PEUL63611-02

OKI

ML63611

User's Manual

CMOS 4-bit microcontroller

Preliminary

SECOND EDITION

ISSUE DATE: Jun. 2001

NOTICE

1. The information contained herein can change without notice owing to product and/or technical improvements. Before using the product, please make sure that the information being referred to is up-to-date.

2. The outline of action and examples for application circuits described herein have been chosen as an explanation for the standard action and performance of the product. When planning to use the product, please ensure that the external conditions are reflected in the actual circuit, assembly, and program designs.

3. When designing your product, please use our product below the specified maximum ratings and within the specified operating ranges including, but not limited to, operating voltage, power dissipation, and operating temperature.

4. Oki assumes no responsibility or liability whatsoever for any failure or unusual or unexpected operation resulting from misuse, neglect, improper installation, repair, alteration or accident, improper handling, or unusual physical or electrical stress including, but not limited to, exposure to parameters beyond the specified maximum ratings or operation outside the specified operating range.

5. Neither indemnity against nor license of a third party's industrial and intellectual property right, etc. is granted by us in connection with the use of the product and/or the information and drawings contained herein. No responsibility is assumed by us for any infringement of a third party's right which may result from the use thereof.

6. The products listed in this document are intended for use in general electronics equipment for commercial applications (e.g., office automation, communication equipment, measurement equipment, consumer electronics, etc.). These products are not authorized for use in any system or application that requires special or enhanced quality and reliability characteristics nor in any system or application where the failure of such system or application may result in the loss or damage of property, or death or injury to humans.

   Such applications include, but are not limited to, traffic and automotive equipment, safety devices, aerospace equipment, nuclear power control, medical equipment, and life-support systems.

7. Certain products in this document may need government approval before they can be exported to particular countries. The purchaser assumes the responsibility of determining the legality of export of these products and will take appropriate and necessary steps at their own expense for these.

8. No part of the contents contained herein may be reprinted or reproduced without our prior permission.

Preface

This manual describes the hardware of Oki's original CMOS 4-bit microcontroller ML63611.

Refer to the "nX-4/250, 300 Core Instruction Manual" for details of the 4-bit CPU core nX-4/250 which is built in the ML63611.

The manuals related to the ML63611 is shown below.

- nX-4/250, 300 Core Instruction Manual:
  Describes the base architecture and instruction set of nX-4/250 core and nX-4/300 core.

- SASM63K User's Manual:
  Describes the structured assembler operation and assembler language specification.

- Dr.63611 User's Manual:
  Describes the hardware of the emulator.

- DT63K Debugger User's Manual:
  Describes the debugger commands and the hardware of the simulator.

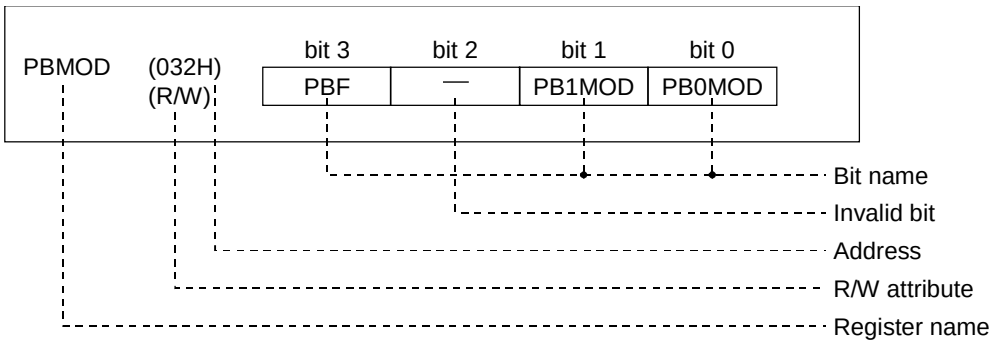
- MOGTOOL Mask Option Generator User's Manual:
  Describes the mask option settings and the generator operation.

Notation

Classification	Notation	Description
■ Numeric value	xxh, xxH	Represents a hexadecimal number.
	xxb	Represents a binary number.
■ Unit	word, W	1 word = 16 bits
	byte, B	1 byte = 2 nibbles = 8 bits
	nibble, N	1 nibble = 4 bits
	mega-, M	10^6
	kilo-, K	$2^{10} = 1024$
	kilo-, k	$10^3 = 1000$
	milli-, m	10^{-3}
	micro-, µ	10^{-6}
	nano-, n	10^{-9}
	second, s (lower case)	second
	KB	1 KB = 1 kilobyte = 1024 bytes
	MB	1 MB = 1 megabyte = 2^{20} bytes = 1,048,576 bytes
■ Symbol	⚠ Note:	Gives more information about mistakable items.
■ Terminology	"H" level	Indicates high side voltage signal levels V_{IH} and V_{OH} as specified by the electrical characteristics.
	"L" level	Indicates low side voltage signal levels V_{IL} and V_{OL} as specified by the electrical characteristics.

■ Register description

Invalid bit : When read, a value of "1" is always obtained. Write operations are invalid.
R/W attribute : "R" indicates data can be read and "W" indicates data can be written.



Contents

Chapter 1

Overview

1. Overview

1.1 Overview

The ML63611 is a CMOS 4-bit microcontroller using Oki's original CPU core nX-4/250.

The ML63611 is provided with the mask options of eight items of selection including (1.5 V or 3.0 V) power supply specifications and (With or Without) the regulator circuit for the LCD bias reference voltage.

When a 3.0 V power supply specificaiton is selected, the halver circuit can be used to decrease power consumption. The halver circuit cannot be used when a 1.5 V power supply specification is selected.

When "With the regulator circuit for the LCD bias reference voltage" is selected, the LCD bias reference voltage will be generated based on the output voltage of the regulator circuit. When "Without the regulator circuit for the LCD bias reference voltage" is selected, the LCD bias reference voltage will be generated based on the power supply voltage; for this reason, the LCD bias voltage decreases as the power supply voltage decreases, causing the display density of the LCD panel to thin down.

The ML63611 has incorporated in it an 8K-word program memory, a 1K-nibble data memory, four input ports, four output ports (only when the mask option of LCD driver pins is selected), 16 I/O ports, a melody circuit, a serial port, four 8-bit timers, and a 64-segment LCD driver (60 segment lines and 4 common lines, max.). (A part of the SEG pins can also be selected as output port pins or COM pins depending on the mask option.)

 Note:

In this manual, for convenience of description, the symbols OPTION A, OPTION B, OPTION C, and OPTION D are used in accordance with the mask option selection of a power supply specification (1.5 V or 3.0 V) and the regulator circuit for the LCD bias reference voltage (With or Without), as shown below.

- OPTION A: 1.5 V power supply specification (halver circuit disabled), without the regulator circuit for the LCD bias reference voltage
- OPTION B: 1.5 V power supply specification (halver circuit disabled), with the regulator circuit for the LCD bias reference voltage
- OPTION C: 3.0 V power supply specification (halver circuit enabled), without the regulator circuit for the LCD bias reference voltage
- OPTION D: 3.0 V power supply specification (halver circuit enabled), with the regulator circuit for the LCD bias reference voltage

1.2 Features

The ML63611 has the following features.

a. Extensive instruction set
- 407 instructions
  Transfer, rotate, increment/decrement, arithmetic operations, compare, logic operations, mask operations, bit operations, ROM table reference, stack operations, flag operations, jump, conditional branch, call/return, control

b. Wide variety of addressing modes
- Indirect addressing mode for 4 types of data memory with current bank register, extra bank register, HL register and XY register
- Data memory bank internal direct addressing mode

c. Processing speed

- 2 clocks per machine cycle, with most instructions executed in 1 machine cycle
- Minimum instruction execution time: 61 µs (@ 32.768 kHz system clock)
  10 µs (@ 200 kHz system clock)
  2.86 µs (@ 700 kHz system clock)

d. Clock generator circuit

- Low-speed clock:
  Crystal oscillation (32.768 kHz)
- High-speed clock:
  OPTION A, OPTION B: RC oscillation (200 kHz max.)
  OPTION C, OPTION D: Ceramic oscillation or RC oscillation selected with software (700 kHz max.)

e. Program memory space

- 8K words
- The basic instruction length is 16 bits per word.

f. Data memory space

- 1024 nibbles

g. Stack level

- Call stack level: 16
- Register stack level: 16

h. Ports

Input port (Port 0.0 to Port 0.3):
Selectable as input with pull-up resistor/high-impedance input
Provided with the reset function that resets the system when there is a simultaneous key depression of multiple bits (2, 3, or 4 bits).

Output port: Selectable as N-channel open drain output/CMOS output
Enabled only when the SEG pins (L32 to L35) are selected as the output port by the mask option.

Input-output port (Port A.0 to Port A.3, Port B.0 to Port B.3, Port C.0 to Port C.3, Port E.0 to Port E.3):
Selectable as input with pull-up resistor/input with pull-down resistor/high-impedance input
Selectable as P-channel open drain output/N-channel open drain output/CMOS output/high-impedance output

Number of ports:

	Input ports	Output ports	Input-output ports
Chip products	1 port × 4 bits	1 port × 4 bits (mask option)	4 ports × 4 bits

i. Melody output

- Melody frequency: 529 Hz to 2979 Hz (@ 32.768 kHz)
- Tone length: 63 varieties
- Tempo: 15 varieties
- Melody data: Stored in program memory
- Buzzer driver signal output: 4 kHz (@ 32.768 kHz)

j. LCD driver

Segment-type LCD drivers built-in

The following pin modes can be specified for L0 to L63 by the mask option generator setting. (Refer to the "MOGTOOL Mask Option Generator User's Manual".)

"●" in the table below indicates that that particular function can be selected.

	L0 to L3	L4 to L31	L32 to L35	L36 to L39	L40 to L63
SEG Pins	●	●	●	●	●
COM Pins	●*1	—	—	●*1	—
Output Port Pins	—	—	●*2	—	—

*1 Can be selected as a COM pin in 1-bit unit (L0 to L3, L36 to L39). A maximum of four pins can be selected as COM pins.

*2 Can be selected as an output port in 4-bit unit (L32 to L35).
N-channel open drain output or CMOS output can be specified for each bit.

Number of segments : 64 (60 SEG. × 4 COM. Max.)
Duty : 1/1 to 1/4 duty (fixed to 1/2 duty when at 1/2 bias)
Bias : Selectable as 1/2 or 1/3 bias
OPTION B, OPTION D: Regulator circuit used (0.95/1.90/2.85 V)
OPTION A, OPTION C: Regulator circuit not used (directly connected to the power supply voltage (1.5/3.0/4.5 V))
Frame frequency : 64 Hz (at 1/1, 1/2, 1/4 duty), 85.3 Hz (at 1/3 duty)
Contrast : OPTION B, OPTION D: Adjustable up to 16 levels (in steps of 0.03 V)
OPTION A, OPTION C: Adjustment not available
Display modes : Selectable as all-ON mode/all-OFF mode/power down mode/normal display mode

k. RC oscillation type A/D converter

- 2 channels (time sharing is used)

l. System reset function

- System reset by RESET pin (2 kHz sampling function provided)
- System reset that resets the system when the combined bits (2, 3, or 4 bits) of the input port (Port 0) are all set to a "H" level
  (Whether system reset is disabled or enabled, the number of bits to be combined, and the polarity can be specified by mask option. Refer to the "MOGTOOL Mask Option Generator User's Manual".)
  2 bits : P0.0, P0.1
  3 bits : P0.0, P0.1, P0.2
  4 bits : P0.0, P0.1, P0.2, P0.3

m. Battery check

- Applies to the OPTION C and OPTION D. Does not apply to the OPTION A and OPTION B.
- Function that detects battery low voltage
- Selection of judgment voltage by software (LD1 and LD0 bit settings of BLDCON)

LD1	LD0	Judgment voltage (V)	Accuracy (V)	Comments
0	0	1.8 ±0.1	±0.1	Ta = 25°C
0	1	2.0 ±0.1	±0.1	Ta = 25°C
1	0	2.4 ±0.1	±0.1	Ta = 25°C
1	1	2.6 ±0.1	±0.1	Ta = 25°C

n. Timers, counters

- 8-bit timer: 4 channels
  Selectable as auto-reload mode, capture mode, clock frequency measurement mode
- Watchdog timer: 1 channel
- 100 Hz timer: 1 channel
  1/100 sec. measurement possible
- 15-bit TBC: 1 channel
  1 Hz, 2 Hz, 4 Hz, 8 Hz, 16 Hz, 32 Hz, 64 Hz, 128 Hz signals can be read

o. Serial port

- Mode: UART mode, synchronous mode
- Communication speed in UART mode: 1200 bps, 2400 bps, 4800 bps, 9600 bps
- Clock frequency in synchronous mode: 32.768 kHz (internal clock mode); external clock frequency
- Data length: 5 to 8 bits

p. Interrupt factors

- External interrupt (4 sources) : Selectable as rising edge/falling edge/both rising and falling edges
- Internal interrupt (14 sources) : Watchdog timer interrupt × 1
  Melody end interrupt × 1
  ADC interrupt × 1
  Timer interrupt × 4
  Serial port reception interrupt × 1
  Serial port transmission interrupt × 1
  1/100 timer (10 Hz) interrupt × 1
  Time base interrupt × 4 (2, 4, 16, and 32 Hz)

q. Operating temperature

- –20 to +70°C

r. Power supply voltage

OPTION A, OPTION B (1.5 V versions): 1.3 to 1.7 V

Note: The operation will only be at the battery voltage and no voltage halver circuit can be used.

OPTION C, OPTION D (3.0 V versions): 1.8 to 3.6 V

Note: It is possible to select by software to use the output of the halver circuit as the power supply of the voltage regulator circuit when the battery voltage is in the range 2.4 to 3.6 V, and to use the battery voltage itself as the power supply of the voltage regulator circuit when the battery voltage is in the range 1.8 to 2.4 V.
It is possible to detect whether the battery voltage is 2.4 V or 1.8 V using the BLD function.
• When the halver circuit is ON: 2.4 to 3.6 V
• When the halver circuit is OFF: 1.8 to 2.4 V

s. Supply current

- In the HALT mode, with the LCD display OFF, low-speed operation, and at normal temperature:

OPTION A (1.5 V power supply specification, without the regulator circuit for the LCD bias reference voltage): Typ. 1.06 µA, 2.4 µA Max.

OPTION B (1.5 V power supply specification, with the regulator circuit for the LCD bias reference voltage): Typ. 1.4 µA, 2.8 µA Max.

OPTION C (3.0 V power supply specification, without the regulator circuit for the LCD bias reference voltage): Typ. 0.53 µA, 1.2 µA Max.

OPTION D (3.0 V power supply specification, with the regulator circuit for the LCD bias reference voltage): Typ. 0.70 µA, 1.4 µA Max.

t. Packages available

Package	Product name
Chip (116-pad)	ML63611-xxxWA (Here, 'xxx' denotes the code number.)

1.3 Mask Options

There are eight items in the mask option of the ML63611.

Make the settings for the following items using the MOGTOOL mask option generator. Refer to the "MOGTOOL Mask Option Generator User's Manual" for details of the method of making the settings.

1) Selection of power supply voltage
Select a power supply specification for the power supply voltage to be used as either a 1.5 V power supply specification (1.3 to 1.7 V) or a 3.0 V power supply specification (1.8 to 3.6 V).

 Note:

When a 1.5 V power supply specification (OPTION A and OPTION B) is selected, the halver circuit and the battery low detect circuit cannot be used.

2) Selection of the regulator circuit for the LCD bias reference voltage
Select the LCD bias reference voltage as either the output of the regulator circuit or the power supply voltage.

 Note:

When power is supplied from the battery:

When "Without the regulator circuit for the LCD bias reference voltage" is selected with the mask option, the LCD bias reference voltage will be generated based on the power supply voltage. When a 1.5 V power supply specification is selected, V_{DD1} will be the pin for the LCD bias reference voltage, and when a 3.0 V power supply specification is selected, V_{DD2} will be the pin for the LCD bias reference voltage. In addition, the LCD bias voltage will decrease as the power supply voltage decreases, causing the display density of the LCD panel to thin down.

When "With the regulator circuit for the LCD bias reference voltage" is selected, the display density will be kept constant even if the battery voltage decreases.

3) Selection of the initial state of Port 0
Select the initial state of Port 0 as either "input with pull-down resistor" or "input with pull-up resistor". This selection determines the initial value of P0PUD (P0CON1).

 Note:

This selection applies to all four bits and it is not possible to make this selection separately for each bit.

4) Selection of simultaneous key depression reset function of Port 0
Select the simultaneous key depression reset function and the number of bits (pins) that can be pressed simultaneously.
The pins that are set according to the number of bits pressed simultaneously are fixed as follows:
2 bits: P0.0, P0.1; 3 bits: P0.0, P0.1, P0.2; 4 bits: P0.0, P0.1, P0.2, P0.3.

 Note:

The system reset mode will be entered at the second falling edge of the 1 Hz signal.

5) Selection of MDB pin output voltage level
Select whether to make the output voltage level of the melody output pin (MDB: negative logic) either V_{DD} or V_{SS} when the melody is OFF.

6) SEG/COM/PORT/DATA selection of the LCD driver pins
It is possible to make the pins L0 to L3 and L36 to L39 either SEG pins or COM pins. However, it is a maximum of four pins that can be selected as COM pins.
It is possible to make the pins L32 to L35 either SEG pins or output port pins.
The pins L4 to L31 and L40 to L63 are always SEG pins.
The segment register corresponding to the pins L0 to L63 can also be used as a DATA area.

Notes:

- When the selection is made as output port pins, the selection applies to all four bits.
- When the segment register is selected as the DATA area, the corresponding pins will still be outputting the segment waveforms, and hence should be left open.

7) Selection of the register address and data of the LCD driver pins
The allocation of the register address and data is set for each LCD driver pin.

Note:

It is not possible to make multiple settings for the same address and the same bit.

8) Selection of whether or not to detect stoppage of low-speed clock oscillations
Select whether or not to detect stoppage of the low-speed clock oscillations and to transfer to the system reset mode.

1.4 Function List

Table 1-1 lists the functions of the ML63611. The solid black circles within the chart indicate that the product has the particular function.

Table 1-1 Function List

Function	Symbol	OPTION A	OPTION B	OPTION C	OPTION D	Reference page
Power supply voltage	V_{DD}	1.5 V (1.3 to 1.7 V)		3.0 V (1.8 to 3.6 V)		—
STACK RAM						2-1
Call	STACK	16 levels				2-5
Register	STACK	16 levels				2-6
Internal memory						2-7
Program memory	ROM	8160 (× 16 bits)				2-7
Data memory	RAM	1024 (× 4 bits)				2-8
System reset generation circuit	RST	2 kHz sampling function, simultaneous key depression reset function of Port 0, detection of stoppage of the low-speed clock oscillations				3-2
Interrupt	INT	External: 4 sources; internal: 14 sources				4-1
Clock generator circuit						5-1
Low-speed clock	XT	Crystal oscillation circuit (32.768 kHz)				5-2
		32.768 kHz				
High-speed clock	OSC	RC oscillation circuit, ceramic oscillation circuit				5-3
		200 kHz max.		700 kHz max.		
Time base counter	TBC	15 bits × 1 ch				6-1
Timer	TIMER	8 bits × 4 ch				7-1
100 Hz timer counter	100HzTC	1 ch				8-1
Watchdog timer	WDT	1 ch				9-1
Port						10-1
Input-only port	INPUT PORT	1 port × 4 bits				—
Port 0	P0	4 bits (P0.0, P0.1, P0.2, P0.3)				10-2
Output-only port	OUTPUT PORT	1 port × 4 bits (selectable by mask option)				—
Port LP	LP0	4 bits (LP0.0, LP0.1, LP0.2, LP0.3)				13-9
Input-output port	I/O PORT	4 ports × 4 bits				—
Port A	PA	4 bits (PA.0, PA.1, PA.2, PA.3)				10-7
Port B	PB	4 bits (PB.0, PB.1, PB.2, PB.3)				10-11
Port C	PC	4 bits (PC.0, PC.1, PC.2, PC.3)				10-19
Port E	PE	4 bits (PE.0, PE.1, PE.2, PE.3)				10-27
Melody driver	MELODY	529 to 2979 Hz				11-1
Serial port	SIO	Asynchronous (UART) or synchronous				12-1
LCD driver	LCD	64 segments (60 seg. by 4 com.)				13-1
Segment register	SEGREG	256 × 4 bits				13-10
Bias generator circuit	BIAS	1/2 or 1/3 bias				13-12
Battery low detect circuit	BLD	—	—	1.8 V, 2.0 V, 2.4 V, 2.6 V		14-1
Power supply circuit (POWER)						15-1
Halver circuit	V/H	—	—	●	●	15-1
Voltage regulator circuit						15-1
For internal logic circuits	V/R1	●	●	●	●	15-1
For low-speed oscillation	V/R2	●	●	●	●	15-1
For LCD bias generation reference	V/R3	—	●	—	●	15-1
A/D converter	ADC	2 channels, of RC oscillation type				16-1

OPTION A (C): 1.5 V (3.0 V), Without regulator circuit for LCD bias

OPTION B (D): 1.5 V (3.0 V), With regulator circuit for LCD bias

1.5 Block Diagram

Block diagrams of the ML63611 are shown in Figures 1-1, 1-2, 1-3 and 1-4.

Asterisks (*) indicate port secondary functions.

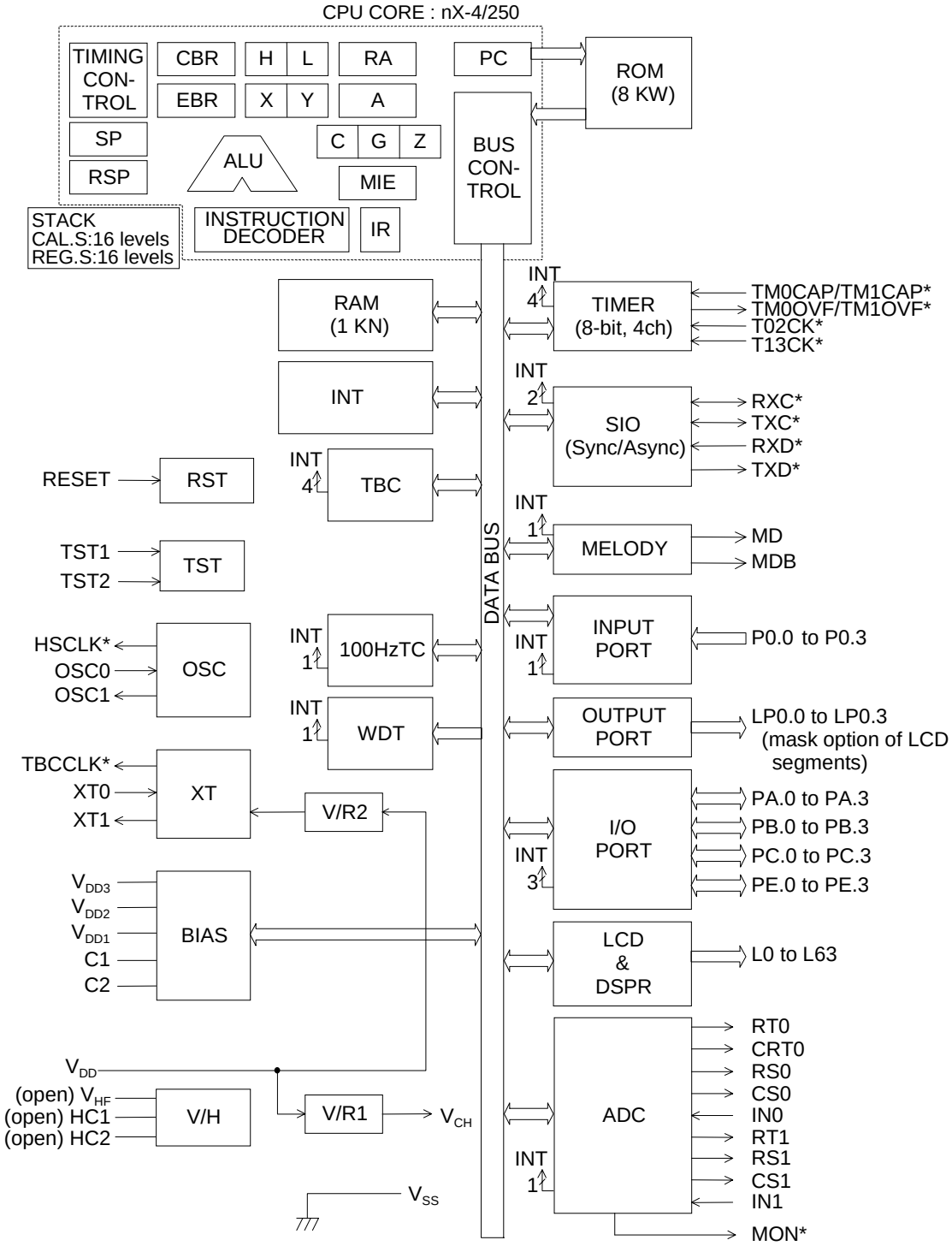


Figure 1-1 OPTION A Block Diagram

Asterisks (*) indicate port secondary functions.

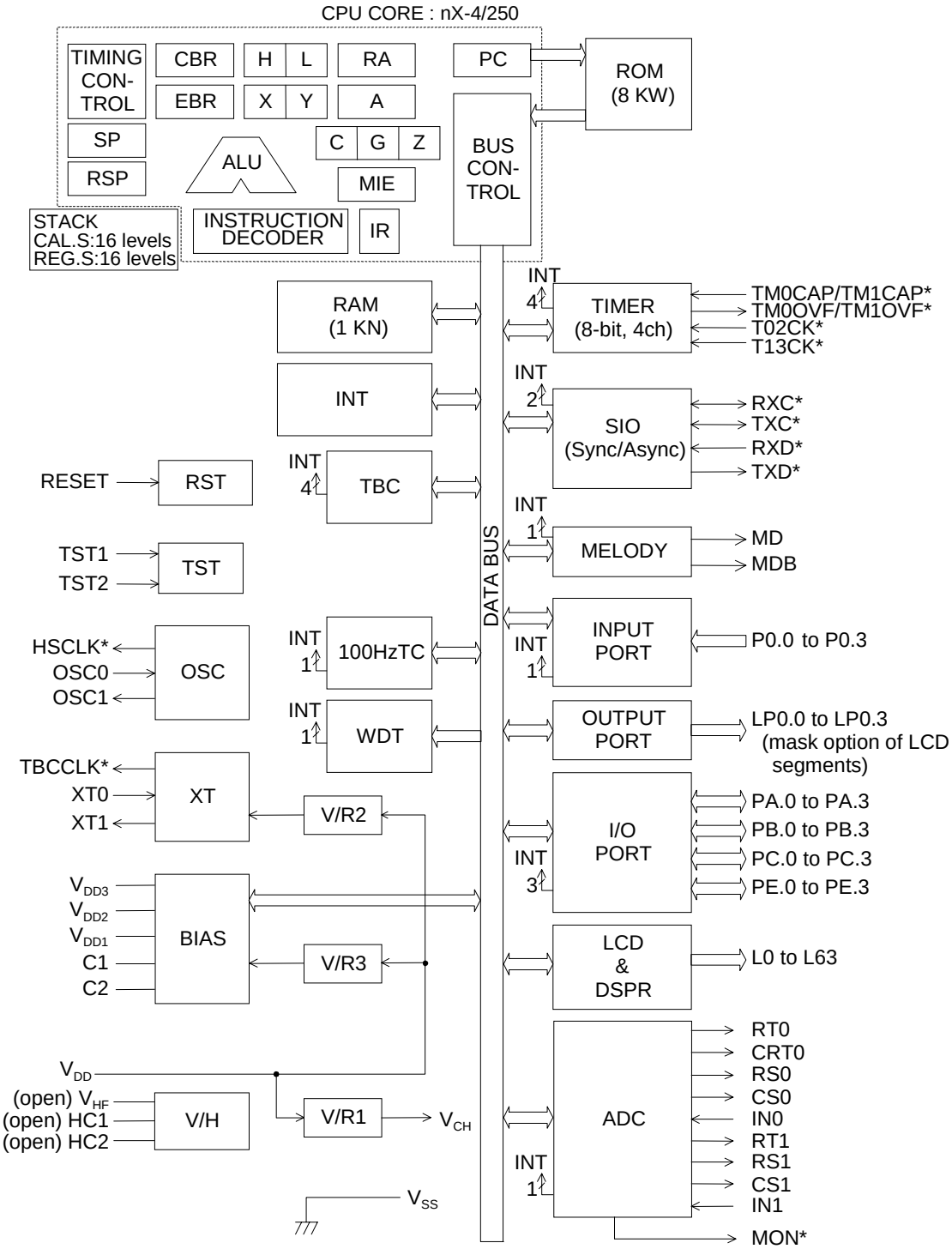


Figure 1-2 OPTION B Block Diagram

Asterisks (*) indicate port secondary functions.

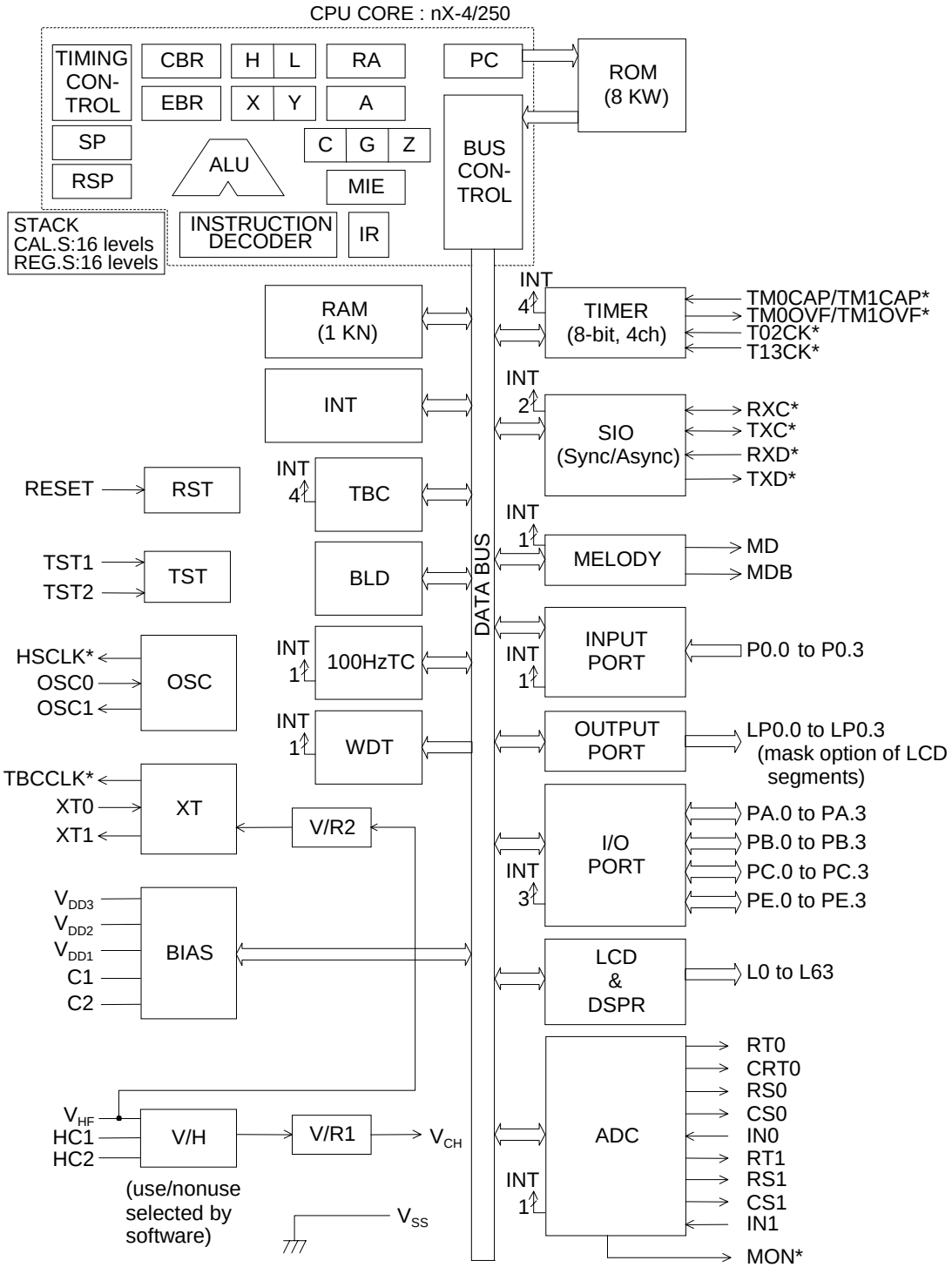


Figure 1-3 OPTION C Block Diagram

Asterisks (*) indicate port secondary functions.

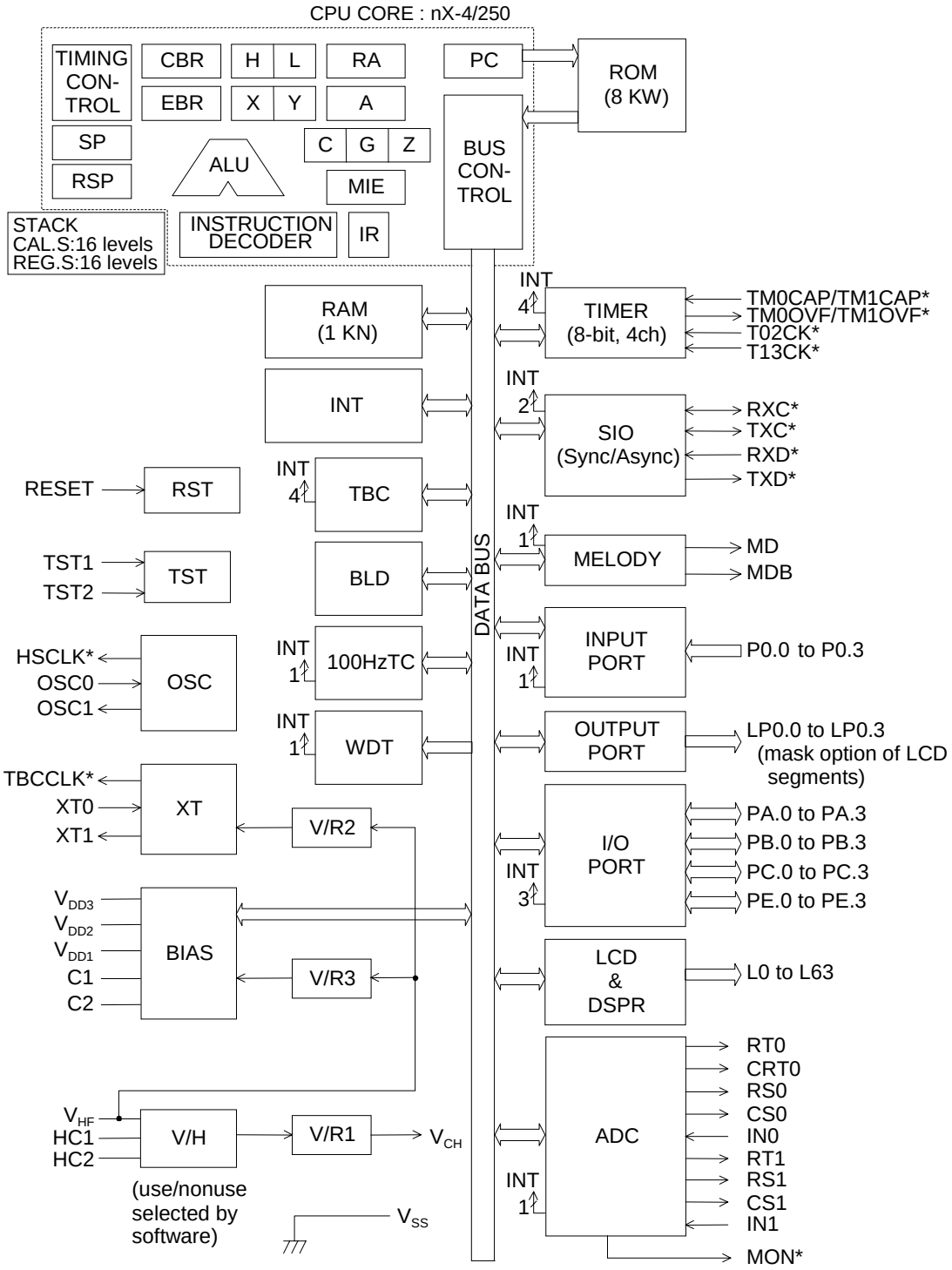
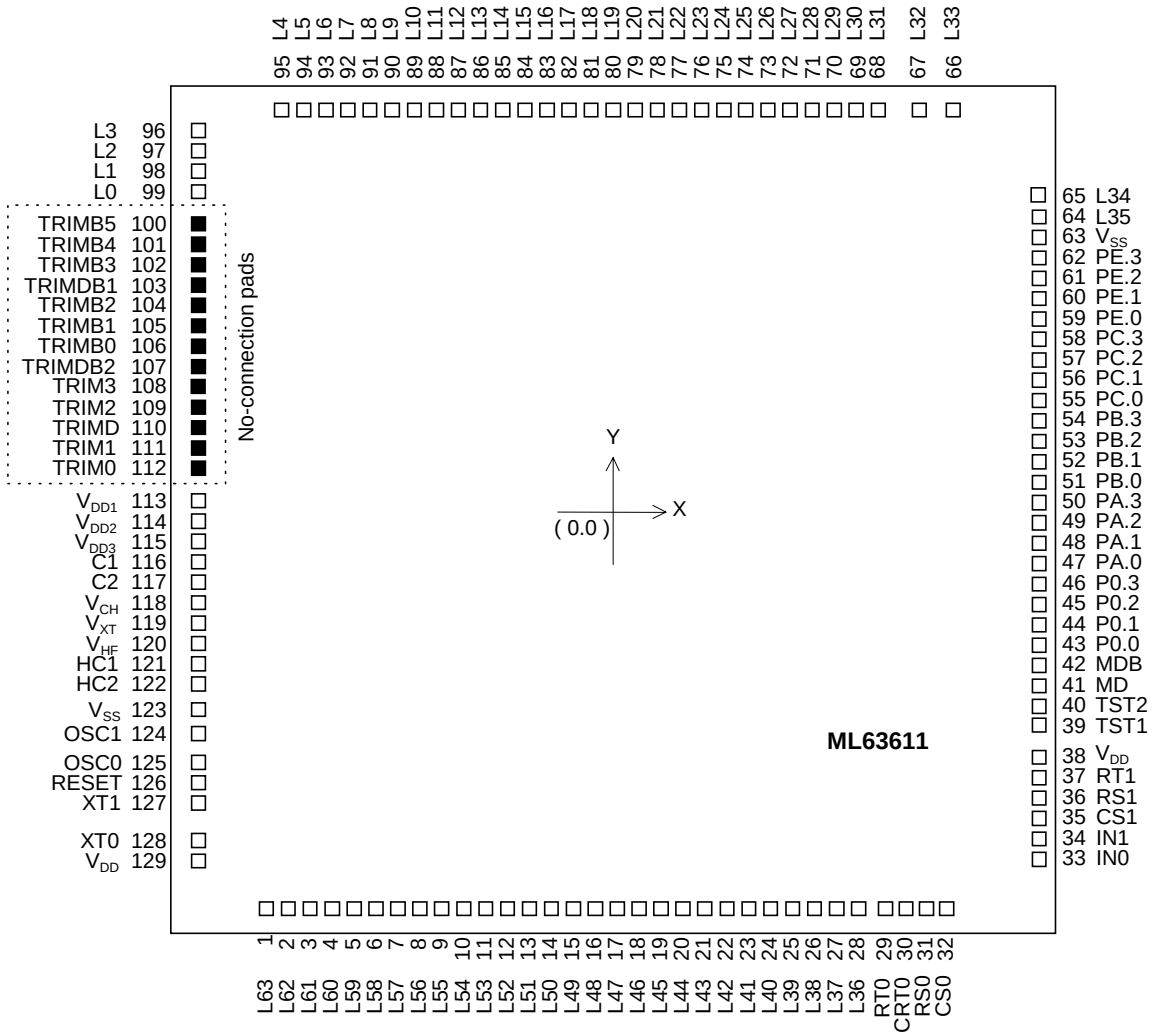


Figure 1-4 OPTION D Block Diagram

1.6 Pin Configuration

1.6.1 ML63611 Pin Configuration

The ML63611 chip pin configuration and pad coordinates are shown in Figures 1-5 and Table 1-2 respectively.



Chip Size	:	5.20 mm × 5.20 mm
Chip Thickness	:	350 µm (typ.) (280 µm: available as required)
Coordinate Origin	:	Chip center
Pad Hole Size	:	80 µm × 80 µm
Pad Size	:	90 µm × 90 µm
Minimum Pad Pitch	:	115 µm
Number of bonding pads	:	116 (total number of pads: 129)

Notes: The chip substrate voltage is V_{SS}.
Do not bond pins 100 to 112 (marked by "■"). Leave them open.

Figure 1-5 ML63611 Chip Pin Configuration (Top View)

Table 1-2 ML63611 Pad Coordinates

Chip center: X = 0, Y = 0

Pad No.	Pad Name	X (µm)	Y (µm)	Pad No.	Pad Name	X (µm)	Y (µm)
1	L63			34	IN1		
2	L62			35	CS1		
3	L61			36	RS1		
4	L60			37	RT1		
5	L59			38	V_{DD}		
6	L58			39	TST1		
7	L57			40	TST2		
8	L56			41	MD		
9	L55			42	MDB		
10	L54			43	P0.0		
11	L53			44	P0.1		
12	L52			45	P0.2		
13	L51			46	P0.3		
14	L50			47	PA.0		
15	L49	Undetermined		48	PA.1	Undetermined	
16	L48			49	PA.2		
17	L47			50	PA.3		
18	L46			51	PB.0		
19	L45			52	PB.1		
20	L44			53	PB.2		
21	L43			54	PB.3		
22	L42			55	PC.0		
23	L41			56	PC.1		
24	L40			57	PC.2		
25	L39			58	PC.3		
26	L38			59	PE.0		
27	L37			60	PE.1		
28	L36			61	PE.2		
29	RT0			62	PE.3		
30	CRT0			63	V_{SS}		
31	RS0			64	L35		
32	CS0			65	L34		
33	IN0			66	L33		

Table 1-2 ML63611 Pad Coordinates (continued)

Chip center: X = 0, Y = 0

Pad No.	Pad Name	X (μm)	Y (μm)
67	L32		
68	L31		
69	L30		
70	L29		
71	L28		
72	L27		
73	L26		
74	L25		
75	L24		
76	L23		
77	L22		
78	L21		
79	L20		
80	L19		
81	L18		
82	L17		
83	L16	Undetermined	
84	L15		
85	L14		
86	L13		
87	L12		
88	L11		
89	L10		
90	L9		
91	L8		
92	L7		
93	L6		
94	L5		
95	L4		
96	L3		
97	L2		
98	L1		
99	L0		
100	TRIMB5	No-connection	
101	TRIMB4		
102	TRIMB3		
103	TRIMDB1		
104	TRIMB2		
105	TRIMB1		
106	TRIMB0		
107	TRIMDB2		
108	TRIM3		
109	TRIM2		
110	TRIMD		
111	TRIM1		
112	TRIM0		
113	V_{DD1}		
114	V_{DD2}	Undetermined	
115	V_{DD3}		
116	C1		
117	C2		
118	V_{CH}		
119	V_{XT}		
120	V_{HF}		
121	HC1		
122	HC2		
123	V_{SS}		
124	OSC1		
125	OSC0		
126	RESET		
127	XT1		
128	XT0		
129	V_{DD}		

1.7 Pin Descriptions

1.7.1 Descriptions of the Basic Functions of Each Pin

The basic functions of each pin of the ML63611 are listed in Table 1-3. Use of a slash ("/") in a pin name indicates that the pin has a secondary function. See Table 1-4 for the secondary functions.
In the I/O column, "—" indicates a power supply pin, "I" indicates an input pin, "O" indicates an output pin, and "I/O" indicates an input/output pin.

Table 1-3 Pin Description (Basic Functions)

Classification	Pin name	Pad No.	I/O	Function
Power Supply	V_{DD}	38, 129	—	Positive power supply
	V_{SS}	63, 123		Negative power supply
	V_{DD1}	113		Power supply pins for LCD bias voltage (internally generated): A capacitor (1.0 μF) should be connected between V_{DD1} and V_{SS}, between V_{DD2} and V_{SS}, and between V_{DD3} and V_{SS}. For the OPTION A, connect V_{DD1} with V_{DD}; for the OPTION C, connect V_{DD2} with V_{DD}.
	V_{DD2}	114		
	V_{DD3}	115		
	C1	116		Capacitor connection pins for LCD bias voltage generation: A capacitor (1.0 μF) should be connected between C1 and C2.
	C2	117		
	V_{HF}	120		Power supply pin for the internal regulator: A capacitor (0.1 μF) should be connected between this pin and V_{SS}. Leave this pin open for the OPTION A and OPTION B.
	V_{XT}	119		Power supply pin for the voltage regulator circuit for low-speed oscillation: A capacitor (0.1 μF) should be connected between this pin and V_{SS}.
	V_{CH}	118		Power supply pin for the voltage regulator circuit for internal logic: A capacitor C_1 (0.1 μF) should be connected between this pin and $V_{SS.}$
	HC1	121		Capacitor connection pins for the halver circuit: A capacitor (0.1 μF) should be connected between HC1 and HC2. Leave these pins open for the OPTION A and OPTION B.
	HC2	122		
Oscillation	XT0	128	I	Low-speed clock oscillation pins: Connect a crystal between XT0 and XT1, and connect capacitor (C_G) between XT0 and $V_{SS.}$
	XT1	127	O	
	OSC0	125	I	High-speed clock oscillation pins: Ceramic oscillation or RC oscillation is selected by the software. In the OPTION A and OPTION B, only RC oscillation is available. If ceramic oscillation is selected, connect a ceramic resonator between OSC0 and OSC1, and connect capacitor (C_{L0}, C_{L1}) between OSC0 and V_{SS} and between OSC1 and V_{SS}. If RC oscillation is selected, connect external oscillation resistor (R_{OSH}) between OSC0 and OSC1.
	OSC1	124	O	
Test	TST1	39	I	Input pins for testing: A pull-down resistor is internally connected to these pins.
	TST2	40		

Table 1-3 Pin Description (Basic Functions) (continued)

Classification	Pin name	Pad No.	I/O	Function
Reset	RESET	126	I	Reset input pin: 2 kHz sampling circuit is equipped. Holding this pin to "H" level for 1 ms or more puts this device into a reset state. Then, setting this pin to "L" level starts executing an instruction from address 0000H. A pull-down resistor is internally connected to this pin.
Melody	MD	41	O	Melody output pin (non-inverted output)
	MDB	42		Melody output pin (inverted output): V_{DD} or V_{SS} is selectable for the pin output voltage when melody output is turned off.
Port	P0.0	43	I	4-bit input port: Pull-up resistor input, pull-down resistor input, or high-impedance input is selectable for each bit. A system reset function is provided that resets the system when there is a simultaneous key depression of multiple bits.
	P0.1	44		
	P0.2	45		
	P0.3	46		
	PA.0	47	I/O	4-bit input-output ports: In input mode, pull-up resistor input, pull-down resistor input, or high-impedance input is selectable for each bit. In output mode, P-channel open drain output, N-channel open drain output, CMOS output, or high-impedance output is selectable for each bit.
	PA.1	48		
	PA.2	49		
	PA.3	50		
	PB.0	51	I/O	
	PB.1	52		
	PB.2	53		
	PB.3	54		
	PC.0	55	I/O	
	PC.1	56		
	PC.2	57		
	PC.3	58		
	PE.0	59	I/O	
	PE.1	60		
	PE.2	61		
	PE.3	62		

Table 1-3 Pin Description (Basic Functions) (continued)

Classification	Pin name	Pad No.	I/O	Function
LCD	L0	99	O	These pins can be selected as LCD segment signal output pins (L0 to L3) or common signal output pins by the mask option. The common signal can be selected from among COM1 to COM4.
	L1	98		
	L2	97		
	L3	96		
	L4	95	O	Output pins dedicated to the LCD segment signal (L4 to L31).
	L5	94		
	L6	93		
	L7	92		
	L8	91		
	L9	90		
	L10	89		
	L11	88		
	L12	87		
	L13	86		
	L14	85		
	L15	84		
	L16	83		
	L17	82		
	L18	81		
	L19	80		
	L20	79		
	L21	78		
	L22	77		
	L23	76		
	L24	75		
	L25	74		
	L26	73		
	L27	72		
	L28	71		
	L29	70		
	L30	69		
	L31	68		
	L32/ LP0.3	67	O	These pins can be selected as LCD segment signal output pins (L32 to L35) or output port pins (LP0.0 to LP0.3) by the mask option.
	L33/ LP0.2	66		
	L34/ LP0.1	65		
	L35/ LP0.0	64		
	L36	28	O	These pins can be selected as output pins dedicated to the LCD segment signal (L36 to L39) or common signal output pins by the mask option. The common signal can be selected from among COM1 to COM4.
	L37	27		
	L38	26		
	L39	25		

Table 1-3 Pin Description (Basic Functions) (continued)

Classification	Pin name	Pad No.	I/O	Function
LCD	L40	24	O	Output pins dedicated to the LCD segment signal (L40 to L63).
	L41	23		
	L42	22		
	L43	21		
	L44	20		
	L45	19		
	L46	18		
	L47	17		
	L48	16		
	L49	15		
	L50	14		
	L51	13		
	L52	12		
	L53	11		
	L54	10		
	L55	9		
	L56	8		
	L57	7		
	L58	6		
	L59	5		
	L60	4		
	L61	3		
	L62	2		
	L63	1		
A/D Converter	RT0	29	O	Resistance temperature sensor connection pin (for channel 0)
	CRT0	30		Resistance/capacitance temperature sensor connection pin (for channel 0)
	RS0	31		Reference resistor connection pin (for channel 0)
	CS0	32		Reference capacitor connection pin (for channel 0)
	IN0	33	I	Input pin for RC oscillator circuit (for channel 0)
	IN1	34		Input pin for RC oscillator circuit (for channel 1)
	CS1	35	O	Reference capacitor connection pin (for channel 1)
	RS1	36		Reference resistor connection pin (for channel 1)
	RT1	37		Resistance temperature sensor connection pin (for channel 1)

1.7.2 Descriptions of the Secondary Functions of Each Pin

The secondary functions of each pin of the ML63611 are listed in Table 1-4.

Table 1-4 Pin Description (Secondary Functions)

Classification	Pin name	Pad No.	I/O	Function
External Interrupt	PB.0/INT0	51	I	External 0 interrupt input pins: The change of input signal level causes an interrupt to occur. The Port B Interrupt Enable register (PBIE) enables or disables an interrupt for each bit.
	PB.1/INT0	52		
	PB.2/INT0	53		
	PB.3/INT0	54		
	PC.0/INT1	55	I	External 1 interrupt input pins: The change of input signal level causes an interrupt to occur. The Port C Interrupt Enable register (PCIE) enables or disables an interrupt for each bit.
	PC.1/INT1	56		
	PC.2/INT1	57		
	PC.3/INT1	58		
	PE.3/INT2	62	I	External 2 interrupt input pin: The change of input signal level causes an interrupt to occur.
	P0.0/INT5	43	I	External 5 interrupt input pins: The change of input signal level causes an interrupt to occur. The Port 0 Interrupt Enable register (P0IE) enables or disables an interrupt for each bit.
	P0.1/INT5	44		
	P0.2/INT5	45		
	P0.3/INT5	46		
Capture	PB.0/TM0CAP	51	I	Timer 0 capture trigger input pin
	PB.1/TM1CAP	52		Timer 1 capture trigger input pin
Timer	PB.0/TM0OVF	51	O	Timer 0 (TM0) overflow flag output pin
	PB.1/TM1OVF	52		Timer 1 (TM1) overflow flag output pin
	PB.2/T02CK	53	I	External clock input pin for Timer 0 (TM0) and Timer 2 (TM2)
	PB.3/T13CK	54		External clock input pin for Timer 1 (TM1) and Timer 3 (TM3)
Serial Port	PC.0/RXD	55	I	Serial port receive data input pin
	PC.1/TXC	56	I/O	Sync serial port clock input-output pin: Transmit sync clock input-output pin when a serial port is used synchronously. Transmit clock output when this device is used as a master processor. Transmit clock input when this device is used as a slave processor.
	PC.2/RXC	57		Sync serial port clock input-output pin: Receive sync clock input-output pin when a serial port is used synchronously. Receive clock output when this device is used as a master processor. Receive clock input when this device is used as a slave processor.
	PC.3/TXD	58	O	Serial port transmit data output pin
Monitor	PE.0/MON	59	O	Pin for monitoring the RC oscillation clock for the A/D converter
	PE.1/TBCCLK	60	O	Low-speed oscillation clock (TBCCLK) monitoring pin
	PE.2/HSCLK	61	O	High-speed oscillation clock (HSCLK) monitoring pin

1.7.3 Handling of Unused Pins

Table 1-5 shows how unused pins should be handled.

Table 1-5 Handling of Unused Pins

Pin	Recommended pin handling
OSC0, OSC1	Open
V_{HF}, HC1, HC2	Open
TST1, TST2	Connect to V_{SS}
P0.0 to P0.3	Open
PA.0 to PA.3	Open
PB.0 to PB.3	Open
PC.0 to PC.3	Open
PE.0 to PE.3	Open
L0 to L63	Open
MD, MDB	Open
RT0, CRT0, RS0, CS0	Open
IN0, IN1	Open
CS1, RS1, RT1	Open

Notes:

1. If a pin set as a high impedance input is left unconnected, the supply current may become excessive. Therefore, it is recommended that unused input ports and input/output ports be set as inputs with either a pull-down or pull-up resistor.
2. When test pins TST1 and TST2 are left unconnected, malfunction may result if there is a large amount of external noise. Therefore, it is recommended to permanently connect TST1 and TST2 to V_{SS}.

1.8 Basic Timing

1.8.1 Basic Timing of CPU Operation

The low-speed oscillation clock from the XT0/XT1 pins or the high-speed oscillation clock from the OSC0/OSC1 pins are used without frequency division as the system clock (CLK). The system clock signal is in phase with the signal from the XT1 pin or the OSC1 pin.

As shown in Figure 1-6, a single machine cycle is composed of two states, S1 and S2. One state is the interval from a falling edge of CLK to the falling edge of the next CLK.

Instructions are processed in machine cycle units and each instruction is executed in 1 to 3 machine cycles. Instructions are classified according to the number of machine cycles: 1-machine-cycle instructions (M1), 2-machine-cycle instructions (M1 + M2), and 3-machine-cycle instructions (M1 + M2 + M3).

Most instructions are executed in 1 machine cycle.

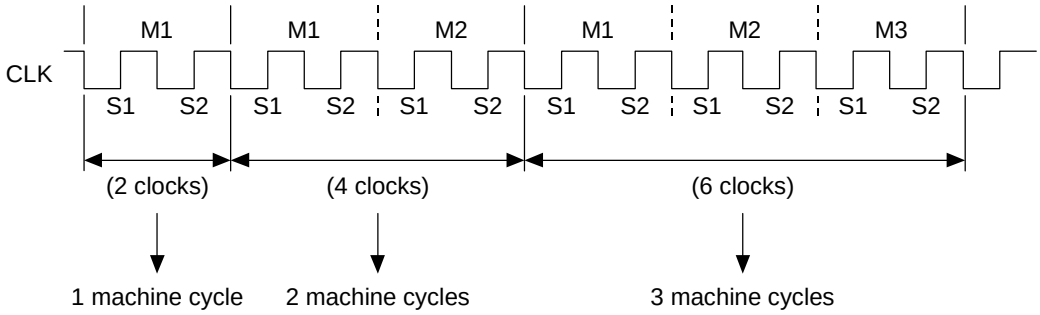


Figure 1-6 Clock Configuration of Each Machine Cycle

1.8.2 Port I/O Basic Timing

Figure 1-7 shows the basic I/O timing.

During the execution of an instruction that outputs data to a port, setting data (data A) is output at the rising edge of the clock in the S2 state during the machine cycle of that instruction.

During the execution of an instruction that inputs data from a port, data at the input pin (data B) is captured internally while the clock is at a "H" level in the S1 state during the machine cycle of that instruction. That data is transferred to the accumulator at the start of the next machine cycle.

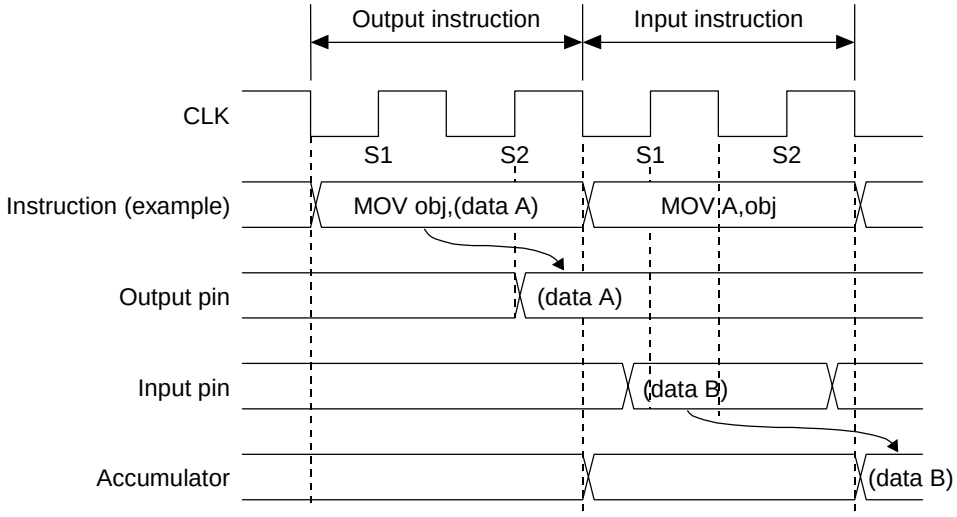


Figure 1-7 Port I/O Basic Timing

Notes:

Regarding input signals

"0" will be captured in the internal register if a "L" level is input to the input pin even once (① of Figure 1-8) during the data capture interval.

