this bit is set to "1".

5.13.5 Programming notes

(1) To reduce current consumption, turn the analog comparator OFF (CMP0ON = CMP1ON = "0") when it is not necessary.

(2) After the analog comparator has been turned ON, a maximum time of 3 msec is necessary until output stabilizes. Consequently, you should allow an adequate waiting time after turning the analog comparator ON, before reading the comparison result.

(3) Since the input terminals of the analog comparator are common to the I/O ports, the I/O control registers (IOC14–IOC17) corresponding to the channel to be used must be set to the input mode.

5.14 Supply Voltage Detection (SVD) Circuit

5.14.1 Configuration of SVD circuit

The E0C88112 has a built-in supply voltage detection (SVD) circuit configured with a 4-bit successive approximation A/D converter.
The SVD circuit has 16 sampling levels (level 0–level 15) for supply voltage, and this can be controlled by software.
In addition, an initial reset signal can be generated when the supply voltage drops to level 0 or less. This is selected by the mask option.
Figure 5.14.1.1 shows the configuration of the SVD circuit.

5.14.2 Operation of SVD circuit

■ Sampling control of the SVD circuit

The SVD circuit has two operation modes: continuous sampling and 1/4 Hz auto-sampling mode. Operation mode selection is done by the SVD control registers SVDON and SVDSP as shown in Table 5.14.2.1. When both bits of SVDON and SVDSP are set to "1", continuous sampling is selected.

Table 5.14.2.1 Correspondence between control register and operation mode

SVDON	SVDSP	Operating mode
0	0	SVD circuit OFF
0	1	1/4 Hz auto-sampling ON
1	×	Continuous sampling ON

In both operation modes, reading SVDON can confirm whether the SVD circuit is operating (BUSY) or on standby (READY); "1" indicates BUSY and "0" indicates READY.
When executing an SLP instruction while the SVD circuit is operating, the stop operation of the OSC1 oscillation circuit is kept waiting until the sampling is completed. The two bits of SVDON and SVDSP are automatically reset to "0" by hardware while waiting for completion of sampling.
To reduce current consumption, turn the SVD circuit OFF when it is not necessary.

■ Detection result

The SVD circuit A/D converts the supply voltage (VDD–VSS) by 4-bit resolution and sets the result thereof into the SVD0–SVD3 register.
The data in SVD0–SVD3 correspond to the detection levels as shown in Table 5.14.2.2 and the detection data is maintained until the next sampling.

For the correspondence between the detection level and the supply voltage, see "7 ELECTRICAL CHARACTERISTICS".
An interval of 7.8 msec (f_{OSC1} = 32.768 kHz) is required from the start of supply voltage sampling by the SVD circuit to completion by writing the result into SVD0–SVD3. Therefore, when reading SVD0–SVD3 before sampling is finished, the previous result will be read.

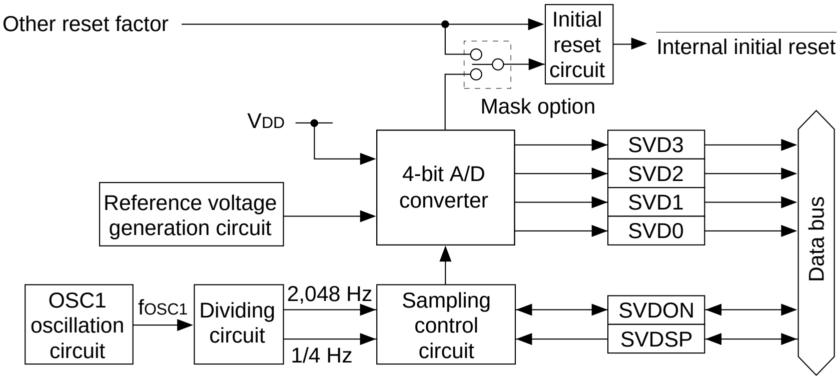


Fig.5.14.1.1 Configuration of SVD circuit

Table 5.14.2.2 Supply voltage detection results

SVD3	SVD2	SVD1	SVD0	Detection level
1	1	1	1	Level 15
1	1	1	0	Level 14
1	1	0	1	Level 13
1	1	0	0	Level 12
1	0	1	1	Level 11
1	0	1	0	Level 10
1	0	0	1	Level 9
1	0	0	0	Level 8
0	1	1	1	Level 7
0	1	1	0	Level 6
0	1	0	1	Level 5
0	1	0	0	Level 4
0	0	1	1	Level 3
0	0	1	0	Level 2
0	0	0	1	Level 1
0	0	0	0	Level 0

■ Timing of sampling

Next, we will explain the timing for two operation modes.

(1) Continuous sampling mode

This mode is selected when "1" is written to SVDON and sampling of the supply voltage is done continuously in 7.8 msec cycles.

The SVD circuit starts operation in synchronization with the internal 2,048 Hz signal and performs one sampling in 16 clock cycles.

The sampling is done continuously without setting the standby time and the result is latched to SVD0–SVD3 in every 16 clock cycles.

Cancellation of continuous sampling is done by writing "0" to SVDON. The SVD circuit maintains ON status until completion of sampling and then goes OFF.

After writing "0" to SVDON, SVDON reads "1" until the SVD circuit actually goes OFF.

Figure 5.14.2.1 shows the timing chart of the continuous sampling.

(2) 1/4 Hz auto-sampling mode

This mode is selected when "0" is written to SVDON and "1" is written to SVDSP. In this case, supply voltage sampling is done in every 4 seconds.

The sampling time is 7.8 msec as in continuous sampling, and the result in SVD0–SVD3 is updated every 4 seconds.

Cancellation of 1/4 Hz auto-sampling is done by writing "0" to SVDSP. If the SVD circuit is sampling, SVD circuit waits until completion and then turns OFF. In addition, "1" is read from SVDON while the SVD circuit is sampling.

Figure 5.14.2.2 shows the timing chart of the 1/4 Hz auto-sampling.

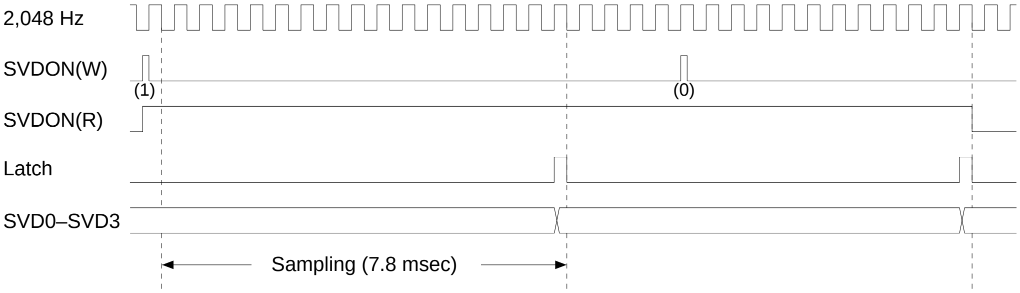


Fig. 5.14.2.1 Timing chart of continuous sampling

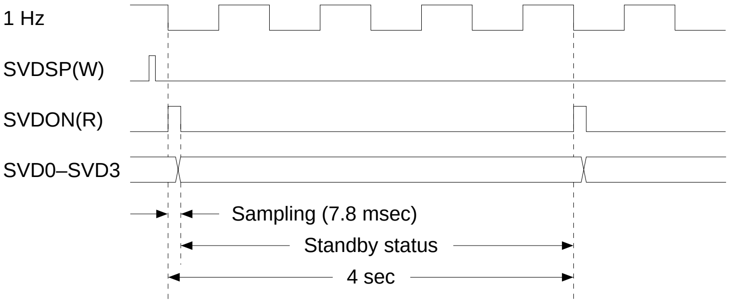


Fig. 5.14.2.2 Timing chart of 1/4 Hz auto-sampling

■ Reset function at low voltage detection

To avoid CPU runaway due to a supply voltage drop, an initial reset function when the supply voltage drops to level 0 or less can be selected by the mask option.

The SVD circuit shifts to continuous sampling status when it detects level 0 (SVD3–SVD0 = 0000B) four successive times. At this time, the internal initial reset signal is generated. The reset status continues until the supply voltage returns to level 2 (SVD3–SVD0 = 0010B) or higher.

When the reset status is canceled by the restoration of the supply voltage, the SVD circuit returns to its previous status. Continuous sampling status continues in case of the previous status was continuous sampling. Then CPU starts the reset exception processing.

Figure 5.14.2.3 shows the timing chart of the initial reset signal generation. (Example when using 1/4 Hz auto-sampling.)

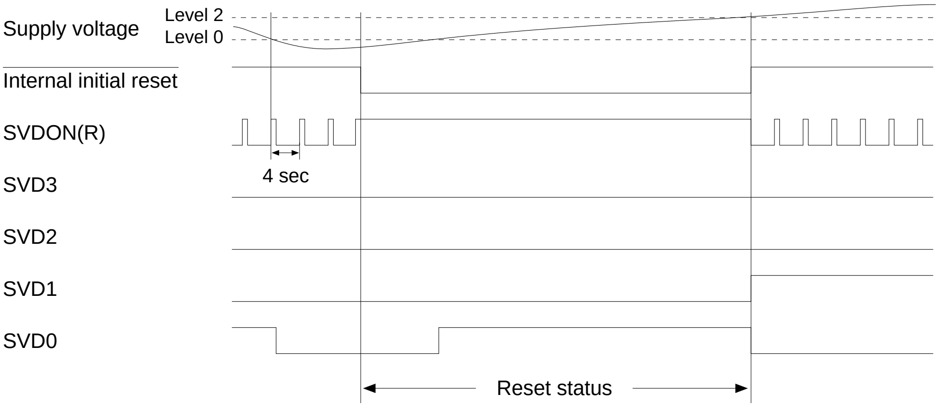


Fig. 5.14.2.3 Timing chart of the initial reset signal generation

5.14.3 Control of SVD circuit

Table 5.14.3.1 shows the SVD circuit control bits.

Table 5.14.3.1 SVD circuit control bits

Address	Bit	Name	Function		1	0	SR	R/W	Comment
00FF12	D7	–	–		–	–	–		Constantry "0" when being read
	D6	–	–		–	–	–		
	D5	SVDSP	SVD auto-sampling control		On	Off	0	R/W	These registers are reset to "0" when SLP instruction is executed.
	D4	SVDON	SVD continuous sampling control/status	R W	Busy On	Ready Off	1→0*1 0	R/W	
	D3	SVD3	SVD detection level				X	R	*2
	D2	SVD2	SVD3 SVD2 SVD1 SVD0 Detection level 1 1 1 1 Level 15				X	R	
	D1	SVD1	1 1 1 0 Level 14 : : : : :				X	R	
	D0	SVD0	0 0 0 0 Level 0				X	R	

*1 After initial reset, this status is set "1" until conclusion of hardware first sampling.

*2 Initial values are set according to the supply voltage detected at first sampling by hardware. Until conclusion of first sampling, SVD0–SVD3 data are undefined.

SVDON: 00FF12H•D4

Controls the turning ON/OFF of the continuous sampling mode.

When "1" is written: Continuous sampling ON
When "0" is written: Continuous sampling OFF

When "1" is read: BUSY
When "0" is read: READY

The continuous sampling mode goes ON when "1" is written to SVDON and goes OFF, when "0" is written.
In the ON status, sampling of the supply voltage is done continuously in 7.8 msec cycles and the detection result is latched to SVD0–SVD3.
SVDON can be read, and "1" indicates SVD circuit operation (BUSY) and "0" indicates standby (READY).
At initial reset and in the SLEEP status, SVDON is set to "0" (continuous sampling OFF/READY).

SVDSP: 00FF12H•D5

Controls the turning ON/OFF of the 1/4 Hz auto-sampling mode.

When "1" is written: Auto-sampling ON
When "0" is written: Auto-sampling OFF
Reading: Valid

The 1/4 Hz auto-sampling mode goes ON when "1" is written to SVDSP and goes OFF, when "0" is written.
In the ON status, sampling is done in every 4 seconds and "1" is read from SVDON during the actual sampling period (7.8 msec).
At initial reset and in the SLEEP status, SVDSP is set to "0" (auto-sampling OFF).

SVD0–SVD3: 00FF12H•D0–D3

The detection result of the SVD is set.
The reading data correspond to the detection levels as shown in Table 5.14.3.2 and the data is maintained until the next sampling.

Table 5.14.3.2 Supply voltage detection results

SVD3	SVD2	SVD1	SVD0	Detection level
1	1	1	1	Level 15
1	1	1	0	Level 14
1	1	0	1	Level 13
1	1	0	0	Level 12
1	0	1	1	Level 11
1	0	1	0	Level 10
1	0	0	1	Level 9
1	0	0	0	Level 8
0	1	1	1	Level 7
0	1	1	0	Level 6
0	1	0	1	Level 5
0	1	0	0	Level 4
0	0	1	1	Level 3
0	0	1	0	Level 2
0	0	0	1	Level 1
0	0	0	0	Level 0

For the correspondence between the detection level and the supply voltage, see "7 ELECTRICAL CHARACTERISTICS".
The initial value at initial reset is set according to the supply voltage detected at first sampling by hardware. Data of this bit is undefined until this sampling is completed.

5.14.4 Programming notes

(1) To reduce current consumption, turn the SVD circuit OFF (SVDON = SVDSP = "0") when it is not necessary.

(2) When executing an SLP instruction while the SVD circuit is operating, the stop operation of the OSC1 oscillation circuit is kept waiting until the sampling is completed. The two bits of SVDON and SVDSP are automatically reset to "0" by hardware while waiting for completion of sampling.

5.15 Interrupt and Standby Status

■ Types of interrupts

Six systems and 15 types of interrupts have been provided for the E0C88112.

External interrupt
- K00–K07 input interrupt (2 types)
- K10 and K11 input interrupt (1 type)

Internal interrupt
- Clock timer interrupt (4 types)
- Stopwatch interrupt (3 types)
- Programmable timer interrupt (2 types)
- Serial interface interrupt (3 types)

An interrupt factor flag that indicates the generation of an interrupt factor and an interrupt enable register that sets enable/disable for interrupt requests have been provided for each interrupt and interrupt generation can be optionally set for each factor.
In addition, an interrupt priority register has been provided for each system of interrupts and the priority of interrupt processing can be set to 3 levels in each system.
Figure 5.15.1 shows the configuration of the interrupt circuit.

Refer to the explanations of the respective peripheral circuits for details on each interrupt.

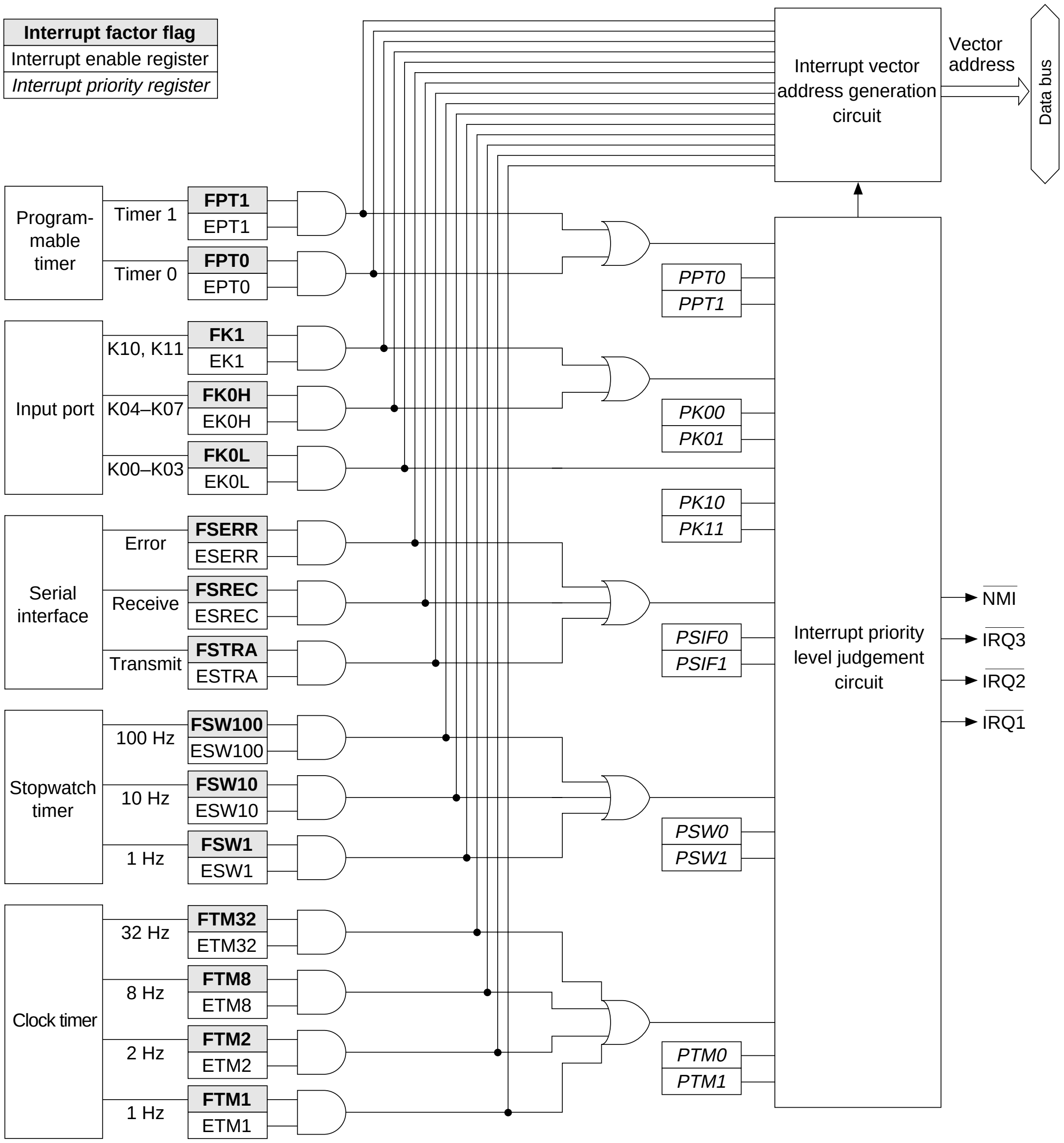


Fig. 5.15.1 Configuration of interrupt circuit

■ HALT status

By executing the program's HALT instruction, the E0C88112 shifts to the HALT status.
Since CPU operation stops in the HALT status, power consumption can be reduced with only peripheral circuit operation.
Cancellation of the HALT status is done by initial reset or an optional interrupt request, and the CPU restarts program execution from an exception processing routine.
See the "E0C88 Core CPU Manual" for the HALT status and reactivation sequence.

■ SLEEP status

By executing the program's SLP instruction, the E0C88112 shifts to the SLEEP status.
Since the operation of the CPU and peripheral circuits stop completely in the SLEEP status, power consumption can be reduced even more than in the HALT status.
Cancellation of the SLEEP status is done by initial reset or an input interrupt from the input port. The CPU reactivates after waiting 8,192/fOSC1 seconds of oscillation stabilization time. At this time, the CPU restarts program execution from an exception processing routine (input interrupt routine).

Note: Since oscillation is unstable for a short time after reactivation from the SLEEP status, the wait time is not always 250 msec even when using the 32.768 kHz crystal oscillator for the OSC1 oscillation circuit.

5.15.1 Interrupt generation conditions

The interrupt factor flags that indicate the generation of their respective interrupt factors are provided for the previously indicated 6 systems and 15 types of interrupts and they will be set to "1" by the generation of a factor.

In addition, interrupt enable registers with a 1 to 1 correspondence to each of the interrupt factor flags are provided. An interrupt is enabled when "1" is written and interrupt is disabled when "0" is written.

The CPU manages the enable/disable of interrupt requests at the interrupt priority level. An interrupt priority register that sets the priority level is provided for each of the interrupts of the 6 systems and the CPU accepts only interrupts above the level that has been indicated with the interrupt flags (I0 and I1).
Consequently, the following three conditions are necessary for the CPU to accept the interrupt.

(1) The interrupt factor flag has been set to "1" by generation of an interrupt factor.
(2) The interrupt enable register corresponding to the above has been set to "1".
(3) The interrupt priority register corresponding to the above has been set to a priority level higher than the interrupt flag (I0 and I1) setting.

The CPU initially samples the interrupt for the first op-code fetch cycle of each instruction. Thereupon, the CPU shifts to the exception processing when the above mentioned conditions have been established.
See the "E0C88 Core CPU Manual" for the exception processing sequence.

5.15.2 Interrupt factor flag

Table 5.15.2.1 shows the correspondence between the factors generating an interrupt and the interrupt factor flags.
The corresponding interrupt factor flags are set to "1" by generation of the respective interrupt factors.
The corresponding interrupt factor can be confirmed by reading the flags through software.

Table 5.15.2.1 Interrupt factors

Interrupt factor	Interrupt factor flag	
Programmable timer 1 underflow	FPT1	(00FF25 D7)
Programmable timer 0 underflow	FPT0	(00FF25 D6)
Non matching of the K10 and K11 inputs and the input comparison registers KCP10 and KCP11	FK1	(00FF25 D5)
Non matching of the K04–K07 inputs and the input comparison registers KCP04–KCP07	FK0H	(00FF25 D4)
Non matching of the K00–K03 inputs and the input comparison registers KCP00–KCP03	FK0L	(00FF25 D3)
Serial interface receiving error (in asynchronous mode)	FSERR	(00FF25 D2)
Serial interface receiving completion	FSREC	(00FF25 D1)
Serial interface transmitting completion	FSTRA	(00FF25 D0)
Falling edge of the stopwatch timer 100 Hz signal	FSW100	(00FF24 D6)
Falling edge of the stopwatch timer 10 Hz signal	FSW10	(00FF24 D5)
Falling edge of the stopwatch timer 1 Hz signal	FSW1	(00FF24 D4)
Rising edge of the clock timer 32 Hz signal	FTM32	(00FF24 D3)
Rising edge of the clock timer 8 Hz signal	FTM8	(00FF24 D2)
Rising edge of the clock timer 2 Hz signal	FTM2	(00FF24 D1)
Rising edge of the clock timer 1 Hz signal	FTM1	(00FF24 D0)

Interrupt factor flag that has been set to "1" is reset to "0" by writing "1".
At initial reset, the interrupt factor flags are reset to "0".

Note: When executing the RETE instruction without resetting the interrupt factor flag after an interrupt has been generated, the same interrupt will be generated. Consequently, the interrupt factor flag corresponding to that routine must be reset (writing "1") in the interrupt processing routine.

5.15.3 Interrupt enable register

The interrupt enable register has a 1 to 1 correspondence with each interrupt factor flag and enable/disable of interrupt requests can be set.

When "1" is written to the interrupt enable register, an interrupt request is enabled, and is disabled when "0" is written. This register also permits reading, thus making it possible to confirm that a status has been set.
At initial reset, the interrupt enable registers are set to "0" and shifts to the interrupt disable status.
Table 5.15.3.1 shows the correspondence between the interrupt enable registers and the interrupt factor flags.

5.15.4 Interrupt priority register and interrupt priority level

The interrupt priority registers shown in Table 5.15.4.1 are set to each system of interrupts and the interrupt priority levels for the CPU can be set to the optional priority level (0–3). As a result, it is possible to have multiple interrupts that match the system's interrupt processing priority levels.

The interrupt priority level between each system can optionally be set to three levels by the interrupt priority register. However, when more than one system is set to the same priority level, they are processed according to the default priority level.

Table 5.15.4.2 Setting of interrupt priority level

P*1	P*0	Interrupt priority level	
1	1	Level 3	($\overline{\text{IRQ3}}$)
1	0	Level 2	($\overline{\text{IRQ2}}$)
0	1	Level 1	($\overline{\text{IRQ1}}$)
0	0	Level 0	(non)

Table 5.15.3.1 Interrupt enable registers and interrupt factor flags

Interrupt	Interrupt factor flag		Interrupt enable register	
Programmable timer 1	FPT1	(00FF25 D7)	EPT1	(00FF23 D7)
Programmable timer 0	FPT0	(00FF25 D6)	EPT0	(00FF23 D6)
K10 and K11 input	FK1	(00FF25 D5)	EK1	(00FF23 D5)
K04–K07 input	FK0H	(00FF25 D4)	EK0H	(00FF23 D4)
K00–K03 input	FK0L	(00FF25 D3)	EK0L	(00FF23 D3)
Serial interface receiving error	FSERR	(00FF25 D2)	ESERR	(00FF23 D2)
Serial interface receiving completion	FSREC	(00FF25 D1)	ESREC	(00FF23 D1)
Serial interface transmitting completion	FSTRA	(00FF25 D0)	ESTRA	(00FF23 D0)
Stopwatch timer 100 Hz	FSW100	(00FF24 D6)	ESW100	(00FF22 D6)
Stopwatch timer 10 Hz	FSW10	(00FF24 D5)	ESW10	(00FF22 D5)
Stopwatch timer 1 Hz	FSW1	(00FF24 D4)	ESW1	(00FF22 D4)
Clock timer 32 Hz	FTM32	(00FF24 D3)	ETM32	(00FF22 D3)
Clock timer 8 Hz	FTM8	(00FF24 D2)	ETM8	(00FF22 D2)
Clock timer 2 Hz	FTM2	(00FF24 D1)	ETM2	(00FF22 D1)
Clock timer 1 Hz	FTM1	(00FF24 D0)	ETM1	(00FF22 D0)

Table 5.15.4.1 Interrupt priority register

Interrupt	Interrupt priority register	
Programmable timer interrupt	PPT0, PPT1	(00FF21 D2, D3)
K10 and K11 input interrupt	PK10, PK11	(00FF21 D0, D1)
K00–K07 input interrupt	PK00, PK01	(00FF20 D6, D7)
Serial interface interrupt	PSIF0, PSIF1	(00FF20 D4, D5)
Stopwatch timer interrupt	PSW0, PSW1	(00FF20 D2, D3)
Clock timer interrupt	PTM0, PTM1	(00FF20 D0, D1)

At initial reset, the interrupt priority registers are all set to "0" and each interrupt is set to level 0. Furthermore, the priority levels in each system have been previously decided and they cannot be changed.

The CPU can mask each interrupt by setting the interrupt flags (I0 and I1). The relation between the interrupt priority level of each system and interrupt flags is shown in Table 5.15.4.3, and the CPU accepts only interrupts above the level indicated by the interrupt flags.

The $\overline{\mathrm{NMI}}$ (watchdog timer) that has level 4 priority, is always accepted regardless of the setting of the interrupt flags.

Table 5.15.4.3 Interrupt mask setting of CPU

I1	I0	Acceptable interrupt
1	1	Level 4 ($\overline{\mathrm{NMI}}$)
1	0	Level 4, Level 3 ($\overline{\mathrm{IRQ3}}$)
0	1	Level 4, Level 3, Level 2 ($\overline{\mathrm{IRQ2}}$)
0	0	Level 4, Level 3, Level 2, Level 1 ($\overline{\mathrm{IRQ1}}$)

After an interrupt has been accepted, the interrupt flags are written to the level of that interrupt. However, interrupt flags after an $\overline{\mathrm{NMI}}$ has been accepted are written to level 3 (I0 = I1 = "1").

Table 5.15.4.4 Interrupt flags after acceptance of interrupt

Accepted interrupt priority level	I1	I0
Level 4 ($\overline{\mathrm{NMI}}$)	1	1
Level 3 ($\overline{\mathrm{IRQ3}}$)	1	1
Level 2 ($\overline{\mathrm{IRQ2}}$)	1	0
Level 1 ($\overline{\mathrm{IRQ1}}$)	0	1

The set interrupt flags are reset to their original value on return from the interrupt processing routine. Consequently, multiple interrupts up to 3 levels can be controlled by the initial settings of the interrupt priority registers alone. Additional multiplexing can be realized by rewriting the interrupt flags and interrupt enable register in the interrupt processing routine.

Note: Beware. If the interrupt flags have been rewritten (set to lower priority) prior to resetting an interrupt factor flag after an interrupt has been generated, the same interrupt will be generated again.

5.15.5 Exception processing vectors

When the CPU accepts an interrupt request, it starts exception processing following completion of the instruction being executed. In exception processing, the following operations branch the program.

(1) In the minimum mode, the program counter (PC) and system condition flag (SC) are moved to stack and in the maximum mode, the code bank register (CB), PC and SC are moved.

(2) The branch destination address is read from the exception processing vector corresponding to each exception processing (interrupt) factor and is placed in the PC.

An exception vector is 2 bytes of data in which the top address of each exception (interrupt) processing routine has been stored and the vector addresses correspond to the exception processing factors as shown in Table 5.15.5.1.

Table 5.15.5.1 Vector address and exception processing correspondence

Vector address	Exception processing factor	Priority
000000H	Reset	High
000002H	Zero division	↑
000004H	Watchdog timer ($\overline{\mathrm{NMI}}$)	
000006H	Programmable timer 1 interrupt	
000008H	Programmable timer 0 interrupt	
00000AH	K10, K11 input interrupt	
00000CH	K04–K07 input interrupt	
00000EH	K00–K03 input interrupt	
000010H	Serial I/F error interrupt	
000012H	Serial I/F receiving complete interrupt	
000014H	Serial I/F transmitting complete interrupt	
000016H	Stopwatch timer 100 Hz interrupt	
000018H	Stopwatch timer 10 Hz interrupt	
00001AH	Stopwatch timer 1 Hz interrupt	
00001CH	Clock timer 32 Hz interrupt	
00001EH	Clock timer 8 Hz interrupt	
000020H	Clock timer 2 Hz interrupt	↓
000022H	Clock timer 1 Hz interrupt	Low
000024H	System reserved (cannot be used)	No priority rating
000026H : 0000FEH	Software interrupt	

Note: An exception processing vector is fixed at 2 bytes, so it cannot specify a branch destination bank address. Consequently, to branch from multiple banks to a common exception processing routine, the top portion of an exception processing routine must be described within the common area (000000H–007FFFH).

5.15.6 Control of interrupt

Table 5.15.6.1 shows the interrupt control bits.

Table 5.15.6.1 Interrupt control bits

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF20	D7	PK01	K00–K07 interrupt priority register	PK01 PK00 PSIF1 PSIF0 PSW1 PSW0 Priority PTM1 PTM0 level 1 1 Level 3 1 0 Level 2 0 1 Level 1 0 0 Level 0		0	R/W	
	D6	PK00						
	D5	PSIF1	Serial interface interrupt priority register			0	R/W	
	D4	PSIF0						
	D3	PSW1	Stopwatch timer interrupt priority register			0	R/W	
	D2	PSW0						
	D1	PTM1	Clock timer interrupt priority register			0	R/W	
	D0	PTM0						
00FF21	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	PPT1	Programmable timer interrupt priority register	PPT1 PPT0 Priority PK11 PK10 level 1 1 Level 3 1 0 Level 2 0 1 Level 1 0 0 Level 0		0	R/W	
	D2	PPT0						
	D1	PK11	K10 and K11 interrupt priority register			0	R/W	
	D0	PK10						
00FF22	D7	–	–	–	–	–		"0" when being read
	D6	ESW100	Stopwatch timer 100 Hz interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D5	ESW10	Stopwatch timer 10 Hz interrupt enable register					
	D4	ESW1	Stopwatch timer 1 Hz interrupt enable register					
	D3	ETM32	Clock timer 32 Hz interrupt enable register					
	D2	ETM8	Clock timer 8 Hz interrupt enable register					
	D1	ETM2	Clock timer 2 Hz interrupt enable register					
	D0	ETM1	Clock timer 1 Hz interrupt enable register					
00FF23	D7	EPT1	Programmable timer 1 interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D6	EPT0	Programmable timer 0 interrupt enable register					
	D5	EK1	K10 and K11 interrupt enable register					
	D4	EK0H	K04–K07 interrupt enable register					
	D3	EK0L	K00–K03 interrupt enable register					
	D2	ESERR	Serial I/F (error) interrupt enable register					
	D1	ESREC	Serial I/F (receiving) interrupt enable register					
	D0	ESTRA	Serial I/F (transmitting) interrupt enable register					
00FF24	D7	–	–	–	–	–		"0" when being read
	D6	FSW100	Stopwatch timer 100 Hz interrupt factor flag	(R) Interrupt factor is generated (W) Reset	(R) No interrupt factor is generated (W) No operation	0	R/W	
	D5	FSW10	Stopwatch timer 10 Hz interrupt factor flag					
	D4	FSW1	Stopwatch timer 1 Hz interrupt factor flag					
	D3	FTM32	Clock timer 32 Hz interrupt factor flag					
	D2	FTM8	Clock timer 8 Hz interrupt factor flag					
	D1	FTM2	Clock timer 2 Hz interrupt factor flag					
	D0	FTM1	Clock timer 1 Hz interrupt factor flag					
00FF25	D7	FPT1	Programmable timer 1 interrupt factor flag	(R) Interrupt factor is generated (W) Reset	(R) No interrupt factor is generated (W) No operation	0	R/W	
	D6	FPT0	Programmable timer 0 interrupt factor flag					
	D5	FK1	K10 and K11 interrupt factor flag					
	D4	FK0H	K04–K07 interrupt factor flag					
	D3	FK0L	K00–K03 interrupt factor flag					
	D2	FSERR	Serial I/F (error) interrupt factor flag					
	D1	FSREC	Serial I/F (receiving) interrupt factor flag					
	D0	FSTRA	Serial I/F (transmitting) interrupt factor flag					

Refer to the explanations on the respective peripheral circuits for the setting content and control method for each bit.

5.15.7 Programming notes

(1) When executing the RETE instruction without resetting the interrupt factor flag after an interrupt has been generated, the same interrupt will be generated. Consequently, the interrupt factor flag corresponding to that routine must be reset (writing "1") in the interrupt processing routine.

(2) Beware. If the interrupt flags (I0 and I1) have been rewritten (set to lower priority) prior to resetting an interrupt factor flag after an interrupt has been generated, the same interrupt will be generated again.

(3) An exception processing vector is fixed at 2 bytes, so it cannot specify a branch destination bank address. Consequently, to branch from multiple banks to a common exception processing routine, the front portion of an exception processing routine must be described within the common area (000000H–007FFFH).

5.16 Notes for Low Current Consumption

The E0C88112 can turn circuits, which consume a large amount of power, ON or OFF by control registers.
You can reduce power consumption by creating a program that operates the minimum necessary circuits using these control registers.

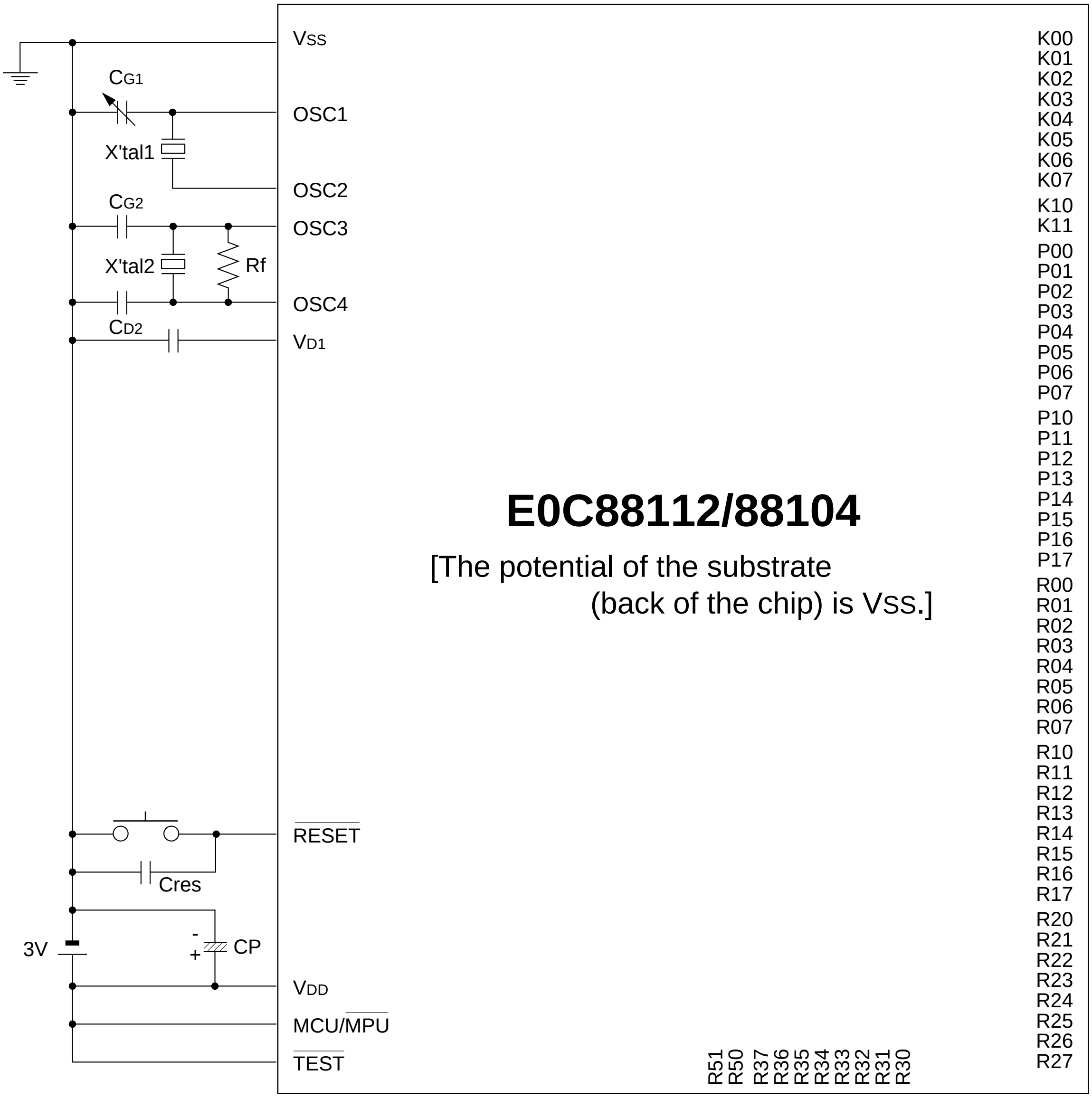
Next, which circuit systems' operation can be controlled and their control registers (instructions) are explained. You should refer to these when programming.

See Chapter 7, "ELECTRICAL CHARACTERISTICS" for the current consumption.

Table 5.17.1 Circuit systems and control registers

Circuit type	Control register (Instruction)	Status at time of initial resetting
CPU	HALT and SLP instructions	Operation status
Oscillation circuit	CLKCHG, OSCC	OSC1 clock (CLKCHG = "0") OSC3 oscillation OFF (OSCC = "0")
Operating mode	VDC0, VDC1	Normal mode (VDC0 = VDC1 = "0")
SVD circuit	SVDON, SVDSP	OFF status (SVDON = SVDSP = "0")
Analog comparator	CMP0ON, CMP1ON	OFF status (CMP0ON = CMP1ON = "0")

6 BASIC EXTERNAL WIRING DIAGRAM



Recommended values for external parts

Symbol	Name	Recommended value
X'tal1	Crystal oscillator	32.768 kHz
X'tal2	Crystal oscillator	4 MHz
Rf	Feedback resistor	1 MΩ
CG1	Trimmer capacitor	5–25 pF
CG2	Gate capacitor	15 pF
CD2	Drain capacitor	15 pF
CP	Capacitor for power supply	3.3 µF
Cres	Capacitor for RESET terminal	0.47 µF

7 ELECTRICAL CHARACTERISTICS

7.1 Absolute Maximum Rating

(V_{SS} = 0 V)

Item	Symbol	Condition	Rated value	Unit	Note
Power voltage	V_{DD}		-0.3 to +7.0	V	
Input voltage	V_I		-0.3 to V_{DD} + 0.3	V	
Output voltage	V_O		-0.3 to V_{DD} + 0.3	V	1
High level output current	I_{OH}	1 terminal	-5	mA	
		Total of all terminals	-20	mA	
Low level output current	I_{OL}	1 terminal	5	mA	
		Total of all terminals	20	mA	
Permitted loss	P_D		200	mW	2
Operating temperature	Topr		-40 to +85	°C	
Storage temperature	Tstg		-65 to +150	°C	

Note) 1 Case that to Nch open drain output by the mask option is included.

2 In case of plastic package.

7.2 Recommended Operating Conditions

(V_{SS} = 0 V, Ta = -40 to 85°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Note
Operating power voltage (Normal mode)	V_{DD}		2.4		5.5	V	
Operating power voltage (Low power mode)	V_{DD}		1.8		3.5	V	
Operating power voltage (High speed mode)	V_{DD}		3.5		5.5	V	
Operating frequency (Normal mode)	f_{OSC1}	V_{DD} = 2.4 to 5.5 V	30.000	32.768	50.000	kHz	3
	f_{OSC3}		0.03		4.2	MHz	3
Operating frequency (Low power mode)	f_{OSC1}	V_{DD} = 1.8 to 3.5 V	30.000	32.768	50.000	kHz	3
Operating frequency (High speed mode)	f_{OSC1}	V_{DD} = 3.5 to 5.5 V	30.000	32.768	50.000	kHz	3
	f_{OSC3}		0.03		8.2	MHz	3
Capacitor between V_{D1} and V_{SS}	C1			0.1		µF	

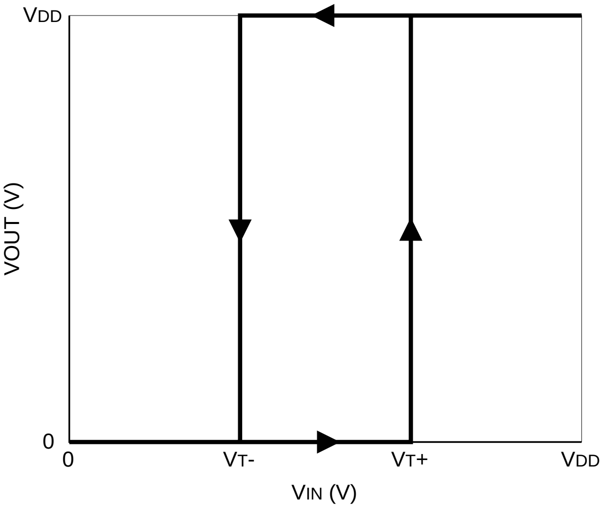
Note) 3 When an external clock is input from the OSC1 terminal by the mask option, do not connect anything to the OSC2 terminal, and when an external clock is input from the OSC3 terminal, do not connect to the OSC4 terminal.

7.3 DC Characteristics

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Note
High level input voltage (1)	V_{IH1}	Kxx, Pxx, MCU/$\overline{\text{MPU}}$	0.8V_{DD}		V_{DD}	V	
Low level input voltage (1)	V_{IL1}	Kxx, Pxx, MCU/$\overline{\text{MPU}}$	0		0.2V_{DD}	V	
High level input voltage (2) (Normal mode)	V_{IH2}	OSC1, OSC3	1.6		V_{DD}	V	4
High level input voltage (2) (Low power mode)	V_{IH2}	OSC1	1.0		V_{DD}	V	4
High level input voltage (2) (High speed mode)	V_{IH2}	OSC1, OSC3	2.4		V_{DD}	V	4
Low level input voltage (2) (Normal mode)	V_{IL2}	OSC1, OSC3	0		0.6	V	4
Low level input voltage (2) (Low power mode)	V_{IL2}	OSC1	0		0.3	V	4
Low level input voltage (2) (High speed mode)	V_{IL2}	OSC1, OSC3	0		0.9	V	4
High level schmitt input voltage	V_{T+}	$\overline{\text{RESET}}$	0.5V_{DD}		0.9V_{DD}	V	
Low level schmitt input voltage	V_{T-}	$\overline{\text{RESET}}$	0.1V_{DD}		0.5V_{DD}	V	
High level output current	I_{OH}	Pxx, Rxx, V_{OH} = 0.9 V_{DD}			-0.5	mA	
Low level output current	I_{OL}	Pxx, Rxx, V_{OL} = 0.1 V_{DD}	0.5			mA	
Input leak current	I_{LI}	Kxx, Pxx, $\overline{\text{RESET}}$, MCU/$\overline{\text{MPU}}$	-1		1	µA	
Output leak current	I_{LO}	Pxx, Rxx	-1		1	µA	
Input pull-up resistance	R_{IN}	Kxx, Pxx, $\overline{\text{RESET}}$, MCU/$\overline{\text{MPU}}$	100		500	kΩ	5
Input terminal capacitance	C_{IN}	Kxx, Pxx V_{IN} = 0 V, f = 1 MHz, Ta = 25°C			15	pF	

Note) 4 When external clock is selected by mask option.
5 When addition of pull-up resistor is selected by mask option.



7.4 Analog Circuit Characteristics

■ SVD circuit

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Note
SVD voltage	V_{SVD}	Level 1 → Level 0	Typ×0.92	1.82	Typ×1.08	V	
		Level 2 → Level 1		2.00		V	
		Level 3 → Level 2		2.18		V	
		Level 4 → Level 3		2.36		V	
		Level 5 → Level 4		2.54		V	
		Level 6 → Level 5		2.72		V	
		Level 7 → Level 6		2.90		V	
		Level 8 → Level 7		3.08		V	
		Level 9 → Level 8		3.26		V	
		Level 10 → Level 9	Typ×0.88	3.45	Typ×1.12	V	
		Level 11 → Level 10		3.65		V	
		Level 12 → Level 11		3.85		V	
		Level 13 → Level 12		4.05		V	
		Level 14 → Level 13		4.25		V	
		Level 15 → Level 14		4.50		V	

■ Analog comparator circuit

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Note
Analog comparator	V_{CMIP}	Non-inverted input (CMPP)	0.7		V_{DD} - 0.7	V	6
operating voltage input range	V_{CMIM}	Inverted input (CMPM)	0.7		V_{DD} - 0.7	V	6
Analog comparator offset voltage	V_{CMOF}	V_{CMIP} = 0.7 V to V_{DD} - 0.7 V V_{CMIM} = 0.7 V to V_{DD} - 0.7 V			20	mV	6
Analog comparator stability time	t_{CMP1}				1	mS	7
Analog comparator response time	t_{CMP2}	V_{CMIP} = 0.7 V to V_{DD} - 0.7 V V_{CMIM} = 0.7 V to V_{DD} - 0.7 V V_{CMIP} = V_{CMIM} ± 0.025 V			2	mS	6 8

Note) 6 When "without pull-up resistor" (comparator input terminal) is selected by mask option.

7 Stability time is the time from turning the circuit ON until the circuit is stabilized.

8 Response time is the time that the output result responds to the input signal.

7.5 Power Current Consumption

Unless otherwise specified: V_{DD} = Within the operating voltage in each operating mode, V_{SS} = 0 V, Ta = 25°C, OSC1 = 32.768 kHz crystal oscillation, C_G = 10pF, OSC3 = External clock input, Non heavy load protection mode, C1 = 0.1 µF

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Note
Power current (Normal mode)	I_{DD1}	In SLEEP status *1		0.3	1	µA	
	I_{DD2}	In HALT status *2		2	5	µA	
	I_{DD3}	CPU is in operating (32.768 kHz) *3		14	18	µA	
	I_{DD4}	CPU is in operating (1 MHz) *4		0.45	0.60	mA	
	I_{HVL}	In heavy load protection mode		25	50	µA	9
Power current (Low power mode)	I_{DD1}	In SLEEP status *1		0.2	1	µA	
	I_{DD2}	In HALT status *2		1	5	µA	
	I_{DD3}	CPU is in operating (32.768 kHz) *3		8	12	µA	
	I_{HVL}	In heavy load protection mode		15	30	µA	9
Power current (High speed mode)	I_{DD1}	In SLEEP status *1		1	3	µA	
	I_{DD2}	In HALT status *2		5	10	µA	
	I_{DD3}	CPU is in operating (32.768 kHz) *3		24	30	µA	
	I_{DD4}	CPU is in operating (1 MHz) *4		0.70	1.00	mA	
	I_{HVL}	In heavy load protection mode		35	70	µA	9
SVD circuit current	I_{SVDN}	V_{DD} = 3.0 V		30	60	µA	10
	I_{SVDH}	In heavy load protection mode		25	75	µA	9
Analog comparator circuit current	I_{CMP1}	CMPXDT = "1"		40	100	µA	
	I_{CMP2}	CMPXDT = "0"		4	10	µA	
OSC1 CR oscillation current	I_{CR1}			20	50	µA	11

*1 OSC1: Stop, OSC3: Stop, CPU, ROM, RAM: SLEEP status, Clock timer: Stop, Others: Stop status

*2 OSC1: Oscillating, OSC3: Stop, CPU, ROM, RAM: HALT status, Clock timer: Operating, Others: Stop status

*3 OSC1: Oscillating, OSC3: Stop, CPU, ROM, RAM: Operating in 32.768 kHz, Clock timer: Operating, Others: Stop status

*4 OSC1: Oscillating, OSC3: Oscillating, CPU, ROM, RAM: Operating in 1 MHz, Clock timer: Operating, Others: Stop status

Note) 9 It is the value of current which flows in the heavy load protection circuit when in the heavy load protection mode (OSC3 ON or buzzer ON).

10 The value in ***x*** V can be found by the following expression: I_{SVDN} (V_{DD} = ***x*** V) = (***x*** × 20) - 30 (Typ. value), I_{SVDN} (V_{DD} = ***x*** V) = (***x*** × 30) - 30 (Max. value)

11 When OSC1 CR oscillation circuit is selected by the mask option.

7.6 AC Characteristics

■ External memory access

• Read cycle (Normal operating mode)

Condition: VDD = 2.4 to 5.5 V, VSS = 0 V, Ta = -40 to 85°C, VIH1 = 0.8VDD, VIL1 = 0.2VDD, VIH2 = 1.6 V, VIL2 = 0.6 V, VOH = 0.8VDD, VOL = 0.2VDD, CL = 100 pF (load capacitance)

Item	Symbol	Min.	Typ.	Max.	Unit	Note
Address set-up time in read cycle	tras	tc+tl-100+n•tc/2			nS	12
Address hold time in read cycle	trah	th-80			nS	
Read signal pulse width	trp	tc-20+n•tc/2			nS	12
Data input set-up time in read cycle	trds	300			nS	
Data input hold time in read cycle	trdh	0			nS	

Note) 12 Substitute the number of states for wait insertion in n.

• Read cycle (High speed operating mode)

Condition: VDD = 3.5 to 5.5 V, VSS = 0 V, Ta = -40 to 85°C, VIH1 = 0.8VDD, VIL1 = 0.2VDD, VIH2 = 2.4 V, VIL2 = 0.9 V, VOH = 0.8VDD, VOL = 0.2VDD, CL = 100 pF (load capacitance)

Item	Symbol	Min.	Typ.	Max.	Unit	Note
Address set-up time in read cycle	tras	tc+tl-50+n•tc/2			nS	12
Address hold time in read cycle	trah	th-40			nS	
Read signal pulse width	trp	tc-10+n•tc/2			nS	12
Data input set-up time in read cycle	trds	150			nS	
Data input hold time in read cycle	trdh	0			nS	

Note) 12 Substitute the number of states for wait insertion in n.

• Read cycle (Low power operating mode)

Condition: VDD = 1.8 to 3.5 V, VSS = 0 V, Ta = -40 to 85°C, VIH1 = 0.8VDD, VIL1 = 0.2VDD, VIH2 = 1.0 V, VIL2 = 0.3 V, VOH = 0.8VDD, VOL = 0.2VDD, CL = 100 pF (load capacitance)

Item	Symbol	Min.	Typ.	Max.	Unit	Note
Address set-up time in read cycle	tras	15			µS	
Address hold time in read cycle	trah	5			µS	
Read signal pulse width	trp	10			µS	
Data input set-up time in read cycle	trds	10			µS	
Data input hold time in read cycle	trdh	0			µS	

• Write cycle (Normal operating mode)

Condition: VDD = 2.4 to 5.5 V, VSS = 0 V, Ta = -40 to 85°C, VIH1 = 0.8VDD, VIL1 = 0.2VDD, VIH2 = 1.6 V, VIL2 = 0.6 V, VOH = 0.8VDD, VOL = 0.2VDD, CL = 100 pF (load capacitance)

Item	Symbol	Min.	Typ.	Max.	Unit	Note
Address set-up time in write cycle	twas	tc-180			nS	
Address hold time in write cycle	twah	th-80			nS	
Write signal pulse width	twp	tl-40+n•tc/2			nS	13
Data output set-up time in write cycle	twds	tc-180+n•tc/2			nS	13
Data output hold time in write cycle	twdh	th-80		th+80	nS	

Note) 13 Substitute the number of states for wait insertion in n.

• Write cycle (High speed operating mode)

Condition: V_{DD} = 3.5 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 85°C, V_{IH1} = 0.8V_{DD}, V_{IL1} = 0.2V_{DD}, V_{IH2} = 2.4 V, V_{IL2} = 0.9 V, V_{OH} = 0.8V_{DD}, V_{OL} = 0.2V_{DD}, CL = 100 pF (load capacitance)

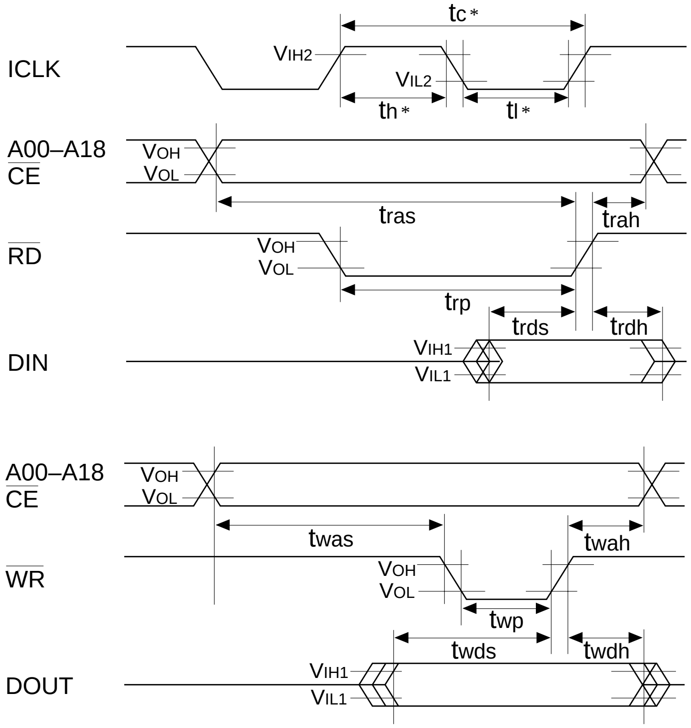
Item	Symbol	Min.	Typ.	Max.	Unit	Note
Address set-up time in write cycle	twas	tc-90			nS	
Address hold time in write cycle	twah	th-40			nS	
Write signal pulse width	twp	tl-20+n•tc/2			nS	13
Data output set-up time in write cycle	twds	tc-90+n•tc/2			nS	13
Data output hold time in write cycle	twdh	th-40		th+40	nS	

Note) 13 Substitute the number of states for wait insertion in n.

• Write cycle (Low power operating mode)

Condition: V_{DD} = 1.8 to 3.5 V, V_{SS} = 0 V, Ta = -40 to 85°C, V_{IH1} = 0.8V_{DD}, V_{IL1} = 0.2V_{DD}, V_{IH2} = 1.0 V, V_{IL2} = 0.3 V, V_{OH} = 0.8V_{DD}, V_{OL} = 0.2V_{DD}, CL = 100 pF (load capacitance)

Item	Symbol	Min.	Typ.	Max.	Unit	Note
Address set-up time in write cycle	twas	10			µS	
Address hold time in write cycle	twah	5			µS	
Write signal pulse width	twp	5			µS	
Data output set-up time in write cycle	twds	10			µS	
Data output hold time in write cycle	twdh	5		20	µS	



* In the case of crystal oscillation and ceramic oscillation: th = 0.5tc ± 0.05tc, tl = tc - th (1/tc: oscillation frequency)

* In the case of CR oscillation: th = 0.5tc ± 0.10tc, tl = tc - th (1/tc: oscillation frequency)

■ Serial interface

• Clock synchronous master mode (Normal operating mode)

Condition: V_{DD} = 2.4 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 85°C, V_{IH1} = 0.8V_{DD}, V_{IL1} = 0.2V_{DD}, V_{OH} = 0.8V_{DD}, V_{OL} = 0.2V_{DD}

Item	Symbol	Min.	Typ.	Max.	Unit	Note
Transmitting data output delay time	t_{smd}			200	nS	
Receiving data input set-up time	t_{sms}	500			nS	
Receiving data input hold time	t_{smh}	200			nS	

• Clock synchronous master mode (High speed operating mode)

Condition: V_{DD} = 3.5 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 85°C, V_{IH1} = 0.8V_{DD}, V_{IL1} = 0.2V_{DD}, V_{OH} = 0.8V_{DD}, V_{OL} = 0.2V_{DD}

Item	Symbol	Min.	Typ.	Max.	Unit	Note
Transmitting data output delay time	t_{smd}			100	nS	
Receiving data input set-up time	t_{sms}	250			nS	
Receiving data input hold time	t_{smh}	100			nS	

• Clock synchronous master mode (Low power operating mode)

Condition: V_{DD} = 1.8 to 3.5 V, V_{SS} = 0 V, Ta = -40 to 85°C, V_{IH1} = 0.8V_{DD}, V_{IL1} = 0.2V_{DD}, V_{OH} = 0.8V_{DD}, V_{OL} = 0.2V_{DD}

Item	Symbol	Min.	Typ.	Max.	Unit	Note
Transmitting data output delay time	t_{smd}			5	µS	
Receiving data input set-up time	t_{sms}	10			µS	
Receiving data input hold time	t_{smh}	5			µS	

• Clock synchronous slave mode (Normal operating mode)

Condition: V_{DD} = 2.4 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 85°C, V_{IH1} = 0.8V_{DD}, V_{IL1} = 0.2V_{DD}, V_{OH} = 0.8V_{DD}, V_{OL} = 0.2V_{DD}

Item	Symbol	Min.	Typ.	Max.	Unit	Note
Transmitting data output delay time	t_{ssd}			500	nS	
Receiving data input set-up time	t_{sss}	200			nS	
Receiving data input hold time	t_{ssh}	200			nS	

• Clock synchronous slave mode (High speed operating mode)

Condition: V_{DD} = 3.5 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 85°C, V_{IH1} = 0.8V_{DD}, V_{IL1} = 0.2V_{DD}, V_{OH} = 0.8V_{DD}, V_{OL} = 0.2V_{DD}

Item	Symbol	Min.	Typ.	Max.	Unit	Note
Transmitting data output delay time	t_{ssd}			250	nS	
Receiving data input set-up time	t_{sss}	100			nS	
Receiving data input hold time	t_{ssh}	100			nS	

• Clock synchronous slave mode (Low power operating mode)

Condition: V_{DD} = 1.8 to 3.5 V, V_{SS} = 0 V, Ta = -40 to 85°C, V_{IH1} = 0.8V_{DD}, V_{IL1} = 0.2V_{DD}, V_{OH} = 0.8V_{DD}, V_{OL} = 0.2V_{DD}

Item	Symbol	Min.	Typ.	Max.	Unit	Note
Transmitting data output delay time	t_{ssd}			10	µS	
Receiving data input set-up time	t_{sss}	5			µS	
Receiving data input hold time	t_{ssh}	5			µS	

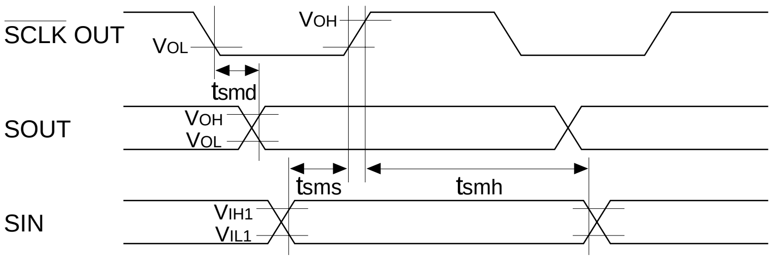
• Asynchronous system (All operating mode)

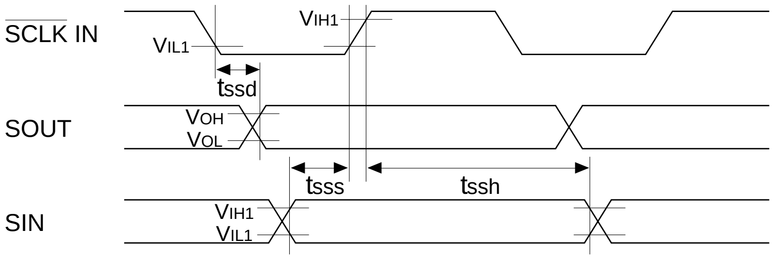
Condition: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 85°C

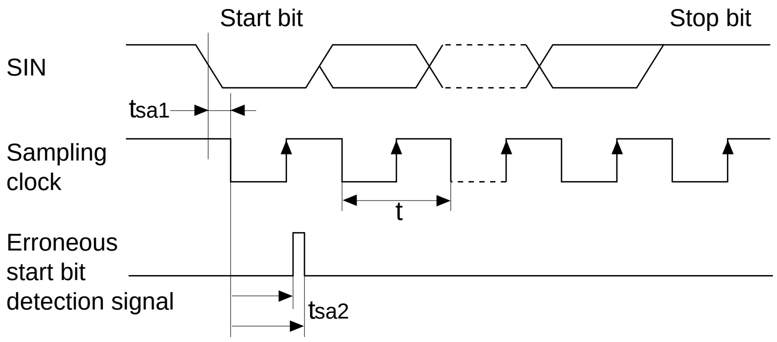
Item	Symbol	Min.	Typ.	Max.	Unit	Note
Start bit detection error time	t_{sa1}	0		t/16	S	14
Erroneous start bit detection range time	t_{sa2}	9t/16		10t/16	S	15

Note) 14 Start bit detection error time is a logical delay time from inputting the start bit until internal sampling begins operating. (Time as far as AC is excluded.)