"1" will be captured in the internal register only if a "H" level is maintained (② of Figure 1-8) throughout the data capture interval.

Therefore, if noise occurs in the input data, implement noise reduction measures with the program and peripheral devices.

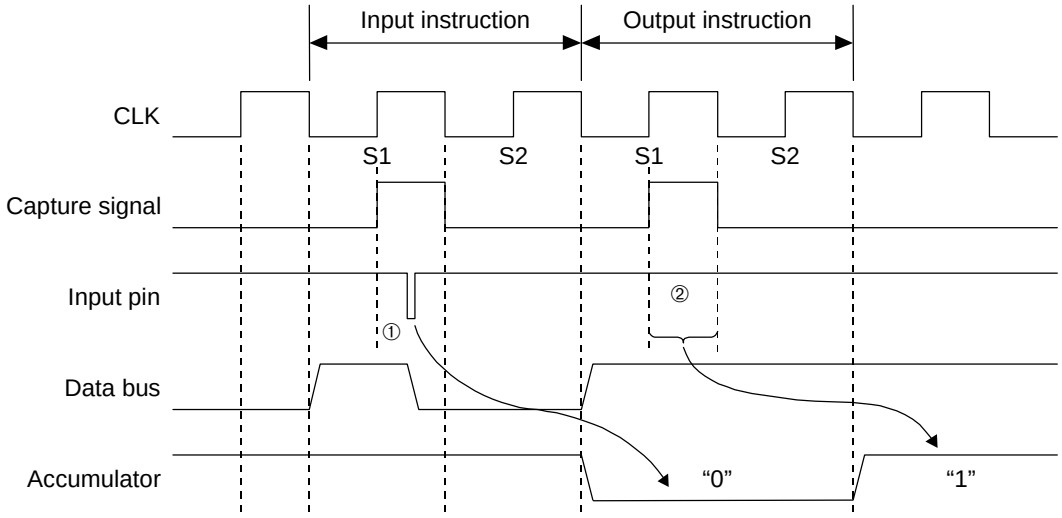


Figure 1-8 Input Data Example

1.8.3 Interrupt Basic Timing

Figure 1-9 shows the basic interrupt timing.

As shown in the figure, when an interrupt factor is generated, the interrupt factor is sampled at the falling edge of CLK and an interrupt request (IRQ) is set at the first half of S1.

When an interrupt condition is established and the CPU receives an interrupt, the interrupt routine will start beginning from the next machine cycle.

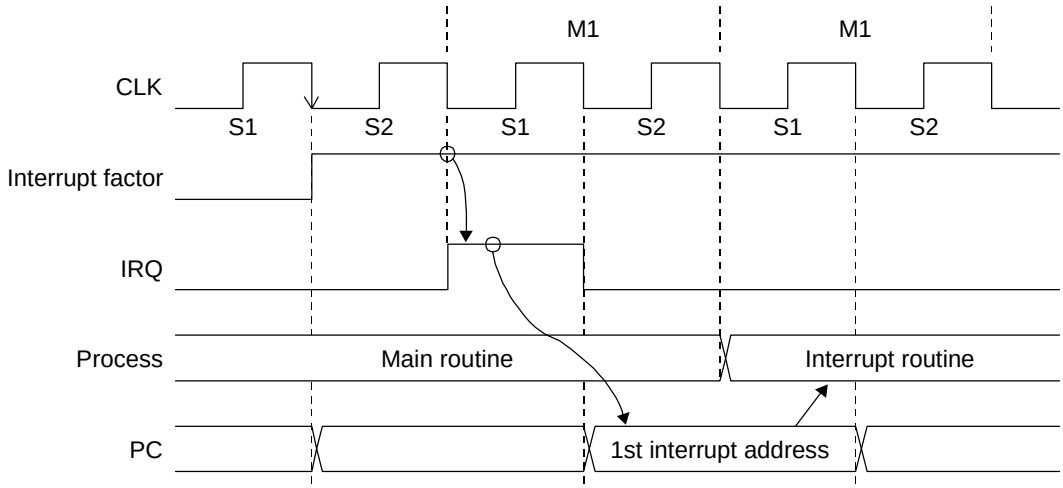


Figure 1-9 Interrupt Basic Timing

Chapter 2

CPU and Memory Spaces

2. CPU and Memory Spaces

2.1 Overview

The ML63611 is equipped with Oki's original CPU core nX-4/250.

The instruction set of the nX-4/250 core consists of 407 types of instructions.

The memory space consists of a 16-bit wide program memory space and a 4-bit wide data memory space. A stack for saving the program counter during a subroutine call or interrupt (call stack) and a stack for saving registers during a PUSH instruction (register stack) are provided separately from the memory space.

The program memory space is used for program data, ROM table data and melody note data.

In the data memory space, special function registers (SFRs) are located in BANK 0, the LCD display register (DSPR) in BANK 1, and data RAM in BANKS 2 to 5.

2.2 Registers

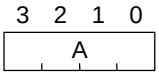
The nX-4/250 core processes data mainly with the accumulator and register set.

The register set is a programming model consisting of the HL and XY registers that store data memory addresses, the current bank register (CBR), the extra bank register (EBR), the RA register that stores program memory addresses, registers that control program flow, and registers that control flags and memory.

2.2.1 Accumulator (A)

The accumulator (A) is the central register for various arithmetic operations.

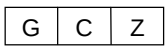
At system reset, the accumulator is initialized to "0". When an interrupt occurs, a "PUSH HL" instruction can be used if necessary to save the accumulator on the register stack. The accumulator can be restored with a "POP HL" instruction.



Accumulator

2.2.2 Flag Register

The flag register consists of 3 flags: the carry flag (C), the zero flag (Z) and the G flag (G). When an interrupt occurs, a "PUSH HL" instruction can be used if necessary to save the flag register on the register stack. The flag register can be restored with a "POP HL" instruction.



Flag register

2.2.2.1 Carry Flag (C)

The carry flag (C) is a 1-bit flag that is loaded with a carry during addition or a borrow during subtraction. At system reset, the carry flag is initialized to "0".

2.2.2.2 Zero Flag (Z)

The zero flag (Z) is a 1-bit flag that is set to "1" when the contents of the accumulator (A) are loaded with "0H". The zero flag is set to "0" when the contents of the accumulator (A) are loaded with a value other than "0H". However, the XCH instruction does not change the zero flag. At system reset, the zero flag is initialized to "0".

2.2.2.3 G Flag (G)

The G flag (G) changes to "1" when the HL, XY or RA registers overflow as the result of execution of a post-increment register indirect addressing instruction or as the result of an increment instruction for the HL, XY or RA registers. At system reset, the G flag is initialized to "0".

2.2.3 Master Interrupt Enable Flag (MIE)

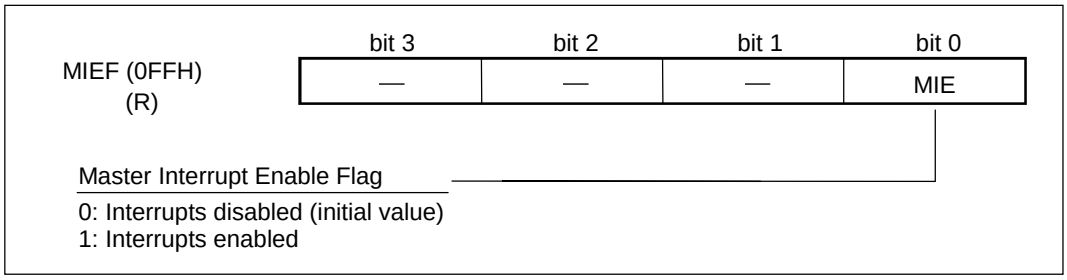
MIE (bit 0 of MIEF) is a flag that disables or enables all interrupts except for the watchdog timer interrupt. MIEF is a 4-bit register in which bit 0 is the master interrupt enable flag (MIE).

If MIE is "0", all interrupts are disabled. If MIE is "1", all interrupts are enabled (with the exception of the watchdog timer).

When any interrupt is received, MIE is cleared to "0". MIE is set to "1" by execution of a return from interrupt instruction (RTI instruction).

If multi-level interrupt processing is to be performed, execute a RTI instruction (MIE←"1") during the interrupt processing routines.

At system reset, MIE is initialized to "0". MIEF only supports data reference (R) of data memory through addressing instructions.



Note:

When setting MIE, use "EI" instructions (MIE← "1") and "DI" instructions (MIE← "0").

2.2.4 Current Bank Register (CBR), Extra Bank Register (EBR), HL Register (HL), XY Register (XY)

The CBR, EBR, HL, and XY registers are used for indirect addressing of data memory.

The CBR and EBR registers indicate the data memory bank. The HL and XY registers indicate addresses in the bank. CBR is also used in combination with 8-bit data in the instruction code for direct addressing within the current bank.

Figure 2-1 shows the various register combinations.

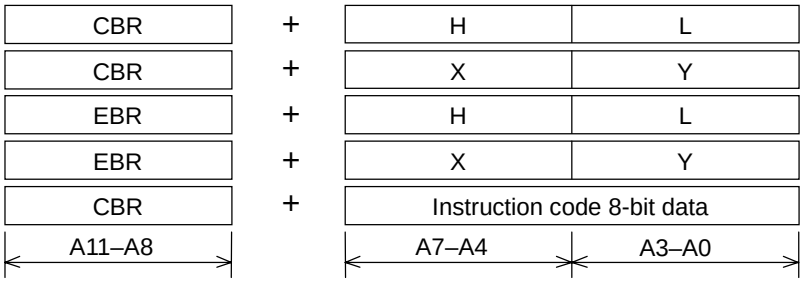


Figure 2-1 Various Register Combinations

A11 to A0 in Figure 2-1 indicate data memory addresses (4K nibbles max.).

At system reset, the CBR, EBR, HL, and XY registers are initialized to "0".

When an interrupt occurs, a "PUSH HL" or "PUSH XY" instruction can be used if necessary to save the CBR, EBR, HL, and XY registers on the register stack. These registers can be restored with a "POP HL" or "POP XY" instruction.

The CBR, EBR, HL, and XY registers are assigned to special function register (SFR) addresses 0F9H to 0FEH.

		bit 3	bit 2	bit 1	bit 0
EBR	(0FEH) (R/W)	e_3	e_2	e_1	e_0
CBR	(0FDH) (R/W)	c_3	c_2	c_1	c_0
H	(0FCH) (R/W)	h_3	h_2	h_1	h_0
L	(0FBH) (R/W)	l_3	l_2	l_1	l_0
X	(0FAH) (R/W)	x_3	x_2	x_1	x_0
Y	(0F9H) (R/W)	y_3	y_2	y_1	y_0

2.2.5 Program Counter (PC)

The program counter (PC) is a counter with 16 valid bits that specifies the program memory space.

2.2.6 RA Registers (RA3, RA2, RA1, RA0)

The RA registers are used for indirect program memory addressing (ROM table reference instructions).

Figure 2-2 shows the address configuration of the RA registers.

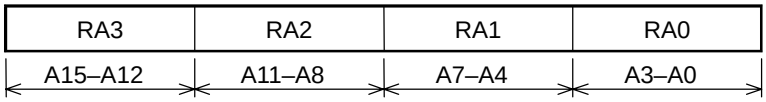
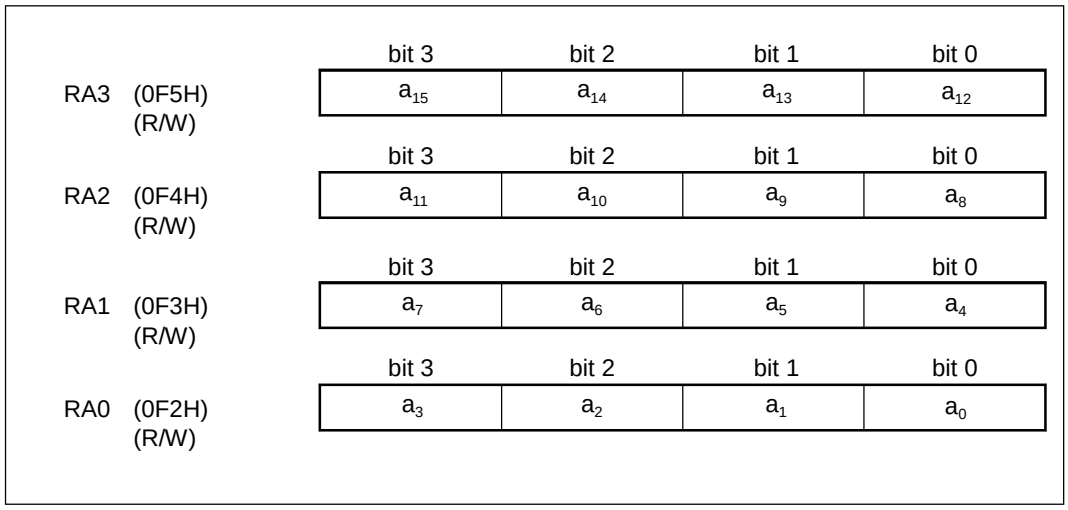


Figure 2-2 Address Configuration of RA3 to RA0 Registers

Within the A15 to A0 of Figure 2-2, A14 to A0 indicate program memory addresses (32K words max.).

RA3 to RA0 are assigned to special function register (SFR) addresses 0F2H to 0F5H.



At system reset, RA3 to RA0 are initialized to "0".

Note:

When executing a ROM table reference instruction that uses RA registers, do not use addresses located in the SFR area to transfer ROM table data to RA registers, otherwise indirect addressing of program memory will not operate properly.

2.2.7 Stack Pointer (SP) and Call Stack

The stack pointer (SP) is a pointer that indicates the call stack address where the program counter is saved when a subroutine call or interrupt occurs.

The SP is a 4-bit up/down counter that is incremented during stack saves and is decremented during stack restores.

The call stack has 16 levels from address 0H to address 0FH. Because the hardware requires 1 level of the call stack during program execution, only 15 levels can be used for stack saves. The contents of the call stack cannot be read or written by the program.

Figure 2-3 shows the relation between SP and the call stack.

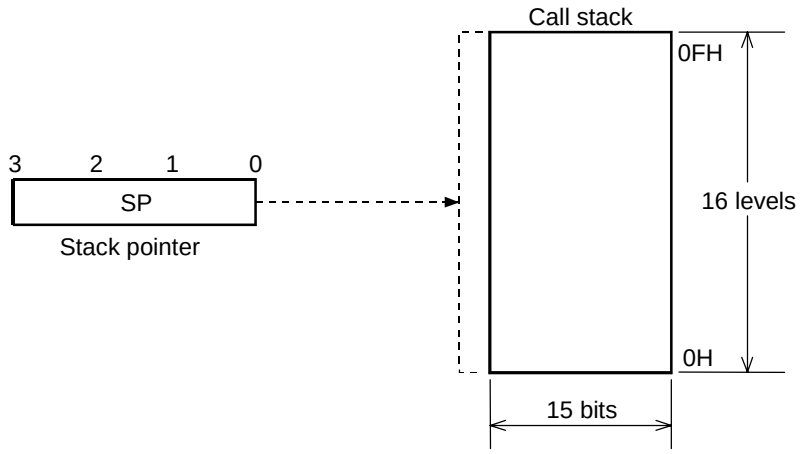


Figure 2-3 Relation between SP and Call Stack

SP is assigned to special function register (SFR) address 0F7H.

	bit 3	bit 2	bit 1	bit 0
SP (0F7H) (R)	SP3	SP2	SP1	SP0

At system reset, SP is initialized to "0" and points to address "0H" of the call stack. SP is a read-only register and writes are invalid.

2.2.8 Register Stack Pointer (RSP) and Register Stack

The register stack pointer (RSP) is a pointer that indicates the register stack address for saving various registers.

RSP is a 4-bit up/down counter that is incremented during stack saves (execution of PUSH instructions) and is decremented during stack restores (execution of POP instructions).

The register stack has 16 levels from address 0H to address 0FH. The contents of the register stack cannot be read or written by the program.

Figure 2-4 shows the relation between RSP and the register stack.

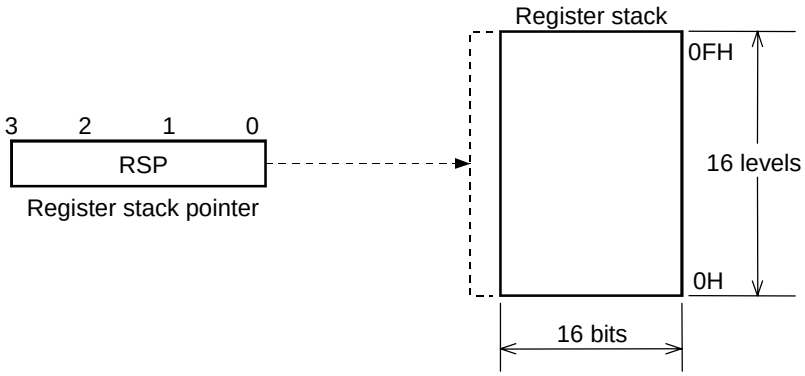


Figure 2-4 Relation between RSP and Register Stack

The various registers shown in Figure 2-5 are saved onto and restored from the register stack by PUSH and POP instructions.

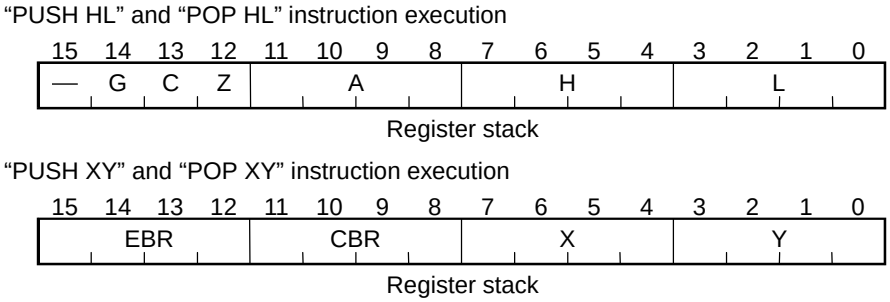


Figure 2-5 Register Save/Restore by Execution of PUSH/POP Instructions

RSP is assigned to special function register (SFR) address 0F6H.

	bit 3	bit 2	bit 1	bit 0
RSP (0F6H) (R/W)	rsp3	rsp2	rsp1	rsp0

At system reset, RSP is initialized to "0" and points to address "0H" of the register stack.

2.3 Memory Spaces

2.3.1 Program Memory Space

The program memory space is the read-only memory that stores program data.

The program memory space has a data length of 16 bits and extends from address 0000H to address 1FFFH.

In addition to program data, the program memory can also store ROM table data and the melody data. Figure 2-6 shows the configuration of the program memory space.

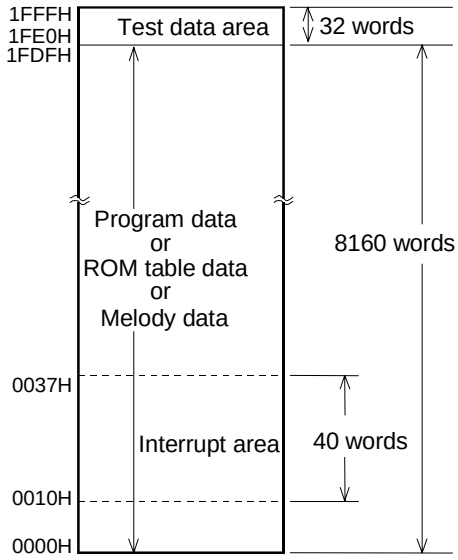


Figure 2-6 Program Memory Space Configuration

After system reset, instruction execution begins at address 0000H. The interrupt area from address 0010H to address 0037H contains starting addresses of the interrupt processing routines that are executed when interrupts are generated. (Refer to Chapter 4, "Interrupt".)

ROM table data is transferred to data memory by ROM table reference instructions.

The melody data defines the tone, tone length, and end tone used in the melody circuit. After an MSA instruction specifies the starting address, the melody data is automatically transferred to the melody circuit when a melody data interrupt occurs. (Refer to Chapter 11, "Melody Driver".)

Because the test data area contains program data for testing, it cannot be used as a program data area.

2.3.2 Data Memory Space

The data memory space contains data RAM and special function registers (SFRs).

The data memory consists of 10 banks. One bank unit is 256 nibbles. BANK 0 is allocated as a SFR area, BANK 1 as the LCD display register, and BANK 2 and the following BANKS are data RAM.

Figure 2-7 shows the configuration of the data memory space.

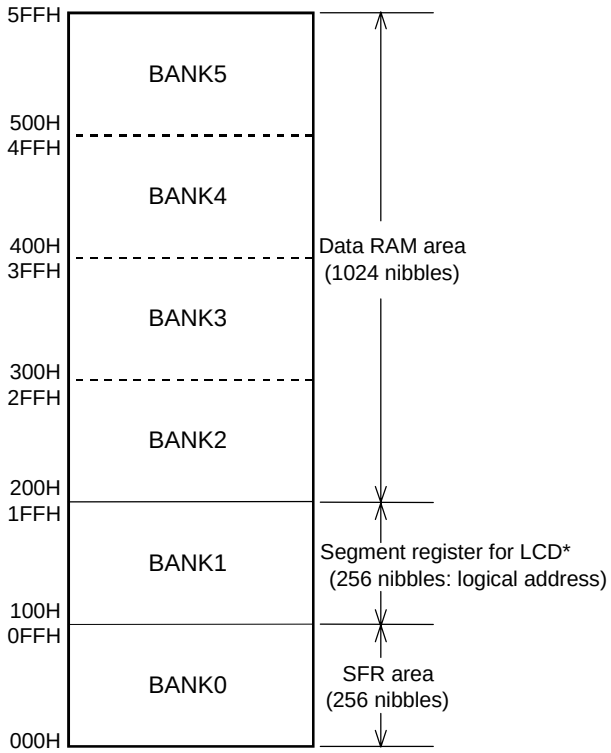


Figure 2-7 Data Memory Space Configuration

Note:

*: The segment register for LCD is allocated in the address range from 100H to 1FFH.
The user can freely set the segment register's address and bit that correspond to each segment pin. Even when L32 to L35, which have a port option, are used as output ports, it is possible to freely set their corresponding address and bit.

Chapter 3

CPU Control Functions

3. CPU Control Functions

3.1 Overview

Operating states, including system reset, are classified as follows.

- Normal operation mode
- System reset mode
- Halt mode

Figure 3-1 shows the CPU operating state transition diagram.

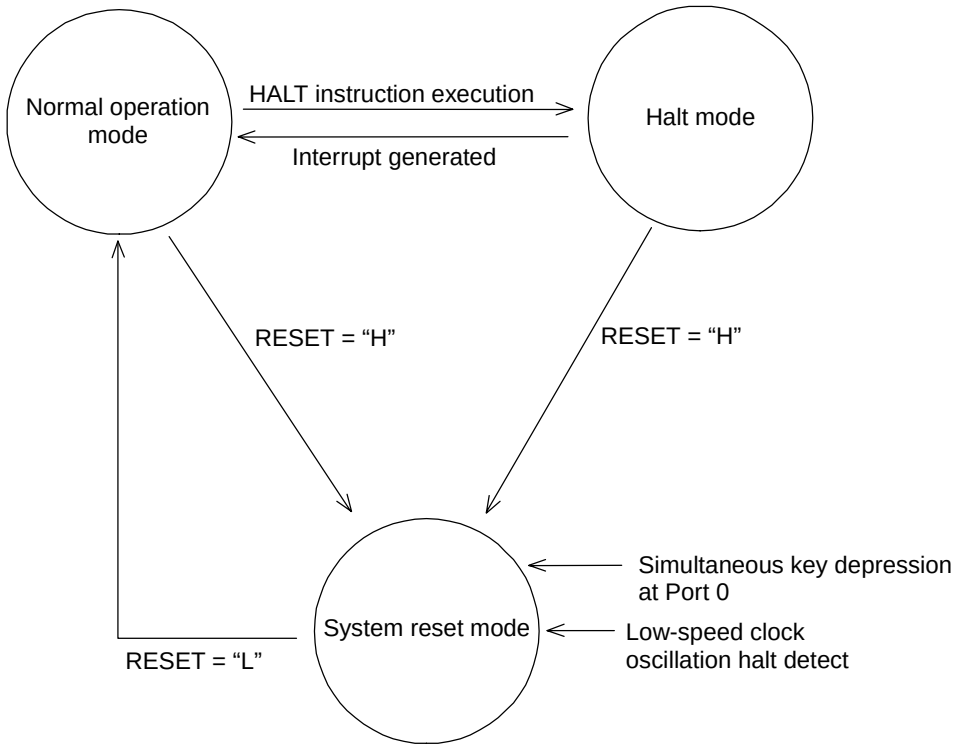


Figure 3-1 Operating State Transition Diagram

The normal operation mode is the state in which the CPU executes instructions sequentially.

The system reset mode begins when a reset input causes the CPU to begin system reset processing where registers and pins are initialized. The CPU remains in this state until instruction execution begins. After system reset processing, instruction execution begins from address 0000H.

The halt mode is the state in which the CPU is halted (instruction execution suspended) but internal peripheral functions continue to operate. During the halt mode, the PC is not incremented. Even upon entering the halt mode, port and peripheral functions will not change. Transfer to the halt mode is accomplished by executing a "HALT" instruction.

3.2 System Reset Mode (RST)

3.2.1 Transfer to and State of System Reset Mode

The ML63611 transfers to the system reset mode due to the following three causes:

(1) When the RESET pin is taken to the "H" level
When recognizing the reset signal, the signal input to the RESET signal is being sampled at 2 kHz. Therefore, the width of the "H" level pulse applied to the RESET pin should be 1 ms or more.
(2) Detection of stoppage of the low frequency clock oscillations (mask option selection)
(3) Simultaneous key depression at the input port 0 (P0.0 to P0.3)
The system reset mode will be entered when there is a simultaneous key depression of a maximum of 4 bits (pins) of Port 0. However, since this sampling is being made at 1 Hz, continue to press for more than about 2 to 3 seconds.
The number of bits pressed simultaneously can be selected by mask option to be 2 bits (P0.0, P0.1), 3 bits (P0.0, P0.1, and P0.2), or 4 bits (P0.0, P0.1, P0.2, and P0.3). For details of the mask option settings, refer to Section 1.3, "Mask Options" and the "MOGTOOL Mask Option Generator User's Manual".

The following operations are performed in the system reset mode.

(1) CPU is initialized.
(2) Bias reference voltage supply (V/R1, V/R2, V/R3) is energized.
(3) All LCD driver outputs are turned OFF and the outputs change to the V_{SS} level.
(4) All special function registers (SFRs) are initialized. However, data RAM and the segment register for LCD are not initialized.

After system reset processing, instruction execution begins from address 0000H.

Figure 3-2 shows the system reset generator circuit and Figure 3-3 shows the signals when a system reset is generated.

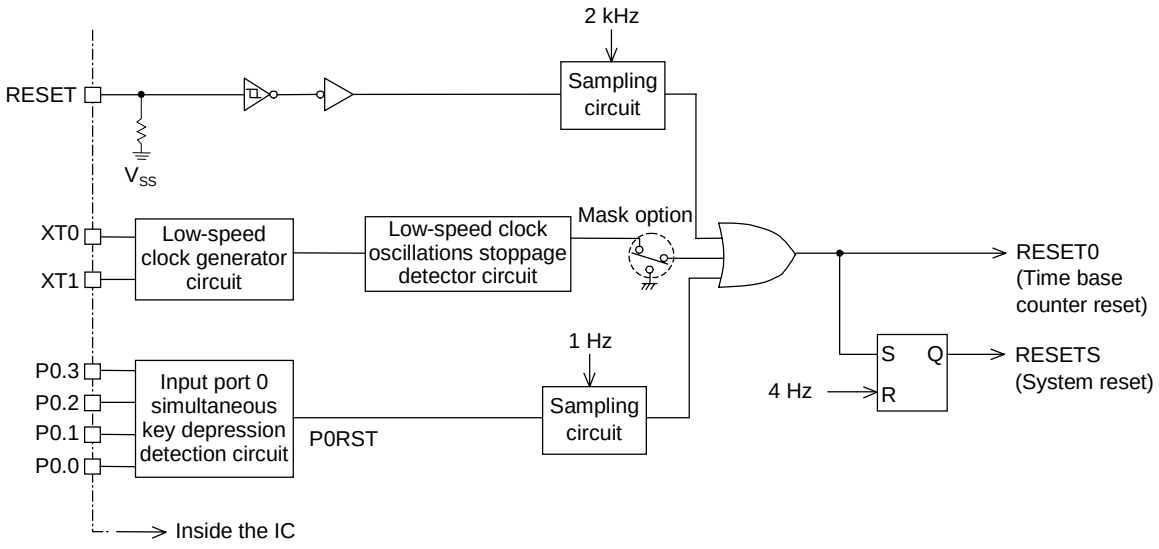


Figure 3-2 System Reset Generator Circuit

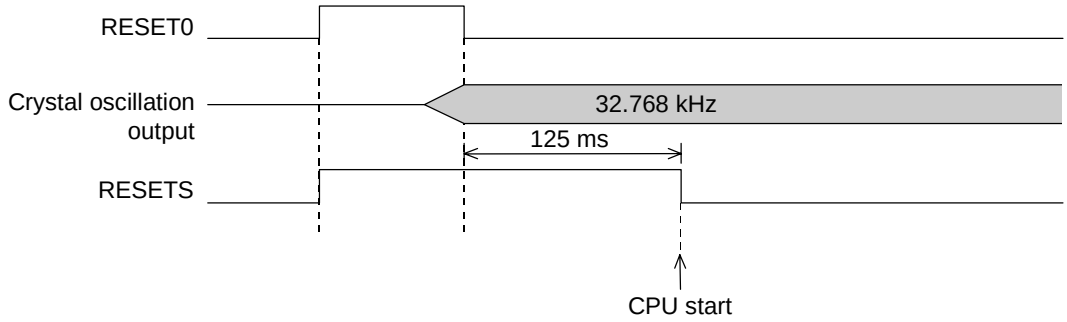
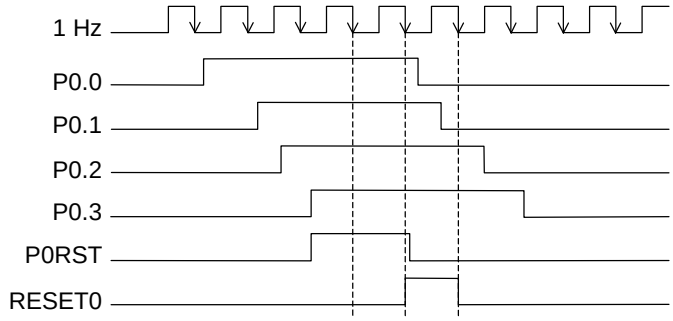
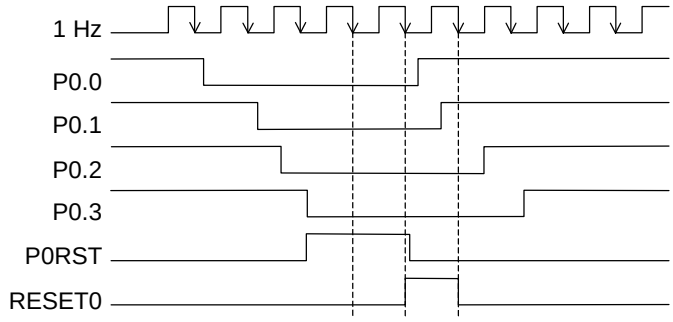


Figure 3-3 Signals When System Reset is Generated

Figure 3-4 shows the timing of transferring to the system reset mode when there is simultaneous key depression at the input Port 0. When all four bits are pressed simultaneously, all of P0.0 to P0.3 go to the "H" (or the "L") level, and the transfer to the system reset mode is made at the second falling edge of the 1 Hz signal.



(a) When P0PUD = 0
(in the case of the 4-bit simultaneous key depression detection setting)



(b) When P0PUD = 1
(in the case of the 4-bit simultaneous key depression detection setting)

Figure 3-4 Timing of Transferring to the System Reset Mode When There Is Simultaneous Key Depression at the Input Port

Notes:

- A system reset is given priority over all other executions, and the executions up to that point in time will be terminated. Therefore, the contents of the RAM and the display registers that are not initialized during the system reset are not guaranteed.
- When transferring to the system reset mode if there is simultaneous key depression at the input port, keep the bits pressed for more than about 2 to 3 seconds.
- When transferring to the system reset mode by taking the RESET pin to the "H" level, make sure that the width of the RESET pulse is 1 ms or more.

3.3 Halt Mode

3.3.1 Transfer to and State of Halt Mode

Transfer to the halt mode is performed by the software when a HALT instruction is executed.

When a HALT instruction is executed, the CPU enters the HALT mode at the S2 state of the HALT instruction.

Oscillation and time base counter operation continue while in the halt mode.

If an interrupt request occurs at the same time as execution of a HALT instruction, interrupt processing has priority and the HALT instruction will not be executed. After the HALT instruction performs the equivalent operation of a NOP instruction, the interrupt routine is entered. When an RTI instruction is used to complete the interrupt routine, the main routine is resumed beginning from the instruction immediately following the HALT instruction.

Figure 3-5 shows the timing when a HALT instruction and interrupt request occur simultaneously.

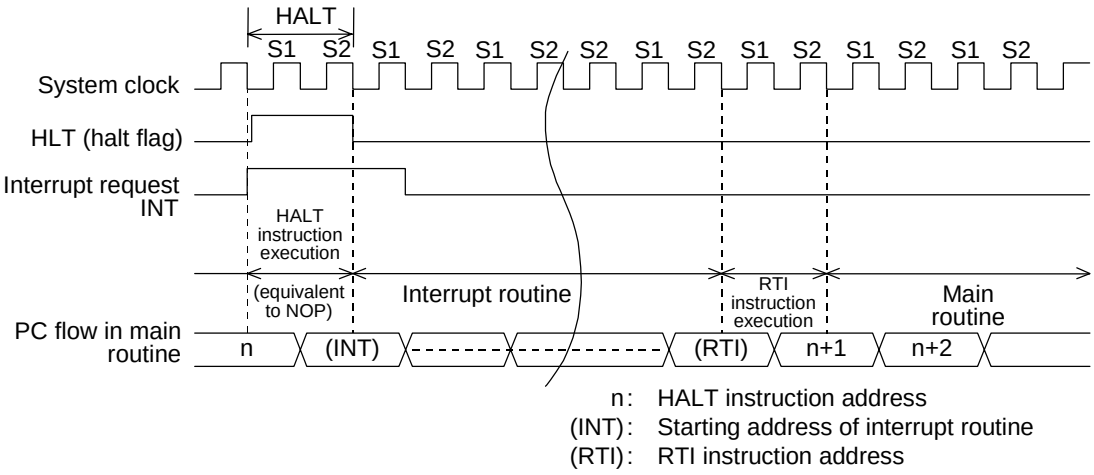


Figure 3-5 Timing of Simultaneous HALT Instruction and Interrupt Request

Note:

While an interrupt request is generated, execution of a HALT instruction will not transfer operation to the halt mode.

3.3.2 Halt Mode Release

The following two methods are available to release the halt mode.

- Release by interrupt generation (transfer to normal operation mode)
- Release by RESET pin (transfer to system reset mode)

3.3.2.1 Release of Halt Mode by Interrupt

If the halt mode is to be released by an interrupt, the enable flag of the interrupt used for release must be set to "1" prior to entering the halt mode. When the halt mode is released by an interrupt, operation transfers to the normal operation mode.

Figure 3-6 shows the timing of transferring to the halt mode by execution of a HALT instruction and of releasing the halt mode by an interrupt.

When the halt mode is released by an interrupt request, the first instruction immediately following the HALT instruction is executed and then the interrupt routine is entered. When an RTI instruction is used to complete the interrupt routine, the main routine is resumed beginning from the second instruction after the HALT instruction.

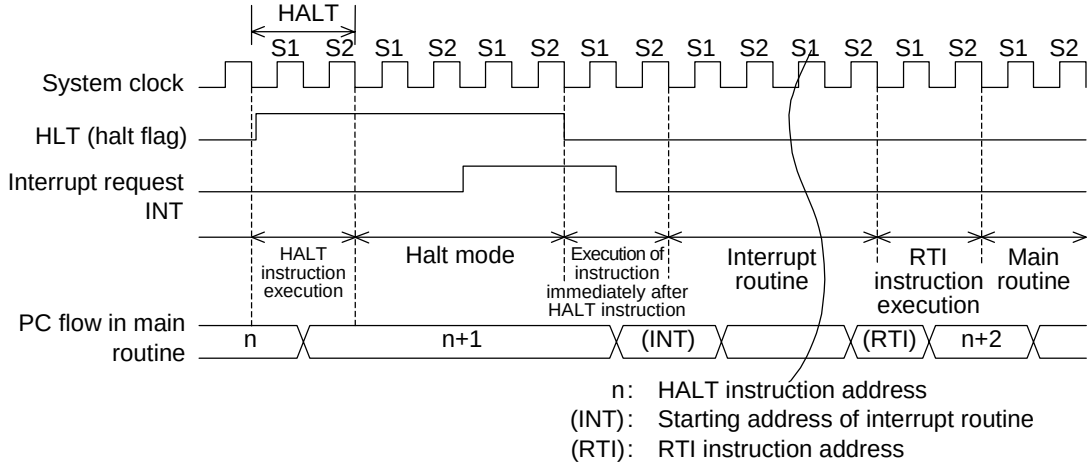


Figure 3-6 Timing of Transfer to Halt Mode and Release of Halt Mode by Interrupt

Note:

If the halt mode is to be released, set individual interrupt enable flags to "1". If an individual interrupt enable flag is "0", the corresponding interrupt request signal cannot reset the HLT flag, regardless of whether the master interrupt enable flag (MIE) is "0" or "1".

3.3.2.2 Release of Halt Mode by RESET Pin

If the RESET pin is held at a "H" level for 1 ms or more, the CPU is released from the halt mode and transfers to the system reset mode. The CPU also transfers to the system reset mode when there is a Port 0 simultaneous key depression (for 2 to 3 seconds) or the low-speed clock oscillation is stopped (selected by mask option).

3.3.3 Melody Data Interrupt and Halt Mode Release

The halt mode is not released by a melody data interrupt.

The melody data interrupt is different from a conventional interrupt in that the melody data interrupt is a hardware processing interrupt used for transfer of melody data to the melody circuit. It is not dependent on the program.

When this interrupt is generated, the instruction immediately after the HALT instruction is executed, then the melody data is transferred to the melody circuit, and the HALT instruction is executed again. This sequence is indicated in Figure 3-7.

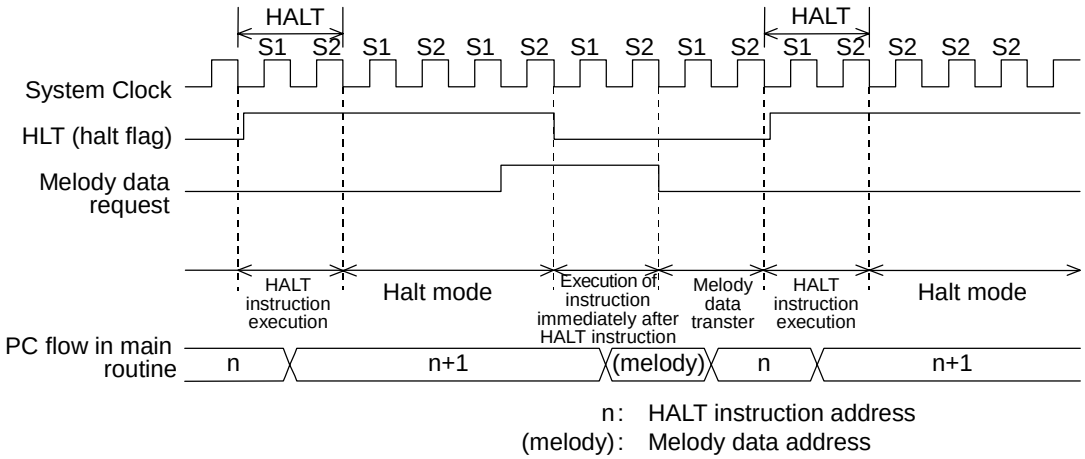


Figure 3-7 Melody Data Request Interrupt Operation

3.3.4 Note Concerning HALT Instruction

As described above, the instruction immediately after the HALT instruction may be executed any number of times. For this reason, always place an NOP instruction immediately after the HALT instruction.

(Example)

•
•
•
HALT
NOP
•
•
•

Chapter 4

Interrupt (INT)

4. Interrupt (INT)

4.1 Overview

The ML63611 supports 18 interrupt factors: 4 external interrupts and 14 internal interrupts.

With the exception of the watchdog timer interrupt, interrupt enable/disable is controlled by the master interrupt enable flag (MIE) and the individual interrupt enable registers (IE0 to IE4). Watchdog timer interrupt is a non-maskable interrupt.

When interrupt conditions are met, the interrupt routine is executed from the interrupt start address.

Table 4-1 indicates a list of interrupt factors, and Figure 4-1 shows the interrupt control equivalent circuit.

Table 4-1 Interrupt Factors

Priority	Interrupt factor	Symbol	Interrupt start address
1	Watchdog timer interrupt	WDTINT	0010H
2	Melody end interrupt	MDINT	0012H
3	External interrupt 0 (PB 4-bit OR input)	XI0INT	0014H
4	External interrupt 1 (PC 4-bit OR input)	XI1INT	0016H
5	External interrupt 2 (PE.3)	XI2INT	0018H
6	A/D interrupt	ADINT	001AH
7	External interrupt 5 (P0 4-bit OR input)	XI5INT	001EH
8	Timer 0 interrupt	TM0INT	0020H
9	Timer 1 interrupt	TM1INT	0022H
10	Timer 2 interrupt	TM2INT	0024H
11	Timer 3 interrupt	TM3INT	0026H
12	Serial port receive interrupt	SRINT	0028H
13	Serial port transmit interrupt	STINT	002AH
14	T10 Hz interrupt	T10HzINT	002EH
15	32 Hz interrupt	32HzINT	0030H
16	16 Hz interrupt	16HzINT	0032H
17	4 Hz interrupt	4HzINT	0034H
18	2 Hz interrupt	2HzINT	0036H

If multiple interrupts are detected simultaneously, the lowest interrupt start address is given priority.

For details on interrupt operation, refer to Chapter 6 (Time Base Counter), Chapter 7 (Timers), Chapter 8 (100 Hz Timer Counter), Chapter 9 (Watchdog Timer), Chapter 10 (Ports), Chapter 11 (Melody Driver), Chapter 12 (Serial Port), and Chapter 16 (A/D Converter).

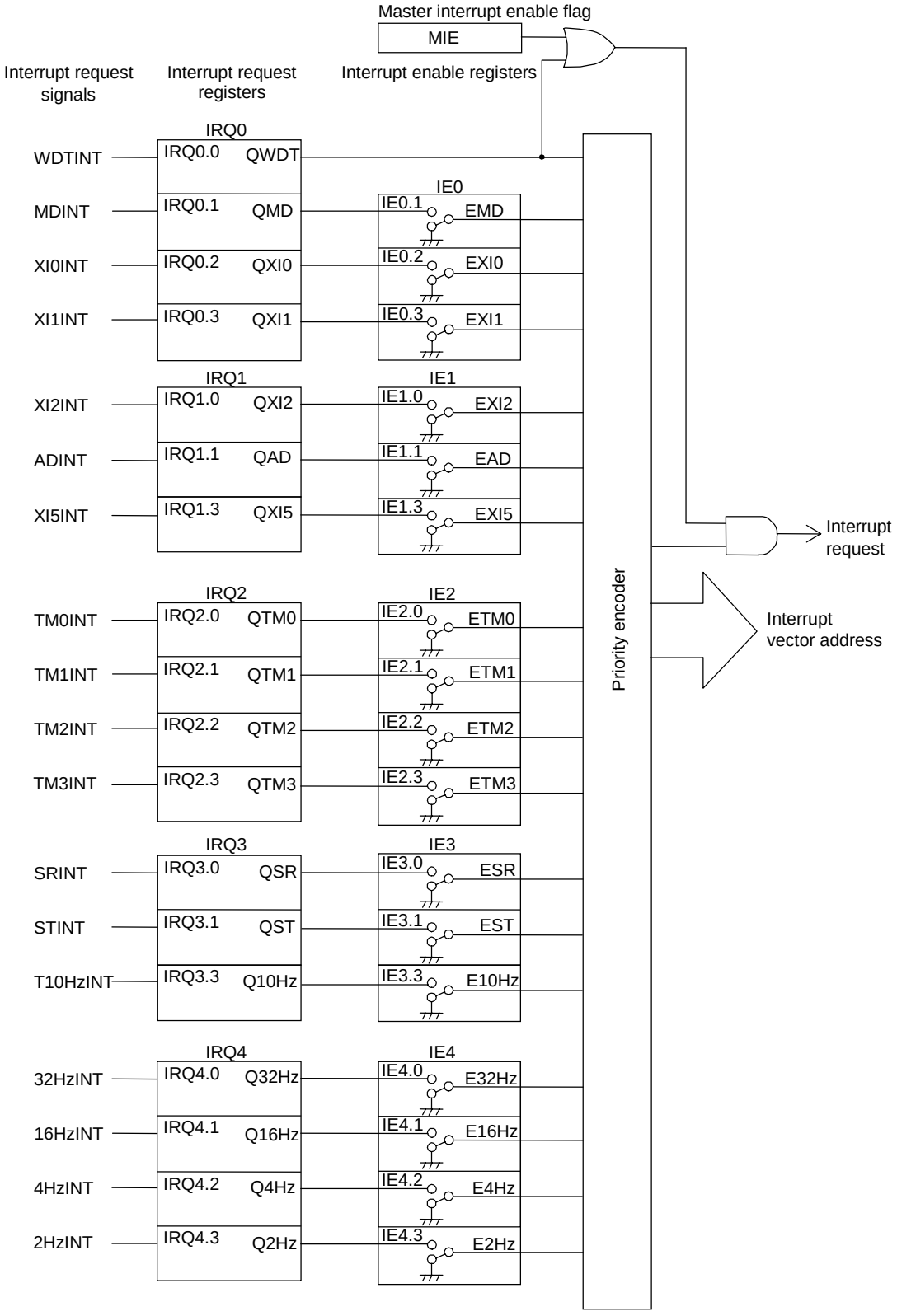


Figure 4-1 Interrupt Control Equivalent Circuit

4.2 Interrupt Registers

The following three types of registers are used to control interrupts.

(1) Master interrupt enable register (MIEF)
(2) Interrupt enable registers (IE0 to IE4)
(3) Interrupt request registers (IRQ0 to IRQ4)

These registers are described below.

(1) Master interrupt enable register (MIEF)

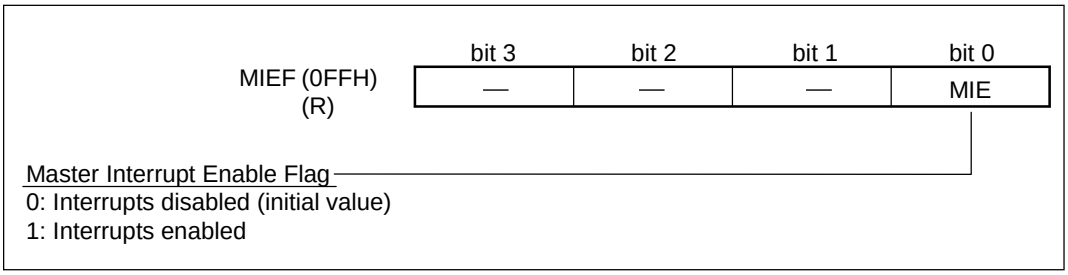
MIEF is a 4-bit register in which bit 0 is the master interrupt enable flag (MIE).

MIE (bit 0 of MIEF) is a flag that disables or enables all interrupts except for the watchdog timer interrupt. If MIE is "0", all interrupts are disabled. If MIE is "1", all interrupts are enabled (with the exception of the watchdog timer).

When any interrupt is received, MIE is cleared to "0". MIE is set to "1" by execution of a return from interrupt instruction (RTI instruction).

If multi-level interrupt processing is to be performed, execute an RTI instruction (MIE←"1") during the interrupt processing routines.

At system reset, MIE is initialized to "0". MIEF only supports data reference (R) of data memory through addressing instructions.



Note:

When setting MIE, use "EI" instructions (MIE←"1") and "DI" instructions (MIE←"0").

(2) Interrupt enable registers (IE0 to IE4)

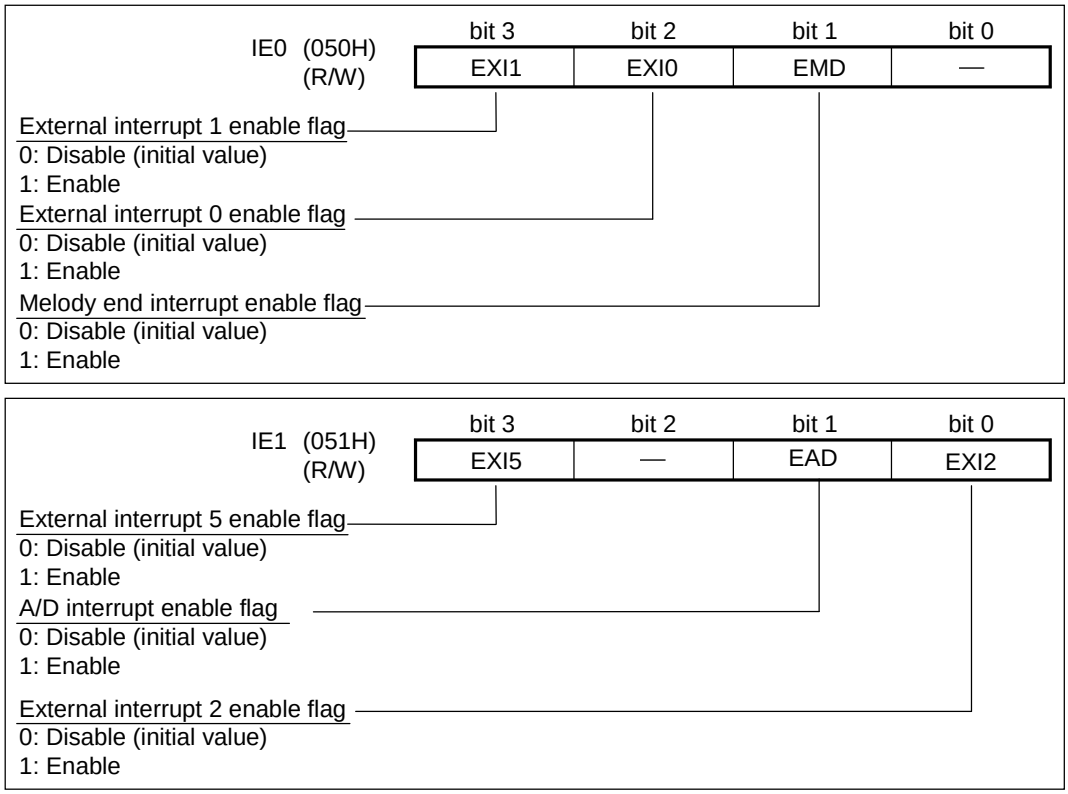
IE0, IE1, IE2, IE3, and IE4 are registers that consist of 4 bits each.

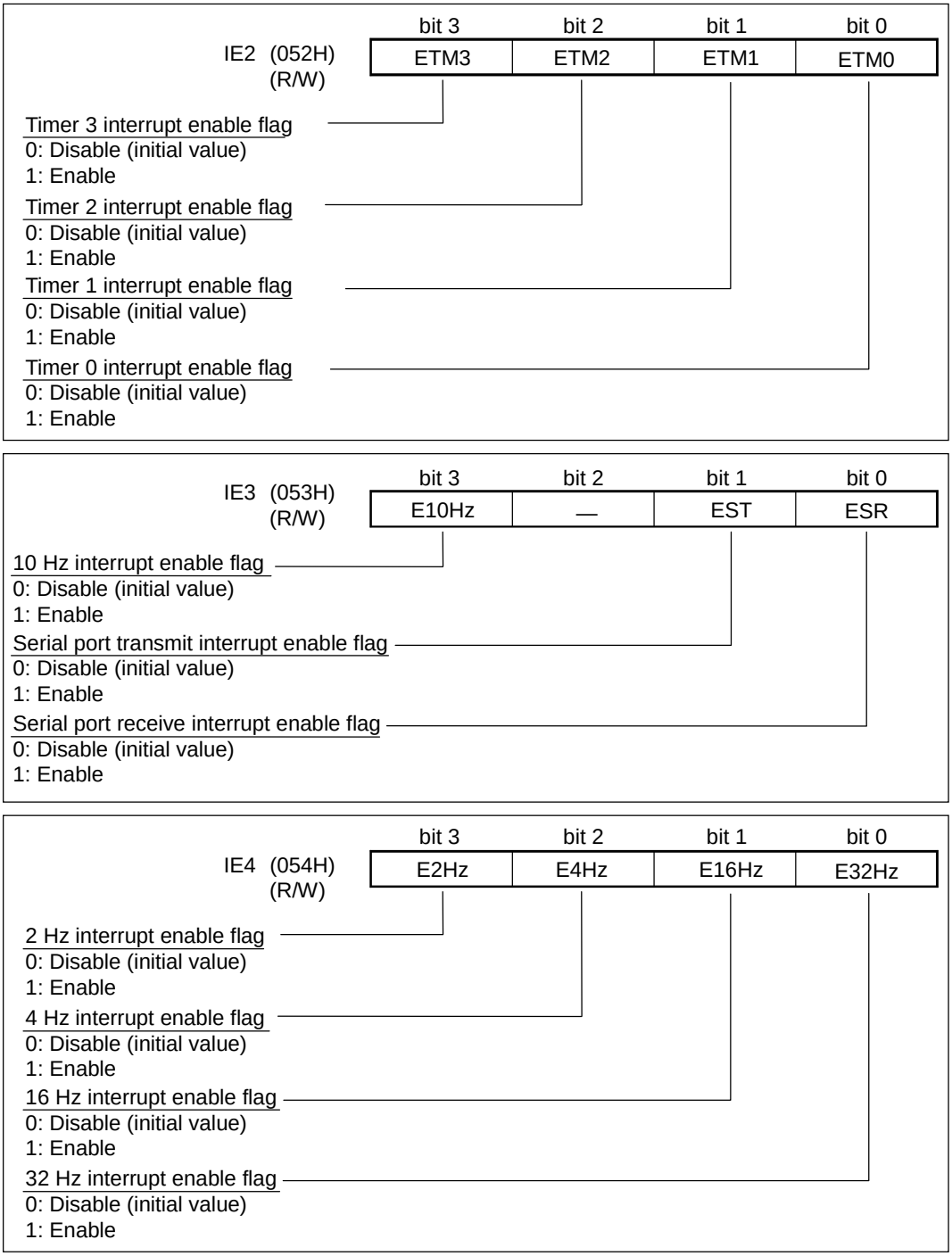
A logical AND of the corresponding bits of an interrupt enable register (IE0 to IE4) and an interrupt request register (IRQ0 to IRQ4) determines whether or not each interrupt request is issued to the CPU. The watchdog timer interrupt is non-maskable, and is therefore not dependent upon the interrupt enable registers (IE0 to IE4) and the master interrupt enable register (MIEF).

If multiple interrupts request the CPU at the same time, the interrupts are accepted in order of highest priority and low priority interrupts are placed on hold (see Table 4-1 for the order of priority).

When an interrupt is received, the master interrupt enable flag (MIE) is cleared to "0". The corresponding bits in the interrupt enable registers (IE0 to IE4) do not change.

At system reset, each bit of IE0 through IE4 is initialized to "0".




IE2 (052H)
(R/W)
bit 3
bit 2
bit 1
bit 0
ETM3
ETM2
ETM1
ETM0
Timer 3 interrupt enable flag
0: Disable (initial value)
1: Enable
Timer 2 interrupt enable flag
0: Disable (initial value)
1: Enable
Timer 1 interrupt enable flag
0: Disable (initial value)
1: Enable
Timer 0 interrupt enable flag
0: Disable (initial value)
1: Enable
IE3 (053H)
(R/W)
bit 3
bit 2
bit 1
bit 0
E10Hz
—
EST
ESR
10 Hz interrupt enable flag
0: Disable (initial value)
1: Enable
Serial port transmit interrupt enable flag
0: Disable (initial value)
1: Enable
Serial port receive interrupt enable flag
0: Disable (initial value)
1: Enable
IE4 (054H)
(R/W)
bit 3
bit 2
bit 1
bit 0
E2Hz
E4Hz
E16Hz
E32Hz
2 Hz interrupt enable flag
0: Disable (initial value)
1: Enable
4 Hz interrupt enable flag
0: Disable (initial value)
1: Enable
16 Hz interrupt enable flag
0: Disable (initial value)
1: Enable
32 Hz interrupt enable flag
0: Disable (initial value)
1: Enable

(3) Interrupt request registers (IRQ0 to IRQ4)

IRQ0, IRQ1, IRQ2, IRQ3 and IRQ4 are registers that consist of 4 bits each.