15 Erroneous start bit detection range time is a logical range to detect whether a LOW level (start bit) has been input again after a start bit has been detected and the internal sampling clock has started.
When a HIGH level is detected, the start bit detection circuit is reset and goes into a wait status until the next start bit. (Time as far as AC is excluded.)







■ Input clock

• OSC1, OSC3 external clock (Normal operating mode)

Condition: V_{DD} = 2.4 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 85°C, V_{IH2} = 1.6 V, V_{IL2} = 0.6 V

Item		Symbol	Min.	Typ.	Max.	Unit	Note
OSC1 input clock time	Cycle time	t_{o1cy}	20		32	µS	
	"H" pulse width	t_{o1h}	10		16	µS	
	"L" pulse width	t_{o1l}	10		16	µS	
OSC3 input clock time	Cycle time	t_{o3cy}	250		32,000	nS	
	"H" pulse width	t_{o3h}	125		16,000	nS	
	"L" pulse width	t_{o3l}	125		16,000	nS	
Input clock rising time		t_{osr}			25	nS	
Input clock falling time		t_{osf}			25	nS	

• OSC1, OSC3 external clock (High speed operating mode)

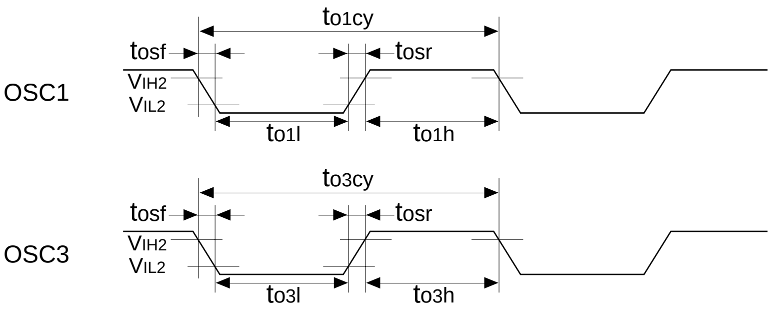
Condition: V_{DD} = 3.5 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 85°C, V_{IH2} = 2.4 V, V_{IL2} = 0.9 V

Item		Symbol	Min.	Typ.	Max.	Unit	Note
OSC1 input clock time	Cycle time	t_{o1cy}	20		32	µS	
	"H" pulse width	t_{o1h}	10		16	µS	
	"L" pulse width	t_{o1l}	10		16	µS	
OSC3 input clock time	Cycle time	t_{o3cy}	125		32,000	nS	
	"H" pulse width	t_{o3h}	62.5		16,000	nS	
	"L" pulse width	t_{o3l}	62.5		16,000	nS	
Input clock rising time		t_{osr}			25	nS	
Input clock falling time		t_{osf}			25	nS	

• OSC1, OSC3 external clock (Low power operating mode)

Condition: V_{DD} = 1.8 to 3.5 V, V_{SS} = 0 V, Ta = -40 to 85°C, V_{IH2} = 1.0 V, V_{IL2} = 0.3 V

Item		Symbol	Min.	Typ.	Max.	Unit	Note
OSC1 input clock time	Cycle time	t_{o1cy}	20		32	µS	
	"H" pulse width	t_{o1h}	10		16	µS	
	"L" pulse width	t_{o1l}	10		16	µS	
Input clock rising time		t_{osr}			25	nS	
Input clock falling time		t_{osf}			25	nS	



• $\overline{\text{SCLK}}$, EVIN input clock (Normal operating mode)

Condition: V_{DD} = 2.4 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 85°C, V_{IH1} = 0.8V_{DD}, V_{IL1} = 0.2V_{DD}

Item		System	Min.	Typ.	Max.	Unit	Note
$\overline{\text{SCLK}}$ input clock time	Cycle time	tsccy	4			µS	
	"H" pulse width	tsch	2			µS	
	"L" pulse width	tscl	2			µS	
EVIN input clock time (With noise rejector)	Cycle time	tevcy	64 / f_{OSC1}			S	
	"H" pulse width	tevh	32 / f_{OSC1}			S	
	"L" pulse width	tevl	32 / f_{OSC1}			S	
EVIN input clock time (Without noise rejector)	Cycle time	tevcy	4			µS	
	"H" pulse width	tevh	2			µS	
	"L" pulse width	tevl	2			µS	
Input clock rising time		tckr			25	nS	
Input clock falling time		tckf			25	nS	

• $\overline{\text{SCLK}}$, EVIN input clock (High speed operating mode)

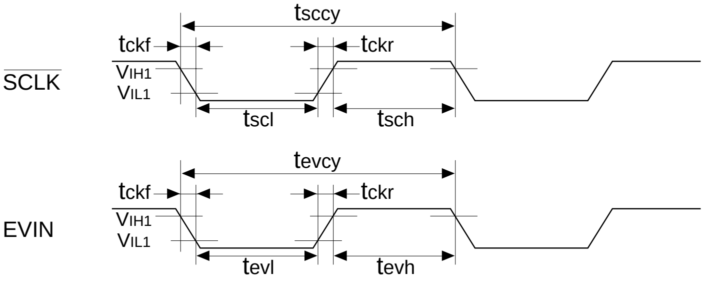
Condition: V_{DD} = 3.5 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 85°C, V_{IH1} = 0.8V_{DD}, V_{IL1} = 0.2V_{DD}

Item		System	Min.	Typ.	Max.	Unit	Note
$\overline{\text{SCLK}}$ input clock time	Cycle time	tsccy	2			µS	
	"H" pulse width	tsch	1			µS	
	"L" pulse width	tscl	1			µS	
EVIN input clock time (With noise rejector)	Cycle time	tevcy	64 / f_{OSC1}			S	
	"H" pulse width	tevh	32 / f_{OSC1}			S	
	"L" pulse width	tevl	32 / f_{OSC1}			S	
EVIN input clock time (Without noise rejector)	Cycle time	tevcy	2			µS	
	"H" pulse width	tevh	1			µS	
	"L" pulse width	tevl	1			µS	
Input clock rising time		tckr			25	nS	
Input clock falling time		tckf			25	nS	

• $\overline{\text{SCLK}}$, EVIN input clock (Low power operating mode)

Condition: V_{DD} = 1.8 to 3.5 V, V_{SS} = 0 V, Ta = -40 to 85°C, V_{IH1} = 0.8V_{DD}, V_{IL1} = 0.2V_{DD}

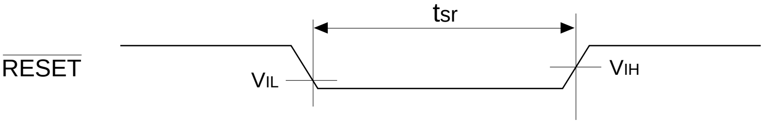
Item		System	Min.	Typ.	Max.	Unit	Note
$\overline{\text{SCLK}}$ input clock time	Cycle time	tsccy	100			µS	
	"H" pulse width	tsch	50			µS	
	"L" pulse width	tscl	50			µS	
EVIN input clock time (With noise rejector)	Cycle time	tevcy	64 / f_{OSC1}			S	
	"H" pulse width	tevh	32 / f_{OSC1}			S	
	"L" pulse width	tevl	32 / f_{OSC1}			S	
EVIN input clock time (Without noise rejector)	Cycle time	tevcy	100			µS	
	"H" pulse width	tevh	50			µS	
	"L" pulse width	tevl	50			µS	
Input clock rising time		tckr			25	nS	
Input clock falling time		tckf			25	nS	



- **$\overline{\text{RESET}}$ input clock (All operating mode)**

Condition: VDD = 1.8 to 5.5 V, VSS = 0 V, Ta = -40 to 85°C, VIH = 0.5VDD, VIL = 0.1VDD

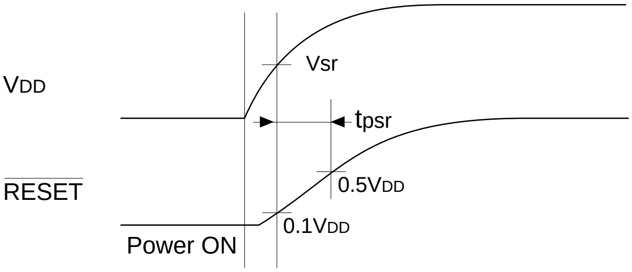
Item	Symbol	Min.	Typ.	Max.	Unit	Note
$\overline{\text{RESET}}$ input time	tsr	100			µS	

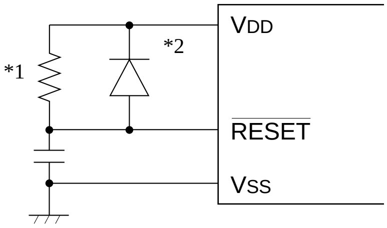


■ Power ON reset

Condition: VSS = 0 V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Note
Operating power voltage	Vsr	2.4			V	
$\overline{\text{RESET}}$ input time	tpsr	10			mS	





*1 When the built-in pull up resistor is not used.

*2 Because the potential of the $\overline{\text{RESET}}$ terminal not reached VDD level or higher.

■ Operating mode switching

Condition: VDD = 1.8 to 5.5 V, VSS = 0 V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Note
Stabilization time	tvdc	5			mS	16

Note) 16 Stabilization time is the time from switching on the operating mode until operating mode is stabilized. For example, when turning the OSC3 oscillation circuit on, stabilization time is needed after the operating mode is switched on.

7.7 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 oscillator is used for OSC3, use the oscillator manufacturer's recommended values for constants such as capacitance and resistance. The oscillation start time is important because it becomes the wait time when OSC3 clock is used. (If OSC3 is used as CPU clock before oscillation stabilizes, the CPU may malfunction.)

■ OSC1 (Crystal)

Unless otherwise specified: V_{DD} = Within the operating voltage in each operating mode, V_{SS} = 0 V, Ta = 25°C, Crystal oscillator = C2-TYPE*, C_{G1} = 25 pF, C_{D1} = Built-in

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Note
Oscillation start time	tsta				3	S	
External gate capacitance	C_{G1}	Including board capacitance	5	10	25	pF	17
Built-in gate capacitance	C_{G1}	In case of the chip		15		pF	18
Built-in drain capacitance	C_{D1}	In case of the chip		15		pF	
Frequency/IC deviation	∂f/∂IC	V_{DD} = constant	-10		10	ppm	
Frequency/power voltage deviation	∂f/∂V				1	ppm/V	
Frequency adjustment range	∂f/∂C_G	V_{DD} = constant, C_G = 5 to 25pF	25			ppm	
Frequency/operating mode devistion	∂f/∂MD	V_{DD} = constant			20	ppm	

* C2-TYPE Made by Seiko Epson corporation

Note) 17 When gate capacitance is not built in.

18 When gate capacitance is built in.

■ OSC1 (CR)

Unless otherwise specified: V_{DD} = 2.4 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Note
Oscillation start time	tsta				3	mS	
Frequency/IC deviation	∂f/∂IC	R_{CR1} = constant	-25		25	%	

■ OSC3 (Crystal)

Unless otherwise specified: V_{DD} = Within the operating voltage in each operating mode, V_{SS} = 0 V, Ta = 25°C, Crystal oscillator = CA-301 4MHz / CA-301 8MHz*, R_F = 1MΩ, C_{G2} = C_{D2} = 15pF

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Note
Oscillation start time (Normal mode)	tsta	4.0 MHz crystal oscillator			20	mS	19
Oscillation start time (High speed mode)	tsta	8.0 MHz crystal oscillator			20	mS	19

* CA-301 4MHz / CA-301 8MHz Made by Seiko Epson corporation

Note) 19 The crystal oscillation start time changes by the crystal oscillator to be used, C_{G2} and C_{D2}.

■ OSC3 (Ceramic)

Unless otherwise specified: V_{DD} = Within the operating voltage in each operating mode, V_{SS} = 0 V, Ta = 25°C, Ceramic oscillator = CSA4.00MG / CSA8.00MTZ*, R_F = 1MΩ, C_{G2} = C_{D2} = 30pF

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Note
Oscillation start time (Normal mode)	tsta	4.0 MHz ceramic oscillator			5	mS	
Oscillation start time (High speed mode)	tsta	8.0 MHz ceramic oscillator			5	mS	

* CSA4.00MG / CSA8.00MTZ Made by Murata Mfg. corporation

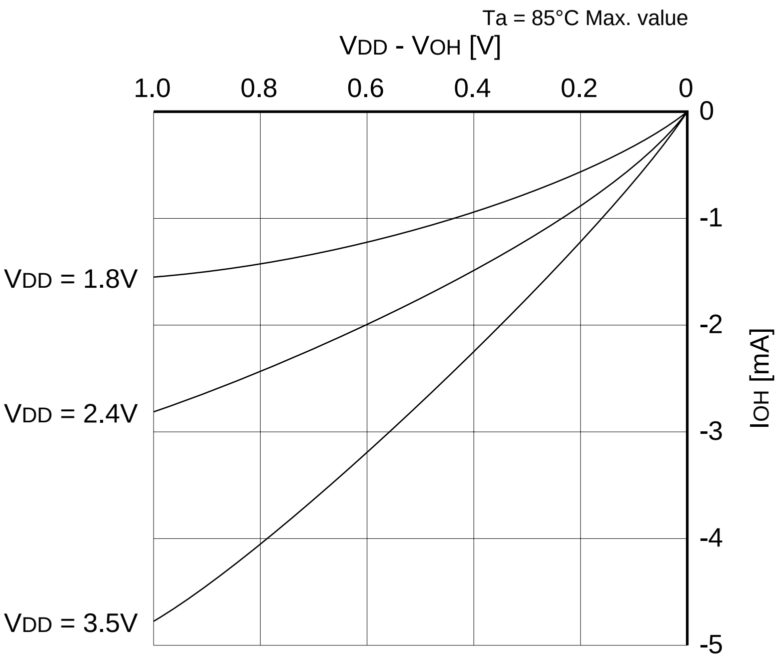
■ OSC3 (CR)

Unless otherwise specified: V_{DD} = Within the operating voltage in each operating mode, V_{SS} = 0 V, Ta = -40 to 85°C

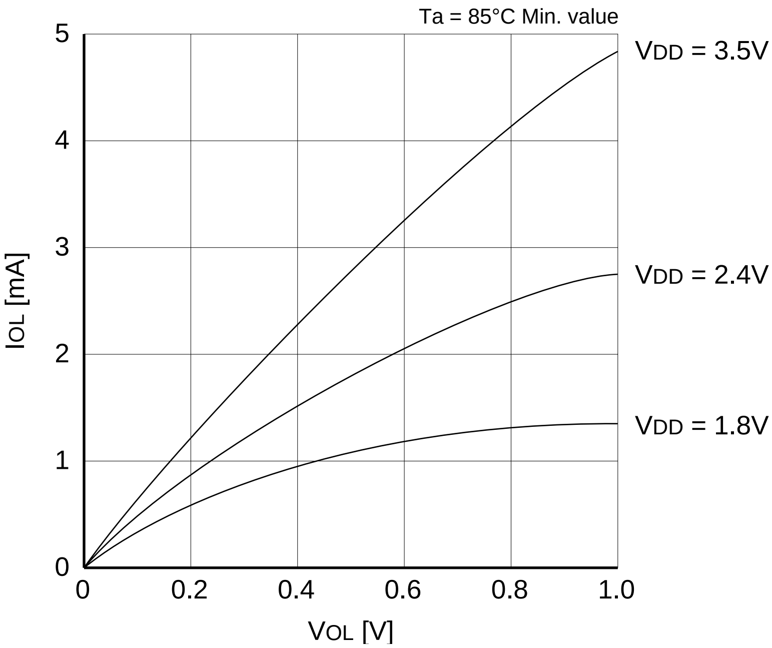
Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Note
Oscillation start time (Normal mode)	tsta				1	mS	
Oscillation start time (High speed mode)	tsta				1	mS	
Frequency/IC deviation (Normal mode)	∂f/∂IC	R_{CR3} = constant	-25		25	%	
Frequency/IC deviation (High speed mode)	∂f/∂IC	R_{CR3} = constant	-25		25	%	

7.8 Characteristics Curves (reference value)

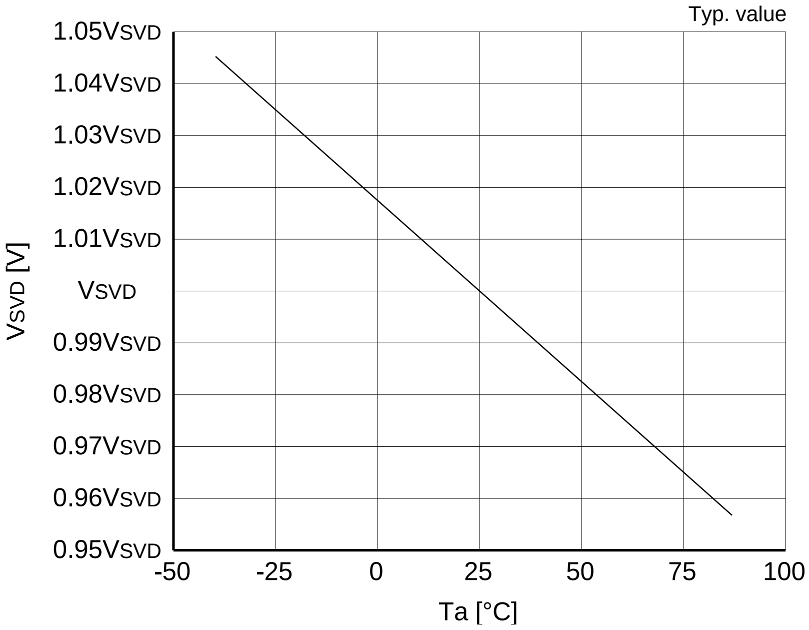
■ High level output current-voltage characteristic



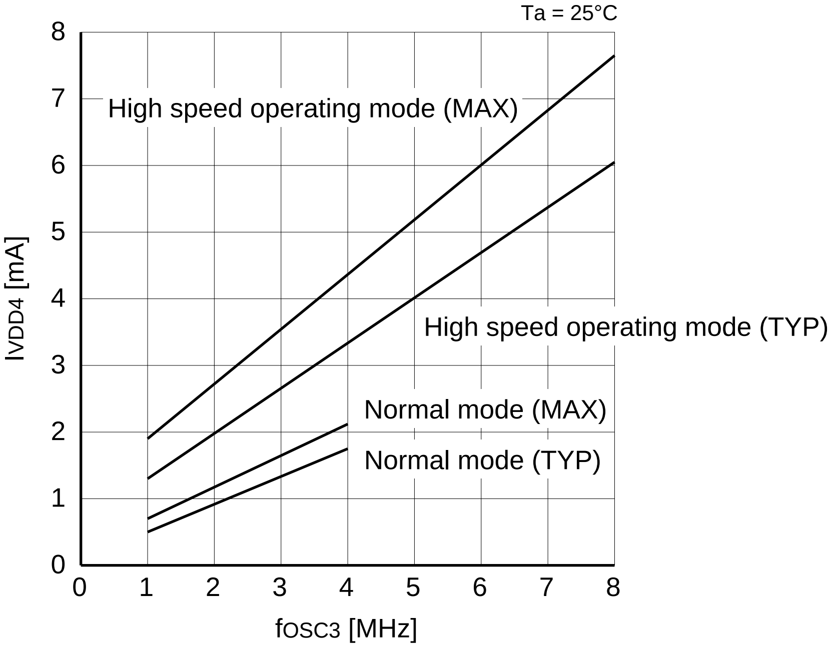
■ Low level output current-voltage characteristic



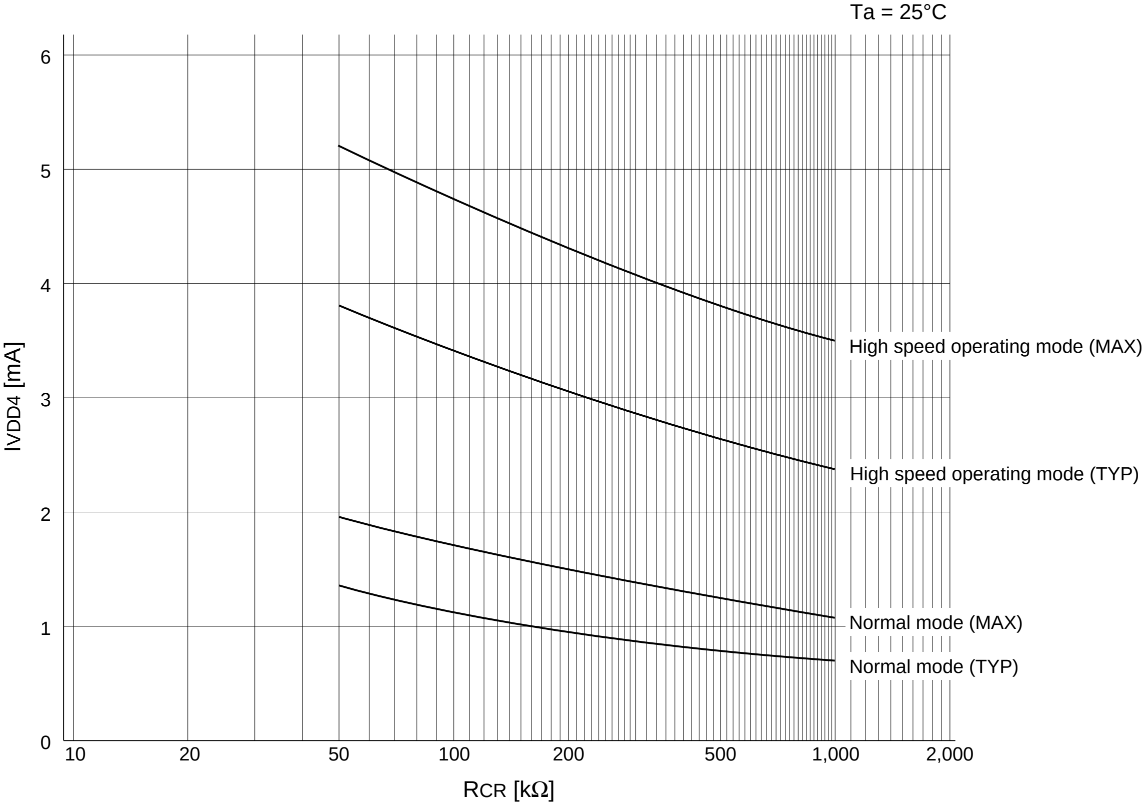
■ SVD voltage-ambient temperature characteristic



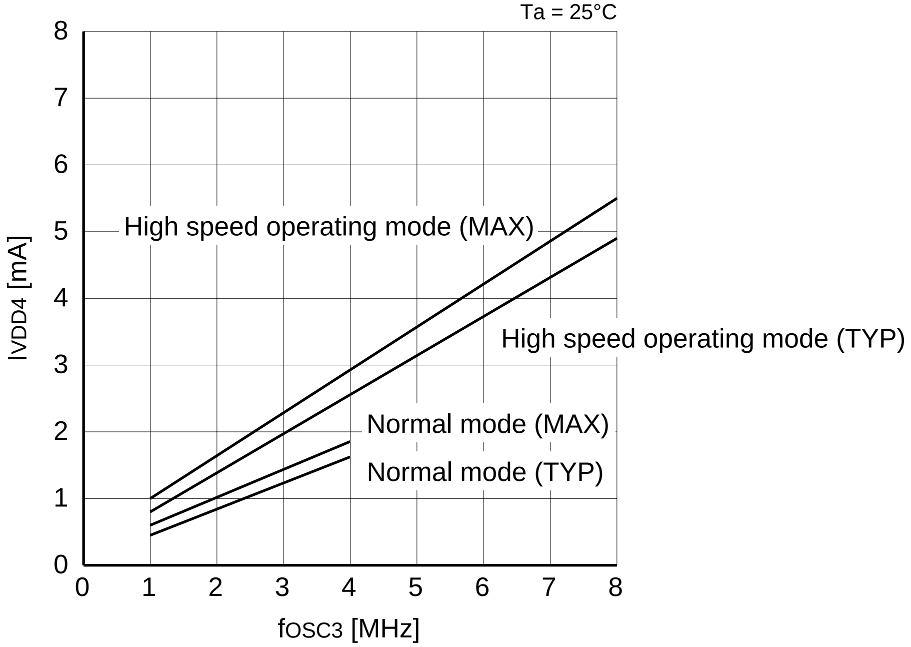
■ Power current (During operation with OSC3) <Crystal oscillation>



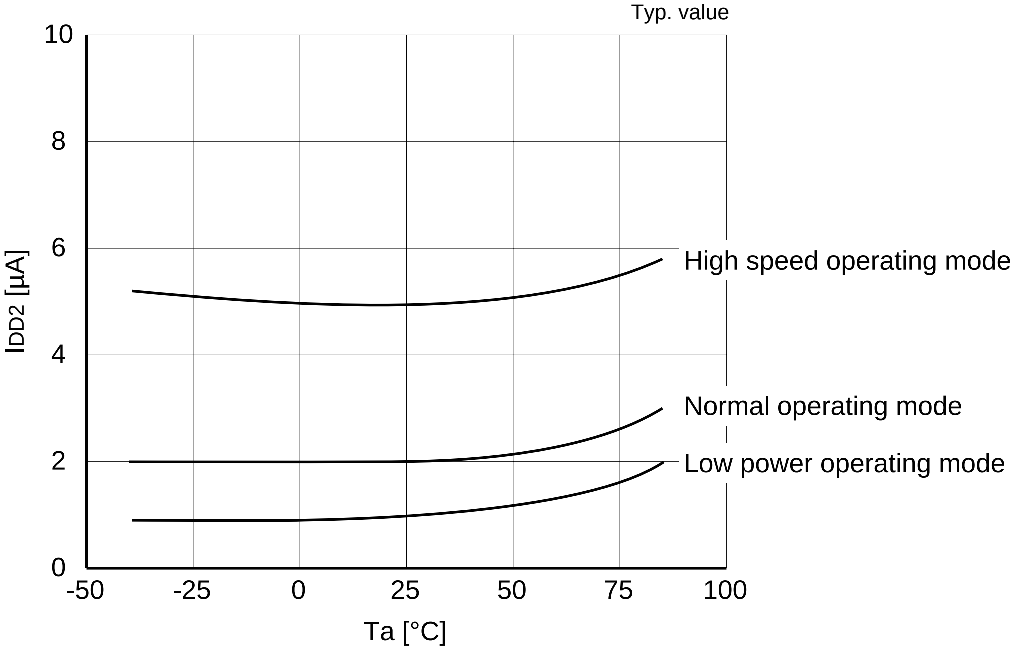
■ Power current (During operation with OSC3) <CR oscillation>



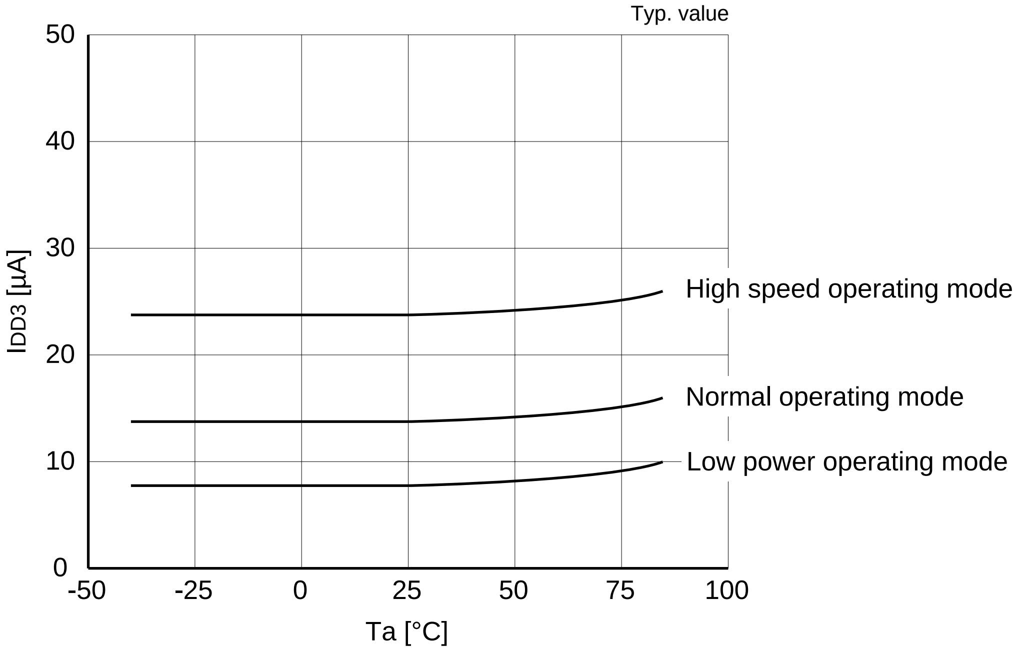
■ Power current (During operation with OSC3) <External clock>



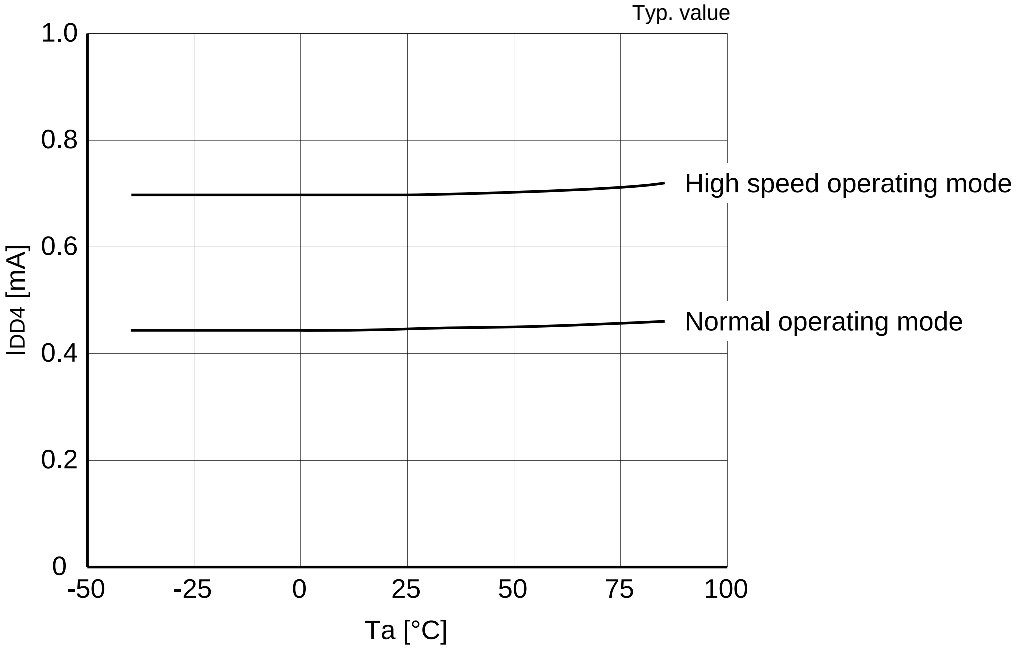
■ Power current-ambient temperature characteristic (In HALT status)



■ Power current-ambient temperature characteristic (CPU is under 32.768 kHz operation)



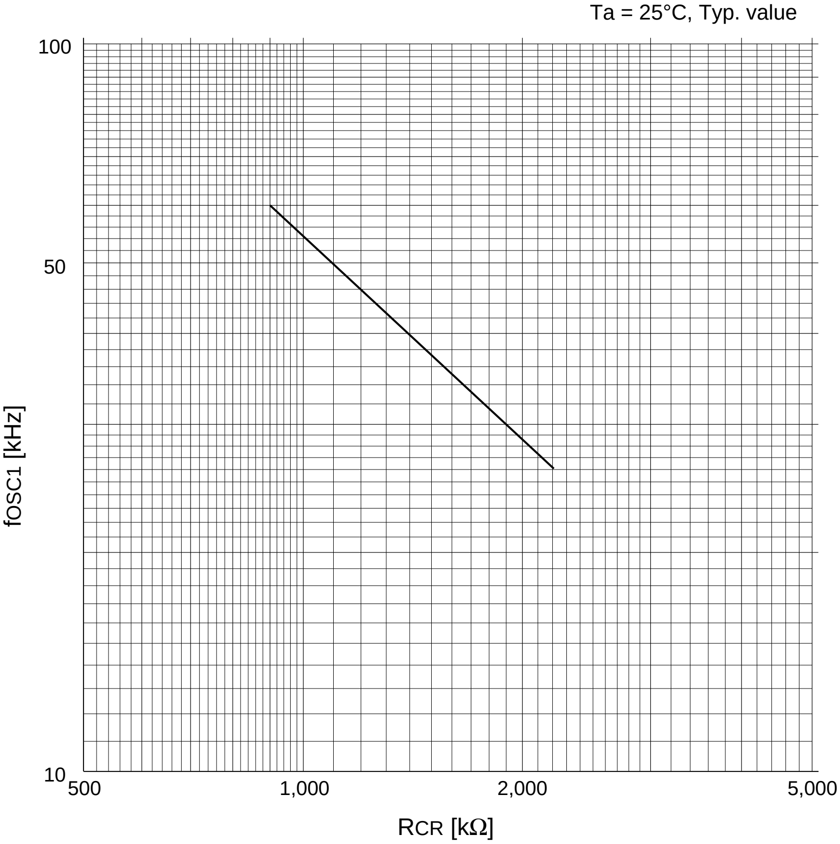
■ Power current-ambient temperature characteristic (CPU is under 1 MHz operation)

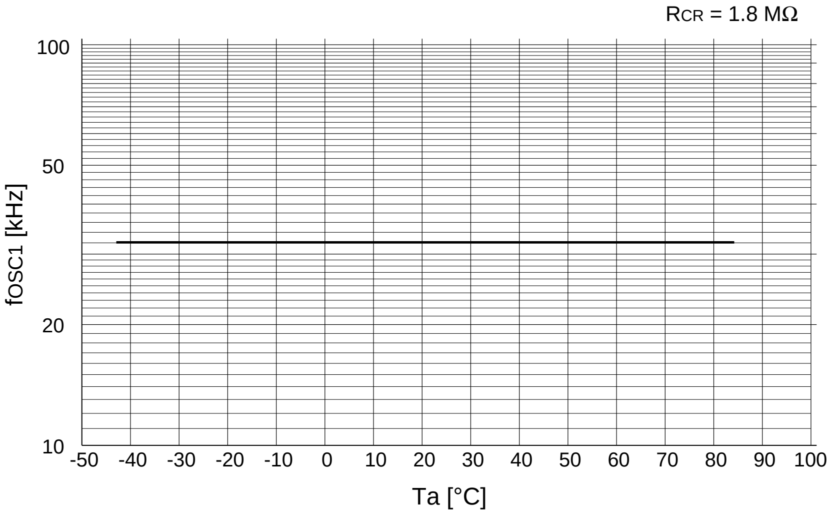


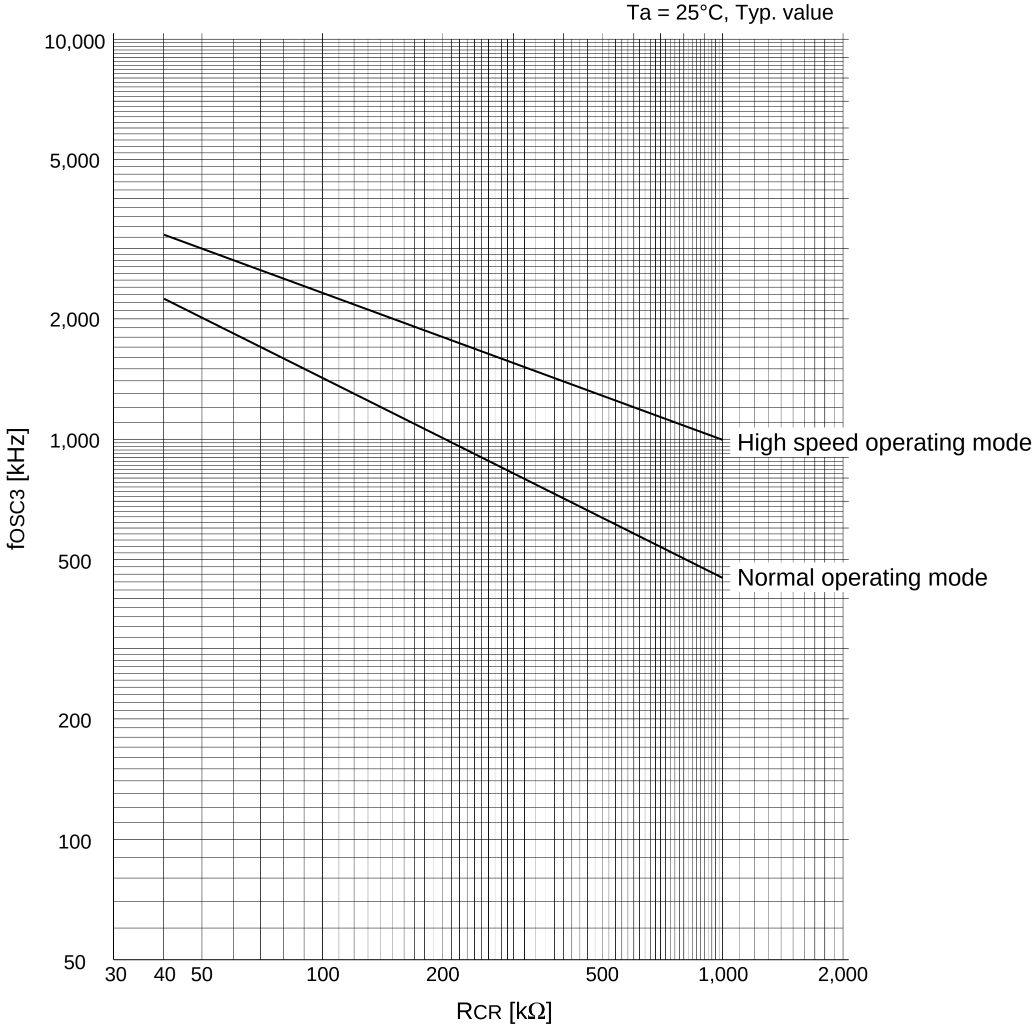
■ CR oscillation frequency characteristic

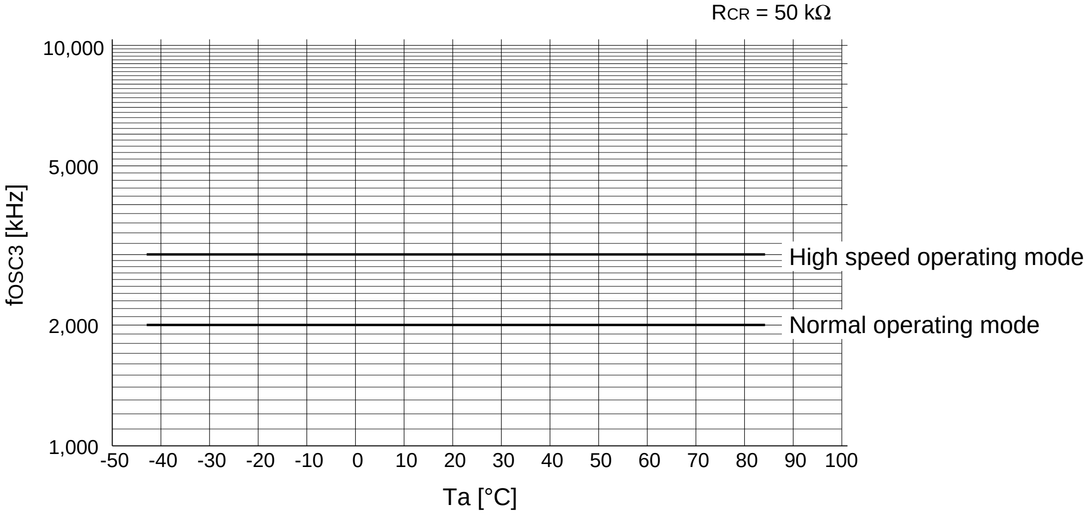
Note: Oscillation frequency changes depending on the conditions (components used, board pattern, etc.). In particular, the OSC3 oscillation frequency changes extensively depending on the product form (chip, plastic package or ceramic package) and board capacitance. Therefore, use the following charts for reference only and select the resistance value after evaluating the actual product. (The resistance value of OSC3 should be set to $R_{CR} \geq 15$ kΩ.)

• Oscillation frequency resistor characteristic





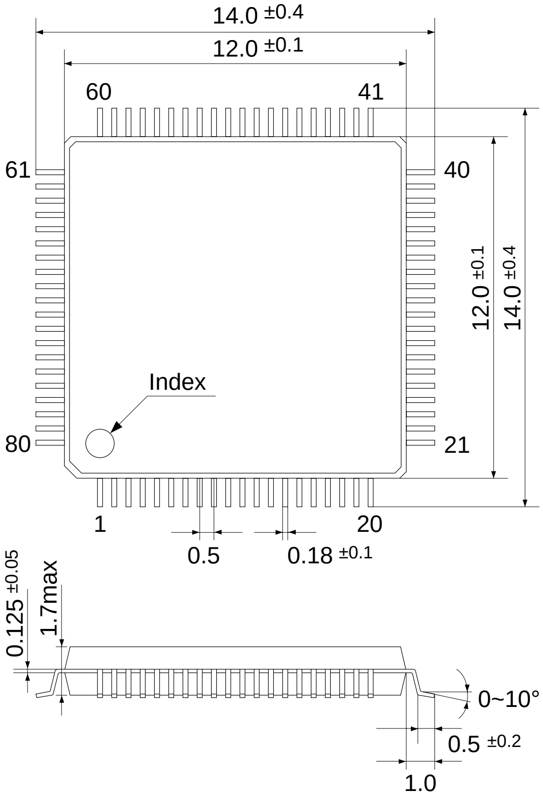

Ta = 25°C, Typ. value
10,000
5,000
2,000
1,000
500
200
100
50
fOSC3 [kHz]
30
40
50
100
200
500
1,000
2,000
RCR [kΩ]
High speed operating mode
Normal operating mode


RCR = 50 kΩ
10,000
5,000
2,000
1,000
fOSC3 [kHz]
-50
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80
90
100
Ta [°C]
High speed operating mode
Normal operating mode

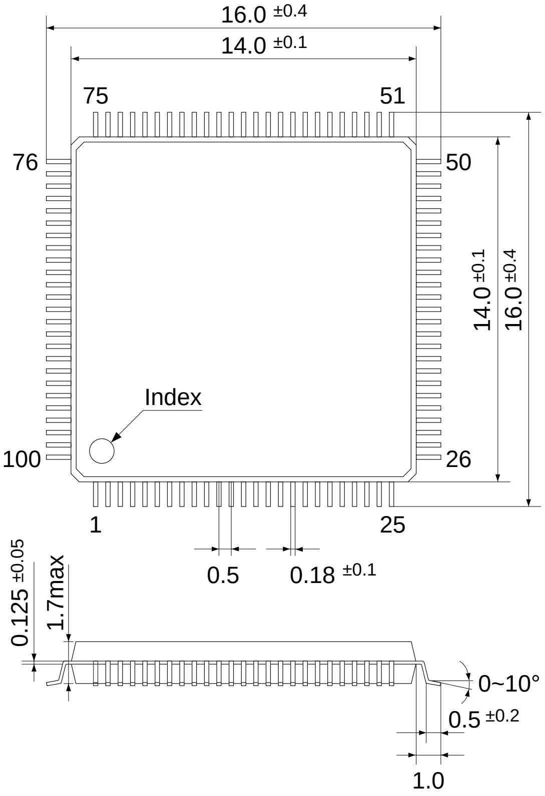
8 PACKAGE

8.1 Plastic Package

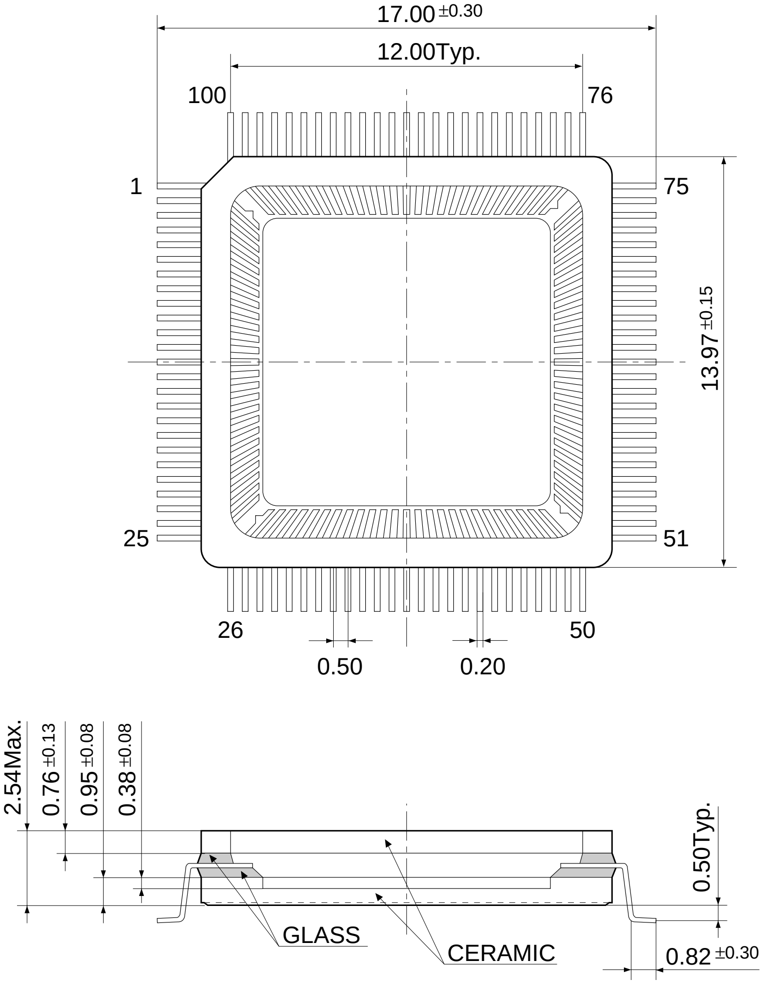
QFP14-80 pin



QFP15-100 pin



8.2 Ceramic Package

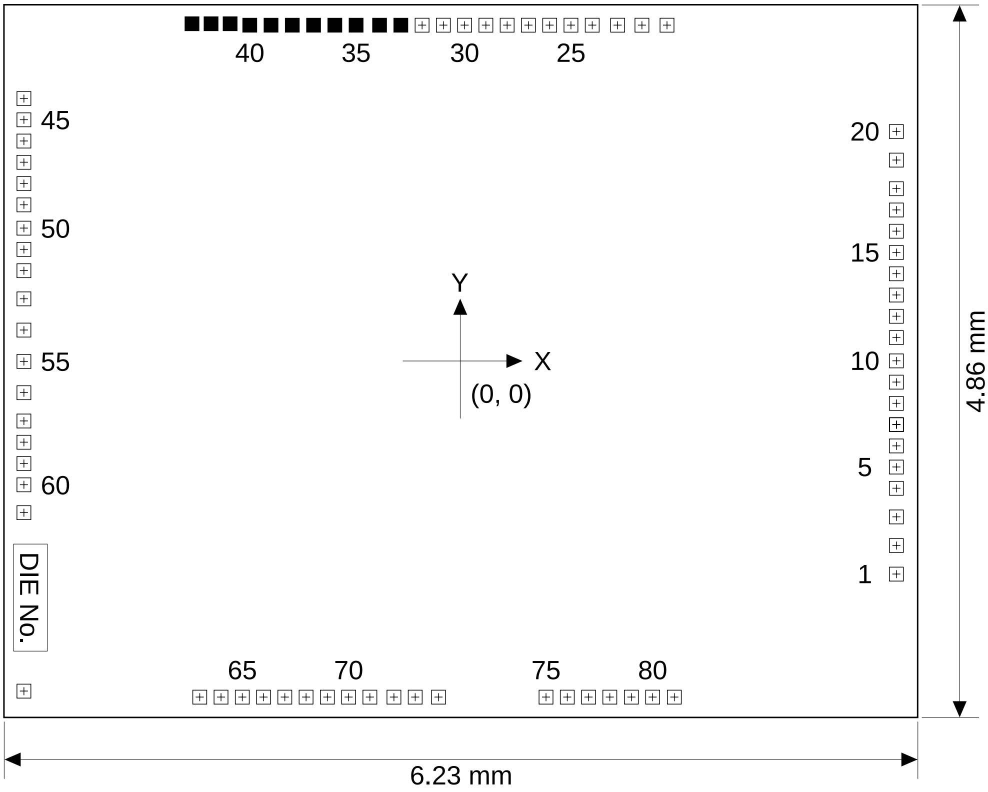


Note: There is no ceramic package corresponding to the QFP14-80 pin plastic package. Signals corresponding to the pins are the same as the QFP15-100 pin package.

9 PAD LAYOUT

9.1 Diagram of Pad Layout

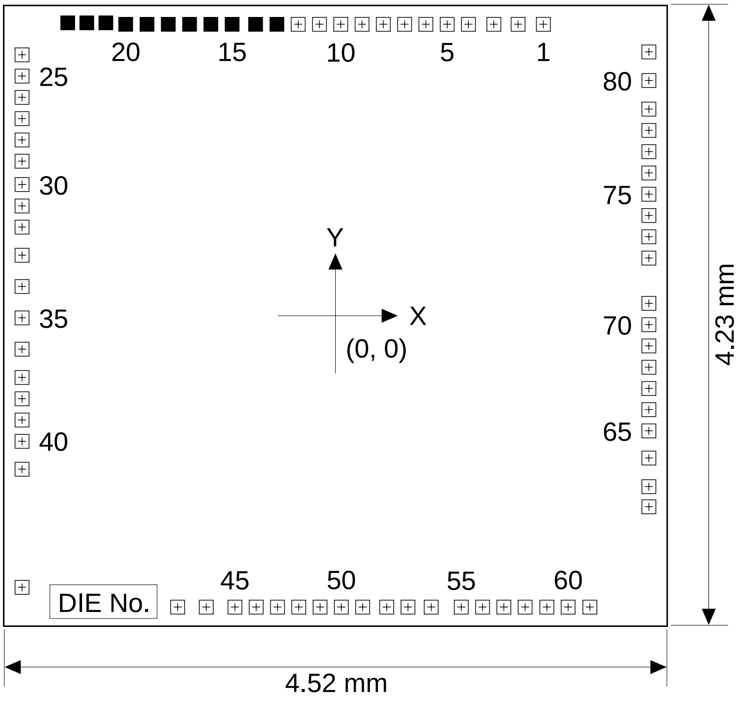
E0C88112



Chip thickness: 0.4 mm
Pad opening: 95 µm

■ Pads are used for the IC outgoing test, so you should not bond them.

E0C88104



Chip thickness: 0.4 mm
Pad opening: 95 µm

■ Pads are used for the IC outgoing test, so you should not bond them.

9.2 Pad Coordinates

Table 9.2.1 Pad coordinates (E0C88112) (Unit: mm)

No.	Name	X	Y	No.	Name	X	Y	No.	Name	X	Y
1	R00/A0	2.975	-1.452	28	R33/$\overline{\text{CE3}}$	0.321	2.291	55	P06/D6	-2.974	-0.002
2	R01/A1	2.975	-1.257	29	R34/FOUT	0.176	2.291	56	P05/D5	-2.974	-0.215
3	R02/A2	2.975	-1.062	30	R35	0.031	2.291	57	P04/D4	-2.974	-0.408
4	R03/A3	2.975	-0.867	31	R36	-0.114	2.291	58	P03/D3	-2.974	-0.553
5	R04/A4	2.975	-0.722	32	R37	-0.259	2.291	59	P02/D2	-2.974	-0.698
6	R05/A5	2.975	-0.578	33	*	-0.404	2.291	60	P01/D1	-2.974	-0.843
7	R06/A6	2.975	-0.433	34	*	-0.549	2.291	61	P00/D0	-2.974	-1.033
8	R07/A7	2.975	-0.288	35	*	-0.709	2.291	62	$\overline{\text{TEST}}$	-2.974	-2.250
9	R10/A8	2.975	-0.143	36	*	-0.854	2.291	63	$\overline{\text{RESET}}$	-1.775	-2.291
10	R11/A9	2.975	0.002	37	*	-0.999	2.291	64	K11/$\overline{\text{BREQ}}$	-1.630	-2.291
11	R12/A10	2.975	0.161	38	*	-1.144	2.291	65	K10/EVIN	-1.485	-2.291
12	R13/A11	2.975	0.306	39	*	-1.289	2.291	66	K07	-1.340	-2.291
13	R14/A12	2.975	0.451	40	*	-1.434	2.291	67	K06	-1.195	-2.291
14	R15/A13	2.975	0.596	41	*	-1.568	2.301	68	K05	-1.050	-2.291
15	R16/A14	2.975	0.741	42	*	-1.698	2.301	69	K04	-0.905	-2.291
16	R17/A15	2.975	0.886	43	*	-1.828	2.301	70	K03	-0.760	-2.291
17	R20/A16	2.975	1.031	44	R50/BZ	-2.974	1.791	71	K02	-0.615	-2.291
18	R21/A17	2.975	1.176	45	R51/$\overline{\text{BACK}}$	-2.974	1.646	72	K01	-0.451	-2.291
19	R22/A18	2.975	1.371	46	P17/CMPM1	-2.974	1.501	73	K00	-0.306	-2.291
20	R23/$\overline{\text{RD}}$	2.975	1.566	47	P16/CMPP1	-2.974	1.356	74	MCU/$\overline{\text{MPU}}$	-0.147	-2.291
21	R24/$\overline{\text{WR}}$	1.411	2.291	48	P15/CMPM0	-2.974	1.211	75	V_{DD}	0.586	-2.291
22	R25	1.240	2.291	49	P14/CMPP0	-2.974	1.066	76	OSC4	0.731	-2.291
23	R26	1.074	2.291	50	P13/$\overline{\text{SRDY}}$	-2.974	0.907	77	OSC3	0.876	-2.291
24	R27/TOUT	0.901	2.291	51	P12/$\overline{\text{SCLK}}$	-2.974	0.762	78	V_{D1}	1.021	-2.291
25	R30/$\overline{\text{CE0}}$	0.756	2.291	52	P11/SOUT	-2.974	0.617	79	OSC2	1.168	-2.291
26	R31/$\overline{\text{CE1}}$	0.611	2.291	53	P10/SIN	-2.974	0.425	80	OSC1	1.313	-2.291
27	R32/$\overline{\text{CE2}}$	0.466	2.291	54	P07/D7	-2.974	0.212	81	V_{SS}	1.461	-2.291

* Pads (No.33–43) are used for the IC shipment test, so you should not bond them.

Table 9.2.2 Pad coordinates (E0C88104) (Unit: mm)

No.	Name	X	Y	No.	Name	X	Y	No.	Name	X	Y
1	R24/$\overline{\text{WR}}$	1.416	1.986	28	P15/CMPM0	-2.135	1.198	55	V_{DD}	0.857	-1.986
2	R25	1.244	1.986	29	P14/CMPP0	-2.135	1.053	56	OSC4	1.002	-1.986
3	R26	1.079	1.986	30	P13/$\overline{\text{SRDY}}$	-2.135	0.894	57	OSC3	1.147	-1.986
4	R27/TOUT	0.906	1.986	31	P12/$\overline{\text{SCLK}}$	-2.135	0.748	58	V_{D1}	1.292	-1.986
5	R30/$\overline{\text{CE0}}$	0.761	1.986	32	P11/SOUT	-2.135	0.604	59	OSC2	1.440	-1.986
6	R31/$\overline{\text{CE1}}$	0.616	1.986	33	P10/SIN	-2.135	0.412	60	OSC1	1.585	-1.986
7	R32/$\overline{\text{CE2}}$	0.471	1.986	34	P07/D7	-2.135	0.199	61	V_{SS}	1.733	-1.986
8	R33/$\overline{\text{CE3}}$	0.326	1.986	35	P06/D6	-2.135	-0.015	62	R00/A0	2.135	-1.302
9	R34/FOUT	0.181	1.986	36	P05/D5	-2.135	-0.228	63	R01/A1	2.135	-1.165
10	R35	0.036	1.986	37	P04/D4	-2.135	-0.421	64	R02/A2	2.135	-0.970
11	R36	-0.109	1.986	38	P03/D3	-2.135	-0.566	65	R03/A3	2.135	-0.785
12	R37	-0.254	1.986	39	P02/D2	-2.135	-0.711	66	R04/A4	2.135	-0.640
13	*	-0.399	1.986	40	P01/D1	-2.135	-0.856	67	R05/A5	2.135	-0.496
14	*	-0.544	1.986	41	P00/D0	-2.135	-1.046	68	R06/A6	2.135	-0.351
15	*	-0.704	1.986	42	$\overline{\text{TEST}}$	-2.135	-1.851	69	R07/A7	2.135	-0.206
16	*	-0.849	1.986	43	$\overline{\text{RESET}}$	-1.075	-1.986	70	R10/A8	2.135	-0.061
17	*	-0.994	1.986	44	K11/$\overline{\text{BREQ}}$	-0.880	-1.986	71	R11/A9	2.135	0.084
18	*	-1.139	1.986	45	K10/EVIN	-0.685	-1.986	72	R12/A10	2.135	0.393
19	*	-1.284	1.986	46	K07	-0.540	-1.986	73	R13/A11	2.135	0.538
20	*	-1.429	1.986	47	K06	-0.395	-1.986	74	R14/A12	2.135	0.683
21	*	-1.564	1.996	48	K05	-0.250	-1.986	75	R15/A13	2.135	0.828
22	*	-1.694	1.996	49	K04	-0.105	-1.986	76	R16/A14	2.135	0.973
23	*	-1.824	1.996	50	K03	0.040	-1.986	77	R17/A15	2.135	1.118
24	R50/BZ	-2.135	1.778	51	K02	0.185	-1.986	78	R20/A16	2.135	1.263
25	R51/$\overline{\text{BACK}}$	-2.135	1.633	52	K01	0.349	-1.986	79	R21/A17	2.135	1.408
26	P17/CMPM1	-2.135	1.488	53	K00	0.494	-1.986	80	R22/A18	2.135	1.603
27	P16/CMPP1	-2.135	1.343	54	MCU/$\overline{\text{MPU}}$	0.652	-1.986	81	R23/$\overline{\text{RD}}$	2.135	1.798

* Pads (No.13–23) are used for the IC shipment test, so you should not bond them.

II E0C88112/88104 Technical Software

Contents

PREFACE *II-1*

Description

Notes for Program Example Use

1 SYSTEM INITIALIZATION *II-2*

Initialization for E0C88112 single chip mode

2 SYSTEM CONTROLLER AND BUS CONTROL *II-4*

(1) E0C88112 Single chip mode
(2) E0C88112 MCU Expanded 64K mode
(3) E0C88112 MPU Expanded 64K mode
(4) E0C88112 MPU Expanded 512K maximum mode

3 WATCHDOG TIMER *II-9*

(1) Watchdog timer reset
(2) Watchdog timer ($\overline{\text{NMI}}$) interrupt processing

4 OSCILLATION CIRCUIT *II-11*

(1) Switching from OSC1 to OSC3
(2) Switching from OSC3 to OSC1

5 INPUT PORTS (K PORTS) *II-14*

(1) Data reading from normal input port (K port)
(2) Key scan for 8 x 2 key matrix
(3) Interrupt condition setting and interrupt processing for input port (K port)

6 OUTPUT PORTS (R PORTS) *II-21*

(1) Normal DC output
(2) High impedance output control
(3) FOUT output control

7 I/O PORTS (P PORTS) *II-27*

(1) Normal data input/output of I/O port
(2) Scan for 2 x 2 switch matrix

8 SERIAL INTERFACE 1 (CLOCK SYNCHRONOUS INTERFACE) *II-33*

(1) Initialization for clock synchronous serial interface (master mode)
(2) Receiving of clock synchronous serial interface (master mode)
(3) Transmitting of clock synchronous serial interface (master mode)

9 SERIAL INTERFACE 2 (ASYNCHRONOUS INTERFACE) *II-42*

(1) Initialization for asynchronous serial interface (8-bit mode)
(2) Receiving of asynchronous serial interface (8-bit mode)
(3) Transmitting of asynchronous serial interface (8-bit mode)

10 CLOCK TIMER *II-54*

(1) Initialization for clock timer
(2) Clock timer interrupt processing

11 STOPWATCH TIMER II-59

(1) Initialization for stopwatch timer

(2) Stopwatch timer interrupt processing

12 PROGRAMMABLE TIMER II-64

(1) Initialization and interrupt processing for 8-bit reload timer (two channels)

(2) Initialization and interrupt processing for 16-bit one-shot timer (one channel)

(3) Initialization and interrupt processing for 8-bit event counter

(4) Initialization and interrupt processing for 16-bit pulse width measurement timer

(5) 16-bit reload timer pulse output

13 SOUND GENERATOR II-88

(1) Initialization for sound generator

(2) Normal buzzer output

(3) Buzzer output with digital envelope

(4) One-shot buzzer output

14 ANALOG COMPARATOR.. II-93

(1) Initialization for analog comparator

(2) Data reading for analog comparator

15 SVD (SUPPLY VOLTAGE DETECTION) CIRCUIT II-95

(1) Supply voltage detection in 1/4 Hz auto sampling mode

(2) Supply voltage detection in continuously sampling mode

16 INTERRUPT (EXCEPTION) PROCESSING II-97

(1) Interrupt level setting and enables interrupt

(2) Zero division exception processing

(3) Watchdog timer ($\overline{\text{NMI}}$) interrupt processing

(4) Interrupt processing for peripheral circuit

17 EXPANDED MODE II-105

(1) Access for program memory outside logical space

(2) Data block transfer between pages

(3) Access for data outside page

Appendix A Table of E0C88112/88104 Input/Output Port Terminals.............. II-108

Appendix B Instruction List.. II-110

Appendix C Programming Notes II-127

PREFACE

In this part, example of a control programs for each peripheral circuit are described. Basic initialization and control routines are shown in the program examples use a relocatable method and are based on the assumption that the cross assembler asm88 for the E0C88 Family is being used. When you create an application program referring to these examples, use them after completion of the program by adding the necessary functions.