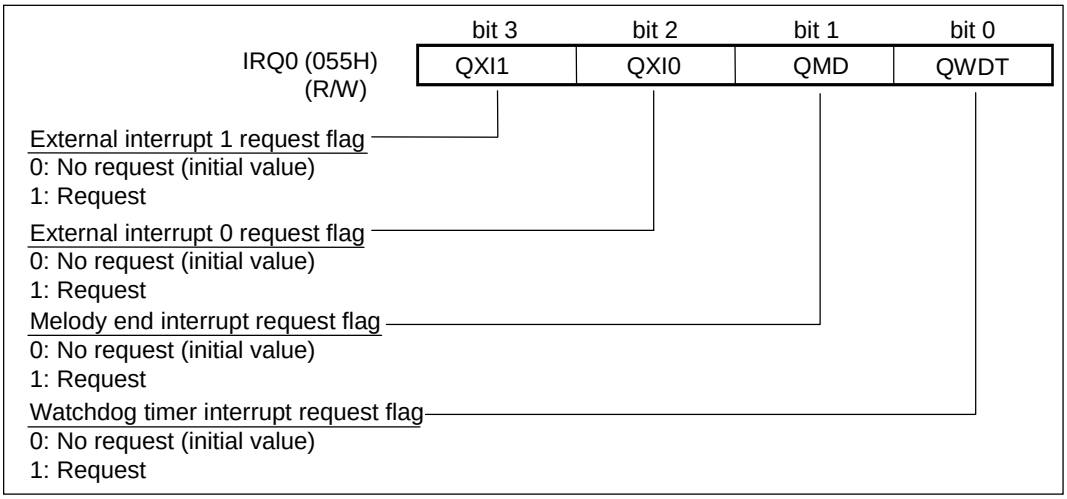
When an interrupt request is generated, the corresponding bit of the interrupt request register is set to "1" in the first half of the S1 state of the next instruction. So that the CPU can receive interrupt requests, set the master interrupt enable flag (MIE) to "1" and set the appropriate flag of the corresponding interrupt enable register (IE0 to IE4) to "1".

The watchdog timer interrupt is non-maskable and does not depend upon the interrupt enable register or the master interrupt enable register (MIEF).

Setting the appropriate bits of an interrupt request register to "1" allows software interrupts to be generated.

When an interrupt request is received, the corresponding bits of IRQ0 to IRQ4 are cleared to "0".

At system reset, each bit of IRQ0 through IRQ4 is initialized to "0".



bit 3: QXI1 (reQuest eXternal Interrupt 1)

The external interrupt 1 request flag.
The external interrupt 1 is assigned as the secondary function of each bit of port C (PC.0 to PC.3).
External interrupt 1 requests are generated by a 4-bit ORed input.

bit 2: QXI0 (reQuest eXternal Interrupt 0)

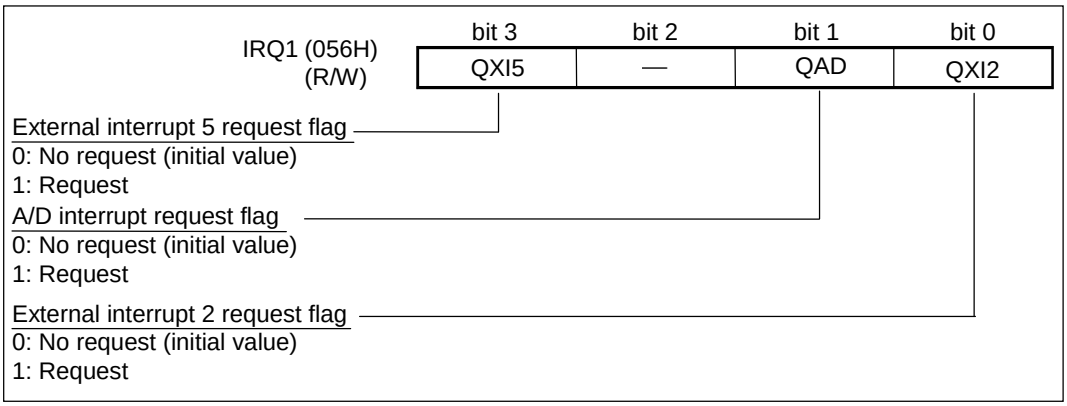
The external interrupt 0 request flag.
The external interrupt 0 is assigned as the secondary function of each bit of port B (PB.0 to PB.3).
External interrupt 0 requests are generated by a 4-bit ORed input.

bit 1: QMD (reQuest Melody Driver)

Melody end interrupt request flag.
Melody end interrupts are generated when the melody driver outputs the end note data (END bit = "1").

bit 0: QWDT (reQuest WatchDog Timer)

Watchdog timer interrupt request flag.
When the watchdog timer is started and then overflow occurs, an interrupt is requested. The watchdog timer interrupt is non-maskable and does not depend upon the interrupt enable registers or the master interrupt enable register (MIE).



bit 3: QXI5 (reQuest eXternal Interrupt 5)

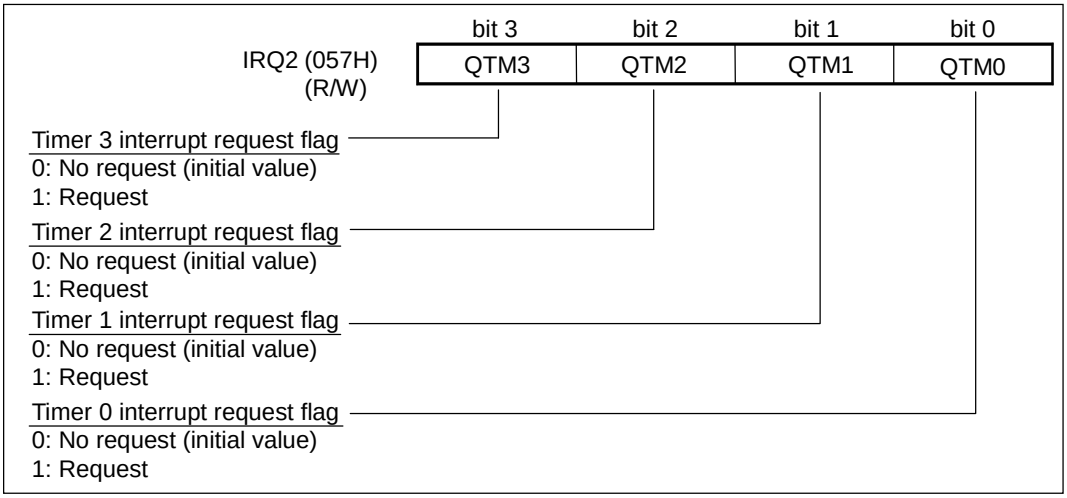
External interrupt 5 request flag.
The external interrupt 5 is assigned as a secondary function to each bit (P0.0 to 0.3) of port 0.
An external interrupt request is generated through the 4-bit ORed input.

bit 1: QAD (reQuest AD)

A/D interrupt request flag.
If the selected counter A or B overflows, an interrupt request is generated.

bit 0: QXI2 (reQuest eXternal Interrupt 2)

External interrupt 2 request flag.
The external interrupt 2 is assigned as a secondary function of port E.3 (PE.3). Generation of the external interrupt 2 is triggered by the falling edge of the 128 Hz or 4 kHz output of the time base counter.



bit 3: QTM3 (reQuest TiMer 3)

Timer 3 interrupt request flag.
A timer 3 interrupt request is generated whenever timer 3 overflows.

bit 2: QTM2 (reQuest TiMer 2)

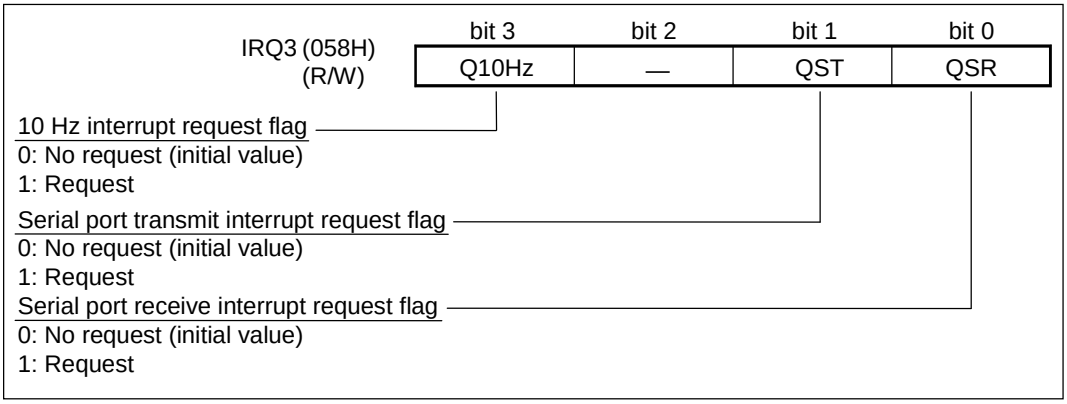
Timer 2 interrupt request flag.
A timer 2 interrupt request is generated whenever timer 2 overflows.

bit 1: QTM1 (reQuest TiMer 1)

Timer 1 interrupt request flag.
A timer 1 interrupt request is generated whenever timer 1 overflows.

bit 0: QTM0 (reQuest TiMer 0)

Timer 0 interrupt request flag.
A timer 0 interrupt request is generated whenever timer 0 overflows.



bit 3: Q10Hz (reQuest 10 Hz)

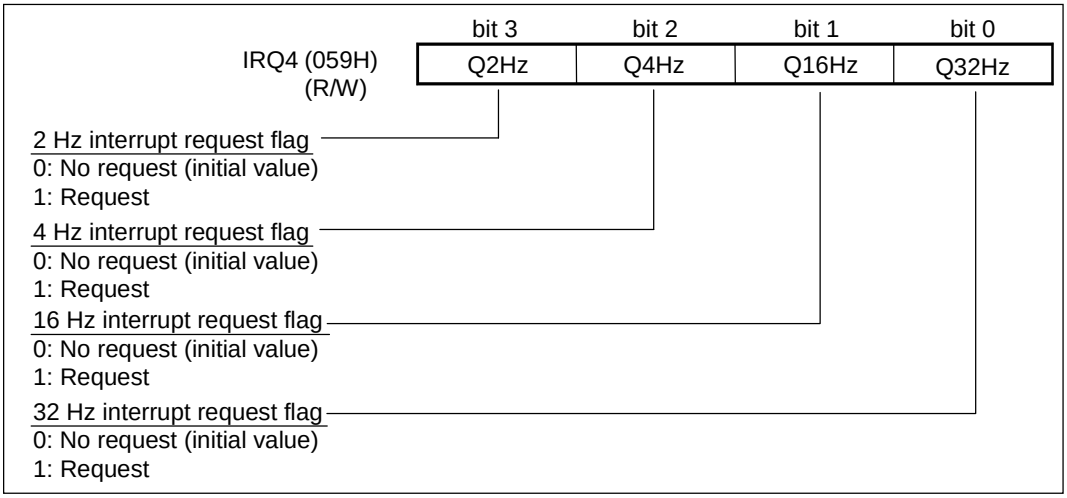
10 Hz interrupt request flag.
A 10 Hz interrupt request is generated whenever the 10 Hz carry generated by the 100 Hz timer counter is output.

bit 1: QST

Serial port transmit interrupt request flag.
A serial port transmit interrupt request is generated when a serial port transmit operation is completed.

bit 0: QSR

Serial port receive interrupt request flag.
A serial port receive interrupt is generated when a serial port receive operation is completed.



bit 3: Q2Hz (reQuest 2 Hz)

2 Hz interrupt request flag.
A 2 Hz interrupt request is generated at every falling edge of the 2 Hz output of the time base counter.

bit 2: Q4Hz (reQuest 4 Hz)

4 Hz interrupt request flag.
A 4 Hz interrupt request is generated at every falling edge of the 4 Hz output of the time base counter.

bit1: Q16Hz (reQuest 16 Hz)

16 Hz interrupt request flag.
A 16 Hz interrupt request is generated at every falling edge of the 16 Hz output of the time base counter.

bit 0: Q32Hz (reQuest 32 Hz)

32 Hz interrupt request flag.
A 32 Hz interrupt request is generated at every falling edge of the 32 Hz output of the time base counter.

4.3 Interrupt Sequence

4.3.1 Interrupt Processing

While MIE is "1", operation transfers to interrupt processing when individual interrupt factors are generated.

The watchdog timer interrupt is non-maskable and regardless of the MIE flag status, operation will shift to interrupt processing when the watchdog timer interrupt factor is generated.

The following processes are performed when an interrupt is generated.

(1) MIE and the corresponding interrupt request flag are cleared to "0".
(2) The program counter (PC) is saved on the call stack.
(3) The call stack pointer (SP) is incremented by 1. (SP←SP+1)
(4) The starting address of the interrupt routine is loaded into the program counter (PC).

Interrupt processing is performed in 0 machine cycles.

Figure 4-2 shows the stack contents after an interrupt is generated.

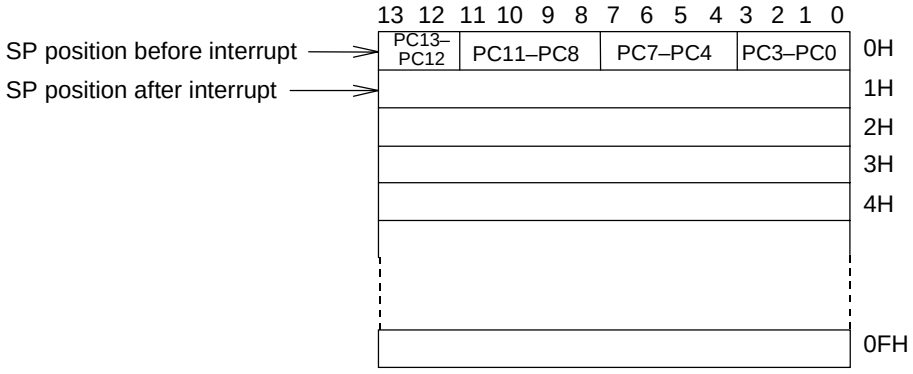


Figure 4-2 Call Stack Contents after Interrupt Generation

4.3.2 Return from an Interrupt Routine

Return from a watchdog timer interrupt routine is performed with an "RTNMI" instruction.

Return from all other interrupt routines is performed with an "RTI" instruction.

Execution of "RTI" and "RTNMI" instructions both require 1 machine cycle.

When returning from an interrupt routine, the CPU performs the following processes.

(1) The call stack pointer (SP) is decremented by 1. (SP←SP–1)
(2) MIE is set to "1" (when an "RTNMI" instruction is used, MIE is restored to its state prior to the interrupt).
(3) 1 is added to the call stack contents and that value is loaded into the program counter (PC)

Note:

- While the MIE flag is "0" (interrupt disabled state), if a watchdog timer interrupt is processed and an "RTI" instruction is executed, the MIE flag will be set to "1" and interrupts enabled.
- Use "RTNMI" instructions to return from watchdog timer interrupts only. Use "RTI" instructions for normal interrupts.

4.3.3 Interrupt Hold Instructions

Interrupt requests are not received after execution of interrupt hold instruction. They are received after execution of an instruction other than interrupt hold instructions.

The interrupt hold instructions follow.

- ROM table reference instructions
- Stack operation instructions
- Jump instructions
- Conditional branch instructions
- Call/return instructions
- "EI" (set MIE flag) instructions, "DI" (clear MIE flag) instructions and "MSA cadr15" (start melody output) instructions within control instructions

Note:

If interrupt hold instructions are to be used consecutively, consider that a generated interrupt will be put on hold for a considerable amount of time before the interrupt routine begins.

Chapter 5

Clock Generator Circuit (OSC)

5. Clock Generator Circuit (OSC)

5.1 Overview

The ML63611 has built in it a low-speed clock generator circuit and a high-speed clock generator circuit, and the system clock that becomes the basic operating clock of the CPU section is controlled using the frequency control register (FCON).

The low-speed clock generator circuit is configured using a crystal oscillator circuit. A crystal unit (32.768 kHz) is connected between the pins XT0 and XT1, and a capacitor (5 to 25 pF) is connected between the pins XT0 and V_{SS}. The output (TBCCLK) of the low-speed clock generator circuit becomes the basic operating clock for the time base counter etc., and is used as the system clock.

In the high-speed clock generator circuit, it is possible to select by software either the ceramic oscillation mode or the RC oscillation mode. In the OPTION A and OPTION B, however, only the RC oscillation mode is available. For the ceramic oscillator mode, connect a ceramic unit (700 kHz max.) between the pins OSC0 and OSC1, and connect a capacitor C_{L0} between the pins OSC0 and V_{SS} and a capacitor C_{L1} between the pins OSC1 and V_{SS}. When not using the high-speed clock generator circuit, leave the pins OSC0 and OSC1 open. It is possible to select the output (HSCLK) of the high-speed clock generator circuit as the clock for the timer etc., and will be used as the system clock.

The frequency control register (FCON) is a 4-bit special function register that selects the system clock (TBCCLK or HSCLK) and the starting and stopping of the operation of the high-speed clock generator circuit.
During a system reset, the operation is made only of the low-speed clock generator circuit (the high-speed clock generator circuit will be in the halted state), and TBCCLK will be selected as the system clock.

5.2 Clock Generator Circuit Configuration

Figure 5-1 shows a block diagram of the clock generator circuit.

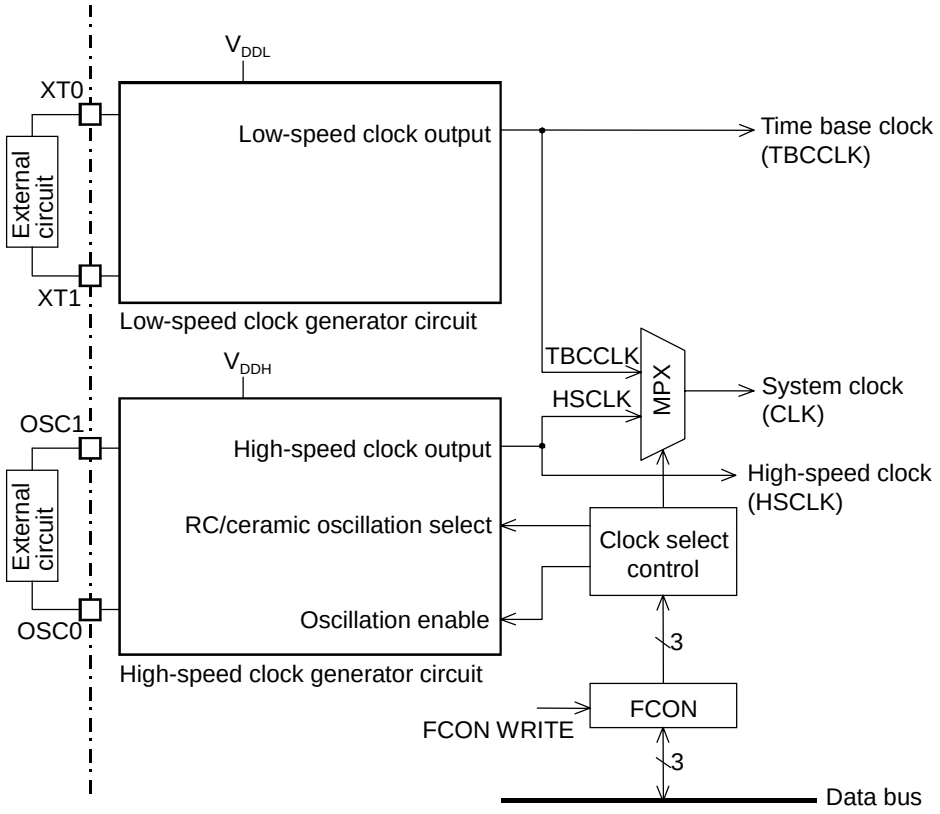


Figure 5-1 Clock Generator Circuit Configuration

5.3 Low-Speed Clock Generator Circuit

The low-speed clock generator circuit is configured using a crystal oscillator circuit. A crystal unit (32.768 kHz) is connected between the pins XT0 and XT1, and a capacitor (5 to 25 pF) is connected between the pins XT0 and V_{SS}. The output (TBCCLK) of the low-speed clock generator circuit becomes the basic operating clock for the time base counter etc., and is used as the system clock.

For the crystal oscillation mode, attach an external crystal unit and a capacitor (C_G) as shown in Figure 5-2.

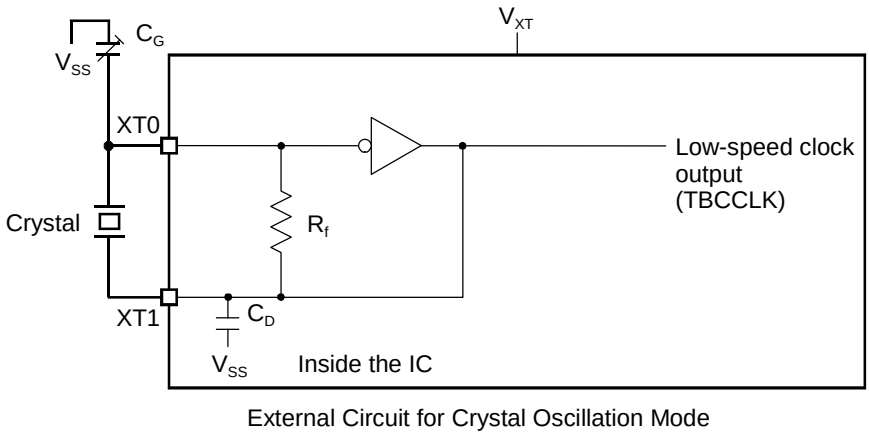


Figure 5-2 External Circuit for Low-Speed Clock Oscillation

Note:

The crystal and the capacitor C_G should be connected in close proximity to the XT0 and XT1 pins.

Table 5-1 shows an example external component to be attached when the low-speed side crystal oscillation mode is selected.

Table 5-1 Example External Component for the Low-Speed Side Crystal Oscillation Mode

C_G	f_{XT}	Model name/Manufacturer
12 pF	32.768 kHz	VT-150, VT-200/SII

5.4 High-Speed Clock Generator Circuit

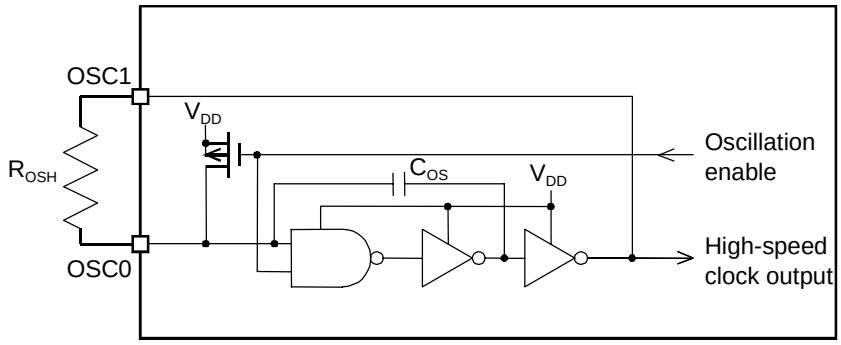
The high-speed clock generator circuit has two modes, the RC oscillation mode and ceramic oscillation mode. In the OPTION A and OPTION B, only the RC oscillation mode is available. Oscillation modes are set by OSCSEL (bit 2 of FCON). The oscillation frequency is a maximum of 200 kHz for the OPTION A and OPTION B, and a maximum of 700 kHz for the OPTION C and OPTION D.

- OSCSEL = "0" : RC oscillation mode
- OSCSEL = "1" : ceramic oscillation mode

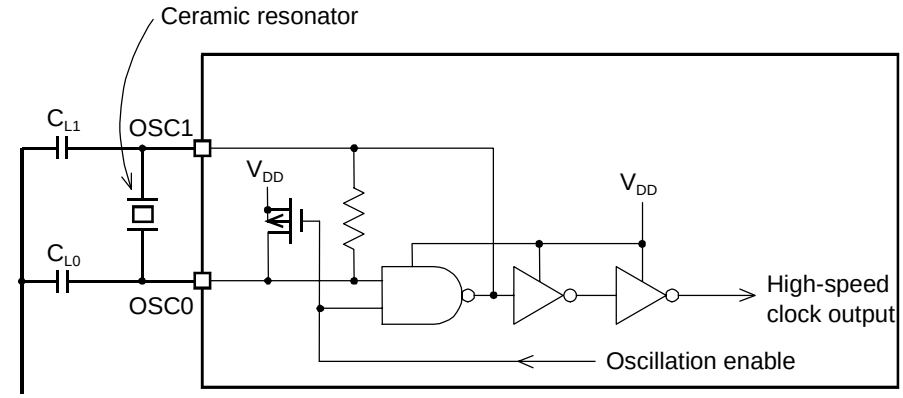
If the high-speed clock is not to be used, leave the OSC0 and OSC1 pins open (unconnected).

For the RC oscillation mode, attach an external resistor, R_{OSH}, as shown in Figure 5-3(a).

For the ceramic oscillation mode, attach an external ceramic unit and capacitors as shown in Figure 5-3(b).



(a) External Circuit for RC Oscillation Mode



(b) External Circuit for Ceramic Oscillation Mode

Figure 5-3 External Circuits for High-Speed Clock Oscillation

Notes:

In the OPTION A and OPTION B, only the RC oscillation mode is available.
The high-speed clock frequency of the OPTION A and OPTION B is a maximum of 200 kHz.
The high-speed clock frequency of the OPTION C and OPTION D is a maximum of 700 kHz.

Table 5-2 lists typical values of oscillation frequency when the high-speed side RC oscillation mode is selected. Table 5-3 lists example external components to be attached when the high-speed side ceramic oscillation mode is selected.

Table 5-2 Typical Oscillation Frequencies for the High-Speed Side RC Oscillation Mode

R_{OSH} (kΩ)	V_{DD} (V)	f_{ROSH}
400	1.3 to 3.6	200 kHz ±30%
100	1.8 to 3.6	700 kHz ±30%

Table 5-3 Example External Components for the High-Speed Side Ceramic Oscillation Mode

C_{L0} (pF)	C_{L1} (pF)	Ceramic unit
330	330	CSB200D (200 kHz)*
220	220	CSB300D (300 kHz)*
150	150	CSB500E (500 kHz)*

* Ceramic unit manufactured by Murata MFG. Co., Ltd.

5.5 System Clock Control

The system clock is the basic operation clock of the CPU.

The clock can be selected as follows with the CPUCLK (bit 0 of FCON) setting.

- CPUCLK = "0" (initial value)
  The output of the low-speed clock generator circuit (TBCCLK) is the system clock.
- CPUCLK = "1"
  The output of the high-speed clock generator circuit (HSCLK) is the system clock.

When HSCLK is selected as the system clock, the high-speed clock must be in the oscillating state (ENOSC = "1"). The low-speed clock generator circuit will continue to oscillate even when the high-speed generator circuit is selected.

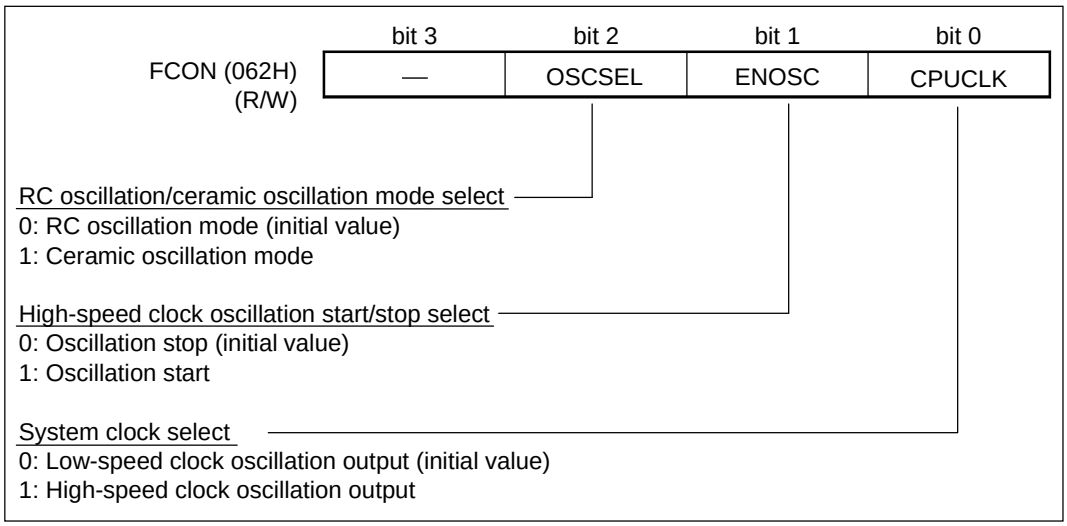
To reduce the total power consumption in applications that use the high-speed clock generator circuit, the following clock controls are generally implemented in software.

- During normal operation, the output of the low-speed clock generator circuit (CPUCLK = "0") should be the system clock.
- Only when high-speed operation is necessary should the high-speed clock oscillate (ENOSC = "1") and output of the high-speed clock generator circuit (CPUCLK = "1") should be selected.

For details of the system clock select timing, refer to section 5.7, "System Clock Select Timing".

5.6 Frequency Control Register (FCON)

FCON is a special function register (SFR) that selects the system clock.



bit 2: OSCSEL

This bit selects the RC oscillation mode or the ceramic oscillation mode of the high-speed clock generator circuit. At system reset, this bit is cleared to "0", selecting the RC oscillation mode.

bit 1: ENOSC

This bit starts and stops oscillation of the high-speed clock generator circuit. At system reset, this bit is cleared to "0", stopping oscillation of the high-speed clock generator circuit.

bit 0: CPUCLK

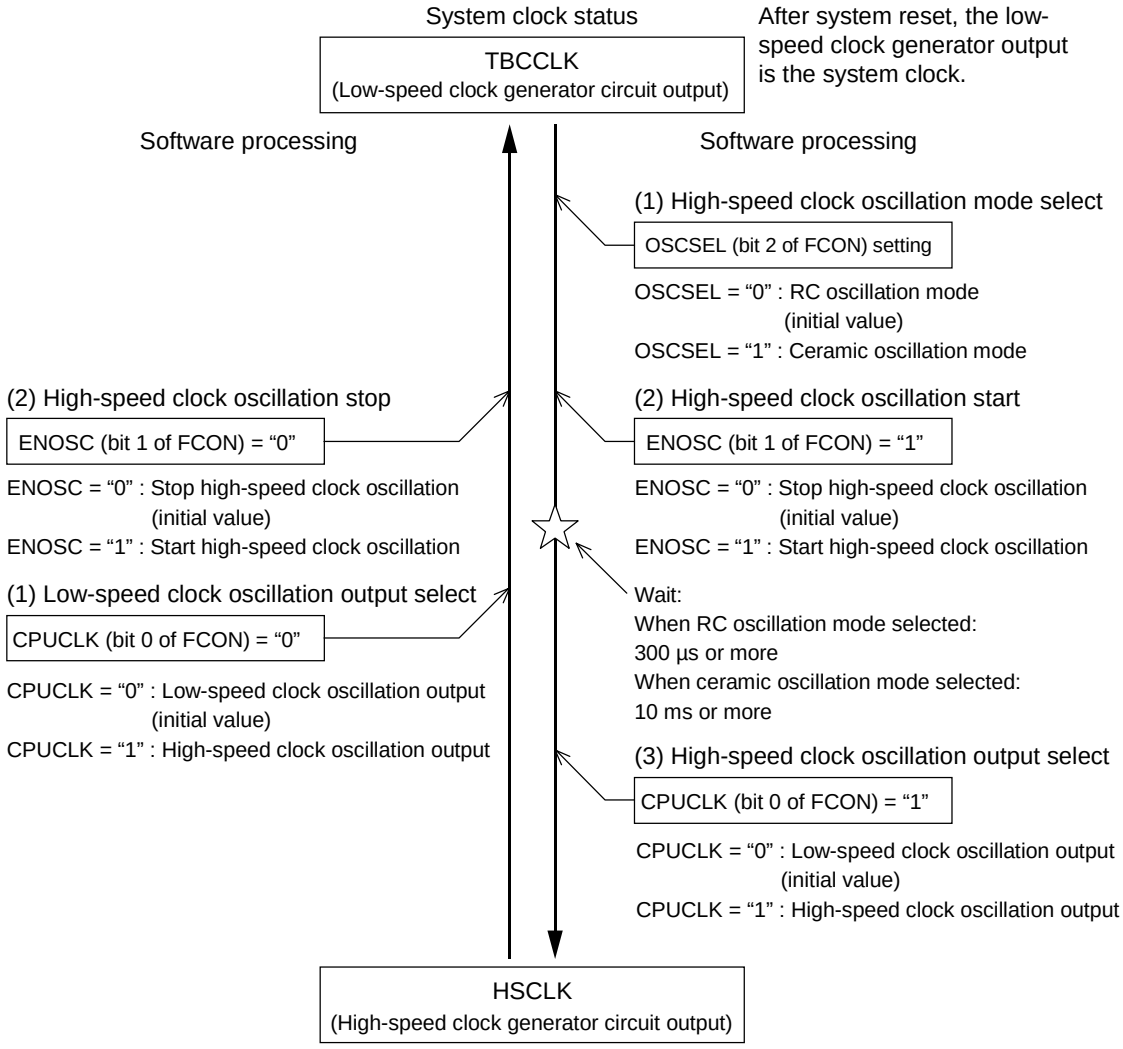
This bit selects the system clock, the basic operation clock of the CPU. At system reset, this bit is cleared to "0", selecting output of the low-speed clock generator circuit (TBCCLK).

5.7 System Clock Select Timing

After system reset, the system clock is TBCCLK.

When high-speed operation is necessary, switch the system clock to HSCLK.

A flowchart of system clock operation is shown below.



When ENOSC (bit 1 of FCON) is set to "1", oscillation starts in the mode selected by OSCSEL. At the same time, the internal logic power supply (V_{CH}) switches from the voltage regulator circuit (V/R1) output level (approx. 1.15 V) to the V_{DD} level. Next, if CPUCLK is set to "1", the system clock switches from crystal oscillation output (TBCCLK) to high-speed clock output (HSCLK).

Figure 5-4 shows the system clock select timing and status of the internal logic power supply (V_{CH}).

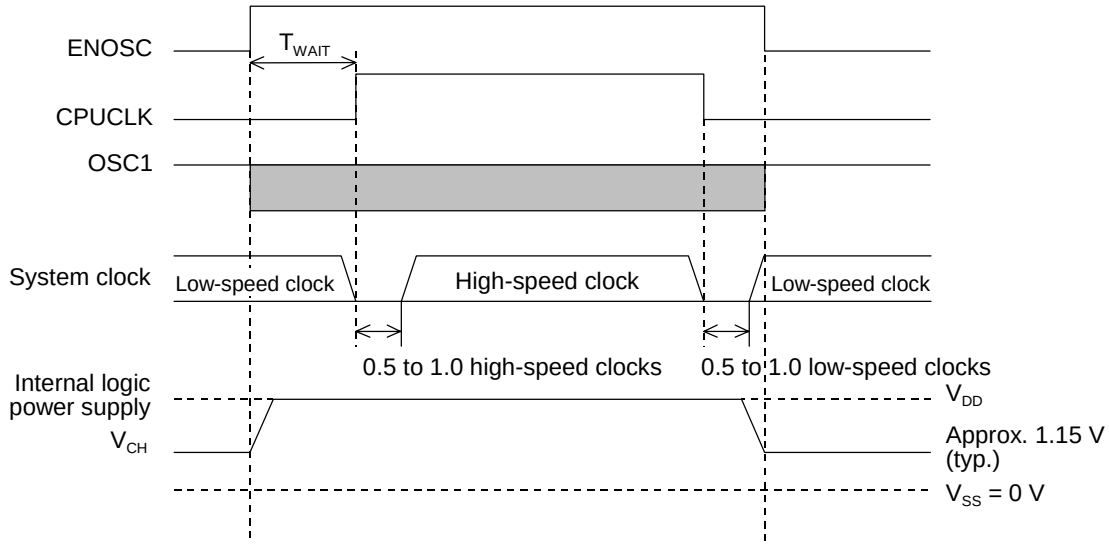


Figure 5-4 System Clock Select Timing

In the ceramic oscillation mode, 10 ms are required from the time when ENOSC is set to "1" until the high-speed clock generator circuit enters the oscillating state. Therefore, in this mode, when switching CPUCLK to a high-speed setting, wait for an interval of at least T_{WAIT} = 10 ms after the rising edge of ENOSC.

In the RC oscillation mode, oscillation begins soon after setting ENOSC to "1". When switching CPUCLK to a high-speed setting, wait for an interval of at least T_{WAIT} = 300 µs after the rising edge of ENOSC.

When switching from the high-speed mode to the low-speed mode, set the CPUCLK bit to "0", and sometime after the next instruction, set the ENOSC bit to "0".

Chapter 6

Time Base Counter (TBC)

6. Time Base Counter (TBC)

6.1 Overview

The time base counter (TBC) is a 15-bit internal counter, which generates the clock supplied to internal peripheral functions.

The TBC clock is a time base clock (TBCCLK).

TBC outputs are used for functions such as time base interrupts and various other circuits.
TBC8–11 and TBC12–15 can be read/reset by software.

The TBC generates an interrupt request at the falling edge of 32 Hz/16 Hz/4 Hz/2 Hz output.

The TBC is initialized to 0000H at system reset.

6.2 Time Base Counter Configuration

The configuration of the time base counter (TBC) is shown in Figure 6-1.

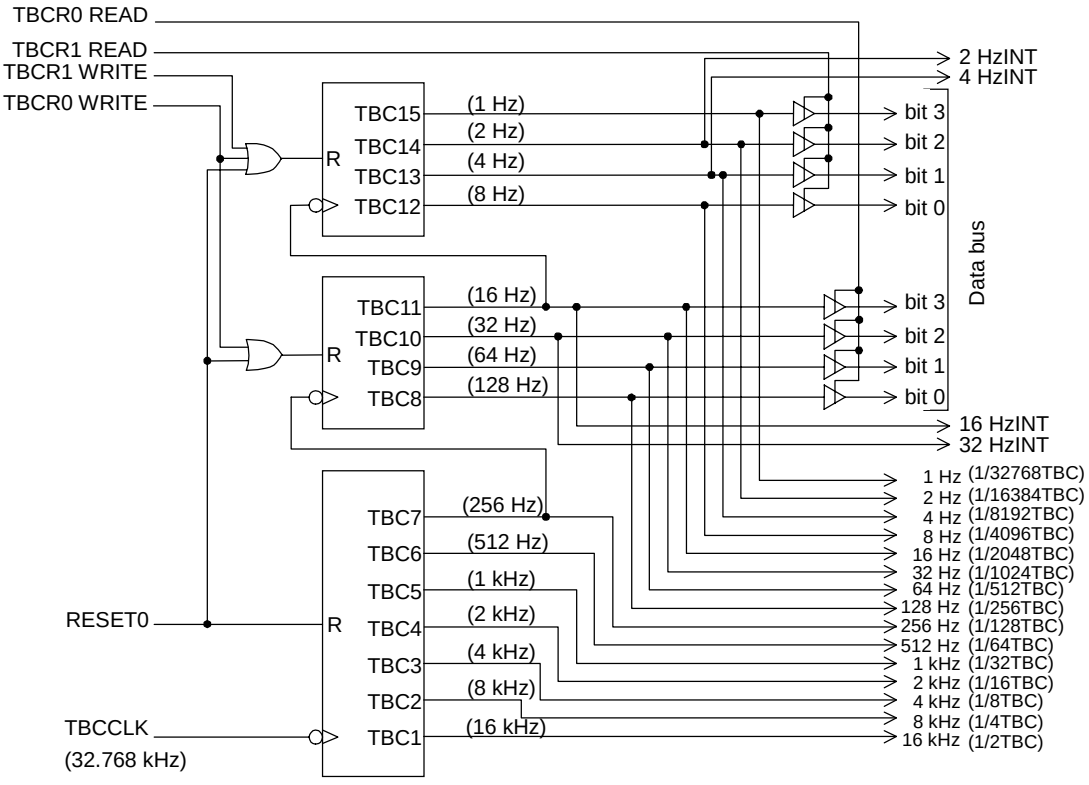


Figure 6-1 Time Base Counter (TBC) Configuration

6.3 Time Base Counter Registers

Time base counter register 0 (TBCR0), time base counter register 1 (TBCR1)

These 4-bit special function registers (SFRs) are used to read the 1 to 8 Hz and 16 to 128 Hz outputs of the time base counter.

A write operation to TBCR0 sets both the 1 to 8 Hz and 16 to 128 Hz outputs to "0", and a write operation to TBCR1 sets the 1 to 8 Hz output to "0".

	bit 3	bit 2	bit 1	bit 0
TBCR0 (060H) (R/W)	16 Hz	32 Hz	64 Hz	128 Hz
	bit 3	bit 2	bit 1	bit 0
TBCR1 (061H) (R/W)	1 Hz	2 Hz	4 Hz	8 Hz

6.4 Time Base Counter Operation

After system reset the time base counter (TBC) begins to count up from 0000H. The count is incremented at the falling edge of the TBCCLK.

TBC 32 Hz/16 Hz/4 Hz/2 Hz outputs are used as time base interrupts. At each output falling edge, four bits of interrupt request register 4 (IRQ4) are set to "1", namely bit 3 (Q32Hz), bit 2 (Q16Hz), bit 1 (Q4Hz) and bit 0 (Q2Hz), requesting an interrupt to the CPU. TBC outputs are also used as clocks for various circuits.

TBC 1 to 8 Hz output and 16 to 128 Hz output can be read through the time base counter register 0/1 (TBCR0/TBCR1).

A write operation to TBCR1 sets the 1 to 8 Hz output counter to "0", and a write operation to TBCR0 sets both the 1 to 8 Hz and 16 to 128 Hz output counters to "0". The write data in these write operations has no significance. For example, the "MOV TBCR0, A" instruction can be used to write, but is not dependent on accumulator content in any way. When write is executed to TBCR0 and TBCR1 and the 1 to 8 Hz and 16 to 128 Hz counters reset, interrupt requests are generated if 32 Hz/16 Hz/4 Hz/2 Hz outputs have been set to "1". To disable these interrupts, first set the master interrupt enable flag (MIE) or interrupt enable register 4 (IE4) to "0", execute the write operation to TBCR 0/1, and set the interrupt request flag 4 (IRQ4) to "0".

Figure 6-2 shows interrupt generation timing and time base counter output reset timing by writing "1" to TBCR0 and TBCR1.

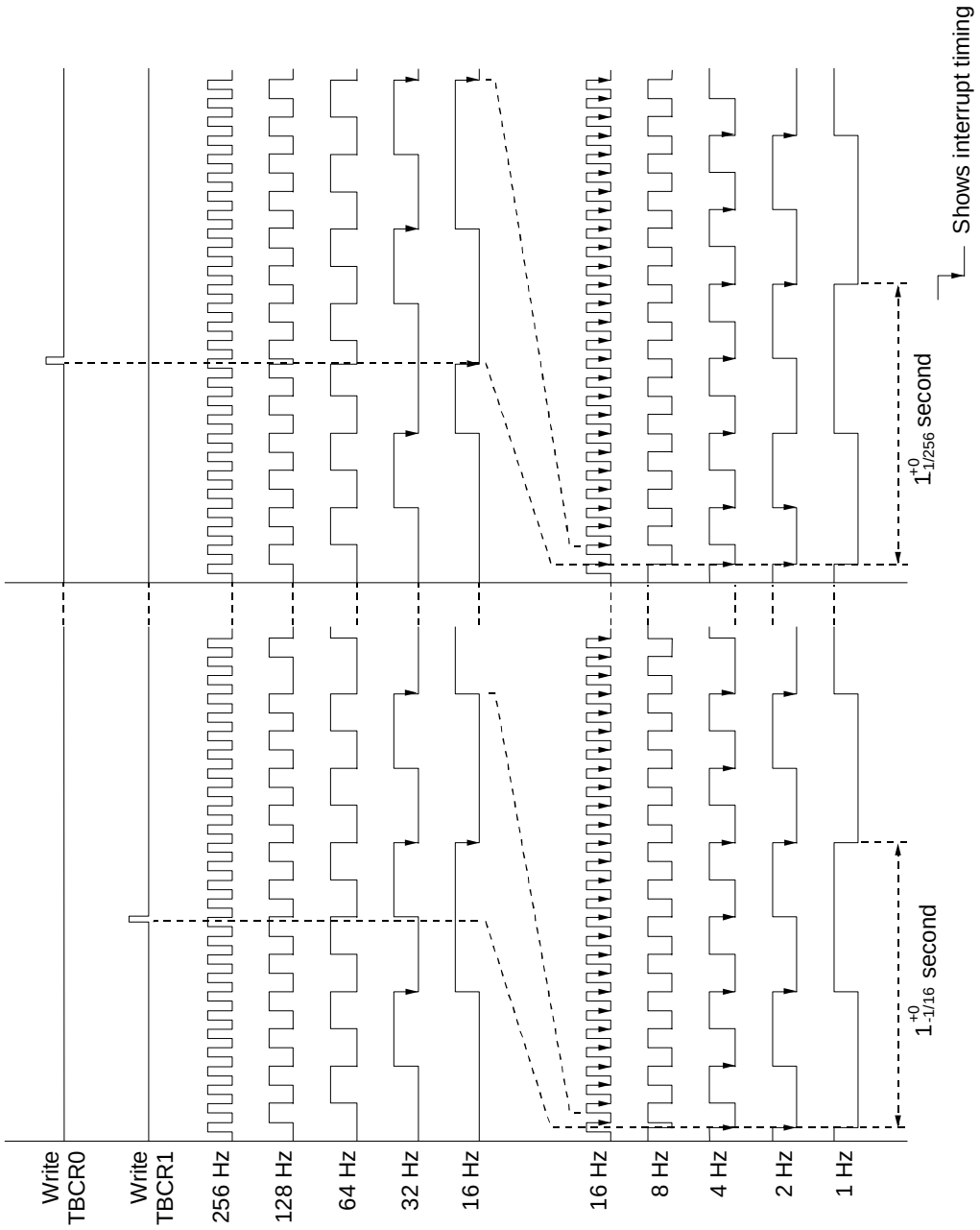


Figure 6-2 Interrupt Timing and Reset Timing by Writing "1" to TBCR0 and TBCR1

Chapter 7

Timers (TIMER)

7. Timers (TIMER)

7.1 Overview

The ML63611 has four internal 8-bit timers (0 to 3). Timers 0 and 1, or timers 2 and 3, can be used in tandem as a 16-bit timer.

Timers 0 and 1 have three operation modes: auto-reload mode, capture mode and frequency measurement mode. Timers 2 and 3 have two modes: auto-reload and frequency measurement. Timer clock may be set to the time base clock (TBCCLK: 32.768 kHz), the high-speed clock (HSCLK), or an external clock. When using the timers as a 16-bit timer, the overflow signals of timers 0 and 2 are used as the clocks for timers 1 and 3, respectively.

Timers can be used not only for pulse generation and time measurement, but as baud rate generators used in serial transmission.

	Timer 0	Timer 1	Timer 2	Timer 3
8-bit timer	●	●	●	●
16-bit timer	● (Timer 0 overflow signal is used as clock for timer 1)		● (Timer 2 overflow signal is used as clock for timer 3)	
Clock	TBCCLK/HSCLK/External clock (T02CK, T13CK)			
Auto-reload mode	●	●	●	●
Capture mode	●	●	—	—
Frequency measurement mode	●	●	●	●

7.2 Timer Configuration

Figures 7-1 through 7-4 show the configuration of timers 0 to 3.

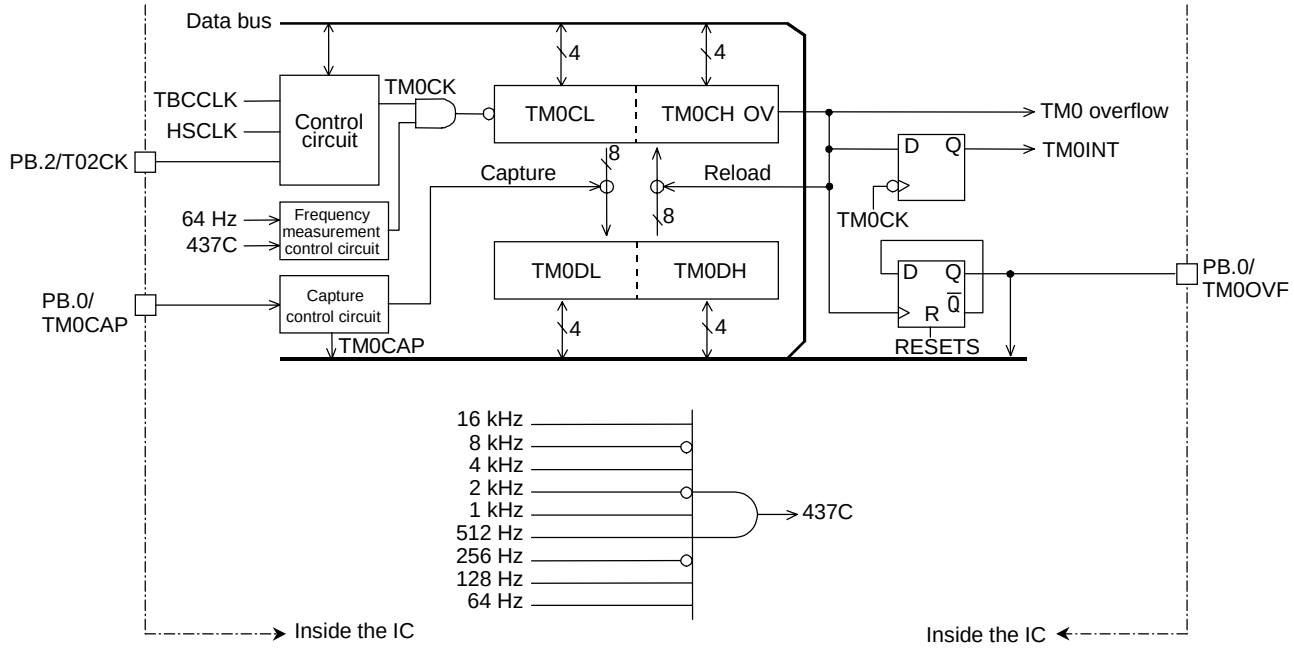


Figure 7-1 Timer 0 Configuration

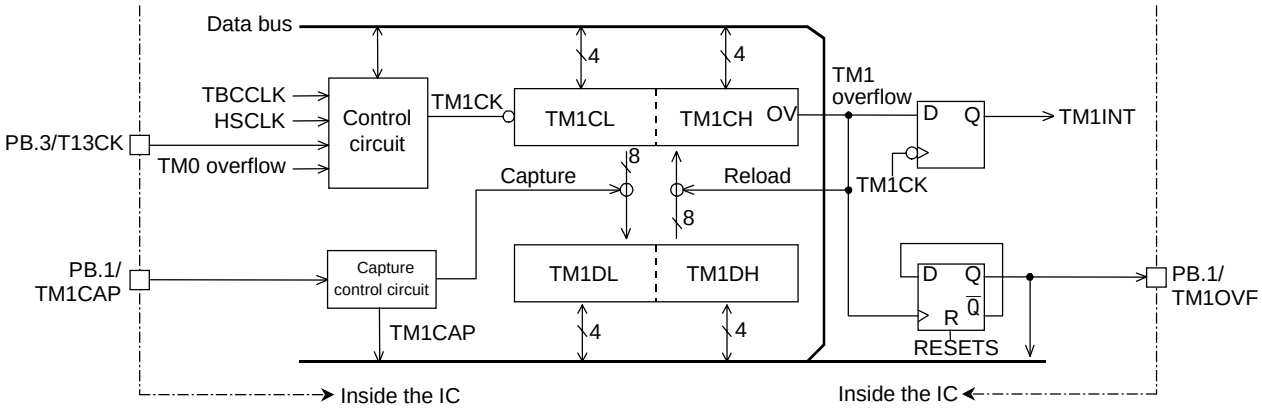


Figure 7-2 Timer 1 Configuration

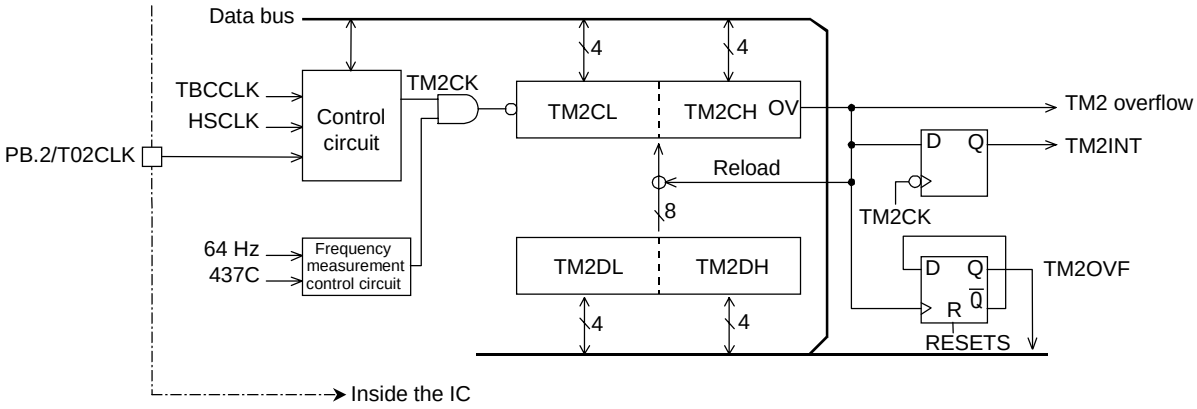


Figure 7-3 Timer 2 Configuration

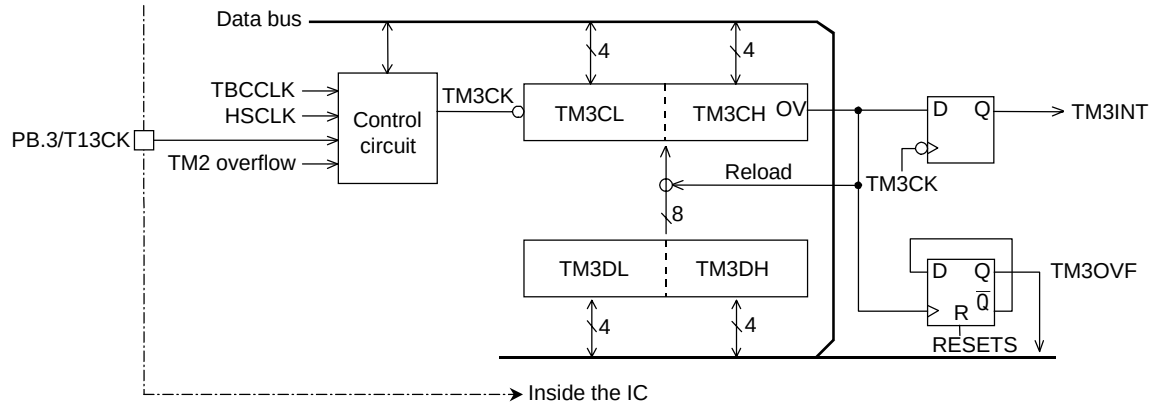


Figure 7-4 Timer 3 Configuration

7.3 Timer Registers

The following four types of registers are used for timer control.

(1) Timer data registers
(TM0DL, TM0DH, TM1DL, TM1DH, TM2DL, TM2DH, TM3DL, TM3DH)
(2) Timer counter registers
(TM0CL, TM0CH, TM1CL, TM1CH, TM2CL, TM2CH, TM3CL, TM3CH)
(3) Timer control registers
(TM0CON0, TM0CON1, TM1CON0, TM1CON1, TM2CON0, TM2CON1, TM3CON0, TM3CON1)
(4) Timer status registers
(TM0STAT, TM1STAT, TM2STAT, TM3STAT)

The registers are described below for each type.

(1) Timer data registers
(TM0DL, TM0DH, TM1DL, TM1DH, TM2DL, TM2DH, TM3DL, TM3DH)

- During the auto-reload mode, timer data registers store the reload values.
- During the capture mode, timer data registers store the capture data.
  Writing to a timer data register causes the contents of the timer counter register to be transferred to the timer data register.
- At system reset, all valid bits are cleared to "0".
- Note regarding register values:
  When a value is written to any timer counter register, the same value is also written to the corresponding timer data register.
  However, when writing to a timer data register, the same value is not written to the corresponding timer counter register.

Timer 0 Registers

		bit 3	bit 2	bit 1	bit 0
TM0DL (Timer 0 lower)	(068H) (R/W)	T0D3	T0D2	T0D1	T0D0
TM0DH (Timer 0 upper)	(069H) (R/W)	T0D7	T0D6	T0D5	T0D4

Timer 1 Registers

		bit 3	bit 2	bit 1	bit 0
TM1DL (Timer 1 lower)	(06AH) (R/W)	T1D3	T1D2	T1D1	T1D0
TM1DH (Timer 1 upper)	(06BH) (R/W)	T1D7	T1D6	T1D5	T1D4

Timer 2 Registers

		bit 3	bit 2	bit 1	bit 0
TM2DL (Timer 2 lower)	(076H) (R/W)	T2D3	T2D2	T2D1	T2D0
TM2DH (Timer 2 upper)	(077H) (R/W)	T2D7	T2D6	T2D5	T2D4

Timer 3 Registers

		bit 3	bit 2	bit 1	bit 0
TM3DL (Timer 3 lower)	(078H) (R/W)	T3D3	T3D2	T3D1	T3D0
TM3DH (Timer 3 upper)	(079H) (R/W)	T3D7	T3D6	T3D5	T3D4

(2) Timer counter registers
(TM0CL, TM0CH, TM1CL, TM1CH, TM2CL, TM2CH, TM3CL, TM3CH)

- 8-bit binary counter operation
- At system reset, all valid bits are cleared to "0".
- Note regarding register values:
  When a value is written to any timer counter register, the same value is also written to the corresponding timer data register.
  However, when writing to a timer data register, the same value is not written to the corresponding timer counter register.