Description

Program examples are shown by each peripheral circuit or function, according to the following items.

I/O MAP	Indicates the I/O memory map that controls the peripheral circuit. See Part I in this manual, "E0C88112/ 88104 Technical Hardware", for details of the control registers and operation.
Specification	Indicates the purpose, function, etc., of the example routine.
Flowchart	Indicates a flowchart of the example.
Note	Indicates matters that require attention when using the example routine and for programming of the peripheral circuit.
Source List	A source code listing using the relocatable method in assembly language. See the "E0C88 Core CPU Manual" for details of the instructions and the "E0C88 Family Structured Assembler Manual" for the assembly language and the format of the source list.

Notes for Program Example Use

Take the following precautions when reading this manual and using the described routines:

(1) Each program example has been modularized as a low-level routine that controls hardware directly, and examples such as a concrete application have not been included. For a routine to be added by the user, an external declaration with a label such as "user_program" should be made and the program will branch to the label. Because the name "user_program" is not very descriptive, you should modify the label name to reflect its function.

(2) In the program examples, 8-bit absolute addressing has been used for I/O memory access. Consequently, the program loads the upper 8 bits (0FFH) of the I/O memory base address (00FF00H) into the BR register. This part in the flowchart is described as (BR setting) and it is set in each program example. If you use another addressing mode, rewriting this part is necessary.

(3) These routines do not specify bank or page. When using in the expanded mode, set the bank and page if necessary.

(4) Input, output and I/O port terminals of the E0C88112 are shared the a bus and special output, and these functions are set by software. Be aware that the port configuration will be changed by these setting. Refer to the terminal configuration tables according to the mode and special output settings which have been mentioned in the "Appendix".

(5) Unary operators set in the asm88 cross assembler have been used for the program examples. These unary operators get the values below from a constant or a label operand.

low ...	Presents the lower 8 bits of the expression.
high ..	Returns the upper 8 bits of a 16-bit expression.
boc ...	Calculates a bank value from the physical address.
loc	Calculates a logical address in a bank from the physical address.
pod ..	Calculates a page value from the physical address.
lod	Calculates a logical address in a page from the physical address.

(6) Seiko Epson assumes no responsibility for any consequences arising from the use of the programs described.

Note: The program examples are created for the E0C88112. The built-in ROM capacity of the E0C88104 is different from the E0C88112, but the other peripheral circuits are made with the same configuration.

1 SYSTEM INITIALIZATION

I/O Map

Refer to the peripheral circuit descriptions in this manual.

Table 1.1 Configuration

Model	Internal ROM	Internal RAM
E0C88112	12K byte	256 byte
E0C88104	4K byte	256 byte

Specification

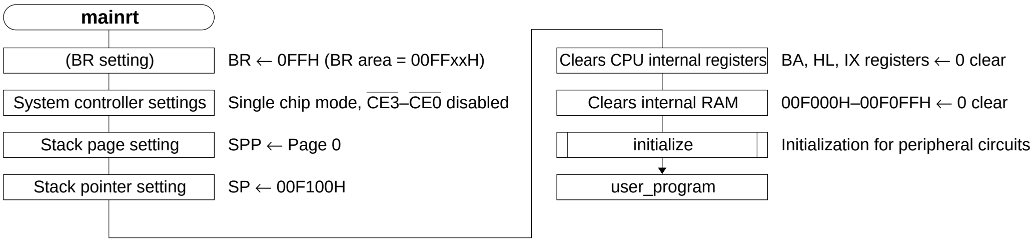
Initialization for E0C88112 single chip mode

mainrt: Initialization for E0C88112 single chip mode

Settings of the base register, CPU mode, $\overline{\text{CE}}$ output, stack page, stack pointer, wait and bus authority release signal and clearing of RAM are done sequentially.

Flowchart

Initialization for E0C88112 single chip mode


mainrt
(BR setting)
BR ← 0FFH (BR area = 00FFxxH)
System controller settings
Single chip mode, $\overline{\text{CE3}}$–$\overline{\text{CE0}}$ disabled
Stack page setting
SPP ← Page 0
Stack pointer setting
SP ← 00F100H
Clears CPU internal registers
BA, HL, IX registers ← 0 clear
Clears internal RAM
00F000H–00F0FFH ← 0 clear
initialize
Initialization for peripheral circuits
user_program

Notes

(1) Interrupts have been set to their initial status (all disabled) except for the watchdog timer ($\overline{\text{NMI}}$) interrupt which cannot be masked.

(2) Be sure to declare the watchdog timer ($\overline{\text{NMI}}$) interrupt processing routine and the vector address, regardless of whether or not the watchdog timer is used.

(3) For peripheral circuit initialization, you must create a separate routine according to the system configuration to be used. (external call: initialize)

(4) For the interrupt flags (I0 and I1), set them to adapt to the interrupt factor and priority level of the peripheral I/O that will be enabled.

(5) When using the peripheral I/O interrupt, declare the front address of the peripheral I/O interrupt processing routine in a vector address corresponding to the interrupt in the order of lower and upper. (Vector address: 000006H–000023H)

(6) Vector addresses 000026H–0000FFH can be set for software interrupts. In this case as well as the above, declare the front address of the software interrupt processing routine in a vector address of the software interrupt in the order of lower and upper.

(7) The vector addresses 000024H and 000025H cannot be used since this is a system reserved area for the E0C88112.

(8) In this initialization routine example, the vector address setting and program have been allocated from 000100H for the sake of convenience.

Source List

Initialization for E0C88112 single chip mode

```
         external    initialize,watchdog_reset
         external    user_program
         public      mainrt
;
reset_vector   equ   000000h                  ;reset vector address
main           equ   000100h                  ;program start address offset
;
br_io          equ   0ffh                     ;base reg. address (set i/o area)
mcu            equ   00ff00h                  ;mcu mode system control address
spp            equ   00ff01h                  ;stack pointer page address
mode           equ   00ff02h                  ;mpu//mcu mode control address
;
sp_112         equ   00f100h                  ;E0C88112 stack pointer top address
;
internal_ram   equ   00f000h                  ;E0C88112 internal ram top address
;
         code
intr_vectors:
         org   intr_vectors+reset_vector
         dw    mainrt                         ;initial reset program address
;
         org   intr_vectors+main
;*************************************************************************
;*                                                                       *
;*       E0C88112 mcu single-chip mode initialize                        *
;*                                                                       *
;*************************************************************************
mainrt:
         ld    br,#br_io                      ;set br reg. address to 0ffxxh
         ld    [br:low mcu],#00110000b        ;single chip mode, /ce3-/ce0 disable
         ld    [br:low spp],#00h              ;set stack pointer page to 0
         ld    sp,#lod sp_112                 ;satck pointer top address set
         ld    [br:low mode],#00000000b       ;set mode reg.
         ld    ba,#0000h                      ;internal reg. clear
         ld    hl,#0000h
         ld    ix,#0000h
;internal ram clear
         carl  watchdog_reset                 ;watchdog timer reset ***
         ld    iy,#lod internal_ram           ;E0C88112 internal ram top address
mainrt00:
         ld    [iy],a                         ;clear data set
         inc   iy                             ;poniter increment
         cp    iy,#lod internal_ram+0100h     ;internal ram end ?
         jrs   nz,mainrt00
;
         carl  initialize                     ;initialize i/o area ***
         jrl   user_program                   ;jump user program
;
;start user program
;
         end
```

2 SYSTEM CONTROLLER AND BUS CONTROL

I/O Map (MCU mode)

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF00 (MCU)	D7	BSMD1	Bus mode (CPU mode) BSMD1 BSMD0 Mode 1 1 512K (Maximum)			0	R/W	
	D6	BSMD0	1 0 512K (Minimum) 0 1 64K 0 0 Single chip			0	R/W	
	D5	CEMD1	Chip enable mode CEMD1 CEMD0 Mode 1 1 64K ($\overline{CE0}$)			1	R/W	Only for 64K bus mode
	D4	CEMD0	1 0 32K ($\overline{CE0}$, $\overline{CE1}$) 0 1 16K ($\overline{CE1}$–$\overline{CE3}$) 0 0 8K ($\overline{CE0}$–$\overline{CE3}$)			1	R/W	
	D3	CE3	$\overline{CE3}$ (R33) $\overline{CE}$ signal output Enable/Disable Enable: $\overline{CE}$ signal output Disable: DC (R3x) output	$\overline{CE3}$ enable	$\overline{CE3}$ disable	0	R/W	In the Single chip mode, these setting are fixed at DC output.
	D2	CE2	$\overline{CE2}$ (R32)	$\overline{CE2}$ enable	$\overline{CE2}$ disable	0	R/W	
	D1	CE1	$\overline{CE1}$ (R31)	$\overline{CE1}$ enable	$\overline{CE1}$ disable	0	R/W	
	D0	CE0	$\overline{CE0}$ (R30)	$\overline{CE0}$ enable	$\overline{CE0}$ disable	0	R/W	
00FF01	D7	SPP7	Stack pointer page address (MSB)	1	0	0	R/W	
	D6	SPP6		1	0	0	R/W	
	D5	SPP5	< SP page allocatable address >	1	0	0	R/W	
	D4	SPP4	• Single chip mode: only 0 page	1	0	0	R/W	
	D3	SPP3	• 64K mode: only 0 page	1	0	0	R/W	
	D2	SPP2	• 512K (min) mode: 0–27H page	1	0	0	R/W	
	D1	SPP1	• 512K (max) mode: 0–27H page	1	0	0	R/W	
	D0	SPP0	(LSB)	1	0	0	R/W	
00FF02	D7	EBR	Bus release enable register K11 (K11 and R51 terminal specification) R51	$\overline{BREQ}$ $\overline{BACK}$	Input port Output port	0	R/W	
	D6	WT2	Wait control register WT2 WT1 WT0 Number of state 1 1 1 14 1 1 0 12			0	R/W	
	D5	WT1	1 0 1 10 1 0 0 8 0 1 1 6					
	D4	WT0	0 1 0 4 0 0 1 2 0 0 0 No wait					
	D3	CLKCHG	CPU operating clock switch	OSC3	OSC1	0	R/W	
	D2	OSCC	OSC3 oscillation On/Off control	On	Off	0	R/W	
	D1	VDC1	Operating mode selection VDC1 VDC0 Operating mode			0	R/W	
	D0	VDC0	1 × High speed (V_{D1}=3.3V) 0 1 Low power (V_{D1}=1.3V) 0 0 Normal (V_{D1}=2.2V)					

Note: All the interrupts including $\overline{NMI}$ are disabled, until you write the optional value into both the "00FF00H" and "00FF01H" addresses.

I/O Map (MPU mode)

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF00 (MPU)	D7	BSMD1	Bus mode (CPU mode) BSMD1 BSMD0 Mode 1 1 512K (Maximum)			*	R/W	* Initial setting can be selected among 3 types (64K, 512K min and 512K max) by mask option setting.
	D6	BSMD0	1 0 512K (Minimum) 0 1 64K 0 0 * Option selection			*	R/W	
	D5	CEMD1	Chip enable mode CEMD1 CEMD0 Mode 1 1 64K ($\overline{CE0}$)			1	R/W	Only for 64K bus mode
	D4	CEMD0	1 0 32K ($\overline{CE0}$, $\overline{CE1}$) 0 1 16K ($\overline{CE0}$–$\overline{CE3}$) 0 0 8K ($\overline{CE0}$–$\overline{CE3}$)			1	R/W	
	D3	CE3	$\overline{CE3}$ (R33) $\overline{CE}$ signal output Enable/Disable	$\overline{CE3}$ enable	$\overline{CE3}$ disable	0	R/W	
	D2	CE2	$\overline{CE2}$ (R32) Enable: $\overline{CE}$ signal output	$\overline{CE2}$ enable	$\overline{CE2}$ disable	0	R/W	
	D1	CE1	$\overline{CE1}$ (R31) Disable: DC (R3x) output	$\overline{CE1}$ enable	$\overline{CE1}$ disable	0	R/W	
	D0	CE0	$\overline{CE0}$ (R30)	$\overline{CE0}$ enable	$\overline{CE0}$ disable	1	R/W	
00FF01	D7	SPP7	Stack pointer page address (MSB)	1	0	0	R/W	
	D6	SPP6		1	0	0	R/W	
	D5	SPP5	< SP page allocatable address >	1	0	0	R/W	
	D4	SPP4	• Single chip mode: only 0 page	1	0	0	R/W	
	D3	SPP3	• 64K mode: only 0 page	1	0	0	R/W	
	D2	SPP2	• 512K (min) mode: 0–27H page	1	0	0	R/W	
	D1	SPP1	• 512K (max) mode: 0–27H page	1	0	0	R/W	
	D0	SPP0	(LSB)	1	0	0	R/W	
00FF02	D7	EBR	Bus release enable register K11 (K11 and R51 terminal specification) R51	$\overline{BREQ}$ $\overline{BACK}$	Input port Output port	0	R/W	
	D6	WT2	Wait control register WT2 WT1 WT0 Number of state 1 1 1 14 1 1 0 12			0	R/W	
	D5	WT1	1 0 1 10 1 0 0 8 0 1 1 6					
	D4	WT0	0 1 0 4 0 0 1 2 0 0 0 No wait					
	D3	CLKCHG	CPU operating clock switch	OSC3	OSC1	0	R/W	
	D2	OSCC	OSC3 oscillation On/Off control	On	Off	0	R/W	
	D1	VDC1	Operating mode selection VDC1 VDC0 Operating mode			0	R/W	
	D0	VDC0	1 × High speed (V_{D1}=3.3V) 0 1 Low power (V_{D1}=1.3V) 0 0 Normal (V_{D1}=2.2V)					

Note: All the interrupts including $\overline{NMI}$ are disabled, until you write the optional value into both the "00FF00H" and "00FF01H" addresses.

Specifications

System controller settings and bus control

(1) single_chip: E0C88112 Single chip mode
(2) mcu64k_112: E0C88112 MCU Expanded 64K mode
(3) mpu_64k: E0C88112 MPU Expanded 64K mode
(4) mpu512k_max: E0C88112 MPU Expanded 512K maximum mode

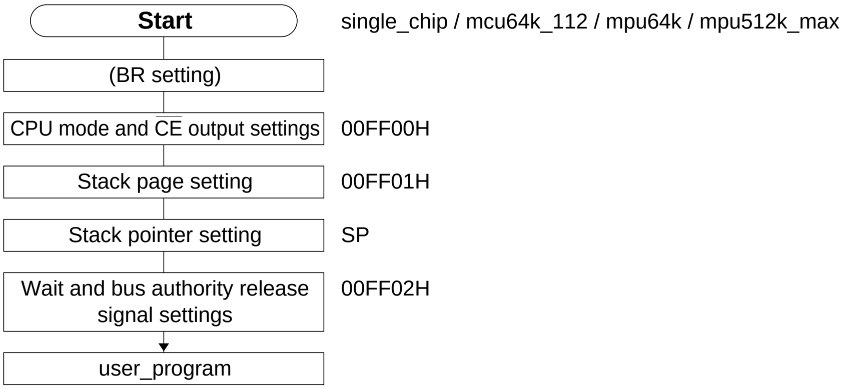
Each of the routines sets the system controller and bus as shown in the table below.

Table 2.1 Setting contents of each routine

Address	Setting item	(1)	(2)	(3)	(4)
0FF00H	CPU mode	Single chip	64K	64K	512K
	Chip enable mode	No	$\overline{CE1}$–$\overline{CE0}$ (32K)	$\overline{CE0}$ (64K)	$\overline{CE3}$–$\overline{CE0}$ (128K)
	$\overline{CE}$ signal output	No	$\overline{CE1}$–$\overline{CE0}$	$\overline{CE0}$	$\overline{CE3}$–$\overline{CE0}$
0FF01H	Stack page	Page 0	←	←	Page 27H
0FF02H	Bus release	No	←	←	Use
	Wait control	No	4	8	0
	CPU operating clock	OSC1	←	←	←
	OSC3 oscillation circuit	Off	←	←	←
	Operating mode	Normal mode	←	←	←

Flowchart

System controller settings and bus control (1), (2), (3), (4)


Start
single_chip / mcu64k_112 / mpu64k / mpu512k_max
(BR setting)
CPU mode and CE output settings
00FF00H
Stack page setting
00FF01H
Stack pointer setting
SP
Wait and bus authority release signal settings
00FF02H
user_program

Notes

(1) Prior to any other processing, be sure to set the system controller and bus control in an initialization routine executed immediately after an initial reset.

(2) When using the MPU mode, the output of $\overline{\text{CE0}}$ signal is set to valid at initial reset. Be sure not to set the $\overline{\text{CE0}}$ output to invalid when setting the system controller.

(3) The $\overline{\text{CE0}}$–$\overline{\text{CE3}}$ output terminals are shared with the R30–R33 terminals. Consequently, the terminals which have been set for $\overline{\text{CE}}$ outputs cannot be used as a general purpose output port, including the high impedance control. Moreover, since the output terminals shift to LOW if "0" is written to the R30–R33 registers prior setting the $\overline{\text{CE}}$ outputs, be sure to avoid this.

(4) When using the bus release function, the K11 and R51 terminals function as the $\overline{\text{BREQ}}$ and $\overline{\text{BACK}}$ terminals, respectively. Consequently, K11 and R51 cannot be used as an input port and a output port.

Source List

System controller settings and bus control

```
         external    user_program
         public      single_chip,mcu64k_112,mpu_64k,mpu512k_max
;
br_io          equ   0ffh                     ;base reg. address (set i/o area)
mcu            equ   00ff00h                  ;mcu mode system control address
mpu            equ   00ff00h                  ;mpu mode system control address
spp            equ   00ff01h                  ;stack pointer page address
mode           equ   00ff02h                  ;mcu//mpu mode control address
;
sp_112         equ   00f800h                  ;E0C88112 stack pointer
;
         code
```

(1) E0C88112 Single chip mode

```
;************************************************************************
;*                                                                      *
;*        single chip mode with E0C88112                                *
;*                                                                      *
;************************************************************************
single_chip:
         ld    br,#br_io                      ;set br reg. address to 0ffxxh
         ld    [br:low mcu],#00110000b        ;single chip mode /ce3-/ce0 disable
         ld    [br:low spp],#00h              ;set stack pointer page to 0          (1)
         ld    sp,#lod sp_112                 ;stack pointer set
         ld    [br:low mode],#00000000b       ;set mode reg. to initial value
         jrl   user_program                   ;jump user program
```

(2) E0C88112 MCU Expanded 64K mode

```
;************************************************************************
;*                                                                      *
;*        mcu 64k mode with E0C88112                                    *
;*                                                                      *
;************************************************************************
mcu64k_112:
         ld    br,#br_io                      ;set br reg. address to 0ffxxh
         ld    [br:low mcu],#01100011b        ;mcu 64k mode  /ce1-/ce0(32kb) enable
         ld    [br:low spp],#00h              ;set stack pointer page to 0          (2)
         ld    sp,#lod sp_112                 ;stack pointer set
         ld    [br:low mode],#00100000b       ;set mode reg. to 4 wait states
         jrl   user_program                   ;jump user program
```

Source List

(3) E0C88112 MPU Expanded 64K mode

```
;*************************************************************************
;*                                                                       *
;*        mpu 64k mode with E0C88112                                     *
;*                                                                       *
;*************************************************************************
mpu_64k:
         ld     br,#br_io                       ;set br reg. address to 0ffxxh
         ld     [br:low mcu],#01110001b         ;mpu 64k mode /ce0(64kb) enable
         ld     [br:low spp],#00h               ;set stack pointer page to 0
         ld     sp,#lod sp_112                  ;stack pointer set
         ld     [br:low mode],#01000000b        ;set mode reg. to 8 wait states
         jrl    user_program                    ;jump user program
```

(3)

(4) E0C88112 MPU Expanded 512K maximum mode

```
;*************************************************************************
;*                                                                       *
;*        mpu 512k maximum mode with E0C88112                            *
;*                                                                       *
;*************************************************************************
mpu512k_max:
         ld     br,#br_io                       ;set br reg. address to 0ffxxh
         ld     [br:low mcu],#11111111b         ;mpu 512k mode /ce3-/ce0(128kb) enable
         ld     [br:low spp],#27h               ;set stack pointer page 27h
         ld     sp,#lod sp_112                  ;stack pointer set
         ld     [br:low mode],#10000000b        ;set mode reg. to 0 wait states
;                                               ;and breq,/back enable
         jrl    user_program                    ;jump user program
         end
```

(4)

3 WATCHDOG TIMER

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF40	D7	–	–	–	–	–		"0" when being read
	D6	FOUT2	FOUT frequency selection FOUT2 FOUT1 FOUT0 Frequency 0 0 0 f_{OSC1} / 1			0	R/W	
	D5	FOUT1	0 0 1 f_{OSC1} / 2 0 1 0 f_{OSC1} / 4 0 1 1 f_{OSC1} / 8 1 0 0 f_{OSC3} / 1			0	R/W	
	D4	FOUT0	1 0 1 f_{OSC3} / 2 1 1 0 f_{OSC3} / 4 1 1 1 f_{OSC3} / 8			0	R/W	
	D3	FOUTON	FOUT output control	On	Off	0	R/W	
	D2	WDRST	Watchdog timer reset	Reset	No operation	–	W	Constantly "0" when being read
	D1	TMRST	Clock timer reset	Reset	No operation	–	W	
	D0	TMRUN	Clock timer Run/Stop control	Run	Stop	0	R/W	

Specifications

Watchdog timer processing

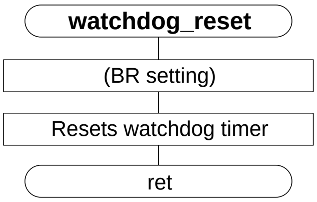
Vector address setting for watchdog timer ($\overline{\text{NMI}}$) interrupt

(1) watchdog_reset: Watchdog timer reset

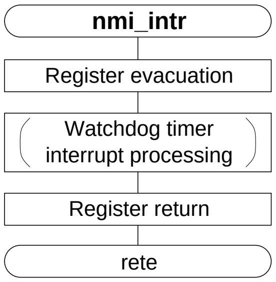
(2) nmi_intr: Watchdog timer ($\overline{\text{NMI}}$) interrupt processing

Flowchart

(1) Watchdog timer reset



(2) Watchdog timer ($\overline{\text{NMI}}$) interrupt processing



Notes

(1) Since the watchdog timer ($\overline{\text{NMI}}$) interrupt cannot be masked, be sure to declare the watchdog timer ($\overline{\text{NMI}}$) interrupt processing routine and the vector address, regardless of whether or not the watchdog timer is used.

(2) In this program example for the watchdog timer, the vector address setting and program have been allocated from 003000H for the sake of convenience.

Source List

Watchdog timer processing

```
         public       watchdog_reset,nmi_intr
;
nmi_vector     equu   000004h                  ;watchdog /nmi interrupt routine
watchdog       equ    003000h                  ;program start address offset
br_io          equ    0ffh                     ;base reg. address (set i/o area)
rtm_mode       equ    00ff40h                  ;timer mode set address
;
;
         code
```

Vector address setting for watchdog timer ($\overline{NMI}$) interrupt

```
intr_vectors:
;
         org    intr_vectors+nmi_vector
         dw     nmi_intr
;
```

(1) Watchdog timer reset

```
         org    intr_vectors+watchdog
;************************************************************************
;*                                                                      *
;*        watchdog timer reset                                          *
;*                                                                      *
;************************************************************************
watchdog_reset:
         ld     br,#br_io                      ;set br reg. address to 0ffxxh          (1)
         or     [br:low rtm_mode],#00000100b   ;watchdogtimer reset
         ret
;
```

(2) Watchdog timer ($\overline{NMI}$) interrupt processing

```
;************************************************************************
;*                                                                      *
;*        /nmi (watchdog) interrupt routine                             *
;*                                                                      *
;************************************************************************
nmi_intr:
         push   ale
;
;
;        /nmi (watchdog) interrupt routine                                             (2)
;
;
         pop    ale
         rete
         end
```

4 OSCILLATION CIRCUIT

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF02	D7	EBR	Bus release enable register K11 (K11 and R51 terminal specification) R51	$\overline{\text{BREQ}}$ $\overline{\text{BACK}}$	Input port Output port	0	R/W	
	D6	WT2	Wait control register WT2 WT1 WT0 Number of state 1 1 1 14 1 1 0 12 1 0 1 10 1 0 0 8 0 1 1 6 0 1 0 4 0 0 1 2 0 0 0 No wait			0	R/W	
	D5	WT1						
	D4	WT0						
	D3	CLKCHG	CPU operating clock switch	OSC3	OSC1	0	R/W	
	D2	OSCC	OSC3 oscillation On/Off control	On	Off	0	R/W	
	D1	VDC1	Operating mode selection VDC1 VDC0 Operating mode 1 × High speed (V_{D1}=3.3V) 0 1 Low power (V_{D1}=1.3V) 0 0 Normal (V_{D1}=2.2V)			0	R/W	
	D0	VDC0						

Specifications

CPU clock switching

(1) osc1toosc3: Switching from OSC1 to OSC3

Checks supply voltage and switches system clock from OSC1 (low power mode, V_{D1} = 1.3 V) to OSC3 (normal mode, V_{D1} = 2.2 V).

(2) osc3toosc1: Switching from OSC3 to OSC1

Switches system clock from OSC3 (normal mode, V_{D1} = 2.2 V) to OSC1 (low power mode, V_{D1} = 1.3 V).

Notes

(1) Delay routines for the OSC3 oscillation stabilization waiting time, V_{D1} voltage stabilization waiting time (wait time until OSC3 turns on after operating mode switching, 5 msec or more), etc. are not included in this program example, so it is necessary to create them separately using a hardware timer or software timer. (external call: osc3_delay, vd1_delay)

(2) Switching operating modes when the supply voltage is lower than the V_{D1} setting may cause a malfunction. Hence, perform operating mode switching only after making sure that the power voltage of SVD is more than the V_{D1} setting voltage (absolute value). (external call: svd_auto)
The program example sets the NG flag (vdd_ngf) and terminates processing without switching the system clock, when the supply voltage is lower than the V_{D1} setting.

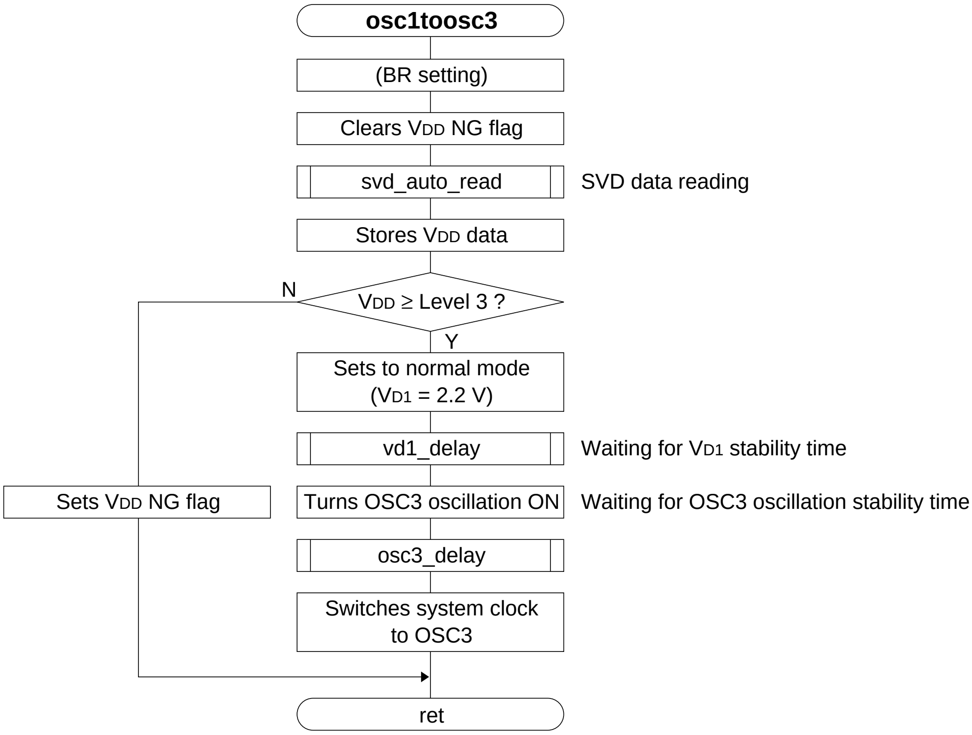
(3) When switching from OSC3 to OSC1 (V_{D1} = 2.2 V → 1.3 V), the program example does not perform special checking of the supply voltage of SVD if the supply voltage is already more than the V_{D1} setting.

(4) Pay special attention the delay routine setting since the OSC3 oscillation stabilization waiting time varies somewhat depending on the oscillator and externally attached parts used.

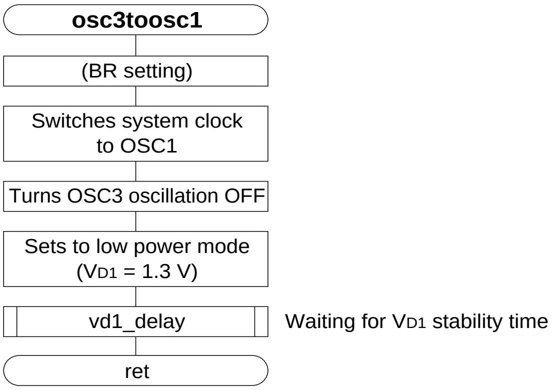
(5) Because of operating voltage considerations, both modes (low power mode and high speed mode) cannot be used in one application.

Flowchart

(1) Switching from OSC1 to OSC3


osc1toosc3
(BR setting)
Clears VDD NG flag
svd_auto_read
SVD data reading
Stores VDD data
VDD ≥ Level 3 ?
N
Y
Sets to normal mode (VD1 = 2.2 V)
vd1_delay
Waiting for VD1 stability time
Sets VDD NG flag
Turns OSC3 oscillation ON
Waiting for OSC3 oscillation stability time
osc3_delay
Switches system clock to OSC3
ret

(2) Switching from OSC3 to OSC1


osc3toosc1
(BR setting)
Switches system clock to OSC1
Turns OSC3 oscillation OFF
Sets to low power mode (VD1 = 1.3 V)
vd1_delay
Waiting for VD1 stability time
ret

Source List

CPU clock switching

```
          external    osc3_delay,vd1_delay
          external    svd_auto_read
          public      osc1toosc3,osc3toosc1
          public      vdd_ngf,vdd_data
;
br_io           equ   0ffh                    ;base reg. address (set i/o area)
mode            equ   00ff02h                 ;mcu//mpu mode control address
;
          data
vdd_ngf:  db    [1]                           ;vdd ng flag
vdd_data: db    [1]                           ;vdd detection data
          code
```

(1) Switching from OSC1 to OSC3

```
;**************************************************************************
;*                                                                        *
;* change osc1(low power mode [vd1=1.3v]) to osc3(normal mode [vd1=2.2v]) *
;*                                                                        *
;**************************************************************************
osc1toosc3:
          ld    br,#br_io                     ;set br reg. address to 0ffxxh
          xor   a,a
          ld    [lod vdd_ngf],a               ;vdd ng flag clear
          ld    [lod vdd_data],a              ;vdd data store
          carl  svd_auto_read                 ;svd check ***
          ld    [lod vdd_data],a              ;vdd store
          cp    a,#03h                        ;areg=svd data
          jrs   c,osc1toosc300                ;vdd >= level 3
;
          and   [br:low mode],#11111100b      ;change mode to normal (vd1 to 2.2v)   (1)
          or    [br:low mode],#00000100b      ;osc3 clock on
          carl  osc3_delay                    ;osc3 start up delay ***
          or    [br:low mode],#00001000b      ;change system clock to osc3
          jrs   osc1toosc301
;
osc1toosc300:
          ld    a,#0ffh
          ld    [lod vdd_ngf],a               ;vdd ng flag set
osc1toosc301:
          ret
;
```

(2) Switching from OSC3 to OSC1

```
;**************************************************************************
;*                                                                        *
;* change osc3(normal mode [vd1=2.2v]) to osc1(low power mode [vd1=1.3v]) *
;*                                                                        *
;**************************************************************************
osc3toosc1:
          ld    br,#br_io                     ;set br reg. address to 0ffxxh
          and   [br:low mode],#11110111b      ;change system clock to osc1
          and   [br:low mode],#11111011b      ;osc3 clock off                        (2)
          or    [br:low mode],#00000001b      ;change mode to low power (vd1 to 1.3v)
          carl  vd1_delay                     ;vd1 delay ***
          ret
;
          end
```

5 INPUT PORTS (K PORTS)

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF50	D7	SIK07	K07 interrupt selection register	Interrupt enable	Interrupt disable	0	R/W	
	D6	SIK06	K06 interrupt selection register					
	D5	SIK05	K05 interrupt selection register					
	D4	SIK04	K04 interrupt selection register					
	D3	SIK03	K03 interrupt selection register					
	D2	SIK02	K02 interrupt selection register					
	D1	SIK01	K01 interrupt selection register					
	D0	SIK00	K00 interrupt selection register					
00FF51	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	–	–	–	–	–		
	D2	–	–	–	–	–		
	D1	SIK11	K11 interrupt selection register	Interrupt enable	Interrupt disable	0	R/W	
	D0	SIK10	K10 interrupt selection register					
00FF52	D7	KCP07	K07 interrupt comparison register	Interrupt generated at falling edge	Interrupt generated at rising edge	1	R/W	
	D6	KCP06	K06 interrupt comparison register					
	D5	KCP05	K05 interrupt comparison register					
	D4	KCP04	K04 interrupt comparison register					
	D3	KCP03	K03 interrupt comparison register					
	D2	KCP02	K02 interrupt comparison register					
	D1	KCP01	K01 interrupt comparison register					
	D0	KCP00	K00 interrupt comparison register					
00FF53	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	–	–	–	–	–		
	D2	–	–	–	–	–		
	D1	KCP11	K11 interrupt comparison register	Falling edge	Rising edge	1	R/W	
	D0	KCP10	K10 interrupt comparison register					
00FF54	D7	K07D	K07 input port data	High level input	Low level input	–	R	
	D6	K06D	K06 input port data					
	D5	K05D	K05 input port data					
	D4	K04D	K04 input port data					
	D3	K03D	K03 input port data					
	D2	K02D	K02 input port data					
	D1	K01D	K01 input port data					
	D0	K00D	K00 input port data					
00FF55	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	_		
	D4	–	–	–	–	_		
	D3	–	–	–	–	_		
	D2	–	–	–	–	_		
	D1	K11D	K11 input port data	High level input	Low level input	–	R	
	D0	K10D	K10 input port data					

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF20	D7	PK01	K00–K07 interrupt priority register	PK01 PK00 PSIF1 PSIF0 PSW1 PSW0 Priority PTM1 PTM0 level 1 1 Level 3 1 0 Level 2 0 1 Level 1 0 0 Level 0		0	R/W	
	D6	PK00						
	D5	PSIF1	Serial interface interrupt priority register			0	R/W	
	D4	PSIF0						
	D3	PSW1	Stopwatch timer interrupt priority register			0	R/W	
	D2	PSW0						
	D1	PTM1	Clock timer interrupt priority register			0	R/W	
	D0	PTM0						
00FF21	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	PPT1	Programmable timer interrupt priority register	PPT1 PPT0 Priority PK11 PK10 level 1 1 Level 3 1 0 Level 2 0 1 Level 1 0 0 Level 0		0	R/W	
	D2	PPT0						
	D1	PK11	K10 and K11 interrupt priority register			0	R/W	
	D0	PK10						
00FF23	D7	EPT1	Programmable timer 1 interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D6	EPT0	Programmable timer 0 interrupt enable register					
	D5	EK1	K10 and K11 interrupt enable register					
	D4	EK0H	K04–K07 interrupt enable register					
	D3	EK0L	K00–K03 interrupt enable register					
	D2	ESERR	Serial I/F (error) interrupt enable register					
	D1	ESREC	Serial I/F (receiving) interrupt enable register					
	D0	ESTRA	Serial I/F (transmitting) interrupt enable register					
00FF25	D7	FPT1	Programmable timer 1 interrupt factor flag	(R) Interrupt factor is generated (W) Reset	(R) No interrupt factor is generated (W) No operation	0	R/W	
	D6	FPT0	Programmable timer 0 interrupt factor flag					
	D5	FK1	K10 and K11 interrupt factor flag					
	D4	FK0H	K04–K07 interrupt factor flag					
	D3	FK0L	K00–K03 interrupt factor flag					
	D2	FSERR	Serial I/F (error) interrupt factor flag					
	D1	FSREC	Serial I/F (receiving) interrupt factor flag					
	D0	FSTRA	Serial I/F (transmitting) interrupt factor flag					

Specifications

Control of input port (K port)

Vector address setting for input port (K port) interrupt

(1) input_normal: Data reading from normal input port (K port)

(2) input_keyscan:Key scan for 8 x 2 key matrix

Assumes the key matrix has been configured with input and output as shown in Figure 5.1, and specifies the key pressed and then stores the data into the RAM area named key_data.

<Conditions>

K07–K00 ports: Input with pull-up resistor (mask option setting)
R01, R00 ports: Nch open drain output (mask option setting)

key_data: 1 word

	7	6	5	4	3	2	1	0	
key_data +0	K07	K06	K05	K04	K03	K02	K01	K00	(R00)
+1	K07	K06	K05	K04	K03	K02	K01	K00	(R01)

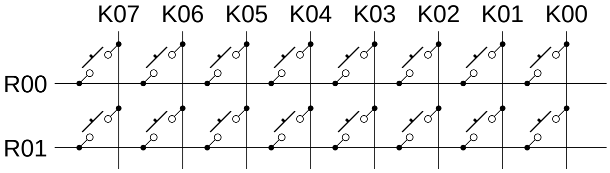


Fig. 5.1 Key matrix

(3) input_keywait, input_keyintr: Interrupt condition setting and interrupt processing for input port (K port)

Generates an $\overline{\text{IRQ3}}$ interrupt when changing the input port K10 and K11 from HIGH to LOW.

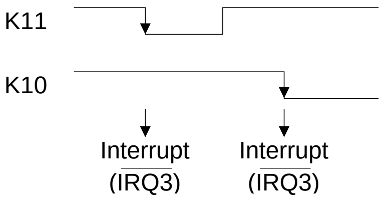
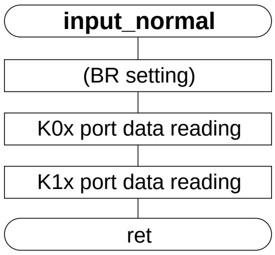

K11
K10
Interrupt ($\overline{\text{IRQ3}}$)
Interrupt ($\overline{\text{IRQ3}}$)

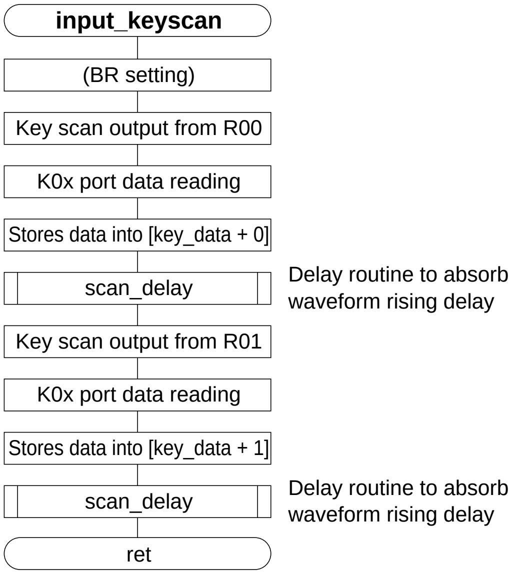
Fig. 5.2 Interrupt generation timing

Flowchart

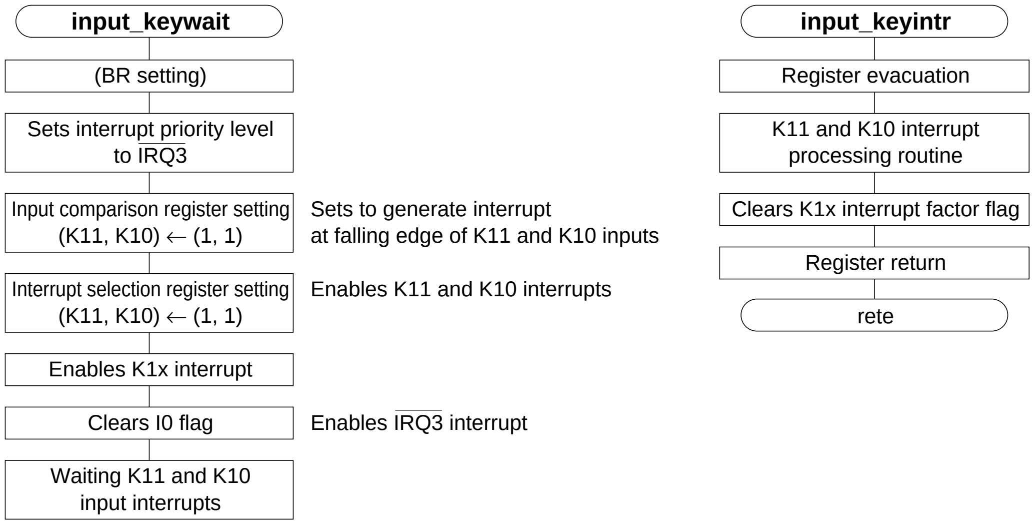
(1) Data reading from normal input port (K port)



(2) Key scanning on 8 x 2 key matrix


input_keyscan
(BR setting)
Key scan output from R00
K0x port data reading
Stores data into [key_data + 0]
scan_delay
Delay routine to absorb waveform rising delay
Key scan output from R01
K0x port data reading
Stores data into [key_data + 1]
scan_delay
Delay routine to absorb waveform rising delay
ret

(3) Interrupt condition setting and interrupt processing for input port (K port)


input_keywait
(BR setting)
Sets interrupt priority level to $\overline{\text{IRQ3}}$
Input comparison register setting (K11, K10) ← (1, 1)
Sets to generate interrupt at falling edge of K11 and K10 inputs
Interrupt selection register setting (K11, K10) ← (1, 1)
Enables K11 and K10 interrupts
Enables K1x interrupt
Clears I0 flag
Enables $\overline{\text{IRQ3}}$ interrupt
Waiting K11 and K10 input interrupts
input_keyintr
Register evacuation
K11 and K10 interrupt processing routine
Clears K1x interrupt factor flag
Register return
rete

Notes

(1) When the pull-up resistor option has been set to "with resistor", a delay in the waveform rise time will occur depending on the time constant of the input gate capacitance when changing the input terminal from LOW to HIGH. For this reason, set an appropriate wait time (for reference, approximately 500 μsec) for the introduction of the input port. In particular, special attention should be paid to key scanning for key matrix formation.

(2) Note that the K11 terminal cannot be used as an input port when the K11 terminal has been set for input of the bus release request ($\overline{\text{BREQ}}$) signal.
See Part I in this manual, "E0C88112/88104 Technical Hardware", for details of the bus release sequence.

(3) The K10 terminal doubles as the input terminal of the programmable timer/event counter with input port functions sharing the input signal as it is. For this reason, when the K10 terminal has been set to the input terminal of the programmable timer/event counter, pay attention to interrupt setting.
See "12 PROGRAMMABLE TIMER", for the control of the programmable timer/event counter.

(4) To reset the interrupt factor flag, write "1" into the corresponding flags alone, using the AND or LD instruction. When the OR logic operation instruction has been used, "1" is written for the interrupt factor flags that have been set to "1" within the same address and those flags are then clear.

(5) The interrupt flags (I1 and I0) have not been reset in the interrupt processing routine of this program example, so an interrupt lower than $\overline{\text{IRQ3}}$ level is disabled at the time of generation. When you wish to accept the next interrupt after an interrupt has been generated, re-setting of the interrupt flags or resetting the interrupt factor flag is necessary after due consideration for the nesting level.

(6) A noise reject circuit is not included in the input port (K port).
In particular, when input port data is read using an interrupt, the interrupt may generate one of the another by key chattering. For this reason, some measure must be devised such as adding noise reject processing in software or with an external.

(7) In this program example for input port (K port), the vector address setting and program have been allocated from 003000H for the sake of convenience.

Source List

Control of input port (K port)

```
        external     scan_delay
        public       input_normal,input_keyscan,input_keywait,input_keyintr
        public       key_data
;
k1x_vector     equ   00000ah                  ;k1x interrupt vector address offset
keyinput       equ   003000h                  ;program start address offset
br_io          equ   0ffh                     ;base reg. address (set i/o area)
sik1           equ   00ff51h                  ;interrupt selection reg. for k1x
kcp1           equ   00ff53h                  ;interrupt comparison reg. for k1x
k0d            equ   00ff54h                  ;input port data from k0x
k1d            equ   00ff55h                  ;input port data from k1x
r0d            equ   00ff73h                  ;r0x output data
intr_pr1       equ   00ff21h                  ;interrupt priority reg. 0
intr_en1       equ   00ff23h                  ;interrupt enable reg. 0
intr_fac1      equ   00ff25h                  ;interrupt factor flag reg. 0
;
        data
key_data:      dw    [1]
        code
```

Source List

Vector address setting for input port (K port) interrupt

```
intr_vectors:
;
        org    intr_vectors+k1x_vector
        dw     input_keyintr                  ;k1x interrupt processing routine
```

(1) Data reading from normal input port (K port)

```
        org    intr_vectors+keyinput
;*************************************************************************
;*                                                                       *
;*      k(input) port read (normal)                                      *
;*         a <- k0x(complementary)                                       *
;*         b <- k1x(complementary)                                       *
;*                                                                       *
;*************************************************************************
input_normal:
        ld     br,#br_io                      ;set br reg. address to 0ffxxh
        ld     a,[br:low k0d]                 ;k07-00 port read                      (1)
        ld     b,[br:low k1d]                 ;k11-00 port read
        ret
;
```

(2) Key scanning on 8 x 2 key matrix

```
;*************************************************************************
;*                                                                       *
;*      k(input) port read (key scan)                                    *
;*         k07 k06 k05 k04 k03 k02 k01 k00(pull up)                      *
;*         r00(n-ch. o.d)                                                *
;*         r01(n-ch. o.d)                                                *
;*                                                                       *
;*         key_data+0(r10) <- k07 k06 k05 k04 k03 k02 k01 k00           *
;*         key_data+1(r11) <- r07 k06 k05 k04 k03 k02 k01 k00           *
;*                                                                       *
;*************************************************************************
input_keyscan:
        ld     br,#br_io                      ;set br reg. address to 0ffxxh
        and    [br:low r0d],#11111110b        ;r00 key scan output
        ld     a,[br:low k0d]                 ;k0x port read
        ld     [lod key_data+0],a             ;key_data save
        carl   scan_delay                     ;key scan delay ***                    (2)
        and    [br:low r0d],#11111101b        ;r01 key scan output
        ld     a,[br:low k0d]                 ;k0x port read
        ld     [lod key_data+1],a             ;key_data save
        carl   scan_delay                     ;key scan delay ***
        ret
```

(3) Interrupt condition setting and interrupt processing for input port (K port)

```
;*************************************************************************
;*                                                                       *
;*      k(input) port read (interrupt)                                   *
;*         k11,10 <- /irq3 falling edge ("h" - "l") interrupt            *
;*                                                                       *
;*************************************************************************
input_keywait:
        ld     br,#br_io                      ;set br reg. address to 0ffxxh
        or     [br:low intr_pr1],#00000011b   ;set pk11 and 10 to /irq3
        ld     [br:low kcp1],#00000011b       ;k11,10 falling edge (h -> l)          (3)
        ld     [br:low sik1],#00000011b       ;k11,10 interrupt enable
        or     [br:low intr_en1],#00100000b   ;ek1 (k11,10) interrupt en.
```

Source List

```
         ld    [br:low r0d],#00000000b        ;waiting key on r0d scan low output
         ld    a,sc
         and   a,#00111111b
         or    a,#10000000b
         ld    sc,a                           ;i0 flag clear (en. /irq3 intr.)
;waiting k11,10 interrupt
;
;*************************************************************************
;*                                                                       *
;*       k1x interrupt processing routine                                *
;*                                                                       *
;*************************************************************************
input_keyintr:
         push  ale
;
;        k11 and 10 interrupt processing routine
;
         and   [br:low intr_fac1],#00100000b  ;clear fk1 (k11,10) flag
         pop   ale
         rete
;
         end
```

(3)

6 OUTPUT PORTS (R PORTS)

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF70	D7	HZR51	R51 high impedance control	High impedance	Complementary	0	R/W	
	D6	HZR50	R50 high impedance control					
	D5	HZR4H	R/W register	1	0	0	R/W	Reserved register
	D4	HZR4L	R/W register					
	D3	HZR1H	R14–R17 high impedance control	High impedance	Complementary	0	R/W	
	D2	HZR1L	R10–R13 high impedance control					
	D1	HZR0H	R04–R07 high impedance control					
	D0	HZR0L	R00–R03 high impedance control					
00FF71	D7	HZR27	R27 high impedance control	High impedance	Complementary	0	R/W	
	D6	HZR26	R26 high impedance control					
	D5	HZR25	R25 high impedance control					
	D4	HZR24	R24 high impedance control					
	D3	HZR23	R23 high impedance control					
	D2	HZR22	R22 high impedance control					
	D1	HZR21	R21 high impedance control					
	D0	HZR20	R20 high impedance control					
00FF72	D7	HZR37	R37 high impedance control	High impedance	Complementary	0	R/W	
	D6	HZR36	R36 high impedance control					
	D5	HZR35	R35 high impedance control					
	D4	HZR34	R34 high impedance control					
	D3	HZR33	R33 high impedance control					
	D2	HZR32	R32 high impedance control					
	D1	HZR31	R31 high impedance control					
	D0	HZR30	R30 high impedance control					
00FF73	D7	R07D	R07 output port data	High	Low	1	R/W	
	D6	R06D	R06 output port data					
	D5	R05D	R05 output port data					
	D4	R04D	R04 output port data					
	D3	R03D	R03 output port data					
	D2	R02D	R02 output port data					
	D1	R01D	R01 output port data					
	D0	R00D	R00 output port data					
00FF74	D7	R17D	R17 output port data	High	Low	1	R/W	
	D6	R16D	R16 output port data					
	D5	R15D	R15 output port data					
	D4	R14D	R14 output port data					
	D3	R13D	R13 output port data					
	D2	R12D	R12 output port data					
	D1	R11D	R11 output port data					
	D0	R10D	R10 output port data					
00FF75	D7	R27D	R27 output port data	High	Low	1	R/W	
	D6	R26D	R26 output port data					
	D5	R25D	R25 output port data					
	D4	R24D	R24 output port data					
	D3	R23D	R23 output port data					
	D2	R22D	R22 output port data					
	D1	R21D	R21 output port data					
	D0	R20D	R20 output port data					

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF76	D7	R37D	R37 output port data	High	Low	1	R/W	
	D6	R36D	R36 output port data					
	D5	R35D	R35 output port data					
	D4	R34D	R34 output port data					
	D3	R33D	R33 output port data					
	D2	R32D	R32 output port data					
	D1	R31D	R31 output port data					
	D0	R30D	R30 output port data					
00FF77	D7	R47D	R/W register	1	0	1	R/W	Reserved register
	D6	R46D	R/W register					
	D5	R45D	R/W register					
	D4	R44D	R/W register					
	D3	R43D	R/W register					
	D2	R42D	R/W register					
	D1	R41D	R/W register					
	D0	R40D	R/W register					
00FF78	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	–	–	–	–	–		
	D2	–	–	–	–	–		
	D1	R51D	R51 output port data	High	Low	1	R/W	
	D0	R50D	R50 output port data	High	Low	0	R/W	
00FF30	D7	–	–	–	–	–		Constantry "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	MODE16	8/16-bit mode selection	16-bit x 1	8-bit x 2	0	R/W	
	D3	CHSEL	TOUT output channel selection	Timer 1	Timer 0	0	R/W	
	D2	PTOUT	TOUT output control	On	Off	0	R/W	
	D1	CKSEL1	Prescaler 1 source clock selection	f_{OSC3}	f_{OSC1}	0	R/W	
	D0	CKSEL0	Prescaler 0 source clock selection	f_{OSC3}	f_{OSC1}	0	R/W	
00FF44	D7	–	–	–	–	–		Constantry "0" when being read
	D6	BZSTP	One-shot buzzer forcibly stop	Forcibly stop	No operation	–	W	
	D5	BZSHT	One-shot buzzer trigger/status R	Busy	Ready	0	R/W	
			W	Trigger	No operation			
	D4	SHTPW	One-shot buzzer duration width selection	125 msec	31.25 msec	0	R/W	
	D3	ENRTM	Envelope attenuation time	1 sec	0.5 sec	0	R/W	
	D2	ENRST	Envelope reset	Reset	No operation	–	W	"0" when being read
	D1	ENON	Envelope On/Off control	On	Off	0	R/W	*1
	D0	BZON	Buzzer output control	On	Off	0	R/W	

*1 Reset to "0" during one-shot output.

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF40	D7	–	–	–	–	–		"0" when being read
	D6	FOUT2	FOUT frequency selection FOUT2 FOUT1 FOUT0 Frequency 0 0 0 f_{OSC1} / 1 0 0 1 f_{OSC1} / 2 0 1 0 f_{OSC1} / 4 0 1 1 f_{OSC1} / 8 1 0 0 f_{OSC3} / 1 1 0 1 f_{OSC3} / 2 1 1 0 f_{OSC3} / 4 1 1 1 f_{OSC3} / 8			0	R/W	
	D5	FOUT1				0	R/W	
	D4	FOUT0				0	R/W	
	D3	FOUTON	FOUT output control	On	Off	0	R/W	
	D2	WDRST	Watchdog timer reset	Reset	No operation	–	W	Constantly "0" when being read
	D1	TMRST	Clock timer reset	Reset	No operation	–	W	
	D0	TMRUN	Clock timer Run/Stop control	Run	Stop	0	R/W	

Specifications

Control of output port (R port)

(1) initoutput_normal, output_normal: Normal DC output

Sets the R3x port to complementary output and outputs HIGH and LOW to R35–R37.

(2) init_hiz, output_hiz: High impedance output control

First sets the R5x port to complementary output and then switches between high impedance output and complementary output to operate the high impedance control register.

(3) fout_init, fout_control: FOUT output control

Controls the turning ON/OFF of the FOUT output.

Notes

(1) Besides normal DC output, output port terminals are shared with the special output shown in Table 6.1, and which is used can be selected in software. When using special output, it should be noted so that the port cannot be used as output port.

For control of special output except for FOUT output (R34 terminal), see the following chapters:

- TOUT output (R27) .... "12 PROGRAMMABLE TIMER"
- BZ output (R50) .... "13 SOUND GENERATOR"

Table 6.1 Special output

Output port	Special output
R27	TOUT output
R34	FOUT output
R50	BZ output

(2) Please note that in accordance with the bus mode and system controller settings or when using bus release for DMA transfer, the following output port terminals are used for the address bus, $\overline{RD}$/$\overline{WR}$ signals, $\overline{CE3}$–$\overline{CE0}$ signals and $\overline{BACK}$ outputs and cannot be used as an output port.

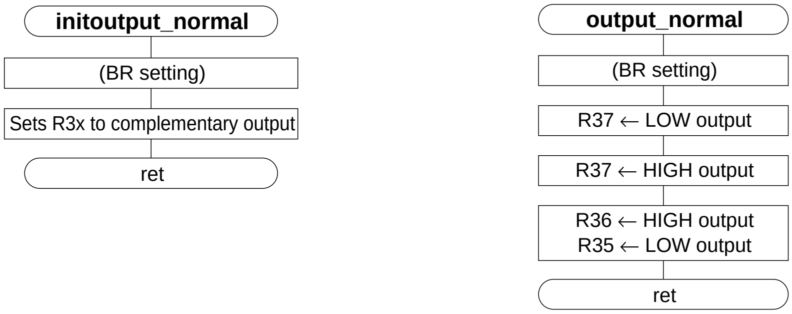
Notes

Table 6.2
Combined output terminal

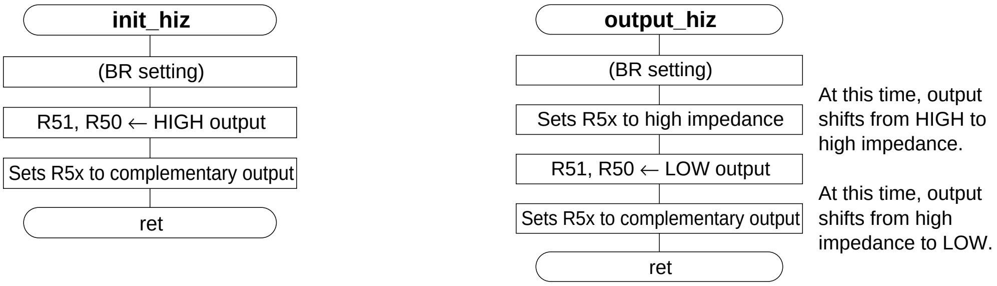
Output port	Special output
R00–R07	A0–A7
R10–R17	A8–A15
R20–R22	A16–A18
R23	$\overline{\text{RD}}$ signal
R24	$\overline{\text{WR}}$ signal
R30–R33	$\overline{\text{CE0}}$–$\overline{\text{CE3}}$ signals
R51	$\overline{\text{BACK}}$ signal

Flowchart

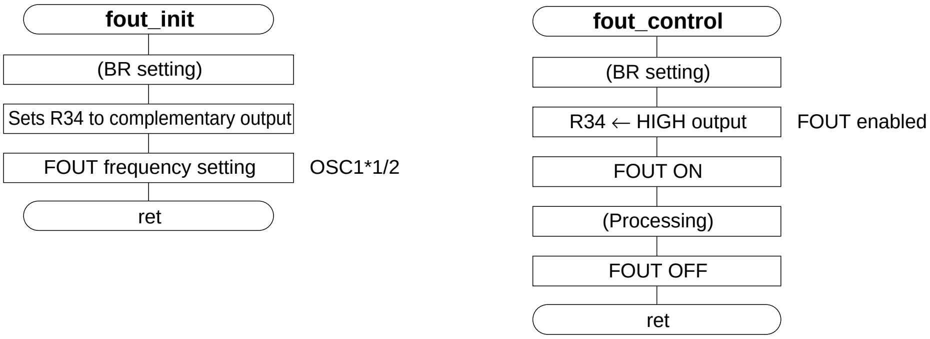
(1) Normal DC output



(2) High impedance output control


init_hiz
(BR setting)
R51, R50 ← HIGH output
Sets R5x to complementary output
ret
output_hiz
(BR setting)
Sets R5x to high impedance
R51, R50 ← LOW output
Sets R5x to complementary output
ret
At this time, output shifts from HIGH to high impedance.
At this time, output shifts from high impedance to LOW.