Timer 0 Registers

		bit 3	bit 2	bit 1	bit 0
TM0CL (Timer 0 lower)	(06CH) (R/W)	T0C3	T0C2	T0C1	T0C0
TM0CH (Timer 0 upper)	(06DH) (R/W)	T0C7	T0C6	T0C5	T0C4

Timer 1 Registers

		bit 3	bit 2	bit 1	bit 0
TM1CL (Timer 1 lower)	(06EH) (R/W)	T1C3	T1C2	T1C1	T1C0
TM1CH (Timer 1 upper)	(06FH) (R/W)	T1C7	T1C6	T1C5	T1C4

Timer 2 Registers

		bit 3	bit 2	bit 1	bit 0
TM2CL (Timer 2 lower)	(07AH) (R/W)	T2C3	T2C2	T2C1	T2C0
TM2CH (Timer 2 upper)	(07BH) (R/W)	T2C7	T2C6	T2C5	T2C4

Timer 3 Registers

		bit 3	bit 2	bit 1	bit 0
TM3CL (Timer 3 lower)	(07CH) (R/W)	T3C3	T3C2	T3C1	T3C0
TM3CH (Timer 3 upper)	(07DH) (R/W)	T3C7	T3C6	T3C5	T3C4

(3) Timer control registers
(TM0CON0, TM0CON1, TM1CON0, TM1CON1, TM2CON0, TM2CON1, TM3CON0, TM3CON1)

- Timer control registers select the operation mode and clock for each timer.
- At system reset, all valid bits are cleared to "0".
- Note regarding register values:
  When a value is written to any timer counter register, the same value is also written to the corresponding timer data register.
  However, when writing to a timer data register, the same value is not written to the corresponding timer counter register.

Timer 0 Registers

To use timer 1 in combination as a 16-bit timer, set timer 1 control registers TM1CON0 and TM1CON1.

TM0CON0 (070H) (R/W)

bit 3	bit 2	bit 1	bit 0
—	FMEAS0	TM0ECAP	TM0RUN

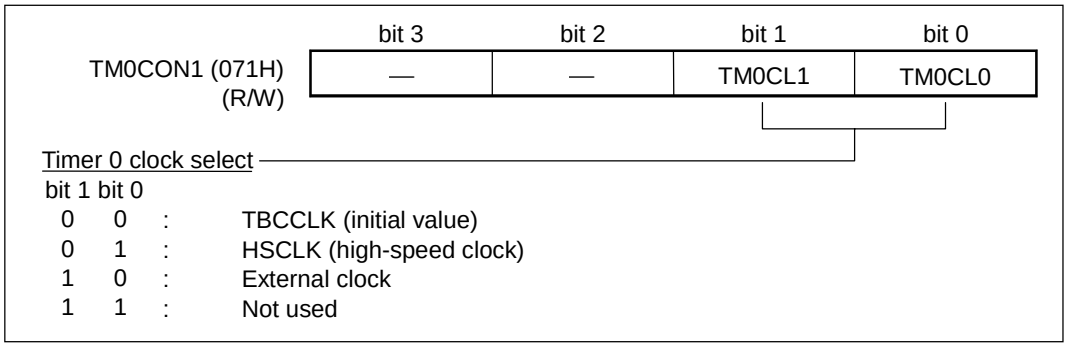
Timer 0 mode select

bit 2	bit 1	bit 0		
0	0	0	:	Auto-reload mode stop (initial value)
0	0	1	:	Auto-reload mode operation
0	1	0	:	Capture mode stop
0	1	1	:	Capture mode operation
1	0	0	:	Frequency measurement mode operation
1	0	1	:	Not used
1	1	0	:	Not used
1	1	1	:	Not used

bit 2, 1, 0: FMEAS0, TM0ECAP, TM0RUN

These bits select the timer 0 operation mode.

The timer 0 operation mode can be selected as auto-reload mode, capture mode, or frequency measurement mode.



bit 1, 0: TM0CL1, TM0CL0

These bits select the timer 0 clock.

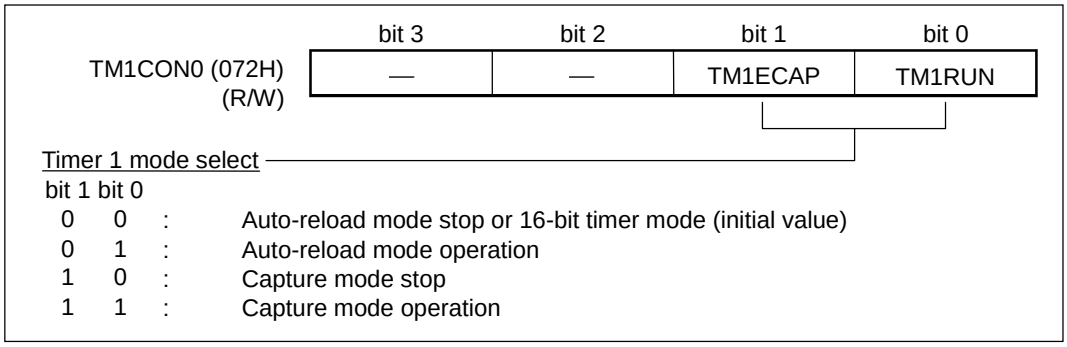
The timer 0 clock can be selected as TBCCLK (low-speed clock), HSCLK (high-speed clock), or external clock (T02CK: secondary function of PB.2).

Note:

If HSCLK is used as the clock, after ENOSC (bit 1 of FCON) is set to "1", wait for the following time interval before starting timer operation.

- Wait at least 10 ms when using ceramic oscillation.
- Wait at least 300 μs when using RC oscillation.

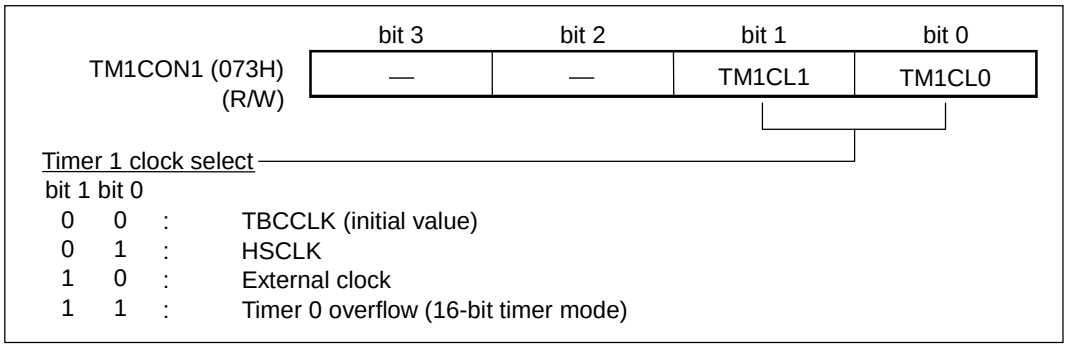
Timer 1 Registers



bit 1, 0: TM1ECAP, TM1RUN

These bits select the timer 1 operation mode.

The timer 1 operation mode can be selected as auto-reload mode, capture mode, or 16-bit timer mode.



bit 1, 0: TM1CL1, TM1CL0

These bits select the timer 1 clock.

The timer 1 clock can be selected as TBCCLK (low-speed clock), HSCLK (high-speed clock), external clock (T13CK: secondary function of PB.2), or the timer 0 overflow flag.

When using as a 16-bit timer, select timer 0 overflow for the clock.

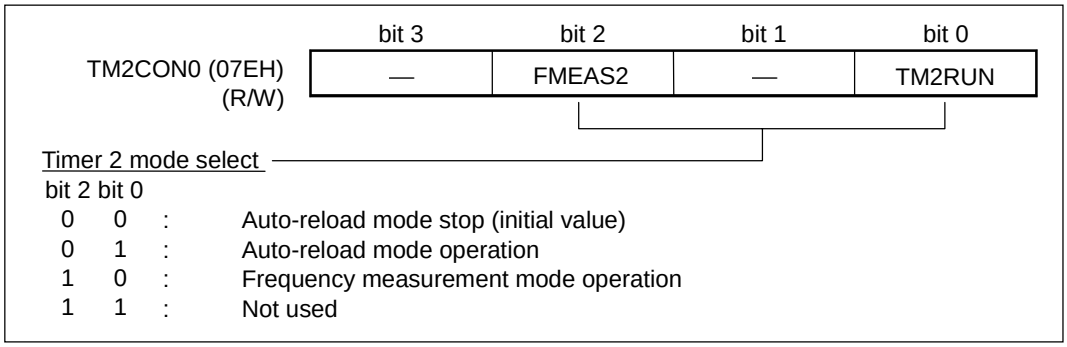
Note:

If HSCLK is used as the clock, after ENOSC (bit 1 of FCON) is set to "1", wait for the following time interval before starting timer operation.

- Wait at least 10 ms when using ceramic oscillation.
- Wait at least 300 μs when using RC oscillation.

Timer 2 Registers

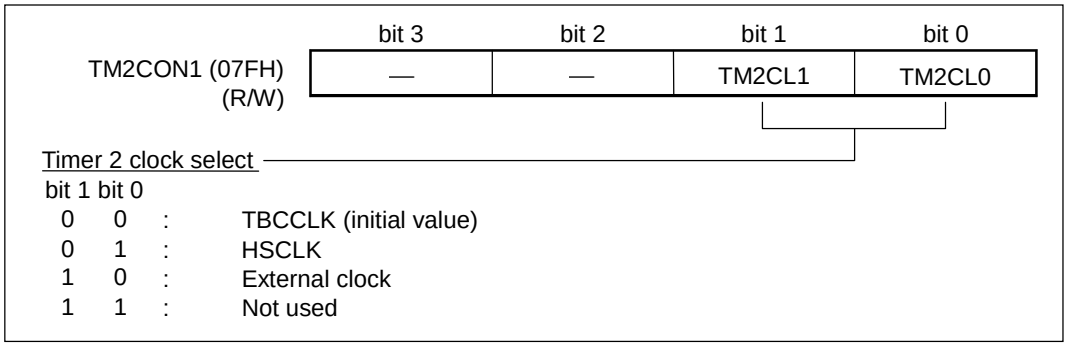
To use timer 3 in combination as a 16-bit timer, set timer 3 control registers TM3CON0 and TM3CON1.



bit 2, 0: FMEAS2, TM2RUN

These bits select the timer 2 operation mode.

The timer 2 operation mode can be selected as auto-reload mode or frequency measurement mode.



bit 1, 0: TM2CL1, TM2CL0

These bits select the timer 2 clock.

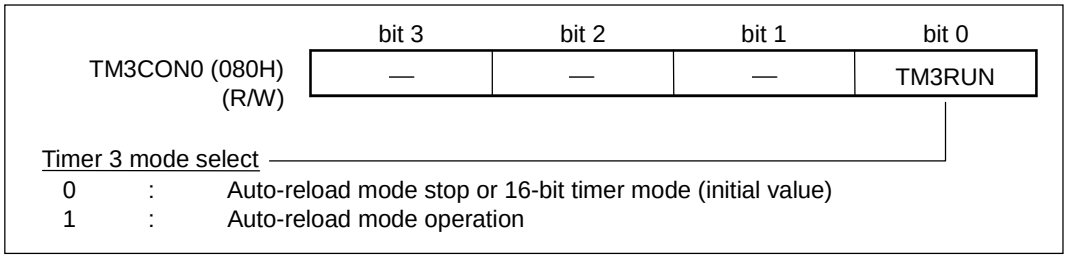
The timer 2 clock can be selected as TBCCLK (low-speed clock), HSCLK (high-speed clock), or external clock (T02CK: secondary function of PB.2).

Note:

If HSCLK is used as the clock, after ENOSC (bit 1 of FCON) is set to "1", wait for the following time interval before starting timer operation.

- Wait at least 10 ms when using ceramic oscillation.
- Wait at least 300 μs when using RC oscillation.

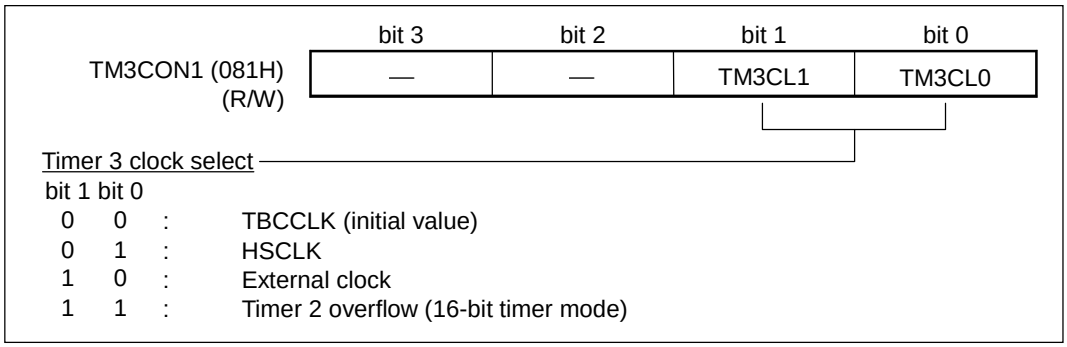
Timer 3 Registers



bit 0: TM3RUN

This bit selects the timer 3 operation mode.

The timer 3 operation mode can be selected as auto-reload mode or 16-bit timer mode.



bit 1, 0: TM3CL1, TM3CL0

These bits select the timer 3 clock.

The timer 3 clock can be selected as TBCCLK (low-speed clock), HSCLK (high-speed clock), external clock (T13CK: secondary function of PB.3), or the timer 2 overflow flag.

When using as a 16-bit timer, select timer 2 overflow for the clock.

Note:

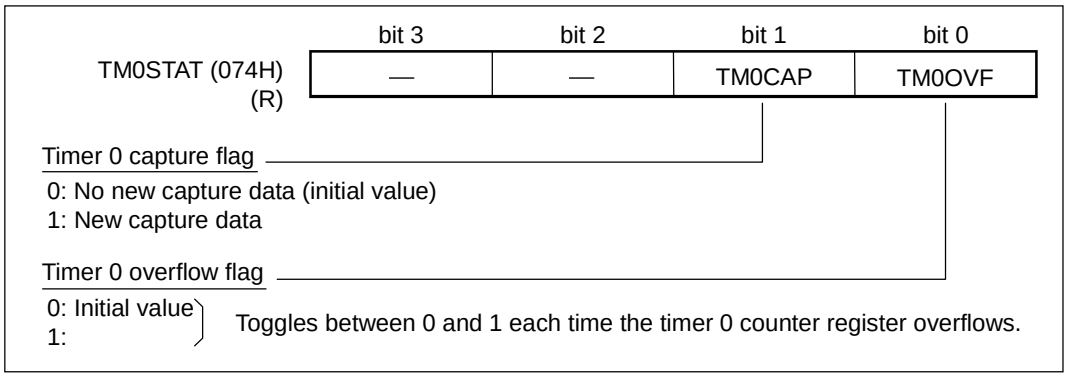
If HSCLK is used as the clock, after ENOSC (bit 1 of FCON) is set to "1", wait for the following time interval before starting timer operation.

- Wait at least 10 ms when using ceramic oscillation.
- Wait at least 300 μs when using RC oscillation.

(4) Timer status registers (TM0STAT, TM1STAT, TM2STAT, TM3STAT)

- Timer status registers read the status of each timer.
- At system reset, all valid bits are cleared to "0".

Timer 0 Registers



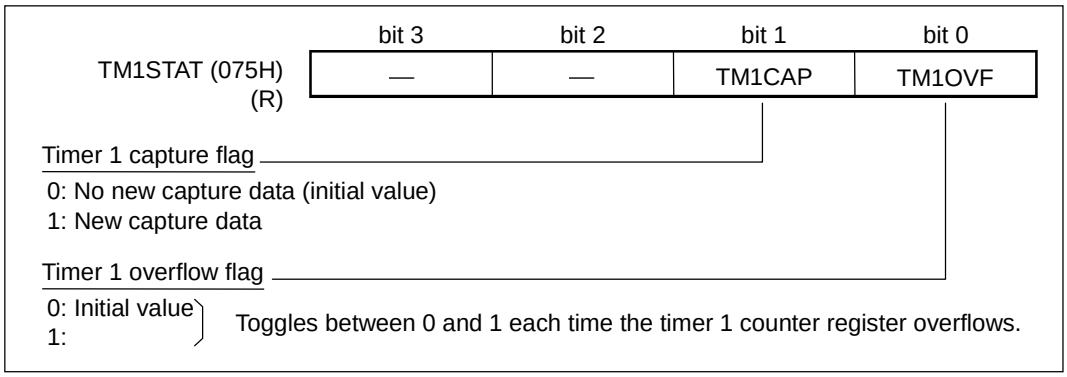
bit 1: TM0CAP (TiMer0 CAPture)

This bit indicates whether or not new capture data is present.
When TM0CAP = "0":
A value of "0" indicates that there has been no new capture data since system reset or since the last time TM0CAP was read.
When TM0CAP = "1":
A value of "1" indicates that there is new capture data since system reset or since the last time TM0CAP was read. Additional captures are disabled.
At system reset, TM0CAP is cleared to "0".
In the capture mode, if the level of the capture input pin (PB.0/TM0CAP) changes and a capture is generated, TM0CAP is automatically set to "1".
If TM0STAT is read, TM0CAP is automatically cleared to "0".

bit 0: TM0OVF (TiMer0 OVerFlow)

This bit indicates that the timer counter register has overflowed.
This bit toggles between "0" and "1" whenever overflow occurs.
At system reset, TM0OVF is cleared to "0".

Timer 1 Registers



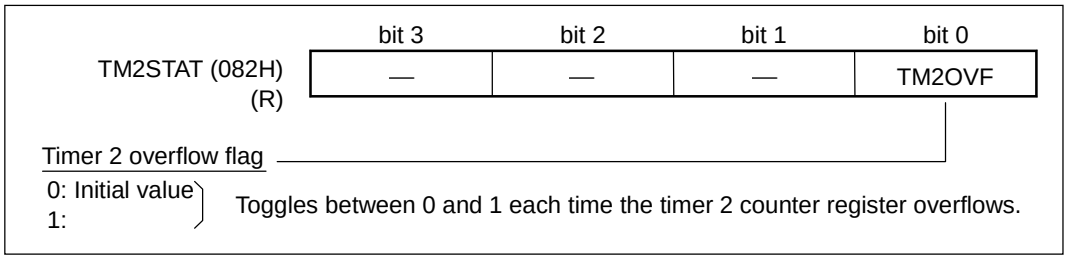
bit 1: TM1CAP (TiMer1 CAPture)

This bit indicates whether or not new capture data is present.
When TM1CAP = "0":
A value of "0" indicates that there has been no new capture data since system reset or since the last time TM1CAP was read.
When TM1CAP = "1":
A value of "1" indicates that there is new capture data since system reset or since the last time TM0CAP was read. Additional captures are disabled.
At system reset, TM1CAP is cleared to "0".
In the capture mode, if the level of the capture input pin (PB.1/TM1CAP) changes and a capture is generated, TM1CAP is automatically set to "1".
If TM1STAT is read, TM1CAP is automatically cleared to "0".

bit 0: TM1OVF (TiMer1 OVerFlow)

This bit indicates that the timer counter register has overflowed.
This bit toggles between "0" and "1" whenever overflow occurs.
At system reset, TM1OVF is cleared to "0".

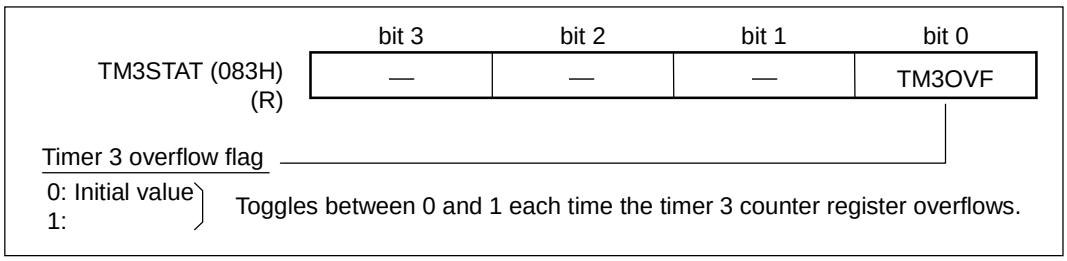
Timer 2 Register



bit 0: TM2OVF (TiMer2 OVerFlow)

This bit indicates that the timer counter register has overflowed.
This bit toggles between "0" and "1" whenever overflow occurs.
At system reset, TM2OVF is cleared to "0".

Timer 3 Register



bit 0: TM3OVF (TiMer3 OVerFlow)

This bit indicates that the timer counter register has overflowed.
This bit toggles between "0" and "1" whenever overflow occurs.
At system reset, TM3OVF is cleared to "0".

[Supplement] List of Timer Registers

Timer 0 Registers

Name	Symbol	Address	R/W	Initial value
Timer 0 data register L	TM0DL	068H	R/W	0H
Timer 0 data register H	TM0DH	069H		0H
Timer 0 counter register L	TM0CL	06CH	R/W	0H
Timer 0 counter register H	TM0CH	06DH		0H
Timer 0 control register 0	TM0CON0	070H	R/W	8H
Timer 0 control register 1	TM0CON1	071H		0CH
Timer 0 status register	TM0STAT	074H	R	0CH

Timer 1 Registers

Name	Symbol	Address	R/W	Initial value
Timer 1 data register L	TM1DL	06AH	R/W	0H
Timer 1 data register H	TM1DH	06BH		0H
Timer 1 counter register L	TM1CL	06EH	R/W	0H
Timer 1 counter register H	TM1CH	06FH		0H
Timer 1 control register 0	TM1CON0	072H	R/W	0CH
Timer 1 control register 1	TM1CON1	073H		0CH
Timer 1 status register	TM1STAT	075H	R	0CH

Timer 2 Registers

Name	Symbol	Address	R/W	Initial value
Timer 2 data register L	TM2DL	076H	R/W	0H
Timer 2 data register H	TM2DH	077H		0H
Timer 2 counter register L	TM2CL	07AH	R/W	0H
Timer 2 counter register H	TM2CH	07BH		0H
Timer 2 control register 0	TM2CON0	07EH	R/W	0AH
Timer 2 control register 1	TM2CON1	07FH		0CH
Timer 2 status register	TM2STAT	082H	R	0EH

Timer 3 Registers

Name	Symbol	Address	R/W	Initial value
Timer 3 data register L	TM3DL	078H	R/W	0H
Timer 3 data register H	TM3DH	079H		0H
Timer 3 counter register L	TM3CL	07CH	R/W	0H
Timer 3 counter register H	TM3CH	07DH		0H
Timer 3 control register 0	TM3CON0	080H	R/W	0EH
Timer 3 control register 1	TM3CON1	081H		0CH
Timer 3 status register	TM3STAT	083H	R	0EH

7.4 Timer Operation

7.4.1 Timer Clock

The timer clock can be selected as TBCCLK (low-speed clock: 32.768 kHz), HSCLK (high-speed clock), or an external clock. By using timer 0 and timer 2 overflow signals as clocks for timer 1 and timer 3, respectively, the timers can be used in pairs as 16-bit timers.

If the high-speed clock (HSCLK) is to be used, after setting ENOSC (bit 1 of FCON), wait at least 10 ms in the ceramic oscillation mode or 300 µs in the RC oscillation mode before operating the timer.

The external clock is input to a port assigned as a secondary function port. In the case of timers 0 and 2, PB.2/T02CK is used as the input pin for the external clock. In the case of timers 1 and 3, PB.3/T13CK is used as the input pin for the external clock. Since the external clock is sampled by the system clock (CLK), the high- and low-levels of the external clock should be longer than 1 cycle of the system clock (CLK).

7.4.2 Timer Data Registers

TM0DL, TM0DH, TM1DL, TM1DH, TM2DL, TM2DH, TM3DL and TM3DH are 4-bit registers.

In the auto-reload mode, the timer data registers save values that are reloaded into the timer counter registers when the timer counter registers overflow.

In the capture mode, the timer data registers save the value of the timer counter registers when a capture signal is input. Each timer data register can be read/written by software. Writing to timer data registers does not change the contents of the timer counter registers.

7.4.3 Timer Counter Registers

TM0CL and TM0CH, TM1CL and TM1CH, TM2CL and TM2CH, and TM3CL and TM3CH are 8-bit binary counters that are incremented at the falling edge of the timer clock.

Each timer counter register can be read/written by software. However, if the CPU clock and timer clock are different, values that are read or written during the count operation cannot be guaranteed. If an external clock is used as the timer clock, reading/writing is always possible.

When a value is written to any timer counter register, the same value is also written to the corresponding timer data register.

7.4.4 Timer Interrupt Requests and Overflow Flags

Timers generate timer interrupt requests when the timer counter register overflows. The overflow flag toggles between "1" and "0" at each overflow. The output of the overflow flag of timers 0 and 1 can be output to secondary port functions PB.0/TM0OVF and PB.1/TM1OVF pins.

Figure 7-5 indicates the operation timing for timer counter register overflow. Table 7-1 lists timer interrupts.

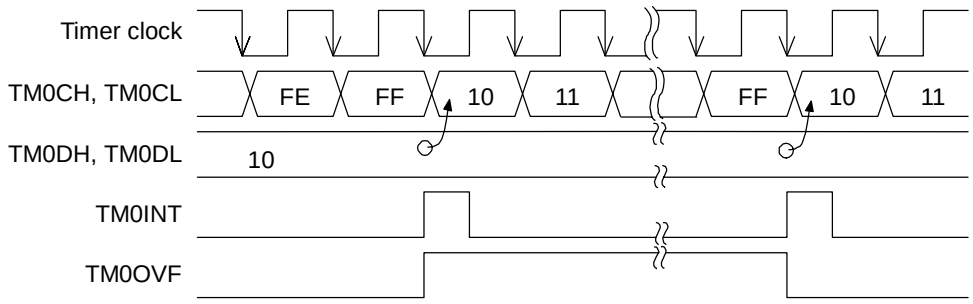


Figure 7-5 Timer Counter Register Overflow Timing (for Timer 0)

Table 7-1 List of Timer Interrupts

Interrupt factor	Symbol	IRQ flag (IRQ2)	IE flag (IE2)	Interrupt vector address
Timer 0 interrupt	TM0INT	QTM0	ETM0	0020H
Timer 1 interrupt	TM1INT	QTM1	ETM1	0022H
Timer 2 interrupt	TM2INT	QTM2	ETM2	0024H
Timer 3 interrupt	TM3INT	QTM3	ETM3	0026H

When the master interrupt enable flag (MIE) is set to "1" with the interrupt enable flags (ETM0–3) set to "1", and a timer overflow occurs, a CPU interrupt request is generated.

7.4.5 Auto-Reload Mode Operation

Timers 0 to 3 can be used as auto-reload mode timers. The setup method is as follows.

- Timer 0: Set FMEAS0 (bit 2 of TM0CON0) to "0", and set TM0ECAP (bit 1 of TM0CON0) to "0".
- Timer 1: Set TM1ECAP (bit 1 of TM1CON0) to "0".
- Timer 2: Set FMEAS2 (bit 2 of TM2CON0) to "0".
- Timer 3: No setup needed.

In the auto-reload mode, each time the timer counter register overflows, the timer data register value is reloaded into the timer counter register, and counting begins from the value. Setting the RUN bits (TM0RUN, TM1RUN, TM2RUN, TM3RUN) for each timer control register to "1" will restart the count, and resetting to "0" stops the count.

In the 16-bit timer mode for timers 0 and 1 the TM1RUN bit is disabled, and start/stop is controlled with the TM0RUN bit. In the 16-bit timer mode for timers 2 and 3 the TM3RUN bit is disabled, and start/stop is controlled with the TM2RUN bit.

Figure 7-6 shows auto-reload mode timing for pulse generation when timers 0 and 1 are used as a 16-bit timer.

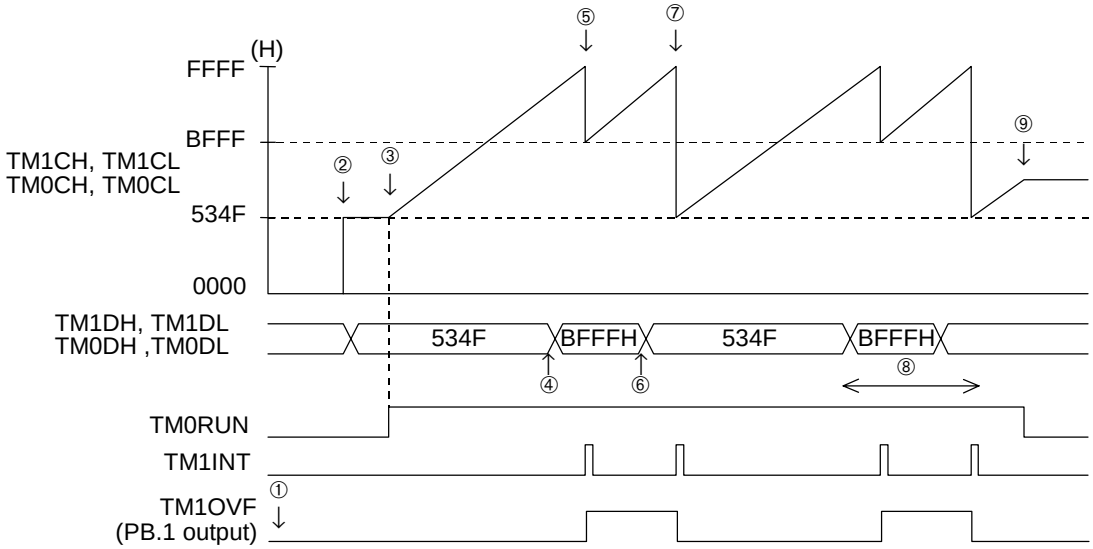


Figure 7-6 Auto-Reload Mode Timing

The operation procedures are as follows.

① Set PB.1 to the output mode (TM1OVF) secondary function.

② Write 534FH to the timer data and timer counter registers.
TM1DH = TM1CH = 5H (bits 15–12)
TM1DL = TM1CL = 3H (bits 11–8)
TM0DH = TM0CH = 4H (bits 7–4)
TM0DL = TM0CL = FH (bits 3–0)

③ If TM0CON and TM1CON are set to auto-reload mode and TM0RUN is set to "1", the timer counter register will start to count from 534FH.

④ Before the timer counter register overflows, write the next reload value BFFFH to the timer data register.

⑤ When the timer counter register overflows, BFFFH is set to the timer counter register, timer interrupt TM1INT is generated and timer 1 overflow flag TM1OVF toggles. The timer counter register continues to count up from BFFFH.

⑥ Before the timer counter register overflows, write the next reload value 534FH to the timer data register.

⑦ When the timer counter register overflows, 534FH is set to the timer counter register, timer interrupt TM1INT is generated and timer 1 overflow flag TM1OVF toggles. The timer counter register resumes counting from address 534FH.

⑧ Repeat steps ④ through ⑦. This allows a user-defined pulse to be output from PB.1/TM1OVF.

⑨ Halt the count by resetting TM0RUN to "0".

Figure 7-7 shows TM0RUN count start/halt timing.

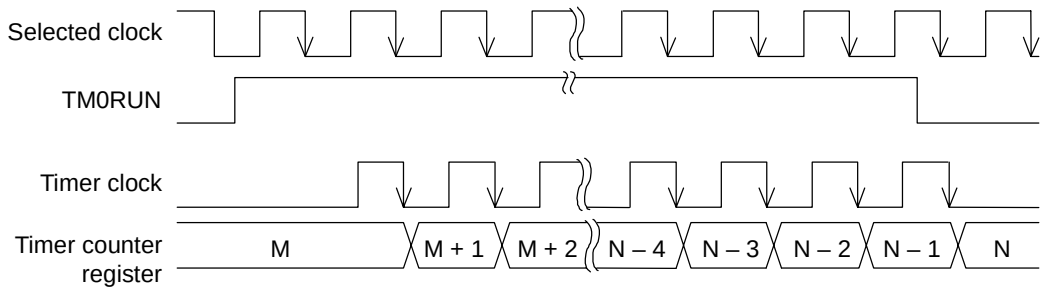


Figure 7-7 TM0RUN Count Start/Halt Timing

When TM0RUN is set to "1", the timer counter starts to count from the second falling edge of the selected clock. When TM0RUN is reset to "0", the counter stops counting at the falling edge of the selected clock which appears immediately after the TM0RUN falling edge.

7.4.6 Capture Mode Operation

Timer 0 and timer 1 can be used as capture mode timers.

In a capture mode, a change in the capture input (PB.0/TM0CAP, PB.1/TM1CAP) level during operation of the timer counter register triggers loading of the value of the timer counter register into the timer data register.

Methods to set the capture mode for each timer are listed below.

- Timer 0: Set TM0ECAP (bit 1 of TM0CON0) to "1", and set FMEAS0 (bit 2 of TM0CON0) to "0".
- Timer 1: Set TM1ECAP (bit 1 of TM1CON0) to "1".

In the capture mode, reloading the timer data register data into the timer counter register is inhibited, and when the timer counter register overflows, counting is restarted from 00H.

When a capture occurs, the capture flags (TM0CAP, TM1CAP) of the timer status registers (TM0STAT, TM1STAT) are set to "1". Additional captures are disabled while the capture flags are "1". The capture flags are assigned to bit 0 of the timer status registers, and are automatically cleared to "0" when the timer status registers are read.

If both the TM1CL1 and TM1CL0 bits of the timer 1 control register 1 (TM1CON1) are set to "1" and timer 0 overflow is selected as the clock, the 16-bit capture mode will be set. In this case, the PB.0/TM0CAP pin is the capture trigger input.

Figure 7-8 shows the timer 0 capture mode timing for pulse width measurement.

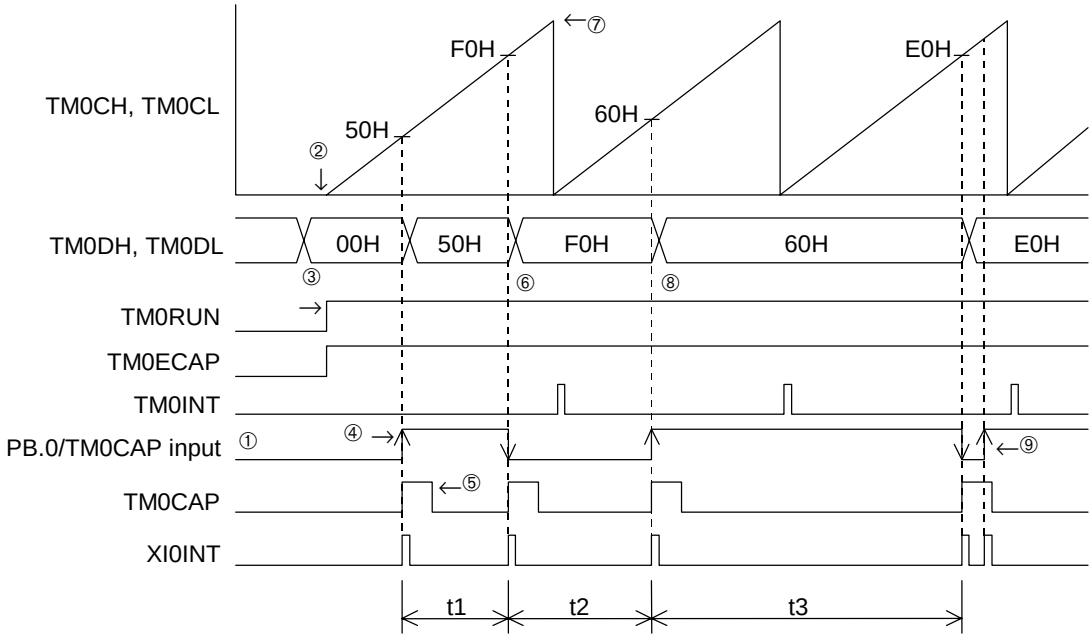


Figure 7-8 Capture Mode Timing

The operation procedure is listed below.

① Set PB.0/TM0CAP to input mode, and enable XI0INT and TM0INT.

② Clear all bits of the timer counter registers and timer data registers to zero.

③ Set TM0CON0 to the capture mode, and set TM0RUN to "1" to begin upward counting.

④ If the PB.0/TM0CAP input changes, the TM0CH/TM0CL value is captured by TM0DH/TM0DL and TM0CAP is set to "1" (first capture). The CPU detects this through XI0INT and reads the values of TM0DH/TM0DL.

⑤ After the TM0DH/TM0DL read is complete, TM0CAP is cleared to "0" to wait for the next capture.

⑥ If the PB.0/TM0CAP input changes, repeat operations ④ and ⑤ (second capture).
The high-level pulse width t1 of the PB.0 input can be determined as follows.

$$t1 = (F0H - 50H) \times t_{CLK} \qquad t_{CLK}: \text{TMCLK cycle}$$

⑦ TM0INT is generated when the timer counter register overflows. When overflow occurs, the timer counter register changes from FFH to 00H and continues upward counting.

⑧ If the PB.0/TM0CAP input changes, repeat operations ④ and ⑤ (third capture). Because the counter overflows once during the interval between the second capture and the third capture, the low-level pulse width t2 of the PB.0 input can be determined as follows.

$$t2 = (60H - F0H + 100H) \times t_{CLK}$$

⑨ While TM0CAP = "1", there is no capture even when PB.0/TM0CAP changes.

Figure 7-9 shows the capture timing and Figure 7-10 shows the capture signal (CAPT) generator circuit.

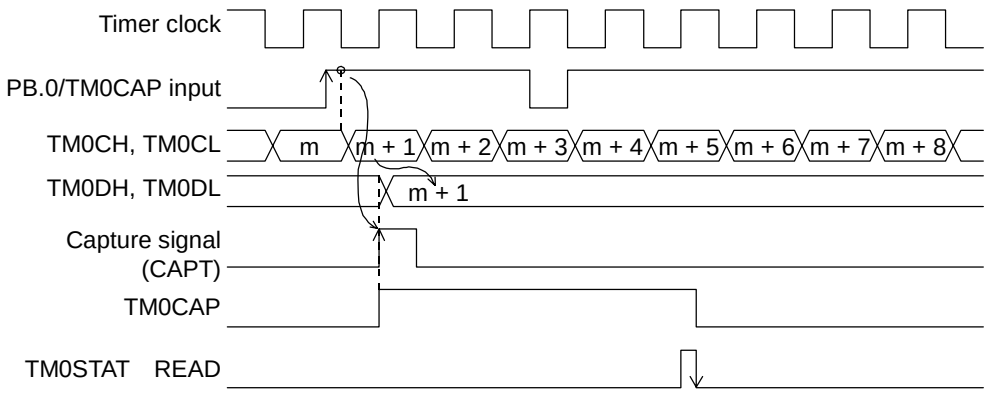


Figure 7-9 Capture Timing

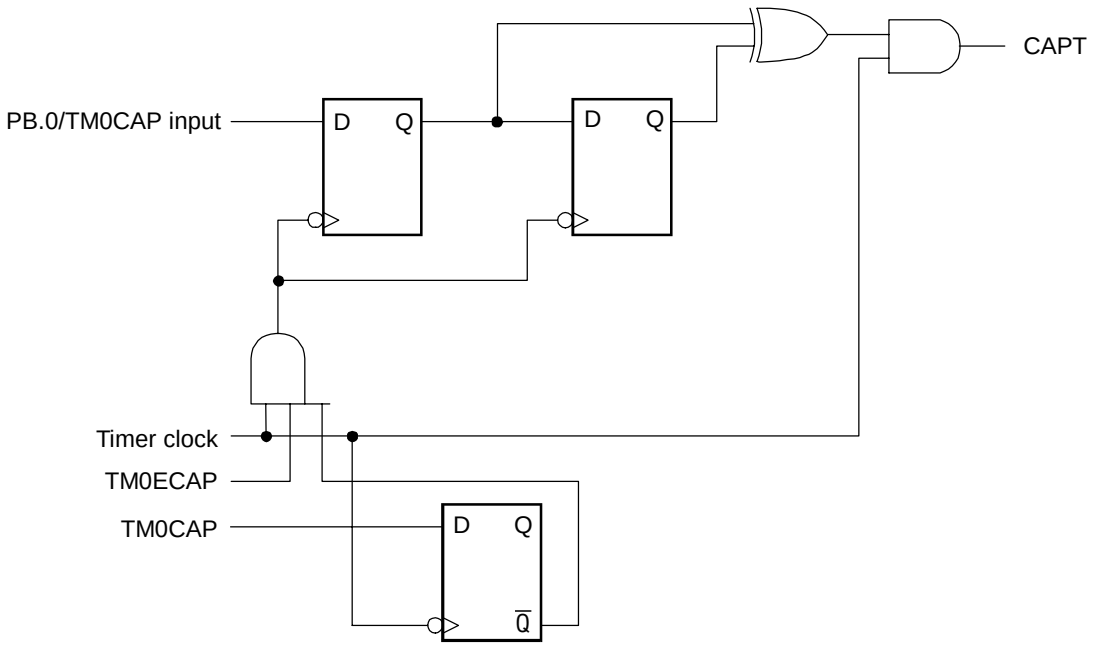


Figure 7-10 Capture Signal (CAPT) Generator Circuit

Note:

Set the pulse width of the capture trigger signal to be input to more than or equal to two cycles of the timer clock. Typical examples of valid pulse widths of the trigger signal are shown below.

- When a low-speed clock (32.768 kHz) is used as the timer clock, a pulse width of 62 μs or more is required.
- When a high-speed clock (2 MHz) is used as the timer clock, a pulse width of 1 μs or more is required.
- When an external clock is used as the timer clock, a pulse width of more than or equal to two cycles of the external clock is required.

7.4.7 Frequency Measurement Mode Operation

The frequency measurement mode is used to measure the frequency of the RC oscillator clock, which has wide product variation.

Timers 0 and 1, and timers 2 and 3 can be used in the frequency measurement mode. These timers are set as follows for the frequency measurement mode:

- Timer 0: Set FMEAS0 (bit 2 of TM0CON0) to "1", and set TM0ECAP (bit 1 of TM0CON0) and TM0RUN (bit 0 of TM0CON0) to "0".
- Timer 1: Set TM1ECAP (bit 1 of TM1CON0) and TM1RUN (bit 0 of TM1CON0) to "0".
- Timer 2: Set FMEAS2 (bit 2 of TM2CON0) to "1", and set TM2RUN (bit 0 of TM2CON0) to "0".
- Timer 3: Set TM3RUN (bit 0 of TM3CON0) to "0".

The count obtained in the frequency measurement mode can be used to determine the auto-reload mode timer data register value, thereby making the timer overflow to generate various signals with required cycles. During serial transmission, the timer 3 interrupt signal (TM3INT) is used as the baud rate clock.

Figure 7-11 indicates frequency measurement mode timing when timers 2 and 3 are used as a 16-bit timer.

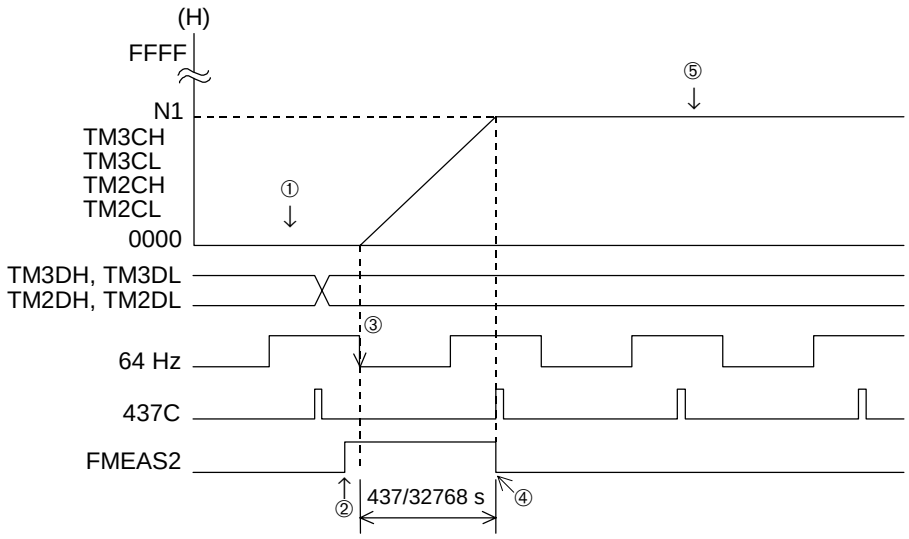


Figure 7-11 Frequency Measurement Mode Timing

The operation sequence for Figure 7-11 is as follows.

① Timer 3 control registers 0 and 1 (TM3CON0, TM3CON1) are set for 16-bit timer mode, and the timer counter and timer data register are cleared to "0". Enable the high-speed clock by the frequency control register (FCON) and the timer clock is set to HSCLK.
② Wait 10 ms or more in the ceramic oscillation mode or 300 µs or more in the RC oscillation mode after starting the high-speed clock and set FMEAS2 to "1" to enter the frequency measurement mode.
③ When FMEAS2 is "1", the counter starts at the 64 Hz falling edge.
④ When the 437C signal is "1", FMEAS2 is reset to "0", and the counter stops at the falling edge of the next clock. The 437C signal is a pulse signal which rises in 437/32768 second after the 64 Hz falling edge.
⑤ Timer counter register value N1 is read.

Assuming that the ceramic oscillation clock is exactly 700 kHz, value N1 read from the timer counter register is:

N1 = 700000 × 437/32768
= 9335 (decimal)
= 2477 (hexadecimal)
= 0010 0100 0111 0111 (binary)
(truncated)

Because 437/32768 second is equivalent to 128 clocks at 9600 Hz (more precisely, 9598 Hz), a division of the count by 128 provides the frequency ratio (N2) between 700 kHz and 9600 Hz. Because $128 = 2^7$, that can be determined by merely truncating the righthand seven digits of N1 (binary), yielding

N2 = 9335/128 = 001001000 (binary)
= 48 (hexadecimal)
= 72 (decimal)

This indicates that 9600 Hz is about 72 times the cycle of 700 kHz, which means that the timer data register should be set to FFB8H so that the counter overflows every 72 counts of the 700 kHz clock in auto-reload mode. As a result, overflow produces a TM3INT cycle t_{TM3INT} of

$t_{TM3INT} = 1/700000 \times 72 = 0.102857$ ms (9722 Hz)

In the same way, assuming that RC oscillation clock is 600 kHz due to manufacturing variation, we get

N1 = 600000 × 437/32768 = 8001 (decimal)
= 1F41 (hexadecimal)
= 0001 1111 0100 0001 (binary)
(truncated)

Truncating the righthand seven digits of N1 (binary), we get

N2 = 8001/128 = 000111110 (binary)
= 3E (hexadecimal)
= 62 (decimal)

Set the timer data register to FFC2H so that the counter overflows every 62 counts of the 600 kHz clock in auto-reload mode. As a result, overflow produces a TM3INT cycle t_{TM3INT} of

$t_{TM3INT} = 1/600000 \times 62 = 0.10333$ ms (9677 Hz)

In this way the frequency measurement mode can be applied to generate TM3INT signals with precision cycles. These TM3INT signals can be supplied to the serial port as a baud rate clock.

Changing the value of N2 makes it possible to generate baud rates of 4800 Hz, 2400 Hz or user-defined rates. The precision of the generated baud rate clock is within ±2% for 9600 Hz, and within ±1% for 4800 Hz or lower.

Figure 7-12 illustrates the operation of baud ratae clock for an RC oscillator clock frequency of 600 kHz.

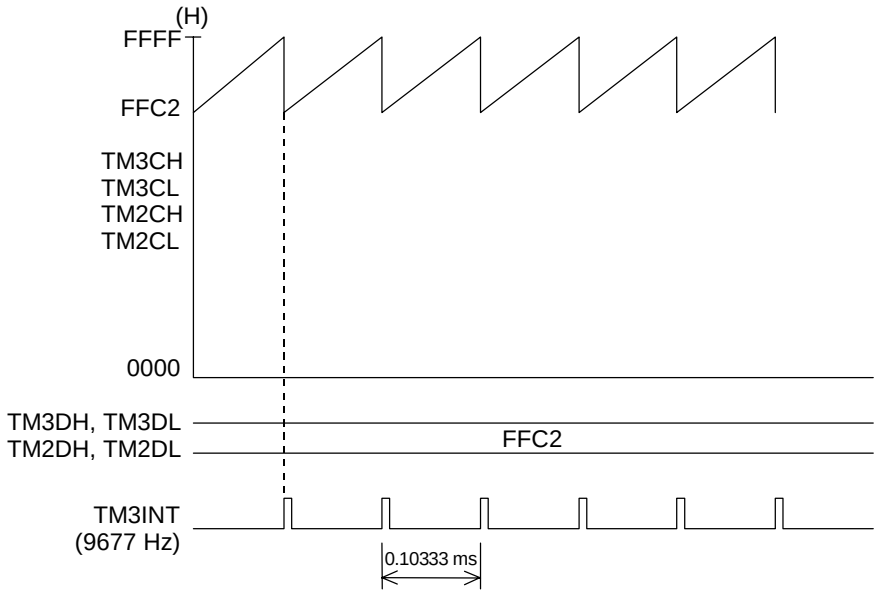


Figure 7-12 Baud Rate Clock Generation

Chapter 8

100 Hz Timer Counter (100HzTC)

8. 100 Hz Timer Counter (100HzTC)

8.1 Overview

The 100 Hz timer counter has a circuit that divides the TBC6 output (512 Hz) of the time base counter to generate a 10 Hz interrupt. The 100 Hz timer consists of a 5/6-base counter and two decimal counters.

8.2 100 Hz Timer Counter Configuration

Figure 8-1 indicates the configuration of the 100 Hz timer counter.

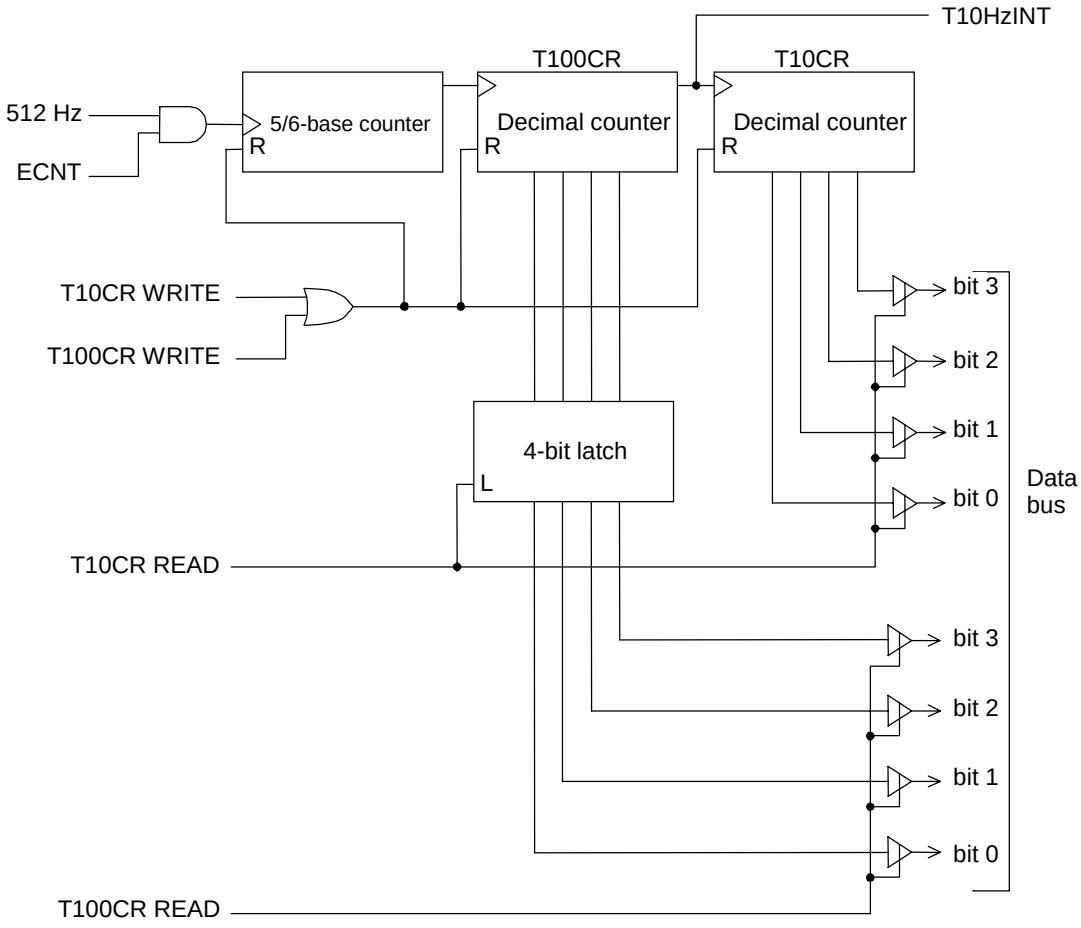
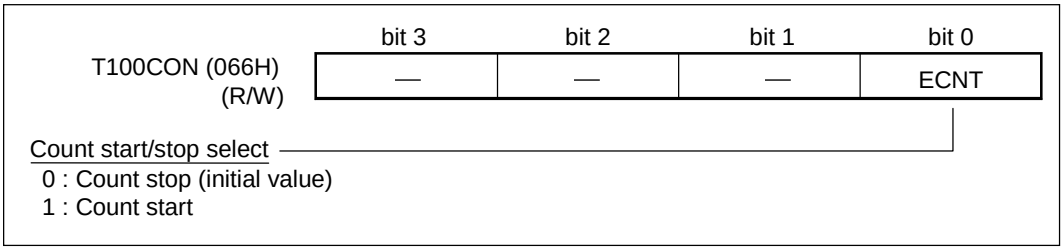


Figure 8-1 100 Hz Timer Counter Configuration

8.3 100 Hz Timer Counter Registers

(1) 100 Hz timer counter control register (T100CON)

This is a 4-bit special function register (SFR) controlling the 100 Hz timer counter.

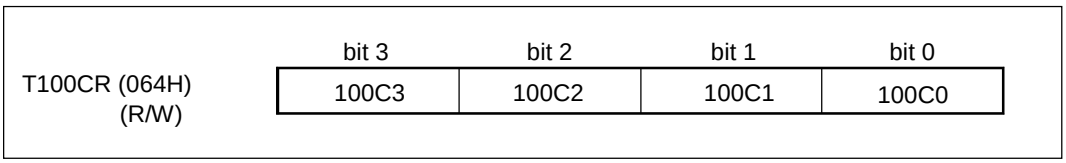


bit 0: ECNT
This bit controls count start/stop for the 100 Hz timer counter internal counter. Count starts when set to "1". At system reset the value is reset to "0" and counting is stopped.

(2) 100 Hz counter register (T100CR)

This is a 4-bit special function register (SFR) to read the 100 Hz counter of the 100 Hz timer counter. The content of the T100CR is latched by a 4-bit latch in T10CR read operation, so the value of the T100CR must always be read after reading T10CR.

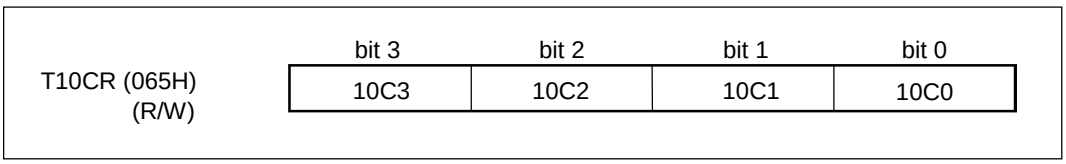
When data is written in T100CR, both T100CR and T10CR are reset to "0".



(3) 10 Hz counter register (T10CR)

A 4-bit special function register (SFR) to read the 10 Hz counter in the 100 Hz timer counter.

When data is written in T10CR, both T100CR and T10CR are reset to "0".



8.4 100 Hz Timer Counter Operation

The 100 Hz timer counter begins counting when bit 0 (ECNT) of the 100 Hz timer counter control register (T100CON) is set to "1". The 512 Hz output of the time base counter is divided into 100 Hz by the 5/6-base counter.

The 100 Hz signal is input to the 100 Hz counter (T100CR) and the carry output of that counter is input to the 10 Hz counter (T10CR). The T10HzINT signal, which is the carry output (10 Hz) of the T100CR 100 Hz counter, also generates an interrupt request, setting bit 3 (Q10Hz) of interrupt request registers 3 (IRQ3) to "1".

If either T100CR or T10CR is written to, both are reset to "0". The write data used has no significance. For example, the "MOV T100CR,A" instruction is not dependent on the contents of the accumulator.

If T10CR is read, the contents of T100CR at that time are latched to the 4-bit latch. Therefore, the contents of T100CR at the time T10CR is read can be read correctly.

Chapter 9

Watchdog Timer (WDT)

9. Watchdog Timer (WDT)

9.1 Overview

The watchdog timer is a circuit to detect CPU malfunction. The WDT consists of a 9-bit watchdog timer counter (WTDC) counting the 256 Hz output of the TBC7 of the time base counter (TBC), and a watchdog timer control register (WDTCON) to start and clear WDTC.

9.2 Watchdog Timer Configuration

Figure 9-1 shows the configuration of the watchdog timer.