(3) FOUT output control



Source List

Control of output port (R port)

```
         public      initoutput_normal,output_normal
         public      init_hiz,output_hiz
         public      fout_init,fout_control
;
br_io          equ   0ffh                     ;base reg. address (set i/o area)
hzr_ex         equ   00ff70h                  ;expand output control reg.
hzr3           equ   00ff72h                  ;r3x output control reg.
r3d            equ   00ff76h                  ;r3x output data
r5d            equ   00ff78h                  ;r5x output data
rtm_mode       equ   00ff40h                  ;timer mode set reg.
         code
```

(1) Normal DC output

```
;************************************************************************
;*                                                                      *
;*       r(output) port control (normal)                                *
;*          r37    <- "l" then "h        (complementary)                *
;*          r36,35 <- "h","l"            (complementary)                *
;*                                                                      *
;************************************************************************
;*** initialize routine
initoutput_normal:
         ld    br,#br_io                      ;set br reg. address to 0ffxxh
         ld    [br:low hzr3],#00000000b       ;set r3x complementary output
         ret
;************************************************************************
;*** control routine
output_normal:
         ld    br,#br_io                      ;set br reg. address to 0ffxxh
         and   [br:low r3d],#01111111b        ;r37 <- "l" output
         or    [br:low r3d],#10000000b        ;r37 <- "h" output
         ld    a,[br:low r3d]                 ;r3x output port read
         and   a,#10011111b
         or    a,#01000000b
         ld    [br:low r3d],a                 ;r36 <- "h" and r35 <- "l" output
         ret
```

(1)

(2) High impedance output control

```
;************************************************************************
;*                                                                      *
;*       r(output) port control (hi-z)                                  *
;*          r50,51 <- "h","h"            (complementary at init.)       *
;*                                                                      *
;*                 <- "hi-z"                                            *
;*                 <- "l","l"            (complementary)                *
;*                                                                      *
;************************************************************************
;*** initialize rotine
init_hiz:
         ld    br,#br_io                      ;set br reg. address to 0ffxxh
         or    [br:low r5d],#00000011b        ;r51,50 <- "h"
         and   [br:low hzr_ex],#00111111b     ;r5x <- complementary output
         ret

;************************************************************************
;*** control routine
output_hiz:
         ld    br,#br_io                      ;set br reg. address to 0ffxxh
         or    [br:low hzr_ex],#11000000b     ;r5x <- high impedance ("hi-z")
         and   [br:low r5d],#11111100b        ;r51,50 <- "l" output
         and   [br:low hzr_ex],#00111111b     ;r5x <- complementary output
         ret
;
```

(2)

Source List

(3) FOUT output control

```
;**************************************************************************
;*                                                                        *
;*       fout control                                                     *
;*                                                                        *
;**************************************************************************
;*** initialize rotine
fout_init:
         ld    br,#br_io                       ;set br reg. address to 0ffxxh
         and   [br:low hzr3],#11101111b        ;set r34 complementary output
         ld    a,[br:low rtm_mode]
         and   a,#00000111b
         or    a,#00010000b                    ;set fout=fosc1/2
         ld    [br:low rtm_mode],a
         ret
;**************************************************************************
;*** control routine                                                            (3)
fout_control:
         or    [br:low r3d],#00010000b         ;r34="h" (fout enable)
         or    [br:low rtm_mode],#00001000b    ;fout on
;
;
;other processing
;
;
         and   [br:low rtm_mode],#11110111b    ;fout off
         ret
;
         end
```

7 I/O PORTS (P PORTS)

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF60	D7	IOC07	P07 I/O control register	Output	Input	0	R/W	
	D6	IOC06	P06 I/O control register					
	D5	IOC05	P05 I/O control register					
	D4	IOC04	P04 I/O control register					
	D3	IOC03	P03 I/O control register					
	D2	IOC02	P02 I/O control register					
	D1	IOC01	P01 I/O control register					
	D0	IOC00	P00 I/O control register					
00FF61	D7	IOC17	P17 I/O control register	Output	Input	0	R/W	
	D6	IOC16	P16 I/O control register					
	D5	IOC15	P15 I/O control register					
	D4	IOC14	P14 I/O control register					
	D3	IOC13	P13 I/O control register					
	D2	IOC12	P12 I/O control register					
	D1	IOC11	P11 I/O control register					
	D0	IOC10	P10 I/O control register					
00FF62	D7	P07D	P07 I/O port data	High	Low	1	R/W	
	D6	P06D	P06 I/O port data					
	D5	P05D	P05 I/O port data					
	D4	P04D	P04 I/O port data					
	D3	P03D	P03 I/O port data					
	D2	P02D	P02 I/O port data					
	D1	P01D	P01 I/O port data					
	D0	P00D	P00 I/O port data					
00FF63	D7	P17D	P17 I/O port data	High	Low	1	R/W	
	D6	P16D	P16 I/O port data					
	D5	P15D	P15 I/O port data					
	D4	P14D	P14 I/O port data					
	D3	P13D	P13 I/O port data					
	D2	P12D	P12 I/O port data					
	D1	P11D	P11 I/O port data					
	D0	P10D	P10 I/O port data					

Specifications

Control of I/O port (P port)

(1) initio_normal, io_normal: Normal data input/output of I/O port

Sets P0x port as input and P1x port as output, and then waits for a HIGH input to P07 port. When P07 shifts to HIGH, reads P0x input data and outputs 55H to P1x.

(2) init_switch, io_switch: Scan for 2 x 2 switch matrix

Assumes the switch matrix has been configured with input and output as shown in Figure 7.1, and specifies the key pressed and then stores the data into the RAM area named switch_data.

<*Conditions*>

P10, P11 ports:	Input with pull-up resistor
R51, R50 ports:	Nch open drain output (software setting)
switch_data:	1 byte

	7	6	5	4	3	2	1	0
switch_data	0	0	0	0	P11/R51	P10/R51	P11/R50	P10/R50

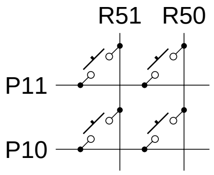
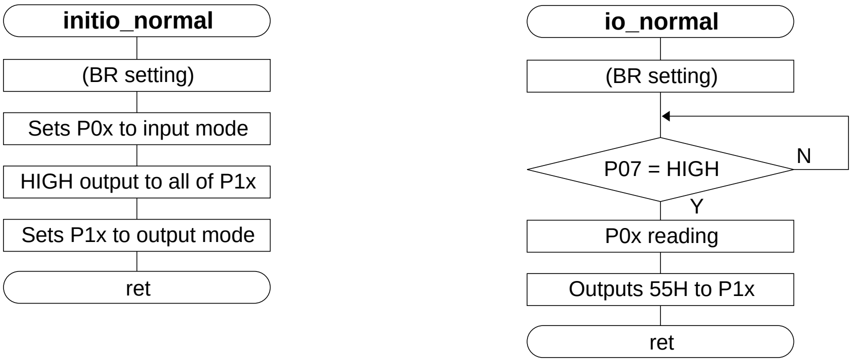


Fig. 7.1 Switch matrix

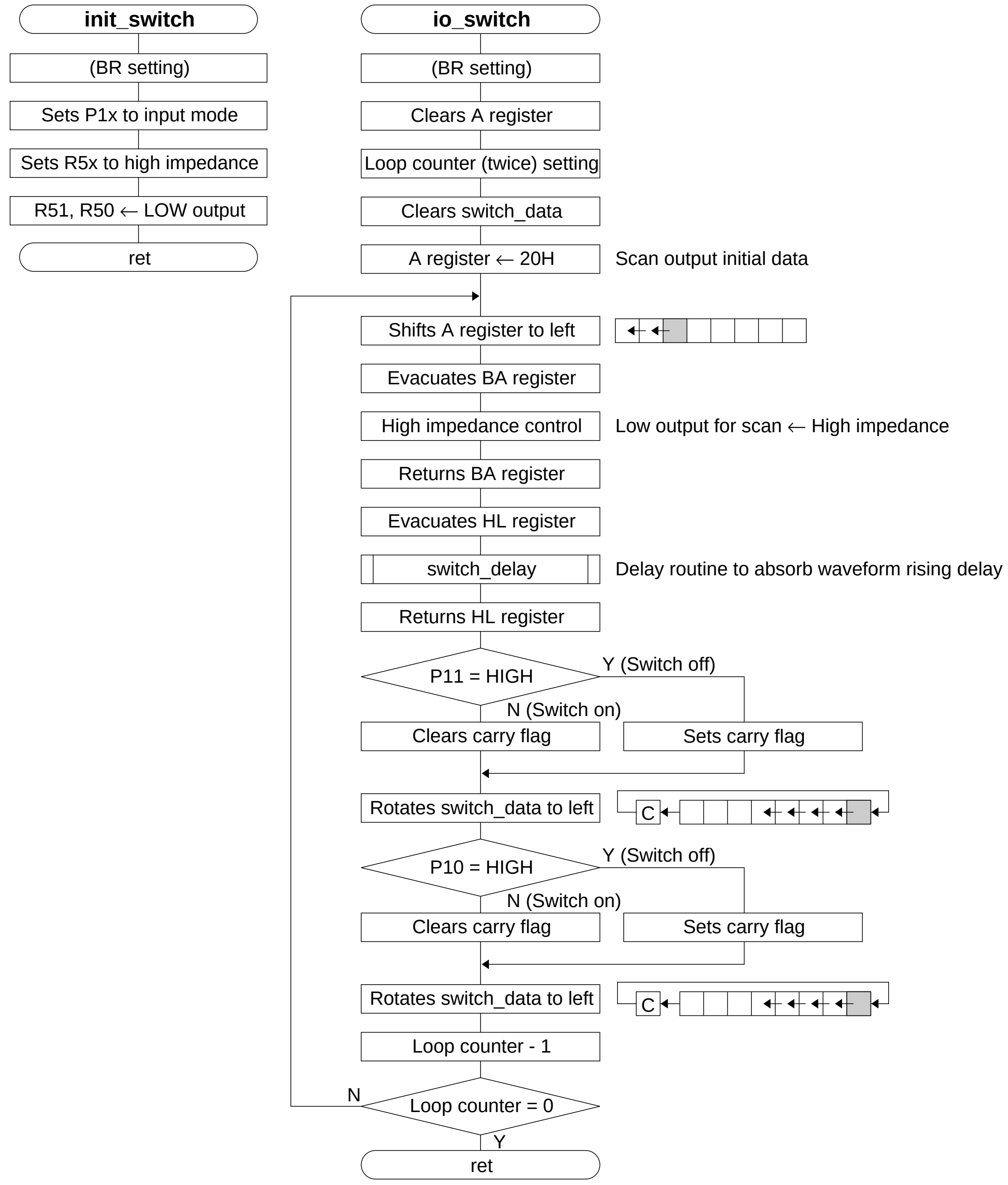
Flowchart

(1) Normal data input/output of I/O port



Flowchart

(2) Scan for 2 x 2 switch matrix



Notes

(1) In the input mode, when changing the port terminal from LOW to HIGH with a pull-up resistor, a delay in the waveform rise time will occur depending on the time constant of the pull-up resistor and input gate capacity. Hence, when reading data from the input port, set an appropriate wait time (for reference, approximately 500 μsec). Care is particularly required in key scanning for key matrix configuration. (external call: switch_delay)

(2) Besides normal DC output, I/O port terminals are shared with the special output shown in Table 7.1, and which is used can be selected in software. When using special output, it should be noted so that the port cannot be used as I/O port.

Table 7.1 Special input/output

I/O port	Special output
P10	SIN
P11	SOUT
P12	$\overline{\text{SCLK}}$
P13	$\overline{\text{SRDY}}$
P14	CMPP0
P15	CMPM0
P16	CMPP1
P17	CMPM1

For details of each control procedure, see the following chapters:

- SIN (P10), SOUT (P11), $\overline{\text{SCLK}}$ (P12), $\overline{\text{SRDY}}$ (P13)
  .... "8 SERIAL INTERFACE 1 (CLOCK SYNCHRONOUS SYSTEM)"
  .... "9 SERIAL INTERFACE 2 (ASYNCHRONOUS SYSTEM)"
- CMPP0 (P14), CMPM0 (P14), CMPP1 (P16), CMPM1 (P17)
  .... "14 ANALOG COMPARATOR"

(3) Please note that in accordance with the bus mode and system controller settings, P0x terminals are used for the data bus and cannot be used as an I/O port.

Table 7.2 Combined data bus terminal

I/O port	Special output
P00–P07	D0–D7

Source List

Control of I/O port (P port)

```
         external    switch_delay
         public      initio_normal,io_normal
         public      init_switch,io_switch
         public      switch_data
;
br_io          equ   0ffh                      ;base reg. address (set i/o area)
ioc0           equ   00ff60h                   ;p0x i/o control reg.
ioc1           equ   00ff61h                   ;p1x i/o control reg.
p0d            equ   00ff62h                   ;p0x port data
p1d            equ   00ff63h                   ;p1x port data
hzr_ex         equ   00ff70h                   ;expand output control reg.
r5d            equ   00ff78h                   ;r5x output data
;
         data
switch_data:   db    [1]
;
         code
```

(1) Normal data input/output of I/O port

```
;*************************************************************************
;*                                                                       *
;*       p(i/o) port control (normal)                                    *
;*          p0x (input:gate direct)                                      *
;*          p1x (output) <- p17-10 (all "h")                             *
;*                                                                       *
;*          p07 (input)  <- waits  "l" to "h" then p0x data read         *
;*          p0x (input)  <- p0x port data read                           *
;*                                                                       *
;*************************************************************************
;*** initialize routine
initio_normal:
         ld    br,#br_io                       ;set br reg. address to 0ffxxh
         ld    [br:low ioc0],#00000000b        ;set ioc0 (p07-00=input)
         ld    [br:low p1d],#11111111b         ;p17-10(output) <- "h"
         ld    [br:low ioc1],#11111111b        ;set ioc1 (p17-10=output)
         ret
;*************************************************************************
;*** control routine
io_normal:
         ld    br,#br_io                       ;set br reg. address to 0ffxxh
io_normal00:
         bit   [br:low p0d],#10000000b         ;p07 = "h" ?
         jrs   z,io_normal00
;
         ld    a,[br:low p0d]                  ;p0x input
         ld    [br:low p1d],#01010101b         ;p1x output
;
         ret
```

(1)

Source List

(2) Scan for 2 x 2 switch matrix

```
;*************************************************************************
;*                                                                       *
;*       p(i/o) port control (internal pull up delay)                    *
;*          r51  r50(n-channel open drain)                               *
;*          p11(pull up)                                                 *
;*          p10(pull up)                                                 *
;*                       7  6  5  4    3       2       1       0         *
;*          switch data  0  0  0  0 r51/p11 r51/p10 r50/p11 r50/p10      *
;*                                                                       *
;*************************************************************************
;*** initialize routine
init_switch:
         ld    br,#br_io                       ;set br reg. address to 0ffxxh
         ld    [br:low ioc1],#00000000b        ;set ioc1 (p17-10=input)
         or    [br:low hzr_ex],#11000000b      ;r5x <- high impledance "hi-z"
         and   [br:low r5d],#11111100b         ;r5x <- "l"
         ret
;*************************************************************************
;*** control routine
io_switch:
         ld    br,#br_io                       ;set br reg. address to 0ffxxh
         xor   a,a
         ld    b,#2                            ;switch scan loop counter
         ld    hl,#lod switch_data             ;switch data buffer
         ld    [hl],#0                         ;clear switch data buffer
         ld    a,#00100000b                    ;scan init. data set
switch00:
         sll   a                               ;scan data move bit0 to 7
         push  ba                              ;escape scan data
         ld    b,[br:low hzr_ex]               ;r5x hi-z control ("hi-z" <-> "l")
         and   b,#00111111b
         or    a,b
         ld    [br:low hzr_ex],a               ;r5x scan data control with hi-z
         pop   ba
         push  hl
         carl  switch_delay                    ;switch scan delay ***
         pop   hl
         bit   [br:low p1d],#00000010b         ;compare p11 port level
         jrs   nz,switch01
;switch (p11) on "l"
         and   sc,#11111101b                   ;clear carry flag
         jrs   switch02
;switch (p11) off "h"
switch01:
         or    sc,#00000010b                   ;set carry flag
switch02:
         rl    [hl]                            ;set switch data buffer
         bit   [br:low p1d],#00000001b         ;compare p10 port level
         jrs   nz,switch03
;switch (p10) on "l"
         and   sc,#11111101b                   ;clear carry flag
         jrs   switch04
;switch (p10) off "h"
switch03:
         or    sc,#00000010b                   ;set carry flag
switch04:
         rl    [hl]                            ;set switch data buffer
         djr   nz,switch00
;
         ret
;
         end
```

(2)

8 SERIAL INTERFACE 1 (CLOCK SYNCHRONOUS INTERFACE)

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF48	D7	–	–	–	–	–		"0" when being read
	D6	EPR	Parity enable register	With parity	Non parity	0	R/W	Only for
	D5	PMD	Parity mode selection	Odd	Even	0	R/W	asynchronous mode
	D4	SCS1	Clock source selection SCS1 SCS0 Clock source 1 1 Programmable timer 1 0 f_{OSC3} / 4 0 1 f_{OSC3} / 8 0 0 f_{OSC3} / 16			0	R/W	In the clock synchronous slave mode, external clock is selected.
	D3	SCS0				0	R/W	
	D2	SMD1	Serial I/F mode selection SMD1 SMD0 Mode 1 1 Asynchronous 8-bit 1 0 Asynchronous 7-bit 0 1 Clock synchronous slave 0 0 Clock synchronous master			0	R/W	
	D1	SMD0				0	R/W	
	D0	ESIF	Serial I/F enable register	Serial I/F	I/O port	0	R/W	
00FF49	D7	–	–	–	–	–		"0" when being read
	D6	FER	Framing error flag R	Error	No error	0	R/W	Only for
			W	Reset (0)	No operation			asynchronous mode
	D5	PER	Parity error flag R	Error	No error	0	R/W	
			W	Reset (0)	No operation			
	D4	OER	Overrun error flag R	Error	No error	0	R/W	
			W	Reset (0)	No operation			
	D3	RXTRG	Receive trigger/status R	Run	Stop	0	R/W	
			W	Trigger	No operation			
	D2	RXEN	Receive enable	Enable	Disable	0	R/W	
	D1	TXTRG	Transmit trigger/status R	Run	Stop	0	R/W	
			W	Trigger	No operation			
	D0	TXEN	Transmit enable	Enable	Disable	0	R/W	
00FF4A	D7	TRXD7	Transmit/Receive data D7 (MSB)	High	Low	X	R/W	
	D6	TRXD6	Transmit/Receive data D6					
	D5	TRXD5	Transmit/Receive data D5					
	D4	TRXD4	Transmit/Receive data D4					
	D3	TRXD3	Transmit/Receive data D3					
	D2	TRXD2	Transmit/Receive data D2					
	D1	TRXD1	Transmit/Receive data D1					
	D0	TRXD0	Transmit/Receive data D0 (LSB)					
00FF20	D7	PK01	K00–K07 interrupt priority register	PK01 PK00 PSIF1 PSIF0 PSW1 PSW0 PTM1 PTM0 1 1 1 0 0 1 0 0	Priority level Level 3 Level 2 Level 1 Level 0	0	R/W	
	D6	PK00						
	D5	PSIF1	Serial interface interrupt priority register			0	R/W	
	D4	PSIF0						
	D3	PSW1	Stopwatch timer interrupt priority register			0	R/W	
	D2	PSW0						
	D1	PTM1	Clock timer interrupt priority register			0	R/W	
	D0	PTM0						

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF23	D7	EPT1	Programmable timer 1 interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D6	EPT0	Programmable timer 0 interrupt enable register					
	D5	EK1	K10 and K11 interrupt enable register					
	D4	EK0H	K04–K07 interrupt enable register					
	D3	EK0L	K00–K03 interrupt enable register					
	D2	ESERR	Serial I/F (error) interrupt enable register					
	D1	ESREC	Serial I/F (receiving) interrupt enable register					
	D0	ESTRA	Serial I/F (transmitting) interrupt enable register					
00FF25	D7	FPT1	Programmable timer 1 interrupt factor flag	(R) Interrupt factor is generated	(R) No interrupt factor is generated	0	R/W	
	D6	FPT0	Programmable timer 0 interrupt factor flag					
	D5	FK1	K10 and K11 interrupt factor flag					
	D4	FK0H	K04–K07 interrupt factor flag					
	D3	FK0L	K00–K03 interrupt factor flag	(W) Reset	(W) No operation			
	D2	FSERR	Serial I/F (error) interrupt factor flag					
	D1	FSREC	Serial I/F (receiving) interrupt factor flag					
	D0	FSTRA	Serial I/F (transmitting) interrupt factor flag					

Specifications

Clock synchronous serial interface

<Conditions>

P10: SIN (Input)
P11: $\overline{\text{SOUT}}$ (Output)
P12: $\overline{\text{SCLK}}$ (Output)

Function and input/output direction of the I/O port are automatically decided when setting the serial mode.

P13: Slave READY (Input) Hand shake signal from slave side

Vector address setting for serial interface interrupt

(1) sio_init: Initialization for clock synchronous serial interface (master mode)

Sets the following in order to transmit/receive in a clock synchronous system:

- Serial interface function
- Clock synchronous master mode
- Synchronous clock OSC3 x 1/4
- Normal mode (OSC3 oscillation)
- Transmitting/receiving interrupt enable ($\overline{\text{IRQ2}}$)

(2) siorv, siorv_intr: Receiving of clock synchronous serial interface (master mode)

Checks handshake signal (P13) and stores a total of 256 bytes of received data from the slave into a built-in memory receive_buffer one byte at a time, using the receiving interrupt ($\overline{\text{IRQ2}}$).

(3) siotr, siotr_intr: Transmitting of clock synchronous serial interface (master mode)

Checks handshake signal (P13) and outputs a total of 256 bytes of transmitted data from a built-in memory trans_buffer to the slave one byte at a time, using the transmitting interrupt ($\overline{\text{IRQ2}}$).

Notes

(1) External routines are called for switching to OSC3 and OSC1. (external call: osc1toosc3, osc3toosc1)

(2) Switching the operating mode when the supply voltage is lower than the VD1 setting may cause a malfunction. Hence, the example routine checks the supply voltage when switching to the normal mode (OSC3) and terminates as a supply voltage error remains unprocessed if the supply voltage is lower than the VD1 setting. For this determination, vdd_ngf flag is used. (See "4 OSCILLATION CIRCUIT".)

(3) When switching from OSC3 to OSC1 (VD1 = 2.2 V → 1.3 V), the program example does not perform special checking of the supply voltage of SVD if the supply voltage is already more than the VD1 setting.

(4) To reset the interrupt factor flag, write "1" into the corresponding flags alone, using the AND or LD instruction. When the OR logic operation instruction has been used, "1" is written for the interrupt factor flags that have been set to "1" within the same address and those flags are then clear.

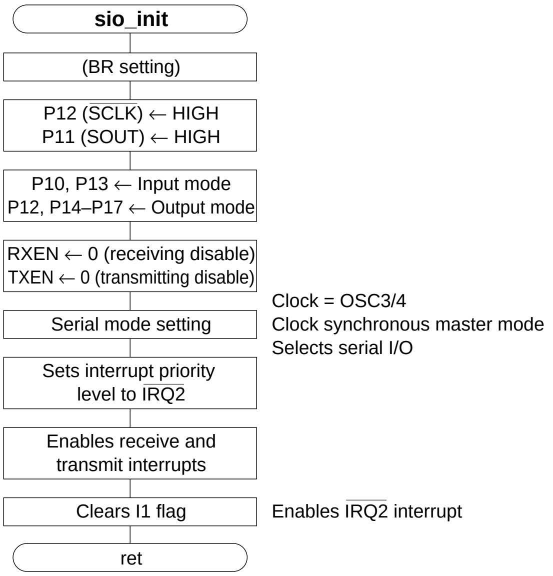
(5) The interrupt flags (I1 and I0) have not been reset in the interrupt processing routine of this program example, so an interrupt lower than $\overline{\text{IRQ2}}$ level is disabled at the time of generation. When you wish to accept the next interrupt after an interrupt has been generated, re-setting of the interrupt flags or resetting the interrupt factor flag is necessary after due consideration for the nesting level.

(6) When you have written "1" for the transmitting/receiving trigger and begin transmitting/receiving, first read the data and be sure to write "1" only on the necessary bits.
Another transmitting/receiving status (receiving status during transmitting, and transmitting status during receiving) has been allocated for reading to the same address as the transmitting/receiving triggers. For example, when directly writing to the transmitting trigger, using the OR instruction during a receiving operation (receiving status = "1"), the receiving status is read once and it is then written as the receiving trigger. It is the same as the current receiving trigger.

(7) In this program example for serial interface 1 (clock synchronous system), the vector address setting and program have been allocated from 003000H for the sake of convenience.

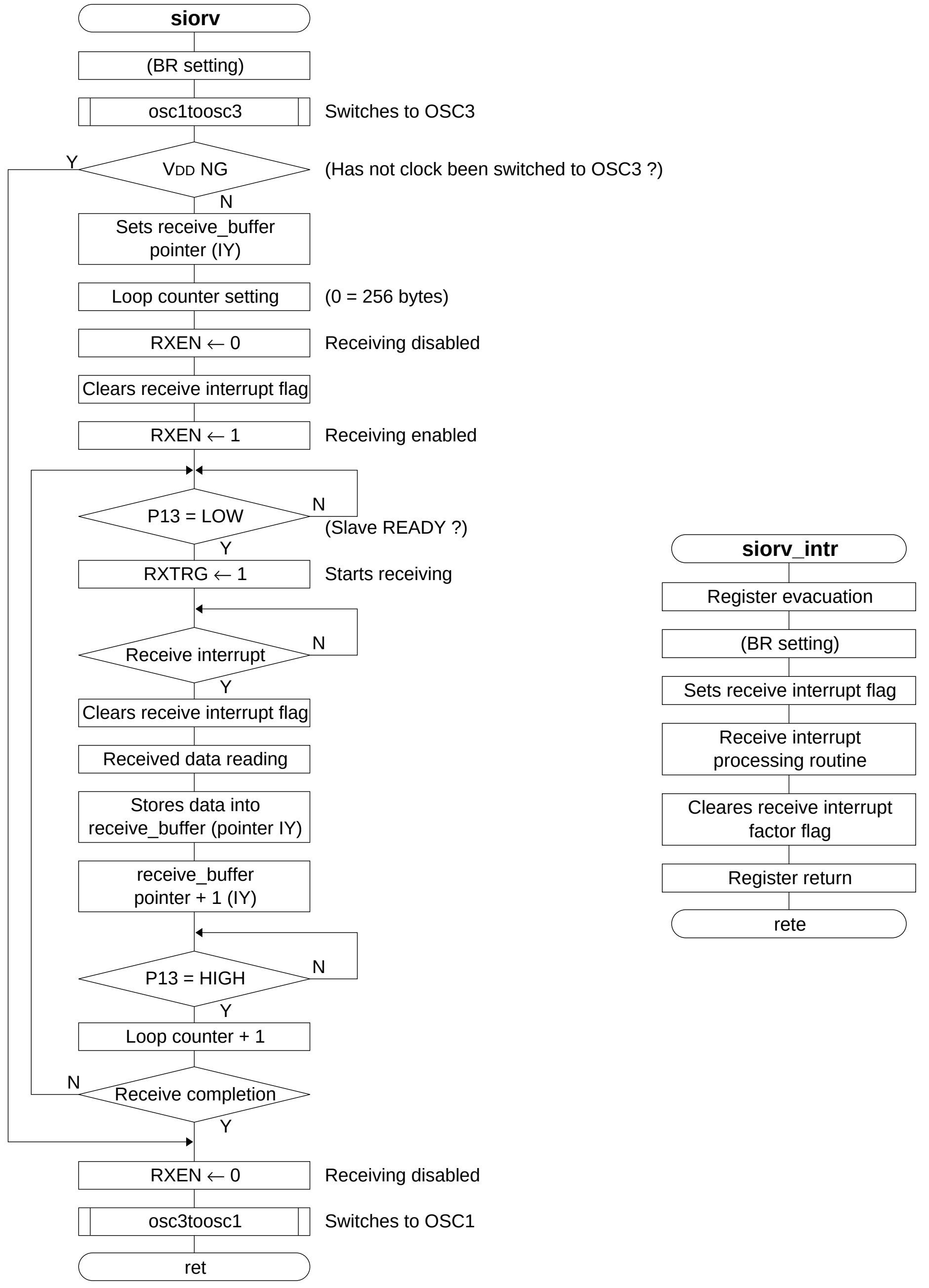
Flowchart

(1) Initialization for clock synchronous serial interface (master mode)


sio_init
(BR setting)
P12 (SCLK) ← HIGH
P11 (SOUT) ← HIGH
P10, P13 ← Input mode
P12, P14–P17 ← Output mode
RXEN ← 0 (receiving disable)
TXEN ← 0 (transmitting disable)
Serial mode setting
Clock = OSC3/4
Clock synchronous master mode
Selects serial I/O
Sets interrupt priority level to IRQ2
Enables receive and transmit interrupts
Clears I1 flag
Enables IRQ2 interrupt
ret

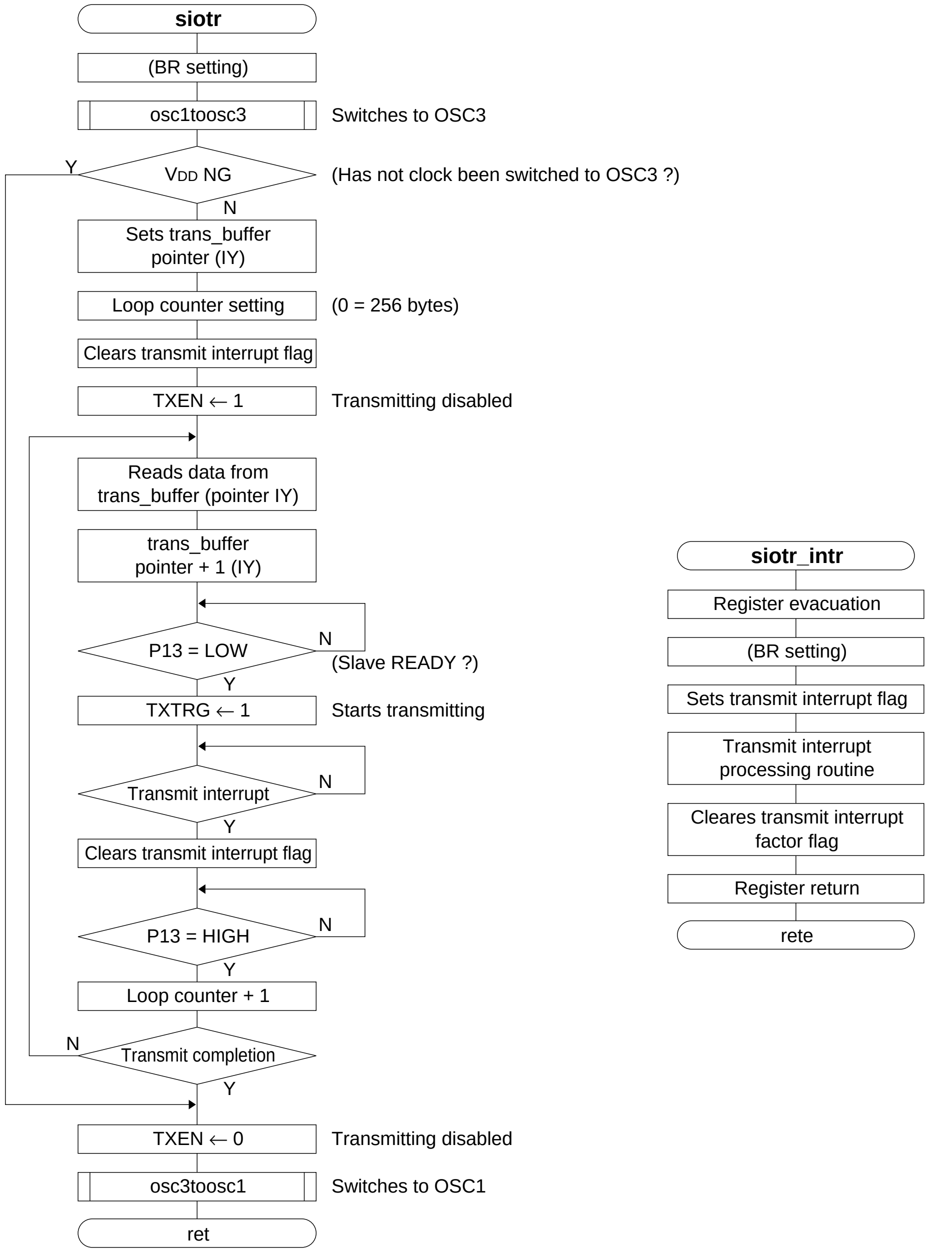
Flowchart

(2) Receiving of clock synchronous serial interface (master mode)



Flowchart

(3) Transmitting of clock synchronous serial interface (master mode)



Source List

Clock synchronous serial interface

```
         external     osc1toosc3,osc3toosc1
         external     vdd_ngf
         public       sio_init,siorv,siotr,siorv_intr,siotr_intr
         public       receive_buffer,trans_buffer,receive_flag,trans_flag
;
siorv_vector   equ   000012h                     ;sio receive interrupt vector offset
siotr_vector   equ   000014h                     ;sio trans interrupt vector offset
sio            equ   003000h                     ;program start address offset
br_io          equ   0ffh                        ;base reg. address (set i/o area)
mode           equ   00ff02h                     ;mode control reg.
ioc1           equ   00ff61h                     ;p1x i/o control reg.
p1d            equ   00ff63h                     ;p1x port data
smd            equ   00ff48h                     ;serial interface mode set reg.
ser            equ   00ff49h                     ;serial interface error and trriger reg
trxd           equ   00ff4ah                     ;trans/recive data reg.
intr_pr0       equ   00ff20h                     ;interrupt priority reg. 0
intr_en1       equ   00ff23h                     ;interrupt enable reg. 1
intr_fac1      equ   00ff25h                     ;interrupt factor reg. 1
;
         data
receive_buffer:db    [256]                       ;sio receive bufffer
trans_buffer:  db    [256]                       ;sio trans buffer
receive_flag:  db    [1]                         ;trans complete flag
trans_flag:    db    [1]                         ;receive complete flag
         code
```

Vector address setting for serial interface interrupt

```
intr_vectors:
;
         org   intr_vectors+siorv_vector
         dw    siorv_intr                      ;sio receive interrupt
;
         org   intr_vectors+siotr_vector
         dw    siotr_intr                      ;sio trans interrupt
;
```

(1) Initialization for clock synchronous serial interface (master mode)

```
         org   intr_vectors+sio
;************************************************************************
;*                                                                      *
;*       sio master mode initialize (p13=slave ready)                   *
;*                                                                      *
;************************************************************************
;*** initialize routine
sio_init:
;p17-14=programmable output,p13=slave ready,p12-10=sio terminal
         ld    br,#br_io                       ;set br reg. address to 0ffxxh
         ld    [br:low p1d],#11110110b         ;/sclk="h",sout="h"
         ld    [br:low ioc1],#11110110b
         ld    [br:low ser],#01110000b         ;rxen=dis.txen=dis.
;serial mode:no-parity,clock=fosc3/4,sio master mode and serial i/o select
         ld    [br:low smd],#00010001b         ;set serial interface mode
         ld    a,[br:low intr_pr0]             ;interrupt priority reg.
         and   a,#11001111b
         or    a,#00100000b                                                 (1)
         ld    [br:low intr_pr0],a             ;set psif1,0 to /irq2
         ld    a,[br:low intr_en1]
         and   a,#01111000b
         or    a,#00000011b
         ld    [br:low intr_en1],a             ;esrec and estra intr. en.

         ld    a,sc
         and   a,#00111111b
         or    a,#01000000b
         ld    sc,a                            ;i1 flag clear (en. /irq2 intr.)
         ret
```

Source List

(2) Receiving of clock synchronous serial interface (master mode)

```
;*********************************************************************
;*                                                                   *
;*       sio master mode receive (p13=slave ready)                   *
;*                                                                   *
;*********************************************************************
;*** control routine
siorv:
        ld    br,#br_io                     ;set br reg. address to 0ffxxh
        carl  osc1toosc3                    ;change osc1 to osc3 ***
        ld    a,[lod vdd_ngf]               ;vdd ng flag
        cp    a,#0ffh
        jrl   z,siorv02                     ;vdd error
;
        ld    iy,#lod receive_buffer        ;receive data buffer
        ld    b,#0                          ;set receive counter (00h=256)
        ld    a,[br:low ser]
        and   a,#00000001b
        ld    [br:low ser],a                ;rxen=0 (dis.) sio reset
        xor   a,a
        ld    [lod receive_flag],a          ;sio receive interrupt flag clear
        ld    a,[br:low ser]
        and   a,#00000001b
        or    a,#00000100b
        ld    [br:low ser],a                ;rxen=1 (en.)
;
;wait slave ready
siorv00:
        bit   [br:low p1d],#00001000b       ;p13(slave ready)="l
        jrs   nz,siorv00                                                  (2)
;
        ld    a,[br:low ser]
        and   a,#00000101b
        or    a,#00001000b
        ld    [br:low ser],a                ;rxtrg=set
;wait sio receive interrupt
siorv01:
        ld    a,[lod receive_flag]          ;sio receive interrput flag
        cp    a,#0ffh
        jrs   nz,siorv01
;
        xor   a,a
        ld    [lod receive_flag],a          ;clear sio receive interrupt flag
        ld    a,[br:low trxd]               ;receive data read
        ld    [iy],a                        ;set receive data buffer
siorv03:
        bit   [br:low p1d],#00001000b
        jrs   z,siorv03
        inc   iy                            ;receive buffer + 1
        djr   nz,siorv00                    ;until buffer end (256 bytes)
;
siorv02:
        ld    a,[br:low ser]
        and   a,#00000001b
        ld    [br:low ser],a                ;rxen=0 (dis.) sio reset
        carl  osc3toosc1                    ;change osc3 to osc1 ***
        ret
```

Source List

(3) Transmitting of clock synchronous serial interface (master mode)

```
;*************************************************************************
;*                                                                       *
;*       sio master mode trans (p13=slave ready)                         *
;*                                                                       *
;*************************************************************************
;*** control routine
siotr:
         ld    br,#br_io                       ;set br reg. address to 0ffxxh
         carl  osc1toosc3                      ;change osc1 to osc3 ***
         ld    a,[lod vdd_ngf]                 ;vdd ng flag
         cp    a,#0ffh
         jrl   z,siotr03                       ;vdd error
;
         ld    iy,#lod trans_buffer            ;trans data buffer
         ld    b,#0                            ;set trans counter (00h=256)
         ld    a,[br:low ser]
         and   a,#00000100b
         ld    [br:low ser],a                  ;txen=0 (dis.) sio reset
         xor   a,a
         ld    [lod trans_flag],a              ;sio trans interrupt flag clear
         ld    a,[br:low ser]
         and   a,#00000100b
         or    a,#00000001b
         ld    [br:low ser],a                  ;txen=en.
;wait slave ready
siotr00:
         ld    a,[iy]                          ;load trans data buffer
         ld    [br:low trxd],a                 ;set trans data
         inc   iy                              ;trans buffer + 1

siotr02:
         bit   [br:low p1d],#00001000b         ;p13(slave ready)="l
         jrs   nz,siotr02
;
         ld    a,[br:low ser]
         and   a,#00000101b
         or    a,#00000010b
         ld    [br:low ser],a                  ;txtrg=set
;wait sio trans interrupt
siotr01:
         ld    a,[lod trans_flag]              ;sio trans interrput flag
         cp    a,#0ffh
         jrs   nz,siotr01
;
         xor   a,a
         ld    [lod trans_flag],a              ;clear sio trans interrupt flag
siotr04:
         bit   [br:low p1d],#00001000b
         jrs   z,siotr04
         djr   nz,siotr00                      ;until buffer end (256 bytes)
;
siotr03:
         ld    a,[br:low ser]
         and   a,#00000100b
         ld    [br:low ser],a                  ;txen=0 (dis.) sio reset
         carl  osc3toosc1                      ;change osc3 to osc1 ***
         ret
```

(3)

Source List

(2) Receiveing interrupt

```
;*************************************************************************
;*                                                                       *
;*       sio master mode receive interrupt processing routine            *
;*                                                                       *
;*************************************************************************
siorv_intr:
         push  ale
;
         ld    br,#br_io                       ;set br reg. address to 0ffxxh
;
         ld    a,#0ffh
         ld    [lod receive_flag],a            ;set sio receive interrupt flag
;                                                                                        (2)
;
;        sio receive interrupt processing routine
;
;
         and   [br:low intr_fac1],#00000010b   ;clear fsrec flag
         pop   ale
         rete
```

(3)Transmitting interrupt

```
;*************************************************************************
;*                                                                       *
;*       sio master trans interrupt processing routine                   *
;*                                                                       *
;*************************************************************************
siotr_intr:
         push  ale
;
         ld    br,#br_io                       ;set br reg. address to 0ffxxh
;
         ld    a,#0ffh
         ld    [lod trans_flag],a              ;set sio trans interrupt flag
;                                                                                        (3)
;
;        sio trans interrupt processing routine
;
;
         and   [br:low intr_fac1],#00000001b   ;clear fstra flag
         pop   ale
         rete
;
         end
```

9 SERIAL INTERFACE 2 *(ASYNCHRONOUS INTERFACE)*

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF48	D7	–	–	–	–	–		"0" when being read
	D6	EPR	Parity enable register	With parity	Non parity	0	R/W	Only for asynchronous mode
	D5	PMD	Parity mode selection	Odd	Even	0	R/W	
	D4	SCS1	Clock source selection SCS1 SCS0 Clock source 1 1 Programmable timer			0	R/W	In the clock synchronous slave mode, external clock is selected.
	D3	SCS0	1 0 f_{OSC3} / 4 0 1 f_{OSC3} / 8 0 0 f_{OSC3} / 16			0	R/W	
	D2	SMD1	Serial I/F mode selection SMD1 SMD0 Mode 1 1 Asynchronous 8-bit			0	R/W	
	D1	SMD0	1 0 Asynchronous 7-bit 0 1 Clock synchronous slave 0 0 Clock synchronous master			0	R/W	
	D0	ESIF	Serial I/F enable register	Serial I/F	I/O port	0	R/W	
00FF49	D7	–	–	–	–	–		"0" when being read
	D6	FER	Framing error flag R	Error	No error	0	R/W	Only for asynchronous mode
			W	Reset (0)	No operation			
	D5	PER	Parity error flag R	Error	No error	0	R/W	
			W	Reset (0)	No operation			
	D4	OER	Overrun error flag R	Error	No error	0	R/W	
			W	Reset (0)	No operation			
	D3	RXTRG	Receive trigger/status R	Run	Stop	0	R/W	
			W	Trigger	No operation			
	D2	RXEN	Receive enable	Enable	Disable	0	R/W	
	D1	TXTRG	Transmit trigger/status R	Run	Stop	0	R/W	
			W	Trigger	No operation			
	D0	TXEN	Transmit enable	Enable	Disable	0	R/W	
00FF4A	D7	TRXD7	Transmit/Receive data D7 (MSB)	High	Low	X	R/W	
	D6	TRXD6	Transmit/Receive data D6					
	D5	TRXD5	Transmit/Receive data D5					
	D4	TRXD4	Transmit/Receive data D4					
	D3	TRXD3	Transmit/Receive data D3					
	D2	TRXD2	Transmit/Receive data D2					
	D1	TRXD1	Transmit/Receive data D1					
	D0	TRXD0	Transmit/Receive data D0 (LSB)					
00FF20	D7	PK01	K00–K07 interrupt priority register	PK01 PK00 PSIF1 PSIF0 PSW1 PSW0 Priority PTM1 PTM0 level 1 1 Level 3 1 0 Level 2 0 1 Level 1 0 0 Level 0		0	R/W	
	D6	PK00						
	D5	PSIF1	Serial interface interrupt priority register			0	R/W	
	D4	PSIF0						
	D3	PSW1	Stopwatch timer interrupt priority register			0	R/W	
	D2	PSW0						
	D1	PTM1	Clock timer interrupt priority register			0	R/W	
	D0	PTM0						

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF23	D7	EPT1	Programmable timer 1 interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D6	EPT0	Programmable timer 0 interrupt enable register					
	D5	EK1	K10 and K11 interrupt enable register					
	D4	EK0H	K04–K07 interrupt enable register					
	D3	EK0L	K00–K03 interrupt enable register					
	D2	ESERR	Serial I/F (error) interrupt enable register					
	D1	ESREC	Serial I/F (receiving) interrupt enable register					
	D0	ESTRA	Serial I/F (transmitting) interrupt enable register					
00FF25	D7	FPT1	Programmable timer 1 interrupt factor flag	(R) Interrupt factor is generated	(R) No interrupt factor is generated	0	R/W	
	D6	FPT0	Programmable timer 0 interrupt factor flag					
	D5	FK1	K10 and K11 interrupt factor flag					
	D4	FK0H	K04–K07 interrupt factor flag					
	D3	FK0L	K00–K03 interrupt factor flag	(W) Reset	(W) No operation			
	D2	FSERR	Serial I/F (error) interrupt factor flag					
	D1	FSREC	Serial I/F (receiving) interrupt factor flag					
	D0	FSTRA	Serial I/F (transmitting) interrupt factor flag					

Specifications

Asynchronous serial interface

<Conditions>

P10: SIN	(Input)	Function and input/output direction of the I/O port are automatically decided when setting the serial mode.
P11: SOUT	(Output)	
P12: Hand shake	(Output)	Unused (In this program example, handshake signals during transmission are ignored.)
P13: Hand shake	(Input)	

Vector address setting for serial interface interrupt

(1) async_init: Initialization for asynchronous serial interface (8-bit mode)

Sets the following in order to transmit/receive in an asynchronous system:

- Serial interface function
- Normal mode (OSC3 oscillation)
- Asynchronous 8-bit mode, even parity
- Transmitting/receiving interrupt enable ($\overline{\text{IRQ2}}$)
- Synchronous clock = Programmable timer

Transmission baud rate clock has been set to 9,600 bps (When OSC3 = 4.9152 MHz) using programmable timer 1 (8 bits).

(2) asyncrv, asyncrv_intr, asyncerr_intr: Receiving of asynchronous serial interface (8-bit mode)

Performs switching to the OSC3 clock and starting the programmable timer, and stores a total of 256 bytes of received data into the built-in memory receive_buffer one byte at a time, using the receiving interrupt ($\overline{\text{IRQ2}}$). At this time, if a receiving error occurs, it suspends receiving processing at that point.

(3) asynctr, async_intr: Transmitting of asynchronous serial interface (8-bit mode)

Performs switching to the OSC3 clock and starting the programmable timer, outputs a total of 256 bytes of transmitted data from a built-in memory trans_buffer one byte at a time, using the transmitting interrupt ($\overline{\text{IRQ2}}$).

Notes

(1) External routines are called for switching to OSC3 and OSC1. (external call: osc1toosc3, osc3toosc1)

(2) Switching the operating mode when the supply voltage is lower than the V_{D1} setting may cause a malfunction. Hence, the example routine checks the supply voltage when switching to the normal mode (OSC3) and terminates as a supply voltage error remains unprocessed if the supply voltage is lower than the V_{D1} setting. For this determination, vdd_ngf flag is used. (See "4 OSCILLATION CIRCUIT".)

(3) When switching from OSC3 to OSC1 (V_{D1} = 2.2 V → 1.3 V), the program example does not perform special checking of the supply voltage of SVD if the supply voltage is already more than the V_{D1} setting.

(4) The example routine does not check the handshake signal when transmitting/receiving. If this routine is used for an actual program, pay attention to the timing of transmitting/receiving, or check the timing using a handshake signal.

(5) The 9,600 bps baud rate has been set on the condition that the 4.9152 MHz OSC3 oscillation clock is used.

(6) To reset the interrupt factor flag, write "1" into the corresponding flags alone, using the AND or LD instruction. When the OR logic operation instruction has been used, "1" is written for the interrupt factor flags that have been set to "1" within the same address and those flags are then clear.

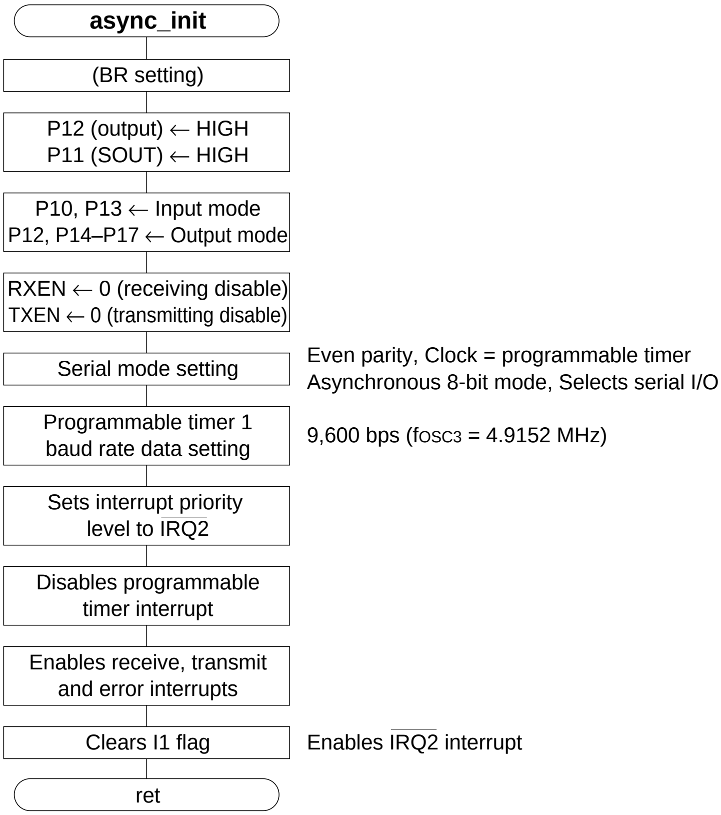
(7) The interrupt flags (I1 and I0) have not been reset in the interrupt processing routine of this program example, so an interrupt lower than$\overline{\text{IRQ2}}$ level is disabled at the time of generation. When you wish to accept the next interrupt after an interrupt has been generated, re-setting of the interrupt flags or resetting the interrupt factor flag is necessary after due consideration for the nesting level.

(8) When you have written "1" for the transmitting/receiving trigger and begin transmitting/receiving, first read the data and be sure to write "1" only on the necessary bits.
Also, when writing "1" to reset the receive error flag to "0", similar care is necessary.
Another transmitting/receiving status (receiving status during transmitting, transmitting status during receiving, and receiving error flag) has been allocated for reading to the same address as the transmitting/receiving triggers. For example, when directly writing to the transmitting trigger, using the OR instruction during a receiving operation (receiving status = "1"), the receiving status is read once and it is then written as the receiving trigger.
Also when the receiving error flag has been set to "1", the receiving error flag is written and reset by an OR instruction. It is the same as setting the receiving trigger or resetting the receiving error flag.

(9) In this program example for serial interface 2 (asynchronous system), the vector address setting and program have been allocated from 003000H for the sake of convenience.

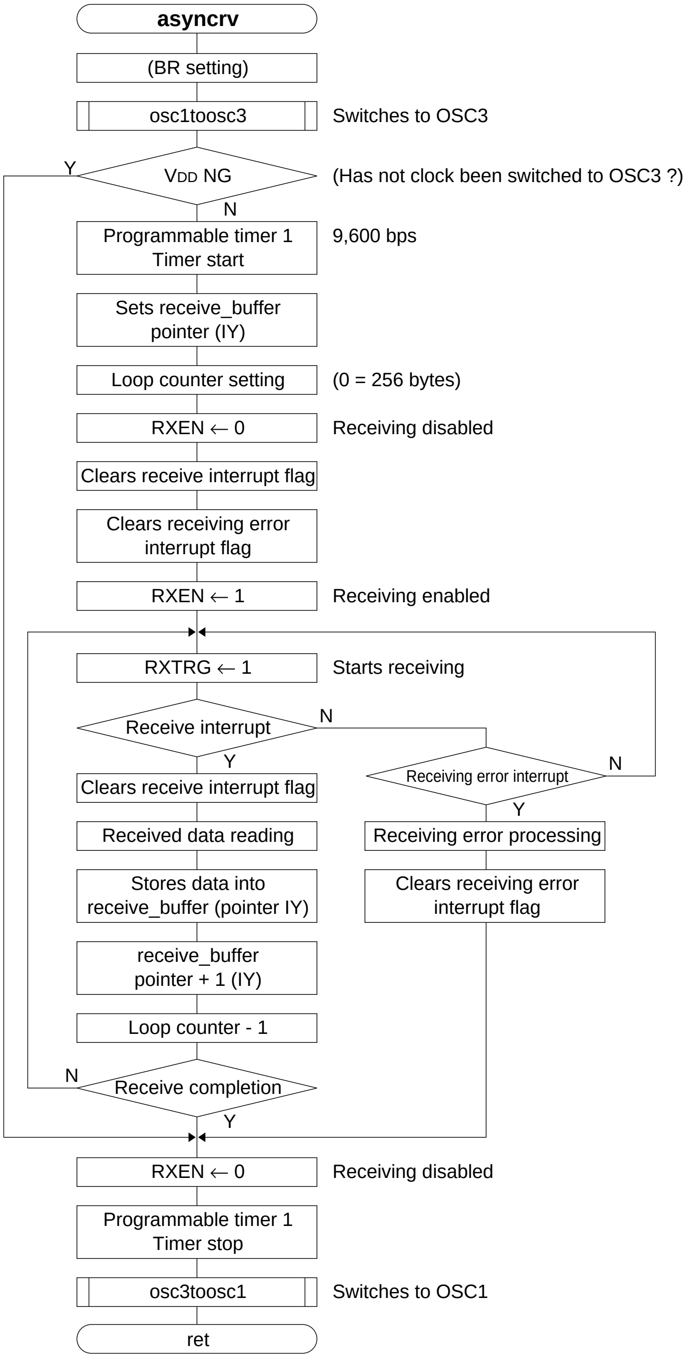
Flowchart

(1) Initialization for asynchronous serial interface (8-bit mode)



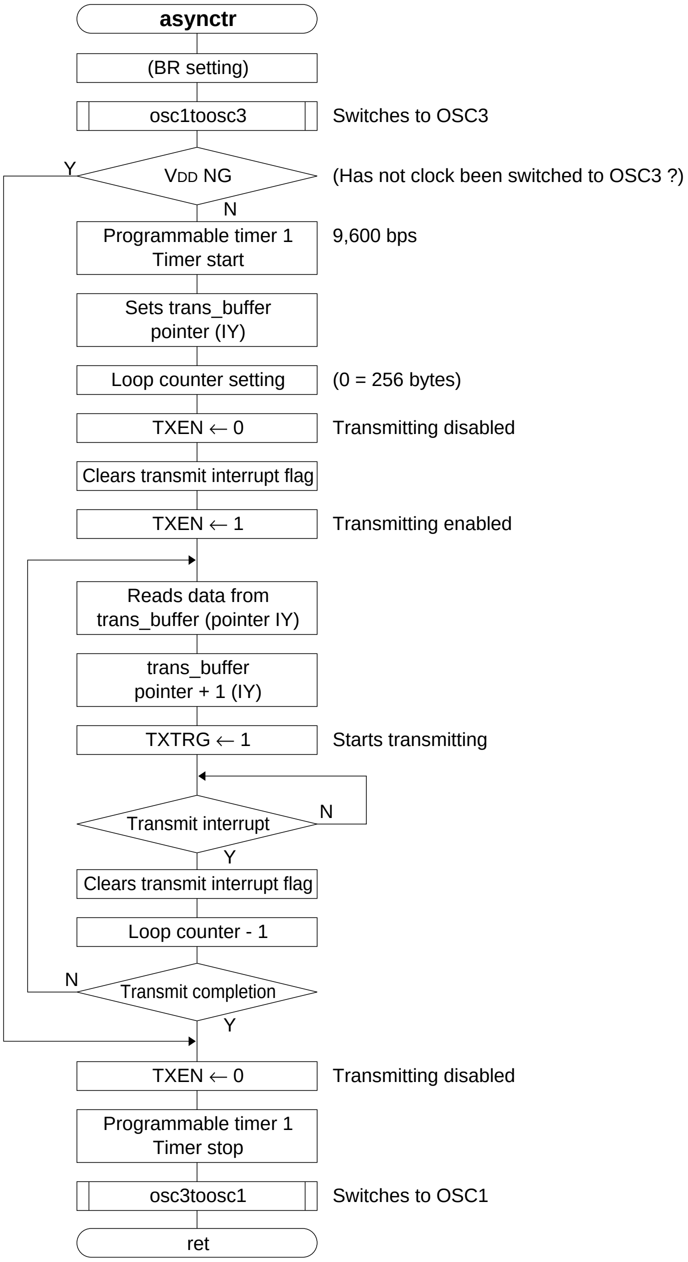
Flowchart

(2) Receiving of asynchronous serial interface (8-bit mode)



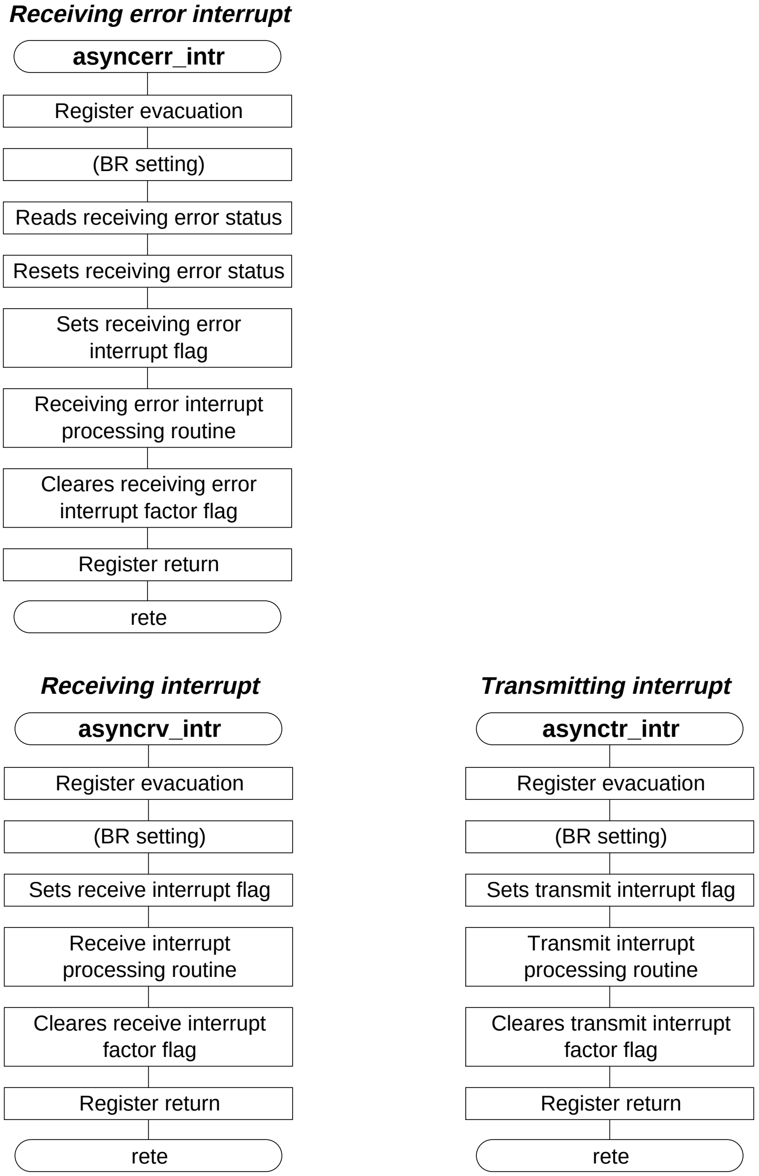
Flowchart

(3) Transmitting of asynchronous serial interface (8-bit mode)



Flowchart

(1) (2) (3) Interrupts of asynchronous serial interface (8-bit mode)


Receiving error interrupt
asyncerr_intr
Register evacuation
(BR setting)
Reads receiving error status
Resets receiving error status
Sets receiving error interrupt flag
Receiving error interrupt processing routine
Cleares receiving error interrupt factor flag
Register return
rete
Receiving interrupt
asyncrv_intr
Register evacuation
(BR setting)
Sets receive interrupt flag
Receive interrupt processing routine
Cleares receive interrupt factor flag
Register return
rete
Transmitting interrupt
asynctr_intr
Register evacuation
(BR setting)
Sets transmit interrupt flag
Transmit interrupt processing routine
Cleares transmit interrupt factor flag
Register return
rete

Source List

Asynchronous serial interface

```
          external    osc1toosc3,osc3toosc1
          external    vdd_ngf
          public      async_init,asyncrv,asynctr
          public      asyncerr_intr,asyncrv_intr,asynctr_intr
          public      receive_buffer,trans_buffer,receive_flag,trans_flag
          public      error_flag,bps_data
;
asyncerr_vector equ   000010h                  ;async error interrupt vector offset
asyncrv_vector  equ   000012h                  ;async receive interrupt vector offset
asynctr_vector  equ   000014h                  ;async trans interrupt vector offset
async           equ   003000h                  ;program start address offset
br_io           equ   0ffh                     ;base reg. address (set i/o area)
mode            equ   00ff02h                  ;mode control reg.
ioc1            equ   00ff61h                  ;p1x i/o control reg.
p1d             equ   00ff63h                  ;p1x port data
smd             equ   00ff48h                  ;serial interface mode set reg.
ser             equ   00ff49h                  ;serial interface error and trriger reg
trxd            equ   00ff4ah                  ;trans/recive data reg.
pt_mode0        equ   00ff30h                  ;programmable timer mode set reg. 0
pt_mode2        equ   00ff32h                  ;programmable timer mode set reg. 2
rld1            equ   00ff34h                  ;programmable timer 1 reload data
intr_pr0        equ   00ff20h                  ;interrupt priority reg. 0
intr_pr1        equ   00ff21h                  ;interrupt priority reg. 1
intr_en1        equ   00ff23h                  ;interrupt enable reg. 1
intr_fac1       equ   00ff25h                  ;interrupt factor reg. 1
;
          data
receive_buffer: db    [256]                    ;async receive bufffer
trans_buffer:   db    [256]                    ;async trans buffer
error_flag:     db    [1]                      ;async error flag
receive_flag:   db    [1]                      ;trans complete flag
trans_flag:     db    [1]                      ;receive complete flag
          code
```