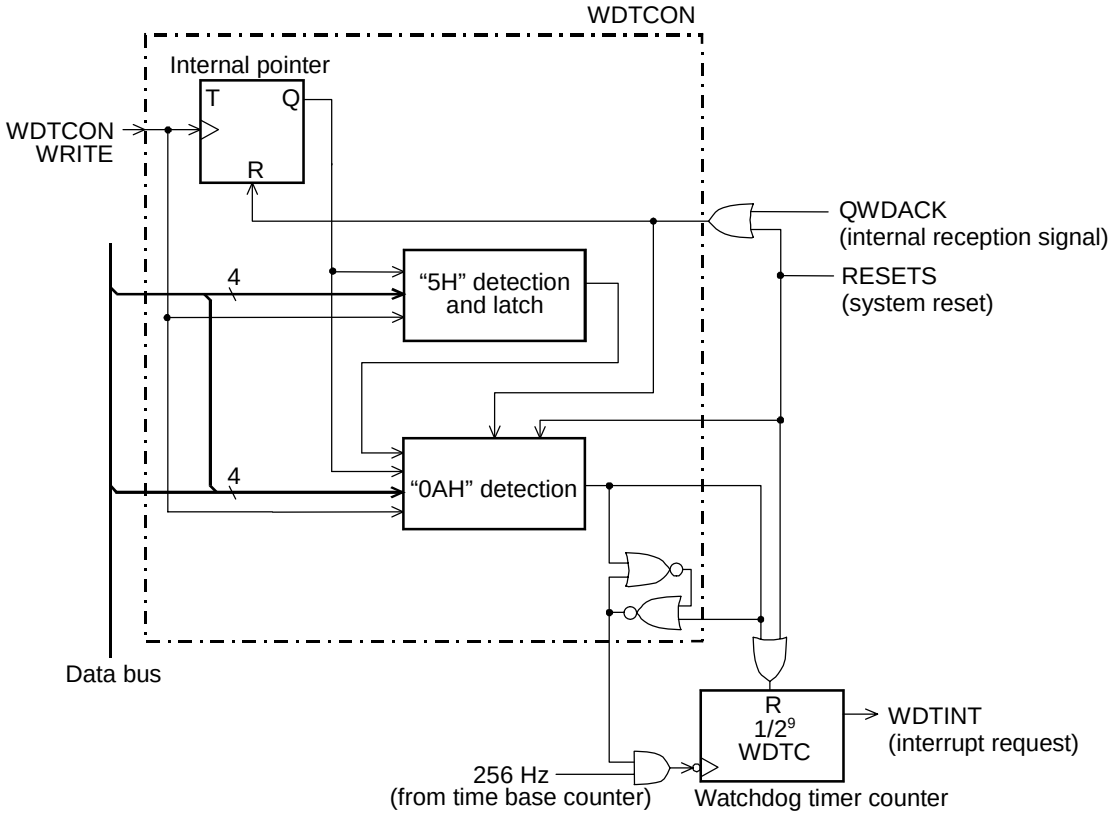


Figure 9-1 Watchdog Timer Configuration

9.3 Watchdog Timer Control Register (WDTCON)

The watchdog timer control register (WDTCON) is a 4-bit write-only special function register (SFR) used to start/clear the watchdog timer counter (WDTC).

	bit 3	bit 2	bit 1	bit 0
WDTCON (09FH) (W)	d3	d2	d1	d0

9.4 Watchdog Timer Operation

At system reset, WDTC (watchdog timer counter) stops counting.

WDTC begins counting by writing "5H" to WDTCON (watchdog timer control register) while the internal pointer is "0", and then writing "0AH" (while the internal pointer is "1").

The internal pointer is cleared to "0" at system reset or when WDTC overflows, and toggles every time a write operation to WDTCON is performed.

After WDTC is activated, WDTC is cleared by writing "5H" to WDTCON while the internal pointer is "0", and then writing "0AH" while the internal pointer is "1". When WDTC overflows (1FFH→000H), a watchdog timer interrupt request (WDTINT) is generated. WDTINT cannot be disabled by the software (non-maskable interrupt) and has the highest level of interrupt priority.

The WDTC overflow cycle (T) is given by:

$$T = \frac{128 \times 512}{32768\ (Hz)} = 2\ s$$

The minus deviation (t) of the WDTC overflow cycle is given by:

$$t = \frac{128}{32768\ (Hz)} = \text{approximately } 3.9\ ms$$

Therefore, the WDTC clear cycle (Ct) can be computed as follows.

$$Ct = T - t = 2\ s - 3.9\ ms = 1.9961\ s$$

If 32.768 kHz is to be used as the low-speed clock, the software must be programmed to clear WDTC within 1.9961 s.

If the CPU malfunctions due to a power failure or other factor and the WDTC cannot be cleared normally, WDTC will overflow and WDTINT will be generated. Program the watchdog timer interrupt routine to handle recovery operations by returning to the normal routine.

Note:

The watchdog timer cannot detect all operating faults. If the CPU malfunctions but WDTC can still be cleared, a fault will not be detected.

Figure 9-2 shows a flowchart of watchdog timer processing.

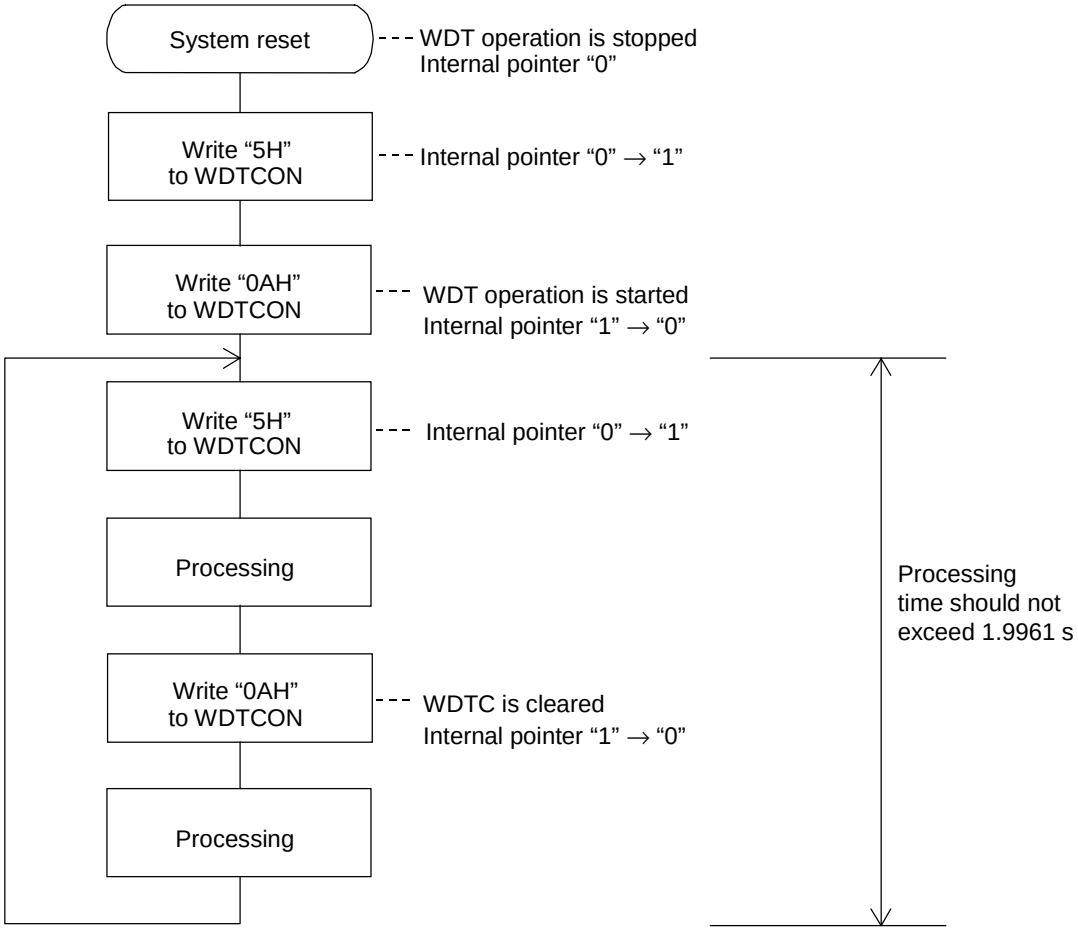


Figure 9-2 Watchdog Timer Processing Flowchart

Figure 9-3 shows the timing chart for watchdog timer operation.

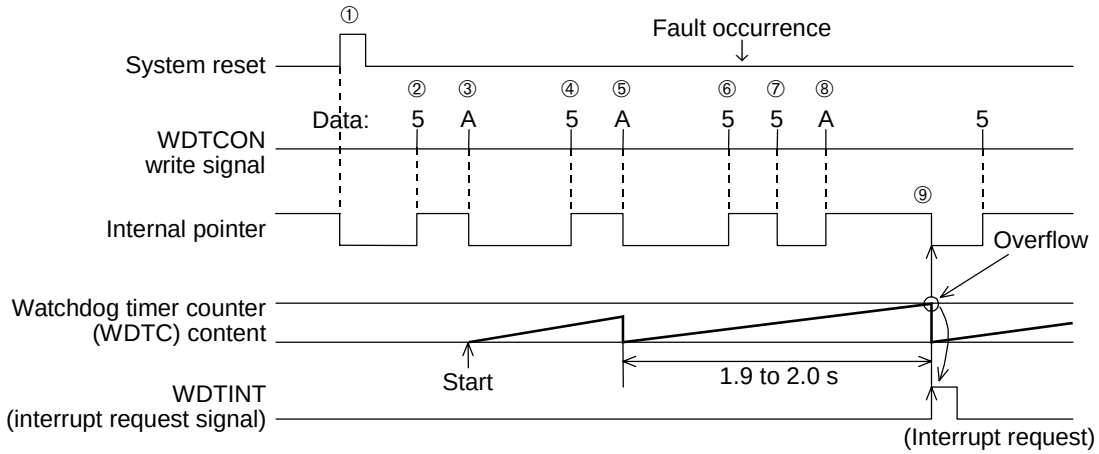


Figure 9-3 Watchdog Timer Operation Timing Chart

The watchdog timer operating sequence is listed below.

① System reset clears the internal pointer and WDTC.

② Write "5H" to WDTCON. (Internal pointer 0→1)

③ Write "0AH" to WDTCON to start WDTC. (Internal pointer 1→0)

④ Write "5H" to WDTCON. (Internal pointer 0→1)

⑤ Write "0AH" to WDTCON to clear WDTC. (Internal pointer 1→0)

⑥ Write "5H" to WDTCON. (Internal pointer 0→1)

⑦ After a fault occurs, "5H" is written to WDTCON but is not accepted since the internal pointer is "1". (Internal pointer 1→0)

⑧ "0AH" is written to WDTCON, but since the internal pointer is "0" and the write of "5H" in step ⑦ was not accepted, WDTC will not be cleared. (Internal pointer 0→1)

⑨ Because WDTC was not cleared, overflow of WDTC will generate the watchdog timer interrupt WDTINT. At this time, the internal pointer is cleared to "0".

Chapter 10

Ports (INPUT, I/O PORT)

10. Ports (INPUT, I/O PORT)

10.1 Overview

The ML63611 has built in it a 4-bit input-only port (Port 0) and 8-bit I/O ports (Port A, Port B, Port C, and Port E).

Port 0 has the three modes of input with pull-up resistor, input with pull-down resistor, and high impedance, one of which can be selected by software. Further, there are the functions of external 5 interrupt and of transfer to system reset due to simultaneous key depression.

The pairs Port A and Port B, and Port C and Port E have the input modes of input with pull-up resistor, input with pull-down resistor, and high impedance, or the output modes of P-channel open drain output, N-channel open drain output, CMOS output, and high impedance, and in each case, one of the modes can be selected by software. In addition, Port B has the external 0 interrupt function and Port C has the external 1 interrupt function.

Port E has the function of RC oscillator clock output for an A/D converter, the low-speed oscillation clock output function, the high-speed oscillation clock output function, and the external 2 interrupt function.

10.2 Ports List

The ports of the ML63611 are shown in Table 10-1.

Table 10-1 Ports List

Port	I/O	Function		Page
Port 0	I	Basic function:	Input-only port Simultaneous key depression reset function	10-2
		Secondary function:	External 5 interrupt (4-bit OR input)	
Port A	I/O	Basic function:	I/O port	10-7
		Secondary function:	None	
Port B	I/O	Basic function:	I/O port	10-11
		Secondary function:	External 0 interrupt (4-bit OR input)	
Port C	I/O	Basic function:	I/O port	10-19
		Secondary function:	Serial port receive data input pin (PC.0/RXD) Synchronous serial port clock I/O pin (for transmission)(PC.1/TXC) Synchronous serial port clock I/O pin (for reception)(PC.2/RXC) Serial port transmit data output pin (PC.3/TXD) External 1 interrupt (4-bit OR input)	
Port E	I/O	Basic function:	I/O port	10-27
		Secondary function:	Output pin of RC oscillation clock for an A/D converter (PE.0/MON) Low-speed oscillation clock output pin (PE.1/TBCCLK) High-speed oscillation clock output pin (PE.2/HSCLK) External 2 interrupt (PE.3/INT2)	

10.3 Port 0 (P0.0–P0.3)

The ML63611 has Port 0, a 4-bit input-only port.

Apart from the input port function, Port 0 has the functions of external interrupts and the function of transfer to system reset due to simultaneous key depression.

10.3.1 Port 0 Configuration

For the input port functions, there are the input modes of input with pull-up resistor, input with pull-down resistor, and high impedance, which can be selected by the Port 0 control registers 0/1 (P0CON0 and P0CON1). Among the external interrupt functions, the external 5 interrupt function has been assigned to this port which detects the OR result of the four input bits. It is possible to set the interrupt enabling and disabling for each bit using the Port 0 interrupt enable register (P0IE).

The function of transfer to system reset due to simultaneous key depression is the function of transferring to the system reset mode when there is simultaneous key depression of 2 bits, 3 bits, or 4 bits, as specified by the mask option, for a period of more than about 2 to 3 seconds. For details, see Section 3.2, "System Reset Mode (RST)".

Figure 10-1 shows the configuration of port 0.

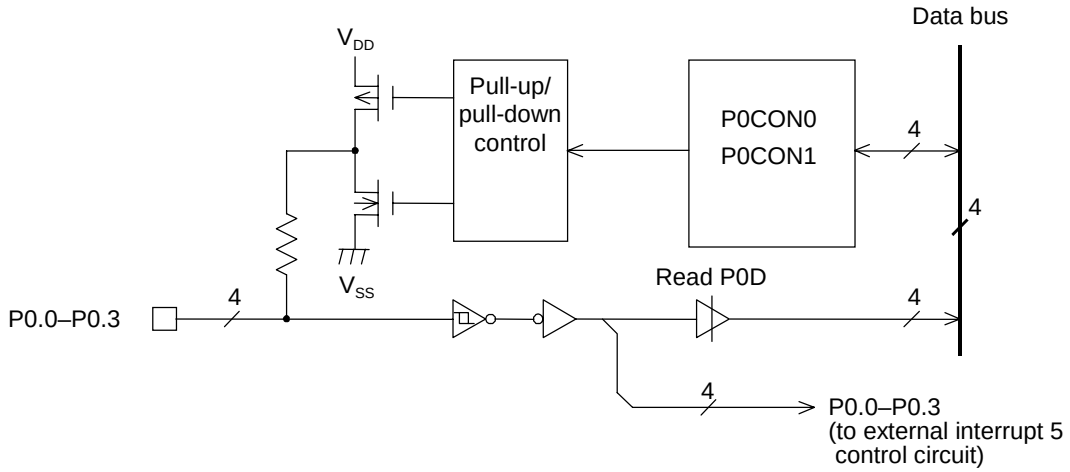


Figure 10-1 Input-Only Port (Port 0) Configuration

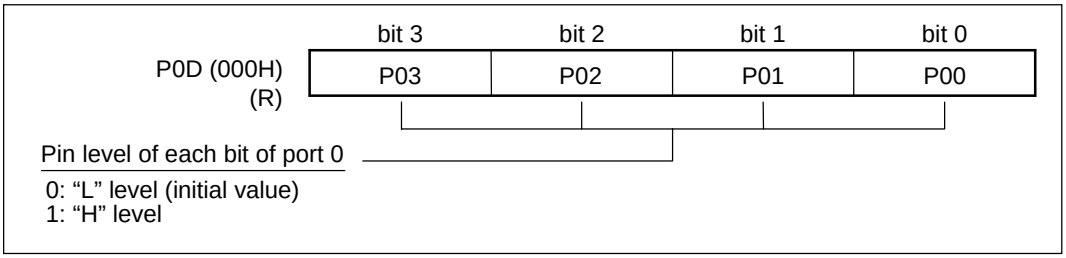
10.3.2 Port 0 Registers

The following registers are used to control port 0:

(1) Port 0 data register (P0D)
(2) Port 0 control registers 0/1 (P0CON0, P0CON1)
(3) Port 0 interrupt enable register (P0IE)

(1) Port 0 data register (P0D)

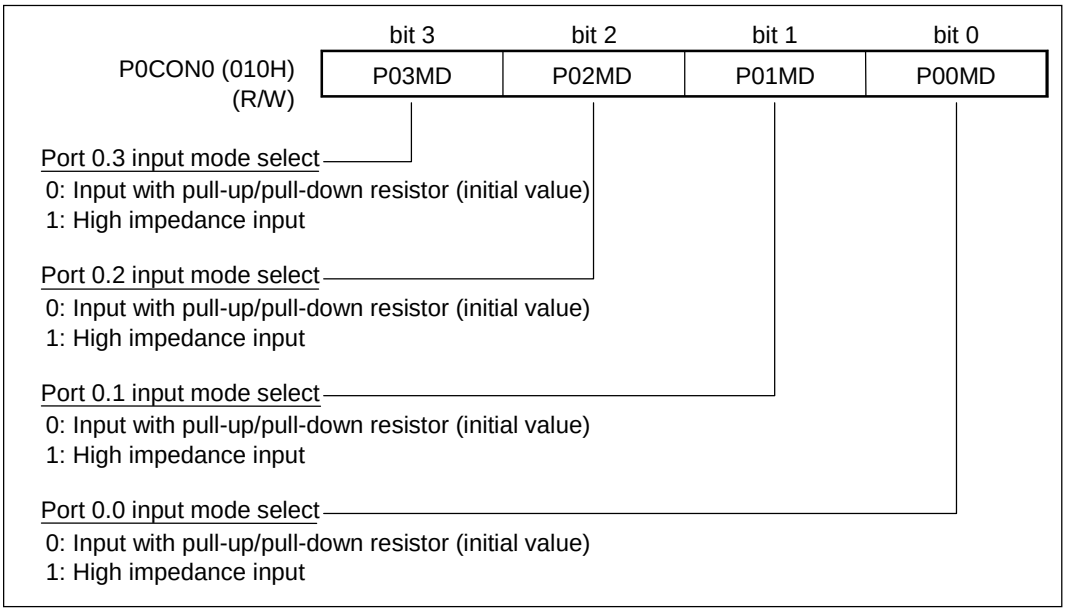
P0D is a 4-bit read-only special function register (SFR) used to read the pin level of each bit of port 0.

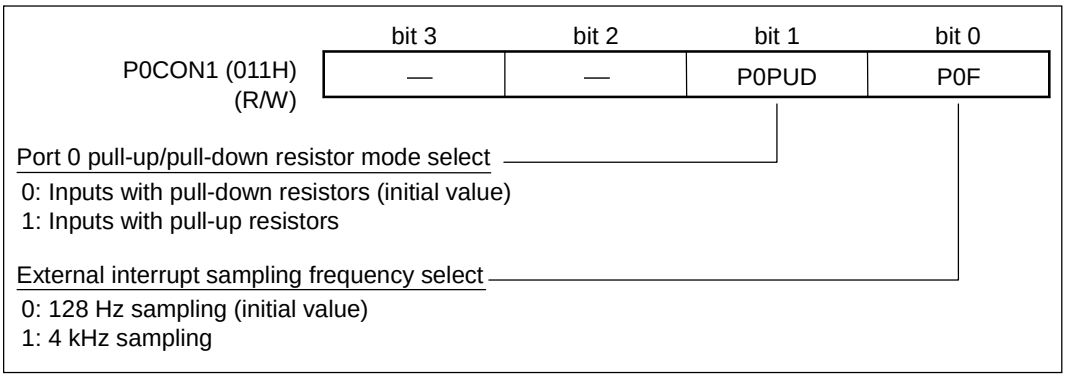


(2) Port 0 control registers 0/1 (P0CON0, P0CON1)

P0CON0 and P0CON1 are 4-bit special function registers (SFRs) that select pull-up or pull-down resistors and select the external interrupt sampling frequency of Port 0 secondary function.

At system reset, all the input modes for the four bits of P0CON0 are set to the input mode set by the mask option, and 128 Hz is selected as the external interrupt sampling frequency (however, all the bits enters the interrupt disabled state).





bit 1: P0PUD

This bit is used to select pull-up or pull-down resistors when any of the port 0 pins are selected by P0CON0 as an input with pull-up/pull-down resistor.

Setting P0PUD to "0" selects pull-down resistors, and setting to "1" selects pull-up resistors. Individual specification of pull-down or pull-up resistors for the pins of port 0.0 to 0.3 is not possible.

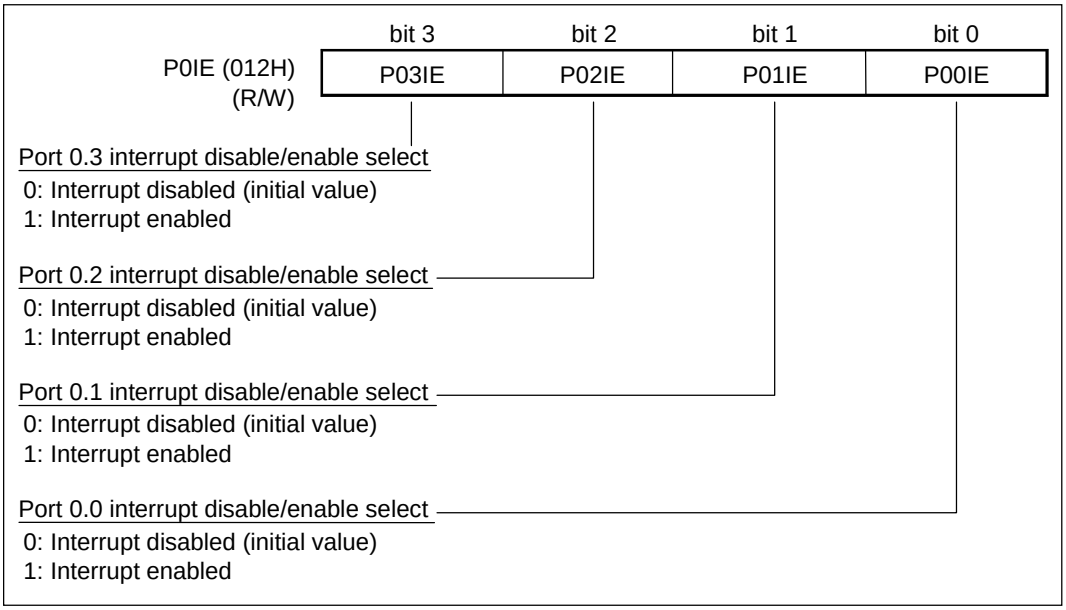
Note:

Use the mask option to specify the initial value at system reset of P0PUD.

(3) Port 0 interrupt enable register (P0IE)

P0IE is a 4-bit special function register (SFR) that enables/disables individual bits when port 0 is used as an external interrupt.

At system reset, all bits in the port interrupt enable register are cleared to "0" and port 0 is initialized to the interrupt disabled state.



10.3.3 Port 0 External Interrupt Function (External Interrupt 5)

Port 0 has external interrupt 5 allocated as secondary function. Individual bits can be enabled/disabled for external interrupt 5.

External interrupt generation for each input of port 0 is triggered by the falling edge of either the 128 Hz or 4 kHz sampling clock from the time base counter.

After the port level changes, interrupt request signal XI5INT is output and external interrupt 5 request flag (QXI5) is set. The maximum time delay from the change in port level until setting QXI5 is one cycle of the sampling clock (128 Hz or 4 kHZ).

Because the port 0 external interrupt 5 is set by a level change at any of the port 0 inputs, each bit of the port must be read to determine which bit of port 0 generated the interrupt.

The interrupt vector address for external interrupt 5 is 001EH.

Figure 10-2 shows an equivalent circuit of external interrupt 5 control.

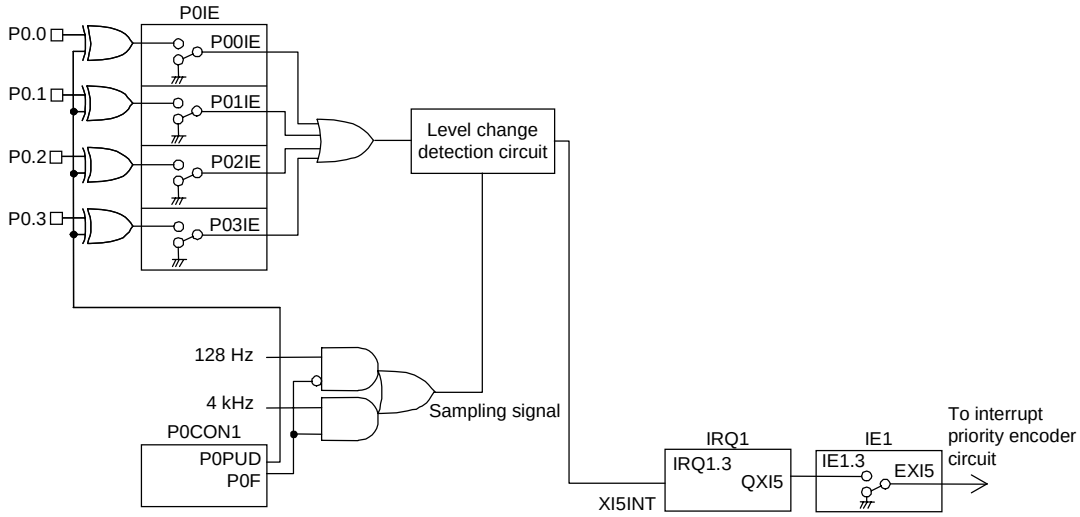


Figure 10-2 Equivalent Circuit of External Interrupt 5 Control

Figure 10-3 shows the timing for generation of external interrupt 5.

(a) P0PUD = "0" (initial value: inputs with pull-down resistors) setting

- When all P0.0 to P0.3 inputs are at a "L" level
  External interrupt 5 is generated when any port 0 input changes to a "H" level.

- When any of P0.0 to P0.3 inputs is at a "H" level
  External interrupt 5 is generated when all the port 0 inputs change to a "L" level.

(b) P0PUD = "1" (inputs with pull-up resistors) setting

- When all P0.0 to P0.3 inputs are at a "H" level
  External interrupt 5 is generated when any port 0 input changes to a "L" level.

- When any of P0.0 to P0.3 inputs is at a "L" level
  External interrupt 5 is generated when all the port 0 inputs change to a "H" level.

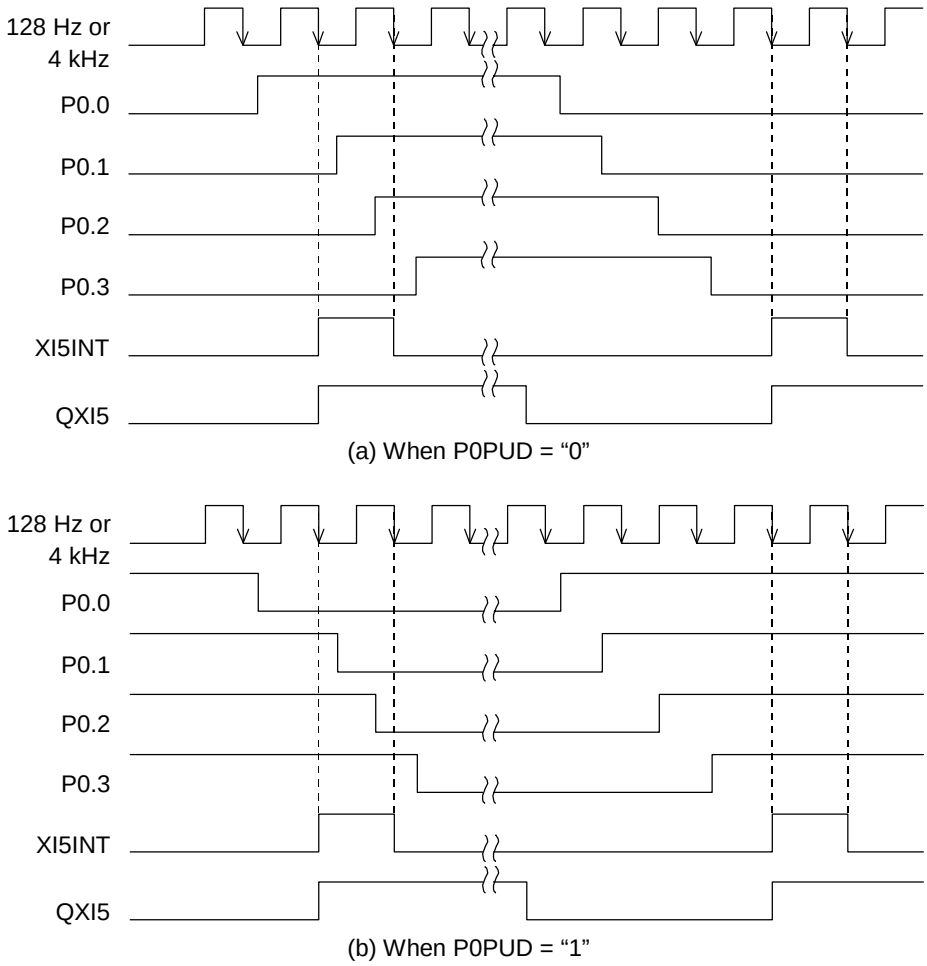


Figure 10-3 Interrupt Generation Timing of External Interrupt 5

10.4 Port A (PA.0–PA.3)

The ML63611 has Port A, a 4-bit input/output port.

10.4.1 Port A Configuration

The circuit configuration for port A is shown in Figure 10-4.

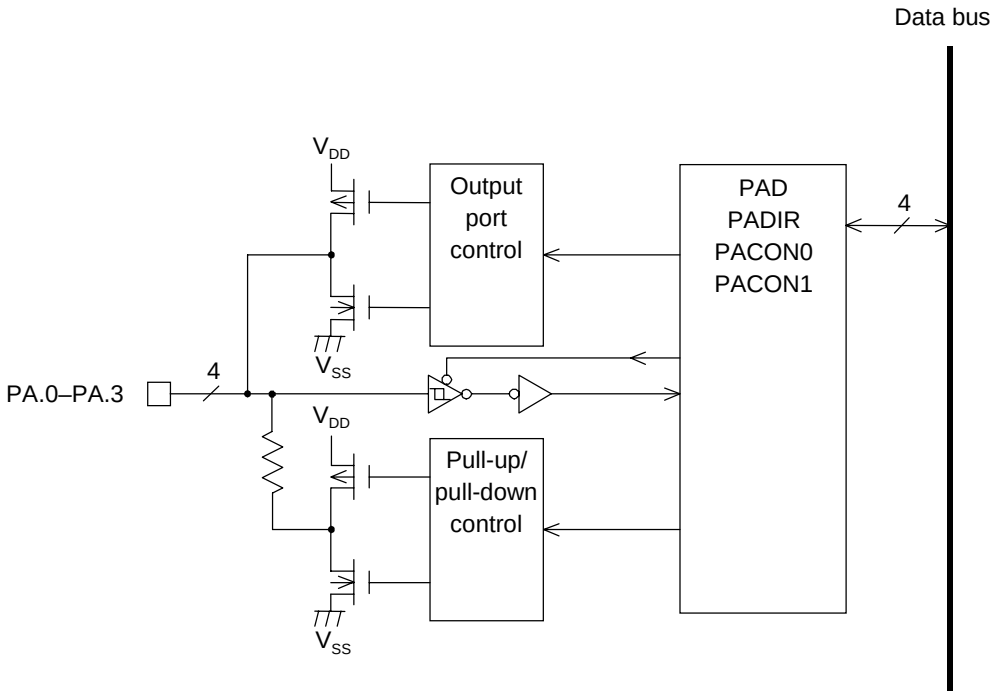


Figure 10-4 Input/Output Port A Configuration

10.4.2 Port A Registers

(1) Port A data register (PAD)

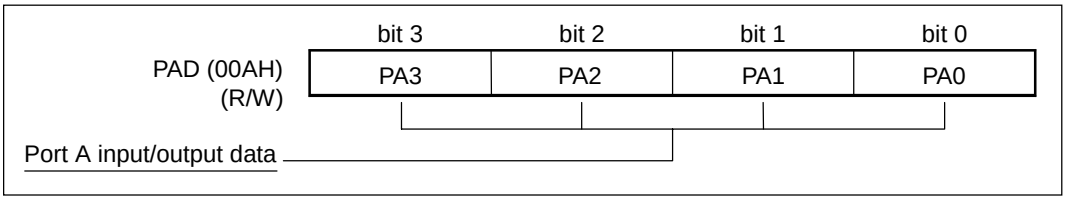
PAD is a 4-bit special function register (SFR) used to set the output values for the port.

When a bit in the port direction register (PADIR) is set to "1" to select the output mode, the content of the corresponding bit in the port data register (PAD) is output to the port (port A).

When a bit in the port data register (PAD) is read with the corresponding port direction register bit set to output, the value of the bit in the port data register is read.

When a bit in the port data register (PAD) is read with the corresponding port direction register bit set to "0" (input mode), the level of the corresponding pin is read.

- Port A



At system reset all bits in the port A data register are set to "0". When data is written to the port A data register, the actual pin change timing is at the rising edge of the system clock for state 2 of the write instruction.

Figure 10-5 shows port A change timing.

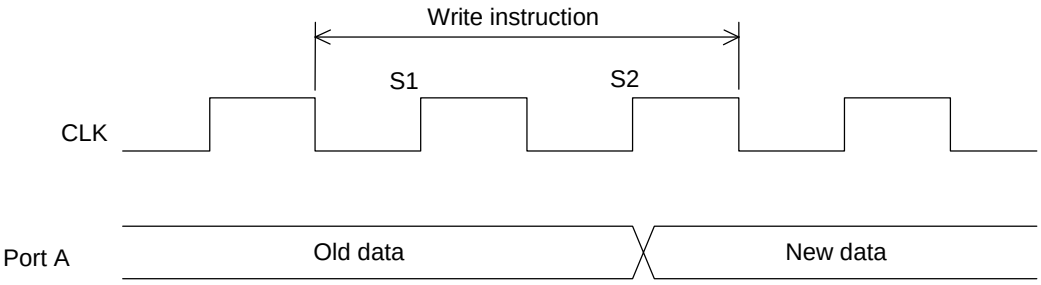


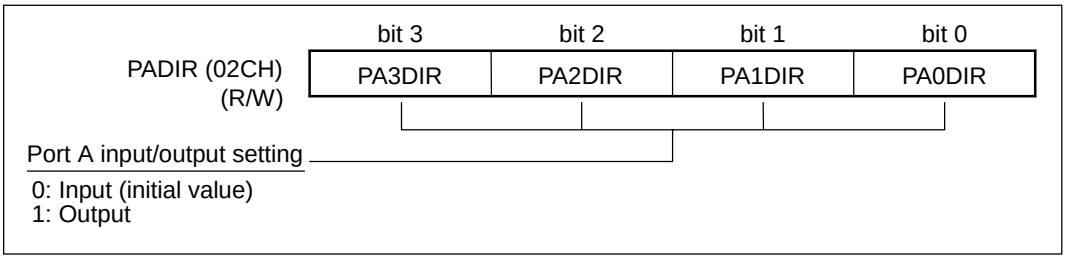
Figure 10-5 Port A Change Timing

(2) Port A direction register (PADIR)

PADIR is a 4-bit special function register (SFR) which specifies the port input/output direction for each bit. Pins corresponding to PADIR bits set to "0" are input, and those corresponding to bits set to "1" are output.

At system reset all bits in PADIR are set to "0", and port A is initialized to input mode.

- Port A



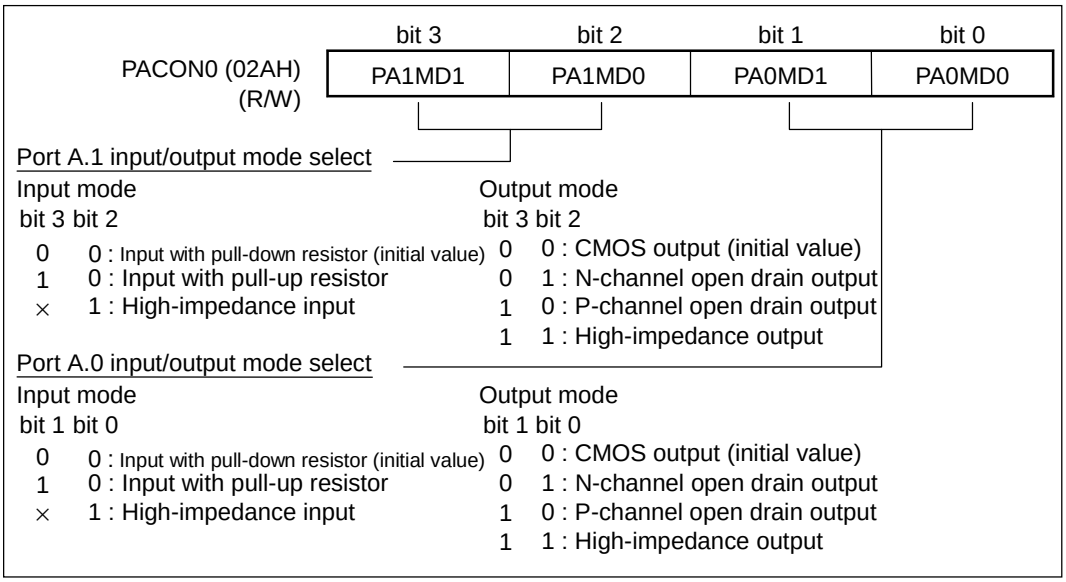
(3) Port A control registers 0/1 (PACON0, PACON1)

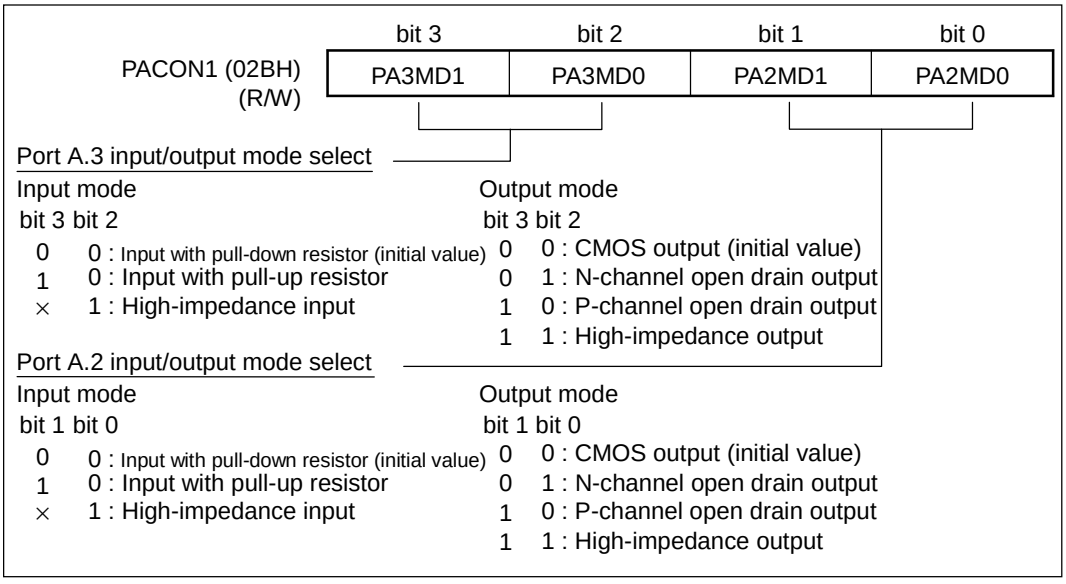
PACON0 and PACON1 are 4-bit special function registers (SFRs) used to select port input/output mode.

The input mode may be pull-down resistor input, pull-up resistor input or high-impedance input.

The output mode may be CMOS output, N-channel open drain output, P-channel open drain output or high-impedance output.

- Port A





10.5 Port B (PB.0–PB.3)

The ML63611 has Port B, a 4-bit input/output port.

10.5.1 Port B Configuration

As its I/O port functions, port B has the input mode that can be pull-up resistor input, pull-down resistor input, or high-impedance input and the output mode that can be P-channel open drain output, N-channel open drain output, CMOS output, or high-impedance output. These modes are selected by the port B control registers 0/1 (PBCON0, PBCON1).

An external 0 interrupt is assigned as an external interrupt function. External 0 interrupts are detected through a 4-bit OR input. Enabling or disabling interrupts can be specified for each bit by the port B interrupt enable register (PBIE).

The circuit configuration for port B is shown in Figure 10-6.

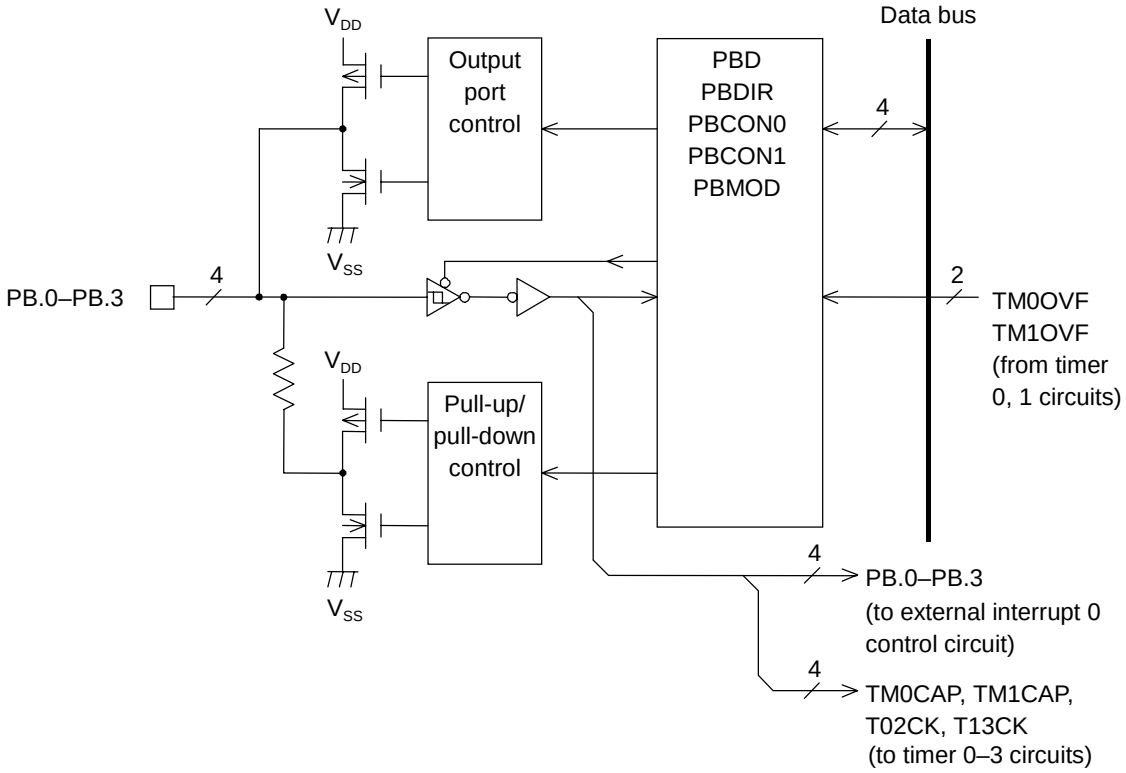


Figure 10-6 Input /Output Port (Port B) Configuration

10.5.2 Port B Registers

The following registers are used to control port B:

(1) Port B data register (PBD)
(2) Port B direction register (PBDIR)
(3) Port B control registers 0/1 (PBCON0, PBCON1)
(4) Port B mode register (PBMOD)
(5) Port B interrupt enable register (PBIE)

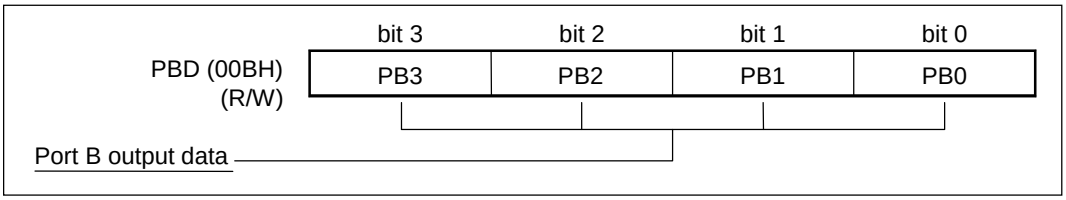
(1) Port B data register (PBD)

PBD is a 4-bit special function register used to set the output values for port B.

When a bit in the port B direction register (PBDIR) is set to "1" to select the output mode, the content of the corresponding bit in the port B data register is output to the port B.

When a bit in the port B data register is read with the corresponding PBDIR bit set to output, the value of the bit in the port B data register is read.

When a bit in the port B data register is read with the corresponding PBDIR bit set to "0" (input mode), the level of the corresponding pin of port B is read.



At system reset all bits in the port B data register (PBD) are set to "0". When data is written to the port B data register, the actual pin change timing is at the rising edge of the system clock for state 2 of the write instruction.

Figure 10-7 indicates port B change timing.

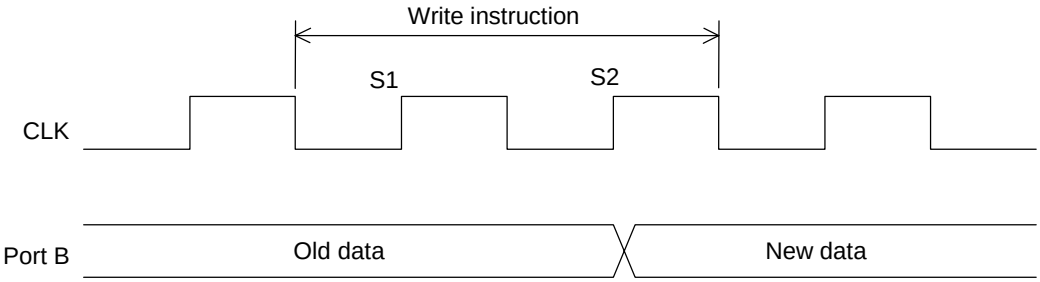
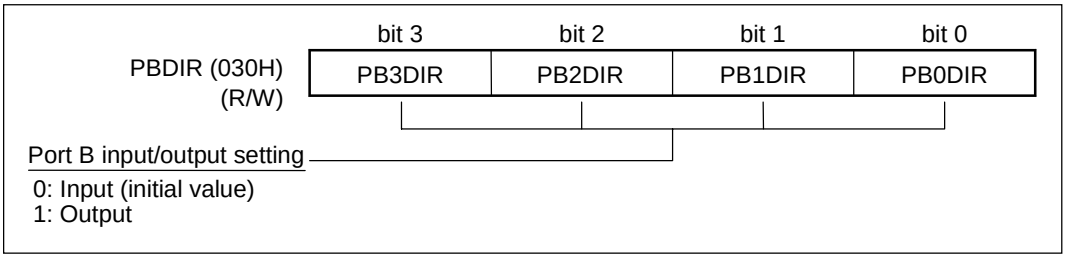


Figure 10-7 Port B Change Timing

(2) Port B direction register (PBDIR)

PBDIR is a 4-bit special function register (SFR) which specifies the port input/output direction for each bit. Pins corresponding to PBDIR bits set to "0" are input, and those corresponding to PBDIR bits set to "1" are output.

At system reset all bits in the port B direction register are reset to "0", and port B is initialized to input mode.



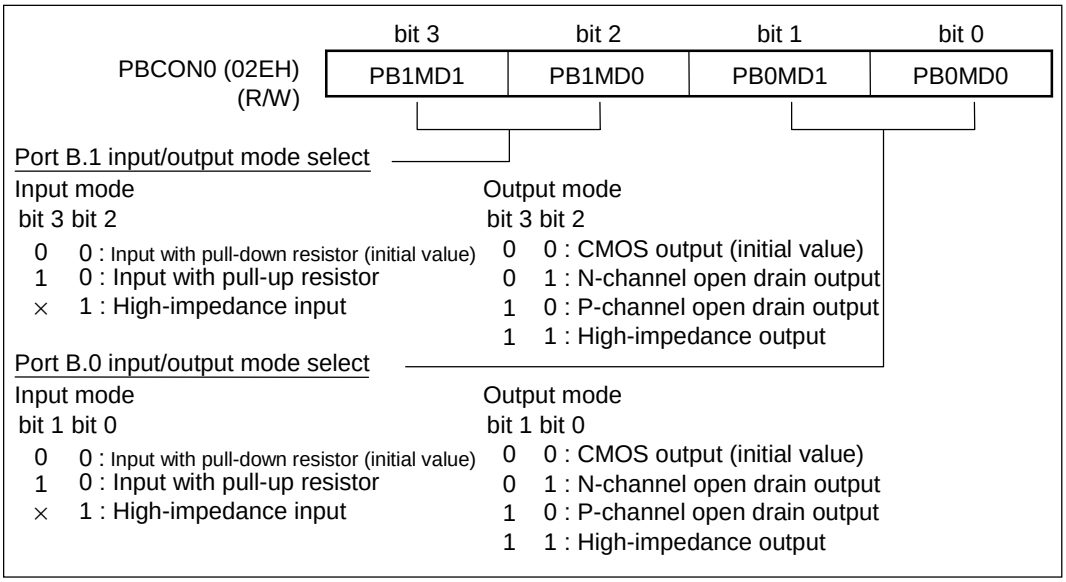
(3) Port B control registers 0/1 (PBCON0, PBCON1)

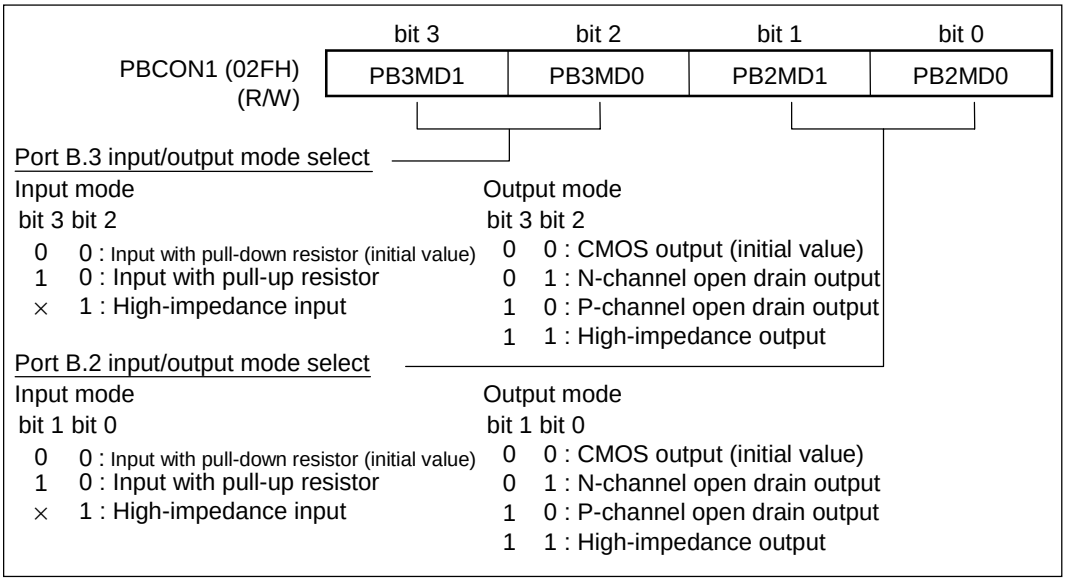
PBCON0 and PBCON1 are 4-bit special function registers (SFRs) used to select port input/output mode.

The input mode can be pull-down resistor input, pull-up resistor input or high-impedance input.

The output mode can be CMOS output, N-channel open drain output, P-channel open drain output or high-impedance output.

At system reset all bits in PBCON0 and PBCON1 are set to "0", and port B is initialized to pull-down resistor input mode and CMOS output mode.





(4) Port B mode register (PBMOD)

PBMOD is a 4-bit special function register (SFR) used to select the sampling frequency when port B is used as an external interrupt. It is also used to select port B secondary functions other than external interrupt.

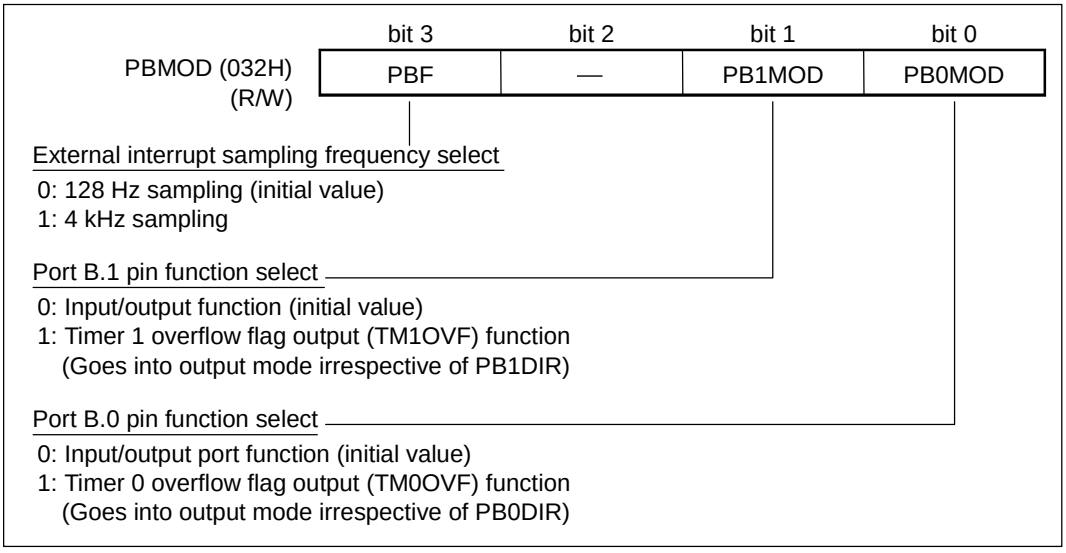
The external interrupt sampling frequency can be selected as either 128 Hz or 4 kHz.

At system reset all the valid bits in PBMOD are initialized to "0".

Port B secondary functions are indicated in Table 10-2.

Table 10-2 Port B Secondary Functions

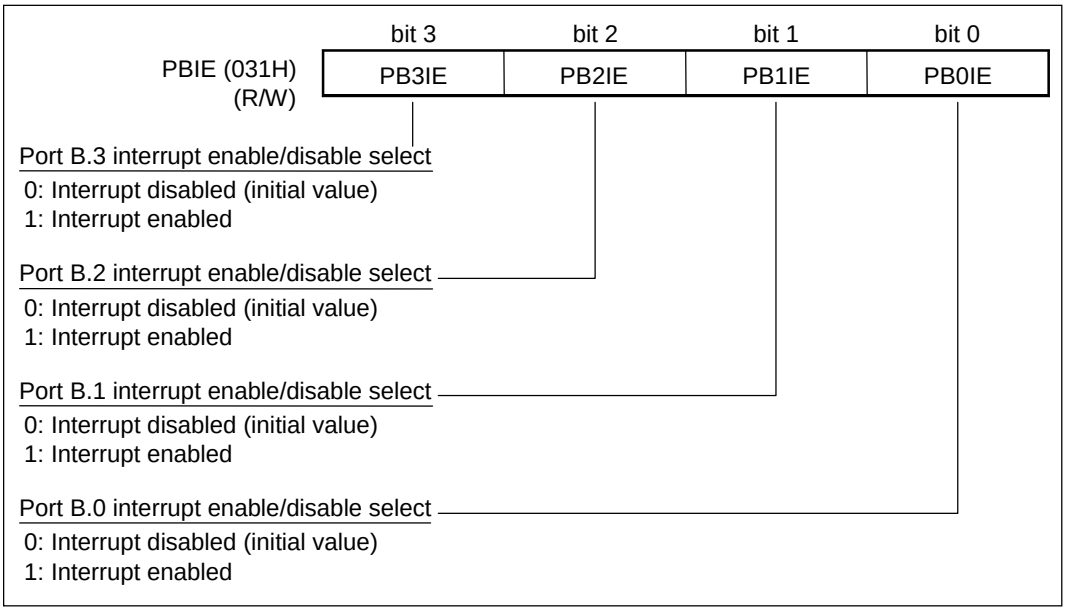
Port	Secondary function	Description
PB.0	TM0CAP	Timer 0 capture input
PB.1	TM1CAP	Timer 1 capture input
PB.2	T02CK	Timer 0, timer 2 external clock input
PB.3	T13CK	Timer 1, timer 3 external clock input
PB.0	TM0OVF	Timer 0 overflow flag output
PB.1	TM1OVF	Timer 1 overflow flag output
PB.0	INT0	External interrupt 0
PB.1		
PB.2		
PB.3		



(5) Port B interrupt enable register (PBIE)

PBIE is a 4-bit special function register (SFR) that enables/disables individual bits when port B is used as an external interrupt input.

At system reset, all bits in PBIE are cleared to "0" and port B is initialized to the interrupt disabled state.



10.5.3 Port B External Interrupt Function (External Interrupt 0)

Port B has external interrupt 0 allocated as secondary function. Individual bits can be enabled/disabled for external interrupt 0.

External interrupt generation for port B is triggered by the falling edge of the 128 Hz or 4 kHz time base counter, which is the sampling clock.

After the port level changes, the interrupt request signal (XI0INT) is output, and the interrupt request flag (QXI0) is set. The maximum delay for this sequence is one cycle of the sampling clock (128 Hz or 4 kHz).

Because the port B external interrupt is set by a level change at any of the port B inputs, each bit of the port must be read to determine which bit of port B generated the interrupt.

The interrupt start address for external interrupt 0 is 0014H.

Figure 10-8 shows the equivalent circuit for external interrupt 0 control.