Vector address setting for serial interface interrupt

```
intr_vectors:
;
          org   intr_vectors+asyncerr_vector
          dw    asyncerr_intr                  ;async error interrupt
;
          org   intr_vectors+asyncrv_vector
          dw    asyncrv_intr                   ;async receive interrupt
;
          org   intr_vectors+asynctr_vector
          dw    asynctr_intr                   ;async trans interrupt
;
```

(1) Initialization for asynchronous serial interface (8-bit mode)

```
          org   intr_vectors+async
bps_data: db    0fh                            ;baud rate(osc3*1/1 9600bps:4.9152mhz)
;************************************************************************
;*                                                                      *
;*       async 8-bit mode initialize (p13 and 12 = hand shake:not use)  *
;*                                                                      *
;************************************************************************
;*** initialize routine
async_init:
;p17-14=programmable output,p13-12=hand shake,p11-10=async terminal          (1)
          ld    br,#br_io                      ;set br reg. address to 0ffxxh
```

Source List

```
         ld    [br:low p1d],#11110110b         ;sout="h" and no hand shake
         ld    [br:low ioc1],#11110110b
         ld    [br:low ser],#01110000b         ;rxen=dis.txen=dis.
;serial mode:even parity,clock=timer 1,async 8-bit mode and serial i/o select
         ld    [br:low smd],#01011111b         ;set serial interface mode
;pt:timer 8bit*2,pulse output=timer 1,pulse output=dis.,clock (timer0&1=fosc3)
         ld    [br:low pt_mode0],#00001011b
         ld    a,[loc bps_data]
         ld    [br:low rld1],a                 ;set reload data reg.
         ld    a,[br:low intr_pr0]             ;interrupt priority reg.
         and   a,#11001111b
         or    a,#00100000b
         ld    [br:low intr_pr0],a             ;set psif=/irq2
         ld    a,[br:low intr_pr1]             ;interrupt priority reg.
         and   a,#11110011b
         or    a,#00001100b
         ld    [br:low intr_pr1],a
         ld    a,[low intr_en1]
         and   a,#01111000b                    ;ept1 interrupt dis.(baud rate control)
         or    a,#00000111b
         ld    [br:low intr_en1],a             ;eserr esrec and estra intr. en
         ld    a,sc
         and   a,#00111111b
         or    a,#01000000b
         ld    sc,a                            ;i1 flag clear (en. /irq2 intr.)
         ret
```

(1)

(2) Receiving of asynchronous serial interface (8-bit mode)

```
;************************************************************************
;*                                                                      *
;*       async 8-bit mode receive (p13 and 12 = hand shake:not use)     *
;*                                                                      *
;************************************************************************
;*** control routine
asyncrv:
         ld    br,#br_io                       ;set br reg. address to 0ffxxh
         carl  osc1toosc3                      ;change osc1 to osc3 ***
         ld    a,[lod vdd_ngf]                 ;vdd ng flag
         cp    a,#0ffh
         jrl   z,asyncrv03
;psc=1/1*fosc3(4.9152mhz),timer1=reload mode and reload data set to timer 1
         ld    [br:low pt_mode2],#00000110b
         or    [br:low pt_mode2],#00000001b    ;timer 1 start (baud rate)
         ld    iy,#lod receive_buffer          ;receive data buffer
         ld    b,#0                            ;set receive counter (00h=256)
         ld    a,[br:low ser]
         and   a,#00000001b
         ld    [br:low ser],a                  ;rxen=0 (dis.) async reset
         xor   a,a
         ld    [lod receive_flag],a            ;async receive interrpt flag clear
         ld    [lod error_flag],a              ;async receive error flag clear
         ld    a,[br:low ser]
         and   a,#00000001b
         or    a,#00000100b
         ld    [br:low ser],a                  ;rxen=1 (en.)
;no hand shake
asyncrv00:
         ld    a,[br:low ser]
         and   a,#00000101b
         or    a,#00001000b
         ld    [br:low ser],a                  ;rxtrg=set and error reset
;wait async receive interrupt
asyncrv01:
```

(2)

Source List

```
         ld    a,[lod receive_flag]            ;async receive interrput flag
         cp    a,#0ffh
         jrs   z,asyncrv02
;
         ld    a,[lod error_flag]              ;async error interrupt flag
         cp    a,#00h
         jrs   z,asyncrv01
;receive error occurrs
;
;
;        async receive error processing
;
;
         xor   a,a
         ld    [lod error_flag],a              ;clear error interrupt flag
         jrs   asyncrv03                                                          (2)
;
;receive no error
asyncrv02:
         xor   a,a
         ld    [lod receive_flag],a            ;clear async receive interrupt flag
         ld    a,[br:low trxd]                 ;receive data read
         ld    [iy],a                          ;set receive data buffer
         inc   iy                              ;receive buffer + 1
         djr   nz,asyncrv00                    ;until buffer end (256 bytes)
;
asyncrv03:
         ld    a,[br:low ser]
         and   a,#00000001b
         ld    [br:low ser],a                  ;rxen=0 (dis.) async reset
         and   [br:low pt_mode2],#00011100b    ;timer 1 stop (baud rate)
         carl  osc3toosc1                      ;change osc3 to osc1 ***
         ret
```

(3) Transmitting of asynchronous serial interface (8-bit mode)

```
;*************************************************************************
;*                                                                       *
;*       async 8-bit mode trans (p13 and 12 = hand shake:not use)        *
;*                                                                       *
;*************************************************************************
;*** control routine
asynctr:
         ld    br,#br_io                       ;set br reg. address to 0ffxxh
         carl  osc1toosc3                      ;change osc1 to osc3 ***
         ld    a,[lod vdd_ngf]                 ;vdd ng flag
         cp    a,#0ffh
         jrl   z,asynctr02                     ;vdd error
;psc=1/1*fosc3(4.9152mhz),timer1=reload mode and reload data set to timer 1
         ld    [br:low pt_mode2],#00000110b
         or    [br:low pt_mode2],#00000001b    ;timer 1 start (baud rate)
         ld    iy,#lod trans_buffer            ;trans data buffer
         ld    b,#0                            ;set trans counter (00h=256)       (3)
         ld    a,[br:low ser]
         and   a,#00000100b
         ld    [br:low ser],a                  ;txen=0 (dis.) async reset
         xor   a,a
         ld    [lod trans_flag],a              ;async trans interrupt flag clear
         ld    a,[br:low ser]
         and   a,#00000100b
         or    a,#00000001b
         ld    [br:low ser],a                  ;txen=en.
;no hand shake
```

Source List

```
asynctr00:
        ld    a,[iy]                        ;load trans data buffer
        ld    [br:low trxd],a               ;set trans data
        inc   iy                            ;trans buffer + 1
        ld    a,[br:low ser]
        and   a,#00000101b
        or    a,#00000010b
        ld    [br:low ser],a                ;txtrg=set
;wait async trans interrupt
asynctr01:
        ld    a,[lod trans_flag]            ;async trans interrpu flag
        cp    a,#0ffh
        jrs   nz,asynctr01                                                (3)
;
        xor   a,a
        ld    [lod trans_flag],a            ;clear async trans interrupt flag
        djr   nz,asynctr00                  ;until buffer end (256 bytes)
;
asynctr02:
        ld    a,[br:low ser]
        and   a,#00000100b
        ld    [br:low ser],a                ;txen=0 (dis.) async reset
        and   [br:low pt_mode2],#00011100b  ;timer 1 stop (baud rate)
        carl  osc3toosc1                    ;change osc3 to osc1 ***
        ret
```

(2) Receiving error interrupt

```
;*************************************************************************
;*                                                                       *
;*      async 8-bit mode error interrupt processing routine              *
;*                                                                       *
;*************************************************************************
asyncerr_intr:
        push  ale
;
        ld    br,#br_io                     ;set br reg. address to 0ffxxh
;
        ld    a,[br:low ser]
        and   a,#01110101b
        ld    [br:low ser],a                ;receive error status reset
        and   a,#01110000b                  ;ignore bits clear
        ld    [lod error_flag],a            ;set async error interrupt flag
;
;
;       async error interrupt processing routine
;
;                                                                         (2)
        and   [br:low intr_fac1],#00000100b ;clear fserr flag
        pop   ale
        rete
```

(2) Receiving interrupt

```
;*************************************************************************
;*                                                                       *
;*      async 8-bit mode receive interrupt processing routine            *
;*                                                                       *
;*************************************************************************
asyncrv_intr:
        push  ale
;
        ld    br,#br_io                     ;set br reg. address to 0ffxxh
```

Source List

```
;
         ld    a,#0ffh
         ld    [lod receive_flag],a            ;set async receive interrupt flag
;
;
;        async receive interrupt processing routine                                (2)
;
;
         and   [br:low intr_fac1],#00000010b   ;clear fsrec flag
         pop   ale
         rete
```

(3) Transmitting interrupt

```
;************************************************************************
;*                                                                      *
;*       async 8-bit mode trans interrupt processing routine            *
;*                                                                      *
;************************************************************************
asynctr_intr:
         push  ale
;
         ld    br,#br_io                       ;set br reg. address to 0ffxxh
;
         ld    a,#0ffh
         ld    [lod trans_flag],a              ;set async trans interrupt flag
;
;
;        async trans interrupt processing routine                                  (3)
;
;
         and   [br:low intr_fac1],#00000001b   ;clear fstra flag
         pop   ale
         rete
         end
```

10 CLOCK TIMER

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF40	D7	–	–	–	–	–		"0" when being read
	D6	FOUT2	FOUT frequency selection FOUT2 FOUT1 FOUT0 Frequency 0 0 0 f_{OSC1} / 1 0 0 1 f_{OSC1} / 2 0 1 0 f_{OSC1} / 4 0 1 1 f_{OSC1} / 8 1 0 0 f_{OSC3} / 1 1 0 1 f_{OSC3} / 2 1 1 0 f_{OSC3} / 4 1 1 1 f_{OSC3} / 8			0	R/W	
	D5	FOUT1				0	R/W	
	D4	FOUT0				0	R/W	
	D3	FOUTON	FOUT output control	On	Off	0	R/W	
	D2	WDRST	Watchdog timer reset	Reset	No operation	–	W	Constantly "0" when being read
	D1	TMRST	Clock timer reset	Reset	No operation	–	W	
	D0	TMRUN	Clock timer Run/Stop control	Run	Stop	0	R/W	
00FF41	D7	TMD7	Clock timer data 1 Hz	High	Low	0	R	
	D6	TMD6	Clock timer data 2 Hz					
	D5	TMD5	Clock timer data 4 Hz					
	D4	TMD4	Clock timer data 8 Hz					
	D3	TMD3	Clock timer data 16 Hz					
	D2	TMD2	Clock timer data 32 Hz					
	D1	TMD1	Clock timer data 64 Hz					
	D0	TMD0	Clock timer data 128 Hz					
00FF20	D7	PK01	K00–K07 interrupt priority register	PK01 PK00 PSIF1 PSIF0 PSW1 PSW0 Priority PTM1 PTM0 level 1 1 Level 3 1 0 Level 2 0 1 Level 1 0 0 Level 0		0	R/W	
	D6	PK00						
	D5	PSIF1	Serial interface interrupt priority register			0	R/W	
	D4	PSIF0						
	D3	PSW1	Stopwatch timer interrupt priority register			0	R/W	
	D2	PSW0						
	D1	PTM1	Clock timer interrupt priority register			0	R/W	
	D0	PTM0						
00FF22	D7	–	–	–	–	–		"0" when being read
	D6	ESW100	Stopwatch timer 100 Hz interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D5	ESW10	Stopwatch timer 10 Hz interrupt enable register					
	D4	ESW1	Stopwatch timer 1 Hz interrupt enable register					
	D3	ETM32	Clock timer 32 Hz interrupt enable register					
	D2	ETM8	Clock timer 8 Hz interrupt enable register					
	D1	ETM2	Clock timer 2 Hz interrupt enable register					
	D0	ETM1	Clock timer 1 Hz interrupt enable register					
00FF24	D7	–	–	–	–	–		"0" when being read
	D6	FSW100	Stopwatch timer 100 Hz interrupt factor flag	(R) Interrupt factor is generated (W) Reset	(R) No interrupt factor is generated (W) No operation	0	R/W	
	D5	FSW10	Stopwatch timer 10 Hz interrupt factor flag					
	D4	FSW1	Stopwatch timer 1 Hz interrupt factor flag					
	D3	FTM32	Clock timer 32 Hz interrupt factor flag					
	D2	FTM8	Clock timer 8 Hz interrupt factor flag					
	D1	FTM2	Clock timer 2 Hz interrupt factor flag					
	D0	FTM1	Clock timer 1 Hz interrupt factor flag					

Specifications

Control of clock timer

Vector address setting for clock timer interrupt

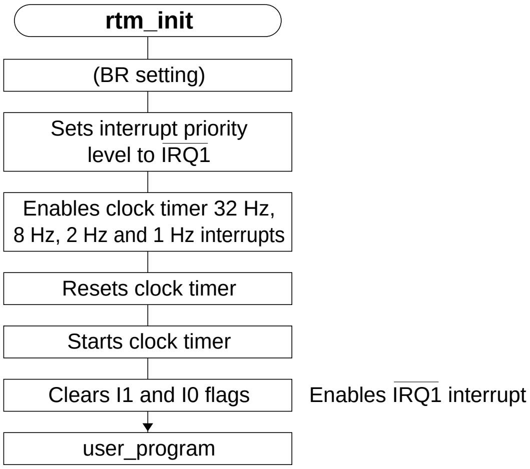
(1) rtm_init: Initialization for clock timer

Enables the respective 32 Hz, 8 Hz, 2 Hz and 1 Hz interrupts of the clock timer, clears the timer data and starts the clock timer. The interrupt level has been set at $\overline{\text{IRQ1}}$.

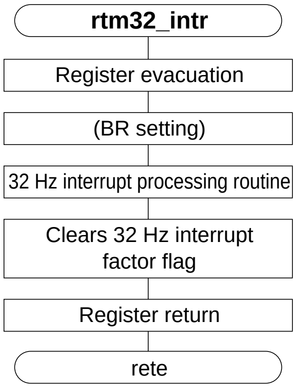
(2) rtm32_intr, rtm8_intr, rtm2_intr, rtm1_intr:Clock timer interrupt processing

Flowchart

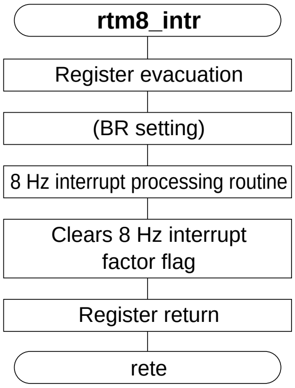
(1) Initialization for clock timer


rtm_init
(BR setting)
Sets interrupt priority level to $\overline{\text{IRQ1}}$
Enables clock timer 32 Hz, 8 Hz, 2 Hz and 1 Hz interrupts
Resets clock timer
Starts clock timer
Clears I1 and I0 flags
Enables $\overline{\text{IRQ1}}$ interrupt
user_program

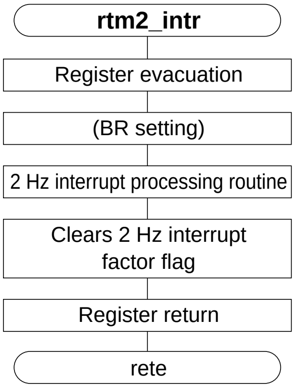
(2) 32 Hz clock timer interrupt processing


rtm32_intr
Register evacuation
(BR setting)
32 Hz interrupt processing routine
Clears 32 Hz interrupt factor flag
Register return
rete

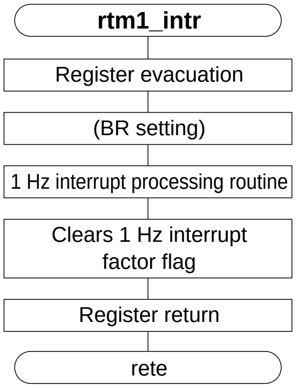
(2) 8 Hz clock timer interrupt processing


rtm8_intr
Register evacuation
(BR setting)
8 Hz interrupt processing routine
Clears 8 Hz interrupt factor flag
Register return
rete

(2) 2 Hz clock timer interrupt processing


rtm2_intr
Register evacuation
(BR setting)
2 Hz interrupt processing routine
Clears 2 Hz interrupt factor flag
Register return
rete

(2) 1 Hz clock timer interrupt processing


rtm1_intr
Register evacuation
(BR setting)
1 Hz interrupt processing routine
Clears 1 Hz interrupt factor flag
Register return
rete

Notes

(1) To reset the interrupt factor flag, write "1" into the corresponding flags alone, using the AND or LD instruction. When the OR logic operation instruction has been used, "1" is written for the interrupt factor flags that have been set to "1" within the same address and those flags are then clear.

(2) The interrupt flags (I1 and I0) have not been reset in the interrupt processing routine of this program example, so an interrupt lower than IRQ1 level is disabled at the time of generation. When you wish to accept the next interrupt after an interrupt has been generated, re-setting of the interrupt flags or resetting the interrupt factor flag is necessary after due consideration for the nesting level.

(3) When stopping the clock timer by writing "0" into the RUN/STOP control register for the clock timer, the clock timer count actually stops when it advances one count with the timing synchronized to the 256 Hz input clock. For this reason, when the clock timer stops, if the 32 Hz, 8 Hz, 2 Hz and 1 Hz interrupt factors are generated, the respective interrupt factor flags are set and if interrupt is enabled, an interrupt is generated. Thus, you should add an interrupt processing and interrupt factor flag resetting, if necessary.

(4) In this program example for the clock timer, the vector address setting and program have been allocated from 003000H for the sake of convenience.

Source List

Control of clock timer

```
         external     user_program
         public       clock_init,clock32_intr,clock8_intr,clock2_intr,clock1_intr
;
clock32_vector equ    00001ch                      ;clock32hz interrupt vector offset
clock8_vector  equ    00001eh                      ;clock8hz interrupt vector offset
clock2_vector  equ    000020h                      ;clock2hz interrupt vector offset
clock1_vector  equ    000022h                      ;clock1hz interrupt vector offset
clock          equ    003000h                      ;program start address offset
br_io          equ    0ffh                         ;base reg. address (set i/o area)
clock_mode     equ    00ff40h                      ;timer mode set reg.
clockd         equ    00ff41h                      ;timer data
intr_pr0       equ    00ff20h                      ;interrupt priority reg. 0
intr_en0       equ    00ff22h                      ;interrupt enable reg. 0
intr_fac0      equ    00ff24h                      ;interrupt factor flag reg.
;
         code
```

Vector address setting for clock timer interrupt

```
intr_vectors:
         org    intr_vectors+clock32_vector
         dw     clock32_intr                       ;clock 32hz interrupt
;
         org    intr_vectors+clock8_vector
         dw     clock8_intr                        ;clock 8hz interrupt
;
         org    intr_vectors+clock2_vector
         dw     clock2_intr                        ;clock 2hz interrupt
;
         org    intr_vectors+clock1_vector
         dw     clock1_intr                        ;clock 1hz interrupt
;
```

Source List

(1) Initialization for clock timer

```
         org    intr_vectors+clock
;*********************************************************************
;*                                                                   *
;*       clock timer initialize (32,8,2 and 1hz interrupt enable)    *
;*                                                                   *
;*********************************************************************
;*** initialize routine
clock_init:
         ld     br,#br_io                        ;set br reg. address to 0ffxxh
         ld     a,[br:low intr_pr0]              ;interrupt priority reg.
         and    a,#11111100b
         or     a,#00000001b
         ld     [br:low intr_pr0],a              ;set ptm=/irq1
;etm32,etm8,etm2 and etm1 (en. /irq1) intr.
         or     [br:low intr_en0],#00001111b
         or     [br:low clock_mode],#00000010b   ;clock timer counter reset       (1)
         or     [br:low clock_mode],#00000001b   ;clock timer start
         and    sc,#00111111b                    ;i1 and i0 flag clear
;*********************************************************************
;*** start clock timer interrupt
;
;
         jrl    user_program
;
;
```

(2) 32 Hz clock timer interrupt processing

```
;*********************************************************************
;*                                                                   *
;*       clock timer 32hz interrupt processing routine               *
;*                                                                   *
;*********************************************************************
clock32_intr:
         push   ale
;
         ld     br,#br_io                        ;set br reg. address to 0ffxxh
;
;
;
;        clock timer 32hz processing routine
;
;
         and    [br:low intr_fac0],#00001000b    ;clear etm32 flag
         pop    ale
         rete
```

(2)

(2) 8 Hz clock timer interrupt processing

```
;*********************************************************************
;*                                                                   *
;*       clock timer 8hz interrupt processing routine                *
;*                                                                   *
;*********************************************************************
clock8_intr:
         push   ale
;
         ld     br,#br_io                        ;set br reg. address to 0ffxxh
;
;
;
```

Source List

```
;       clock timer 8hz processing routine
;
;
        and   [br:low intr_fac0],#00000100b     ;clear etm8 flag
        pop   ale
        rete
```

(2) 2 Hz clock timer interrupt processing

```
;************************************************************************
;*                                                                      *
;*      clock timer 2hz interrupt processing routine                    *
;*                                                                      *
;************************************************************************
clock2_intr:
        push  ale
;
        ld    br,#br_io                         ;set br reg. address to 0ffxxh
;
;
;
;       clock timer 2hz processing routine
;
;
        and   [br:low intr_fac0],#00000010b     ;clear etm2 flag
        pop   ale
        rete
```

(2)

(2) 1 Hz clock timer interrupt processing

```
;************************************************************************
;*                                                                      *
;*      clock timer 1hz interrupt processing routine                    *
;*                                                                      *
;************************************************************************
clock1_intr:
        push  ale
;
        ld    br,#br_io                         ;set br reg. address to 0ffxxh
;
;
;
;       clock timer 1hz processing routine
;
;
        and   [br:low intr_fac0],#00000001b     ;clear etm1 flag
        pop   ale
        rete

        end
```

11 STOPWATCH TIMER

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF42	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	–	–	–	–	–		
	D2	–	–	–	–	–		
	D1	SWRST	Stopwatch timer reset	Reset	No operation	–	W	
	D0	SWRUN	Stopwatch timer Run/Stop control	Run	Stop	0	R/W	
00FF43	D7	SWD7	Stopwatch timer data			0	R	
	D6	SWD6						
	D5	SWD5	BCD (1/10 sec)					
	D4	SWD4						
	D3	SWD3	Stopwatch timer data					
	D2	SWD2						
	D1	SWD1	BCD (1/100 sec)					
	D0	SWD0						
00FF20	D7	PK01	K00–K07 interrupt priority register	PK01 PK00 PSIF1 PSIF0 PSW1 PSW0 Priority PTM1 PTM0 level 1 1 Level 3 1 0 Level 2 0 1 Level 1 0 0 Level 0		0	R/W	
	D6	PK00						
	D5	PSIF1	Serial interface interrupt priority register			0	R/W	
	D4	PSIF0						
	D3	PSW1	Stopwatch timer interrupt priority register			0	R/W	
	D2	PSW0						
	D1	PTM1	Clock timer interrupt priority register			0	R/W	
	D0	PTM0						
00FF22	D7	–	–	–	–	–		"0" when being read
	D6	ESW100	Stopwatch timer 100 Hz interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D5	ESW10	Stopwatch timer 10 Hz interrupt enable register					
	D4	ESW1	Stopwatch timer 1 Hz interrupt enable register					
	D3	ETM32	Clock timer 32 Hz interrupt enable register					
	D2	ETM8	Clock timer 8 Hz interrupt enable register					
	D1	ETM2	Clock timer 2 Hz interrupt enable register					
	D0	ETM1	Clock timer 1 Hz interrupt enable register					
00FF24	D7	–	–	–	–	–		"0" when being read
	D6	FSW100	Stopwatch timer 100 Hz interrupt factor flag	(R) Interrupt factor is generated (W) Reset	(R) No interrupt factor is generated (W) No operation	0	R/W	
	D5	FSW10	Stopwatch timer 10 Hz interrupt factor flag					
	D4	FSW1	Stopwatch timer 1 Hz interrupt factor flag					
	D3	FTM32	Clock timer 32 Hz interrupt factor flag					
	D2	FTM8	Clock timer 8 Hz interrupt factor flag					
	D1	FTM2	Clock timer 2 Hz interrupt factor flag					
	D0	FTM1	Clock timer 1 Hz interrupt factor flag					

Specifications

Control of stopwatch timer

Vector address setting for stopwatch timer interrupt

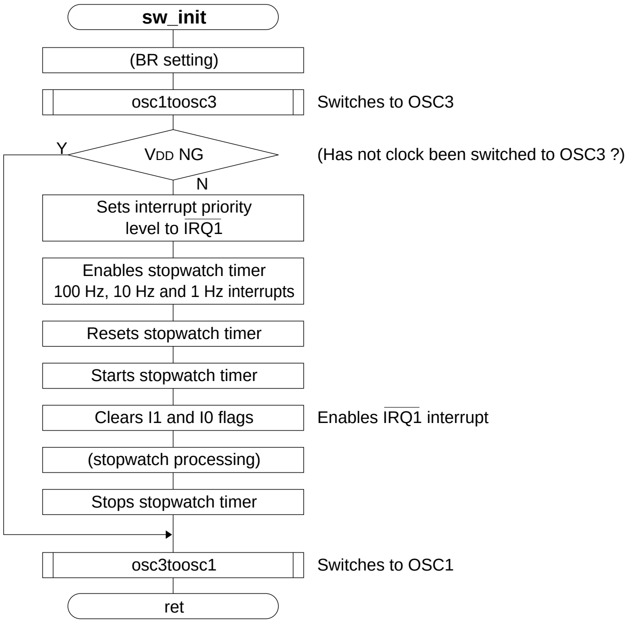
(1) sw_init: Initialization for stopwatch timer

Enables the respective 100 Hz, 10 Hz and 1 Hz interrupts of the stopwatch timer, clears the timer data and starts the stopwatch timer. The interrupt level has been set at $\overline{\text{IRQ1}}$.

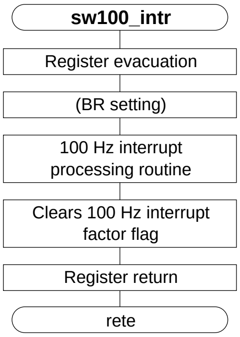
(2) sw100_intr, sw10_intr, sw1_intr: Stopwatch timer interrupt processing

Flowchart

(1) Initialization for stopwatch timer

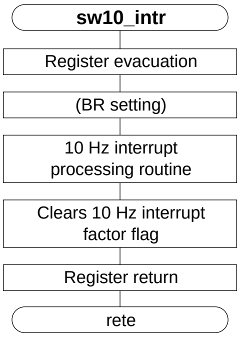


(2) 100 Hz stopwatch timer interrupt processing



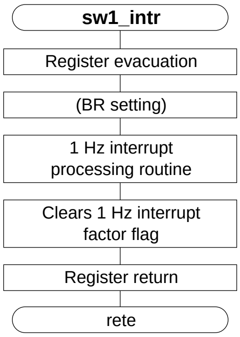
sw100_intr
Register evacuation
(BR setting)
100 Hz interrupt processing routine
Clears 100 Hz interrupt factor flag
Register return
rete

(2) 10 Hz stopwatch timer interrupt processing



sw10_intr
Register evacuation
(BR setting)
10 Hz interrupt processing routine
Clears 10 Hz interrupt factor flag
Register return
rete

(2) 1 Hz stopwatch timer interrupt processing



sw1_intr
Register evacuation
(BR setting)
1 Hz interrupt processing routine
Clears 1 Hz interrupt factor flag
Register return
rete

Notes

(1) External routines are called for switching to OSC3 and OSC1. (external call: osc1toosc3, osc3toosc1)

(2) Switching the operating mode when the supply voltage is lower than the V_{D1} setting may cause a malfunction. Hence, the example routine checks the supply voltage when switching to the normal mode (OSC3) and terminates as a supply voltage error remains unprocessed if the supply voltage is lower than the V_{D1} setting. For this determination, vdd_ngf flag is used. (See "4 OSCILLATION CIRCUIT".)

(3) When switching from OSC3 to OSC1 (V_{D1} = 2.2 V → 1.3 V), the program example does not perform special checking of the supply voltage of SVD if the supply voltage is already more than the V_{D1} setting.

(4) To reset the interrupt factor flag, write "1" into the corresponding flags alone, using the AND or LD instruction. When the OR logic operation instruction has been used, "1" is written for the interrupt factor flags that have been set to "1" within the same address and those flags are then clear.

(5) The interrupt flags (I1 and I0) have not been reset in the interrupt processing routine of this program example, so an interrupt lower than $\overline{\text{IRQ1}}$ level is disabled at the time of generation. When you wish to accept the next interrupt after an interrupt has been generated, re-setting of the interrupt flags or resetting the interrupt factor flag is necessary after due consideration for the nesting level.

(6) When stopping the stopwatch timer by writing "0" into the RUN/STOP control register for the stopwatch timer, the stopwatch timer count actually stops when it advances one count with the timing synchronized to the 256 Hz input clock.
For this reason, when the stopwatch timer stops, if the 100 Hz, 10 Hz, and 1 Hz interrupt factors are generated, the respective interrupt factor flags are set and if interrupt is enabled, an interrupt is generated. Thus, you should add an interrupt processing and interrupt factor flag resetting, if necessary.

(7) In this program example for the stopwatch timer, the vector address setting and program have been allocated from 003000H for the sake of convenience.

Source List

Control of stopwatch timer

```
        external     osc1toosc3,osc3toosc1
        external     vdd_ngf
        public       sw_init,sw100_intr,sw10_intr,sw1_intr
;
sw100_vector   equu   000016h                     ;sw100hz interrupt vector offset
sw10_vector    equ    000018h                     ;sw10hz interrupt vector offset
sw1_vector     equ    00001ah                     ;sw1hz interrupt vector offset
sw             equ    003000h                     ;program start address offset
br_io          equ    0ffh                        ;base reg. address (set i/o area)
sw_mode        equ    00ff42h                     ;stopwatch mode set reg.
swd            equ    00ff43h                     ;stopwatch data
intr_pr0       equ    00ff20h                     ;interrupt priority reg. 0
intr_en0       equ    00ff22h                     ;interrupt enable reg. 0
intr_fac0      equ    00ff24h                     ;interrupt factor flag reg.
;
        code
```

Source List

Vector address setting for stopwatch timer interrupt

```
intr_vectors:
         org   intr_vectors+sw100_vector
         dw    sw100_intr                        ;sw 100hz interrupt
;
         org   intr_vectors+sw10_vector
         dw    sw10_intr                         ;sw 10hz interrupt
;
         org   intr_vectors+sw1_vector
         dw    sw1_intr                          ;sw 1hz interrupt
;
```

(1) Initialization for stopwatch timer

```
         org   intr_vectors+sw
;*************************************************************************
;*                                                                       *
;*       stopwatch initialize (100,10 and 1hz interrupt enable)         *
;*                                                                       *
;*************************************************************************
;*** initialize routine
sw_init:
         ld    br,#br_io                         ;set br reg. address to 0ffxxh
         carl  osc1toosc3                        ;change osc1 to osc3 ***
         ld    a,[lod vdd_ngf]                   ;vdd ng flag
         cp    a,#0ffh
         jrl   z,sw_init00
;
         ld    a,[br:low intr_pr0]               ;interrupt priority reg.
         and   a,#11110011b
         or    a,#00000100b
         ld    [br:low intr_pr0],a               ;set sw=/irq1

;sw100,sw10 and sw1 (en. /irq1) intr.
         or    [br:low intr_en0],#01110000b                                      (1)
         or    [br:low sw_mode],#00000010b       ;stopwatch counter reset
         or    [br:low sw_mode],#00000001b       ;stopwatch start
         and   sc,#00111111b                     ;i1 and i0 flag clear
;*************************************************************************
;*** start stopwatch interrupt
;        (user program)
;
;*************************************************************************
;*** end processing
         and   [br:low sw_mode],#11111110b       ;stopwatch stop
sw_init00:
         carl  osc3toosc1                        ;change osc3 to osc1 ***
         ret
```

(2) 100 Hz stopwatch timer interrupt processing

```
;*************************************************************************
;*                                                                       *
;*       stopwatch 100hz interrupt processing routine                    *
;*                                                                       *
;*************************************************************************
sw100_intr:
         push  ale
;
         ld    br,#br_io                         ;set br reg. address to 0ffxxh
;
;

;                                                                                (2)
;        stopwatch 100hz processing routine
;
;
         and   [br:low intr_fac0],#01000000b     ;clear sw100 flag
         pop   ale
         rete
```

Source List

(2) 10 Hz stopwatch timer interrupt processing

```
;************************************************************************
;*                                                                      *
;*       stopwatch 10hz interrupt processing routine                    *
;*                                                                      *
;************************************************************************
sw10_intr:
         push  ale
;
         ld    br,#br_io                         ;set br reg. address to 0ffxxh
;
;
;
;        stopwatch 10hz processing routine
;
;
         and   [br:low intr_fac0],#00100000b     ;clear sw10 flag
         pop   ale
         rete
```

(2) 1 Hz stopwatch timer interrupt processing

(2)

```
;************************************************************************
;*                                                                      *
;*       stopwatch 1hz interrupt processing routine                     *
;*                                                                      *
;************************************************************************
sw1_intr:
         push  ale
;
         ld    br,#br_io                         ;set br reg. address to 0ffxxh
;
;
;
;        stopwatch 1hz processing routine
;
;
         and   [br:low intr_fac0],#00010000b     ;clear sw1 flag
         pop   ale
         rete
;
         end
```

12 PROGRAMMABLE TIMER

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF30	D7	–	–	–	–	–		Constantry "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	MODE16	8/16-bit mode selection	16-bit x 1	8-bit x 2	0	R/W	
	D3	CHSEL	TOUT output channel selection	Timer 1	Timer 0	0	R/W	
	D2	PTOUT	TOUT output control	On	Off	0	R/W	
	D1	CKSEL1	Prescaler 1 source clock selection	f_{OSC3}	f_{OSC1}	0	R/W	
	D0	CKSEL0	Prescaler 0 source clock selection	f_{OSC3}	f_{OSC1}	0	R/W	
00FF31	D7	EVCNT	Timer 0 counter mode selection	Event counter	Timer	0	R/W	
	D6	FCSEL	Timer 0 function selection – In timer mode	Pulse width measurement	Normal mode	0	R/W	
			In event counter mode	With noise rejector	Without noise rejector			
	D5	PLPOL	Timer 0 pulse polarity selection – Down count timing in event counter mode	Rising edge of K10 input	Falling edge of K10 input	0	R/W	
			In pulse width measurement mode	High level measurement for K10 input	Low level measurement for K10 input			
	D4	PSC01	Timer 0 prescaler dividing ratio selection PSC01 PSC00 Prescaler dividing ratio 1 1 Source clock / 64 1 0 Source clock / 16 0 1 Source clock / 4 0 0 Source clock / 1			0	R/W	
	D3	PSC00				0	R/W	
	D2	CONT0	Timer 0 continuous/one-shot mode selection	Continuous	One-shot	0	R/W	
	D1	PSET0	Timer 0 preset	Preset	No operation	–	W	"0" when being read
	D0	PRUN0	Timer 0 Run/Stop control	Run	Stop	0	R/W	
00FF32	D7	–	–	–	–	–		Constantry "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	PSC11	Timer 1 prescaler dividing ratio selection PSC11 PSC10 Prescaler dividing ratio 1 1 Source clock / 64 1 0 Source clock / 16 0 1 Source clock / 4 0 0 Source clock / 1			0	R/W	
	D3	PSC10				0	R/W	
	D2	CONT1	Timer 1 continuous/one-shot mode selection	Continuous	One-shot	0	R/W	
	D1	PSET1	Timer 1 preset	Preset	No operation	–	W	"0" when being read
	D0	PRUN1	Timer 1 Run/Stop control	Run	Stop	0	R/W	

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF33	D7	RLD07	Timer 0 reload data D7 (MSB)	High	Low	1	R/W	
	D6	RLD06	Timer 0 reload data D6					
	D5	RLD05	Timer 0 reload data D5					
	D4	RLD04	Timer 0 reload data D4					
	D3	RLD03	Timer 0 reload data D3					
	D2	RLD02	Timer 0 reload data D2					
	D1	RLD01	Timer 0 reload data D1					
	D0	RLD00	Timer 0 reload data D0 (LSB)					
00FF34	D7	RLD17	Timer 1 reload data D7 (MSB)	High	Low	1	R/W	
	D6	RLD16	Timer 1 reload data D6					
	D5	RLD15	Timer 1 reload data D5					
	D4	RLD14	Timer 1 reload data D4					
	D3	RLD13	Timer 1 reload data D3					
	D2	RLD12	Timer 1 reload data D2					
	D1	RLD11	Timer 1 reload data D1					
	D0	RLD10	Timer 1 reload data D0 (LSB)					
00FF35	D7	PTD07	Timer 0 counter data D7 (MSB)	High	Low	1	R	
	D6	PTD06	Timer 0 counter data D6					
	D5	PTD05	Timer 0 counter data D5					
	D4	PTD04	Timer 0 counter data D4					
	D3	PTD03	Timer 0 counter data D3					
	D2	PTD02	Timer 0 counter data D2					
	D1	PTD01	Timer 0 counter data D1					
	D0	PTD00	Timer 0 counter data D0 (LSB)					
00FF36	D7	PTD17	Timer 1 counter data D7 (MSB)	High	Low	1	R	
	D6	PTD16	Timer 1 counter data D6					
	D5	PTD15	Timer 1 counter data D5					
	D4	PTD14	Timer 1 counter data D4					
	D3	PTD13	Timer 1 counter data D3					
	D2	PTD12	Timer 1 counter data D2					
	D1	PTD11	Timer 1 counter data D1					
	D0	PTD10	Timer 1 counter data D0 (LSB)					

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF21	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	PPT1	Programmable timer interrupt priority register	PPT1 PPT0 PK11 PK10 1 1 1 0 0 1 0 0	Priority level Level 3 Level 2 Level 1 Level 0	0	R/W	
	D2	PPT0						
	D1	PK11	K10 and K11 interrupt priority register			0	R/W	
	D0	PK10						
00FF23	D7	EPT1	Programmable timer 1 interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D6	EPT0	Programmable timer 0 interrupt enable register					
	D5	EK1	K10 and K11 interrupt enable register					
	D4	EK0H	K04–K07 interrupt enable register					
	D3	EK0L	K00–K03 interrupt enable register					
	D2	ESERR	Serial I/F (error) interrupt enable register					
	D1	ESREC	Serial I/F (receiving) interrupt enable register					
	D0	ESTRA	Serial I/F (transmitting) interrupt enable register					
00FF25	D7	FPT1	Programmable timer 1 interrupt factor flag	(R) Interrupt factor is generated (W) Reset	(R) No interrupt factor is generated (W) No operation	0	R/W	
	D6	FPT0	Programmable timer 0 interrupt factor flag					
	D5	FK1	K10 and K11 interrupt factor flag					
	D4	FK0H	K04–K07 interrupt factor flag					
	D3	FK0L	K00–K03 interrupt factor flag					
	D2	FSERR	Serial I/F (error) interrupt factor flag					
	D1	FSREC	Serial I/F (receiving) interrupt factor flag					
	D0	FSTRA	Serial I/F (transmitting) interrupt factor flag					

Specifications

Control of programmable timer

(1) timer2ch_init, pt1_intr, pt0_intr: Vector address setting for programmable timer interrupt Initialization and interrupt processing for 8-bit reload timer (two channels)

This is an example of using the programmable timer as an 8-bit x 2 system and performs the following settings:

- Count mode 8-bit x 2
- Pulse output channel Timer 0 ⎤ (Invalid)
- Pulse external (TOUT) output OFF ⎦

<Timer 0>

- Timer mode Programmable timer (reload mode)
- Count clock fOSC3 x 1/16
- Reload data 200 (= 800 μsec, when fOSC3 is 4 MHz)

<Timer 1>

- Timer mode Programmable timer (reload mode)
- Count clock fOSC3 x 1/64
- Reload data 250 (= 1 msec, when fOSC3 is 4 MHz)

After setting the above, it enables the timer 1 and timer 0 interrupts, and starts each timer. The interrupt level has been set at $\overline{\text{IRQ3}}$ and the respective interrupts are generated in the cycles according to the reload data.

Specifications

(2) timer1ch_init, pt0_intr: Vector address setting for programmable timer interrupt Initialization and interrupt processing for 16-bit one-shot timer (one channel)

This is an example of using the programmable timer as a 16-bit x 1 system one-shot timer and performs the following settings:

• Count mode	16-bit x 1	
• Pulse output channel	Timer 0	(Invalid)
• Pulse external (TOUT) output	OFF	

<Timer 0>

• Timer mode	Programmable timer (one-shot mode)
• Count clock	f_{OSC3} x 1/4
• Reload data	33,200 (= 33.2 msec, when f_{OSC3} is 4 MHz)

<Timer 1> Cannot be used

After setting the above, it enables the timer 1 interrupt, and starts the timer.

The interrupt level has been set at $\overline{\text{IRQ3}}$ and an interrupt is generated 33.2 msec after starting.

(3) evcnt_init, pt1_intr, evcnt_intr: Vector address setting for programmable timer interrupt Initialization and interrupt processing for 8-bit event counter

This is an example of using the programmable timer as an 8-bit event counter and 8-bit reload timer, and performs the following settings:

• Count mode	8-bit x 2	
• Pulse output channel	Timer 0	(Invalid)
• Pulse external (TOUT) output	OFF	

<Timer 0>

• Timer mode	Event counter (reload mode)
• Input clock	K10 with noise rejector
• Count timing	Falling edge
• Reload data	0FFH (Event counter initial value)

<Timer 1>

• Timer mode	Programmable timer (reload mode)
• Count clock	f_{OSC3} x 1/64
• Reload data	250 (= 4 msec, when f_{OSC3} is 4 MHz)

After setting the above, it enables the the event counter and timer 1 interrupts, and starts each timer.

The interrupt level has been set at $\overline{\text{IRQ3}}$ and an interrupt is generated by the overflow of the event counter or timer 1.

Timer 1 is programmed to generate an interrupt in 4 msec cycles. This example reads the event counter data in the interrupt processing routine and calculates the difference between it and previous count value. This difference is made to the number of clocks that had been input in the 4 msec period.

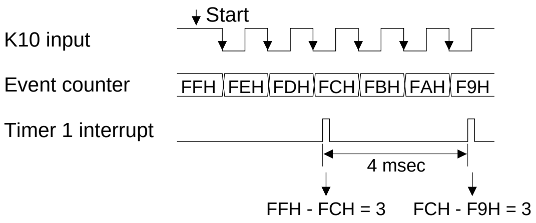


Fig. 12.1 Event counter processing

Specifications

(4) measure_init, measure_intr, k1x_intr: Vector address setting for programmable timer interrupt Initialization and interrupt processing for 16-bit pulse width measurement timer

This is an example of using the programmable timer as a 16-bit pulse width measurement timer and performs the following settings:

• Count mode	16-bit x 1	
• Pulse output channel	Timer 0	(Invalid)
• Pulse external (TOUT) output	OFF	

<Timer 0>

• Timer mode	Pulse width measurement timer (reload mode)
• Measurement period	During LOW input
• Count clock	f_{OSC3} x 1/1
• Reload data	0FFFFH (Pulse width measurement timer initial value)
<Timer 1>	Cannot be used

After setting the above, it enables the timer 1 and K10 input interrupts, and starts the timer. The interrupt level has been set at $\overline{\text{IRQ3}}$ and an interrupt is generated by the overflow of the timer or K10 input.

Since the fall (count start) and rise (count completion) timings of the K10 input cannot be evaluated by programmable timer control only, a K10 input interrupt is used. Furthermore, in order to be able to generate an interrupt at both falling and rising timings, input interrupt timing is reversed by each interrupt generation in the K1x interrupt processing routine.

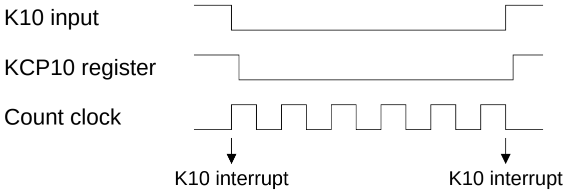


Fig. 12.2 Timing of K10 input interrupt generation

(5) pulsout_init: Vector address setting for programmable timer interrupt 16-bit reload timer pulse output

Outputs TOUT signal from the R27 terminal using the programmable timer as 16-bit reload timer.

• Count mode	16-bit x 1	
• Pulse output channel	Timer 0	(Valid)
• Pulse external (TOUT) output	ON	

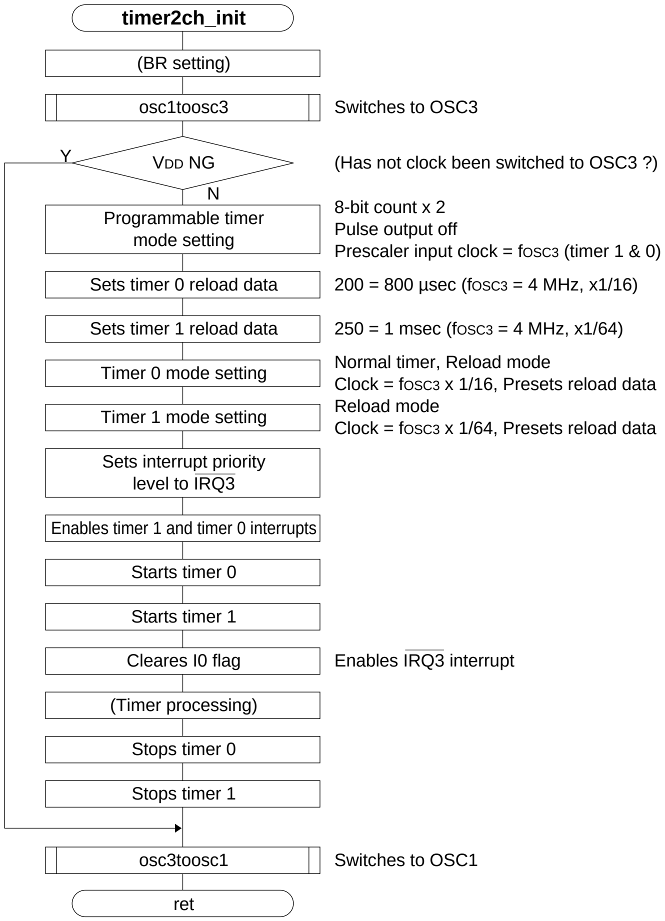
<Timer 0>

• Timer mode	Programmable timer (reload mode)
• Count clock	f_{OSC3} x 1/4
• Reload data	33,200 (= 33.2 msec, when f_{OSC3} is 4 MHz)
<Timer 1>	Cannot be used

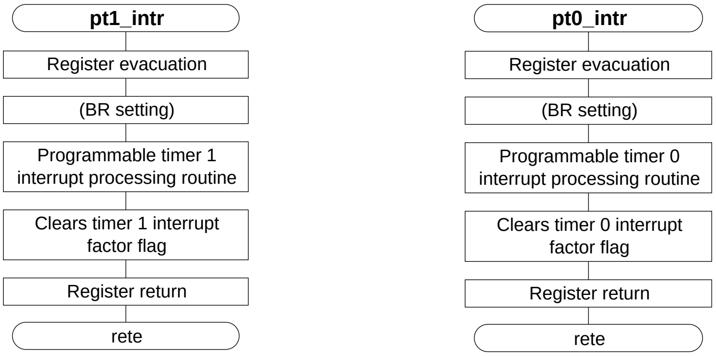
According to the above setting, the clock cycle of the TOUT signal is set at 66.4 msec (approximately 15 Hz).

Flowchart

(1) Initialization for 8-bit reload timer (two channels)

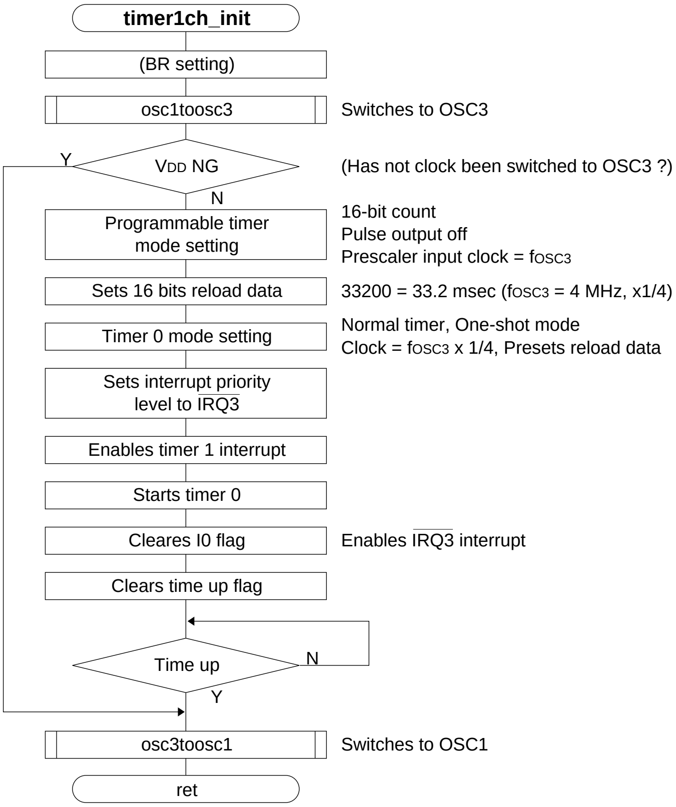

timer2ch_init
(BR setting)
osc1toosc3
Switches to OSC3
VDD NG
Y
N
(Has not clock been switched to OSC3 ?)
Programmable timer mode setting
8-bit count x 2
Pulse output off
Prescaler input clock = fOSC3 (timer 1 & 0)
Sets timer 0 reload data
200 = 800 µsec (fOSC3 = 4 MHz, x1/16)
Sets timer 1 reload data
250 = 1 msec (fOSC3 = 4 MHz, x1/64)
Timer 0 mode setting
Normal timer, Reload mode
Clock = fOSC3 x 1/16, Presets reload data
Timer 1 mode setting
Reload mode
Clock = fOSC3 x 1/64, Presets reload data
Sets interrupt priority level to IRQ3
Enables timer 1 and timer 0 interrupts
Starts timer 0
Starts timer 1
Cleares I0 flag
Enables IRQ3 interrupt
(Timer processing)
Stops timer 0
Stops timer 1
osc3toosc1
Switches to OSC1
ret

(1) Interrupt processing for 8-bit reload timer (two channels)

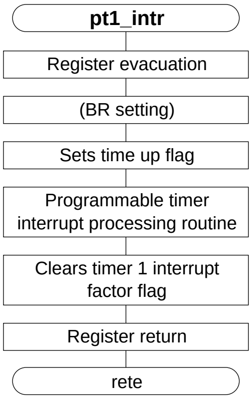

pt1_intr
Register evacuation
(BR setting)
Programmable timer 1 interrupt processing routine
Clears timer 1 interrupt factor flag
Register return
rete
pt0_intr
Register evacuation
(BR setting)
Programmable timer 0 interrupt processing routine
Clears timer 0 interrupt factor flag
Register return
rete

Flowchart

(2) Initialization for 16-bit one-shot timer (one channel)

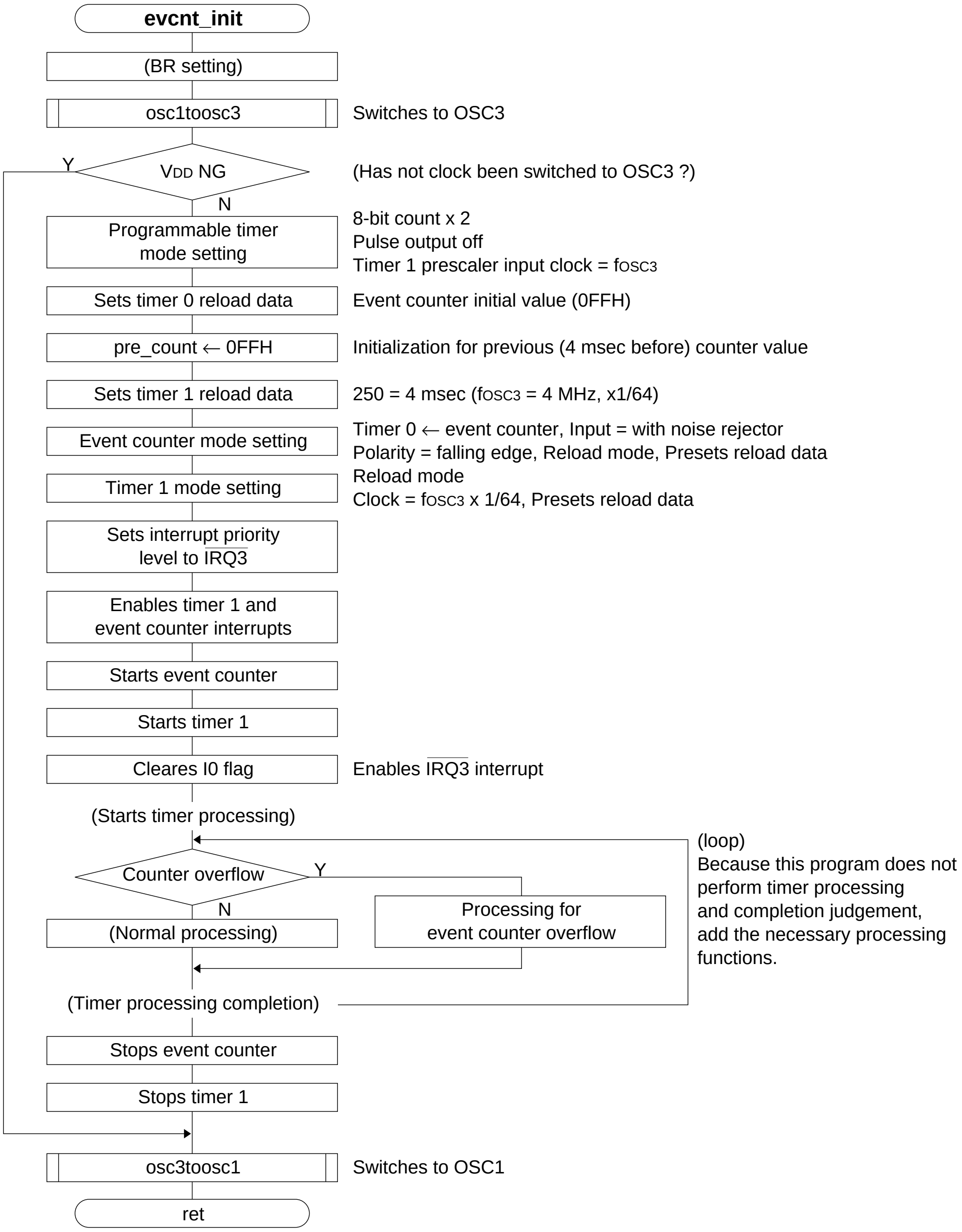

timer1ch_init
(BR setting)
osc1toosc3
Switches to OSC3
VDD NG
Y
N
(Has not clock been switched to OSC3 ?)
Programmable timer mode setting
16-bit count
Pulse output off
Prescaler input clock = fOSC3
Sets 16 bits reload data
33200 = 33.2 msec (fOSC3 = 4 MHz, x1/4)
Timer 0 mode setting
Normal timer, One-shot mode
Clock = fOSC3 x 1/4, Presets reload data
Sets interrupt priority level to IRQ3
Enables timer 1 interrupt
Starts timer 0
Cleares I0 flag
Enables IRQ3 interrupt
Clears time up flag
Time up
N
Y
osc3toosc1
Switches to OSC1
ret

(2) Interrupt processing for 16-bit one-shot timer (one channel)


pt1_intr
Register evacuation
(BR setting)
Sets time up flag
Programmable timer interrupt processing routine
Clears timer 1 interrupt factor flag
Register return
rete

Flowchart

(3) Initialization for 8-bit event counter



Switches to OSC3

(Has not clock been switched to OSC3 ?)

8-bit count x 2
Pulse output off
Timer 1 prescaler input clock = f_{OSC3}

Event counter initial value (0FFH)

Initialization for previous (4 msec before) counter value

250 = 4 msec (f_{OSC3} = 4 MHz, x1/64)

Timer 0 ← event counter, Input = with noise rejector
Polarity = falling edge, Reload mode, Presets reload data

Reload mode
Clock = f_{OSC3} x 1/64, Presets reload data

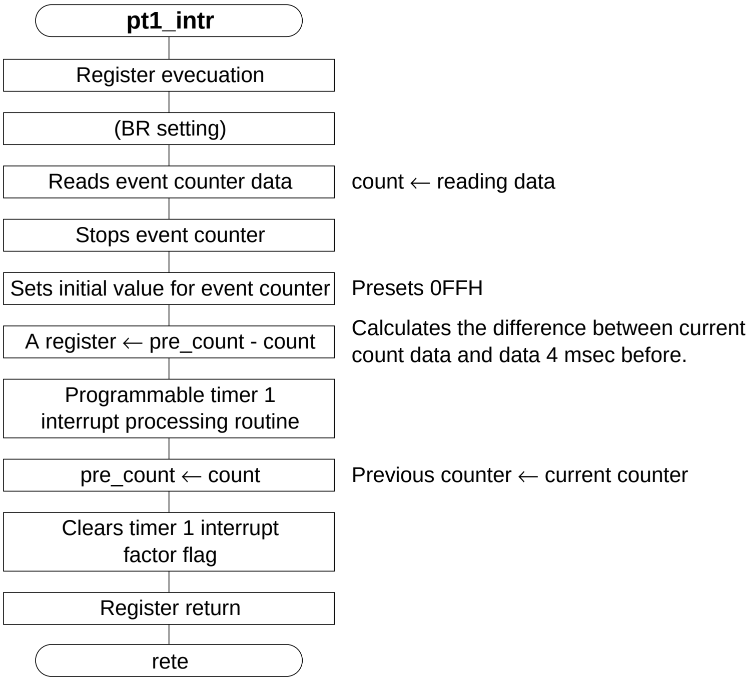
Enables $\overline{\text{IRQ3}}$ interrupt

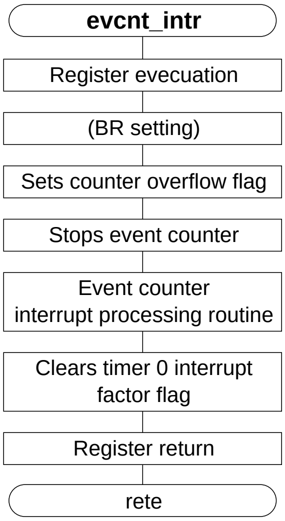
(loop)
Because this program does not perform timer processing and completion judgement, add the necessary processing functions.

Switches to OSC1

Flowchart

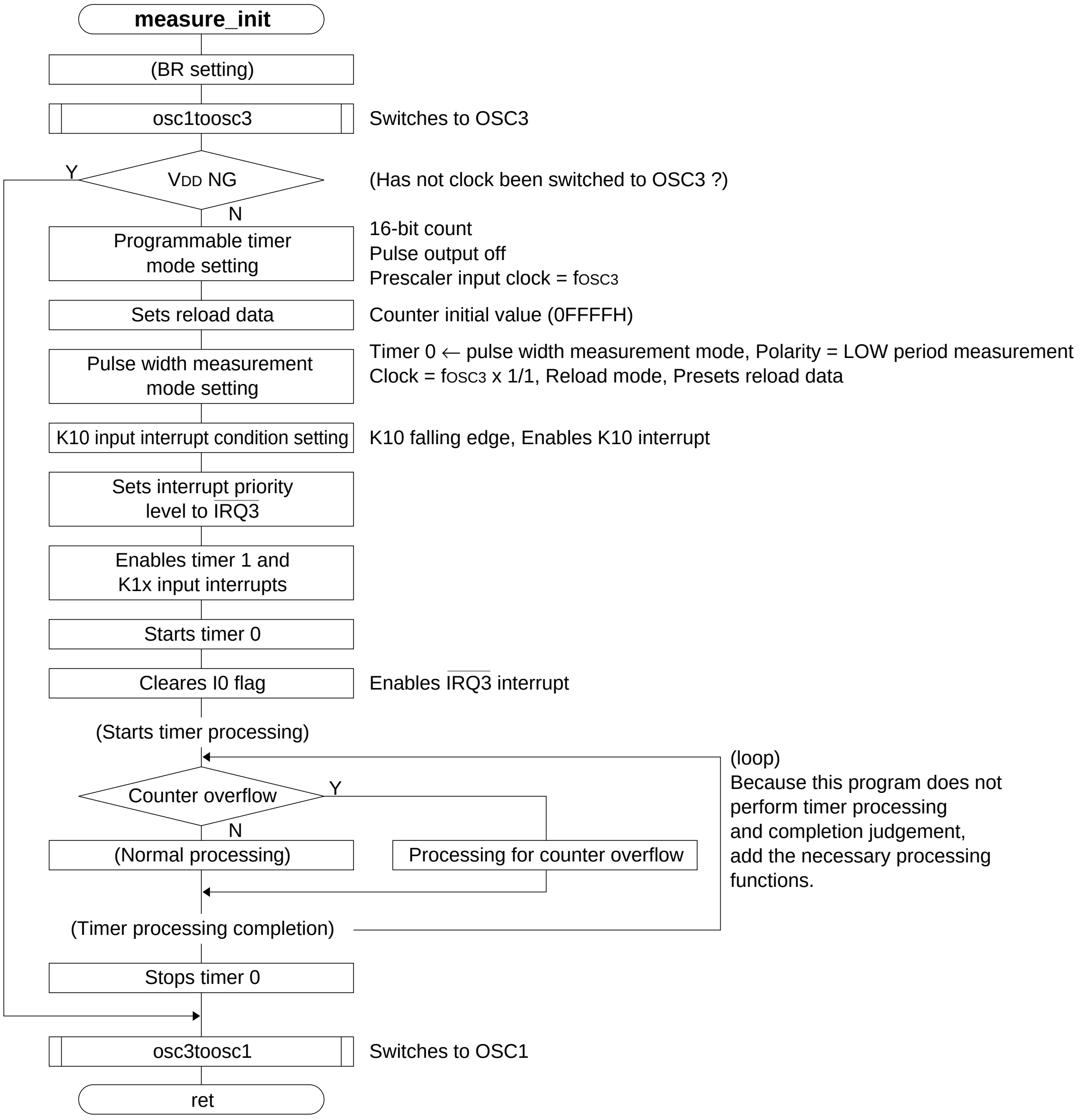
(3) Interrupt processing for 8-bit event counter


pt1_intr
Register evecuation
(BR setting)
Reads event counter data
count ← reading data
Stops event counter
Sets initial value for event counter
Presets 0FFH
A register ← pre_count - count
Calculates the difference between current count data and data 4 msec before.
Programmable timer 1 interrupt processing routine
pre_count ← count
Previous counter ← current counter
Clears timer 1 interrupt factor flag
Register return
rete


evcnt_intr
Register evecuation
(BR setting)
Sets counter overflow flag
Stops event counter
Event counter interrupt processing routine
Clears timer 0 interrupt factor flag
Register return
rete

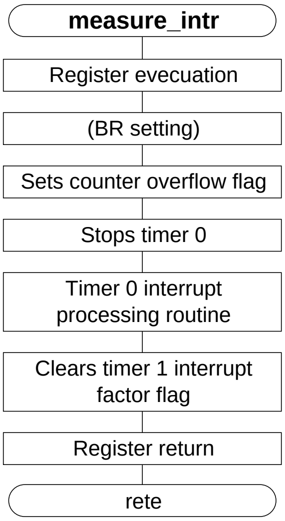
Flowchart

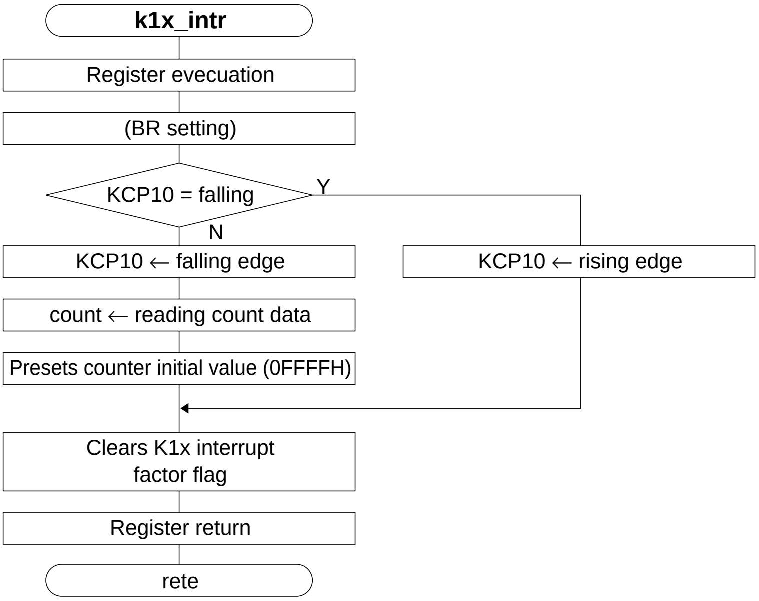
(4) Initialization for 16-bit pulse width measurement timer


measure_init
(BR setting)
osc1toosc3
Switches to OSC3
Y
VDD NG
N
(Has not clock been switched to OSC3 ?)
Programmable timer mode setting
16-bit count
Pulse output off
Prescaler input clock = fOSC3
Sets reload data
Counter initial value (0FFFFH)
Pulse width measurement mode setting
Timer 0 ← pulse width measurement mode, Polarity = LOW period measurement
Clock = fOSC3 x 1/1, Reload mode, Presets reload data
K10 input interrupt condition setting
K10 falling edge, Enables K10 interrupt
Sets interrupt priority level to IRQ3
Enables timer 1 and K1x input interrupts
Starts timer 0
Cleares I0 flag
Enables IRQ3 interrupt
(Starts timer processing)
Counter overflow
Y
N
(Normal processing)
Processing for counter overflow
(loop)
Because this program does not perform timer processing and completion judgement, add the necessary processing functions.
(Timer processing completion)
Stops timer 0
osc3toosc1
Switches to OSC1
ret

Flowchart

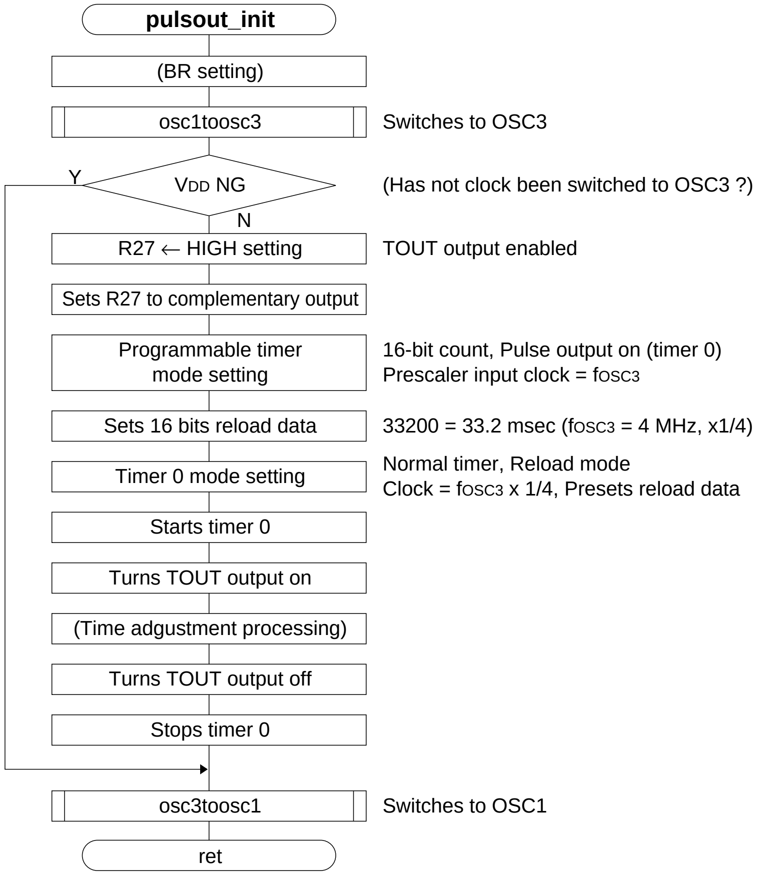
(4) Interrupt processing for 16-bit pulse width measurement timer


measure_intr
Register evecuation
(BR setting)
Sets counter overflow flag
Stops timer 0
Timer 0 interrupt processing routine
Clears timer 1 interrupt factor flag
Register return
rete



Flowchart

(5) 16-bit reload timer pulse output


pulsout_init
(BR setting)
osc1toosc3
Switches to OSC3
VDD NG
Y
N
(Has not clock been switched to OSC3 ?)
R27 ← HIGH setting
TOUT output enabled
Sets R27 to complementary output
Programmable timer mode setting
16-bit count, Pulse output on (timer 0)
Prescaler input clock = fOSC3
Sets 16 bits reload data
33200 = 33.2 msec (fOSC3 = 4 MHz, x1/4)
Timer 0 mode setting
Normal timer, Reload mode
Clock = fOSC3 x 1/4, Presets reload data
Starts timer 0
Turns TOUT output on
(Time adgustment processing)
Turns TOUT output off
Stops timer 0
osc3toosc1
Switches to OSC1
ret

Notes

(1) External routines are called for switching to OSC3 and OSC1. (external call: osc1toosc3, osc3toosc1)

(2) Switching the operating mode when the supply voltage is lower than the VD1 setting may cause a malfunction. Hence, the example routine checks the supply voltage when switching to the normal mode (OSC3) and terminates as a supply voltage error remains unprocessed if the supply voltage is lower than the VD1 setting. For this determination, vdd_ngf flag is used. (See "4 OSCILLATION CIRCUIT".)

(3) When switching from OSC3 to OSC1 (VD1 = 2.2 V → 1.3 V), the program example does not perform special checking of the supply voltage of SVD if the supply voltage is already more than the VD1 setting.

(4) To reset the interrupt factor flag, write "1" into the corresponding flags alone, using the AND or LD instruction. When the OR logic operation instruction has been used, "1" is written for the interrupt factor flags that have been set to "1" within the same address and those flags are then clear.

Notes

(5) The interrupt flags (I1 and I0) have not been reset in the interrupt processing routine of this program example, so an interrupt lower than $\overline{\text{IRQ3}}$ level is disabled at the time of generation. When you wish to accept the next interrupt after an interrupt has been generated, re-setting of the interrupt flags or resetting the interrupt factor flag is necessary after due consideration for the nesting level.

(6) The R27 terminal is the common terminal for the normal DC output port and the TOUT output. When TOUT is being output, set R27 register to "1" and control the signal ON/OFF using the TOUT register.

(7) When the pulse output control is set to off ("0"), the setting of the pulse output channel selection becomes invalid.

(8) When programmable timer 1 is selected as the clock source for the serial interface, pay attention to the setting value for timer 1, the mode selection for timer 1 and the interrupt setting. Be advised that in this case, it is impossible to use it as a 16-bit timer coupling both timer 0 and timer 1.

(9) When coupling programmable timers 0 and 1 for use as a 16-bit timer, the setting of timer 0 becomes valid for timer operation and the setting of timer 1 becomes invalid. However, since an interrupt is generated by the underflow of timer 1, set the interrupt related routine with timer 1.

(10) When stopping the programmable timer by writing "0" into the RUN/STOP control register for the programmable timer 0 and 1, the programmable timer count actually stops when it advances one count with the timing synchronized to the input clock selected with the prescaler dividing clock. For this reason, when the programmable timer stops, if the respective interrupt factors are generated, the respective interrupt factor flags are set and if interrupt is enabled, an interrupt is generated. Thus, you should add an interrupt processing and interrupt factor flag resetting, if necessary.

(11) A noise reject circuit is not included in the input port (K port). For this reason, when the programmable timer is used for event counter in the program example (3) or for pulse width measurement in the example (4), the following operation will occur if there is chattering in the K10 input, so, input waveform shaping or adding external noise reject processing with an external circuit and software is necessary. In case of the event counter in the program example (3), if there is chattering in the K10 input, the chattering may be counted. In the case of pulse width measurement in the program example (4), if there is chattering in the K10 input, successive interrupts may be generated in the measurement start trigger timing of the rising or falling K10 input.

(12) When a down-counter underflow occurs, the one-shot timer mode sets the reload register value to the counter data register, to stop the count. For this reason, when you want to continue the count at the same count number, you should restart to timer. If you want to newly set a different count number, set the new value in the reload register, then set it to the count data register, and then start the timer.

(13) In the examples of programmable timer control programs which use an interrupt, the vector address setting and program have been allocated from 003000H for the sake of convenience. For an example which does not use an interrupt, a specific address has not been allocated as in the examples in other chapters.

Source List

Control of programmable timer 1

```
         external    osc1toosc3,osc3toosc1
         external    vdd_ngf
         public      timer2ch_init,pt1_intr,pt0_intr
;
pt1_vector     equ   000006h                     ;timer 0 interrupt vector offset
pt0_vector     equ   000008h                     ;timer 1 interrupt vector offset
pt             equ   003000h                     ;program start address offset
br_io          equ   0ffh                        ;base reg. address (set i/o area)
pt_mode0       equ   00ff30h                     ;programmable timer mode set reg. 0
pt_mode1       equ   00ff31h                     ;programmable timer mode set reg. 1
pt_mode2       equ   00ff32h                     ;programmable timer mode set reg. 2
rld0           equ   00ff33h                     ;programmable timer 0 reload data
rld1           equ   00ff34h                     ;programmable timer 1 reload data
;
intr_pr1       equ   00ff21h                     ;interrupt priority reg. 1
intr_en1       equ   00ff23h                     ;interrupt enable reg. 1
intr_fac1      equ   00ff25h                     ;interrupt factor flag reg. 1
;
         code
```

Vector address setting for programmable timer interrupt

```
intr_vectors:
         org   intr_vectors+pt1_vector
         dw    pt1_intr                          ;programmable timer 1 interrupt
;
         org   intr_vectors+pt0_vector
         dw    pt0_intr                          ;programmable timer 0 interrupt
;
```

(1) Initialization for 8-bit reload timer (two systems)

```
         org   intr_vectors+pt
timerdata8_0:  db    200                         ;timer 0 reload data (800us at 4mhz/16)
timerdata8_1:  db    250                         ;timer 1 reload data (  1ms at 4mhz/64)
;
;************************************************************************
;*                                                                      *
;*       8-bit * 2-channel reload timer                                 *
;*                                                                      *
;************************************************************************
;*** initialize routine
timer2ch_init:
         ld    br,#br_io                         ;set br reg. address to 0ffxxh
         carl  osc1toosc3                        ;change osc1 to osc3 ***
         ld    a,[lod vdd_ngf]                   ;vdd ng flag
         cp    a,#0ffh
         jrl   z,timer2ch_init00
;
;mode16=8bit*2,chsel=timer0,ptout=off,cksel1&0=fosc3
         ld    [br:low pt_mode0],#00000011b
         ld    a,[loc timerdata8_0]
         ld    [br:low rld0],a                   ;set reload data (timer 0)
         ld    a,[loc timerdata8_1]
         ld    [br:low rld1],a                   ;set reload data (timer 1)
;pt0:evcnt=timer,fcsel=normal timer,plpol=don't care,psc=fosc3/16,rlmd0=reload
;pset0=preset
         ld    [br:low pt_mode1],#00010110b
;pt1:psc=fosc3/64,rlmd1=reload,pset1=preset
         ld    [br:low pt_mode2],#00011110b
         or    [br:low intr_pr1],#00001100b      ;set pt=/irq3
```

Source List

```
         or    [br:low intr_en1],#11000000b      ;ept1&0 intr. en.
         or    [br:low pt_mode1],#00000001b      ;start timer 0
         or    [br:low pt_mode2],#00000001b      ;start timer 1
         ld    a,sc
         and   a,#00111111b
         or    a,#10000000b
         ld    sc,a                              ;i0 flag clear (en. /irq3 intr.)
;*************************************************************************
;*** start programmable timer 0 & 1 interrupt
;        (user program)
;
;*************************************************************************
;*** end processing
         and   [br:low pt_mode1],#11111110b      ;stop timer 0
         and   [br:low pt_mode2],#11111110b      ;stop timer 1
timer2ch_init00:
         carl  osc3toosc1                        ;change osc3 to osc1 ***
         ret
```

(1) Interrupt processing for 8-bit reload timer (two systems)

```
;*************************************************************************
;*                                                                       *
;*    programmable timer 1 interrupt processing routine (reload mode)   *
;*                                                                       *
;*************************************************************************
pt1_intr:
         push  ale
;
         ld    br,#br_io                         ;set br reg. address to 0ffxxh
;
;
;        programmable timer 1 processing
;
;
         and   [br:low intr_fac1],#10000000b     ;clear fpt1 interrupt flag
         pop   ale
         rete
;*************************************************************************
;*                                                                       *
;*    programmable timer 0 interrupt processing routine (reload mode)   *
;*                                                                       *
;*************************************************************************
pt0_intr:
         push  ale
;
         ld    br,#br_io                         ;set br reg. address to 0ffxxh
;
;
;        programmable timer 0 processing
;
;
         and   [br:low intr_fac1],#01000000b     ;clear fpt0 interrupt flag
         pop   ale
         rete
         end
```

Source List

Control of programmable timer 2

```
         external     osc1toosc3,osc3toosc1
         external     vdd_ngf
         public       timer1ch_init,pt0_intr
         public       timeup
;
pt1_vector     equ   000006h                     ;timer 1 interrupt vector offset
pt             equ   003000h                     ;program start address offset
br_io          equ   0ffh                        ;base reg. address (set i/o area)
pt_mode0       equ   00ff30h                     ;programmable timer mode set reg. 0
pt_mode1       equ   00ff31h                     ;programmable timer mode set reg. 1
rld0           equ   00ff33h                     ;programmable timer 0 reload data
rld1           equ   00ff34h                     ;programmable timer 1 reload data
;
intr_pr1       equ   00ff21h                     ;interrupt priority reg. 1
intr_en1       equ   00ff23h                     ;interrupt enable reg. 1
intr_fac1      equ   00ff25h                     ;interrupt factor flag reg. 1
;
         data
timeup:        db    [1]                         ;timeup flag
;
         code
```

Vector address setting for programmable timer interrupt

```
intr_vectors:
         org   intr_vectors+pt1_vector
         dw    pt1_intr                          ;programmable timer 0 interrupt
;
```

(2) Initialization for 16-bit one-shot timer (one system)

```
         org   intr_vectors+pt
timerdata16:   dw    33200                       ;timer16 reload data (33.2ms at 4mhz/4)
;************************************************************************
;*                                                                      *
;*       16-bit * 1-channel one shot timer                              *
;*                                                                      *
;************************************************************************
;*** initialize routine
timer1ch_init:
         ld    br,#br_io                         ;set br reg. address to 0ffxxh
         carl  osc1toosc3                        ;change osc1 to osc3 ***
         ld    a,[lod vdd_ngf]                   ;vdd ng flag
         cp    a,#0ffh
         jrl   z,timer1ch_init00
;
;mode16=16-bit,chsel=timer0,ptout=off,cksel1=dont't care,ckse0=fosc3
         ld    [br:low pt_mode0],#00010001b
         ld    ba,[loc timerdata16]              ;set 16-bit reload data (timer 0 & 1)
         ld    [lod rld0],ba
;pt0:evcnt=timer,fcsel=normal timer,plpol=don't care,psc=fosc3/4,rlmd0=oneshot
;pset0=preset
         ld    [br:low pt_mode1],#00001010b
         or    [br:low intr_pr1],#00001100b      ;set pt=/irq3
         or    [br:low intr_en1],#10000000b      ;ept1 intr. en.
         or    [br:low pt_mode1],#00000001b      ;start timer 0
         ld    a,sc
         and   a,#00111111b
         or    a,#10000000b
         ld    sc,a                              ;i0 flag clear (en. /irq3 intr.)
         xor   a,a
```

Source List

```
         ld    [lod timeup],a
;************************************************************************
;*** start programmable timer 0 (16-bit) interrupt
timer1ch_init01:
         ld    a,[lod timeup]
         cp    a,#0ffh
         jrs   nz,timer1ch_init01
;************************************************************************
;*** end processing
timer1ch_init00:
         carl  osc3toosc1                        ;change osc3 to osc1 ***
         ret
```

(2) Interrupt processing for 16-bit one-shot timer (one system)

```
;************************************************************************
;*                                                                      *
;*   programmable timer 1 interrupt processing routine (one-shot mode)  *
;*                                                                      *
;************************************************************************
pt0_intr:
         push  ale
;
         ld    br,#br_io                         ;set br reg. address to 0ffxxh
         ld    a,#0ffh
         ld    [lod timeup],a                    ;timeup flag set
;
;
;        programmable timer 0 processing
;
;
         and   [br:low intr_fac1],#10000000b     ;clear fpt1 interrupt flag
         pop   ale
         rete
         end
```

Source List

Control of programmable timer 3

```
         external    osc1toosc3,osc3toosc1
         external    vdd_ngf
         public      evcnt_init,evcnt_intr,pt1_intr
         public      pre_count,count,ovf_flag
;
pt1_vector     equ   000006h                      ;timer 1 interrupt vector offset
evcnt_vector   equ   000008h                      ;event counter interrupt vector offset
evcnt          equ   003000h                      ;program start address offset
br_io          equ   0ffh                         ;base reg. address (set i/o area)
pt_mode0       equ   00ff30h                      ;event counter timer mode set reg. 0
pt_mode1       equ   00ff31h                      ;event counter mode set reg. 1
pt_mode2       equ   00ff32h                      ;programmable timer mode set reg. 2
rld0           equ   00ff33h                      ;event counter reload data
rld1           equ   00ff34h                      ;programmable timer 1 reload data
ptd0           equ   00ff35h                      ;event counter counting data
ptd1           equ   00ff36h                      ;programmable timer 1 counter data
;
intr_pr1       equ   00ff21h                      ;interrupt priority reg. 1
intr_en1       equ   00ff23h                      ;interrupt enable reg. 1
intr_fac1      equ   00ff25h                      ;interrupt factor flag reg. 1
;
         data
pre_count:     db    [1]                          ;previous event counter data
count:         db    [1]                          ;present event counter data
ovf_flag:      db    [1]                          ;event counter overflow flag
         code
```

Vector address setting for 8-bit event counter interrupt

```
intr_vectors:
         org   intr_vectors+pt1_vector
         dw    pt1_intr
;
         org   intr_vectors+evcnt_vector
         dw    evcnt_intr                         ;event counter overflow interrupt
;
```

(3) Initialization for 8-bit event counter

```
         org   intr_vectors+evcnt
timerdata8_2:  db    250                          ;timer 1 reload data (4msec at 4mhz/64)
;*************************************************************************
;*                                                                       *
;* 8-bit event counter (timer 0) counting between 4msec (reload timer 1) *
;*                                                                       *
;*************************************************************************
;*** initialize routine
evcnt_init:
         ld    br,#br_io                          ;set br reg. address to 0ffxxh
         carl  osc1toosc3                         ;change osc1 to osc3 ***
         ld    a,[lod vdd_ngf]
         cp    a,#0ffh
         jrl   z,evcnt_init01
;
;mode16=8-bit,chsel=timer 0,pulse output=off,cksel1=fosc3,cksel0=don't care
         ld    [br:low pt_mode0],#00000011b
         ld    a,#0ffh
         ld    [br:low rld0],a                    ;set event counter init data (max.)
         ld    [lod pre_count],a                  ;pre event counter data set
         ld    a,[loc timerdata8_2]
         ld    [br:low rld1],a                    ;set reload data (timer 1)
```

Source List

```
;pt0:evcnt=event counter,fcsel=with noise rejector,plpol=falling edge
;psc1&0=don't care,rlmd0=reload,pset0=preset,prrun0=stop
         ld    [br:low pt_mode1],#11000110b
;pt1:psc=fosc3/64,rlmd1=reload,pset1=preset
         ld    [br:low pt_mode2],#00011110b
         or    [br:low intr_pr1],#00001100b      ;set pt=/irq3
         or    [br:low intr_en1],#11000000b      ;ept1&0 intr. en.
         or    [br:low pt_mode1],#00000001b      ;start event counter
         or    [br:low pt_mode2],#00000001b      ;start timer 1
         ld    a,sc
         and   a,#00111111b
         or    a,#10000000b
         ld    sc,a                              ;i0 flag clear (en. /irq3 intr.)
         xor   a,a
         ld    [lod ovf_flag],a                  ;overflow flag clear
;************************************************************************
;*** start event counter (timer 0) and programmable timer 1 interrupt
;        (user program)
;
loop:
         ld    a,[lod ovf_flag]
         cp    a,#0ffh                           ;event counter overflow ?
         jrs   nz,evcnt_init00                   ;--> normal
;************************************************************************
;*** event counter overflow processing
;        (user program)
;
         jrs   event_init02
;************************************************************************
;*** normal processing
evcnt_init00:
;        (user program)
;
event_init02:
         jrs   loop                              ;-->
;************************************************************************
;*** end processing
         and   [br:low pt_mode1],#11111110b      ;stop event counter
         and   [br:low pt_mode2],#11111110b      ;stop timer 1
evcnt_init01:
         carl  osc3toosc1                        ;change osc3 to osc1 ***
         ret
```

(3) Interrupt processing for 8-bit event counter

```
;************************************************************************
;*                                                                      *
;*   programmable timer 1 interrupt processing routine (reload mode)    *
;*                                                                      *
;************************************************************************
pt1_intr:
         push  ale
;
         ld    br,#br_io                         ;set br reg. address to 0ffxxh
;
         ld    a,[br:low ptd0]                   ;read event counter counting data
         ld    [lod count],a                     ;set present event counter data
         and   [br:low pt_mode1],#11111110b      ;event counter stop
         or    [br:low pt_mode1],#00000010b      ;set event counter next data (max.)
         ld    a,[lod pre_count]
         sub   a,[lod count]                     ;a-reg. = input count number (4 msec)
;
;
```

Source List

```
;        programmable timer 1 processing (based on event counter counting data)
;
;
         ld    a,[lod count]
         ld    [lod pre_count],a                ;move event counter data p(i-1) <- p(i)
         and   [br:low intr_fac1],#10000000b    ;clear fpt1 interrupt flag
         pop   ale
         rete
;***************************************************************************
;*                                                                         *
;* event counter (timer 0) interrupt processing routine (counter overflow) *
;*                                                                         *
;***************************************************************************
evcnt_intr:
         push  ale
;
         ld    br,#br_io                        ;set br reg. address to 0ffxxh
;
         ld    a,#0ffh
         ld    [lod ovf_flag],a                 ;event counter overflow flag set
         and   [br:low pt_mode1],#11111110b     ;event counter stop
;
;
;        event counter overflow processing
;
;
         and   [br:low intr_fac1],#01000000b    ;clear fpt0 interrupt flag
         pop   ale
         rete
         end
```

Source List

Control of programmable timer 4

```
         external     osc1toosc3,osc3toosc1
         external     vdd_ngf
         public       measure_init,measure_intr
         public       count,ovf_flag,k1x_intr
;
measure_vector equ    000006h                    ;measure interrupt vector offset
k1x_vector     equ    00000ah                    ;k1x interrupt vector offset
pm             equ    003000h                    ;program start address offset
br_io          equ    0ffh                       ;base reg. address (set i/o area)
pt_mode0       equ    00ff30h                    ;pulse width measure mode set reg. 0
pt_mode1       equ    00ff31h                    ;pulse width measure mode set reg. 1
rld0           equ    00ff33h                    ;pulse width measure (low)  reload data
rld1           equ    00ff34h                    ;pulse width measure (high) reload data
ptd0           equ    00ff35h                    ;pulse width measure (low)  count data
ptd1           equ    00ff36h                    ;pulse width measure (high) count data
sik1           equ    00ff51h                    ;interrupt selection reg. for k1x
kcp1           equ    00ff53h                    ;interrupt comparison reg. for k1x
k1d            equ    00ff55h                    ;input data from k1x
;
intr_pr1       equ    00ff21h                    ;interrupt priority reg. 1
intr_en1       equ    00ff23h                    ;interrupt enable reg. 1
intr_fac1      equ    00ff25h                    ;interrupt factor flag reg. 1
;
         data
count:         dw     [1]                        ;pulse width measured data
ovf_flag:      db     [1]                        ;event counter overflow flag
         code
```

Vector address setting for 16-bit pulse width measurement timer interrupt

```
intr_vectors:
         org   intr_vectors+measure_vector
         dw    measure_intr                      ;pulse width measure overflow interrupt
;
         org   intr_vectors+k1x_vector
         dw    k1x_intr                          ;k1x interrupt processing routine
;
```

(4) Initialization for 16-bit pulse width measurement timer

```
         org   intr_vectors+pm
;*************************************************************************
;*                                                                       *
;* 16-bit pulse width measurement (timer 0) between k10 "low" input term *
;*                                                                       *
;*************************************************************************
;*** initialize routine
measure_init:
         ld    br,#br_io                         ;set br reg. address to 0ffxxh
         carl  osc1toosc3                        ;change osc1 to osc3 ***
         ld    a,[lod vdd_ngf]                   ;vdd ng flag
         cp    a,#0ffh
         jrl   z,measure_init01
;mode16=16-bit,chsel=timer 0,pulse output=off,cksel1=don't care,cksel0=fosc3
         ld    [br:low pt_mode0],#00011001b
         ld    ba,#0ffffh
         ld    [lod rld0],ba                     ;set measure counter init data (max.)
;pt0:evcnt=timer,fcsel=pulse width measurement,plpol=low level measurement
;psc=fosc3/1,rlmd0=reload,pset0=preset,prrun0=stop
         ld    [br:low pt_mode1],#01000110b
         ld    [br:low kcp1],#00000001b          ;k10 falling edge ("h" -> "l")
```

Source List

```
         ld    [br:low sik1],#00000001b          ;k10 interrupt enable
         or    [br:low intr_pr1],#00001111b      ;set pt & pk1=/irq3
         or    [br:low intr_en1],#10100000b      ;ept1 & ek1 intr. en.
         or    [br:low pt_mode1],#00000001b      ;start pulse measurement
         ld    a,sc
         and   a,#00111111b
         or    a,#10000000b
         ld    sc,a                              ;i0 flag clear (en. /irq3 intr.)
         xor   a,a
         ld    [lod ovf_flag],a                  ;overflow flag clear
;************************************************************************
;*** start measure counter (16-bit timer 0)
;        (user program)
;
wait_loop:
         ld    a,[lod ovf_flag]
         cp    a,#0ffh                           ;measure counter overflow ?
         jrs   nz,measure_init00                 ;--> normal
;************************************************************************
;*** measure counter overflow processing
;        (user program)
;
         jrs   measure_init02
;*** normal processing
measure_init00:
;        (user program)
;
measure_init02:
         jrs   wait_loop                         ;-->
;************************************************************************
;*** end processing
         and   [br:low pt_mode1],#11111110b      ;stop measure counter
measure_init01:
         carl  osc3toosc1                        ;change osc3 to osc1 ***
         ret
```

(4) Interrupt processing for 16-bit pulse width measurement timer

```
;****************************************************************************
;*                                                                          *
;* measure counter (16-bit timer 0) interrupt processing routine (overflow) *
;*                                                                          *
;****************************************************************************
measure_intr:
         push  ale
;
         ld    br,#br_io                         ;set br reg. address to 0ffxxh
;
         ld    a,#0ffh
         ld    [lod ovf_flag],a                  ;event counter overflow flag set
         and   [br:low pt_mode1],#11111110b      ;measure counter stop
;
;
;        measure counter overflow processing
;
;
         and   [br:low intr_fac1],#10000000b     ;clear fpt1 interrupt flag
         pop   ale
         rete
```

Source List

```
;*************************************************************************
;*                                                                       *
;*       k1x interrupt processing routine                                *
;*                                                                       *
;*************************************************************************
k1x_intr:
         push  ale
;
         ld    br,#br_io                         ;set br reg. address to 0ffxxh
;
         bit   [br:low kcp1],#00000001b          ;kcp setting ?
         jrs   z,k1x_intr01
;falling edge -> rising edge
         and   [br:low kcp1],#11111110b          ;set rising edge
         jrs   k1x_intr00
;rising edge -> falling edge
k1x_intr01:
         or    [br:low kcp1],#00000001b          ;set falling edge
         ld    ba,[lod ptd0]
         ld    [lod count],ba                    ;read measure count data
         or    [br:low pt_mode1],#00000010b      ;set measure counter init data (max.)
k1x_intr00:
         and   [br:low intr_fac1],#00100000b     ;clear fk1 interrupt flag
         pop   ale
         rete
         end
```

Source List

Control of programmable timer 5

```
         external     osc1toosc3,osc3toosc1
         external     vdd_ngf
         public       pulsout_init
;
br_io          equ    0ffh                         ;base reg. address (set i/o area)
pt_mode0       equ    00ff30h                      ;programmable timer mode set reg. 0
pt_mode1       equ    00ff31h                      ;programmable timer mode set reg. 1
rld0           equ    00ff33h                      ;programmable timer 0 reload data
rld1           equ    00ff34h                      ;programmable timer 1 reload data
hzr2           equ    00ff71h                      ;r2x output control reg.
r2d            equ    00ff75h                      ;r2x output data
;
intr_pr1       equ    00ff21h                      ;interrupt priority reg. 1
intr_en1       equ    00ff23h                      ;interrupt enable reg. 1
intr_fac1      equ    00ff25h                      ;interrupt factor flag reg. 1
;
         code
```

(5) 16-bit reload timer pulse output

```
;pulse output=66.4ms(approx. 15hz)
timerdata16:   dw     33200                        ;timer16 reload data (33.2ms at 4mhz/4)
;*************************************************************************
;*                                                                       *
;*       pulse out (16-bit) control                                      *
;*                                                                       *
;*************************************************************************
;*** initialize routine
pulsout_init:
         ld     br,#br_io                          ;set br reg. address to 0ffxxh
         carl   osc1toosc3                         ;change osc1 to osc3 ***
         ld     a,[lod vdd_ngf]                    ;vdd ng flag
         cp     a,#0ffh
         jrl    z,pulsout_init00
;
         or     [br:low r2d],#10000000b            ;r27="h" (enable ptout)
         and    [br:low hzr2],#01111111b           ;r27=complementary output
;mode16=16-bit,chsel=timer0,ptout=off,cksel1=don't care,cksel0=fosc3
         ld     [br:low pt_mode0],#00011001b
         ld     ba,[loc timerdata16]               ;set 16-bit counter data (timer 0 & 1)
         ld     [lod rld0],ba
;pt0:evcnt=timer,fcsel=normal timer,plpol=don't care,psc=fosc3/4,rlmd0=reload
;pset0=preset
         ld     [br:low pt_mode1],#00001110b
         or     [br:low pt_mode1],#00000001b       ;start timer 0
         or     [br:low pt_mode0],#00000100b       ;start ptout
;*************************************************************************
;*** start pulse out (16-bit)
;        (user program)
;
;*************************************************************************
;*** end processing
         and    [br:low pt_mode0],#11111011b       ;stop ptout
         and    [br:low pt_mode1],#11111110b       ;stop timer 0
pulsout_init00:
         carl   osc3toosc1                         ;change osc3 to osc1 ***
         ret
         end
```

13 SOUND GENERATOR

I/O Map

Address	Bit	Name	Function		1	0	SR	R/W	Comment
00FF44	D7	–	–		–	–	–		Constantry "0" when being read
	D6	BZSTP	One-shot buzzer forcibly stop		Forcibly stop	No operation	–	W	
	D5	BZSHT	One-shot buzzer trigger/status	R	Busy	Ready	0	R/W	
				W	Trigger	No operation			
	D4	SHTPW	One-shot buzzer duration width selection		125 msec	31.25 msec	0	R/W	
	D3	ENRTM	Envelope attenuation time		1 sec	0.5 sec	0	R/W	
	D2	ENRST	Envelope reset		Reset	No operation	–	W	"0" when being read
	D1	ENON	Envelope On/Off control		On	Off	0	R/W	*1
	D0	BZON	Buzzer output control		On	Off	0	R/W	
00FF45	D7	–	–		–	–	–		"0" when being read
	D6	DUTY2	Buzzer signal duty ratio selection DUTY2–1 | Buzzer frequency (Hz) 2 1 0 | 4096.0/2048.0 | 3276.8/1638.4 | 2730.7/1365.3 | 2340.6/1170.3 0 0 0 | 8/16 | 8/20 | 12/24 | 12/28 0 0 1 | 7/16 | 7/20 | 11/24 | 11/28 0 1 0 | 6/16 | 6/20 | 10/24 | 10/28 0 1 1 | 5/16 | 5/20 | 9/24 | 9/28 1 0 0 | 4/16 | 4/20 | 8/24 | 8/28 1 0 1 | 3/16 | 3/20 | 7/24 | 7/28 1 1 0 | 2/16 | 2/20 | 6/24 | 6/28 1 1 1 | 1/16 | 1/20 | 5/24 | 5/28				0	R/W	
	D5	DUTY1					0	R/W	
	D4	DUTY0					0	R/W	
	D3	–	–				–		"0" when being read
	D2	BZFQ2	Buzzer frequency selection BZFQ2 | BZFQ1 | BZFQ0 | Frequency (Hz) 0 | 0 | 0 | 4096.0 0 | 0 | 1 | 3276.8 0 | 1 | 0 | 2730.7 0 | 1 | 1 | 2340.6 1 | 0 | 0 | 2048.0 1 | 0 | 1 | 1638.4 1 | 1 | 0 | 1365.3 1 | 1 | 1 | 1170.3				0	R/W	
	D1	BZFQ1					0	R/W	
	D0	BZFQ0					0	R/W	

*1 Reset to "0" during one-shot output.

Specifications

Control of sound generator

(1) sound_init: Initialization for sound generator

Enables the buzzer output from R50 terminal.

(2) normal_init, normal_on, normal_off: Normal buzzer output

The normal_init routine sets the duty ratio of the buzzer signal to maximum and the frequency to 4,096 Hz. There is buzzer output when normal_on has been called until normal_off is called.

(3) envelope_init, envelope_on, envelope_reset, envelope_off: Buzzer output with digital envelope

The envelope_init routine sets the buzzer signal frequency to 4,096 Hz and the envelope attenuation time to 1 sec and then turns the envelope ON.

There is buzzer output when envelope_on has been called until envelope_off is called.

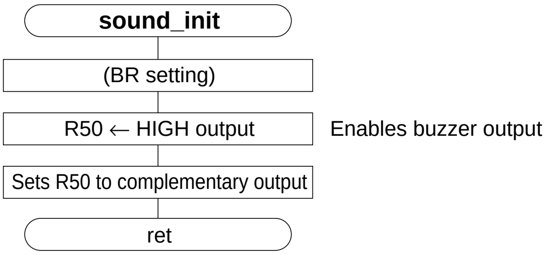
The envelope_reset routine re-sets the buzzer signal frequency to 2,048 Hz and the envelope attenuation time to 0.5 sec and then resets the envelope. The envelope is reset by calling envelope_reset during output period of a buzzer with envelope.

(4) oneshot_ready, oneshot_on, oneshot_off: One-shot buzzer output

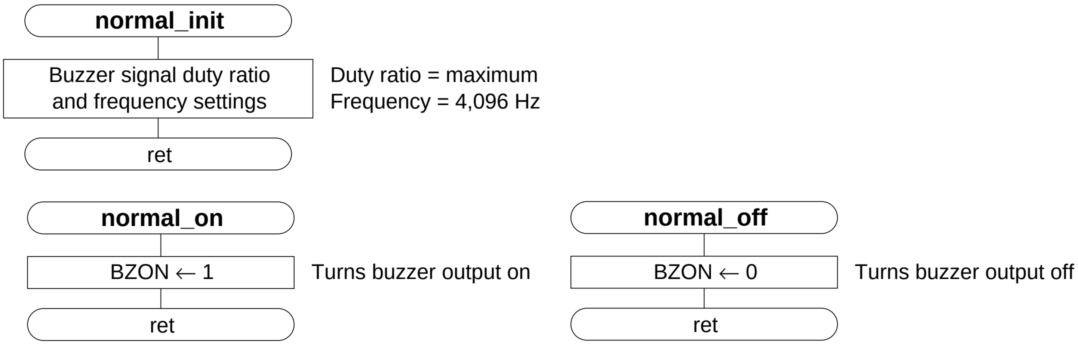
The oneshot_ready routine sets the one-shot buzzer pulse width to 125 msec and waits until the one-shot buzzer output has shifted to READY status. One-shot buzzer output is done by calling oneshot_on. Buzzer output is 125 msec when called by oneshot_on, but even in that time, the one-shot buzzer output can be forcibly terminated by calling oneshot_off.

Flowchart

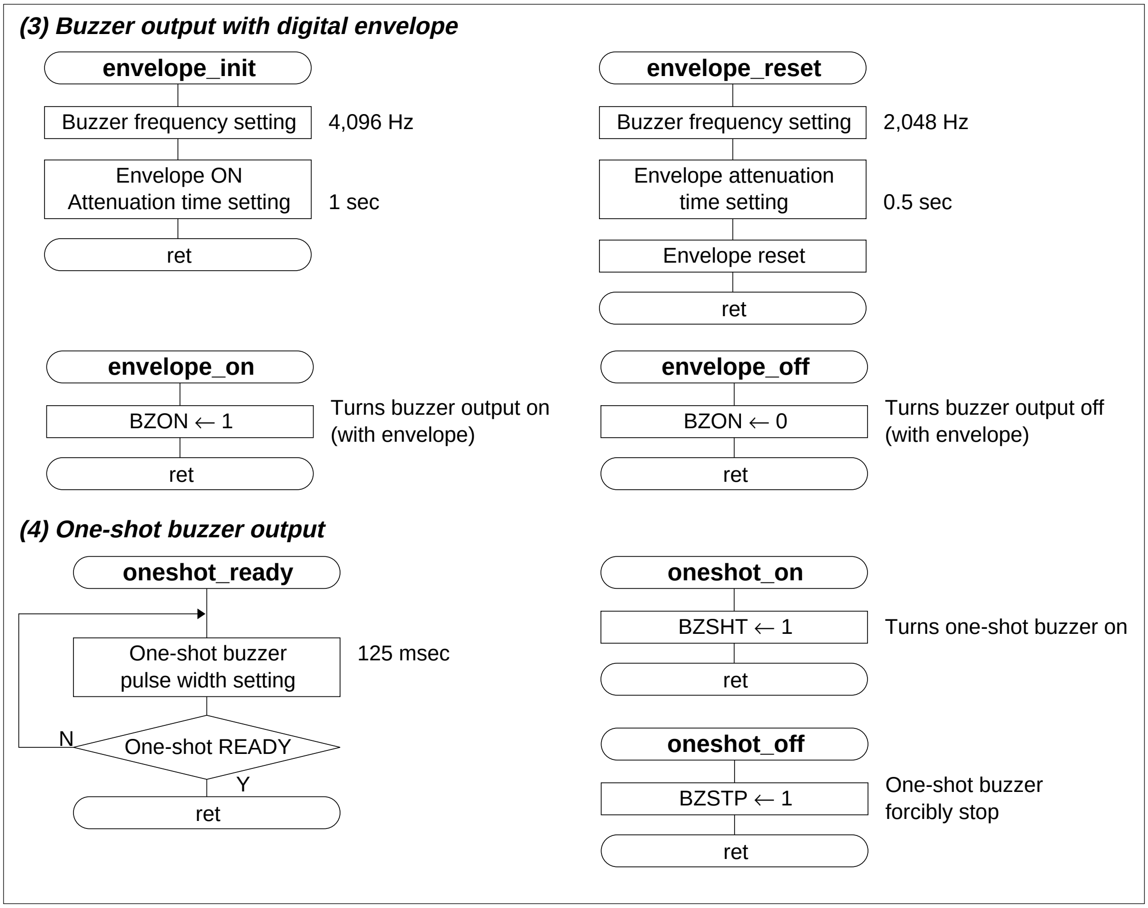
(1) Initialization for sound generator



(2) Normal buzzer output



Flowchart



Note

The R50 terminal is common to the normal DC output port and the buzzer output. When a buzzer circuit has been configured with the R50 terminal, set the R50 register to "1" and control the signal ON/OFF using the BZON register.

Source List

Control of sound generator

```
         public        sound_init
         public        normal_init,normal_on,normal_off
         public        envelope_init,envelope_on,envelope_reset,envelope_off
         public        oneshot_ready,oneshot_on.oneshot_off
;
br_io          equ   0ffh                        ;base reg. address (set i/o area)
sound_mode0    equ   00ff44h                     ;sound generator mode set reg. 0
sound_mode1    equ   00ff45h                     ;sound generator mode set reg. 1
;
hzr_ex         equ   00ff70h                     ;expand output control reg.
r5d            equ   00ff78h                     ;r5x output data
         code
;
```

(1) Initialization for sound generator

```
;************************************************************************
;*                                                                      *
;*       sound genertator control                                       *
;*                                                                      *
;************************************************************************
sound_init:
         ld    br,#br_io                         ;set br reg. address to 0ffxxh
         and   [br:low r5d],#11111110b           ;r50="l" (bzon enable)                (1)
         and   [br:low hzr_ex],#10111111b        ;r50=complementary output
         ret
;
```

(2) Normal buzzer output

```
;************************************************************************
;*** sound normal
normal_init:
         ld    [br:low sound_mode1],#00000000b   ;duty=max.,bzfq=4096hz
         ret
;***
normal_on:
         or    [br:low sound_mode0],#00000001b   ;bzon=enable                          (2)
         ret
;***
normal_off:
         and   [br:low sound_mode0],#11111110b   ;bzon=disable
         ret
;
```

(3) Buzzer output with digital envelope

```
;************************************************************************
;*** sound envelope
envelope_init:
         ld    [br:low sound_mode1],#00000000b   ;duty=don't care,bzfq=4096hz
         or    [br:low sound_mode0],#00001010b   ;enrtm=1sec,enon=on
         ret
;***
envelope_on:
         or    [br:low sound_mode0],#00000001b   ;bzon=enable (with envelope)          (3)
         ret
;***
;envelope reset then on(change envelope release time & buzzer frequency)
envelope_reset:
         ld    [br:low sound_mode1],#00000100b   ;duty=don't care,bzfq=2048hz
```

Source List

```
        ld     a,[br:low sound_mode0]
        and    a,#00000011b                      ;enrtm=0.5sec
        or     a,#00000100b                      ;envelope reset
        ld     [br:low sound_mode0],a
        ret                                                                      (3)
;***
envelope_off:
        and    [br:low sound_mode0],#11111110b   ;bzon=disable
        ret
;
```

(4) One-shot buzzer output

```
;************************************************************************
;*** sound_oneshot
oneshot_ready:
        or     [br:low sound_mode0],#00010000b   ;one shot width=125ms
        bit    [br:low sound_mode0],#00100000b   ;one shot ready ?
        jrs    nz,oneshot_ready
;
        ret
;***
oneshot_on:
        ld     a,[br:low sound_mode0]
        and    a,#00011111b                                                      (4)
        or     a,#00100000b
        ld     [br:low sound_mode0],a            ;one shot buzzer on
        ret
;***
oneshot_off:
        ld     a,[br:low sound_mode0]
        and    a,#00011111b
        or     a,#01000000b
        ld     [br:low sound_mode0],a            ;no status read stop
        ret
;
        end
```

14 ANALOG COMPARATOR

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF13	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	CMP1ON	Comparator 1 On/Off control	On	Off	0	R/W	
	D2	CMP0ON	Comparator 0 On/Off control	On	Off	0	R/W	
	D1	CMP1DT	Comparator 1 data	+ > -	+ < -	0	R	
	D0	CMP0DT	Comparator 0 data	+ > -	+ < -	0	R	

Specifications

Control of analog comparator

(1) comp_init: Initialization for analog comparator

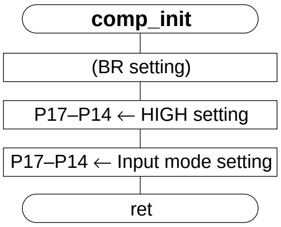
Sets I/O port P17–P14 to the input mode in order to prevent a malfunction.

(2) comp_control: Data reading for analog comparator

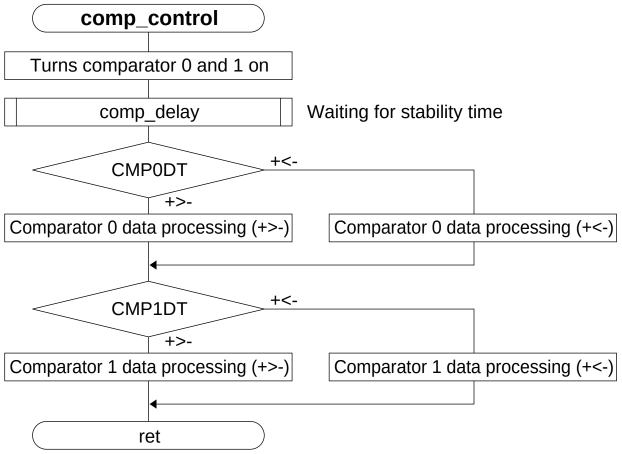
Sets the analog comparator to ON and reads the comparator data after calling a delay routine. Executes subsequent processing according to the results of the read.

Flowchart

(1) Initialization for analog comparator



(2) Data reading for analog comparator


comp_control
Turns comparator 0 and 1 on
comp_delay
Waiting for stability time
CMP0DT
+<-
+>-
Comparator 0 data processing (+>-)
Comparator 0 data processing (+<-)
CMP1DT
+<-
+>-
Comparator 1 data processing (+>-)
Comparator 1 data processing (+<-)
ret

Notes

(1) A delay routine for the operation stabilization waiting time (3 msec, maximum) of the analog comparator is not included in this program example, so it is necessary to create it using a hardware timer or software timer. (external call: comp_delay)

(2) P17–P14 terminals are common to the analog comparator inputs (CMPM0, CMPP0, CMPM1 and CMPP1) and the I/O port, and these are switched to I/O port terminals when the analog comparator is turned OFF. Consequently, for an I/O port which is used for an analog comparator, be sure to set in input mode.

Source List

Control of analog comparator

```
         external    comp_delay
         public      comp_init,comp_control
;
br_io          equ   0ffh                        ;base reg. address (set i/o area)
comp_mode      equ   00ff13h                     ;analog comparator mode set reg.
ioc1           equ   00ff61h                     ;p1x i/o control reg.
p1d            equ   00ff63h                     ;p1x port data
         code
```

(1) Initialization for analog comparator

```
;************************************************************************
;*                                                                      *
;*       comparator control                                             *
;*                                                                      *
;************************************************************************
;*** initialize routine
comp_init:
         ld    br,#br_io                         ;set br reg. address to 0ffxxh
         or    [br:low p1d],#11110000b           ;set p17-14="h"                  (1)
         and   [br:low ioc1],#00001111b          ;set p17-14=input mode
         ret
```

(2) Data reading for analog comparator

```
;************************************************************************
;*** control routine
comp_control:
         or    [br:low comp_mode],#00001100b     ;comparator 0&1 on
         carl  comp_delay                        ;comparator stable delay ***
         bit   [br:low comp_mode],#00000001b     ;comparator 0 on ?
         jrs   z,comp_control00
; comparator 0 : + > -
;
         jrs   comp_control01
; comparator 0 : + < -
comp_control00:
;
comp_control01:
         bit   [br:low comp_mode],#00000010b     ;comparator 1 on ?               (2)
         jrs   z,comp_control02
; comparator 1 : + > -
;
         jrs   comp_control03
; comparator 1 : + < -
comp_control02:
;
; comparator processing end
comp_control03:
         and   [br:low comp_mode],#00001100b
         ret
;
         end
```

15 SVD (SUPPLY VOLTAGE DETECTION) CIRCUIT

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF12	D7	–	–	–	–	–		Constantry "0" when being read
	D6	–	–	–	–	–		
	D5	SVDSP	SVD auto-sampling control	On	Off	0	R/W	These registers are reset to "0" when SLP instruction is executed.
	D4	SVDON	SVD continuous sampling control/status R	Busy	Ready	1→0*1	R/W	
			W	On	Off	0		
	D3	SVD3	SVD detection level			X	R	*2
	D2	SVD2	SVD3 SVD2 SVD1 SVD0 Detection level 1 1 1 1 Level 15 1 1 1 0 Level 14 : : : : : 0 0 0 0 Level 0			X	R	
	D1	SVD1				X	R	
	D0	SVD0				X	R	

*1 After initial reset, this status is set "1" until conclusion of hardware first sampling.

*2 Initial values are set according to the supply voltage detected at first sampling by hardware. Until conclusion of first sampling, SVD0–SVD3 data are undefined.

Specifications

Control of SVD circuit

(1) svd_auto: Supply voltage detection in 1/4 Hz auto sampling mode

After setting the 1/4 Hz auto-sampling mode to turn the SVD circuit ON, reads out SVD detection data into the A register.

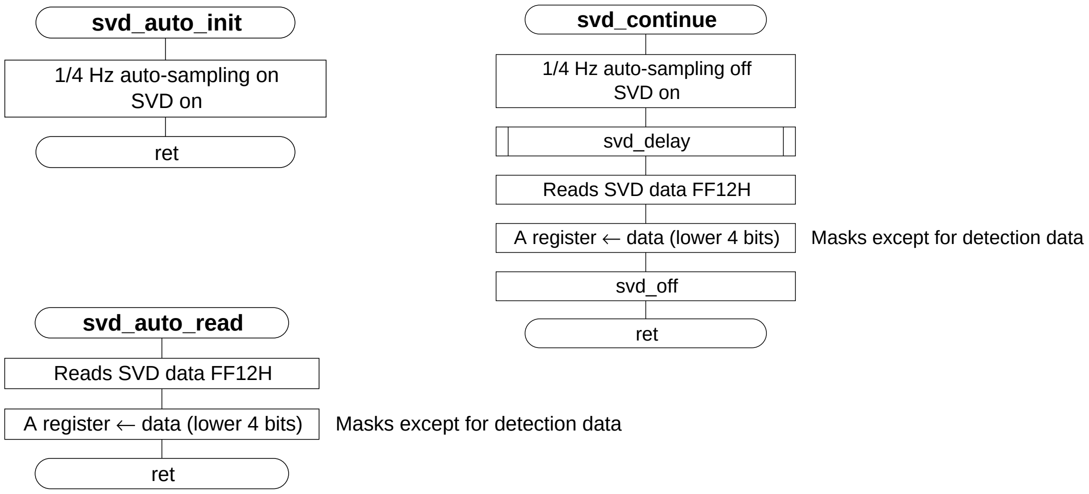
(2) SVD_continue: Supply voltage detection in continuously sampling mode

Sets the continuous sampling mode (cancels the 1/4 Hz auto-sampling mode) to turn the SVD circuit ON, and reads out SVD detection data into A register after calling a delay routine.

Flowchart

(1) Supply voltage detection in 1/4 Hz auto sampling mode

(2) Supply voltage detection in continuously sampling mode



Notes

(1) A delay routine that waits data decision time (approximately 7.8 msec or more) for the SVD circuit has not been included in this program example, so it is necessary to create a separate routine using a hardware timer or software timer. (external call: svd_delay)

(2) In the continuous sampling mode, when reading the detection data without waiting the data decision time (approximately 7.8 msec or more), previous data that has not been updated will be read.

Source List

Control of SVD circuit

```
         external     svd_delay
         public       svd_auto_init,svd_auto_read,svd_contine
;
br_io          equ    0ffh                        ;base reg. address (set i/o area)
svd_mode       equ    00ff12h                     ;supply voltage detector mode set reg.
         code
;
```

(1) Supply voltage detection in 1/4 Hz auto sampling mode

```
;************************************************************************
;*                                                                      *
;*       svd control                                                    *
;*                                                                      *
;************************************************************************
;*** auto sampling mode
svd_auto_init:
         or     [br:low svd_mode],#00100000b      ;auto sampling
         ret
;                                                                            (1)
;
         ld     a,[br:low svd_mode]               ;read svd data
         and    a,#0fh
         ret
```

(2) Supply voltage detection in continuously sampling mode

```
;************************************************************************
;*** continuos mode
svd_continue:
         ld     a,[br:low svd_mode]
         and    a,#00011111b                      ;auto sampling off
         or     a,#00010000b                      ;svd on
         ld     [br:low svd_mode],a
;
;                                                                            (2)
         carl   svd_delay                         ;svd stable delay
;
         ld     a,[br:low svd_mode]               ;read svd data
         and    a,#0fh
         and    [br:low svd_mode],#00001111b      ;svd off
         ret
;
         end
```

16 INTERRUPT (EXCEPTION) PROCESSING

I/O Map

Address	Bit	Name	Function	1	0	SR	R/W	Comment
00FF20	D7	PK01	K00–K07 interrupt priority register	PK01 PK00 PSIF1 PSIF0 PSW1 PSW0 Priority PTM1 PTM0 level 1 1 Level 3 1 0 Level 2 0 1 Level 1 0 0 Level 0		0	R/W	
	D6	PK00						
	D5	PSIF1	Serial interface interrupt priority register			0	R/W	
	D4	PSIF0						
	D3	PSW1	Stopwatch timer interrupt priority register			0	R/W	
	D2	PSW0						
	D1	PTM1	Clock timer interrupt priority register			0	R/W	
	D0	PTM0						
00FF21	D7	–	–	–	–	–		Constantly "0" when being read
	D6	–	–	–	–	–		
	D5	–	–	–	–	–		
	D4	–	–	–	–	–		
	D3	PPT1	Programmable timer interrupt priority register	PPT1 PPT0 Priority PK11 PK10 level 1 1 Level 3 1 0 Level 2 0 1 Level 1 0 0 Level 0		0	R/W	
	D2	PPT0						
	D1	PK11	K10 and K11 interrupt priority register			0	R/W	
	D0	PK10						
00FF22	D7	–	–	–	–	–		"0" when being read
	D6	ESW100	Stopwatch timer 100 Hz interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D5	ESW10	Stopwatch timer 10 Hz interrupt enable register					
	D4	ESW1	Stopwatch timer 1 Hz interrupt enable register					
	D3	ETM32	Clock timer 32 Hz interrupt enable register					
	D2	ETM8	Clock timer 8 Hz interrupt enable register					
	D1	ETM2	Clock timer 2 Hz interrupt enable register					
	D0	ETM1	Clock timer 1 Hz interrupt enable register					
00FF23	D7	EPT1	Programmable timer 1 interrupt enable register	Interrupt enable	Interrupt disable	0	R/W	
	D6	EPT0	Programmable timer 0 interrupt enable register					
	D5	EK1	K10 and K11 interrupt enable register					
	D4	EK0H	K04–K07 interrupt enable register					
	D3	EK0L	K00–K03 interrupt enable register					
	D2	ESERR	Serial I/F (error) interrupt enable register					
	D1	ESREC	Serial I/F (receiving) interrupt enable register					
	D0	ESTRA	Serial I/F (transmitting) interrupt enable register					
00FF24	D7	–	–	–	–	–		"0" when being read
	D6	FSW100	Stopwatch timer 100 Hz interrupt factor flag	(R) Interrupt factor is generated (W) Reset	(R) No interrupt factor is generated (W) No operation	0	R/W	
	D5	FSW10	Stopwatch timer 10 Hz interrupt factor flag					
	D4	FSW1	Stopwatch timer 1 Hz interrupt factor flag					
	D3	FTM32	Clock timer 32 Hz interrupt factor flag					
	D2	FTM8	Clock timer 8 Hz interrupt factor flag					
	D1	FTM2	Clock timer 2 Hz interrupt factor flag					
	D0	FTM1	Clock timer 1 Hz interrupt factor flag					
00FF25	D7	FPT1	Programmable timer 1 interrupt factor flag	(R) Interrupt factor is generated (W) Reset	(R) No interrupt factor is generated (W) No operation	0	R/W	
	D6	FPT0	Programmable timer 0 interrupt factor flag					
	D5	FK1	K10 and K11 interrupt factor flag					
	D4	FK0H	K04–K07 interrupt factor flag					
	D3	FK0L	K00–K03 interrupt factor flag					
	D2	FSERR	Serial I/F (error) interrupt factor flag					
	D1	FSREC	Serial I/F (receiving) interrupt factor flag					
	D0	FSTRA	Serial I/F (transmitting) interrupt factor flag					

Specifications

Interrupt (exception) processing

Setting of interrupt vector address

(1) main: Interrupt level setting and enables interrupt

Sets an interrupt level ($\overline{\text{IRQ3}}$–$\overline{\text{IRQ1}}$) as the below for all interrupts and enables interrupts in the initialization routine (example for 88112 single chip mode) which is executed by reset exception processing.

• Programmable timer interrupt	$\overline{\text{IRQ3}}$
• Input port interrupt	$\overline{\text{IRQ3}}$
• Serial interface interrupt	$\overline{\text{IRQ2}}$
• Stopwatch timer interrupt	$\overline{\text{IRQ1}}$
• Clock timer interrupt	$\overline{\text{IRQ1}}$

(2) zero_div: Zero division exception processing

(3) watchdog: Watchdog timer ($\overline{\text{NMI}}$) interrupt processing

(4) xxx_intr: Interrupt processing for peripheral circuit

Notes

(1) The interrupt level ($\overline{\text{IRQ3}}$–$\overline{\text{IRQ1}}$) can be set to adapt to the system.

(2) Be sure to initialize peripheral circuits which use an interrupt and set interrupt generation conditions beforehand to enable each interrupt.

(3) Interrupt processing for a peripheral circuit enables all interrupts, and exception processing with an interrupt vectors is a precondition. Since an interrupt flag is set by the generation of an interrupt regardless of the interrupt enable register and interrupt flags (I1 and I0), a procedure for polling interrupt factor flags by software can also be used.

(4) Since the watchdog timer ($\overline{\text{NMI}}$) interrupt cannot be masked, be sure to declare the watchdog timer ($\overline{\text{NMI}}$) interrupt processing routine and the vector address, regardless of whether or not the watchdog timer is used.

(5) To reset the interrupt factor flag, write "1" into the corresponding flags alone, using the AND or LD instruction. When the OR logic operation instruction has been used, "1" is written for the interrupt factor flags that have been set to "1" within the same address and those flags are then clear.

(6) The interrupt flags (I1 and I0) have not been reset in the interrupt processing routine of this program example, so an interrupt lower than the set level is disabled at the time of generation. When you wish to accept the next interrupt after an interrupt has been generated, re-setting of the interrupt flags or resetting the interrupt factor flag is necessary after due consideration for the nesting level.