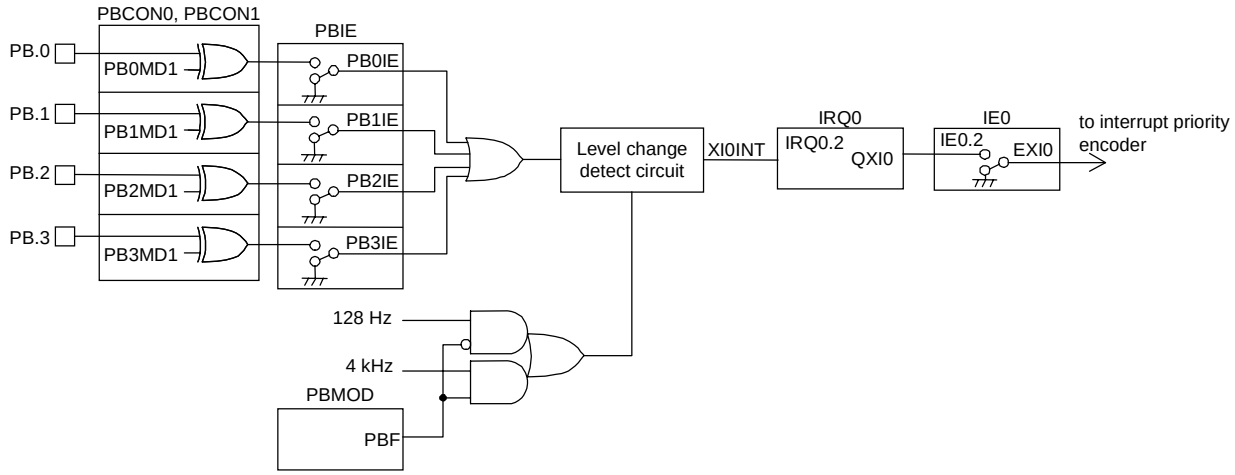


Figure 10-8 External Interrupt 0 Control Equivalent Circuit

Figure 10-9 shows the external interrupt 0 generation timing.

(a) PB0MD1 to PB3MD1 = "0" (initial value: inputs with pull-down resistors or high impedance input) setting

- When all PB.0 to PB.3 inputs are at a "L" level
  External interrupt 0 is generated when any port B input changes to a "H" level.
- When any of PB.0 to PB.3 inputs is at a "H" level
  External interrupt 0 is generated when all the port B inputs change to a "L" level.

(b) PB0MD1 and PB1MD1 = "0" and PB2MD1 and PB3MD1 = "1" (PB.0 and PB.1 selected as inputs with pull-down resistors or high impedance input; PB.2 and PB.3 selected as inputs with pull-up resistors or high impedance input) setting

- When both PB.0 and PB.1 inputs are at a "L" level AND both PB.2 and PB.3 inputs are at a "H" level
  External interrupt 0 is generated when either PB.0 or PB.1 input changes to a "H" level (alternatively, when either PB.2 or PB.3 input changes to a "L" level).
- When either PB.0 or PB.1 input is at a "H" level OR either PB.2 or PB.3 input is at a "L" level
  External interrupt 0 is generated when both PB.0 and PB.1 inputs change to a "L" level AND both PB.2 and PB.3 inputs change to a "H" level.

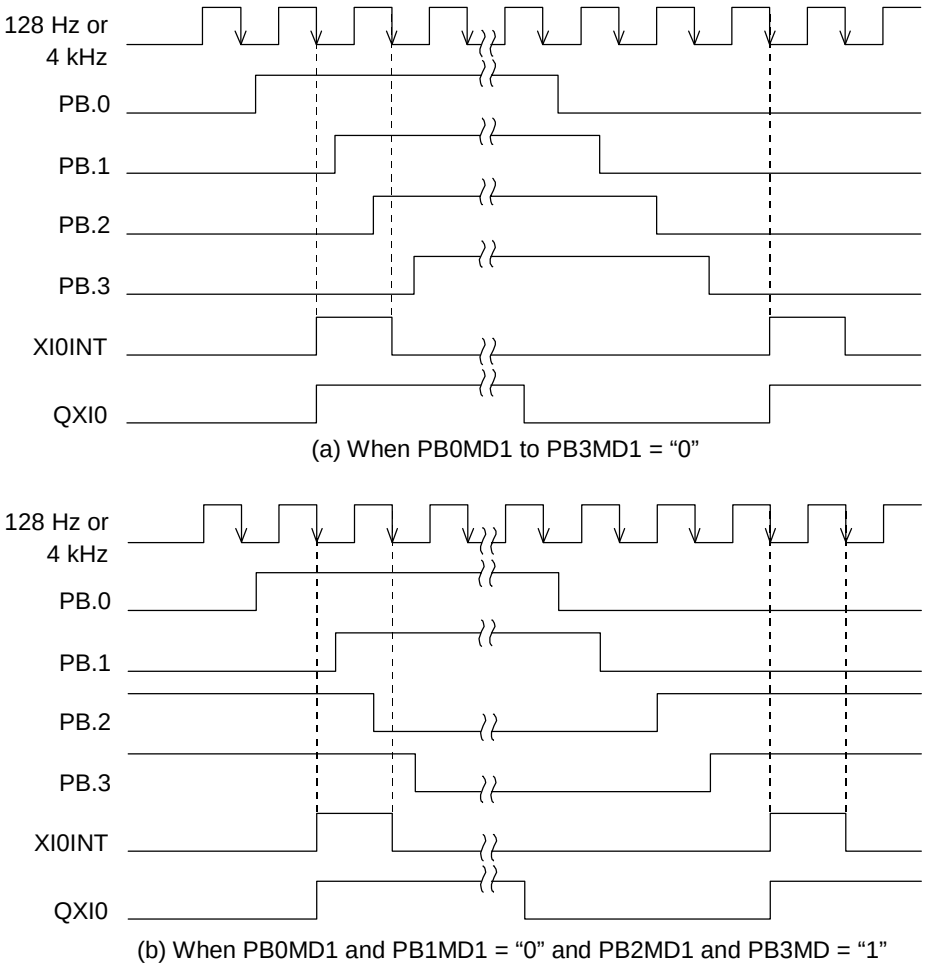


Figure 10-9 Interrupt Generation Timing of External Interrupt 0

10.6 Port C (PC.0–PC.3)

The ML63611 has Port C, a 4-bit input/output port.

10.6.1 Port C Configuration

The circuit configuration for port C is shown in Figure 10-10.

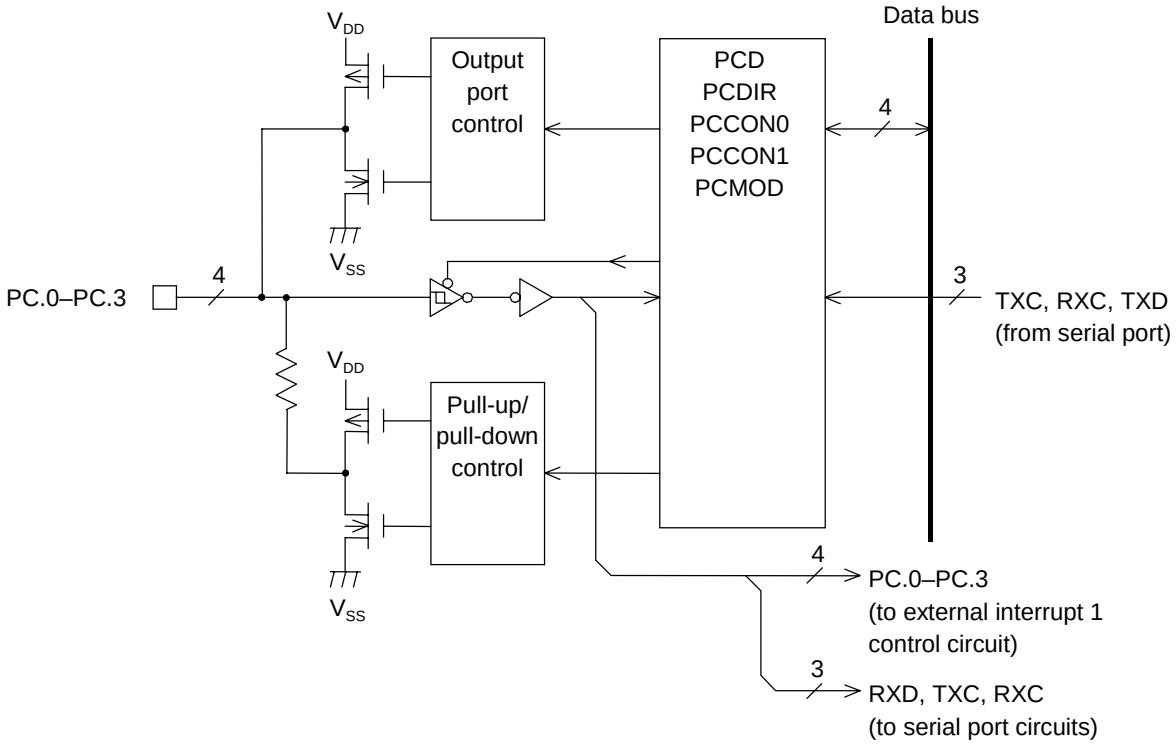


Figure 10-10 Input /Output Port (Port C) Configuration

10.6.2 Port C Registers

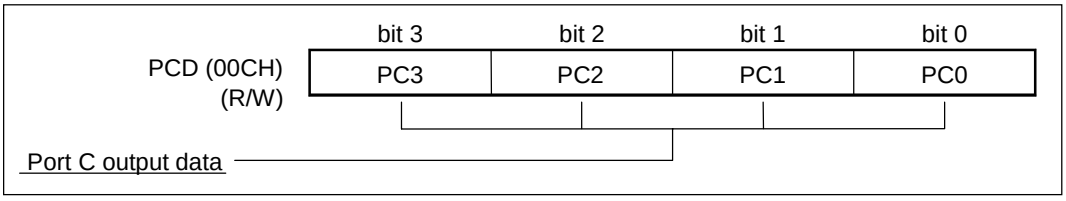
(1) Port C data register (PCD)

PCD is a 4-bit special function register used to set the output values for port C.

When a bit in the port C direction register (PCDIR) is set to "1" to select the output mode, the content of the corresponding bit in the port C data register is output to the port C.

When a bit in the port C data register is read with the corresponding PCDIR bit set to output, the value of the bit in the port C data register is read.

When a bit in the port C data register is read with the corresponding PCDIR bit set to "0" (input mode), the level of the corresponding pin of port C is read.



At system reset all bits in the port C data register (PCD) are set to "0". When data is written to the port C data register, the actual pin change timing is at the rising edge of the system clock for state 2 of the write instruction.

Figure 10-11 indicates port C change timing.

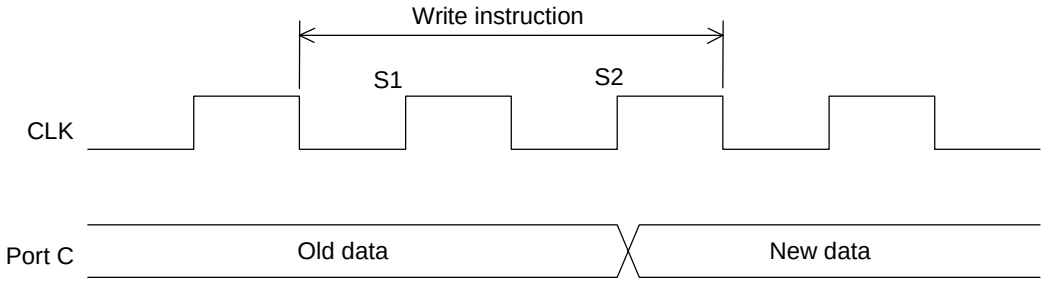
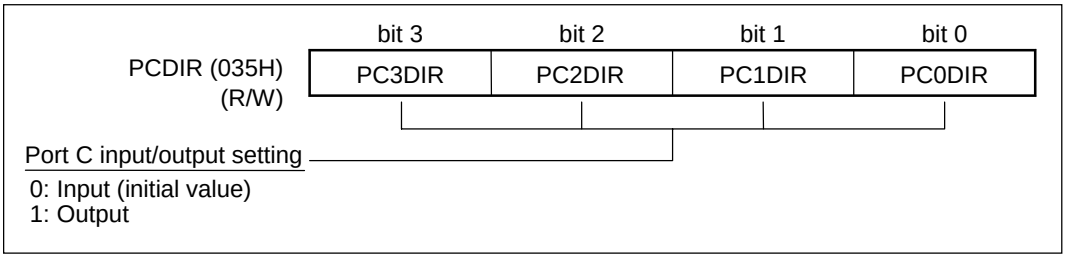


Figure 10-11 Port C Change Timing

(2) Port C direction register (PCDIR)

PCDIR is a 4-bit special function register (SFR) which specifies the port input/output direction for each bit. Pins corresponding to PCDIR bits set to "0" are input, and those corresponding to PCDIR bits set to "1" are output.

At system reset all bits in the port C direction register are reset to "0", and port C is initialized to input mode.



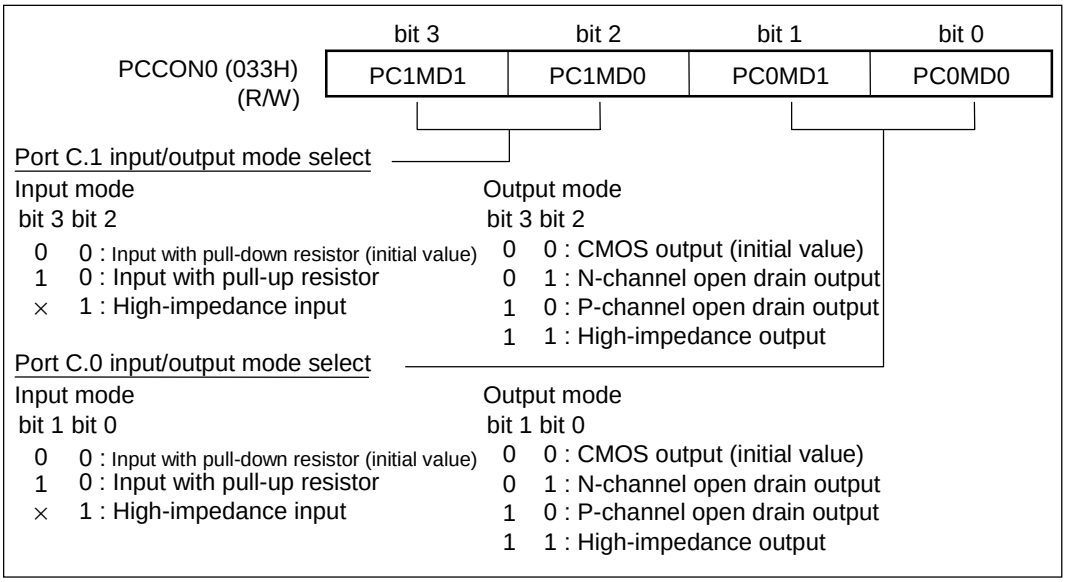
(3) Port C control registers 0/1 (PCCON0, PCCON1)

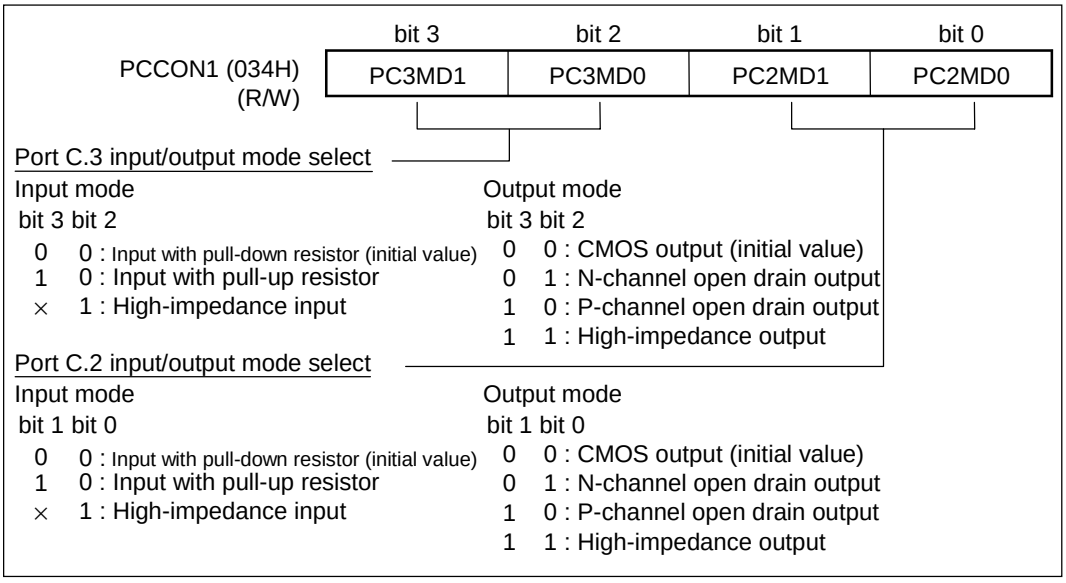
PCCON0 and PCCON1 are 4-bit special function registers (SFRs) used to select port input/output mode.

The input mode can be pull-down resistor input, pull-up resistor input or high-impedance input.

The output mode can be CMOS output, N-channel open drain output, P-channel open drain output or high-impedance output.

At system reset all bits in PCCON0 and PCCON1 are set to "0", and port C is initialized to pull-down resistor input mode and CMOS output mode.





(4) Port C mode registers 0/1 (PCMOD0, PCMOD1)

PCMOD0 and PCMOD1 are 4-bit special function registers (SFRs) used to select the sampling frequency when ports are used for external interrupt, and to select secondary functions other than external interrupt.

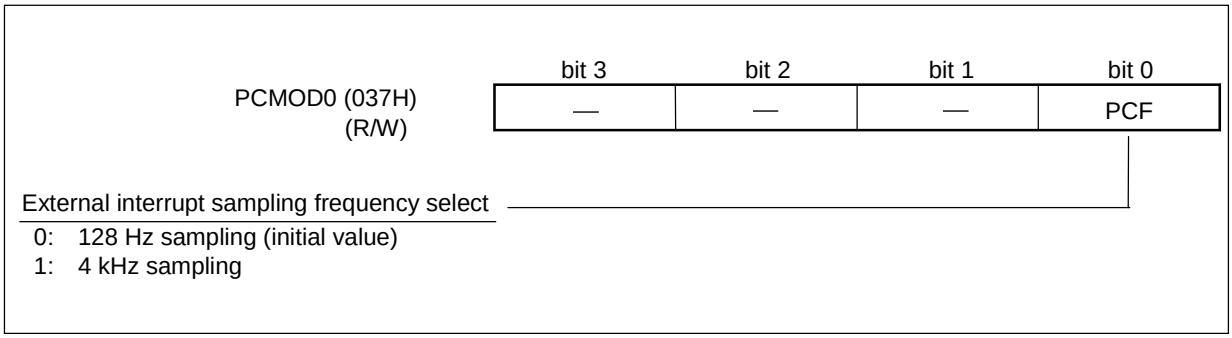
The external interrupt sampling frequency is either 128 Hz or 4 kHz.

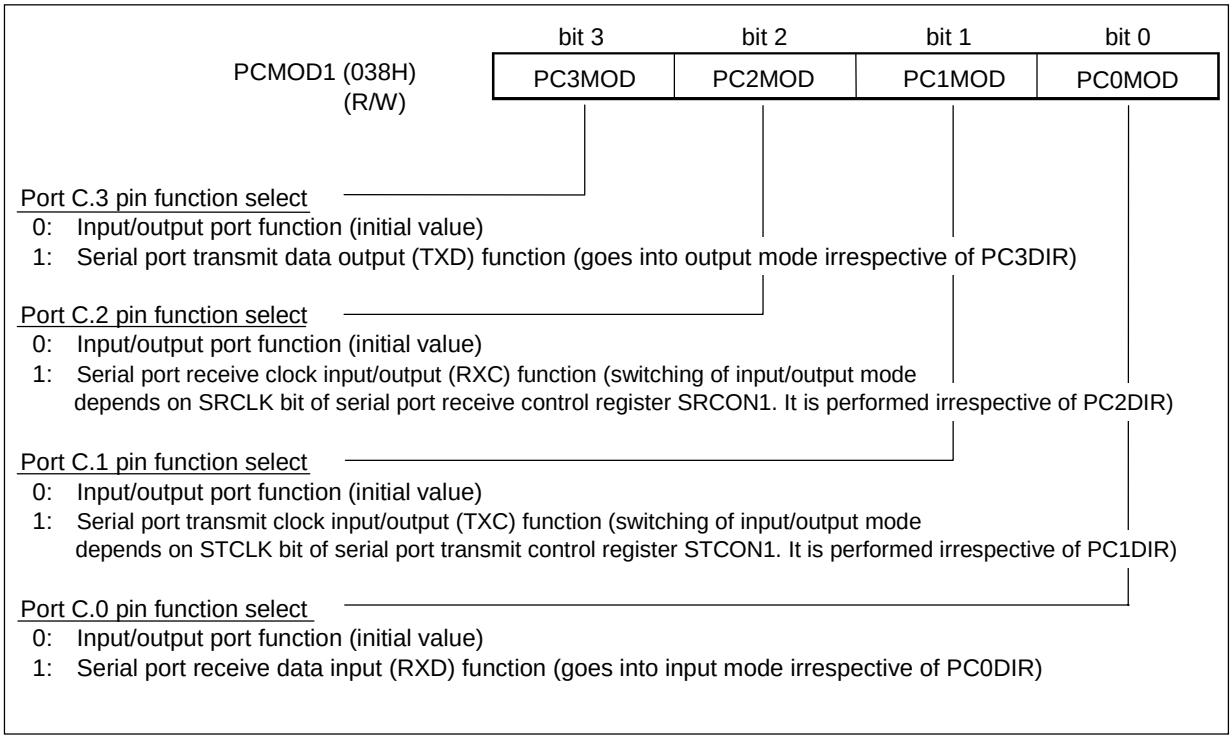
At system reset all lthe valid bits in the PCMOD0 and PCMOD1 are initialized to "0".

Port C secondary functions are indicated in Table 10-3.

Table 10-3 Port C Secondary Functions

Port	Secondary function	Description
PC.0	RXD	Serial port receive data input
PC.1	TXC	Serial port transmit clock I/O
PC.2	RXC	Serial port receive clock I/O
PC.3	TXD	Serial port transmit data output
PC.0	INT1	External interrupt 1
PC.1		
PC.2		
PC.3		

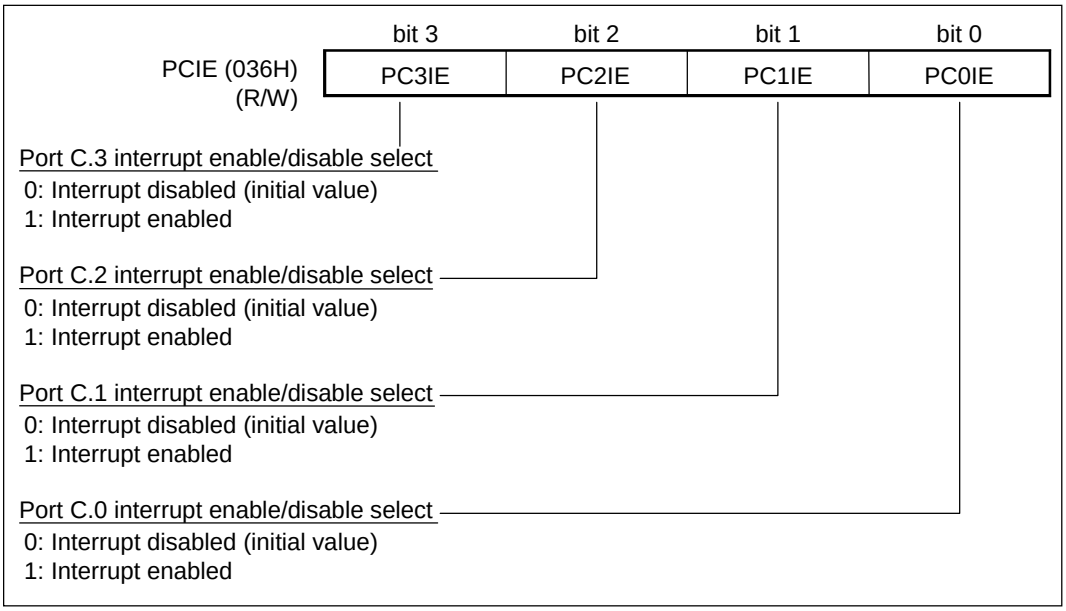

PCMOD0 (037H)
(R/W)
bit 3
bit 2
bit 1
bit 0
—
—
—
PCF
External interrupt sampling frequency select
0: 128 Hz sampling (initial value)
1: 4 kHz sampling


PCMOD1 (038H)
(R/W)
bit 3
bit 2
bit 1
bit 0
PC3MOD
PC2MOD
PC1MOD
PC0MOD
Port C.3 pin function select
0: Input/output port function (initial value)
1: Serial port transmit data output (TXD) function (goes into output mode irrespective of PC3DIR)
Port C.2 pin function select
0: Input/output port function (initial value)
1: Serial port receive clock input/output (RXC) function (switching of input/output mode depends on SRCLK bit of serial port receive control register SRCON1. It is performed irrespective of PC2DIR)
Port C.1 pin function select
0: Input/output port function (initial value)
1: Serial port transmit clock input/output (TXC) function (switching of input/output mode depends on STCLK bit of serial port transmit control register STCON1. It is performed irrespective of PC1DIR)
Port C.0 pin function select
0: Input/output port function (initial value)
1: Serial port receive data input (RXD) function (goes into input mode irrespective of PC0DIR)

(5) Port C interrupt enable register (PCIE)

PCIE is a 4-bit special function register (SFR) that enables/disables individual bits when port C is used as an external interrupt input.

At system reset, all bits in PCIE are cleared to "0" and port C is initialized to the interrupt disabled state.



10.6.3 Port C External Interrupt Function (External Interrupt 1)

Port C has external interrupt 1 allocated as secondary function. Individual bits can be enabled/disabled for external interrupt 1.

External interrupt generation for port C is triggered by the falling edge of the 128 Hz or 4 kHz time base counter, which is the sampling clock.

After the port level changes, the interrupt request signal (XI1INT) is output, and the interrupt request flag (QXI1) is set. The maximum delay for this sequence is one cycle of the sampling clock (128 Hz or 4 kHz).

Because the port C external interrupt 1 is set by a level change at any of the port C inputs, each bit of the port must be read to determine which bit of port C generated the interrupt.

The interrupt start address for external interrupt 1 is 0016H.

Figure 10-12 shows the equivalent circuit for external interrupt 1 control.

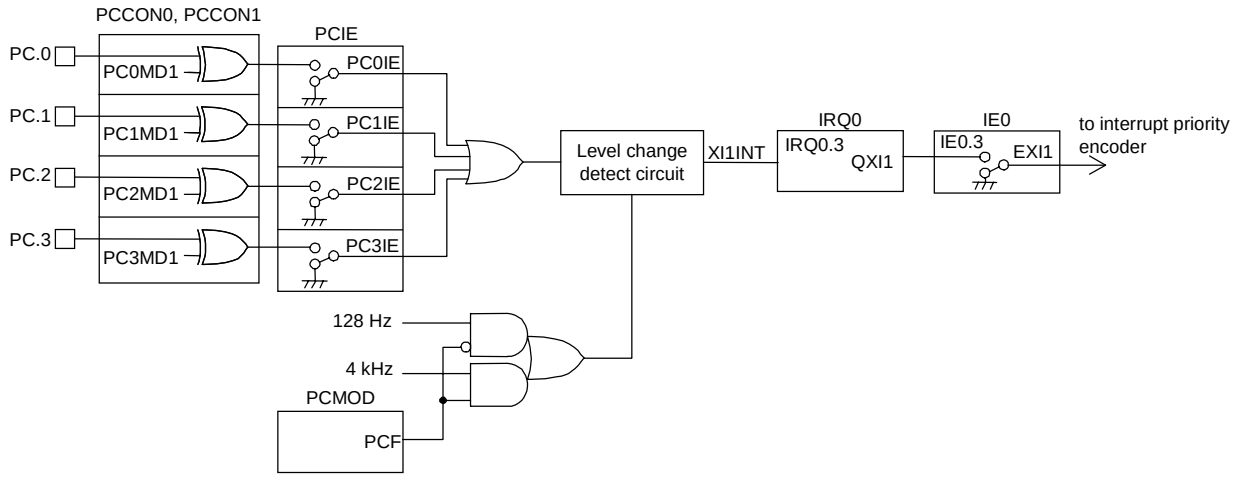


Figure 10-12 External Interrupt 1 Control Equivalent Circuit

Figure 10-13 shows the external interrupt 1 generation timing.

(a) PC0MD1 to PC3MD1 = "0" (initial value: inputs with pull-down resistors or high impedance input) setting

- When all PC.0 to PC.3 inputs are at a "L" level
  External interrupt 1 is generated when any port C input changes to a "H" level.

- When any of PC.0 to PC.3 inputs is at a "H" level
  External interrupt 1 is generated when all the port B inputs change to a "L" level.

(b) PC0MD1 and PC1MD1 = "0" and PC2MD1 and PC3MD1 = "1" (PC.0 and PC.1 selected as inputs with pull-down resistors or high impedance input; PC.2 and PC.3 selected as inputs with pull-up resistors or high impedance input) setting

- When both PC.0 and PC.1 inputs are at a "L" level AND both PC.2 and PC.3 inputs are at a "H" level
  External interrupt 1 is generated when either PC.0 or PC.1 input changes to a "H" level (alternatively, when either PC.2 or PC.3 input changes to a "L" level).

- When either PC.0 or PC.1 input is at a "H" level OR either PC.2 or PC.3 input is at a "L" level External interrupt 1 is generated when both PC.0 and PC.1 inputs change to a "L" level AND both PC.2 and PC.3 inputs change to a "H" level.

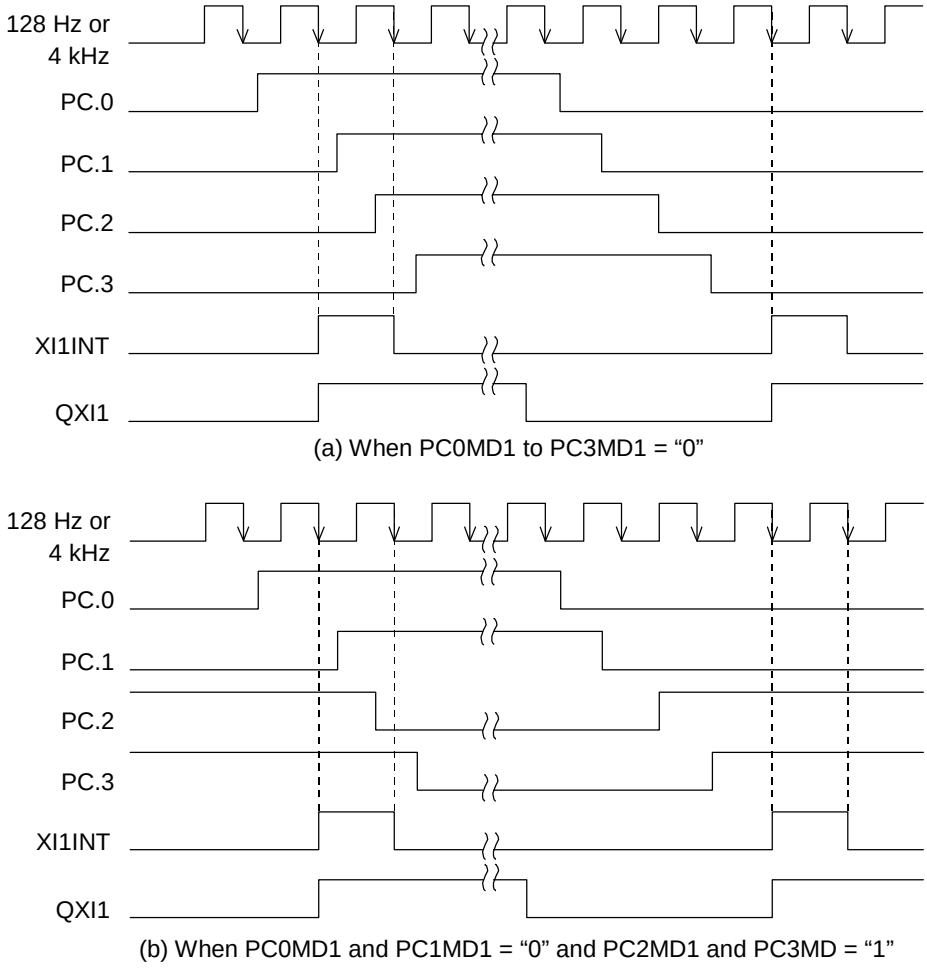


Figure 10-13 Interrupt Generation Timing of External Interrupt 1

10.7 Port E (PE.0–PE.3)

The ML63611 has Port E, a 4-bit input/output port.

Apart from the input/output port function, Port E has the functions of external interrupts, the function of RC oscillator clock output for an A/D converter, the low-speed clock output function, and the high-speed clock output function.

10.7.1 Port E Configuration

For the input port functions, there are the input modes of input with pull-up resistor, input with pull-down resistor, and high impedance, or the output modes of P-channel open drain output, N-channel open drain output, CMOS output, and high impedance, and one of the modes can be selected by the Port E control registers (PECON0 and PECON1).

The external 2 interrupt has been assigned to PE.3.

The function of RC oscillator clock output for an A/D converter, the low-speed clock output function, and the high-speed clock output function have been assigned to PE.0, PE.1, and PE.2, respectively. These functions can be selected by the Port E mode register (PEMOD).

The circuit configuration for port E is shown in Figure 10-14.

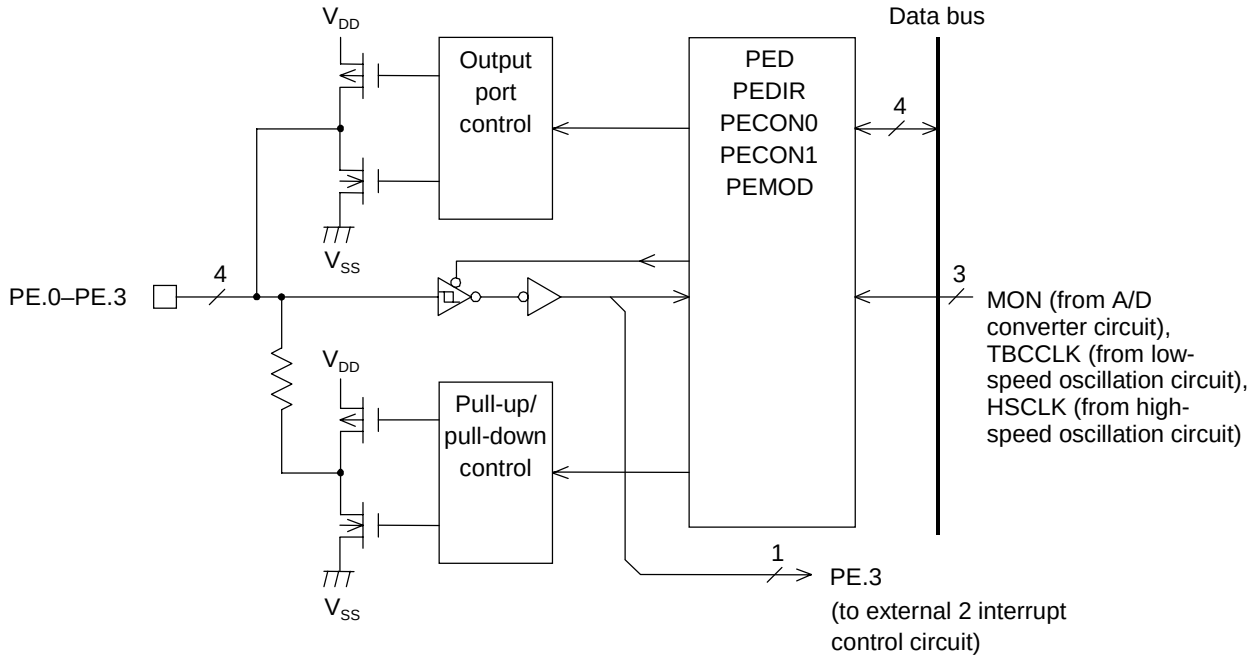


Figure 10-14 Input/Output Port (Port E) Configuration

10.7.2 Port E Registers

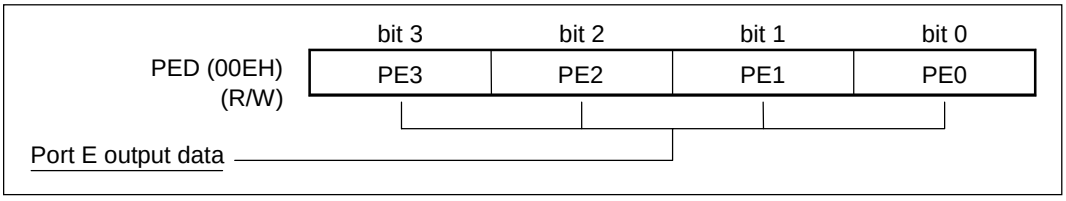
(1) Port E data register (PED)

PED is a 4-bit special function register used to set the output values for port E.

When a bit in the port E direction register (PEDIR) is set to "1" to select the output mode, the content of the corresponding bit in the port E data register is output to the port E.

When a bit in the port E data register is read with the corresponding PEDIR bit set to output, the value of the bit in the port E data register is read.

When a bit in the port E data register is read with the corresponding PEDIR bit set to "0" (input mode), the level of the corresponding pin of port E is read.



At system reset all bits in the port E data register (PED) are reset to "0". When data is written to the port E data register, the pin change timing is at the rising edge of the system clock for state 2 of the write instruction.

Figure 10-15 indicates port E change timing.

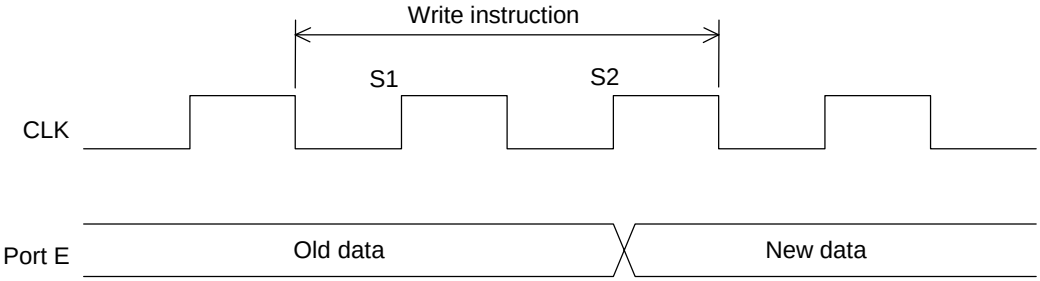
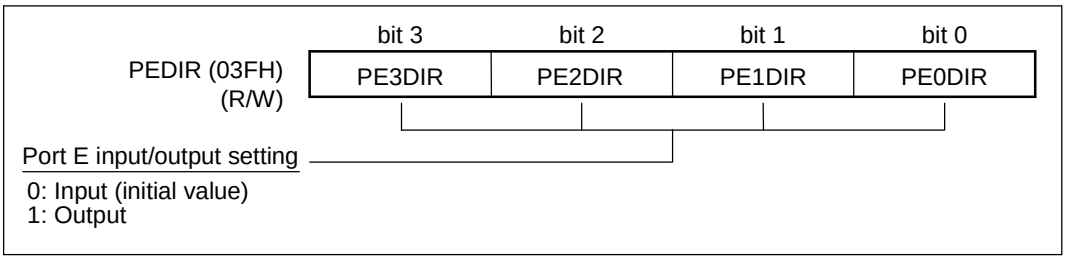


Figure 10-15 Port E Change Timing

(2) Port E direction register (PEDIR)

PEDIR is a 4-bit special function register (SFR) which specifies the port input/output direction for each bit. Pins corresponding to PEDIR bits set to "0" are input, and those corresponding to PEDIR bits set to "1" are output.

At system reset all bits in the port E direction register are set to "0", and port E is initialized to input mode.



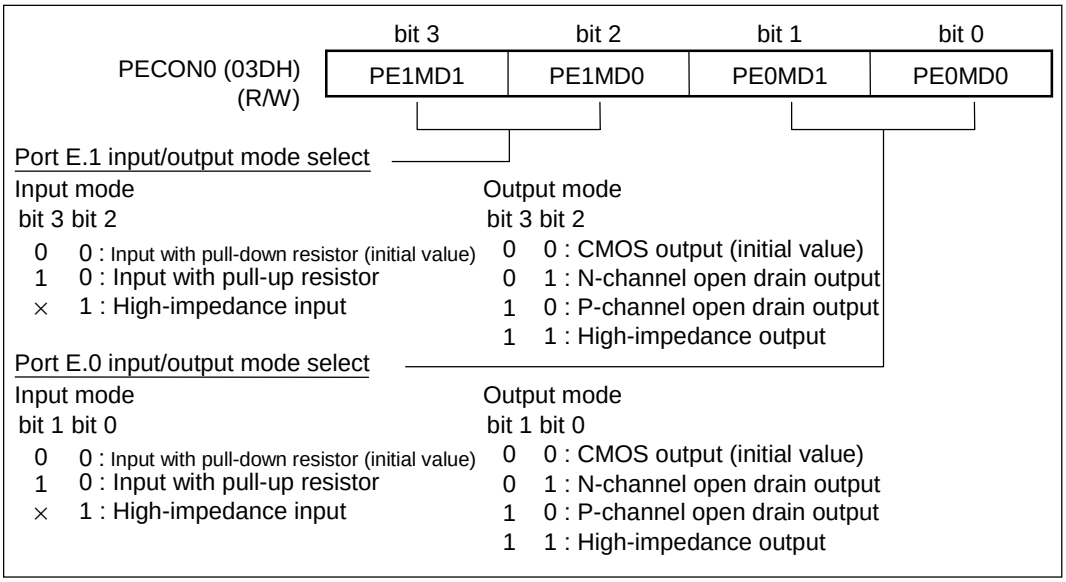
(3) Port E control registers 0/1 (PECON0, PECON1)

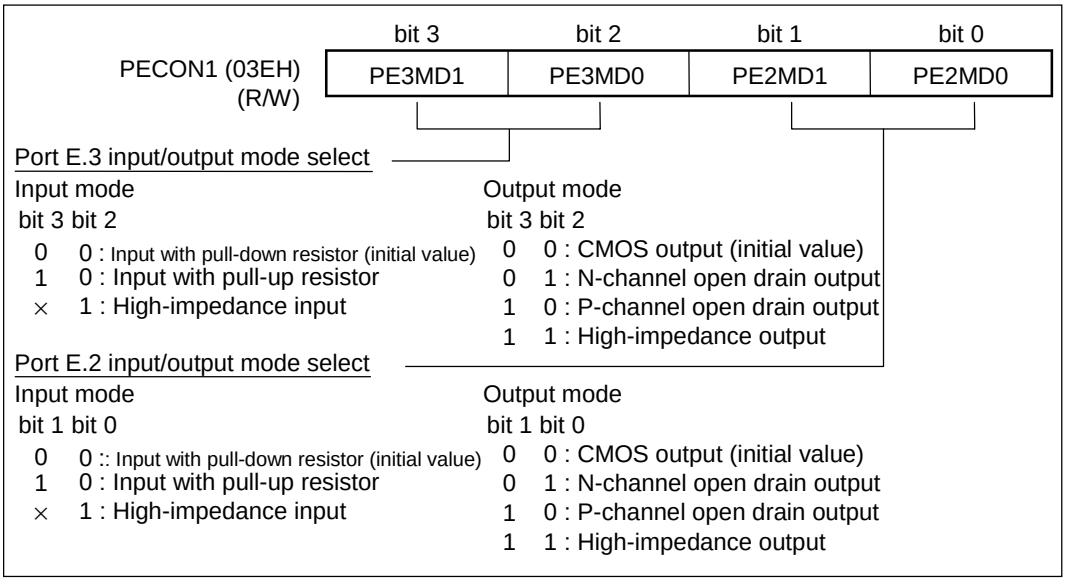
PECON0 and PECON1 are 4-bit special function registers (SFRs) used to select port input/output mode.

The input mode can be pull-down resistor input, pull-up resistor input or high-impedance input.

The output mode can be CMOS output, N-channel open drain output, P-channel open drain output or high-impedance output.

At system reset all bits in PECON0 and PECON1 are set to "0", and port E is initialized to pull-down resistor input mode and CMOS output mode.





(4) Port E mode register (PEMOD)

PEMOD is a 4-bit special function register (SFR) used to select the sampling frequency when PE.3 is used as an external interrupt. It is also used to select port E secondary functions other than external interrupt.

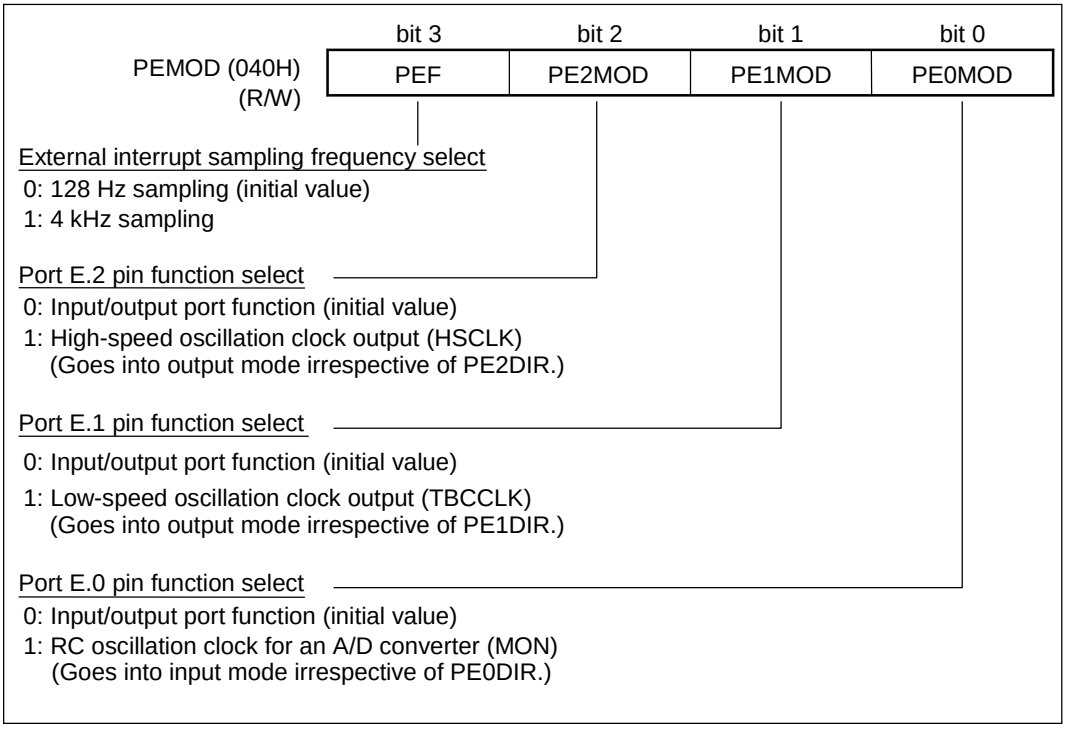
The external interrupt sampling frequency can be selected as either 128 Hz or 4 kHz.

At system reset all bits in PEMOD are initialized to "0".

Port E secondary functions are indicated in Table 10-4.

Table 10-4 Port E Secondary Functions

Port	Secondary function	Description
PE.0	MON	RC oscillation clock output for an A/D converter
PE.1	TBCCLK	Low-speed oscillation clock output
PE.2	HSCLK	High-speed oscillation clock output
PE.3	INT2	External interrupt 2



10.7.3 Port E.3 External Interrupt Function (External Interrupt 2)

Port E.3 has external interrupt 2 allocated as secondary function.

External interrupt generation for PE.3 is triggered by the falling edge of the 128 Hz or 4 kHz time base counter, which is the sampling clock.

After the port level changes, the interrupt request signal (XI2INT) is output, and the interrupt request flag (QXI2) is set. The maximum delay for this sequence is one cycle of the sampling clock (128 Hz or 4 kHz).

The interrupt start address for external interrupt 2 is 0018H.

Figure 10-16 shows the equivalent circuit for external interrupt 2 control.

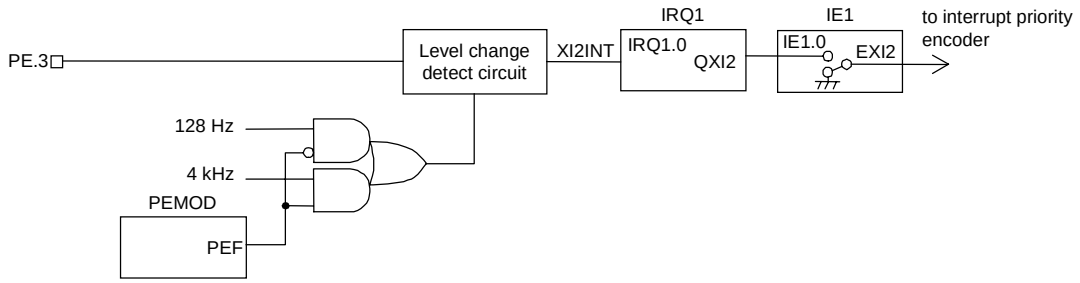


Figure 10-16 External Interrupt 2 Control Equivalent Circuit

Figure 10-17 shows the external interrupt 2 generation timing.

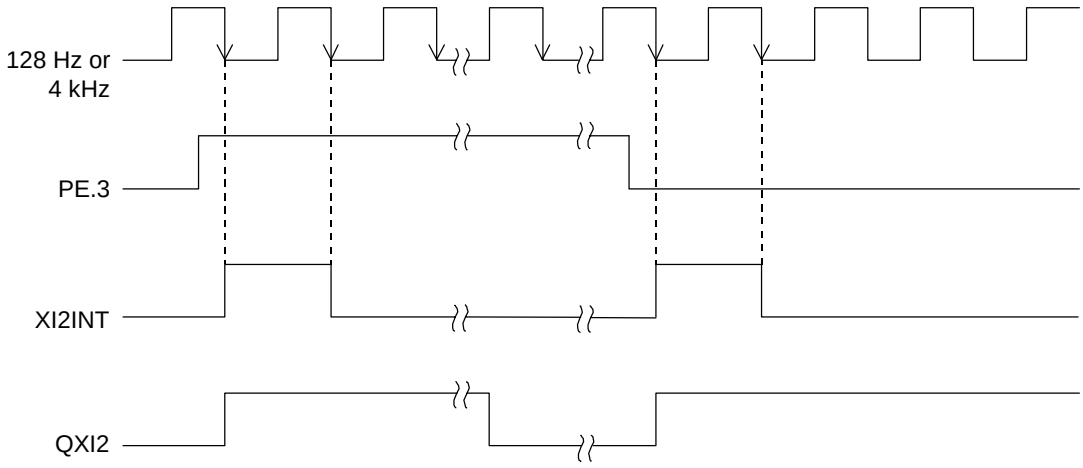


Figure 10-17 External Interrupt 2 Generation Timing

Chapter 11

Melody Driver (MELODY)

11. Melody Driver (MELODY)

11.1 Overview

The ML63611 has an internal melody circuit and buzzer circuit.

While automatically reading melody data in ROM (program memory) as specified by an MSA instruction, the melody circuit outputs a melody signal via the MD and MDB pins.

The output voltage level of the MDB pin when the melody is OFF can be selected as either V_{DD} or V_{SS} using the mask option. Refer to Section 1.3, "Mask Options" and the "MOGTOOL Mask Option Generator User's Manual" for details.

The melody circuit can select 29 different tones, 63 different tone lengths, and 15 different tempos.

The buzzer circuit has four different buzzer output modes at a frequency of 4 kHz. The buzzer driver signal is output via the MD and MDB pins.

Melody output is a higher priority operation than buzzer output.

11.2 Melody Driver Configuration

The melody driver configuration is shown in Figure 11-1.

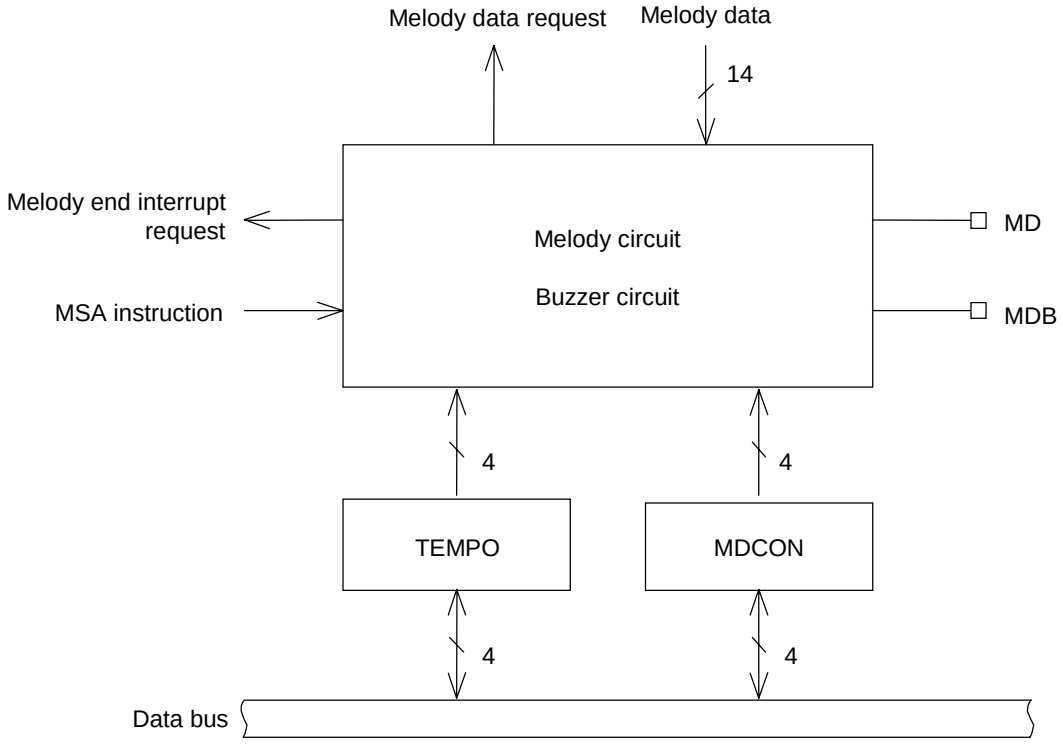
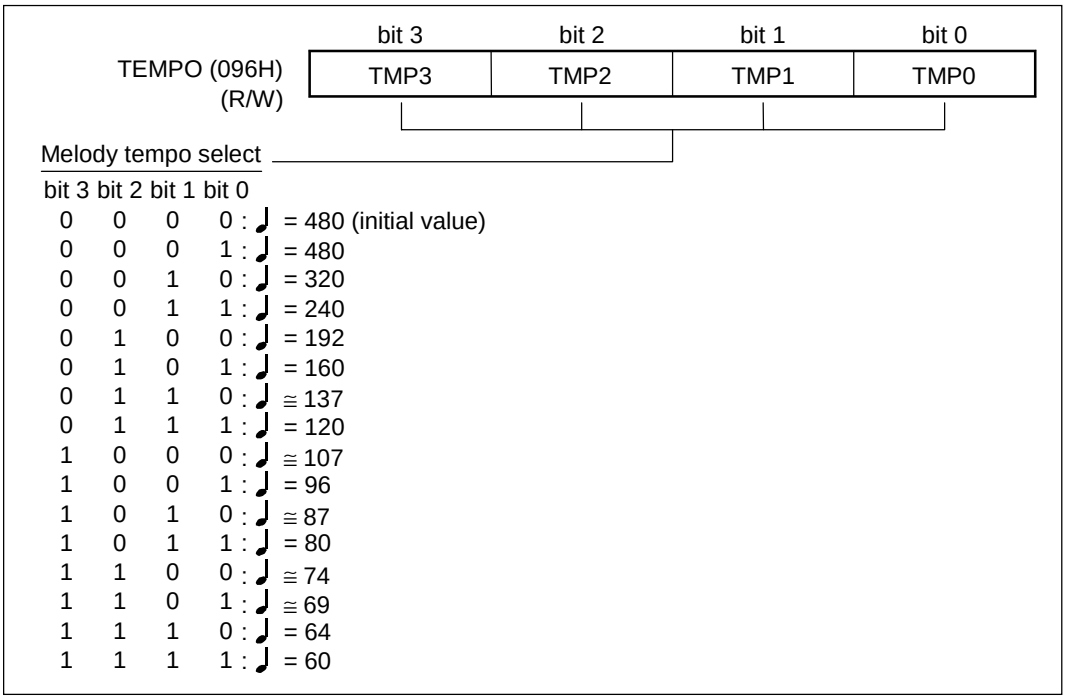


Figure 11-1 Melody Driver Configuration

11.3 Melody Driver Registers

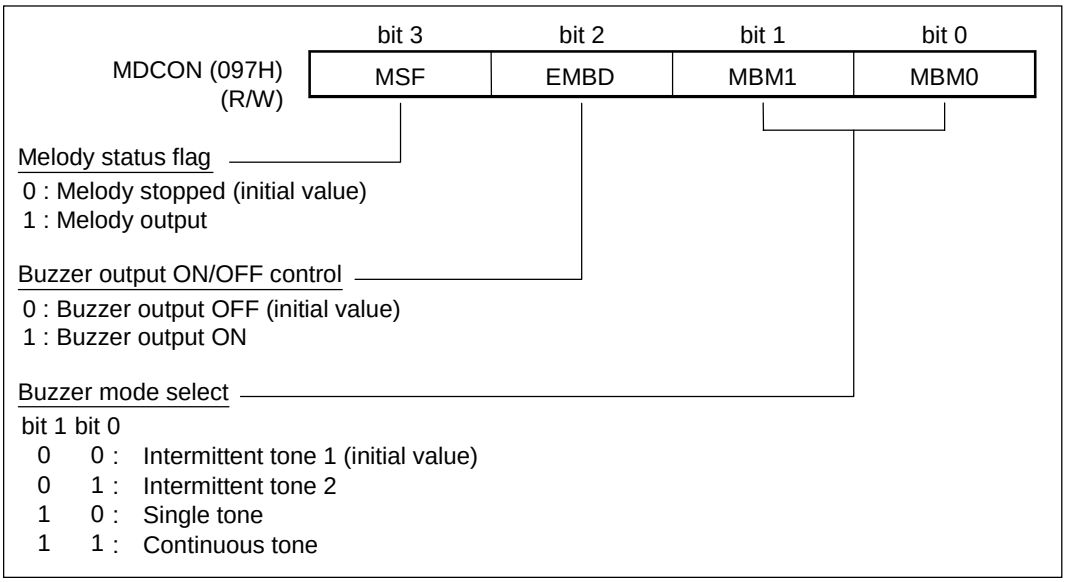
(1) Tempo Register (TEMPO)

TEMPO is a 4-bit special function register (SFR) that sets the tempo of the melody driver.