(7) When permitting interrupt nesting, be careful of the stack size.

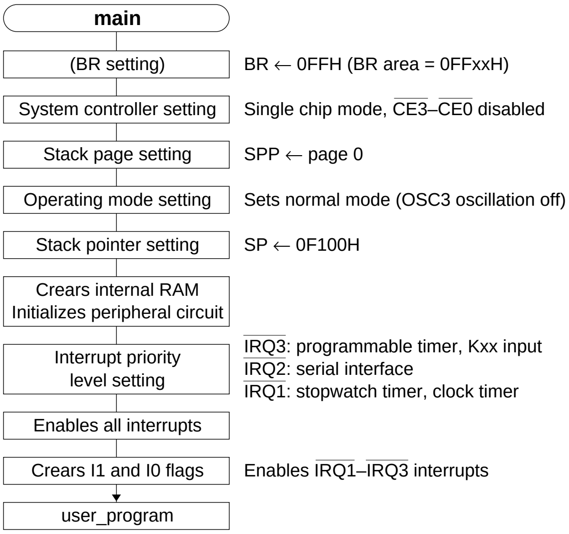
(8) Vector addresses for software interrupts can be set up to 109 and to optional address (two bytes which begin with an even address) from 000026H to 0000FEH.

(9) The vector addresses 000024H and 000025H cannot be used since this is a system reserved area for the E0C88112.

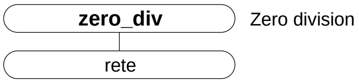
(10) In this program example for interrupt (exception) processing, the vector address setting and program have been allocated from 000100H for the sake of convenience.

Flowchart

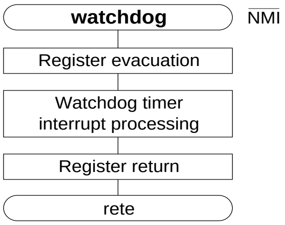
(1) Interrupt level setting and enables interrupt


main
(BR setting)
BR ← 0FFH (BR area = 0FFxxH)
System controller setting
Single chip mode, $\overline{\text{CE3}}$–$\overline{\text{CE0}}$ disabled
Stack page setting
SPP ← page 0
Operating mode setting
Sets normal mode (OSC3 oscillation off)
Stack pointer setting
SP ← 0F100H
Crears internal RAM
Initializes peripheral circuit
Interrupt priority
level setting
$\overline{\text{IRQ3}}$: programmable timer, Kxx input
$\overline{\text{IRQ2}}$: serial interface
$\overline{\text{IRQ1}}$: stopwatch timer, clock timer
Enables all interrupts
Crears I1 and I0 flags
Enables $\overline{\text{IRQ1}}$–$\overline{\text{IRQ3}}$ interrupts
user_program

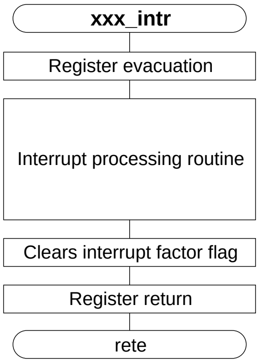
(2) Zero division exception processing



(3) Watchdog timer ($\overline{\text{NMI}}$) interrupt processing



(4) Interrupt processing for peripheral circuit



xxx_intr	Interrupt factor flag	
pt1_intr	FPT1	Programmable timer 0
pt0_intr	FPT0	Programmable timer 1
k1x_intr	FK1	K10 and K11 input port
k0h_intr	FK0H	K07–K04 input port
k0l_intr	FK0L	K03–K00 input port
serr_intr	FSERR	Serial interface error
srv_intr	FSREC	Serial interface receiving
str_intr	FSTRA	Serial interface transmitting
sw100_intr	FSW100	Stopwatch timer 100 Hz
sw10_intr	FSW10	Stopwatch timer 10 Hz
sw1_intr	FSW1	Stopwatch timer 1 Hz
rtm32_intr	FTM32	Clock timer 32 Hz
rtm8_intr	FTM8	Clock timer 8 Hz
rtm2_intr	FTM2	Clock timer 2 Hz
rtm1_intr	FTM1	Clock timer 1 Hz

Source List

Interrupt (exception) processing

```
         external     user_program
         public       main,zero_div,watch_dog
         public       pt1_intr,pt0_intr
         public       k1x_intr,k0h_intr,k0l_intr
         public       serr_intr,srv_intr,str_intr
         public       sw100_intr,sw10_intr,sw1_intr
         public       clock32_intr,clock8_intr,clock2_intr,clock1_intr
;
br_io          equ   0ffh                     ;base reg. address (set i/o area)
mcu            equ   00ff00h                  ;mcu mode system control reg.
spp            equ   00ff01h                  ;stack pointer page address
mode           equ   00ff02h                  ;mcu//mpu mode control address
sp_112         equ   00f100h                  ;E0C88112 stack top address
intr_pr0       equ   00ff20h                  ;interrupt priority reg. 0
intr_pr1       equ   00ff21h                  ;interrupt priority reg. 1
intr_en0       equ   00ff22h                  ;interrupt enable reg. 0
intr_en1       equ   00ff23h                  ;interrupt enable reg. 1
intr_fac0      equ   00ff24h                  ;interrupt factor reg. 0
intr_fac1      equ   00ff25h                  ;interrupt factor reg. 1
;
reserve        equ   000024h                  ;E0C88112 system reserve
soft_intr      equ   000026h                  ;software interrupt vector
offset         equ   000100h                  ;program start address offset
         code
```

Setting of interrupt vector address

```
intr_vectors:
;system interrupt vectors
         dw    main                           ;reset vector
         dw    zero_div                       ;zero divide
         dw    watchdog                       ;watchdog timer(/nmi)
;E0C88112 peripheral interrupt vectors (irq levels can set by software)
         dw    pt1_intr                       ;programmable timer 1 (/irq3)
         dw    pt0_intr                       ;programmable timer 0 (/irq3)
         dw    k1x_intr                       ;k1x input port       (/irq3)
         dw    k0h_intr                       ;k07-04 input port    (/irq3)
         dw    k0l_intr                       ;k03-00 input port    (/irq3)
         dw    serr_intr                      ;serial error         (/irq2)
         dw    srv_intr                       ;serial receive       (/irq2)
         dw    str_intr                       ;serial transmission  (/irq2)
         dw    sw100_intr                     ;stopwatch 100hz      (/irq1)
         dw    sw10_intr                      ;stopwatch 10hz       (/irq1)
         dw    sw1_intr                       ;stopwatch 1hz        (/irq1)
         dw    clock32_intr                   ;clock timer 32hz     (/irq1)
         dw    clock8_intr                    ;clock timer 8hz      (/irq1)
         dw    clock2_intr                    ;clock timer 2hz      (/irq1)
         dw    clock1_intr                    ;clock timer 1hz      (/irq1)
;E0C88112 system reserve
         org   intr_vectors+reserve
;software intrrupt vectors (i.e bios handler and/or general purpose routine(s))
         org   intr_vectors+soft_intr
;
```

Source List

(1) Interrupt level setting and enables interrupt

```
         org    intr_vectors+offset
;************************************************************************
;*                                                                      *
;*       main routine (mcu single chip mode)                            *
;*                                                                      *
;************************************************************************
main:
;
         ld     br,#br_io
;mcu & spp write icludes system interrupt flag reset
         ld     [br:low mcu],#00110000b
         ld     [br:low spp],#0h
         ld     [br:low mode],#00000000b
         ld     sp,#sp_112
;
;
; ram clear and i/o initialize
;
;
;pk0(/irq3),psif(/irq2),psw(/irq1),ptm(/irq1)
         ld     [br:low intr_pr0],#11100101b
;ppt(/irq3),pk1(/irq3)
         ld     [br:low intr_pr1],#00000101b
;esw100,10,1(en),etm32,8,2,1(en.)
         ld     [br:low intr_en0],#01111111b
;ept(en.),ek1(en.)ek0b(en.),ek0a(en.),eserr(en.),esrec(en.),estra(en.)
         ld     [br:low intr_en1],#11111111b
;en. /nmi,/irq3,/irq2,/irq1
         and    sc,#00111111b                     ;i1 & i0 flag clear
;wait for interrupt
         jrl    user_program
;
```

(1)

(2) Zero division exception processing

```
;************************************************************************
;*                                                                      *
;*       zero divide                                                    *
;*                                                                      *
;************************************************************************
zero_div:
         rete
```

(2)

(3) Watchdog timer ($\overline{\text{NMI}}$) interrupt processing

```
;************************************************************************
;*                                                                      *
;*       watchdog timer (/nmi)                                          *
;*                                                                      *
;************************************************************************
watchdog:
         push   ale
         ld     br,#br_io                         ;set br reg. address to 0ffxxh
;
;watchdog timer (/nmi) interrupt processing
;
         pop    ale
         rete
```

(3)

Source List

(4) Interrupt processing for peripheral circuit

```
;************************************************************************
;*                                                                      *
;*       programmable timer 1 (/irq3)                                   *
;*                                                                      *
;************************************************************************
pt1_intr:
        push  ale
;
;       programmable timer 1 interrupt processing                              (4)
;
        and   [br:low intr_fac1],#10000000b   ;clear fpt1 interrupt flag
        pop   ale
        rete
;************************************************************************
;*                                                                      *
;*       programmable timer 0 (/irq3)                                   *
;*                                                                      *
;************************************************************************
pt0_intr:
        push  ale
;
;       programmable timer 0 interrupt processing                              (4)
;
        and   [br:low intr_fac1],#01000000b   ;clear fpt0 interrupt flag
        pop   ale
        rete
;************************************************************************
;*                                                                      *
;*       k1x input port (/irq3)                                         *
;*                                                                      *
;************************************************************************
k1x_intr:
        push  ale
;
;       k1x input port interrupt processing                                    (4)
;
        and   [br:low intr_fac1],#00100000b   ;clear fk1 interrupt flag
        pop   ale
        rete
;************************************************************************
;*                                                                      *
;*       k0h input port (/irq3)                                         *
;*                                                                      *
;************************************************************************
k0h_intr:
        push  ale
;
;       k0h input port interrupt processing                                    (4)
;
        and   [br:low intr_fac1],#00010000b   ;clear fk0b interrupt flag
        pop   ale
        rete
;************************************************************************
;*                                                                      *
;*       k0l input port 0 (/irq3)                                       *
;*                                                                      *
;************************************************************************
k0l_intr:
        push  ale
;
;       k0l input port interrupt processing                                    (4)
;
        and   [br:low intr_fac1],#00001000b   ;clear fk0a interrupt flag
        pop   ale
        rete
```

Source List

```
;************************************************************************
;*                                                                      *
;*      serial error (/irq2)                                            *
;*                                                                      *
;************************************************************************
serr_intr:
        push    ale
;
;       serial error interrupt processing                                          (4)
;
        and     [br:low intr_fac1],#00000100b   ;clear fserr interrupt flag
        pop     ale
        rete
;************************************************************************
;*                                                                      *
;*      serial receive (/irq2)                                          *
;*                                                                      *
;************************************************************************
srv_intr:
        push    ale
;
;       serial receive interrupt processing                                        (4)
;
        and     [br:low intr_fac1],#00000010b   ;clear fsrec interrupt flag
        pop     ale
        rete
;************************************************************************
;*                                                                      *
;*      serial transmission (/irq2)                                     *
;*                                                                      *
;************************************************************************
str_intr:
        push    ale
;
;       serial transmission interrupt processing                                   (4)
;
        and     [br:low intr_fac1],#00000001b   ;clear fstra interrupt flag
        pop     ale
        rete
;************************************************************************
;*                                                                      *
;*      stopwatch 100hz (/irq1)                                         *
;*                                                                      *
;************************************************************************
sw100_intr:
        push    ale
;
;       stopwatch 100hz interrupt processing                                       (4)
;
        and     [br:low intr_fac0],#01000000b   ;clear fsw100 interrupt flag
        pop     ale
        rete
;************************************************************************
;*                                                                      *
;*      stopwatch 10hz (/irq1)                                          *
;*                                                                      *
;************************************************************************
sw10_intr:
        push    ale
;
;       stopwatch 10hz interrupt processing                                        (4)
;
        and     [br:low intr_fac0],#00100000b   ;clear fsw10 interrupt flag
        pop     ale
        rete
```

Source List

```
;*************************************************************************
;*                                                                       *
;*       stopwatch 1hz (/irq1)                                           *
;*                                                                       *
;*************************************************************************
sw1_intr:
        push   ale
;
;       stopwatch 1hz interrupt processing                                      (4)
;
        and    [br:low intr_fac0],#00010000b   ;clear fsw1 interrupt flag
        pop    ale
        rete
;*************************************************************************
;*                                                                       *
;*       clock timer 32hz (/irq1)                                        *
;*                                                                       *
;*************************************************************************
clock32_intr:
        push   ale
;
;       clock timer 32hz interrupt processing                                   (4)
;
        and    [br:low intr_fac0],#00001000b   ;clear ftm32 interrupt flag
        pop    ale
        rete
;*************************************************************************
;*                                                                       *
;*       clock timer 8hz (/irq1)                                         *
;*                                                                       *
;*************************************************************************
clock8_intr:
        push   ale
;
;       clock timer 8hz interrupt processing                                    (4)
;
        and    [br:low intr_fac0],#00000100b   ;clear ftm8 interrupt flag
        pop    ale
        rete
;*************************************************************************
;*                                                                       *
;*       clock timer 2hz (/irq1)                                         *
;*                                                                       *
;*************************************************************************
clock2_intr:
        push   ale
;
;       clock timer 2hz interrupt processing                                    (4)
;
        and    [br:low intr_fac0],#00000010b   ;clear ftm2 interrupt flag
        pop    ale
        rete
;*************************************************************************
;*                                                                       *
;*       clock timer 1hz (/irq1)                                         *
;*                                                                       *
;*************************************************************************
clock1_intr:
        push   ale
;
;       clock timer 1hz interrupt processing                                    (4)
;
        and    [br:low intr_fac0],#00000001b   ;clear ftm1 interrupt flag
        pop    ale
        rete
;
        end
```

17 EXPANDED MODE

Specifications

Memory access in expanded mode

(1) common, common_sub, ex_sub: Access for program memory outside logical space

Branches to a bank outside logical space by setting NB register.

(2) ex_ram, move_data: Data block transfer between pages

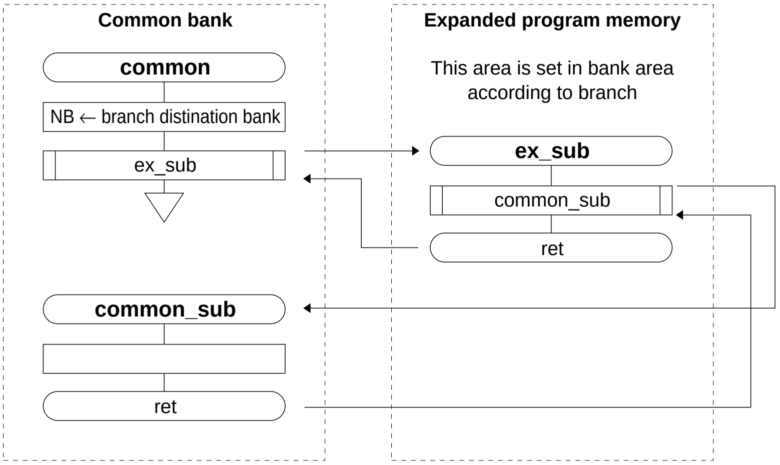
By setting expand page register, copies data (64K bytes) in page 1 to page 2. (Register indirect addressing)

(3) ex_access: Access for data outside page

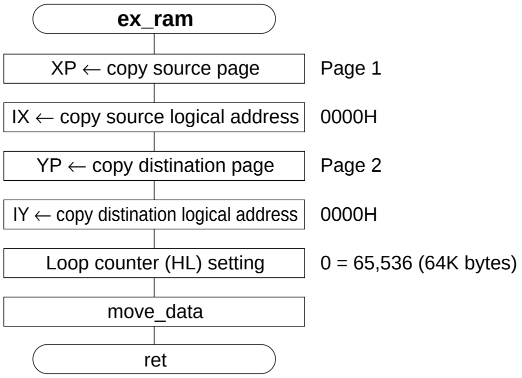
Accesses a data memory area outside of the current page using expand page register.

Flowchart

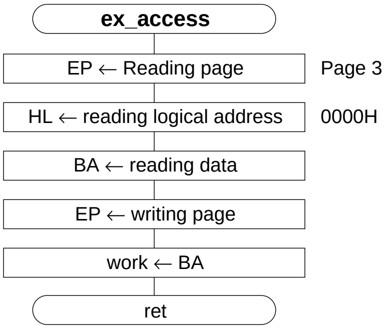
(1) Access for program memory outside logical space



(2) Data block transfer between pages

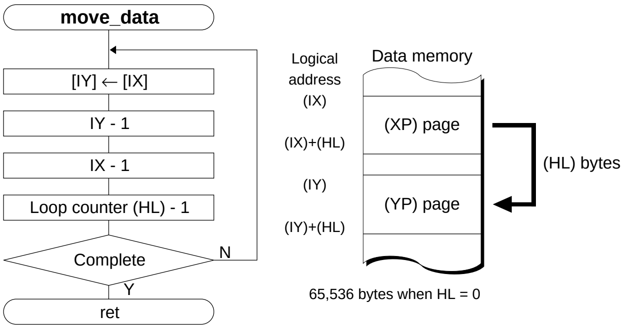

ex_ram
XP ← copy source page
Page 1
IX ← copy source logical address
0000H
YP ← copy distination page
Page 2
IY ← copy distination logical address
0000H
Loop counter (HL) setting
0 = 65,536 (64K bytes)
move_data
ret

(3) Access for data outside page



Flowchart

(3) Access for data outside page



Notes

(1) "boc" is the unary operator which calculates a bank value from the physical address.
"loc" is the unary operator which calculates a logical address in bank from the physical address.

(2) "pod" is the unary operator which calculates a page value from the physical address.
"lod" is the unary operator which calculates a logical address in page from the physical address.

Source List

```
        public       common,common_sub
        public       ex_sub
```

(1) Access for program memory outside logical space

```
        code
;
;************************************************************************
;*                                                                      *
;*       common area (bank0 = 000000h -> 007fffh) example               *
;*                                                                      *
;************************************************************************
common:
;
        ld    nb,#boc ex_sub                  ;set new bank (external area)
        carl  ex_sub                          ;external bank sub routine call ***
;
;************************************************************************
;*                                                                      *
;*       common sub-routine                                             *
;*                                                                      *
;************************************************************************
common_sub:
;
        ret
;************************************************************************
;*                                                                      *
;*       external  area (bank1 -> 15 = 008000h -> 07ffffh) example      *
;*                                                                      *
;************************************************************************
ex_sub:
;
        carl  common_sub                      ;common bank sub routine call ***
;
        ret
;
        end
```

(1)

Source List

```
;
src_data equ    10000h                          ;source data (page1=010000h -> 01ffffh)
dst_data equu   20000h                          ;destination data (page2=020000h -> 02ffffh)
;
ex_work  equ    30000h                          ;external work area
;
work:    dw     [1]
         code
;
```

(2) Data block transfer between pages

```
;*************************************************************************
;*                                                                       *
;*       external ram page control                                       *
;*                                                                       *
;*************************************************************************
ex_ram:
         ld     xp,#pod src_data                ;source data page address
         ld     ix,#lod src_data                ;source data logical top address
         ld     yp,#pod dst_data                ;destination data page address
         ld     iy,#lod dst_data                ;destination data logical top address
         ld     hl,#0                           ;0 = 65,536 (64k byte)
         carl   move_data
         ret
;*************************************************************************
;*                                                                       *
;*       move block data                                                 *      (2)
;*                                                                       *
;*************************************************************************
move_data:
         ld     [iy],[ix]
         inc    iy
         inc    ix
         dec    hl
         jrs    nz,move_data
;
         ret
```

(3) Access for data outside page

```
;*************************************************************************
;*                                                                       *
;*       external page data read and write                               *
;*                                                                       *
;*************************************************************************
ex_access:
         ld     ep,#pod ex_work
         ld     hl,#lod ex_work
         ld     ba,[hl]                                                         (3)
         ld     ep,#pod work
         ld     [lod work],ba
         ret
         end
```

Appendix A Table of E0C88112/88104 Input/Output Port Terminals

Terminal configuration depending on model		E0C88112 E0C88104	K00~07	K10 EVIN	K11 $\overline{\text{BREQ}}$	R00~07 A0~7	R10~17 A8~15	R20~22 A16~18	R23 $\overline{\text{RD}}$	R24 $\overline{\text{WR}}$	R25	R26	R27 TOUT	R30 $\overline{\text{CE0}}$	R31 $\overline{\text{CE1}}$	R32 $\overline{\text{CE2}}$	R33 $\overline{\text{CE3}}$	R34 FOUT	R35~37	R50 BZ	R51 $\overline{\text{BACK}}$	P00~07 D0~7	P10 SIN	P11 SOUT	P12 $\overline{\text{SCLK}}$	P13 $\overline{\text{SRDY}}$	P14 CMPP0	P15 CMPM0	P16 CMPP1	P17 CMPM1
CPU mode	Bus mode	Special output	Input port terminals			Output port terminals																I/O port terminals								
			K00~07	K10	K11	R00~07	R10~17	R20~22	R23	R24	R25	R26	R27	R30	R31	R32	R33	R34	R35~37	R50	R51	P00~07	P10	P11	P12	P13	P14	P15	P16	P17
MCU	Single chip	(Initial setting)	K00~07	K10	K11	R00~07	R10~17	R20~22	R23	R24	R25	R26	R27	R30	R31	R32	R33	R34	R35~37	R50	R51	P00~07	P10	P11	P12	P13	P14	P15	P16	P17
		$\overline{\text{CE0}}$ output (invalid)	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑	↑	↑	↑		↑			↑								
		$\overline{\text{CE1}}$ output (invalid)	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑	↑	↑	↑		↑			↑								
		$\overline{\text{CE2}}$ output (invalid)	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑	↑	↑	↑		↑			↑								
		$\overline{\text{CE3}}$ output (invalid)	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑	↑	↑	↑		↑			↑								
		Bus release	↑	↑	$\overline{\text{BREQ}}$	↑	↑	↑	↑	↑	↑	↑		↑	↑	↑	↑		↑		$\overline{\text{BACK}}$	↑								
		Serial interface	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑	↑	↑	↑		↑			↑	SIN	SOUT	$\overline{\text{SCLK}}$	$\overline{\text{SRDY}}$				
		Comparator 0	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑	↑	↑	↑		↑			↑					CMPP0	CMPM0		
		Comparator 1	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑	↑	↑	↑		↑			↑							CMPP1	CMPM1
		CL output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑	↑	↑	↑		↑			↑								
		FR output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑	↑	↑	↑		↑			↑								
		TOUT output	↑	↑		↑	↑	↑	↑	↑	↑	↑	TOUT	↑	↑	↑	↑		↑			↑								
		FOUT output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑	↑	↑	↑	FOUT	↑			↑								
		BZ output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑	↑	↑	↑		↑	BZ		↑								
	Expanded 64K	(No special output)	K00~07	K10	K11	A0~7	A8~15	R20~22	$\overline{\text{RD}}$	$\overline{\text{WR}}$	R25	R26	R27	R30	R31	R32	R33	R34	R35~37	R50	R51	D0~7	P10	P11	P12	P13	P14	P15	P16	P17
		$\overline{\text{CE0}}$ output	↑	↑		↑	↑	↑	↑	↑	↑	↑		$\overline{\text{CE0}}$					↑			↑								
		$\overline{\text{CE1}}$ output	↑	↑		↑	↑	↑	↑	↑	↑	↑			$\overline{\text{CE1}}$				↑			↑								
		$\overline{\text{CE2}}$ output	↑	↑		↑	↑	↑	↑	↑	↑	↑				$\overline{\text{CE2}}$			↑			↑								
		$\overline{\text{CE3}}$ output	↑	↑		↑	↑	↑	↑	↑	↑	↑					$\overline{\text{CE3}}$		↑			↑								
		Bus release	↑	↑	$\overline{\text{BREQ}}$	↑	↑	↑	↑	↑	↑	↑							↑		$\overline{\text{BACK}}$	↑								
		Serial interface	↑	↑		↑	↑	↑	↑	↑	↑	↑							↑			↑	SIN	SOUT	$\overline{\text{SCLK}}$	$\overline{\text{SRDY}}$				
		Comparator 0	↑	↑		↑	↑	↑	↑	↑	↑	↑							↑			↑					CMPP0	CMPM0		
		Comparator 1	↑	↑		↑	↑	↑	↑	↑	↑	↑							↑			↑							CMPP1	CMPM1
		CL output	↑	↑		↑	↑	↑	↑	↑	↑	↑							↑			↑								
		FR output	↑	↑		↑	↑	↑	↑	↑	↑	↑							↑			↑								
		TOUT output	↑	↑		↑	↑	↑	↑	↑	↑	↑	TOUT						↑			↑								
		FOUT output	↑	↑		↑	↑	↑	↑	↑	↑	↑						FOUT	↑			↑								
		BZ output	↑	↑		↑	↑	↑	↑	↑	↑	↑							↑	BZ		↑								
	Expanded 512K (MIN & MAX)	(No special output)	K00~07	K10	K11	A0~7	A8~15	A16~18	$\overline{\text{RD}}$	$\overline{\text{WR}}$	R25	R26	R27	R30	R31	R32	R33	R34	R35~37	R50	R51	D0~7	P10	P11	P12	P13	P14	P15	P16	P17
		$\overline{\text{CE0}}$ output	↑	↑		↑	↑	↑	↑	↑	↑	↑		$\overline{\text{CE0}}$					↑			↑								
		$\overline{\text{CE1}}$ output	↑	↑		↑	↑	↑	↑	↑	↑	↑			$\overline{\text{CE1}}$				↑			↑								
		$\overline{\text{CE2}}$ output	↑	↑		↑	↑	↑	↑	↑	↑	↑				$\overline{\text{CE2}}$			↑			↑								
		$\overline{\text{CE3}}$ output	↑	↑		↑	↑	↑	↑	↑	↑	↑					$\overline{\text{CE3}}$		↑			↑								
		Bus release	↑	↑	$\overline{\text{BREQ}}$	↑	↑	↑	↑	↑	↑	↑							↑		$\overline{\text{BACK}}$	↑								
		Serial interface	↑	↑		↑	↑	↑	↑	↑	↑	↑							↑			↑	SIN	SOUT	$\overline{\text{SCLK}}$	$\overline{\text{SRDY}}$				
		Comparator 0	↑	↑		↑	↑	↑	↑	↑	↑	↑							↑			↑					CMPP0	CMPM0		
		Comparator 1	↑	↑		↑	↑	↑	↑	↑	↑	↑							↑			↑							CMPP1	CMPM1
		CL output	↑	↑		↑	↑	↑	↑	↑	↑	↑							↑			↑								
		FR output	↑	↑		↑	↑	↑	↑	↑	↑	↑							↑			↑								
		TOUT output	↑	↑		↑	↑	↑	↑	↑	↑	↑	TOUT						↑			↑								
		FOUT output	↑	↑		↑	↑	↑	↑	↑	↑	↑						FOUT	↑			↑								
		BZ output	↑	↑		↑	↑	↑	↑	↑	↑	↑							↑	BZ		↑								

Terminal configuration depending on model		E0C88112	K00~07	K10	K11	R00~07	R10~17	R20~22	R23	R24	R25	R26	R27	R30	R31	R32	R33	R34	R35~37	R50	R51	P00~07	P10	P11	P12	P13	P14	P15	P16	P17
		E0C88104		EVIN	$\overline{\text{BREQ}}$	A0~7	A8~15	A16~18	$\overline{\text{RD}}$	$\overline{\text{WR}}$			TOUT	$\overline{\text{CE0}}$	$\overline{\text{CE1}}$	$\overline{\text{CE2}}$	$\overline{\text{CE3}}$	FOUT		$\overline{\text{BZ}}$	$\overline{\text{BACK}}$	D0~7	SIN	SOUT	$\overline{\text{SCLK}}$	$\overline{\text{SRDY}}$	CMPP0	CMPM0	CMPP1	CMPM1
CPU mode	Bus mode	Special output	Input port terminals			Output port terminals																I/O port terminals								
			K00~07	K10	K11	R00~07	R10~17	R20~22	R23	R24	R25	R26	R27	R30	R31	R32	R33	R34	R35~37	R50	R51	P00~07	P10	P11	P12	P13	P14	P15	P16	P17
MPU	Expanded 64K	(No special output)	K00~07	K10	K11	A0~7	A8~15	R20~22	$\overline{\text{RD}}$	$\overline{\text{WR}}$	R25	R26	R27	$\overline{\text{CE0}}$	R31	R32	R33	R34	R35~37	R50	R51	D0~7	P10	P11	P12	P13	P14	P15	P16	P17
		$\overline{\text{CE0}}$ output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑					↑			↑								
		$\overline{\text{CE1}}$ output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑	$\overline{\text{CE1}}$				↑			↑								
		$\overline{\text{CE2}}$ output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑		$\overline{\text{CE2}}$			↑			↑								
		$\overline{\text{CE3}}$ output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑			$\overline{\text{CE3}}$		↑			↑								
		Bus release	↑	↑	$\overline{\text{BREQ}}$	↑	↑	↑	↑	↑	↑	↑		↑					↑		$\overline{\text{BACK}}$	↑								
		Serial interface	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑					↑			↑	SIN	SOUT	$\overline{\text{SCLK}}$	$\overline{\text{SRDY}}$				
		Comparator 0	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑					↑			↑					CMPP0	CMPM0		
		Comparator 1	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑					↑			↑							CMPP1	CMPM1
		CL output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑					↑			↑								
		FR output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑					↑			↑								
		TOUT output	↑	↑		↑	↑	↑	↑	↑	↑	↑	TOUT	↑					↑			↑								
		FOUT output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑				FOUT	↑			↑								
		BZ output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑					↑	BZ		↑								
	Expanded 512K (MIN & MAX)	(No special output)	K00~07	K10	K11	A0~7	A8~15	A16~18	$\overline{\text{RD}}$	$\overline{\text{WR}}$	R25	R26	R27	$\overline{\text{CE0}}$	R31	R32	R33	R34	R35~37	R50	R51	D0~7	P10	P11	P12	P13	P14	P15	P16	P17
		$\overline{\text{CE0}}$ output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑					↑			↑								
		$\overline{\text{CE1}}$ output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑	$\overline{\text{CE1}}$				↑			↑								
		$\overline{\text{CE2}}$ output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑		$\overline{\text{CE2}}$			↑			↑								
		$\overline{\text{CE3}}$ output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑			$\overline{\text{CE3}}$		↑			↑								
		Bus release	↑	↑	$\overline{\text{BREQ}}$	↑	↑	↑	↑	↑	↑	↑		↑					↑		$\overline{\text{BACK}}$	↑								
		Serial interface	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑					↑			↑	SIN	SOUT	$\overline{\text{SCLK}}$	$\overline{\text{SRDY}}$				
		Comparator 0	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑					↑			↑					CMPP0	CMPM0		
		Comparator 1	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑					↑			↑							CMPP1	CMPM1
		CL output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑					↑			↑								
		FR output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑					↑			↑								
		TOUT output	↑	↑		↑	↑	↑	↑	↑	↑	↑	TOUT	↑					↑			↑								
		FOUT output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑				FOUT	↑			↑								
		BZ output	↑	↑		↑	↑	↑	↑	↑	↑	↑		↑					↑	BZ		↑								

Note: • Blank items will be decided according to other special output settings in the same row.

Appendix B Instruction List

8-bit Trnsfer Instructions (1/3)

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
LD	A,A	40	A←A	1	1	–	–	–	–	–	–	–	–	
	A,B	41	A←B	1	1	–	–	–	–	–	–	–	–	
	A,L	42	A←L	1	1	–	–	–	–	–	–	–	–	
	A,H	43	A←H	1	1	–	–	–	–	–	–	–	–	
	A,BR	CE,C0	A←BR	2	2	–	–	–	–	–	–	–	–	
	A,SC	CE,C1	A←SC	2	2	–	–	–	–	–	–	–	–	
	A,#nn	B0,nn	A←nn	2	2	–	–	–	–	–	–	–	–	
	A,[BR:*ll*]	44,*ll*	A←[BR:*ll*]	3	2	–	–	–	–	–	–	–	–	
	A,[hh*ll*]	CE,D0,*ll*,hh	A←[hh*ll*]	5	4	–	–	–	–	–	–	–	–	
	A,[HL]	45	A←[HL]	2	1	–	–	–	–	–	–	–	–	
	A,[IX]	46	A←[IX]	2	1	–	–	–	–	–	–	–	–	
	A,[IY]	47	A←[IY]	2	1	–	–	–	–	–	–	–	–	
	A,[IX+dd]	CE,40,dd	A←[IX+dd]	4	3	–	–	–	–	–	–	–	–	
	A,[IY+dd]	CE,41,dd	A←[IY+dd]	4	3	–	–	–	–	–	–	–	–	
	A,[IX+L]	CE,42	A←[IX+L]	4	2	–	–	–	–	–	–	–	–	
	A,[IY+L]	CE,43	A←[IY+L]	4	2	–	–	–	–	–	–	–	–	
	A,NB	CE,C8	A←NB	2	2	–	–	–	–	–	–	–	–	*MODEL2/3 only*
	A,EP	CE,C9	A←EP	2	2	–	–	–	–	–	–	–	–	
	A,XP	CE,CA	A←XP	2	2	–	–	–	–	–	–	–	–	
	A,YP	CE,CB	A←YP	2	2	–	–	–	–	–	–	–	–	
LD	B,A	48	B←A	1	1	–	–	–	–	–	–	–	–	
	B,B	49	B←B	1	1	–	–	–	–	–	–	–	–	
	B,L	4A	B←L	1	1	–	–	–	–	–	–	–	–	
	B,H	4B	B←H	1	1	–	–	–	–	–	–	–	–	
	B,#nn	B1,nn	B←nn	2	2	–	–	–	–	–	–	–	–	
	B,[BR:*ll*]	4C,*ll*	B←[BR:*ll*]	3	2	–	–	–	–	–	–	–	–	
	B,[hh*ll*]	CE,D1,*ll*,hh	B←[hh*ll*]	5	4	–	–	–	–	–	–	–	–	
	B,[HL]	4D	B←[HL]	2	1	–	–	–	–	–	–	–	–	
	B,[IX]	4E	B←[IX]	2	1	–	–	–	–	–	–	–	–	
	B,[IY]	4F	B←[IY]	2	1	–	–	–	–	–	–	–	–	
	B,[IX+dd]	CE,48,dd	B←[IX+dd]	4	3	–	–	–	–	–	–	–	–	
	B,[IY+dd]	CE,49,dd	B←[IY+dd]	4	3	–	–	–	–	–	–	–	–	
	B,[IX+L]	CE,4A	B←[IX+L]	4	2	–	–	–	–	–	–	–	–	
	B,[IY+L]	CE,4B	B←[IY+L]	4	2	–	–	–	–	–	–	–	–	
LD	L,A	50	L←A	1	1	–	–	–	–	–	–	–	–	
	L,B	51	L←B	1	1	–	–	–	–	–	–	–	–	
	L,L	52	L←L	1	1	–	–	–	–	–	–	–	–	
	L,H	53	L←H	1	1	–	–	–	–	–	–	–	–	
	L,#nn	B2,nn	L←nn	2	2	–	–	–	–	–	–	–	–	
	L,[BR:*ll*]	54,*ll*	L←[BR:*ll*]	3	2	–	–	–	–	–	–	–	–	
	L,[hh*ll*]	CE,D2,*ll*,hh	L←[hh*ll*]	5	4	–	–	–	–	–	–	–	–	
	L,[HL]	55	L←[HL]	2	1	–	–	–	–	–	–	–	–	
	L,[IX]	56	L←[IX]	2	1	–	–	–	–	–	–	–	–	
	L,[IY]	57	L←[IY]	2	1	–	–	–	–	–	–	–	–	
	L,[IX+dd]	CE,50,dd	L←[IX+dd]	4	3	–	–	–	–	–	–	–	–	
	L,[IY+dd]	CE,51,dd	L←[IY+dd]	4	3	–	–	–	–	–	–	–	–	
	L,[IX+L]	CE,52	L←[IX+L]	4	2	–	–	–	–	–	–	–	–	
	L,[IY+L]	CE,53	L←[IY+L]	4	2	–	–	–	–	–	–	–	–	

* New code bank register NB and expand page registers EP/XP/YP are set only for MODEL2/3. In MODEL0/1, instructions that access these registers cannot be used.

8-bit Trnsfer Instructions (2/3)

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
LD	H,A	58	H←A	1	1	–	–	–	–	–	–	–	–	
	H,B	59	H←B	1	1	–	–	–	–	–	–	–	–	
	H,L	5A	H←L	1	1	–	–	–	–	–	–	–	–	
	H,H	5B	H←H	1	1	–	–	–	–	–	–	–	–	
	H,#nn	B3,nn	H←nn	2	2	–	–	–	–	–	–	–	–	
	H,[BR:*ll*]	5C,*ll*	H←[BR:*ll*]	3	2	–	–	–	–	–	–	–	–	
	H,[hh*ll*]	CE,D3,*ll*,hh	H←[hh*ll*]	5	4	–	–	–	–	–	–	–	–	
	H,[HL]	5D	H←[HL]	2	1	–	–	–	–	–	–	–	–	
	H,[IX]	5E	H←[IX]	2	1	–	–	–	–	–	–	–	–	
	H,[IY]	5F	H←[IY]	2	1	–	–	–	–	–	–	–	–	
	H,[IX+dd]	CE,58,dd	H←[IX+dd]	4	3	–	–	–	–	–	–	–	–	
	H,[IY+dd]	CE,59,dd	H←[IY+dd]	4	3	–	–	–	–	–	–	–	–	
	H,[IX+L]	CE,5A	H←[IX+L]	4	2	–	–	–	–	–	–	–	–	
	H,[IY+L]	CE,5B	H←[IY+L]	4	2	–	–	–	–	–	–	–	–	
LD	BR,A	CE,C2	BR←A	2	2	–	–	–	–	–	–	–	–	
	BR,#hh	B4,hh	BR←hh	2	2	–	–	–	–	–	–	–	–	
LD	SC,A	CE,C3	SC←A	3	2	↕	↕	↕	↕	↕	↕	↕	↕	
	SC,#nn	9F,nn	SC←nn	3	2	↕	↕	↕	↕	↕	↕	↕	↕	
LD	[BR:*ll*],A	78,*ll*	[BR:*ll*]←A	3	2	–	–	–	–	–	–	–	–	
	[BR:*ll*],B	79,*ll*	[BR:*ll*]←B	3	2	–	–	–	–	–	–	–	–	
	[BR:*ll*],L	7A,*ll*	[BR:*ll*]←L	3	2	–	–	–	–	–	–	–	–	
	[BR:*ll*],H	7B,*ll*	[BR:*ll*]←H	3	2	–	–	–	–	–	–	–	–	
	[BR:*ll*],#nn	DD,*ll*,nn	[BR:*ll*]←nn	4	3	–	–	–	–	–	–	–	–	
	[BR:*ll*],[HL]	7D,*ll*	[BR:*ll*]←[HL]	4	2	–	–	–	–	–	–	–	–	
	[BR:*ll*],[IX]	7E,*ll*	[BR:*ll*]←[IX]	4	2	–	–	–	–	–	–	–	–	
	[BR:*ll*],[IY]	7F,*ll*	[BR:*ll*]←[IY]	4	2	–	–	–	–	–	–	–	–	
LD	[hh*ll*],A	CE,D4,*ll*,hh	[hh*ll*]←A	5	4	–	–	–	–	–	–	–	–	
	[hh*ll*],B	CE,D5,*ll*,hh	[hh*ll*]←B	5	4	–	–	–	–	–	–	–	–	
	[hh*ll*],L	CE,D6,*ll*,hh	[hh*ll*]←L	5	4	–	–	–	–	–	–	–	–	
	[hh*ll*],H	CE,D7,*ll*,hh	[hh*ll*]←H	5	4	–	–	–	–	–	–	–	–	
LD	[HL],A	68	[HL]←A	2	1	–	–	–	–	–	–	–	–	
	[HL],B	69	[HL]←B	2	1	–	–	–	–	–	–	–	–	
	[HL],L	6A	[HL]←L	2	1	–	–	–	–	–	–	–	–	
	[HL],H	6B	[HL]←H	2	1	–	–	–	–	–	–	–	–	
	[HL],#nn	B5,nn	[HL]←nn	3	2	–	–	–	–	–	–	–	–	
	[HL],[BR:*ll*]	6C,*ll*	[HL]←[BR:*ll*]	4	2	–	–	–	–	–	–	–	–	
	[HL],[HL]	6D	[HL]←[HL]	3	1	–	–	–	–	–	–	–	–	
	[HL],[IX]	6E	[HL]←[IX]	3	1	–	–	–	–	–	–	–	–	
	[HL],[IY]	6F	[HL]←[IY]	3	1	–	–	–	–	–	–	–	–	
	[HL],[IX+dd]	CE,60,dd	[HL]←[IX+dd]	5	3	–	–	–	–	–	–	–	–	
	[HL],[IY+dd]	CE,61,dd	[HL]←[IY+dd]	5	3	–	–	–	–	–	–	–	–	
	[HL],[IX+L]	CE,62	[HL]←[IX+L]	5	2	–	–	–	–	–	–	–	–	
	[HL],[IY+L]	CE,63	[HL]←[IY+L]	5	2	–	–	–	–	–	–	–	–	
LD	[IX],A	60	[IX]←A	2	1	–	–	–	–	–	–	–	–	
	[IX],B	61	[IX]←B	2	1	–	–	–	–	–	–	–	–	
	[IX],L	62	[IX]←L	2	1	–	–	–	–	–	–	–	–	
	[IX],H	63	[IX]←H	2	1	–	–	–	–	–	–	–	–	
	[IX],#nn	B6,nn	[IX]←nn	3	2	–	–	–	–	–	–	–	–	

8-bit Trnsfer Instructions (3/3)

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
LD	[IX],[BR:*ll*]	64,*ll*	[IX]←[BR:*ll*]	4	2	–	–	–	–	–	–	–	–	
	[IX],[HL]	65	[IX]←[HL]	3	1	–	–	–	–	–	–	–	–	
	[IX],[IX]	66	[IX]←[IX]	3	1	–	–	–	–	–	–	–	–	
	[IX],[IY]	67	[IX]←[IY]	3	1	–	–	–	–	–	–	–	–	
	[IX],[IX+dd]	CE,68,dd	[IX]←[IX+dd]	5	3	–	–	–	–	–	–	–	–	
	[IX],[IY+dd]	CE,69,dd	[IX]←[IY+dd]	5	3	–	–	–	–	–	–	–	–	
	[IX],[IX+L]	CE,6A	[IX]←[IX+L]	5	2	–	–	–	–	–	–	–	–	
	[IX],[IY+L]	CE,6B	[IX]←[IY+L]	5	2	–	–	–	–	–	–	–	–	
LD	[IY],A	70	[IY]←A	2	1	–	–	–	–	–	–	–	–	
	[IY],B	71	[IY]←B	2	1	–	–	–	–	–	–	–	–	
	[IY],L	72	[IY]←L	2	1	–	–	–	–	–	–	–	–	
	[IY],H	73	[IY]←H	2	1	–	–	–	–	–	–	–	–	
	[IY],#nn	B7,nn	[IY]←nn	3	2	–	–	–	–	–	–	–	–	
	[IY],[BR:*ll*]	74,ll	[IY]←[BR:*ll*]	4	2	–	–	–	–	–	–	–	–	
	[IY],[HL]	75	[IY]←[HL]	3	1	–	–	–	–	–	–	–	–	
	[IY],[IX]	76	[IY]←[IX]	3	1	–	–	–	–	–	–	–	–	
	[IY],[IY]	77	[IY]←[IY]	3	1	–	–	–	–	–	–	–	–	
	[IY],[IX+dd]	CE,78,dd	[IY]←[IX+dd]	5	3	–	–	–	–	–	–	–	–	
	[IY],[IY+dd]	CE,79,dd	[IY]←[IY+dd]	5	3	–	–	–	–	–	–	–	–	
	[IY],[IX+L]	CE,7A	[IY]←[IX+L]	5	2	–	–	–	–	–	–	–	–	
	[IY],[IY+L]	CE,7B	[IY]←[IY+L]	5	2	–	–	–	–	–	–	–	–	
LD	[IX+dd],A	CE,44,dd	[IX+dd]←A	4	3	–	–	–	–	–	–	–	–	
	[IX+dd],B	CE,4C,dd	[IX+dd]←B	4	3	–	–	–	–	–	–	–	–	
	[IX+dd],L	CE,54,dd	[IX+dd]←L	4	3	–	–	–	–	–	–	–	–	
	[IX+dd],H	CE,5C,dd	[IX+dd]←H	4	3	–	–	–	–	–	–	–	–	
LD	[IY+dd],A	CE,45,dd	[IY+dd]←A	4	3	–	–	–	–	–	–	–	–	
	[IY+dd],B	CE,4D,dd	[IY+dd]←B	4	3	–	–	–	–	–	–	–	–	
	[IY+dd],L	CE,55,dd	[IY+dd]←L	4	3	–	–	–	–	–	–	–	–	
	[IY+dd],H	CE,5D,dd	[IY+dd]←H	4	3	–	–	–	–	–	–	–	–	
LD	[IX+L],A	CE,46	[IX+L]←A	4	2	–	–	–	–	–	–	–	–	
	[IX+L],B	CE,4E	[IX+L]←B	4	2	–	–	–	–	–	–	–	–	
	[IX+L],L	CE,56	[IX+L]←L	4	2	–	–	–	–	–	–	–	–	
	[IX+L],H	CE,5E	[IX+L]←H	4	2	–	–	–	–	–	–	–	–	
LD	[IY+L],A	CE,47	[IY+L]←A	4	2	–	–	–	–	–	–	–	–	
	[IY+L],B	CE,4F	[IY+L]←B	4	2	–	–	–	–	–	–	–	–	
	[IY+L],L	CE,57	[IY+L]←L	4	2	–	–	–	–	–	–	–	–	
	[IY+L],H	CE,5F	[IY+L]←H	4	2	–	–	–	–	–	–	–	–	
LD	NB,A	CE,CC	NB←A	3	2	–	–	–	–	–	–	–	–	*MODEL2/3 only*
	NB,#bb	CE,C4,bb	NB←bb	4	3	–	–	–	–	–	–	–	–	
LD	EP,A	CE,CD	EP←A	2	2	–	–	–	–	–	–	–	–	
	EP,#pp	CE,C5,pp	EP←pp	3	3	–	–	–	–	–	–	–	–	
LD	XP,A	CE,CE	XP←A	2	2	–	–	–	–	–	–	–	–	
	XP,#pp	CE,C6,pp	XP←pp	3	3	–	–	–	–	–	–	–	–	
LD	YP,A	CE,CF	YP←A	2	2	–	–	–	–	–	–	–	–	
	YP,#pp	CE,C7,pp	YP←pp	3	3	–	–	–	–	–	–	–	–	
EX	A,B	CC	A↔B	2	1	–	–	–	–	–	–	–	–	
	A,[HL]	CD	A↔[HL]	3	1	–	–	–	–	–	–	–	–	
SWAP	A	F6	A$_{(H)}$↔A$_{(L)}$	2	1	–	–	–	–	–	–	–	–	
	[HL]	F7	[HL]$_{(H)}$↔[HL]$_{(L)}$	3	1	–	–	–	–	–	–	–	–	

* New code bank register NB and expand page registers EP/XP/YP are set only for MODEL2/3. In MODEL0/1, instructions that access these registers cannot be used.

16-bit Trnsfer Instructions (1/2)

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
LD	BA,BA	CF,E0	BA←BA	2	2	–	–	–	–	–	–	–	–	
	BA,HL	CF,E1	BA←HL	2	2	–	–	–	–	–	–	–	–	
	BA,IX	CF,E2	BA←IX	2	2	–	–	–	–	–	–	–	–	
	BA,IY	CF,E3	BA←IY	2	2	–	–	–	–	–	–	–	–	
	BA,SP	CF,F8	BA←SP	2	2	–	–	–	–	–	–	–	–	
	BA,PC	CF,F9	BA←PC+2	2	2	–	–	–	–	–	–	–	–	
	BA,#mmnn	C4,nn,mm	BA←mmnn	3	3	–	–	–	–	–	–	–	–	
	BA,[hh*ll*]	B8,*ll*,hh	A←[hh*ll*], B←[hh*ll*+1]	5	3	–	–	–	–	–	–	–	–	
	BA,[HL]	CF,C0	A←[HL], B←[HL+1]	5	2	–	–	–	–	–	–	–	–	
	BA,[IX]	CF,D0	A←[IX], B←[IX+1]	5	2	–	–	–	–	–	–	–	–	
	BA,[IY]	CF,D8	A←[IY], B←[IY+1]	5	2	–	–	–	–	–	–	–	–	
	BA,[SP+dd]	CF,70,dd	A←[SP+dd], B←[SP+dd+1]	6	3	–	–	–	–	–	–	–	–	
LD	HL,BA	CF,E4	HL←BA	2	2	–	–	–	–	–	–	–	–	
	HL,HL	CF,E5	HL←HL	2	2	–	–	–	–	–	–	–	–	
	HL,IX	CF,E6	HL←IX	2	2	–	–	–	–	–	–	–	–	
	HL,IY	CF,E7	HL←IY	2	2	–	–	–	–	–	–	–	–	
	HL,SP	CF,F4	HL←SP	2	2	–	–	–	–	–	–	–	–	
	HL,PC	CF,F5	HL←PC+2	2	2	–	–	–	–	–	–	–	–	
	HL,#mmnn	C5,nn,mm	HL←mmnn	3	3	–	–	–	–	–	–	–	–	
	HL,[hh*ll*]	B9,*ll*,hh	L←[hh*ll*], H←[hh*ll*+1]	5	3	–	–	–	–	–	–	–	–	
	HL,[HL]	CF,C1	L←[HL], H←[HL+1]	5	2	–	–	–	–	–	–	–	–	
	HL,[IX]	CF,D1	L←[IX], H←[IX+1]	5	2	–	–	–	–	–	–	–	–	
	HL,[IY]	CF,D9	L←[IY], H←[IY+1]	5	2	–	–	–	–	–	–	–	–	
	HL,[SP+dd]	CF,71,dd	L←[SP+dd], H←[SP+dd+1]	6	3	–	–	–	–	–	–	–	–	
LD	IX,BA	CF,E8	IX←BA	2	2	–	–	–	–	–	–	–	–	
	IX,HL	CF,E9	IX←HL	2	2	–	–	–	–	–	–	–	–	
	IX,IX	CF,EA	IX←IX	2	2	–	–	–	–	–	–	–	–	
	IX,IY	CF,EB	IX←IY	2	2	–	–	–	–	–	–	–	–	
	IX,SP	CF,FA	IX←SP	2	2	–	–	–	–	–	–	–	–	
	IX,#mmnn	C6,nn,mm	IX←mmnn	3	3	–	–	–	–	–	–	–	–	
	IX,[hh*ll*]	BA,*ll*,hh	IX(L)←[hh*ll*], IX(H)←[hh*ll*+1]	5	3	–	–	–	–	–	–	–	–	
	IX,[HL]	CF,C2	IX(L)←[HL], IX(H)←[HL+1]	5	2	–	–	–	–	–	–	–	–	
	IX,[IX]	CF,D2	IX(L)←[IX], IX(H)←[IX+1]	5	2	–	–	–	–	–	–	–	–	
	IX,[IY]	CF,DA	IX(L)←[IY], IX(H)←[IY+1]	5	2	–	–	–	–	–	–	–	–	
	IX,[SP+dd]	CF,72,dd	IX(L)←[SP+dd], IX(H)←[SP+dd+1]	6	3	–	–	–	–	–	–	–	–	
LD	IY,BA	CF,EC	IY←BA	2	2	–	–	–	–	–	–	–	–	
	IY,HL	CF,ED	IY←HL	2	2	–	–	–	–	–	–	–	–	
	IY,IX	CF,EE	IY←IX	2	2	–	–	–	–	–	–	–	–	
	IY,IY	CF,EF	IY←IY	2	2	–	–	–	–	–	–	–	–	
	IY,SP	CF,FE	IY←SP	2	2	–	–	–	–	–	–	–	–	
	IY,#mmnn	C7,nn,mm	IY←mmnn	3	3	–	–	–	–	–	–	–	–	
	IY,[hh*ll*]	BB,*ll*,hh	IY(L)←[hh*ll*], IY(H)←[hh*ll*+1]	5	3	–	–	–	–	–	–	–	–	
	IY,[HL]	CF,C3	IY(L)←[HL], IY(H)←[HL+1]	5	2	–	–	–	–	–	–	–	–	
	IY,[IX]	CF,D3	IY(L)←[IX], IY(H)←[IX+1]	5	2	–	–	–	–	–	–	–	–	
	IY,[IY]	CF,DB	IY(L)←[IY], IY(H)←[IY+1]	5	2	–	–	–	–	–	–	–	–	
	IY,[SP+dd]	CF,73,dd	IY(L)←[SP+dd], IY(H)←[SP+dd+1]	6	3	–	–	–	–	–	–	–	–	

16-bit Trnsfer Instructions (2/2)

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
LD	SP,BA	CF,F0	SP←BA	2	2	–	–	–	–	–	–	–	–	
	SP,[hh*ll*]	CF,78,*ll*,hh	SP(L)←[hh*ll*], SP(H)←[hh*ll*+1]	6	4	–	–	–	–	–	–	–	–	
	SP,HL	CF,F1	SP←HL	2	2	–	–	–	–	–	–	–	–	
	SP,IX	CF,F2	SP←IX	2	2	–	–	–	–	–	–	–	–	
	SP,IY	CF,F3	SP←IY	2	2	–	–	–	–	–	–	–	–	
	SP,#mmnn	CF,6E,nn,mm	SP←mmnn	4	4	–	–	–	–	–	–	–	–	
LD	[hh*ll*],BA	BC,*ll*,hh	[hh*ll*]←A, [hh*ll*+1]←B	5	3	–	–	–	–	–	–	–	–	
	[hh*ll*],HL	BD,*ll*,hh	[hh*ll*]←L, [hh*ll*+1]←H	5	3	–	–	–	–	–	–	–	–	
	[hh*ll*],IX	BE,*ll*,hh	[hh*ll*]←IX(L), [hh*ll*+1]←IX(H)	5	3	–	–	–	–	–	–	–	–	
	[hh*ll*],IY	BF,*ll*,hh	[hh*ll*]←IY(L), [hh*ll*+1]←IY(H)	5	3	–	–	–	–	–	–	–	–	
	[hh*ll*],SP	CF,7C,*ll*,hh	[hh*ll*]←SP(L), [hh*ll*+1]←SP(H)	6	4	–	–	–	–	–	–	–	–	
LD	[HL],BA	CF,C4	[HL]←A, [HL+1]←B	5	2	–	–	–	–	–	–	–	–	
	[HL],HL	CF,C5	[HL]←L, [HL+1]←H	5	2	–	–	–	–	–	–	–	–	
	[HL],IX	CF,C6	[HL]←IX(L), [HL+1]←IX(H)	5	2	–	–	–	–	–	–	–	–	
	[HL],IY	CF,C7	[HL]←IY(L), [HL+1]←IY(H)	5	2	–	–	–	–	–	–	–	–	
LD	[IX],BA	CF,D4	[IX]←A, [IX+1]←B	5	2	–	–	–	–	–	–	–	–	
	[IX],HL	CF,D5	[IX]←L, [IX+1]←H	5	2	–	–	–	–	–	–	–	–	
	[IX],IX	CF,D6	[IX]←IX(L), [IX+1]←IX(H)	5	2	–	–	–	–	–	–	–	–	
	[IX],IY	CF,D7	[IX]←IY(L), [IX+1]←IY(H)	5	2	–	–	–	–	–	–	–	–	
LD	[IY],BA	CF,DC	[IY]←A, [IY+1]←B	5	2	–	–	–	–	–	–	–	–	
	[IY],HL	CF,DD	[IY]←L, [IY+1]←H	5	2	–	–	–	–	–	–	–	–	
	[IY],IX	CF,DE	[IY]←IX(L), [IY+1]←IX(H)	5	2	–	–	–	–	–	–	–	–	
	[IY],IY	CF,DF	[IY]←IY(L), [IY+1]←IY(H)	5	2	–	–	–	–	–	–	–	–	
LD	[SP+dd],BA	CF,74,dd	[SP+dd]←A, [SP+dd+1]←B	6	3	–	–	–	–	–	–	–	–	
	[SP+dd],HL	CF,75,dd	[SP+dd]←L, [SP+dd+1]←H	6	3	–	–	–	–	–	–	–	–	
	[SP+dd],IX	CF,76,dd	[SP+dd]←IX(L), [SP+dd+1]←IX(H)	6	3	–	–	–	–	–	–	–	–	
	[SP+dd],IY	CF,77,dd	[SP+dd]←IY(L), [SP+dd+1]←IY(H)	6	3	–	–	–	–	–	–	–	–	
EX	BA,HL	C8	BA↔HL	3	1	–	–	–	–	–	–	–	–	
	BA,IX	C9	BA↔IX	3	1	–	–	–	–	–	–	–	–	
	BA,IY	CA	BA↔IY	3	1	–	–	–	–	–	–	–	–	
	BA,SP	CB	BA↔SP	3	1	–	–	–	–	–	–	–	–	

8-bit Arithmetic and Logic Operation Instructions (1/4)

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
ADD	A,A	00	A←A+A	2	1	–	–	★	★	↕	↕	↕	↕	
	A,B	01	A←A+B	2	1	–	–	★	★	↕	↕	↕	↕	
	A,#nn	02,nn	A←A+nn	2	2	–	–	★	★	↕	↕	↕	↕	
	A,[BR:*ll*]	04,*ll*	A←A+[BR:*ll*]	3	2	–	–	★	★	↕	↕	↕	↕	
	A,[hh*ll*]	05,*ll*,hh	A←A+[hh*ll*]	4	3	–	–	★	★	↕	↕	↕	↕	
	A,[HL]	03	A←A+[HL]	2	1	–	–	★	★	↕	↕	↕	↕	
	A,[IX]	06	A←A+[IX]	2	1	–	–	★	★	↕	↕	↕	↕	
	A,[IY]	07	A←A+[IY]	2	1	–	–	★	★	↕	↕	↕	↕	
	A,[IX+dd]	CE,00,dd	A←A+[IX+dd]	4	3	–	–	★	★	↕	↕	↕	↕	
	A,[IY+dd]	CE,01,dd	A←A+[IY+dd]	4	3	–	–	★	★	↕	↕	↕	↕	
	A,[IX+L]	CE,02	A←A+[IX+L]	4	2	–	–	★	★	↕	↕	↕	↕	
	A,[IY+L]	CE,03	A←A+[IY+L]	4	2	–	–	★	★	↕	↕	↕	↕	
	[HL],A	CE,04	[HL]←[HL]+A	4	2	–	–	★	★	↕	↕	↕	↕	
	[HL],#nn	CE,05,nn	[HL]←[HL]+nn	5	3	–	–	★	★	↕	↕	↕	↕	
	[HL],[IX]	CE,06	[HL]←[HL]+[IX]	5	2	–	–	★	★	↕	↕	↕	↕	
	[HL],[IY]	CE,07	[HL]←[HL]+[IY]	5	2	–	–	★	★	↕	↕	↕	↕	
ADC	A,A	08	A←A+A+C	2	1	–	–	★	★	↕	↕	↕	↕	
	A,B	09	A←A+B+C	2	1	–	–	★	★	↕	↕	↕	↕	
	A,#nn	0A,nn	A←A+nn+C	2	2	–	–	★	★	↕	↕	↕	↕	
	A,[BR:*ll*]	0C,*ll*	A←A+[BR:*ll*]+C	3	2	–	–	★	★	↕	↕	↕	↕	
	A,[hh*ll*]	0D,*ll*,hh	A←A+[hh*ll*]+C	4	3	–	–	★	★	↕	↕	↕	↕	
	A,[HL]	0B	A←A+[HL]+C	2	1	–	–	★	★	↕	↕	↕	↕	
	A,[IX]	0E	A←A+[IX]+C	2	1	–	–	★	★	↕	↕	↕	↕	
	A,[IY]	0F	A←A+[IY]+C	2	1	–	–	★	★	↕	↕	↕	↕	
	A,[IX+dd]	CE,08,dd	A←A+[IX+dd]+C	4	3	–	–	★	★	↕	↕	↕	↕	
	A,[IY+dd]	CE,09,dd	A←A+[IY+dd]+C	4	3	–	–	★	★	↕	↕	↕	↕	
	A,[IX+L]	CE,0A	A←A+[IX+L]+C	4	2	–	–	★	★	↕	↕	↕	↕	
	A,[IY+L]	CE,0B	A←A+[IY+L]+C	4	2	–	–	★	★	↕	↕	↕	↕	
	[HL],A	CE,0C	[HL]←[HL]+A+C	4	2	–	–	★	★	↕	↕	↕	↕	
	[HL],#nn	CE,0D,nn	[HL]←[HL]+nn+C	5	3	–	–	★	★	↕	↕	↕	↕	
	[HL],[IX]	CE,0E	[HL]←[HL]+[IX]+C	5	2	–	–	★	★	↕	↕	↕	↕	
	[HL],[IY]	CE,0F	[HL]←[HL]+[IY]+C	5	2	–	–	★	★	↕	↕	↕	↕	
SUB	A,A	10	A←A-A	2	1	–	–	★	★	↕	↕	↕	↕	
	A,B	11	A←A-B	2	1	–	–	★	★	↕	↕	↕	↕	
	A,#nn	12,nn	A←A-nn	2	2	–	–	★	★	↕	↕	↕	↕	
	A,[BR:*ll*]	14,*ll*	A←A-[BR:*ll*]	3	2	–	–	★	★	↕	↕	↕	↕	
	A,[hh*ll*]	15,*ll*,hh	A←A-[hh*ll*]	4	3	–	–	★	★	↕	↕	↕	↕	
	A,[HL]	13	A←A-[HL]	2	1	–	–	★	★	↕	↕	↕	↕	
	A,[IX]	16	A←A-[IX]	2	1	–	–	★	★	↕	↕	↕	↕	
	A,[IY]	17	A←A-[IY]	2	1	–	–	★	★	↕	↕	↕	↕	
	A,[IX+dd]	CE,10,dd	A←A-[IX+dd]	4	3	–	–	★	★	↕	↕	↕	↕	
	A,[IY+dd]	CE,11,dd	A←A-[IY+dd]	4	3	–	–	★	★	↕	↕	↕	↕	
	A,[IX+L]	CE,12	A←A-[IX+L]	4	2	–	–	★	★	↕	↕	↕	↕	
	A,[IY+L]	CE,13	A←A-[IY+L]	4	2	–	–	★	★	↕	↕	↕	↕	
	[HL],A	CE,14	[HL]←[HL]-A	4	2	–	–	★	★	↕	↕	↕	↕	
	[HL],#nn	CE,15,nn	[HL]←[HL]-nn	5	3	–	–	★	★	↕	↕	↕	↕	
	[HL],[IX]	CE,16	[HL]←[HL]-[IX]	5	2	–	–	★	★	↕	↕	↕	↕	
	[HL],[IY]	CE,17	[HL]←[HL]-[IY]	5	2	–	–	★	★	↕	↕	↕	↕	