(2) Melody Driver Control Register (MDCON)

MDCON is a 4-bit special function register (SFR) that controls output of the melody driver.



bit 3: MSF

This flag indicates the melody output status.
When an MSA instruction starts the melody, MSF is set to "1". After output of the last melody data (END bit is "1"), MSF is cleared to "0".
Setting MSF to "0" during melody output will forcibly stop the melody output. If forcibly stopped, the melody output cannot be restarted at the address at which it was stopped.
At system reset, MSF is cleared to "0".

Note:

If MSF (bit 3 of MDCON) is set to "0" to stop melody output forcibly, it is required to set the stop address on the ROM table to the end-data address (8000H). In this case, set MSF to "0" after writing the melody end data that consists of two words of melody (silence with the END bit being "1") data. If this programming is not executed, melody output may not be stopped even if MSF is set to "0". Example programming is shown below.

```
;*Program part***********

    DI                      ; 0. Disable master interrupt (MIE).
    MSA   MDSTOP_DATA       ; 1. Write melody end data to the melody circuit.
    MOV   A, #0             ; 2. Set the MSF to "0".
    MOV   MDCON,A           ;
    MOV   A, #1101b         ; 3. Clear melody end interrupt request (QMD).
    AND   IRQ0, A           ;
    EI                      ; 4. Enable master interrupt (MIE).

;*ROM table data part****
;*Provide two words of melody data so that a melody will always be terminated even if a melody
;*request is issued twice.

MDSTOP_DATA:

    DW    8000H             ; Silence data 1
    DW    8000H             ; Silence data 2

;************************
```

In the Dr.63611 In-Circuit Emulator, melody output will be stopped only by setting MSF to "0"; writing melody end data is not needed.

bit 2: EMBD

This bit turns the buzzer output ON or OFF.
At system reset, EMBD is cleared to "0" and buzzer output is turned OFF.
In the single tone output mode, setting EMBD to "1" turns ON the buzzer output. After the second falling edge of the 32 Hz output, EMBD is cleared to "0" and buzzer output is turned OFF.
If melody output is started during buzzer output, EMBD is cleared to "0" and the buzzer output is turned OFF.

bit 1, 0: MBM1, MBM0

These bits select the buzzer output mode.
Output of two types of intermittent tones, a single tone or a continuous tone can be selected.
At system reset, MBM1 and MBM0 are cleared to "0", selecting output of intermittent tone 1.

Buzzer output mode	Waveform
Intermittent tone 1	Intermittent tone waveform synchronized to 8 Hz output of time base counter
Intermittent tone 2	Intermittent tone waveform synchronized to the logical AND of 8 Hz signal output and a "L" level of 1 Hz signal output of the time base counter
Single tone	Single tone waveform beginning when EMBD is set to "1" until second falling edge of 32 Hz output of time base counter
Continuous tone	Continuous tone waveform that is constant while EMBD is "1"

Figure 11-2 shows the output waveforms of the melody driver output pins.

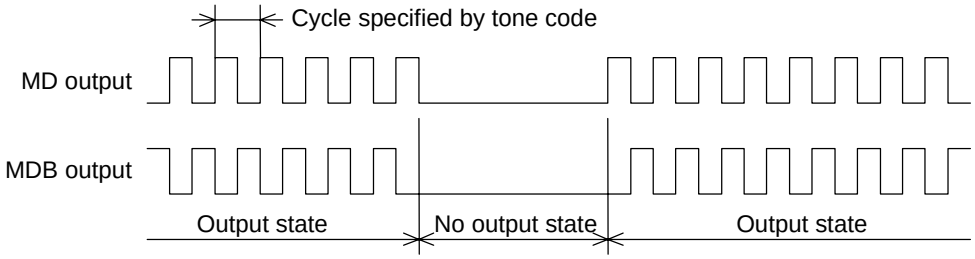


Figure 11-2 Output Waveforms of Melody Driver Output Pins

11.4 Melody Circuit Operation

After the melody tempo is set in the tempo register (TEMP), execution of an MSA instruction will start operation of the melody circuit.

The melody circuit outputs melody data while automatically reading melody data in ROM (program memory) as specified by an MSA instruction. When the last melody data is read (END bit is "1"), the melody circuit generates a melody end interrupt request. At this time, if an MSA instruction is executed, after the last melody data is output, melody output will continue from the melody data specified by the MSA instruction. If an MSA instruction is not executed, the melody output will stop after the last melody data is output.

MSF (bit 3 of MDCON) is a flag indicating the melody output status. When MSF is "1", the melody is being output, and when "0", the melody is stopped. Setting MSF to "0" during melody output will forcibly stop the melody output. If it is required to stop melody output forcibly, describe the program according to the "Note" on page 11-3. If forcibly stopped, the melody output cannot be restarted at the address at which it was stopped.

11.4.1 Tempo Data

Tempo data defines the basic tone length. Tempo data is set in the tempo register (TEMPO).

The tempos (number of counts per minute) set by TEMPO are shown in Table 11-1.

Table 11-1 Melody Tempo

TEMPO					Tempo
TP3–0	TP3	TP2	TP1	TP0	
0H	0	0	0	0	♩ = 480
1H	0	0	0	1	♩ = 480
2H	0	0	1	0	♩ = 320
3H	0	0	1	1	♩ = 240
4H	0	1	0	0	♩ = 192
5H	0	1	0	1	♩ = 160
6H	0	1	1	0	♩ ≅ 137
7H	0	1	1	1	♩ = 120
8H	1	0	0	0	♩ ≅ 107
9H	1	0	0	1	♩ = 96
AH	1	0	1	0	♩ ≅ 87
BH	1	0	1	1	♩ = 80
CH	1	1	0	0	♩ ≅ 74
DH	1	1	0	1	♩ ≅ 69
EH	1	1	1	0	♩ = 64
FH	1	1	1	1	♩ = 60

11.4.2 Melody Data

Melody data is 14-bit format data in the program ROM defining tone, tone length and end tone.

The melody data format is indicated in Figure 13-3.

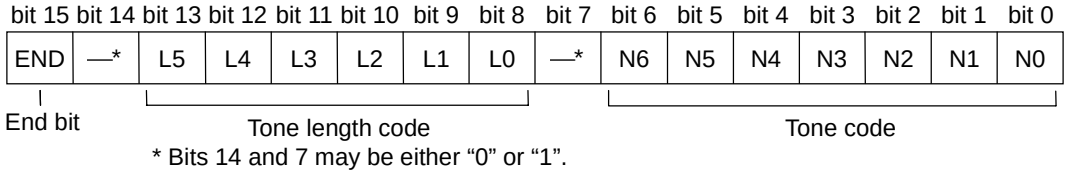


Figure 11-3 Melody Data Format

(1) Tone code

The tone code is set in bits 6 through 0 of the melody data. The frequencies that can be output by the melody circuit are defined as:

$$\frac{65536}{(N + 2)} \text{ Hz (where N is an integer from 4 to 127)}$$

The relation between N and tone code bits is:

$N = 2^6N6 + 2^5N5 + 2^4N4 + 2^3N3 + 2^2N2 + 2^1N1 + 2^0N0$

If N6 through N2 are all set to "0", there is no melody output for the time specified by the tone length code. Values for N1 and N0 are irrelevant.

Table 11-2 indicates the relations between tones and tone codes.

Table 11-2 Tone and Tone Code Correspondence

Tone	Frequency (Hz)	Tone code							
		N6	N5	N4	N3	N2	N1	N0	N6–N0
C^1	529	1	1	1	1	0	1	1	7BH
Cis^1	560	1	1	1	0	0	1	1	73H
D^1	590	1	1	0	1	1	0	1	6DH
Dis^1	624	1	1	0	0	1	1	1	67H
E^1	662	1	1	0	0	0	0	1	61H
F^1	705	1	0	1	1	0	1	1	5BH
Fis^1	745	1	0	1	0	1	1	0	56H
G^1	790	1	0	1	0	0	0	1	51H
Gis^1	840	1	0	0	1	1	0	0	4CH
A^1	886	1	0	0	1	0	0	0	48H
Ais^1	936	1	0	0	0	1	0	0	44H
B^1	993	1	0	0	0	0	0	0	40H
C^2	1057	0	1	1	1	1	0	0	3CH
Cis^2	1111	0	1	1	1	0	0	1	39H
D^2	1192	0	1	1	0	1	0	1	35H

Table 11-2 Tone and Tone Code Correspondence (continued)

Tone	Frequency (Hz)	Tone code							
		N6	N5	N4	N3	N2	N1	N0	N6–N0
Dis^2	1260	0	1	1	0	0	1	0	32H
E^2	1338	0	1	0	1	1	1	1	2FH
F^2	1394	0	1	0	1	1	0	1	2DH
Fis^2	1490	0	1	0	1	0	1	0	2AH
G^2	1560	0	1	0	1	0	0	0	28H
Gis^2	1680	0	1	0	0	1	0	1	25H
A^2	1771	0	1	0	0	0	1	1	23H
Ais^2	1872	0	1	0	0	0	0	1	21H
B^2	1986	0	0	1	1	1	1	1	1FH
C^3	2114	0	0	1	1	1	0	1	1DH
D^3	2341	0	0	1	1	0	1	0	1AH
Dis^3	2521	0	0	1	1	0	0	0	18H
E^3	2621	0	0	1	0	1	1	1	17H
Fis^3	2979	0	0	1	0	1	0	0	14H

(2) Tone length code

The tone length code is set in melody data bits 13 through 8.

Table 11-3 indicates the relation between tone length and tone length code (L5 to L0).

The tone length that is set during execution of the MSA instruction is shorter by approximately 1 to 3 ms.

When all bits are set to "0", the tone length will be the same as the minimum tone length (the tone length with only L0 set to "1").

Table 11-3 Tone Length and Tone Length Code Correspondence

Tone length	Tone length code						
	L5	L4	L3	L2	L1	L0	L5–L0
	1	1	1	1	1	1	3FH
	1	0	1	1	1	1	2FH
	0	1	1	1	1	1	1FH
	0	1	0	1	1	1	17H
	0	0	1	1	1	1	0FH
	0	0	1	0	1	1	0BH
	0	0	0	1	1	1	07H
	0	0	0	1	0	1	05H
	0	0	0	0	1	1	03H
	0	0	0	0	1	0	02H
	0	0	0	0	0	1	01H

Tone lengths specified by the tone length code and the tempo data are expressed by the following:

$1.953125 \times (TP + 1) \times (L + 1)$ ms (where TP is an integer from 1 to 15, and L is an integer from 1 to 63)

TP is a value set in the tempo register (TEMPO), and has the following bit correspondence:

$TP = 2^3TP3 + 2^2TP2 + 2^1TP1 + 2^0TP0$

L is set by the tone length code, and has a bit correspondence with the tone length code as:

$L = 2^5L5 + 2^4L4 + 2^3L3 + 2^2L2 + 2^1L1 + 2^0L0$

(3) END bit

The END bit is set in bit 15 of the melody data. When the output of the last melody data is started (END bit is "1"), the melody circuit generates a melody end interrupt request, and stops the melody after the last melody data is output.

11.4.3 Melody Circuit Application Example

An example melody is shown in Figure 11-4.

Table 11-4 lists the note codes for the melody shown in Figure 11-4.

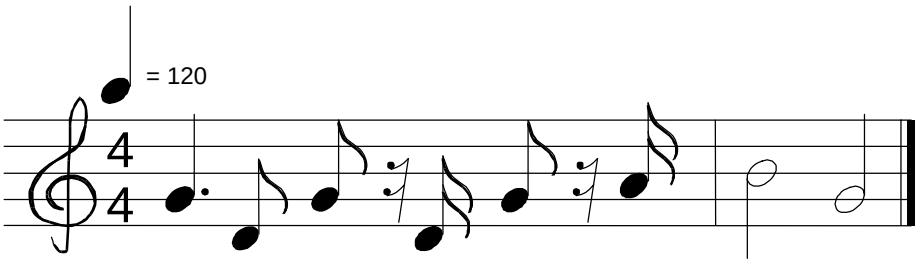


Figure 11-4 Example Melody

Table 11-4 Note Code Table

Note	Note code																Hex
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	END	—*	L5	L4	L3	L2	L1	L0	—*	N6	N5	N4	N3	N2	N1	N0	
G^2	0	0	1	0	1	1	1	1	0	0	1	0	1	0	0	0	2F28H
D^2	0	0	0	0	1	1	1	1	0	0	1	1	0	1	0	1	0F35H
G^2	0	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	0F28H
—	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0700H
D^2	0	0	0	0	0	1	1	1	0	0	1	1	0	1	0	1	0735H
G^2	0	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	0F28H
—	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0700H
A^2	0	0	0	0	0	1	1	1	0	0	1	0	0	0	1	1	0723H
B^2	0	0	1	1	1	1	1	1	0	0	0	1	1	1	1	1	3F1FH
G^2	1	0	1	1	1	1	1	1	0	0	1	0	1	0	0	0	BF28H

* Bits 14 and 7 may be "0" or "1", but in this example they are shown as "0".

11.5 Buzzer Circuit Operation

When EMBD (bit 2 of MDCON) is set to "1", a buzzer driver signal is sent to the melody driver output pins (MD, MDB).

Four buzzer output modes can be selected by MBM1 (bit 1 of MDCON) and MBM0 (bit 0 of MDCON): two types of intermittent tones, a single tone, or a continuous tone output. The buzzer output frequency is 4 kHz and has a 50% duty ratio.

In the intermittent tone 1 mode, a waveform synchronized to the 8 Hz output of the time base counter is output.

In the intermittent tone 2 mode, a waveform synchronized to the logical AND of 8 Hz signal output and a "L" level of 1 Hz signal of the time base counter is output.

In the single tone mode, output starts in synchronization with the rising edge of EMBD. At the second falling edge of the 32 Hz output of the time base counter, EMBD is cleared to "0" and output is stopped.

In the continuous tone mode, output is continued while EMBD is "1".

While the melody is being output (MSF (bit 3 of MDCON) = "1"), the buzzer output is turned OFF. If melody output is started during buzzer output, EMBD is cleared to "0", the buzzer output is stopped, and melody output is given priority.

Figure 11-5 shows the output waveforms of each mode. Shaded sections indicate the 4 kHz output frequency.

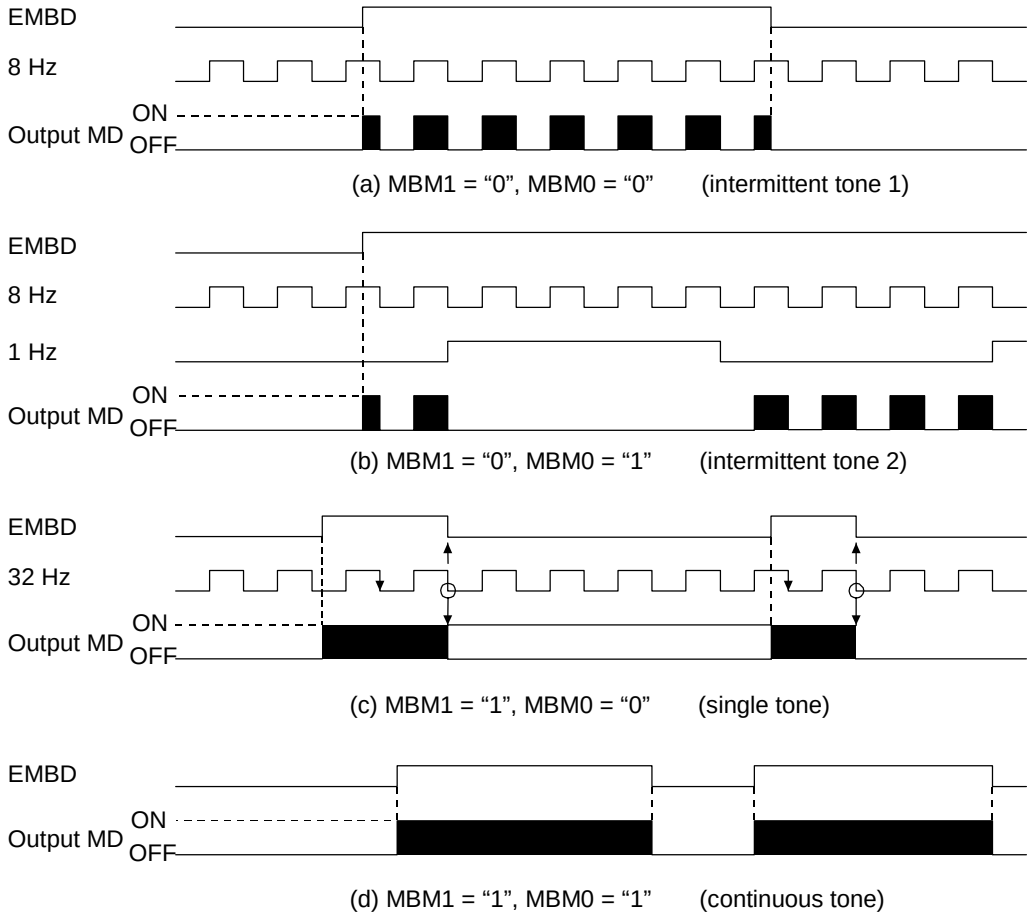


Figure 11-5 Buzzer Driver Output Waveforms in Each Output Mode

Chapter 12

Serial Port (SIO)

12. Serial Port (SIO)

12.1 Overview

The ML63611 has a built-in serial communication port (serial port) for either synchronous or asynchronous communication.

The serial port implements the send and receive circuits in independent circuits, making it possible to send and receive simultaneously.

The send and receive modes can be UART mode (asynchronous communication mode) or synchronous mode (synchronous communication mode).

In synchronous mode an internal clock mode generates the shift clock internally, and an external clock mode receives an external shift clock.

Table 12-1 shows the serial port modes.

Table 12-1 Serial Port Modes

		Mode		Baud rate
Serial port	Send side	UART mode		Can be set to a user-specified value with timers 2, 3 (TM2, 3)
		Synchronous mode	Internal clock mode	32.768 kHz
			External clock mode	From external clock
	Receive side	UART mode		• 9600 bps • 4800 bps • 2800 bps • 1200 bps
		Synchronous mode	Internal clock mode	32.768 kHz
			External clock mode	From external clock

12.2 Serial Port Configuration

Figure 12-1 indicates the serial port configuration.

The serial port consists of the send/receive clock generator circuits, the send/receive control registers, the buffer registers to store send/receive data, send/receive data transfer shift registers, and the send/receive status registers.

PC.0/RXD is the send serial data input pin, PC.3/TXD is the send serial data output pin, PC.1/TXC is the serial send clock I/O pin, and PC.2/RXC is the serial receive clock I/O pin. Set I/O and secondary functions with the port control registers as needed for each communication mode.

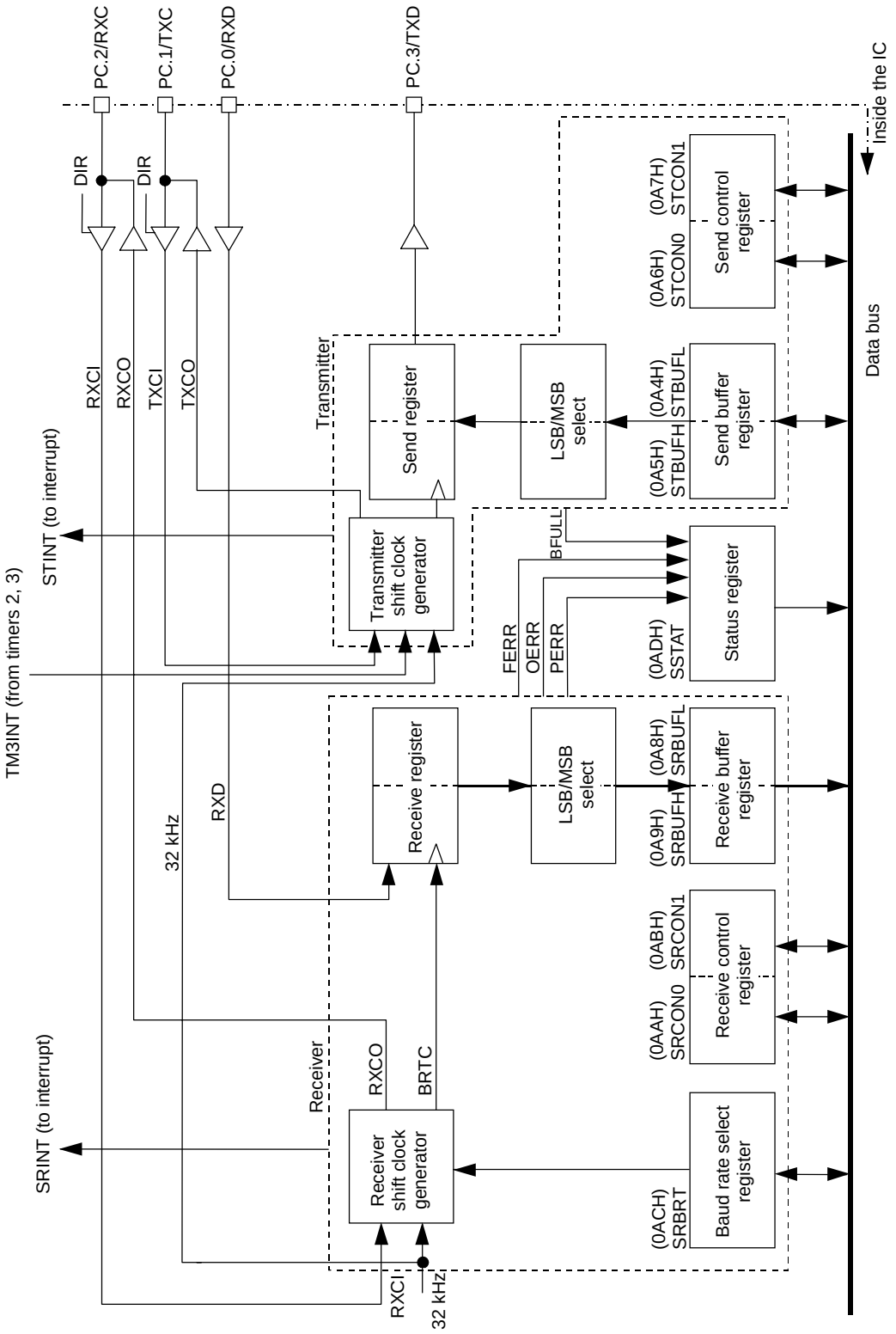
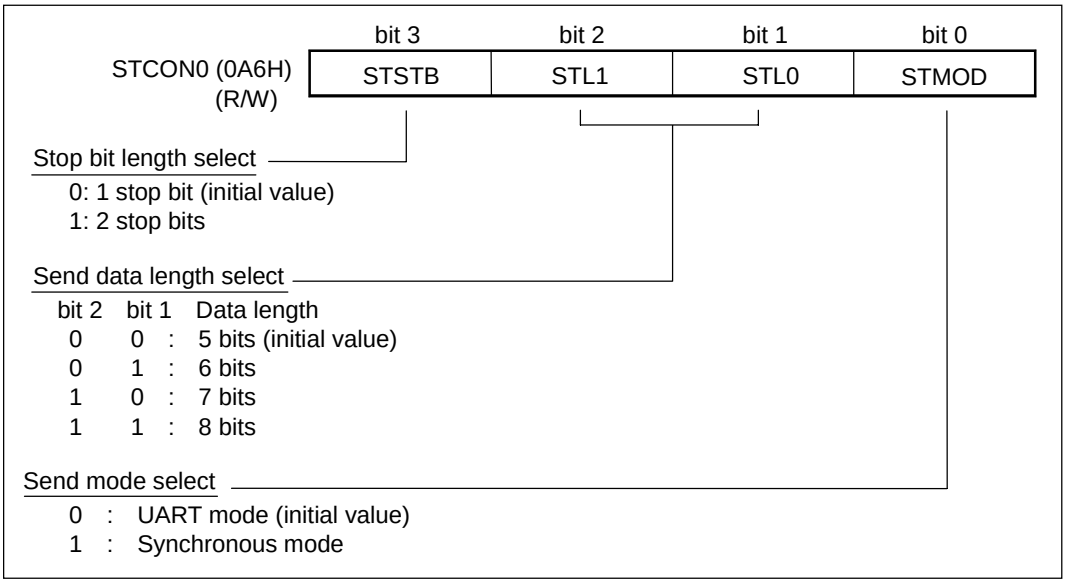


Figure 12-1 Serial Port Configuration

12.3 Serial Port Registers

(1) Send control registers 0/1 (STCON0, STCON1)

STCON0 and STCON1 are 4-bit special function registers (SFRs) to control the serial port send operation. STCON0 and STCON1 are initialized to "0" at system reset.



bit 3: STSTB (Serial Transmission STop Bit)

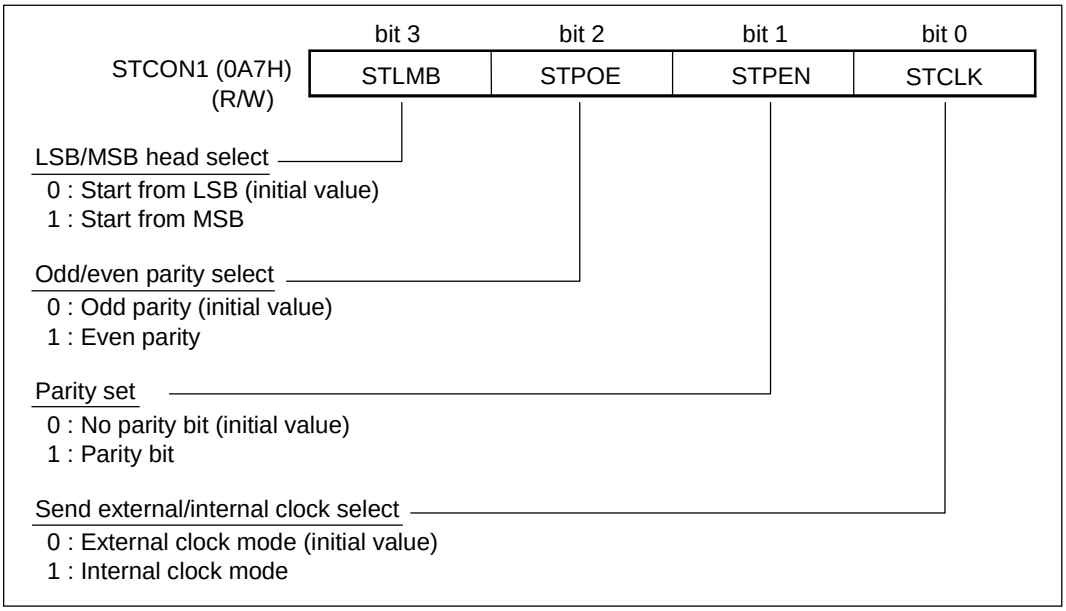
This bit specifies stop bit length. Valid only when bit 0 is "0" (UART mode).

bit 2, 1: STL1 (Serial Transmission Length select bit 1),
STL0 (Serial Transmission Length select bit 0)

These bits specify the send data length.

bit 0: STMOD (Serial Transmission MODe bit)

This bit specifies the serial port send operation mode.



bit 3: STLMB (Serial Transmission Least significant bit first or Most significant Bit first)

This bit specifies either LSB first or MSB first for send data.

bit 2: STPOE (Serial Transmission Parity Odd or Even number bit)

This bit specifies whether the parity bit is even or odd. Valid only when bit 1 is "1" (parity bit).

bit 1: STPEN (Serial Transmission Parity ENable bit)

This bit specifies whether or not a parity bit is added.

bit 0: STCLK (Serial Transmission CLock select bit)

This bit specifies the external/internal send clock for synchronous mode. Valid only when STMOD (bit 0 of STCON0) is "1" (synchronous mode).

(2) Send buffer registers (STBUFL, STBUFH)

	bit 3	bit 2	bit 1	bit 0
STBUFL (0A4H) (R/W)	TB3	TB2	TB1	TB0
STBUFH (0A5H) (R/W)	TB7	TB6	TB5	TB4

STBUFL and STBUFH are 4-bit special function registers (SFRs) that set send data for serial port send operation.

LSB/MSB selection (described later) allows the data send direction (LSB or MSB first) to be specified. Both STBUFL and STBUFH are initialized to "0" at system reset.

Send operation begins when send data is set to STBUFH. Be sure to set send data to STBUFL before setting data to STBUFH.

Also set the baud rate and send mode before beginning send operation.

If send operation is already under way when send data is set to STBUFH, send for the new data begins when the prior send has ended, and at the same time an interrupt request signal (STINT) is generated. In the STINT interrupt routine the program should first write the send data to STBUFL and STBUFH to assure no pauses in the send sequence.

(3) Send register

The send register is a shift register that handles the shift operation in send. At system reset it is cleared to 00H. The send register cannot be directly accessed from the CPU.

The hardware send flow is indicated in Figure 12-2, to explain the timing for transfer of data from STBUFL/H to the send register.

First set the send mode and baud rate. When send data is set to STBUFH, the status (SSTAT) buffer full flag (BFULL) is set to "1", and unless send operation is already under way the content of STBUFL/H is transferred to the send register and send operation begins. When send operation begins the BFULL flag is reset to "0", and the next send data can be set to STBUFL/H.

If prior data send operation is not complete, the send data is held in STBUFL/H until send is completed. In this case BFULL remains set to "1". When the prior send operation is complete the send data will be transferred from STBUFL/H to the send register, and send begins.

Note:

When BFULL is "1" it is possible to set data to STBUFL/H, but prior data set to STBUFL/H that is being held there is overwritten and lost. Always set data after verifying that the BFULL flag is "0".

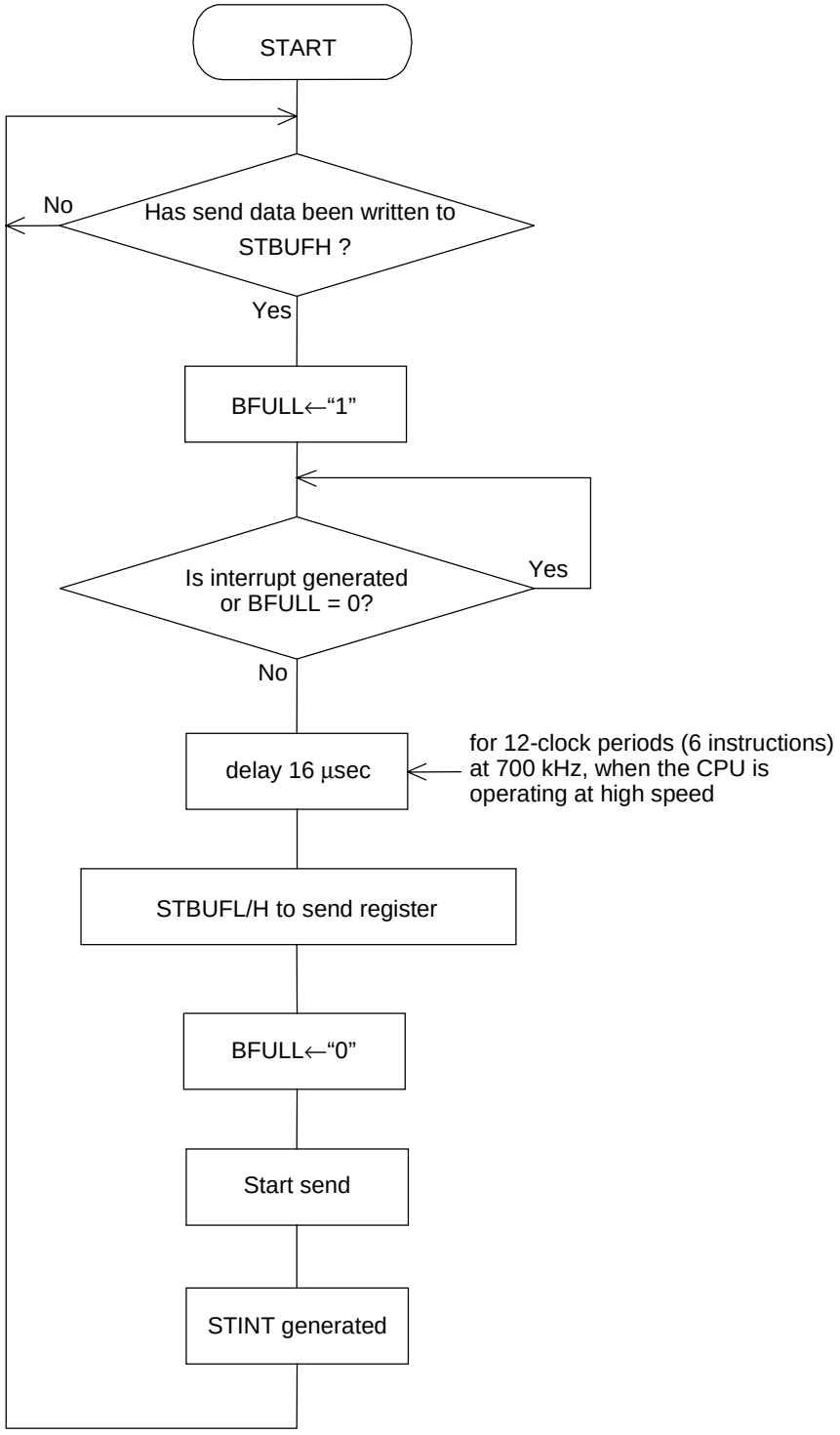
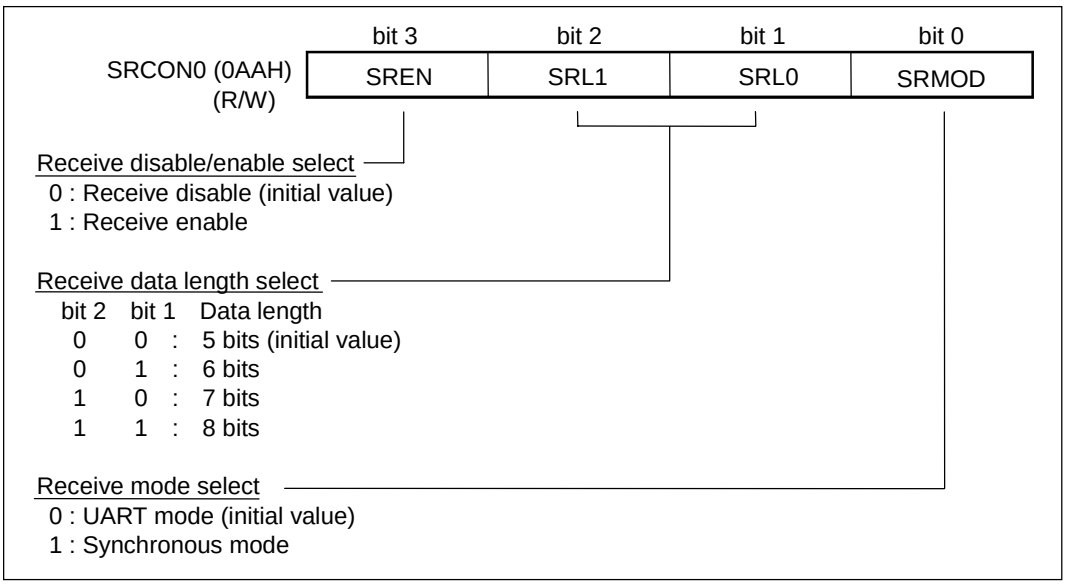


Figure 12-2 Hardware Send Operation Flow

(4) Receive control registers 0/1 (SRCON0, SRCON1)

SRCON0 and SRCON1 are 4-bit special function registers (SFRs) controlling serial port receive operation.

SRCON0 and SRCON1 are initialized to "0" at system reset.



bit 3: SREN (Serial Reception ENable bit)

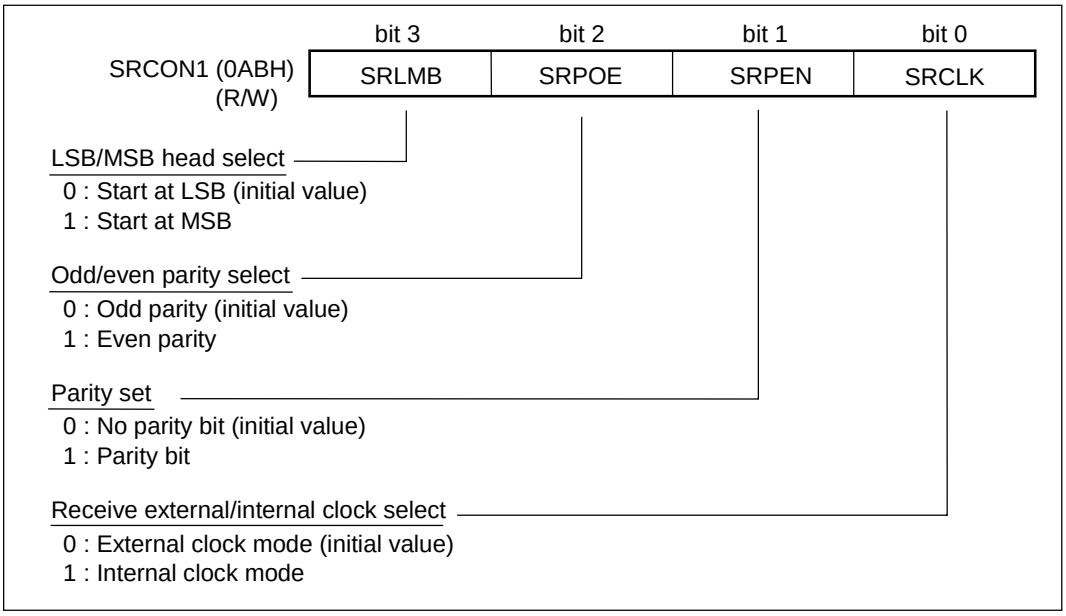
This bit specifies receive operation disable/enable. After receive is enabled in the synchronous mode, this bit is reset to "0" after receiving one frame of data. In the UART mode it does not change.

bit 2, 1: SRL1 (Serial Reception Length select bit 1),
SRL0 (Serial Reception Length select bit 0)

These bits specify the receive data length.

bit 0: SRMOD (Serial Reception MODe bit)

This bit specifies the serial port receive operation mode.



bit 3: SRLMB (Serial Reception Least significant bit first or Most significant Bit first)

This bit specifies either LSB first or MSB first for receive data.

bit 2: SRPOE (Serial Reception Parity Odd or Even number bit)

This bit specifies whether the parity bit is even or odd. Valid only when bit 1 is "1" (parity bit).

bit 1: SRPEN (Serial Reception Parity ENable bit)

This bit specifies whether or not a parity bit is added.

bit 0: SRCLK (Serial Reception CLocK select bit)

This bit specifies the external/internal receive clock for synchronous mode. Valid only when SRMOD (bit 0 of SRCON0) is "1" (synchronous mode).

(5) Receive register

The receive register is the shift register that handles shift operation at receive. It is initialized to 00H at system reset. It cannot be directly accessed by the CPU. When a receive operation is complete, the data read into the receive register is transferred to SRBUFL/H, and at the same time the receive interrupt request signal (SRINT) is generated.

(6) Receive buffer registers (SRBUFL, SRBUFH)

	bit 3	bit 2	bit 1	bit 0
SRBUFL (0A8H) (R)	RB3	RB2	RB1	RB0
SRBUFH (0A9H) (R)	RB7	RB6	RB5	RB4

SRBUFL and SRBUFH are 4-bit special function registers (SFRs) used to hold the received data in serial port reception. SRBUFL and SRBUFH are initialized to "0" at system reset.

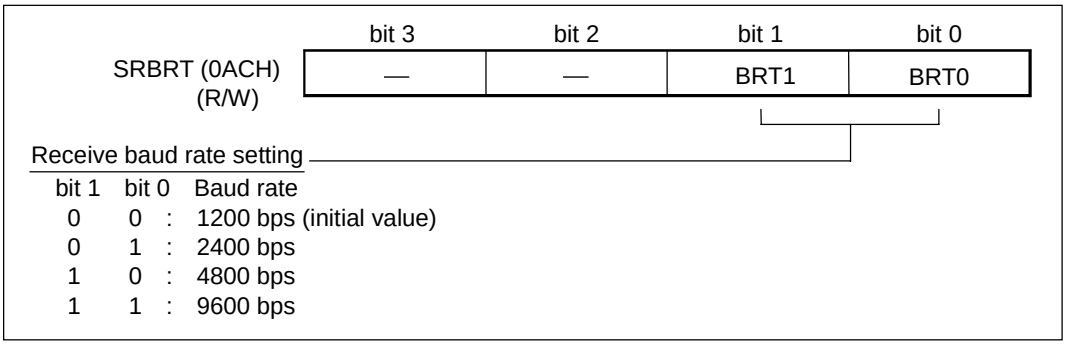
When receive operation is completed the contents of the receive register are sent to SRBUFL/H, and the receive interrupt request (SRINT) is generated. The contents of SRBUFL/H are held until the next receive operation is completed.

If data from a prior receive operation is in SRBUFL/H and new data is received, an overrun error will result. When an overrun error is generated, new received data cannot be loaded into SRBUFL/H.

(7) Receive baud rate setting register (SRBRT)

SRBRT is a 4-bit special function register (SFR) used to set the receive baud rate for serial port receive operation in UART mode.

SRBRT is initialized to 0CH at system reset.



bit 1, 0: BRT1 (Baud RaTe select bit 1), BRT0 (Baud RaTe select bit 0)

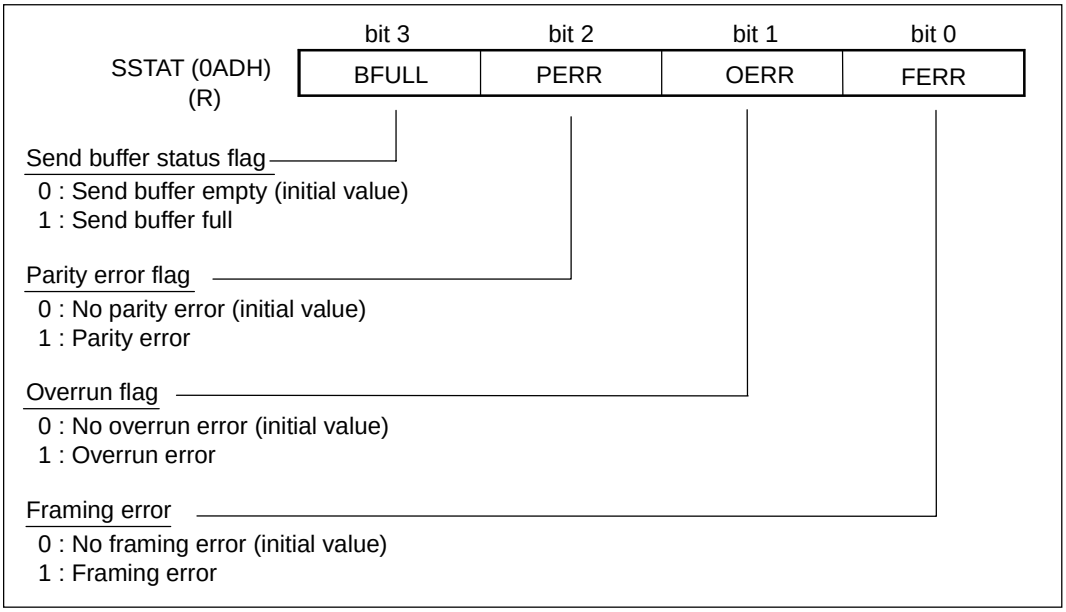
These bits set the receive baud rate.

(8) Serial port status register (SSTAT)

SSTAT is a 4-bit special function register (SFR) used to indicate the status of serial port send/receive.

SSTAT is initialized to "0" at system reset.

SSTAT is a read-only register, and the content is reset every time it is read.



bit 3: BFULL (send Buffer FULL flag)

This bit is enabled in both UART and synchronous modes, and is set to "1" when send data is set to STBUFL/H in the send mode, and reset to "0" when the send data is transferred to the send register.
When BFULL is set to "1" and send data is set (written) to STBUFL/H, the previous data set to those registers is overwritten and lost. Always set data only after verifying that the BFULL flag is "0".

bit 2: PERR (Parity ERRor flag)

This bit is enabled in both UART and synchronous modes, and is set to "1" when the parity for the received data does not match the parity bit attached to the data.

bit 1: OERR (Overrun ERRor flag)

This bit is enabled in both UART and synchronous modes, and is set to "1" when data reception is completed and the data received the previous time has still not been transferred to the CPU. In this case, the new data cannot be transferred to SRBUFL/H.

bit 0: FERR (Framing ERRor flag)

This is only enabled in the UART mode and is set to "1" in the following instances.

(1) when a "1" is detected in start bit sampling
(2) when a "0" is detected in stop bit sampling

In either case a receive interrupt request signal (SRINT) is generated.

12.4 Serial Port Operation Description

12.4.1 Data Format

(1) UART mode

The data format for the UART mode is shown in Figure 12-3.

SRCON0/1 and STCON0/1 can be set to specify a data bit length of 5 to 8 bits. The parity bit can be enabled/disabled. If enabled it can be set to even or odd. Stop bit length can be set to 1 or 2 bits.

The combination of these parameters gives a range of from 7 to 12 bits for send/receive data frames.

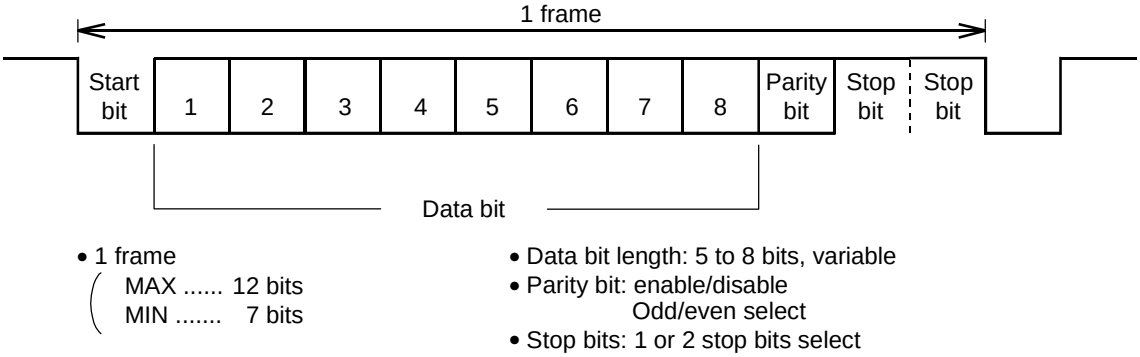


Figure 12-3 UART Mode Data Format

(2) Synchronous mode

The data format for the UART mode is shown in Figure 12-4.

SRCON0/1 and STCON0/1 can be set to specify a data bit length of 5 to 8 bits. The parity bit can be enabled/disabled, and if enabled can be set to even or odd.

The combination of these parameters gives a range of from 5 to 9 bits for send/receive data frames.

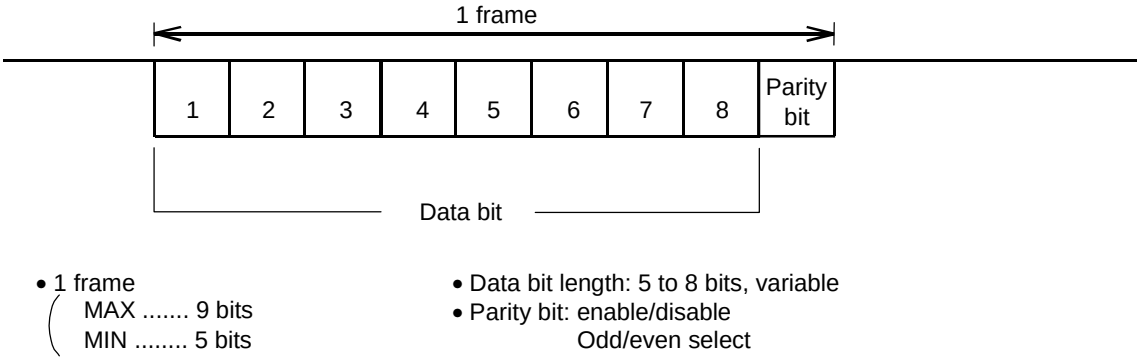


Figure 12-4 Synchronous Mode Data Format

12.4.2 Send Operation Description

The serial port send circuit has a two-stage configuration. This consists of the send register and the send buffer register (STBUFL/H), so it is possible to set send data to STBUFL/H while sending the previous data. When the BFULL flag of the serial port status register (SSTAT) is 1, however, it indicates that STBUFL/H send data has not yet been transferred to the send register. Always verify that the BFULL flag is 0 before transferring data.

(1) UART mode

The UART mode is specified by setting STMOD (bit 0 of STCON0) to "0". Figure 12-5 is the UART mode send timing chart. The UART mode send procedure is described below. The send baud rate is set first, then the timer, and then the send format (data bit length, parity bit, etc.) in STCON0 and STCON1. The TM3INT signal supplied from timers 2 and 3 is the baud rate clock.

① Set send data to STBUFL/H.

② The send data is transferred from STBUFL/H to the send register, and send operation begins. At the same time the serial port send interrupt request (STINT) is generated.

③ Verify that BFULL = "0", then after a delay of 16 μsec, set the next send data to STBUFL/H.

④ When send operation is complete, the send data set to STBUFL/H is transferred to the send register, and send operation begins. At the same time the serial port send interrupt request (STINT) is generated.

Repeat operation ③ the required number of times.

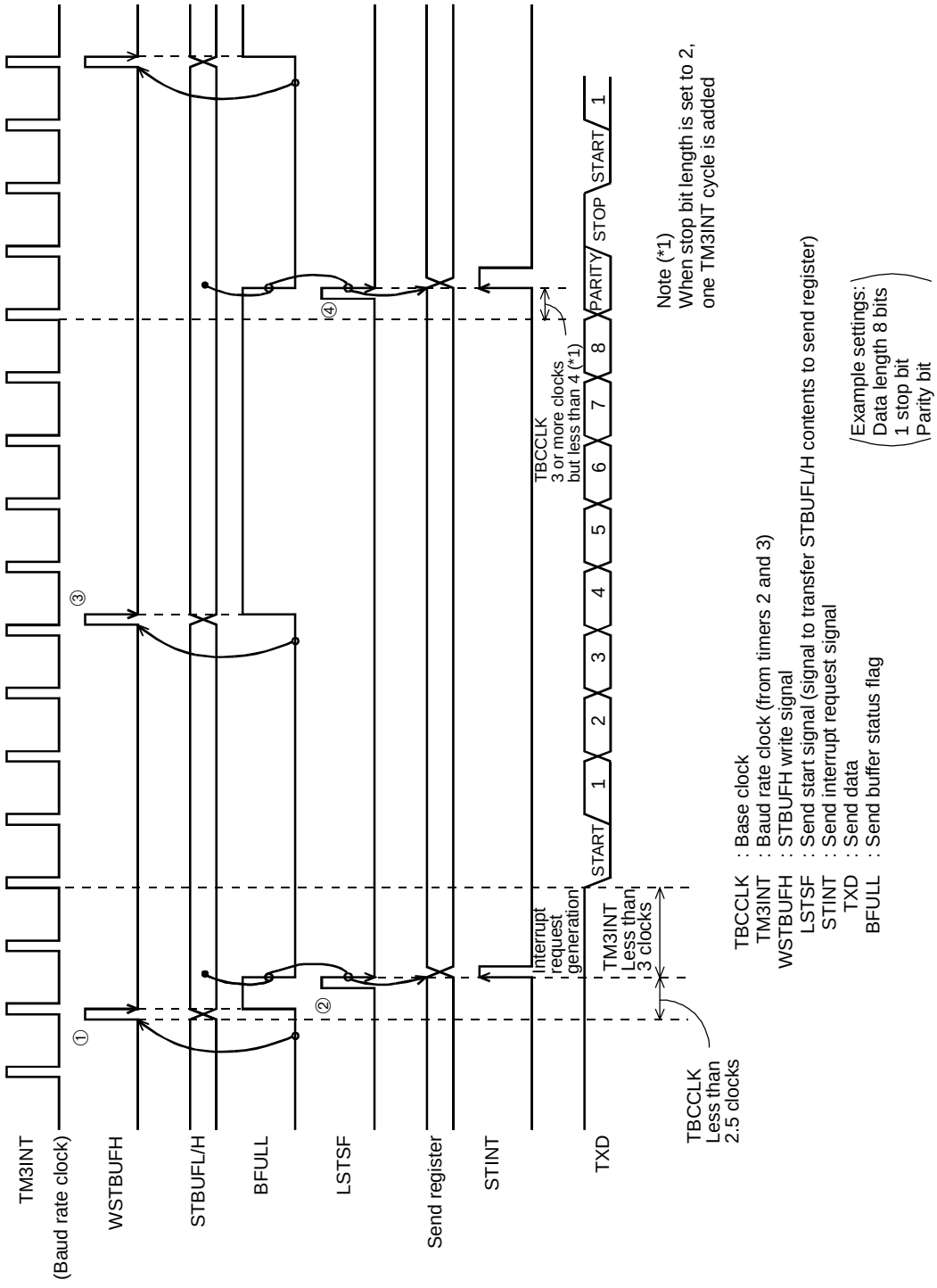


Figure 12-5 UART Mode Send Timing Chart

(2) Synchronous internal clock mode

The synchronous internal clock mode is selected by setting STMOD (bit 0 of STCON0) to "1", and STCLK (bit 0 of STCON1) to "1".

Figure 12-6 is the send timing chart for the synchronous internal clock mode.

The synchronous internal clock send procedure is described below.

First the send format (data bit length, parity bit, etc.) is set to STCON0 and STCON1.

① Set send data to STBUFL/H.

② The send data is transferred from STBUFL/H to the send register, and send operation begins. At the same time the interrupt request signal (STINT) is generated.

③ Check that BFULL = "0", then set the next send data to STBUFL/H.

④ When the send operation is complete, the send data set to STBUFL/H is transferred to the send register, and the send operation begins. At the same time, the serial port send interrupt signal (STINT) is generated.

Repeat step ③ the required number of times.

In the synchronous internal clock mode the send baud rate is fixed at the crystal oscillation frequency, that is, the frequency (32.768 kHz) of the time base clock (TBCCLK).

After data is set to STBUFH, the send clock (TXCO) generates between 2 and 3.5 clocks of the TBCCLK source, and a send operation starts.

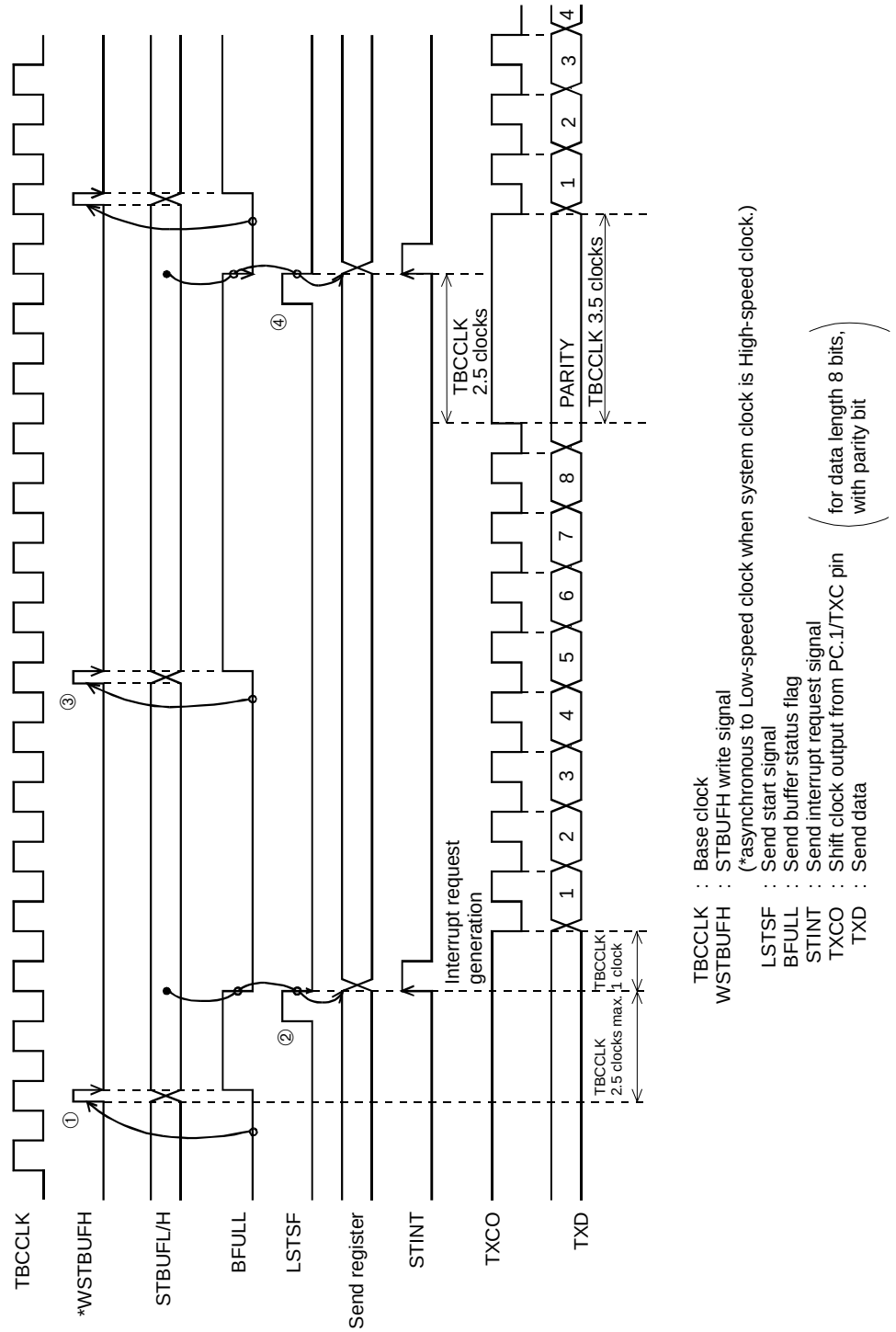


Figure 12-6 Send Timing Chart for Synchronous Internal Clock Mode

(3) Synchronous external clock mode

The synchronous external clock mode is selected by setting STMOD (bit 0 of STCON0) to "1", and STCLK (bit 0 of STCON1) to "0".