8-bit Arithmetic and Logic Operation Instructions (2/4)

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
SBC	A,A	18	A←A-A-C	2	1	–	–	★	★	↕	↕	↕	↕	
	A,B	19	A←A-B-C	2	1	–	–	★	★	↕	↕	↕	↕	
	A,#nn	1A,nn	A←A-nn-C	2	2	–	–	★	★	↕	↕	↕	↕	
	A,[BR:*ll*]	1C,*ll*	A←A-[BR:*ll*]-C	3	2	–	–	★	★	↕	↕	↕	↕	
	A,[hh*ll*]	1D,*ll*,hh	A←A-[hh*ll*]-C	4	3	–	–	★	★	↕	↕	↕	↕	
	A,[HL]	1B	A←A-[HL]-C	2	1	–	–	★	★	↕	↕	↕	↕	
	A,[IX]	1E	A←A-[IX]-C	2	1	–	–	★	★	↕	↕	↕	↕	
	A,[IY]	1F	A←A-[IY]-C	2	1	–	–	★	★	↕	↕	↕	↕	
	A,[IX+dd]	CE,18,dd	A←A-[IX+dd]-C	4	3	–	–	★	★	↕	↕	↕	↕	
	A,[IY+dd]	CE,19,dd	A←A-[IY+dd]-C	4	3	–	–	★	★	↕	↕	↕	↕	
	A,[IX+L]	CE,1A	A←A-[IX+L]-C	4	2	–	–	★	★	↕	↕	↕	↕	
	A,[IY+L]	CE,1B	A←A-[IY+L]-C	4	2	–	–	★	★	↕	↕	↕	↕	
	[HL],A	CE,1C	[HL]←[HL]-A-C	4	2	–	–	★	★	↕	↕	↕	↕	
	[HL],#nn	CE,1D,nn	[HL]←[HL]-nn-C	5	3	–	–	★	★	↕	↕	↕	↕	
	[HL],[IX]	CE,1E	[HL]←[HL]-[IX]-C	5	2	–	–	★	★	↕	↕	↕	↕	
	[HL],[IY]	CE,1F	[HL]←[HL]-[IY]-C	5	2	–	–	★	★	↕	↕	↕	↕	
AND	A,A	20	A←A∧A	2	1	–	–	–	–	↕	–	–	↕	
	A,B	21	A←A∧B	2	1	–	–	–	–	↕	–	–	↕	
	A,#nn	22,nn	A←A∧nn	2	2	–	–	–	–	↕	–	–	↕	
	A,[BR:*ll*]	24,*ll*	A←A∧[BR:*ll*]	3	2	–	–	–	–	↕	–	–	↕	
	A,[hh*ll*]	25,*ll*,hh	A←A∧[hh*ll*]	4	3	–	–	–	–	↕	–	–	↕	
	A,[HL]	23	A←A∧[HL]	2	1	–	–	–	–	↕	–	–	↕	
	A,[IX]	26	A←A∧[IX]	2	1	–	–	–	–	↕	–	–	↕	
	A,[IY]	27	A←A∧[IY]	2	1	–	–	–	–	↕	–	–	↕	
	A,[IX+dd]	CE,20,dd	A←A∧[IX+dd]	4	3	–	–	–	–	↕	–	–	↕	
	A,[IY+dd]	CE,21,dd	A←A∧[IY+dd]	4	3	–	–	–	–	↕	–	–	↕	
	A,[IX+L]	CE,22	A←A∧[IX+L]	4	2	–	–	–	–	↕	–	–	↕	
	A,[IY+L]	CE,23	A←A∧[IY+L]	4	2	–	–	–	–	↕	–	–	↕	
	B,#nn	CE,B0,nn	B←B∧nn	3	3	–	–	–	–	↕	–	–	↕	
	L,#nn	CE,B1,nn	L←L∧nn	3	3	–	–	–	–	↕	–	–	↕	
	H,#nn	CE,B2,nn	H←H∧nn	3	3	–	–	–	–	↕	–	–	↕	
	SC,#nn	9C,nn	SC←SC∧nn	3	2	↓	↓	↓	↓	↓	↓	↓	↓	
	[BR:*ll*],#nn	D8,*ll*,nn	[BR:*ll*]←[BR:*ll*]∧nn	5	3	–	–	–	–	↕	–	–	↕	
	[HL],A	CE,24	[HL]←[HL]∧A	4	2	–	–	–	–	↕	–	–	↕	
	[HL],#nn	CE,25,nn	[HL]←[HL]∧nn	5	3	–	–	–	–	↕	–	–	↕	
	[HL],[IX]	CE,26	[HL]←[HL]∧[IX]	5	2	–	–	–	–	↕	–	–	↕	
	[HL],[IY]	CE,27	[HL]←[HL]∧[IY]	5	2	–	–	–	–	↕	–	–	↕	
OR	A,A	28	A←A∨A	2	1	–	–	–	–	↕	–	–	↕	
	A,B	29	A←A∨B	2	1	–	–	–	–	↕	–	–	↕	
	A,#nn	2A,nn	A←A∨nn	2	2	–	–	–	–	↕	–	–	↕	
	A,[BR:*ll*]	2C,*ll*	A←A∨[BR:*ll*]	3	2	–	–	–	–	↕	–	–	↕	
	A,[hh*ll*]	2D,*ll*,hh	A←A∨[hh*ll*]	4	3	–	–	–	–	↕	–	–	↕	
	A,[HL]	2B	A←A∨[HL]	2	1	–	–	–	–	↕	–	–	↕	
	A,[IX]	2E	A←A∨[IX]	2	1	–	–	–	–	↕	–	–	↕	
	A,[IY]	2F	A←A∨[IY]	2	1	–	–	–	–	↕	–	–	↕	
	A,[IX+dd]	CE,28,dd	A←A∨[IX+dd]	4	3	–	–	–	–	↕	–	–	↕	
	A,[IY+dd]	CE,29,dd	A←A∨[IY+dd]	4	3	–	–	–	–	↕	–	–	↕	
	A,[IX+L]	CE,2A	A←A∨[IX+L]	4	2	–	–	–	–	↕	–	–	↕	

8-bit Arithmetic and Logic Operation Instructions (3/4)

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
OR	A,[IY+L]	CE,2B	A←A∨[IY+L]	4	2	–	–	–	–	↕	–	–	↕	
	B,#nn	CE,B4,nn	B←B∨nn	3	3	–	–	–	–	↕	–	–	↕	
	L,#nn	CE,B5,nn	L←L∨nn	3	3	–	–	–	–	↕	–	–	↕	
	H,#nn	CE,B6,nn	H←H∨nn	3	3	–	–	–	–	↕	–	–	↕	
	SC,#nn	9D,nn	SC←SC∨nn	3	2	↑	↑	↑	↑	↑	↑	↑	↑	
	[BR:*ll*],#nn	D9,*ll*,nn	[BR:*ll*]←[BR:*ll*]∨nn	5	3	–	–	–	–	↕	–	–	↕	
	[HL],A	CE,2C	[HL]←[HL]∨A	4	2	–	–	–	–	↕	–	–	↕	
	[HL],#nn	CE,2D,nn	[HL]←[HL]∨nn	5	3	–	–	–	–	↕	–	–	↕	
	[HL],[IX]	CE,2E	[HL]←[HL]∨[IX]	5	2	–	–	–	–	↕	–	–	↕	
	[HL],[IY]	CE,2F	[HL]←[HL]∨[IY]	5	2	–	–	–	–	↕	–	–	↕	
XOR	A,A	38	A←A∀A	2	1	–	–	–	–	↕	–	–	↕	
	A,B	39	A←A∀B	2	1	–	–	–	–	↕	–	–	↕	
	A,#nn	3A,nn	A←A∀nn	2	2	–	–	–	–	↕	–	–	↕	
	A,[BR:*ll*]	3C,*ll*	A←A∀[BR:*ll*]	3	2	–	–	–	–	↕	–	–	↕	
	A,[hh*ll*]	3D,*ll*,hh	A←A∀[hh*ll*]	4	3	–	–	–	–	↕	–	–	↕	
	A,[HL]	3B	A←A∀[HL]	2	1	–	–	–	–	↕	–	–	↕	
	A,[IX]	3E	A←A∀[IX]	2	1	–	–	–	–	↕	–	–	↕	
	A,[IY]	3F	A←A∀[IY]	2	1	–	–	–	–	↕	–	–	↕	
	A,[IX+dd]	CE,38,dd	A←A∀[IX+dd]	4	3	–	–	–	–	↕	–	–	↕	
	A,[IY+dd]	CE,39,dd	A←A∀[IY+dd]	4	3	–	–	–	–	↕	–	–	↕	
	A,[IX+L]	CE,3A	A←A∀[IX+L]	4	2	–	–	–	–	↕	–	–	↕	
	A,[IY+L]	CE,3B	A←A∀[IY+L]	4	2	–	–	–	–	↕	–	–	↕	
	B,#nn	CE,B8,nn	B←B∀nn	3	3	–	–	–	–	↕	–	–	↕	
	L,#nn	CE,B9,nn	L←L∀nn	3	3	–	–	–	–	↕	–	–	↕	
	H,#nn	CE,BA,nn	H←H∀nn	3	3	–	–	–	–	↕	–	–	↕	
	SC,#nn	9E,nn	SC←SC∀nn	3	2	↕	↕	↕	↕	↕	↕	↕	↕	
	[BR:*ll*],#nn	DA,*ll*,nn	[BR:*ll*]←[BR:*ll*]∀nn	5	3	–	–	–	–	↕	–	–	↕	
	[HL],A	CE,3C	[HL]←[HL]∀A	4	2	–	–	–	–	↕	–	–	↕	
	[HL],#nn	CE,3D,nn	[HL]←[HL]∀nn	5	3	–	–	–	–	↕	–	–	↕	
	[HL],[IX]	CE,3E	[HL]←[HL]∀[IX]	5	2	–	–	–	–	↕	–	–	↕	
	[HL],[IY]	CE,3F	[HL]←[HL]∀[IY]	5	2	–	–	–	–	↕	–	–	↕	
CP	A,A	30	A-A	2	1	–	–	–	–	↕	↕	↕	↕	
	A,B	31	A-B	2	1	–	–	–	–	↕	↕	↕	↕	
	A,#nn	32,nn	A-nn	2	2	–	–	–	–	↕	↕	↕	↕	
	A,[BR:*ll*]	34,*ll*	A-[BR:*ll*]	3	2	–	–	–	–	↕	↕	↕	↕	
	A,[hh*ll*]	35,*ll*,hh	A-[hh*ll*]	4	3	–	–	–	–	↕	↕	↕	↕	
	A,[HL]	33	A-[HL]	2	1	–	–	–	–	↕	↕	↕	↕	
	A,[IX]	36	A-[IX]	2	1	–	–	–	–	↕	↕	↕	↕	
	A,[IY]	37	A-[IY]	2	1	–	–	–	–	↕	↕	↕	↕	
	A,[IX+dd]	CE,30,dd	A-[IX+dd]	4	3	–	–	–	–	↕	↕	↕	↕	
	A,[IY+dd]	CE,31,dd	A-[IY+dd]	4	3	–	–	–	–	↕	↕	↕	↕	
	A,[IX+L]	CE,32	A-[IX+L]	4	2	–	–	–	–	↕	↕	↕	↕	
	A,[IY+L]	CE,33	A-[IY+L]	4	2	–	–	–	–	↕	↕	↕	↕	
	B,#nn	CE,BC,nn	B-nn	3	3	–	–	–	–	↕	↕	↕	↕	
	L,#nn	CE,BD,nn	L-nn	3	3	–	–	–	–	↕	↕	↕	↕	
	H,#nn	CE,BE,nn	H-nn	3	3	–	–	–	–	↕	↕	↕	↕	
	BR,#hh	CE,BF,hh	BR-hh	3	3	–	–	–	–	↕	↕	↕	↕	
	[BR:*ll*],#nn	DB,*ll*,nn	[BR:*ll*]-nn	4	3	–	–	–	–	↕	↕	↕	↕	

8-bit Arithmetic and Logic Operation Instructions (4/4)

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
CP	[HL],A	CE,34	[HL]-A	3	2	–	–	–	–	↕	↕	↕	↕	
	[HL],#nn	CE,35,nn	[HL]-nn	4	3	–	–	–	–	↕	↕	↕	↕	
	[HL],[IX]	CE,36	[HL]-[IX]	4	2	–	–	–	–	↕	↕	↕	↕	
	[HL],[IY]	CE,37	[HL]-[IY]	4	2	–	–	–	–	↕	↕	↕	↕	
BIT	A,B	94	A∧B	2	1	–	–	–	–	↕	–	–	↕	
	A,#nn	96,nn	A∧nn	2	2	–	–	–	–	↕	–	–	↕	
	B,#nn	97,nn	B∧nn	2	2	–	–	–	–	↕	–	–	↕	
	[BR:*ll*],#nn	DC,*ll*,nn	[BR:*ll*]∧nn	4	3	–	–	–	–	↕	–	–	↕	
	[HL],#nn	95,nn	[HL]∧nn	3	2	–	–	–	–	↕	–	–	↕	
INC	A	80	A←A+1	2	1	–	–	–	–	–	–	–	↕	
	B	81	B←B+1	2	1	–	–	–	–	–	–	–	↕	
	L	82	L←L+1	2	1	–	–	–	–	–	–	–	↕	
	H	83	H←H+1	2	1	–	–	–	–	–	–	–	↕	
	BR	84	BR←BR+1	2	1	–	–	–	–	–	–	–	↕	
	[BR:*ll*]	85,*ll*	[BR:*ll*]←[BR:*ll*]+1	4	2	–	–	–	–	–	–	–	↕	
	[HL]	86	[HL]←[HL]+1	3	1	–	–	–	–	–	–	–	↕	
DEC	A	88	A←A-1	2	1	–	–	–	–	–	–	–	↕	
	B	89	B←B-1	2	1	–	–	–	–	–	–	–	↕	
	L	8A	L←L-1	2	1	–	–	–	–	–	–	–	↕	
	H	8B	H←H-1	2	1	–	–	–	–	–	–	–	↕	
	BR	8C	BR←BR-1	2	1	–	–	–	–	–	–	–	↕	
	[BR:*ll*]	8D,*ll*	[BR:*ll*]←[BR:*ll*]-1	4	2	–	–	–	–	–	–	–	↕	
	[HL]	8E	[HL]←[HL]-1	3	1	–	–	–	–	–	–	–	↕	
CPL	A	CE,A0	A←$\overline{\text{A}}$	3	2	–	–	–	–	↕	–	–	↕	
	B	CE,A1	B←$\overline{\text{B}}$	3	2	–	–	–	–	↕	–	–	↕	
	[BR:*ll*]	CE,A2,*ll*	[BR:*ll*]←$\overline{[\text{BR}:ll]}$	5	3	–	–	–	–	↕	–	–	↕	
	[HL]	CE,A3	[HL]←$\overline{[\text{HL}]}$	4	2	–	–	–	–	↕	–	–	↕	
NEG	A	CE,A4	A←0-A	3	2	–	–	★	★	↕	↕	↕	↕	
	B	CE,A5	B←0-B	3	2	–	–	★	★	↕	↕	↕	↕	
	[BR:*ll*]	CE,A6,*ll*	[BR:*ll*]←0-[BR:*ll*]	5	3	–	–	★	★	↕	↕	↕	↕	
	[HL]	CE,A7	[HL]←0-[HL]	4	2	–	–	★	★	↕	↕	↕	↕	
MLT		CE,D8	HL←L*A	12	2	–	–	–	–	↕	0	0	↕	*MODEL1/3*
DIV		CE,D9	L←HL/A, H←Remainder	13	2	–	–	–	–	↕	↕	0	↕	*only*

* Multiplication and division instructions are set only for MODEL1/3. In MODEL0/2, these instructions cannot be used.

16-bit Arithmetic Operation Instructions (1/2)

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
ADD	BA,BA	CF,00	BA←BA+BA	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,HL	CF,01	BA←BA+HL	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,IX	CF,02	BA←BA+IX	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,IY	CF,03	BA←BA+IY	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,#mmnn	C0,nn,mm	BA←BA+mmnn	3	3	–	–	–	–	↕	↕	↕	↕	
	HL,BA	CF,20	HL←HL+BA	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,HL	CF,21	HL←HL+HL	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,IX	CF,22	HL←HL+IX	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,IY	CF,23	HL←HL+IY	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,#mmnn	C1,nn,mm	HL←HL+mmnn	3	3	–	–	–	–	↕	↕	↕	↕	
	IX,BA	CF,40	IX←IX+BA	4	2	–	–	–	–	↕	↕	↕	↕	
	IX,HL	CF,41	IX←IX+HL	4	2	–	–	–	–	↕	↕	↕	↕	
	IX,#mmnn	C2,nn,mm	IX←IX+mmnn	3	3	–	–	–	–	↕	↕	↕	↕	
	IY,BA	CF,42	IY←IY+BA	4	2	–	–	–	–	↕	↕	↕	↕	
	IY,HL	CF,43	IY←IY+HL	4	2	–	–	–	–	↕	↕	↕	↕	
	IY,#mmnn	C3,nn,mm	IY←IY+mmnn	3	3	–	–	–	–	↕	↕	↕	↕	
	SP,BA	CF,44	SP←SP+BA	4	2	–	–	–	–	↕	↕	↕	↕	
	SP,HL	CF,45	SP←SP+HL	4	2	–	–	–	–	↕	↕	↕	↕	
	SP,#mmnn	CF,68,nn,mm	SP←SP+mmnn	4	4	–	–	–	–	↕	↕	↕	↕	
ADC	BA,BA	CF,04	BA←BA+BA+C	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,HL	CF,05	BA←BA+HL+C	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,IX	CF,06	BA←BA+IX+C	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,IY	CF,07	BA←BA+IY+C	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,#mmnn	CF,60,nn,mm	BA←BA+mmnn+C	4	4	–	–	–	–	↕	↕	↕	↕	
	HL,BA	CF,24	HL←HL+BA+C	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,HL	CF,25	HL←HL+HL+C	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,IX	CF,26	HL←HL+IX+C	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,IY	CF,27	HL←HL+IY+C	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,#mmnn	CF,61,nn,mm	HL←HL+mmnn+C	4	4	–	–	–	–	↕	↕	↕	↕	
SUB	BA,BA	CF,08	BA←BA-BA	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,HL	CF,09	BA←BA-HL	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,IX	CF,0A	BA←BA-IX	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,IY	CF,0B	BA←BA-IY	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,#mmnn	D0,nn,mm	BA←BA-mmnn	3	3	–	–	–	–	↕	↕	↕	↕	
	HL,BA	CF,28	HL←HL-BA	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,HL	CF,29	HL←HL-HL	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,IX	CF,2A	HL←HL-IX	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,IY	CF,2B	HL←HL-IY	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,#mmnn	D1,nn,mm	HL←HL-mmnn	3	3	–	–	–	–	↕	↕	↕	↕	
	IX,BA	CF,48	IX←IX-BA	4	2	–	–	–	–	↕	↕	↕	↕	
	IX,HL	CF,49	IX←IX-HL	4	2	–	–	–	–	↕	↕	↕	↕	
	IX,#mmnn	D2,nn,mm	IX←IX-mmnn	3	3	–	–	–	–	↕	↕	↕	↕	
	IY,BA	CF,4A	IY←IY-BA	4	2	–	–	–	–	↕	↕	↕	↕	
	IY,HL	CF,4B	IY←IY-HL	4	2	–	–	–	–	↕	↕	↕	↕	
	IY,#mmnn	D3,nn,mm	IY←IY-mmnn	3	3	–	–	–	–	↕	↕	↕	↕	
	SP,BA	CF,4C	SP←SP-BA	4	2	–	–	–	–	↕	↕	↕	↕	
	SP,HL	CF,4D	SP←SP-HL	4	2	–	–	–	–	↕	↕	↕	↕	
	SP,#mmnn	CF,6A,nn,mm	SP←SP-mmnn	4	4	–	–	–	–	↕	↕	↕	↕	

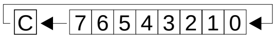
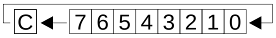
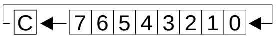
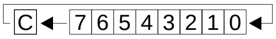
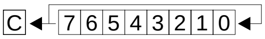
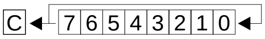
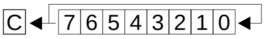
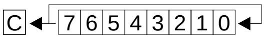
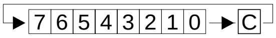
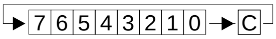
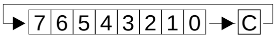
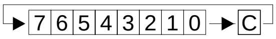
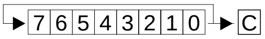
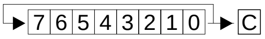
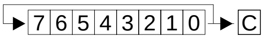
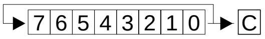
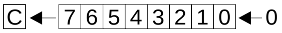
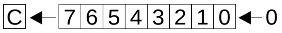
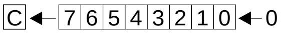
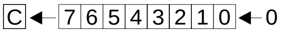
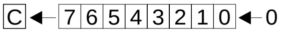
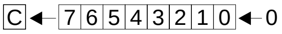
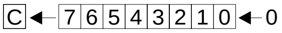
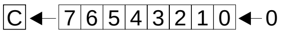
16-bit Arithmetic Operation Instructions (2/2)

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
SBC	BA,BA	CF,0C	BA←BA-BA-C	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,HL	CF,0D	BA←BA-HL-C	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,IX	CF,0E	BA←BA-IX-C	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,IY	CF,0F	BA←BA-IY-C	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,#mmnn	CF,62,nn,mm	BA←BA-mmnn-C	4	4	–	–	–	–	↕	↕	↕	↕	
	HL,BA	CF,2C	HL←HL-BA-C	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,HL	CF,2D	HL←HL-HL-C	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,IX	CF,2E	HL←HL-IX-C	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,IY	CF,2F	HL←HL-IY-C	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,#mmnn	CF,63,nn,mm	HL←HL-mmnn-C	4	4	–	–	–	–	↕	↕	↕	↕	
CP	BA,BA	CF,18	BA-BA	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,HL	CF,19	BA-HL	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,IX	CF,1A	BA-IX	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,IY	CF,1B	BA-IY	4	2	–	–	–	–	↕	↕	↕	↕	
	BA,#mmnn	D4,nn,mm	BA-mmnn	3	3	–	–	–	–	↕	↕	↕	↕	
	HL,BA	CF,38	HL-BA	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,HL	CF,39	HL-HL	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,IX	CF,3A	HL-IX	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,IY	CF,3B	HL-IY	4	2	–	–	–	–	↕	↕	↕	↕	
	HL,#mmnn	D5,nn,mm	HL-mmnn	3	3	–	–	–	–	↕	↕	↕	↕	
	IX,#mmnn	D6,nn,mm	IX-mmnn	3	3	–	–	–	–	↕	↕	↕	↕	
	IY,#mmnn	D7,nn,mm	IY-mmnn	3	3	–	–	–	–	↕	↕	↕	↕	
	SP,BA	CF,5C	SP-BA	4	2	–	–	–	–	↕	↕	↕	↕	
	SP,HL	CF,5D	SP-HL	4	2	–	–	–	–	↕	↕	↕	↕	
	SP,#mmnn	CF,6C,nn,mm	SP-mmnn	4	4	–	–	–	–	↕	↕	↕	↕	
INC	BA	90	BA←BA+1	2	1	–	–	–	–	–	–	–	↕	
	HL	91	HL←HL+1	2	1	–	–	–	–	–	–	–	↕	
	IX	92	IX←IX+1	2	1	–	–	–	–	–	–	–	↕	
	IY	93	IY←IY+1	2	1	–	–	–	–	–	–	–	↕	
	SP	87	SP←SP+1	2	1	–	–	–	–	–	–	–	↕	
DEC	BA	98	BA←BA-1	2	1	–	–	–	–	–	–	–	↕	
	HL	99	HL←HL-1	2	1	–	–	–	–	–	–	–	↕	
	IX	9A	IX←IX-1	2	1	–	–	–	–	–	–	–	↕	
	IY	9B	IY←IY-1	2	1	–	–	–	–	–	–	–	↕	
	SP	8F	SP←SP-1	2	1	–	–	–	–	–	–	–	↕	

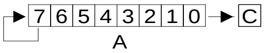
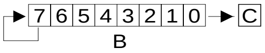
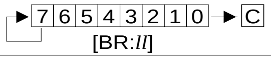
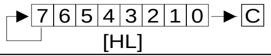
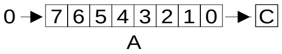
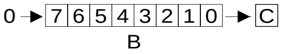
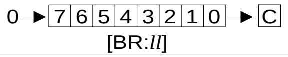
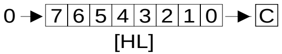
Auxiliary Operation Instructions

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
PACK		DE	B A * m * n → A m n	2	1	–	–	–	–	–	–	–	–	
UPCK		DF	A m n → B A 0 m 0 n	2	1	–	–	–	–	–	–	–	–	
SEP		CE,A8	B A → B A 0******* → 00000000 0******* 1******* → 11111111 1*******	3	2	–	–	–	–	–	–	–	–	

Rotate/Shift Instructions (1/2)

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
RL	A	CE,90	 A	3	2	–	–	–	–	↕	–	↕	↕	
	B	CE,91	 B	3	2	–	–	–	–	↕	–	↕	↕	
	[BR:*ll*]	CE,92,*ll*	 [BR:*ll*]	5	3	–	–	–	–	↕	–	↕	↕	
	[HL]	CE,93	 [HL]	4	2	–	–	–	–	↕	–	↕	↕	
RLC	A	CE,94	 A	3	2	–	–	–	–	↕	–	↕	↕	
	B	CE,95	 B	3	2	–	–	–	–	↕	–	↕	↕	
	[BR:*ll*]	CE,96,*ll*	 [BR:*ll*]	5	3	–	–	–	–	↕	–	↕	↕	
	[HL]	CE,97	 [HL]	4	2	–	–	–	–	↕	–	↕	↕	
RR	A	CE,98	 A	3	2	–	–	–	–	↕	–	↕	↕	
	B	CE,99	 B	3	2	–	–	–	–	↕	–	↕	↕	
	[BR:*ll*]	CE,9A,*ll*	 [BR:*ll*]	5	3	–	–	–	–	↕	–	↕	↕	
	[HL]	CE,9B	 [HL]	4	2	–	–	–	–	↕	–	↕	↕	
RRC	A	CE,9C	 A	3	2	–	–	–	–	↕	–	↕	↕	
	B	CE,9D	 B	3	2	–	–	–	–	↕	–	↕	↕	
	[BR:*ll*]	CE,9E,*ll*	 [BR:*ll*]	5	3	–	–	–	–	↕	–	↕	↕	
	[HL]	CE,9F	 [HL]	4	2	–	–	–	–	↕	–	↕	↕	
SLA	A	CE,80	 A	3	2	–	–	–	–	↕	↕	↕	↕	
	B	CE,81	 B	3	2	–	–	–	–	↕	↕	↕	↕	
	[BR:*ll*]	CE,82,*ll*	 [BR:*ll*]	5	3	–	–	–	–	↕	↕	↕	↕	
	[HL]	CE,83	 [HL]	4	2	–	–	–	–	↕	↕	↕	↕	
SLL	A	CE,84	 A	3	2	–	–	–	–	↕	–	↕	↕	
	B	CE,85	 B	3	2	–	–	–	–	↕	–	↕	↕	
	[BR:*ll*]	CE,86,*ll*	 [BR:*ll*]	5	3	–	–	–	–	↕	–	↕	↕	
	[HL]	CE,87	 [HL]	4	2	–	–	–	–	↕	–	↕	↕	

Rotate/Shift Instructions (2/2)

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
SRA	A	CE,88		3	2	–	–	–	–	↕	0	↕	↕	
	B	CE,89		3	2	–	–	–	–	↕	0	↕	↕	
	[BR:*ll*]	CE,8A,*ll*		5	3	–	–	–	–	↕	0	↕	↕	
	[HL]	CE,8B		4	2	–	–	–	–	↕	0	↕	↕	
SRL	A	CE,8C		3	2	–	–	–	–	0	–	↕	↕	
	B	CE,8D		3	2	–	–	–	–	0	–	↕	↕	
	[BR:*ll*]	CE,8E,*ll*		5	3	–	–	–	–	0	–	↕	↕	
	[HL]	CE,8F		4	2	–	–	–	–	0	–	↕	↕	

Stack Control Instructions

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
PUSH	A	CF,B0	[SP-1]←A, SP←SP-1	3	2	–	–	–	–	–	–	–	–	
	B	CF,B1	[SP-1]←B, SP←SP-1	3	2	–	–	–	–	–	–	–	–	
	L	CF,B2	[SP-1]←L, SP←SP-1	3	2	–	–	–	–	–	–	–	–	
	H	CF,B3	[SP-1]←H, SP←SP-1	3	2	–	–	–	–	–	–	–	–	
	BR	A4	[SP-1]←BR, SP←SP-1	3	1	–	–	–	–	–	–	–	–	
	SC	A7	[SP-1]←SC, SP←SP-1	3	1	–	–	–	–	–	–	–	–	
	BA	A0	[SP-1]←B, [SP-2]←A, SP←SP-2	4	1	–	–	–	–	–	–	–	–	
	HL	A1	[SP-1]←H, [SP-2]←L, SP←SP-2	4	1	–	–	–	–	–	–	–	–	
	IX	A2	[SP-1]←IX(H), [SP-2]←IX(L), SP←SP-2	4	1	–	–	–	–	–	–	–	–	
	IY	A3	[SP-1]←IY(H), [SP-2]←IY(L), SP←SP-2	4	1	–	–	–	–	–	–	–	–	
	EP	A5	[SP-1]←EP, SP←SP-1	3	1	–	–	–	–	–	–	–	–	
	IP	A6	[SP-1]←XP, [SP-2]←YP, SP←SP-2	4	1	–	–	–	–	–	–	–	–	
PUSH	ALL	CF,B8	PUSH BA, HL, IX, IY, BR	12	2	–	–	–	–	–	–	–	–	
	ALE	CF,B9	PUSH BA, HL, IX, IY, BR, EP, IP	15	2	–	–	–	–	–	–	–	–	*MODEL2/3 only*
POP	A	CF,B4	A←[SP], SP←SP+1	3	2	–	–	–	–	–	–	–	–	
	B	CF,B5	B←[SP], SP←SP+1	3	2	–	–	–	–	–	–	–	–	
	L	CF,B6	L←[SP], SP←SP+1	3	2	–	–	–	–	–	–	–	–	
	H	CF,B7	H←[SP], SP←SP+1	3	2	–	–	–	–	–	–	–	–	
	BR	AC	BR←[SP], SP←SP+1	2	1	–	–	–	–	–	–	–	–	
	SC	AF	SC←[SP], SP←SP+1	2	1	↕	↕	↕	↕	↕	↕	↕	↕	
	BA	A8	A←[SP], B←[SP+1], SP←SP+2	3	1	–	–	–	–	–	–	–	–	
	HL	A9	L←[SP], H←[SP+1], SP←SP+2	3	1	–	–	–	–	–	–	–	–	
	IX	AA	IX(L)←[SP], IX(H)←[SP+1], SP←SP+2	3	1	–	–	–	–	–	–	–	–	
	IY	AB	IY(L)←[SP], IY(H)←[SP+1], SP←SP+2	3	1	–	–	–	–	–	–	–	–	
	EP	AD	EP←[SP], SP←SP+1	2	1	–	–	–	–	–	–	–	–	
	IP	AE	YP←[SP], XP←[SP+1], SP←SP+2	3	1	–	–	–	–	–	–	–	–	
POP	ALL	CF,BC	POP BR, IY, IX, HL, BA	11	2	–	–	–	–	–	–	–	–	
	ALE	CF,BD	POP IP, EP, BR, IY, IX, HL, BA	14	2	–	–	–	–	–	–	–	–	*MODEL2/3 only*

* Expand page registers EP/XP/YP are set only for MODEL2/3. In MODEL0/1, instructions that access these registers cannot be used.

Branch Instructions (1/4)

Mnemonic		Machine Code	Condition	Operation	Cycle	Byte	SC							
							I1	I0	U	D	N	V	C	Z
JRS	rr	F1,rr	Unconditionable	*MODEL0/1* PC←PC+rr+1 *MODEL2/3* PC←PC+rr+1, CB←NB	2	2	–	–	–	–	–	–	–	–
JRS	C,rr	E4,rr	C=1	*MODEL0/1* If Condition is true, then PC←PC+rr+1 else PC←PC+2 *MODEL2/3* If Condition is true, then PC←PC+rr+1, CB←NB else PC←PC+2, NB←CB	2	2	–	–	–	–	–	–	–	–
	NC,rr	E5,rr	C=0											
	Z,rr	E6,rr	Z=1											
	NZ,rr	E7,rr	Z=0											
JRS	LT,rr	CE,E0,rr	[N∀V]=1	*MODEL0/1* If Condition is true, then PC←PC+rr+2 else PC←PC+3 *MODEL2/3* If Condition is true, then PC←PC+rr+2, CB←NB else PC←PC+3, NB←CB	3	3	–	–	–	–	–	–	–	–
	LE,rr	CE,E1,rr	Z∨[N∀V]=1											
	GT,rr	CE,E2,rr	Z∨[N∀V]=0											
	GE,rr	CE,E3,rr	[N∀V]=0											
	V,rr	CE,E4,rr	V=1											
	NV,rr	CE,E5,rr	V=0											
	P,rr	CE,E6,rr	N=0											
	M,rr	CE,E7,rr	N=1											
	F0,rr	CE,E8,rr	F0=1											
	F1,rr	CE,E9,rr	F1=1											
	F2,rr	CE,EA,rr	F2=1											
	F3,rr	CE,EB,rr	F3=1											
	NF0,rr	CE,EC,rr	F0=0											
	NF1,rr	CE,ED,rr	F1=0											
	NF2,rr	CE,EE,rr	F2=0											
	NF3,rr	CE,EF,rr	F3=0											
JRL	qqrr	F3,rr,qq	Unconditionable	*MODEL0/1* PC←PC+qqrr+2 *MODEL2/3* PC←PC+qqrr+2, CB←NB	3	3	–	–	–	–	–	–	–	–
JRL	C,qqrr	EC,rr,qq	C=1	*MODEL0/1* If Condition is true, then PC←PC+qqrr+2 else PC←PC+3 *MODEL2/3* If Condition is true, then PC←PC+qqrr+2, CB←NB else PC←PC+3, NB←CB	3	3	–	–	–	–	–	–	–	–
	NC,qqrr	ED,rr,qq	C=0											
	Z,qqrr	EE,rr,qq	Z=1											
	NZ,qqrr	EF,rr,qq	Z=0											
DJR	NZ,rr	F5,rr	B=0	*MODEL0/1* B←B-1, If B=0, then PC←PC+rr+1 else PC←PC+2 *MODEL2/3* B←B-1, If B=0, then PC←PC+rr+1, CB←NB else PC←PC+2, NB←CB	4	2	–	–	–	–	–	–	–	↕

Branch Instructions (2/4)

Mnemonic		Machine Code	Condition	Operation	Cycle	Byte	SC							
							I1	I0	U	D	N	V	C	Z
JP	HL	F4	Unconditionable	*MODEL0/1* PC←HL	2	1	–	–	–	–	–	–	–	–
				MODEL2/3 PC←HL, CB←NB										
	[kk]	FD,kk	Unconditionable	*MODEL0/1* PC(L)←[00kk], PC(H)←[00kk+1]	4	2	–	–	–	–	–	–	–	–
				MODEL2/3 PC(L)←[00kk] PC(H)←[00kk+1], CB←NB										
CARS	rr	F0,rr	Unconditionable	*MODEL0/1* [SP-1]←PC(H), [SP-2]←PC(L), SP←SP-2, PC←PC+rr+1	4	2	–	–	–	–	–	–	–	–
				MODEL2/3 (Minimum mode) [SP-1]←PC(H), [SP-2]←PC(L), SP←SP-2, PC←PC+rr+1, CB←NB										
				MODEL2/3 (Maximum mode) [SP-1]←CB, [SP-2]←PC(H), [SP-3]←PC(L), SP←SP-3, PC←PC+rr+1, CB←NB	5									
CARS	C,rr	E0,rr	C=1	*MODEL0/1* If Condition is true then [SP-1]←PC(H), [SP-2]←PC(L), SP←SP-2, PC←PC+rr+1	4	2	–	–	–	–	–	–	–	–
	NC,rr	E1,rr	C=0	else PC←PC+2	2									
	Z,rr	E2,rr	Z=1	*MODEL2/3 (Minimum mode)* If Condition is true then [SP-1]←PC(H), [SP-2]←PC(L), SP←SP-2, PC←PC+rr+1, CB←NB	4									
				else PC←PC+2, NB←CB	2									
	NZ,rr	E3,rr	Z=0	*MODEL2/3 (Maximum mode)* If Condition is true then [SP-1]←CB, [SP-2]←PC(H), [SP-3]←PC(L), SP←SP-3, PC←PC+rr+1, CB←NB	5									
				else PC←PC+2, NB←CB	2									
CARS	LT,rr	CE,F0,rr	[N∀V]=1	*MODEL0/1* If Condition is true then [SP-1]←PC(H), [SP-2]←PC(L), SP←SP-2, PC←PC+rr+2	5	3	–	–	–	–	–	–	–	–
	LE,rr	CE,F1,rr	Z∨[N∀V]=1											
	GT,rr	CE,F2,rr	Z∨[N∀V]=0											
	GE,rr	CE,F3,rr	[N∀V]=0											
	V,rr	CE,F4,rr	V=1	else PC←PC+3	3									
	NV,rr	CE,F5,rr	V=0	*MODEL2/3 (Minimum mode)* If Condition is true then [SP-1]←PC(H), [SP-2]←PC(L), SP←SP-2, PC←PC+rr+2, CB←NB	5									
	P,rr	CE,F6,rr	N=0											
	M,rr	CE,F7,rr	N=1											
	F0,rr	CE,F8,rr	F0=1											
	F1,rr	CE,F9,rr	F1=1	else PC←PC+3, NB←CB	3									
	F2,rr	CE,FA,rr	F2=1	*MODEL2/3 (Maximum mode)* If Condition is true then [SP-1]←CB, [SP-2]←PC(H), [SP-3]←PC(L), SP←SP-3, PC←PC+rr+2, CB←NB	6									
	F3,rr	CE,FB,rr	F3=1											
	NF0,rr	CE,FC,rr	F0=0											
	NF1,rr	CE,FD,rr	F1=0											
	NF2,rr	CE,FE,rr	F2=0											
	NF3,rr	CE,FF,rr	F3=0	else PC←PC+3, NB←CB	3									

Branch Instructions (3/4)

Mnemonic		Machine Code	Condition	Operation	Cycle	Byte	SC							
							I1	I0	U	D	N	V	C	Z
CARL	qqrr	F2,rr,qq	Unconditionable	*MODEL0/1* [SP-1]←PC(H), [SP-2]←PC(L), SP←SP-2, PC←PC+qqrr+2	5	3	–	–	–	–	–	–	–	–
				MODEL2/3 (Minimum mode) [SP-1]←PC(H), [SP-2]←PC(L), SP←SP-2, PC←PC+qqrr+2, CB←NB										
				MODEL2/3 (Maximum mode) [SP-1]←CB, [SP-2]←PC(H), [SP-3]←PC(L), SP←SP-3, PC←PC+qqrr+2, CB←NB	6									
CARL	C,qqrr	E8,rr,qq	C=1	*MODEL0/1* If Condition is true then [SP-1]←PC(H), [SP-2]←PC(L), SP←SP-2, PC←PC+qqrr+2	5	3	–	–	–	–	–	–	–	–
	NC,qqrr	E9,rr,qq	C=0	else PC←PC+3	3									
	Z,qqrr	EA,rr,qq	Z=1	*MODEL2/3 (Minimum mode)* If Condition is true then [SP-1]←PC(H), [SP-2]←PC(L), SP←SP-2, PC←PC+qqrr+2, CB←NB	5									
				else PC←PC+3, NB←CB	3									
	NZ,qqrr	EB,rr,qq	Z=0	*MODEL2/3 (Maximum mode)* If Condition is true then [SP-1]←CB, [SP-2]←PC(H), [SP-3]←PC(L), SP←SP-3, PC←PC+qqrr+2, CB←NB	6									
				else PC←PC+3, NB←CB	3									
CALL	[hh*ll*]	FB,*ll*,hh	Unconditionable	*MODEL0/1* [SP-1]←PC(H), [SP-2]←PC(L), SP←SP-2, PC(L)←[hh*ll*], PC(H)←[hh*ll*+1]	7	3	–	–	–	–	–	–	–	–
				MODEL2/3 (Minimum mode) [SP-1]←PC(H), [SP-2]←PC(L), SP←SP-2, PC(L)←[hh*ll*], PC(H)←[hh*ll*+1], CB←NB										
				MODEL2/3 (Maximum mode) [SP-1]←CB, [SP-2]←PC(H), [SP-3]←PC(L), SP←SP-3, PC(L)←[hh*ll*], PC(H)←[hh*ll*+1], CB←NB	8									

Branch Instructions (4/4)

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
INT	[kk]	FC,kk	*MODEL0/1* [SP-1]←PC(H), [SP-2]←PC(L), [SP-3]←SC, SP←SP-3, PC(L)←[00kk], PC(H)←[00kk+1]	7	2	–	–	–	–	–	–	–	–	
			MODEL2/3 (Minimum mode) [SP-1]←PC(H), [SP-2]←PC(L), [SP-3]←SC, SP←SP-3, PC(L)←[00kk], PC(H)←[00kk+1], CB←NB											
			MODEL2/3 (Maximum mode) [SP-1]←CB, [SP-2]←PC(H), [SP-3]←PC(L), [SP-4]←SC, SP←SP-4, PC(L)←[00kk], PC(H)←[00kk+1], CB←NB	8										
RET		F8	*MODEL0/1, MODEL2/3 (Minimum mode)* PC(L)←[SP], PC(H)←[SP+1], SP←SP+2	3	1	–	–	–	–	–	–	–	–	
			MODEL2/3 (Maximum mode) PC(L)←[SP], PC(H)←[SP+1], CB←[SP+2], NB←CB, SP←SP+3	4										
RETE		F9	*MODEL0/1, MODEL2/3 (Minimum mode)* SC←[SP], PC(L)←[SP+1], PC(H)←[SP+2], SP←SP+3	4	1	↕	↕	↕	↕	↕	↕	↕	↕	
			MODEL2/3 (Maximum mode) SC←[SP], PC(L)←[SP+1], PC(H)←[SP+2], CB←[SP+3], NB←CB, SP←SP+4	5										
RETS		FA	*MODEL0/1, MODEL2/3 (Minimum mode)* PC(L)←[SP], PC(H)←[SP+1], SP←SP+2, PC←PC+2	5	1	–	–	–	–	–	–	–	–	
			MODEL2/3 (Maximum mode) PC(L)←[SP], PC(H)←[SP+1], CB←[SP+2], NB←CB, SP←SP+3, PC←PC+2	6										

System Control Instructions

Mnemonic		Machine Code	Operation	Cycle	Byte	SC								Comment
						I1	I0	U	D	N	V	C	Z	
NOP		FF	No Operation	2	1	–	–	–	–	–	–	–	–	
HALT		CE,AE	HALT	3	2	–	–	–	–	–	–	–	–	
SLP		CE,AF	SLEEP	3	2	–	–	–	–	–	–	–	–	

Appendix C Programming Notes

System Controller and Bus Control

(1) All the interrupts including $\overline{\text{NMI}}$ are masked, until you write the optional value into both the "00FF00H" and "00FF01H" addresses. Consequently, even if you do not change the content of this address (You use the initial value, as is.), you should still be sure to perform the writing operation using the initialization routine.

(2) When setting stack fields, including page addresses as well, you should write them in the order of the register SPP ("00FF01H") and the stack pointer SP.

Example: When setting the "178000H" address

```
LD   EP, #00H
LD   HL, #0FF01H
LD   [HL], #17H
LD   SP, #8000H
```

During this period the interrupts (including $\overline{\text{NMI}}$) are masked.

Watchdog Timer

The watchdog timer must reset within 3-second cycles by software.

Oscillation Circuit and Operating Mode

(1) When the high speed CPU operation is not necessary, you should operate the peripheral circuits according to the setting outline indicate below.

- *CPU operating clock*
  OSC1
- *OSC3 oscillation circuit*
  OFF (When the OSC3 clock is not necessary for some peripheral circuits.)
- *Operating mode*
  Low power mode
  (When V_{DD}–V_{SS} is 3.5 V or less)
  or Normal mode
  (When V_{DD}–V_{SS} is 3.5 V or more)

(2) Do not turn the OSC3 oscillation circuit ON in the low power mode.
Do not switch over the operating mode (normal mode ↔ high speed mode) in the OSC3 oscillation circuit ON status, as this will cause faulty operation.

(3) When turning ON the OSC3 oscillation circuit after switching the operating mode, you should allow a minimum waiting time of 5 msec.

(4) Since several msec to several tens of msec are necessary for the oscillation to stabilize after turning the OSC3 oscillation circuit ON. Consequently, you should switch the CPU operating clock (OSC1 → OSC3) after allowing for a sufficient waiting time once the OSC3 oscillation goes ON. (The oscillation start time will vary somewhat depending on the oscillator and on the externally attached parts. Refer to the oscillation start time example indicated in Chapter 7, "ELECTRICAL CHARACTERISTICS".)

(5) When switching the clock from OSC3 to OSC1, be sure to switch OSC3 oscillation OFF with separate instructions. Using a single instruction to process simultaneously can cause a malfunction of the CPU.

Input Ports (K Ports)

When changing the input terminal from LOW level to HIGH with the built-in pull-up resistor, a delay in the waveform rise time will occur depending on the time constant of the pull-up resistor and the load capacitance of the terminal. It is necessary to set an appropriate wait time for introduction of an input port. In particular, special attention should be paid to key scan for key matrix formation. Make this wait time the amount of time or more calculated by the following expression.

Wait time = R_{IN} x (C_{IN} + load capacitance on the board) x 1.6 [sec]
R_{IN}: Pull up resistance Max. value
C_{IN}: Terminal capacitance Max. value

Output Ports (R ports)

(1) Since the special output signals (TOUT, FOUT and BZ) are generated asynchronously from the output control registers (PTOUT, FOUTON, BZON, BZSHT and BZSTP), when the signals is turned ON or OFF by the output control register settings, a hazard of a 1/2 cycle or less is generated.

(2) When the FOUT frequency is made "f_{OSC3}/n", you must turn on the OSC3 oscillation circuit before outputting FOUT. A time interval of several msec to several 10 msec, from the turning ON of the OSC3 oscillation circuit to until the oscillation stabilizes, is necessary, due to the oscillation element that is used. Consequently, if an abnormality occurs as the result of an unstable FOUT signal being output externally, you should allow an adequate waiting time after turning ON of the OSC3 oscillation, before turning outputting FOUT. (The oscillation start time will vary somewhat depending on the oscillator and on the externally attached parts. Refer to the oscillation start time example indicated in Chapter 7, "ELECTRICAL CHARACTERISTICS".)
At initial reset, OSC3 oscillation circuit is set to OFF state.

(3) The SLP instruction has executed when the special output signals (TOUT, FOUT and BZ) are in the enable status, an unstable clock is output for the special output at the time of return from the SLEEP state. Consequently, when shifting to the SLEEP state, you should set the special output signal to the disable status prior to executing the SLP instruction.

I/O Ports (P Ports)

(1) When changing the port terminal from LOW level to HIGH with the built-in pull-up resistor, a delay in the waveform rise time will occur depending on the time constant of the pull-up resistor and the load capacitance of the terminal. It is necessary to set an appropriate wait time for introduction of an I/O port. Make this wait time the amount of time or more calculated by the following expression.

Wait time = R_{IN} x (C_{IN} + load capacitance on the board) x 1.6 [sec]
R_{IN}: Pull up resistance Max. value
C_{IN}: Terminal capacitance Max. value

(2) When the analog comparator is used, "0" must always be set for the I/O control registers (IOC14–IOC15 or IOC16–IOC17, or both) of I/O ports which will become input terminals.

Serial Interface

(1) Be sure to initialize the serial interface mode in the transmitting/receiving disable status (TXEN = RXEN = "0").

(2) Do not perform double trigger (writing "1") to TXTRG (RXTRG) when the serial interface is in the transmitting (receiving) operation. Furthermore, do not execute the SLP instruction. (When executing the SLP instruction, set TXEN = RXEN = "0".)

(3) In the clock synchronous mode, since one clock line ($\overline{\text{SCLK}}$) is shared for both transmitting and receiving, transmitting and receiving cannot be performed simultaneously. (Half duplex only is possible in clock synchronous mode.)
Consequently, be sure not to write "1" to RXTRG (TXTRG) when TXTRG (RXTRG) is "1".

(4) When a parity error or flaming error is generated during receiving in the asynchronous mode, the receiving error interrupt factor flag FSERR is set to "1" prior to the receiving complete interrupt factor flag FSREC for the time indicated in Table C.1. Consequently, when an error is generated, you should reset the receiving complete interrupt factor flag FSREC to "0" by providing a wait time in error processing routines and similar routines.
When an overrun error is generated, the receiving complete interrupt factor flag FSREC is not set to "1" and a receiving complete interrupt is not generated.

Table C.1 Time difference between FSERR and FSREC on error generation

Clock source	Time difference
f_{OSC3} / n	1/2 cycles of f_{OSC3} / n
Programmable timer	1 cycle of timer 1 underflow

(5) When the demultiplied signal of the OSC3 oscillation circuit is made the clock source, it is necessary to turn the OSC3 oscillation ON, prior to using the serial interface.
A time interval of several msec to several 10 msec, from the turning ON of the OSC3 oscillation circuit to until the oscillation stabilizes, is necessary, due to the oscillation element that is used. Consequently, you should allow an adequate waiting time after turning ON of the OSC3 oscillation, before starting transmitting/receiving of serial interface. (The oscillation start time will vary somewhat depending on the oscillator and on the externally attached parts. Refer to the oscillation start time example indicated in Chapter 7, "ELECTRICAL CHARACTERISTICS".)
At initial reset, the OSC3 oscillation circuit is set to OFF status.

Clock Timer

(1) The clock timer is actually made to RUN/STOP in synchronization with the falling edge of the 256 Hz signal after writing to the TMRUN register. Consequently, when "0" is written to the TMRUN, the timer shifts to STOP status when the counter is incremented "1". The TMRUN maintains "1" for reading until the timer actually shifts to STOP status.
Figure C.1 shows the timing chart of the RUN/STOP control.

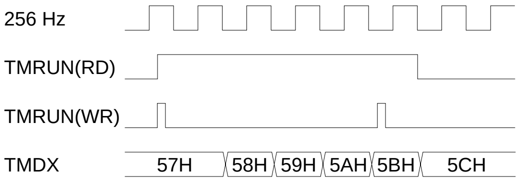


Fig. C.1 Timing chart of RUN/STOP control

(2) The SLP instruction is executed when the clock timer is in the RUN status (TMRUN = "1"). The clock timer operation will become unstable when returning from SLEEP status. Therefore, when shifting to SLEEP status, set the clock timer to STOP status (TMRUN = "0") prior to executing the SLP instruction.

Stopwatch Timer

(1) The stopwatch timer is actually made to RUN/STOP in synchronization with the falling edge of the 256 Hz signal after writing to the SWRUN register. Consequently, when "0" is written to the SWRUN, the timer shifts to STOP status when the counter is incremented "1". The SWRUN maintains "1" for reading until the timer actually shifts to STOP status.
Figure C.2 shows the timing chart of the RUN/STOP control.

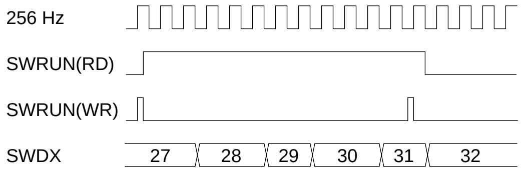


Fig. C.2 Timing chart of RUN/STOP control

(2) The SLP instruction is executed when the stopwatch timer is in the RUN status (SWRUN = "1"). The stopwatch timer operation will become unstable when returning from SLEEP status. Therefore, when shifting to SLEEP status, set the clock timer to STOP status (SWRUN = "0") prior to executing the SLP instruction.

Programmable Timer

(1) The programmable timer is actually made to RUN/STOP in synchronization with the falling edge of the input clock after writing to the PRUN0(1) register. Consequently, when "0" is written to the PRUN0(1), the timer shifts to STOP status when the counter is decremented "1". The PRUN0(1) maintains "1" for reading until the timer actually shifts to STOP status.
Figure C.3 shows the timing chart of the RUN/STOP control.

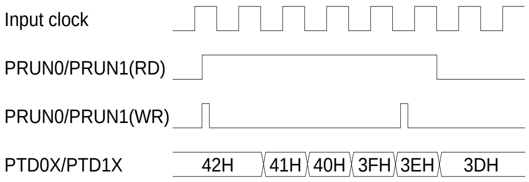


Fig. C.3 Timing chart of RUN/STOP control

The event counter mode is excluded from the above note.

(2) The SLP instruction is executed when the programmable timer is in the RUN status (PRUN0(1) = "1"). The programmable timer operation will become unstable when returning from SLEEP status. Therefore, when shifting to SLEEP status, set the clock timer to STOP status (PRUN0(1) = "0") prior to executing the SLP instruction.
In the same way, disable the TOUT signal (PTOUT = "0") to avoid an unstable clock output to the R27 output port terminal.

(3) Since the TOUT signal is generated asynchronously from the register PTOUT, when the signal is turned ON or OFF by the register setting, a hazard of a 1/2 cycle or less is generated.

(4) When the OSC3 oscillation circuit is made the clock source, it is necessary to turn the OSC3 oscillation ON, prior to using the programmable timer.
From the time the OSC3 oscillation circuit is turning ON until oscillation stabilizes, an interval of several msec to several 10 msec is necessary. Consequently, you should allow an adequate waiting time after turning the OSC3 oscillation circuit ON before starting the count of the programmable timer. (The oscillation start time will vary somewhat depending on the oscillator and on external parts. Refer to the oscillation start time example indicated in Chapter 7, "ELECTRICAL CHARACTERISTICS".)
At initial reset, OSC3 oscillation circuit is set to OFF status.

(5) When the 16-bit mode has been selected, be sure to read the counter data in the order of PTD00–PTD07 and PTD10–PTD17. Moreover, the time interval between reading PTD00–PTD07 and PTD10–PTD17 should be 0.73 msec or less.

Sound Generator

(1) Since the BZ signal is generated asynchronously from the register BZON, when the signal is turned ON or OFF by the register setting, a hazard of a 1/2 cycle or less is generated.

(2) The SLP instruction has executed when the BZ signal is in the enable status (BZON = "1" or BZSHT = "1"), an unstable clock is output from the R50 output port terminal at the time of return from the SLEEP status. Consequently, when shifting to the SLEEP status, you should set the BZ signal to the disable status (BZON = BZSHT = "0") prior to executing the SLP instruction.

(3) The one-shot output is only valid when the normal buzzer output is OFF (BZON = "0") status. The trigger is invalid during ON (BZON = "1") status.

Analog Comparator

(1) To reduce current consumption, turn the analog comparator OFF (CMP0ON = CMP1ON = "0") when it is not necessary.

(2) After the analog comparator has been turned ON, a maximum time of 3 msec is necessary until output stabilizes. Consequently, you should allow an adequate waiting time after turning the analog comparator ON, before reading the comparison result.

(3) Since the input terminals of the analog comparator are common to the I/O ports, the I/O control registers (IOC14–IOC17) corresponding to the channel to be used must be set to the input mode.

SVD (Supply Voltage Detection) Circuit

(1) To reduce current consumption, turn the SVD circuit OFF (SVDON = SVDSP = "0") when it is not necessary.

(2) When executing an SLP instruction while the SVD circuit is operating, the stop operation of the OSC1 oscillation circuit is kept waiting until the sampling is completed. The two bits of SVDON and SVDSP are automatically reset to "0" by hardware while waiting for completion of sampling.

Interrupt (Exception) Processing

(1) When executing the RETE instruction without resetting the interrupt factor flag after an interrupt has been generated, the same interrupt will be generated. Consequently, the interrupt factor flag corresponding to that routine must be reset (writing "1") in the interrupt processing routine.

(2) Beware. If the interrupt flags (I0 and I1) have been rewritten (set to lower priority) prior to resetting an interrupt factor flag after an interrupt has been generated, the same interrupt will be generated again.

(3) An exception processing vector is fixed at 2 bytes, so it cannot specify a branch destination bank address. Consequently, to branch from multiple banks to a common exception processing routine, the front portion of an exception processing routine must be described within the common area (000000H–007FFFH).

EPSON International Sales Operations

AMERICA

S-MOS SYSTEMS, INC.
150 River Oaks Parkway
San Jose, CA 95134, U.S.A.
Phone: +1-408-922-0200 Fax: +1-408-922-0238
Telex: 176079 SMOS SNJUD

S-MOS SYSTEMS, INC.
EASTERN AREA SALES AND TECHNOLOGY CENTER
301 Edgewater Place, Suite 120
Wakefield, MA 01880, U.S.A.
Phone: +1-617-246-3600 Fax: +1-617-246-5443

S-MOS SYSTEMS, INC.
SOUTH EASTERN AREA SALES AND TECHNOLOGY CENTER
4300 Six Forks Road, Suite 430
Raleigh, NC 27609, U.S.A.
Phone: +1-919-781-7667 Fax: +1-919-781-6778

S-MOS SYSTEMS, INC.
CENTRAL AREA SALES AND TECHNOLOGY CENTER
1450 E.American Lane, Suite 1550
Schaumburg, IL 60173, U.S.A.
Phone: +1-847-517-7667 Fax: +1-847-517-7601

EUROPE

- HEADQUARTERS -

EPSON EUROPE ELECTRONICS GmbH
Riesstrasse 15
80992 Muenchen, GERMANY
Phone: +49-(0)89-14005-0 Fax: +49-(0)89-14005-110

- GERMANY -

EPSON EUROPE ELECTRONICS GmbH
SALES OFFICE
Breidenbachstrasse 46
D-51373 Leverkusen, GERMANY
Phone: +49-(0)214-83070-0 Fax: +49-(0)214-83070-10

- UNITED KINGDOM -

EPSON EUROPE ELECTRONICS GmbH
UK BRANCH OFFICE
G6 Doncastle House, Doncastle Road
Bracknell, Berkshire RG12 8PE, ENGLAND
Phone: +44-(0)1344-381700 Fax: +44-(0)1344-381701

- FRANCE -

EPSON EUROPE ELECTRONICS GmbH
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

- 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

- CHINA -

SHANGHAI EPSON ELECTRONICS CO., LTD.
4F, Bldg., 27, No. 69, Gui Jing Road
Caohejing, Shanghai, CHINA
Phone: 21-6485-5552 Fax: 21-6485-0775

- 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

EPSON TAIWAN TECHNOLOGY & TRADING LTD.
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
10F, 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

■ Electronic devices information on the Epson WWW server

http://www.epson.co.jp

First issue OCTOBER 1994, Printed OCTOBER 1998 in Japan Ⓜ B