Figure 12-7 is the send timing chart for the synchronous external clock mode.

The synchronous external clock send procedure is described below.

First set the send format (data bit length, parity bit, etc.) to STCON0 and STCON1.

① Set send data to STBUFL/H.

② The send data is transferred from STBUFL/H to the send register, and at the same time the interrupt request signal (STINT) is generated.

③ Send operation is started by the send shift clock (TXCI).

④ Check that BFULL = "0", then set the next send data to STBUFL/H.

⑤ When the send operation is complete, the send data set to STBUFL/H is transferred to the send register. At the same time, the serial port send interrupt signal (STINT) is generated.

Repeat step ④ the required number of times.

In the synchronous external clock mode the send baud rate is determined by the input shift clock (TXCI).
To send data continuously, keep an interval of at least 3.5 clocks (approx. 107 µs) of TBCCLK for one frame of clocked (TXCI) send data.

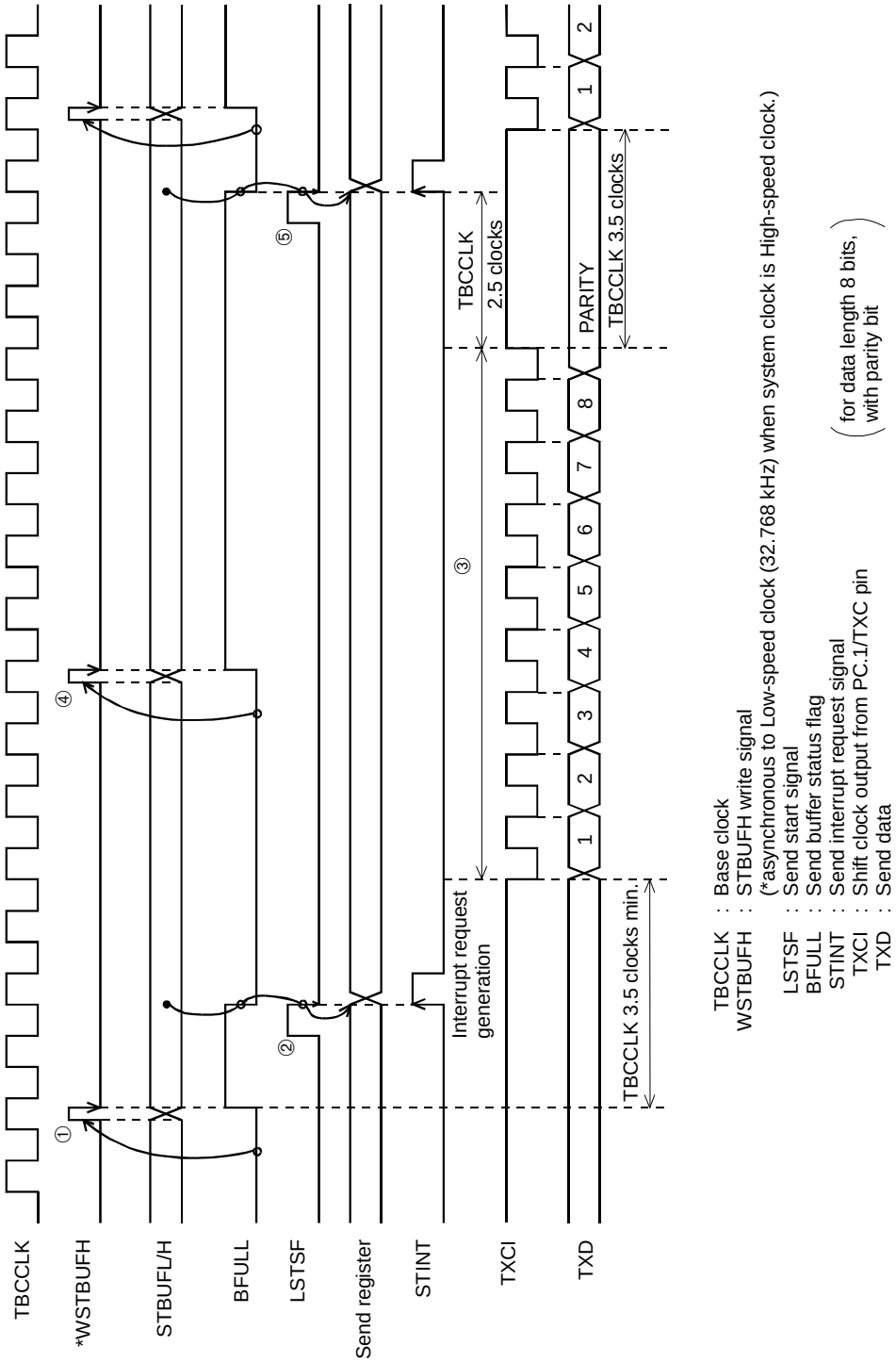


Figure 12-7 Send Timing Chart for Synchronous External Clock Mode

12.4.3 Receive Operation Description

(1) UART mode

The UART mode is specified by setting SRMOD (bit 0 of SRCON0) to "0". Figure 12-8 is the UART mode receive timing chart. The UART mode receive procedure is described below.

First set the receive baud rate in the receive baud rate setting register (SRBRT). Supported baud rates for UART mode receive are 1200, 2400, 4800, and 9600 bps.

Set the receive format (data bit length, parity bit, etc.) in SRCON0 and SRCON1.

① Set SREN (bit 3 of SRCON0) to "1" to enable receive.

② At the negative edge of the receive data (RXD) start bit, receive operation will start.

③ Receive operation ends.

If a framing or overrun error occurs the FERR or OERR flag of the status register (SSTAT) will be set to "1".

④ Received data is transferred to SRBUFL/H.

If a parity error occurs, the PERR flag of the status register (SSTAT) is set to "1".

⑤ The serial port receive interrupt request (SRINT) is generated.

Receive data is received until receive is disabled (SREN = "0"). When receive is ended, reset the receive enable/disable flag (SREN) to "0".

The receive data sampling clock (SRSMPL) is based on the low-speed clock supply, not on the high-speed clock. This allows receive operations to be executed while in the energy-saving mode.

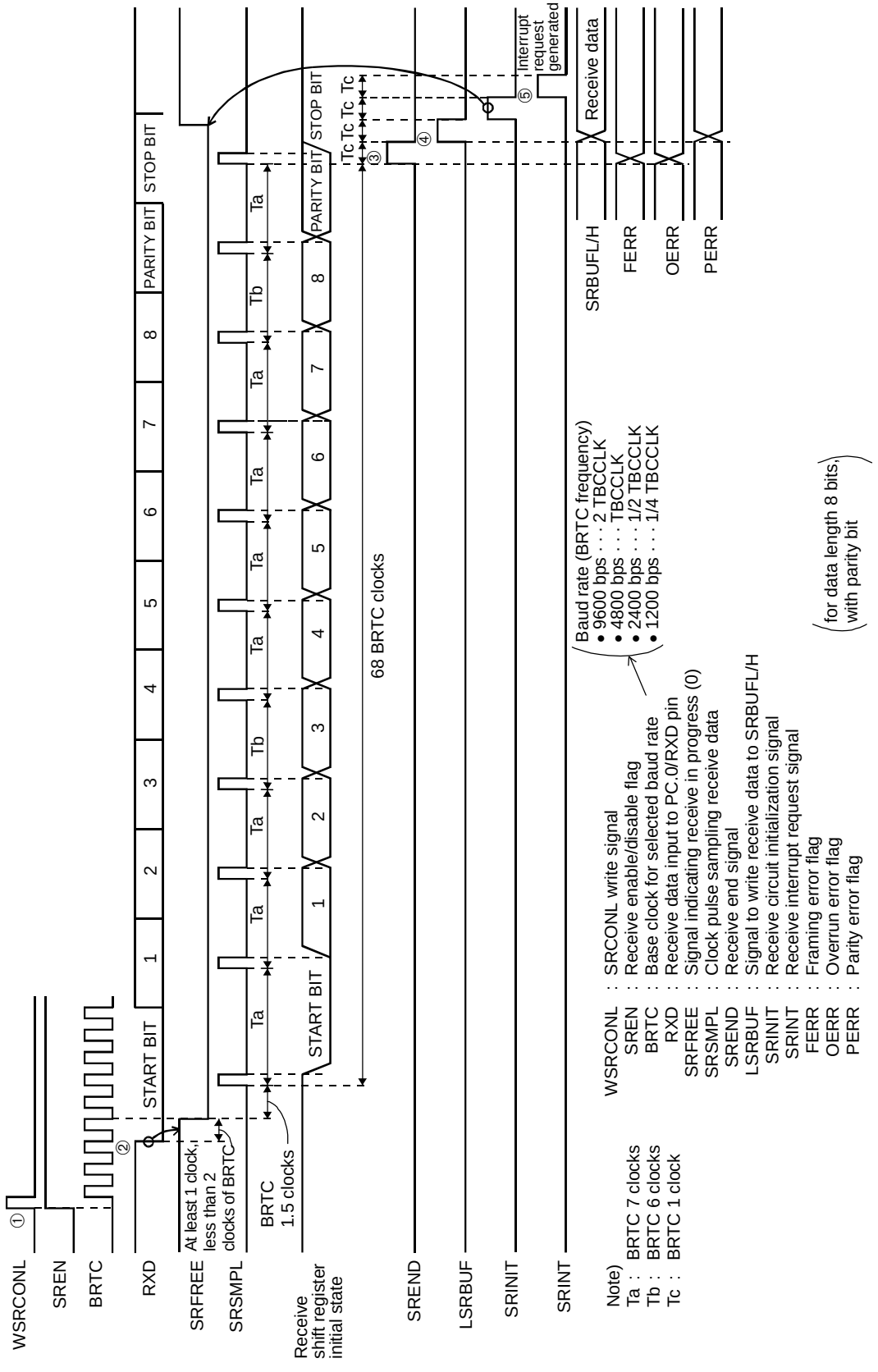


Figure 12-8 UART Mode Receive Timing Chart

(2) Synchronous internal clock mode

The synchronous internal clock mode is selected by setting SRMOD (bit 0 of SRCON0) to "1" and SRCLK (bit 0 of SRCON1) to "1".

Figure 12-9 is the receive timing chart for the synchronous internal clock mode.

The synchronous internal clock receive procedure is indicated below.

First set the receive format (data bit length, parity bit, etc.) in SRCON1 and SRCON0.

① Set SREN (bit 3 of SRCON0) to "1" (receive enable).

② 3 to 4 BRTC clock cycles later the receive shift clock (RXCO) is generated, and the receive operation starts.

(The shift clock is supplied from the PC.2/RXC pin.)

③ At the positive edge of RXCO the data received from the PC.0/RXD pin is written to the receive register.

④ Receive operation ends.

If an overrun error occurs the OERR flag in status register (SSTAT) is set to "1".

⑤ Received data is transferred to SRBUFL/H.

If a parity error occurs, the PERR flag of status register (SSTAT) is set to "1".

⑥ The serial port receive interrupt request signal (SRINT) is generated.

⑦ At the negative edge of SRINT, SREN is reset to "0".

Repeat step ① required number of times. In the synchronous internal clock mode the receive baud rate is fixed to TBCCLK.

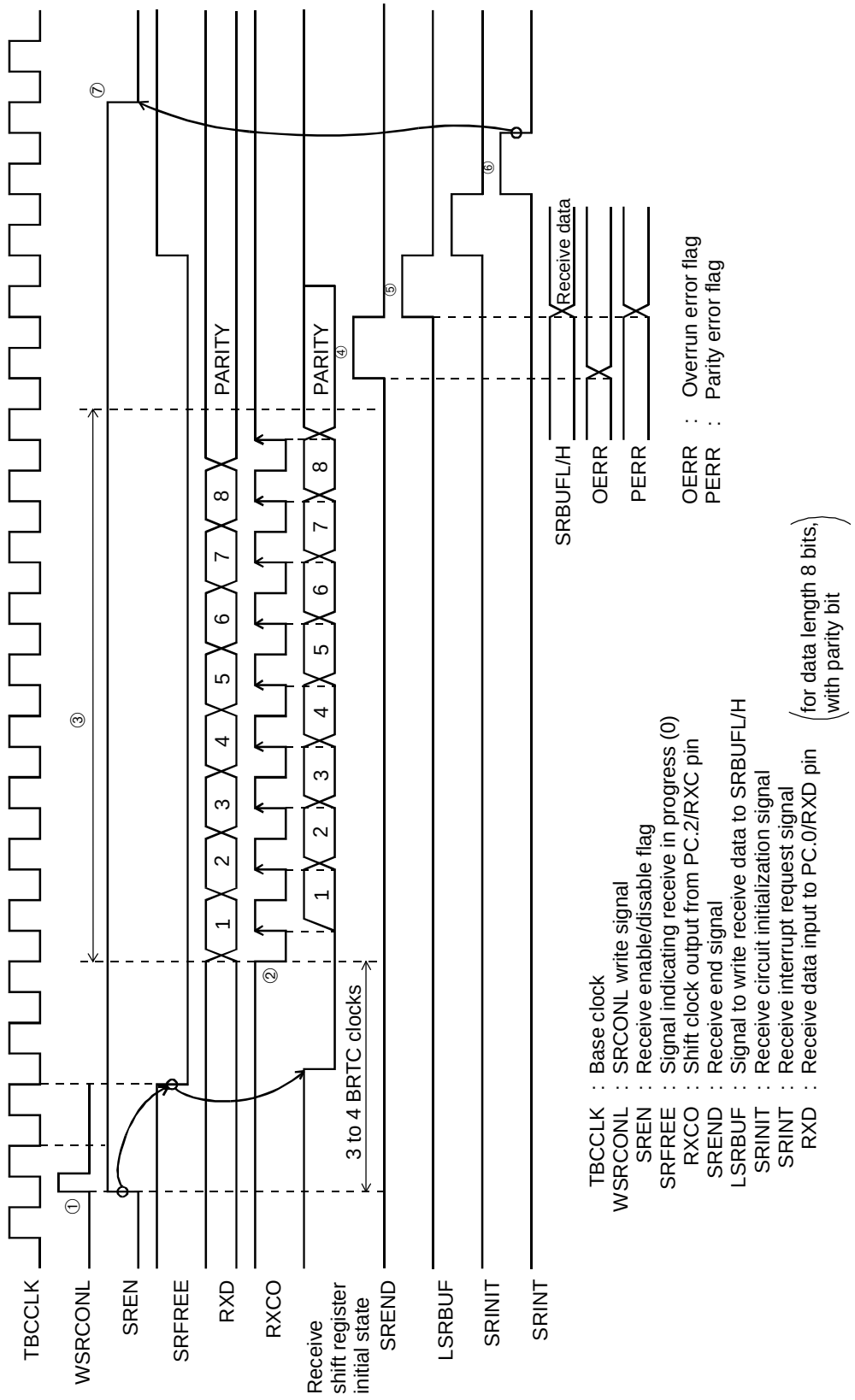


Figure 12-9 Synchronous Internal Clock Mode Receive Timing Chart

(3) Synchronous external clock mode

The synchronous external clock mode is selected by setting SRMOD (bit 0 of SRCON0) to "1" and SRCLK (bit 0 of SRCON1) to "0".

Figure 12-10 is the receive timing chart for the synchronous external clock mode.

The synchronous external clock receive procedure is indicated below.

First set the receive format (data bit length, parity bit, etc.) in SRCON1 and SRCON0.

① Set SREN (bit 3 of SRCON0) to "1" (receive enable).

② At the positive edge of the receive shift clock input through PC.2/RXC pin, the receive data from PC.0/RXD pin is written to the receive register.

③ Receive operation ends.

If an overrun error occurs the OERR flag in status register (SSTAT) is set to "1".

④ Received data is transferred to SRBUFL/H.

If a parity error occurs, the PERR flag of status register (SSTAT) is set to "1".

⑤ The serial port receive interrupt request signal (SRINT) is generated.

⑥ At the negative edge of SRINT, SREN is reset to "0".

Repeat step ① the required number of times.

In the synchronous external clock mode the receive baud rate is determined by the external clock (RXCI). Allow at least five clocks (approx. 153 µs) of TBCCLK between the time the receive is enabled (SREN = "1") and the time the external clock (RXCI) is input.

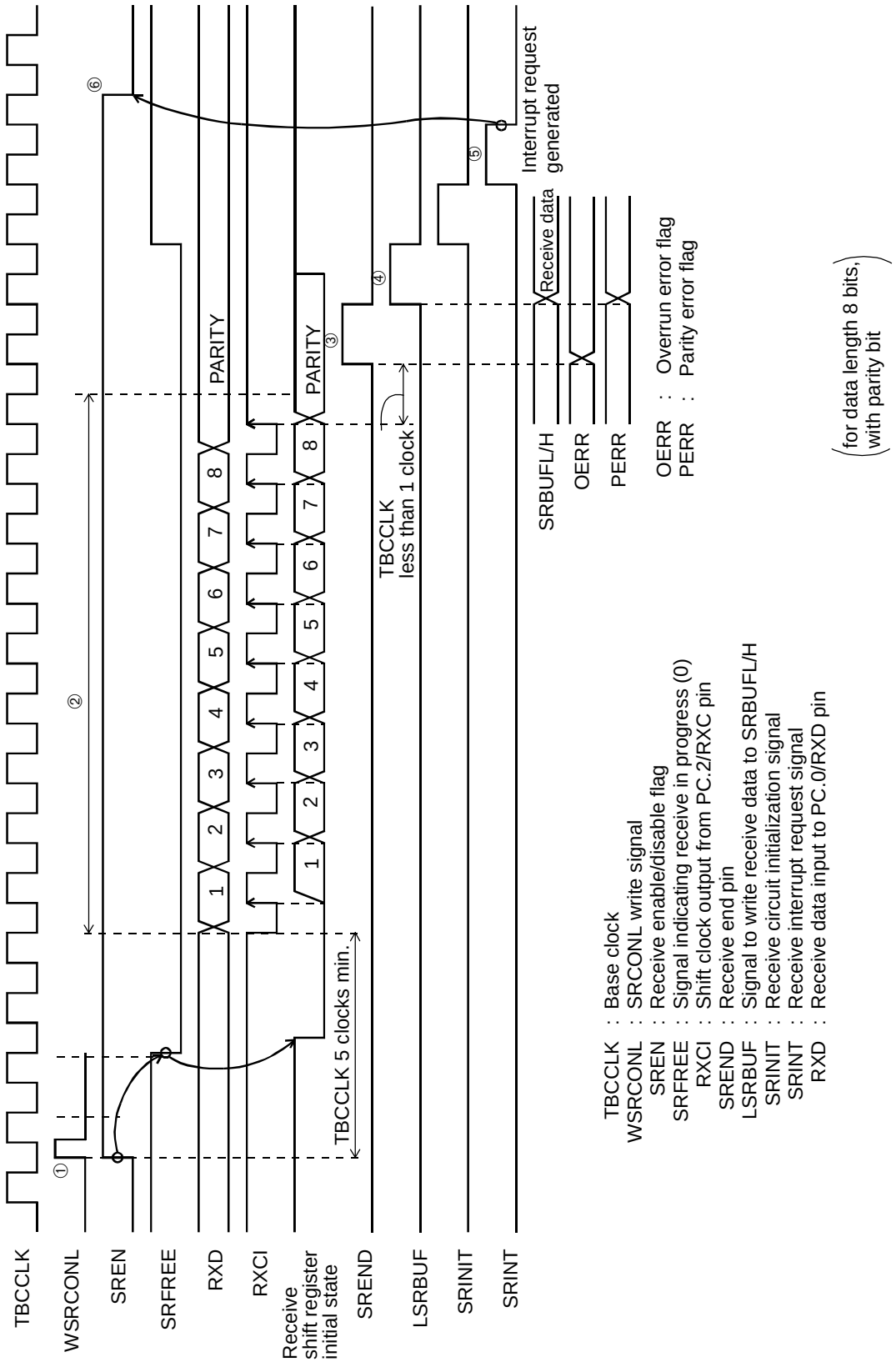


Figure 12-10 Synchronous External Clock Mode Receive Timing Chart

12.5 Send/Receive Data LSB/MSB First Select

Either LSB first or MSB first for send can be selected by setting STLMB (bit 3 of STCON1).

Either LSB first or MSB first for receive can be selected by setting SRLMB (bit 3 of SRCON1).

12.5.1 Selecting Send Data LSB/MSB First

Set STLMB (bit 3 of STCON1) to "0" to select LSB first for send.

The correspondence between LSB first send data and the send buffer register bit is shown in Figure 12-11. In this case, the LSB is TB0 (bit 0 of STBUFL).

Set STLMB to "1" to send the MSB first.

The correspondence between MSB first send data and the send buffer register bit is shown in Figure 12-12. In this case, the MSB is TB7 (bit 3 of STBUFH).

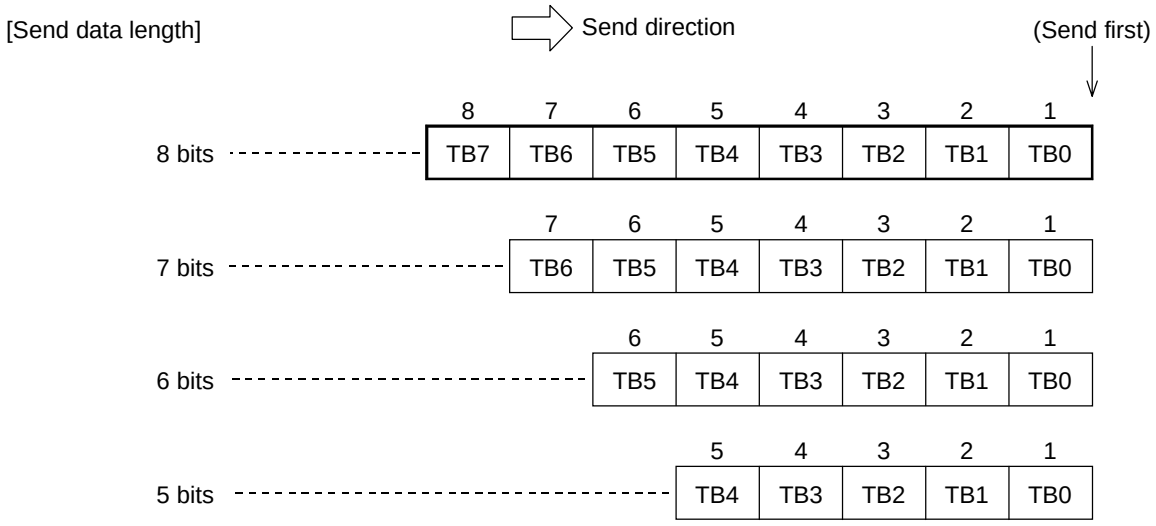


Figure 12-11 Correspondence Between LSB First Send Data and Send Buffer Register

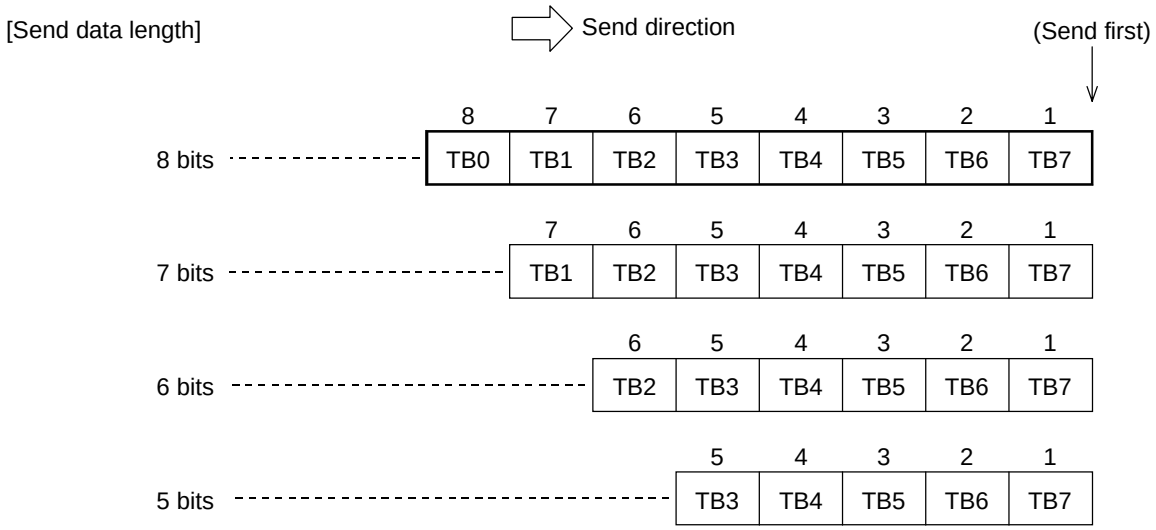


Figure 12-12 Correspondence Between MSB First Send Data and Send Buffer Register

12.5.2 Selecting Receive Data LSB/MSB First

When the LSB is first in receive data, set SRLMB (bit 3 of SRCON1) to "0".

If the MSB is first, set SRLMB to "1".

The correspondence between receive data and SRBUFL/H bits for LSB first receive is shown in Figure 12-13, and for MSB first receive in Figure 12-14.

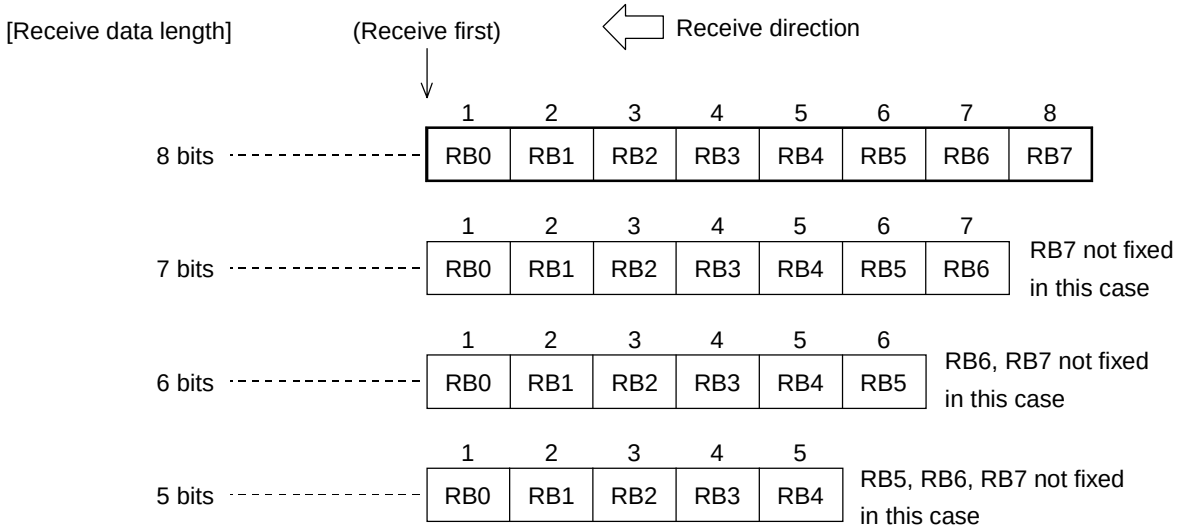


Figure 12-13 Correspondence Between LSB First Receive Data and Receive Buffer Register

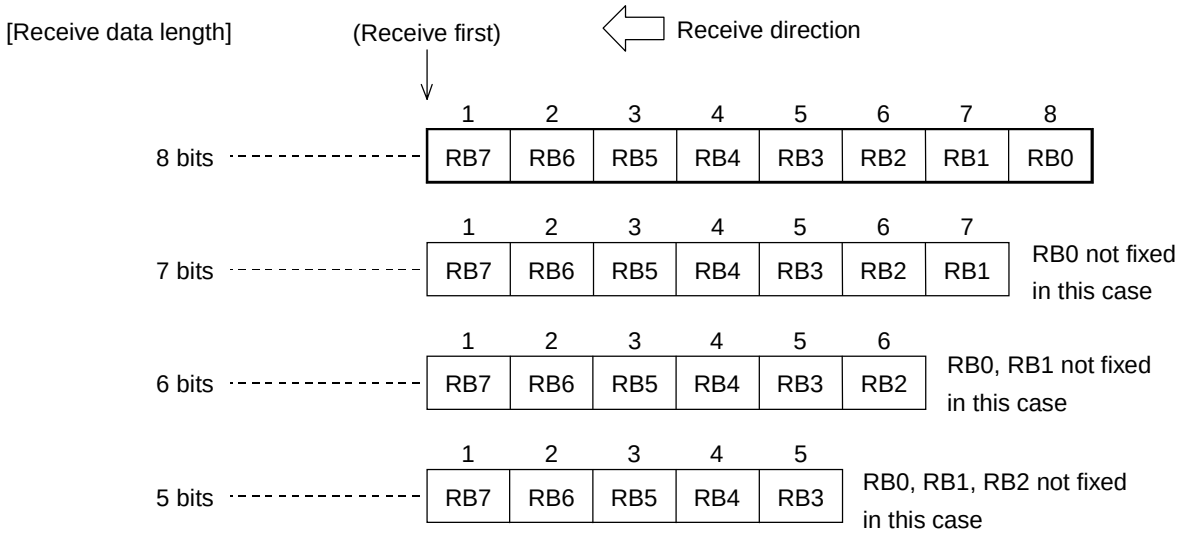


Figure 12-14 Correspondence Between MSB First Receive Data and Receive Buffer Register

Chapter 13

LCD Driver (LCD)

13. LCD Driver (LCD)

13.1 Overview

The ML63611 has built in it a segment type LCD driver (LCD) with 64 outputs.

The LCD driver section consists of a segment register, the display control register 0 (DSPCON0), the display control register 1 (DSPCON1), the display contrast register (DSPCNT), the LCD driver circuits for 64 outputs, and the bias generator circuit (BIAS).

The OPTION B and OPTION D use a voltage regulator circuit for the LCD, but the OPTION A and OPTION C do not use a voltage regulator.

The segment register is a register that can be used by selecting with the mask option. Even when using the segment register as the data area, it is necessary to specify that segment register in the mask option table. The pins of the part of the segment register used as the data area will still be outputting the segment waveforms, and hence should be left open. The allocation of the segment drivers of the different bits of the segment register can be specified freely by the mask option. For details of the mask option settings, refer to Section 1.3, "Mask Options" and the "MOGTOOL Mask Option Generator User's Manual".

DSPCON0 is the register for selecting the bias mode, power down mode, all-ON mode, and the display mode.

DSPCON1 is the register for selecting the duty ratio (1/1 to 1/4).

DSPCNT is the register for selecting the contrast value. However, this will not be effective in the OPTION A and OPTION C.

The LCD driver circuits consist of the segment driver circuits and the common driver circuits, with a total of 64 outputs L0 to L63. Among these driver circuits, a total of 64 circuits, L0 to L3 and L36 to L39, can be selected to be COM pins (COM1 to COM4) by making mask option settings for each bit. In addition, the four bits L32 to L35 can be selected to form an output port in 4-bit unit. The relationship among the duty ratio, bias value, and the maximum number of segments is as follows:

During 1/4 duty and 1/3 bias: 240 segments
During 1/2 duty and 1/2 bias: 124 segments

The OPTION B and OPTION D use a voltage regulator circuit (V/R3) that generates the bias generation reference voltage for the LCD driver. BIAS is a circuit that multiplies the output voltage (V_{DD1} = 0.95 V) of V/R3 and generates the different bias voltages when external capacitors are connected.

The OPTION A and OPTION C do not use the voltage regulator circuit (V/R3) that generates the bias generation reference voltage for the LCD driver. BIAS is a circuit that multiplies the power supply voltage and generates the different bias voltages when external capacitors are connected. In the OPTION A, short the pin V_{DD1} to V_{DD}; in the OPTION C, short the pin V_{DD2} to V_{DD}.

Note:

The OPTION B and OPTION D can keep the display density of the LCD panel constant even if the battery voltage decreases, while in the case of the OPTION A and OPTION C the display density thins down as the battery voltage decreases.

13.2 LCD Driver Configuration

Figure 13-1 shows the LCD driver configuration of the OPTION B and OPTION D; Figure 13-2, the LCD driver configuration of the OPTION A; and Figure 13-3, the LCD driver configuration of OPTION C. Figures 13-4, 13-5, and 13-6 show the peripheral circuits of the LCD driver and segment register.

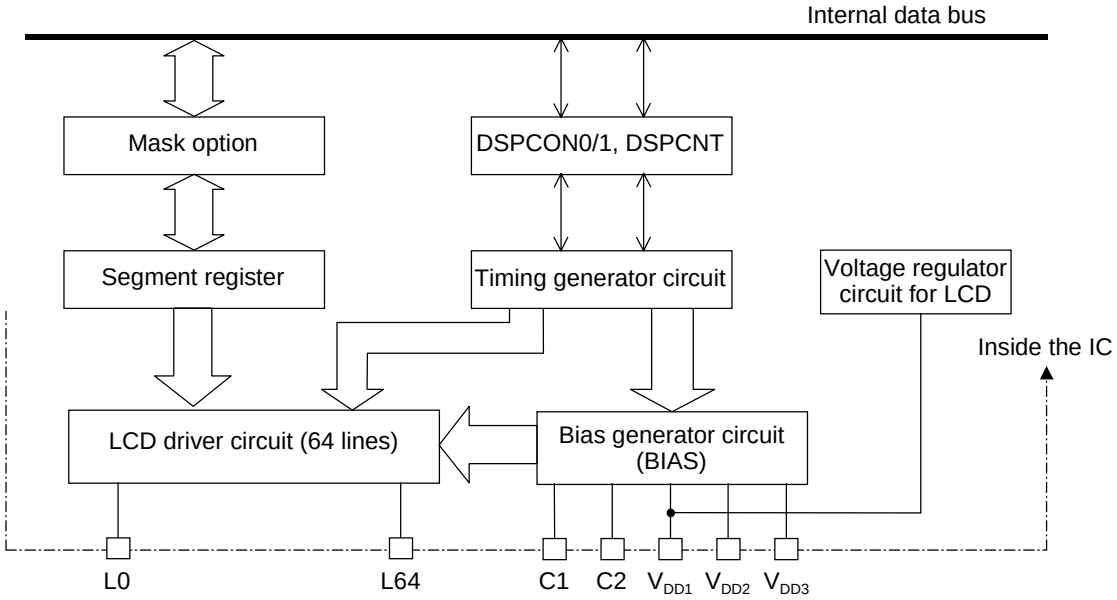


Figure 13-1 OPTION B and OPTION D LCD Driver Configuration

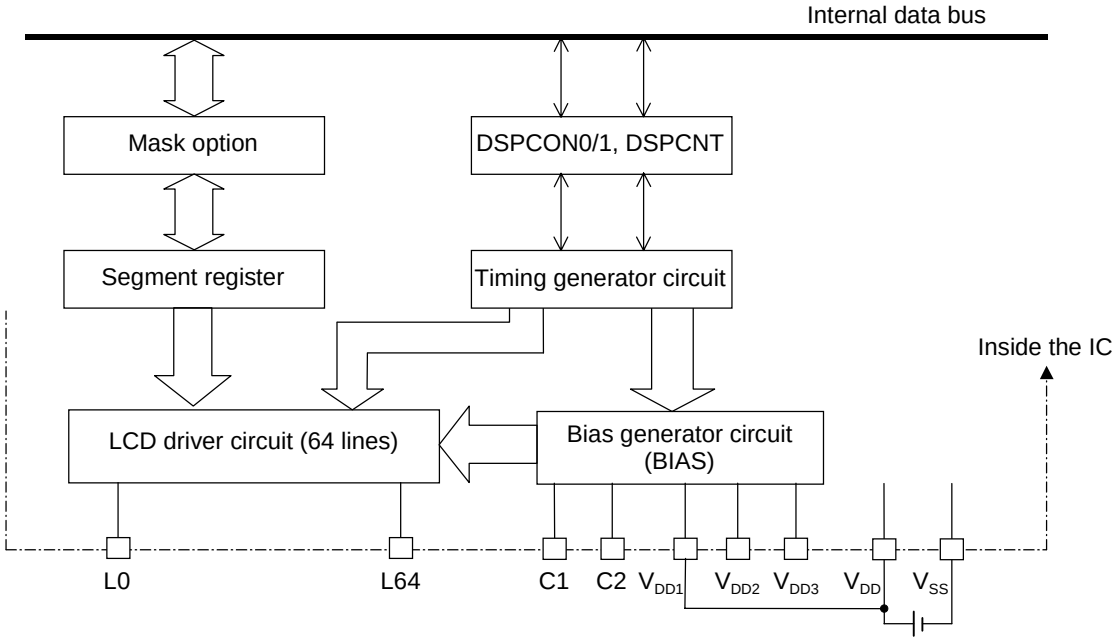


Figure 13-2 OPTION A LCD Driver Configuration

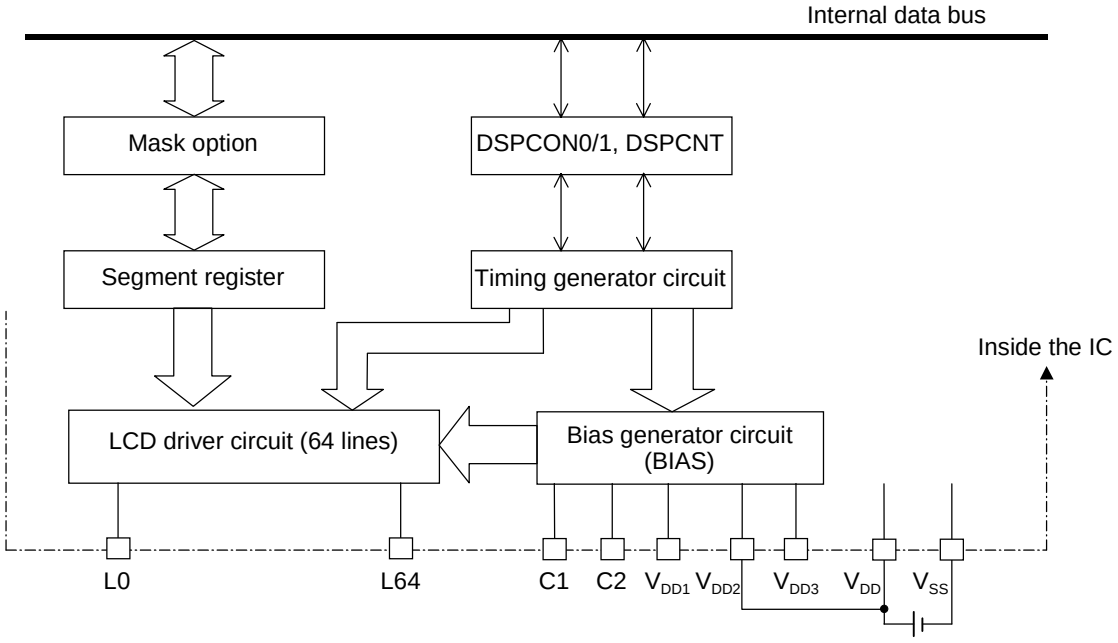


Figure 13-3 OPTION C LCD Driver Configuration

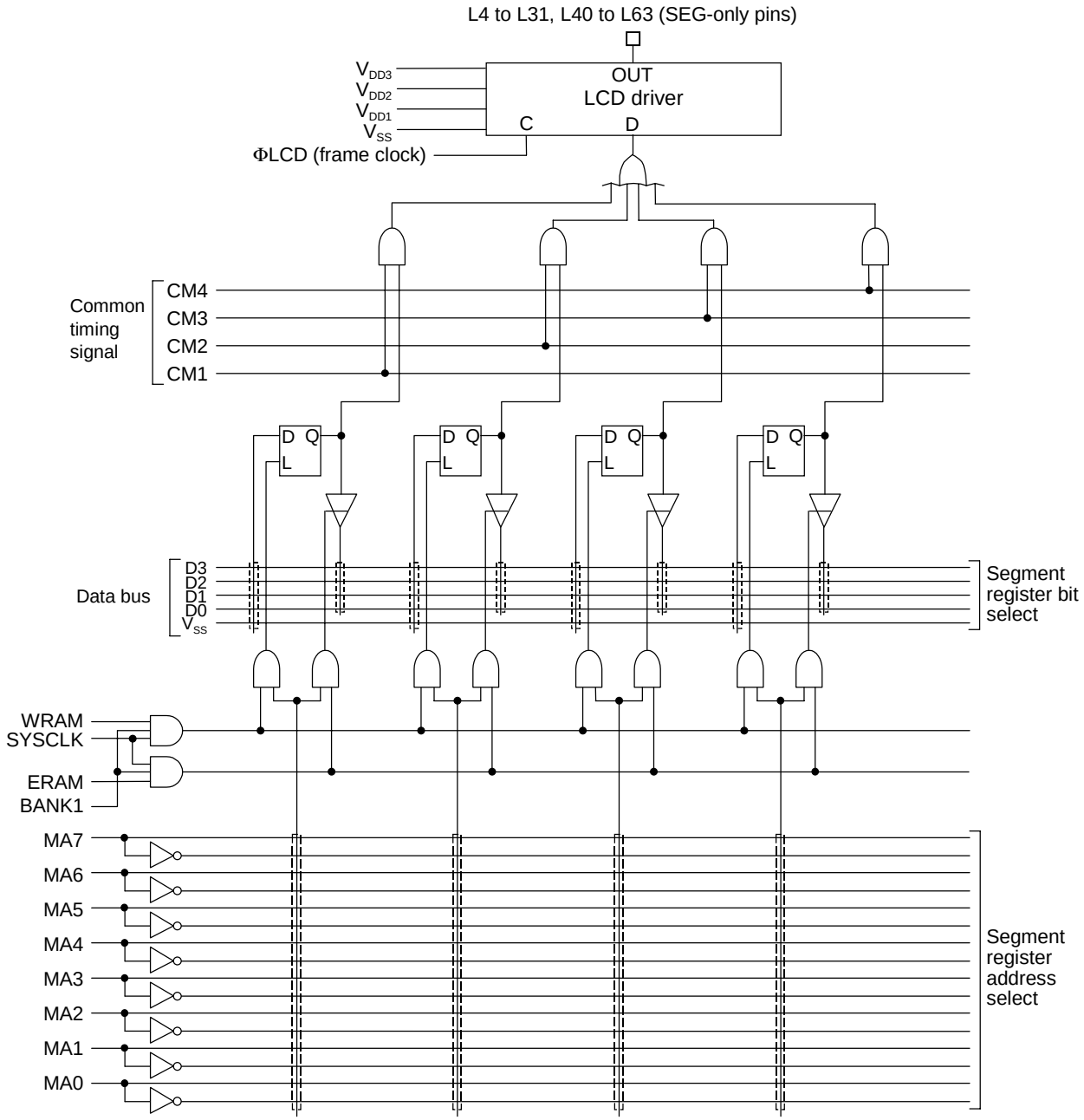


Figure 13-4 LCD Driver and Segment Register
(Circuit configuration of SEG-only pins L14 to L31 and L40 to L63)

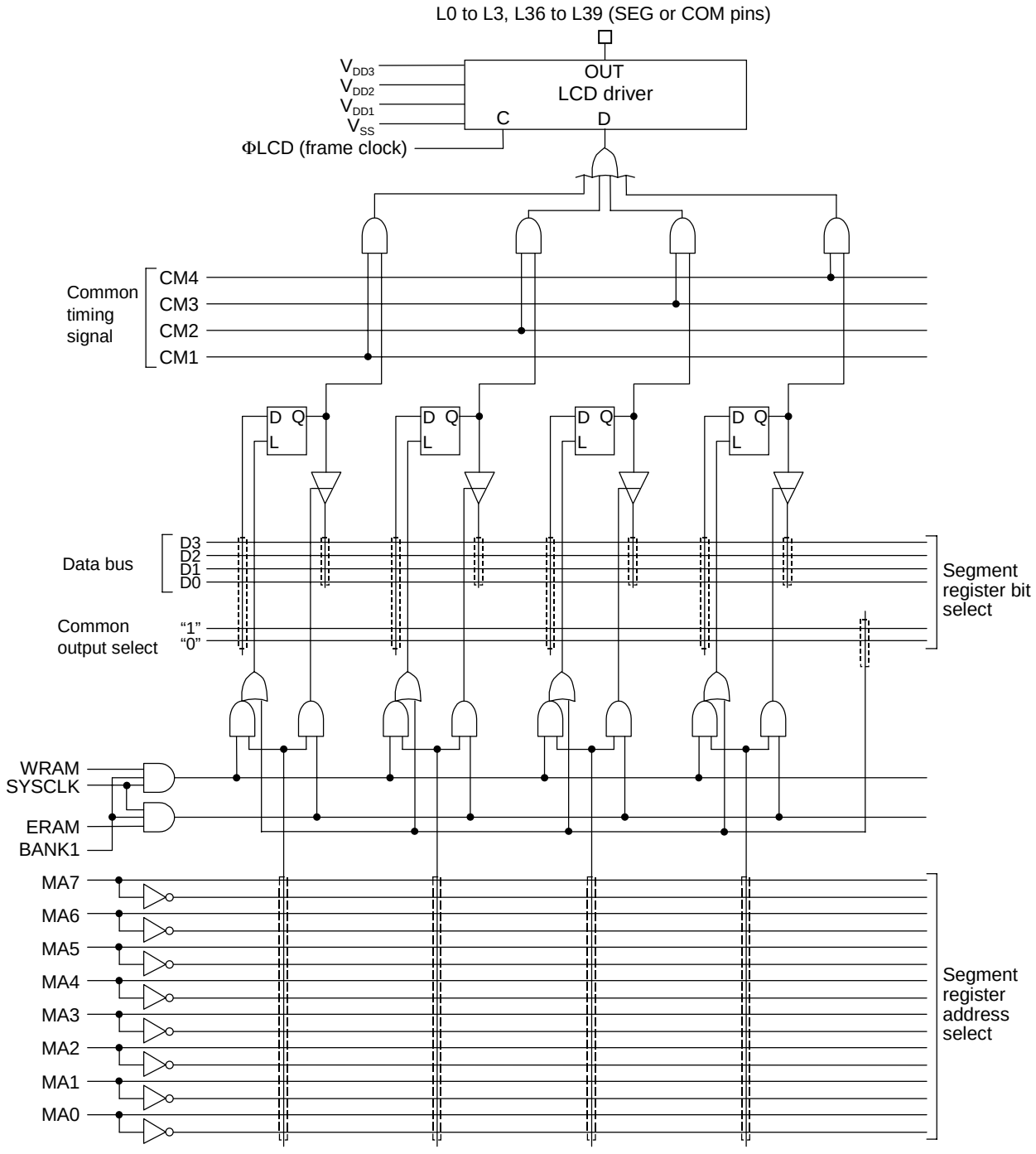


Figure 13-5 LCD Driver and Segment Register
(Circuit configuration of COM select pins L0 to L3 and L36 to L39)

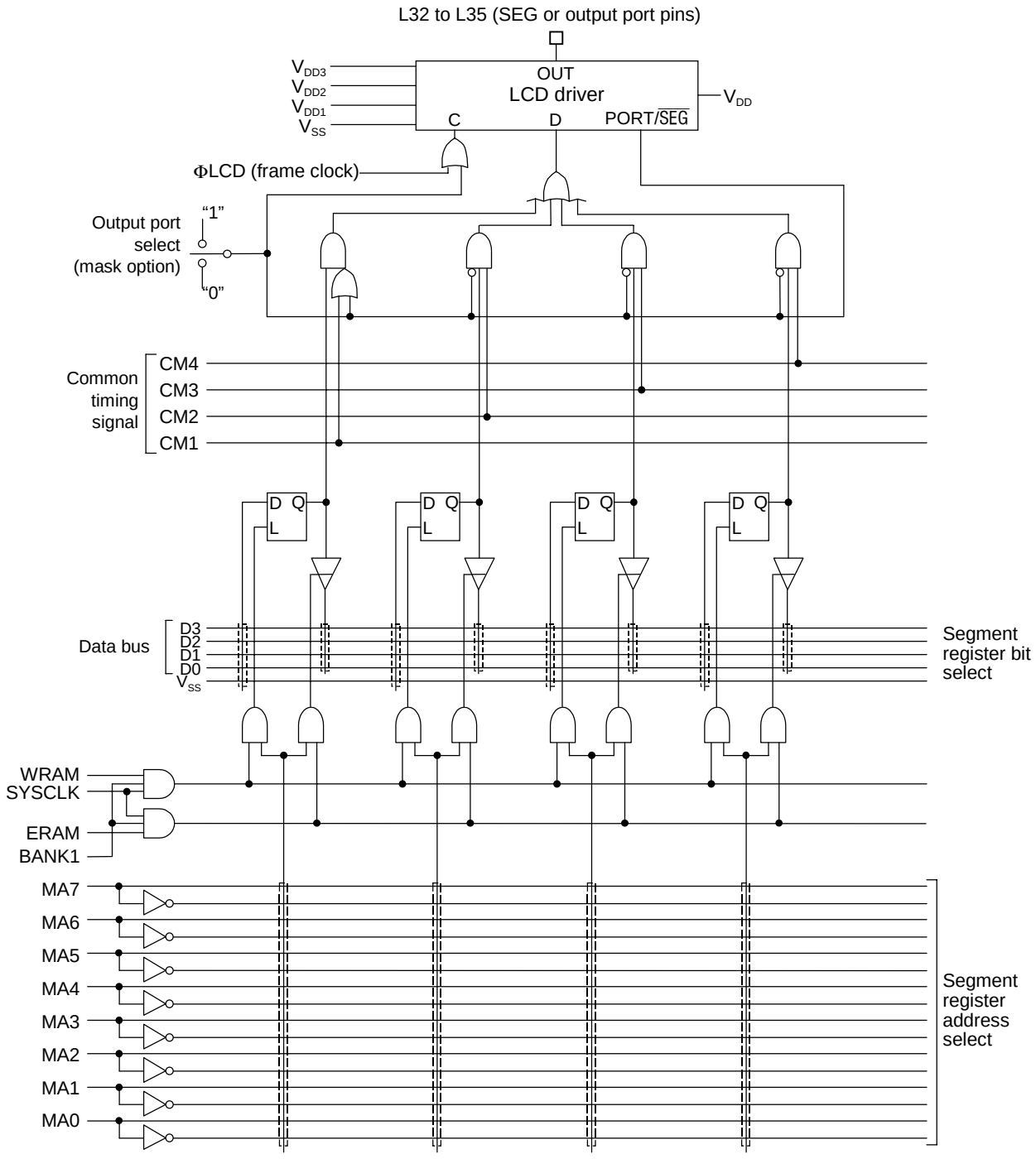
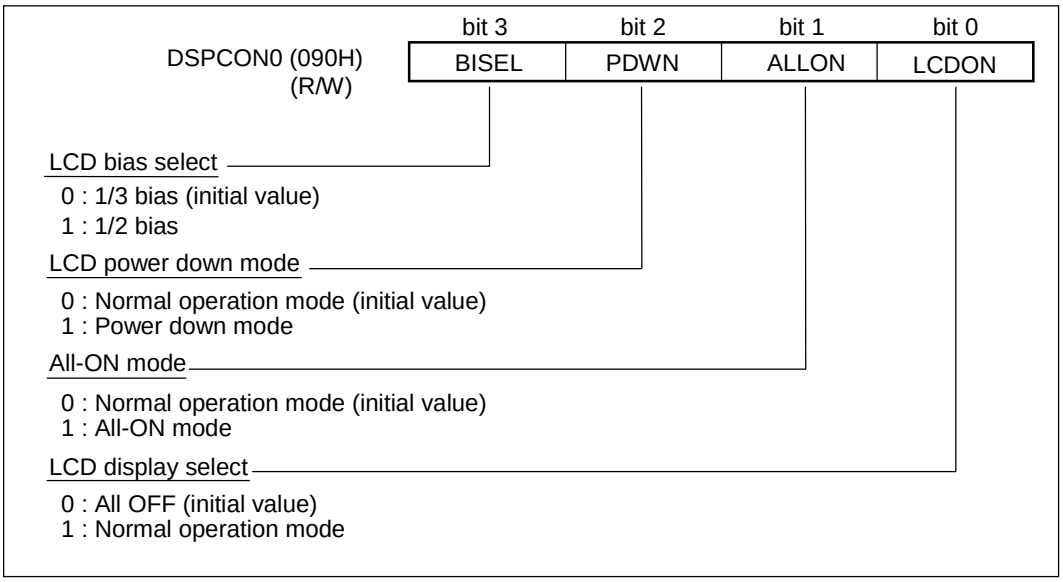


Figure 13-6 LCD Driver and Segment Register
(Circuit configuration of Output port select pins L32 to L35)

13.3 LCD Driver Registers

(1) Display control register 0 (DSPCON0)

DSPCON0 is a 4-bit special function register (SFR) controlling LCD driver operation.



bit 3: BISEL

This bit selects 1/3 or 1/2 bias.
At system reset it is "0", selecting 1/3 bias.

bit 2: PDWN

This bit selects the LCD power down mode. When PDWN is set to "1", the bias generation circuit stops its voltage lowering/raising operation and pins L0 to L63 are all set to the V_{SS} level, reducing supply current. At system reset it is cleared to "0".

bit 1: ALLON

When ALLON is set to "1" all segment drivers are turned on. The ALLON bit has priority over the LCDON bit. At system reset it is cleared to "0".

bit 0: LCDON

When the LCDON bit is set to "1", the display data in the display register is output to the segment drivers. At system reset it is cleared to "0", and all segment drivers are turned off.

(2) Display control register 1 (DSPCON1)

DSPCON1 is a 4-bit special function register (SFR) used to select the LCD driver duty.

At system reset, bits 0 and 1 of DSPCON1 are initialized to "0".

DSPCON1 (091H) (R/W)

bit 3	bit 2	bit 1	bit 0
—	—	DT1	DT0

Duty select

bit 1	bit 0	
0	0	: 1/4 duty (initial value)
0	1	: 1/1 duty
1	0	: 1/2 duty
1	1	: 1/3 duty

Note:

When 1/2 duty is selected, select 1/2 bias with the display control register 0.

(3) Display contrast register (DSPCNT)

DSPCNT is a 4-bit special function register (SFR) used to adjust display contrast.

At system reset, each bit of DSPCNT is initialized to "0".
This adjustment function is disabled in the OPTION A and OPTION C.

DSPCNT (092H) (R/W)

bit 3	bit 2	bit 1	bit 0
CN3	CN2	CN1	CN0

Contrast select

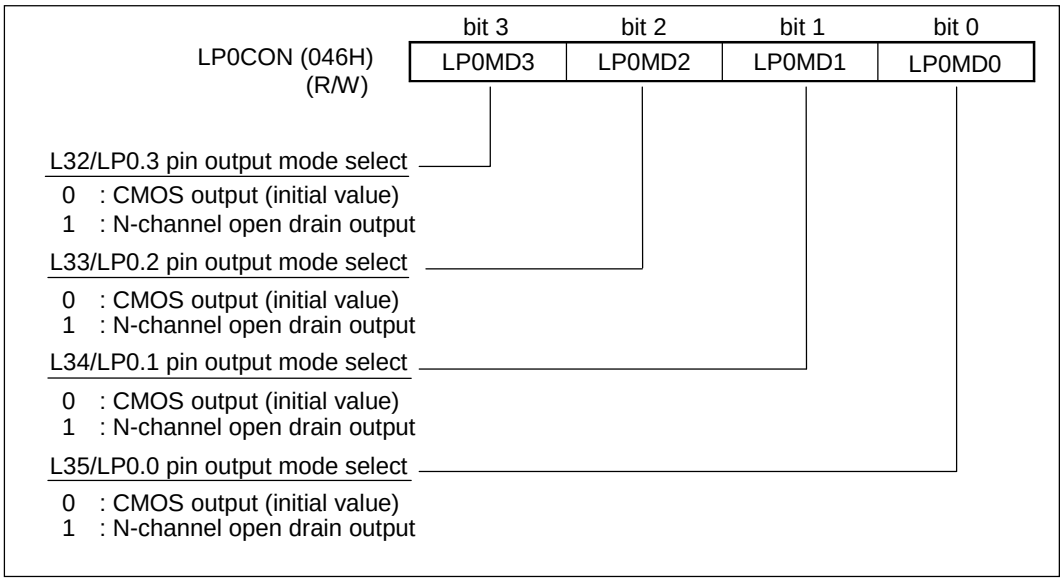
bit 3	bit 2	bit 1	bit 0	
0	0	0	0	: Light (initial value)
0	0	0	1	:
0	0	1	0	:
0	0	1	1	:
0	1	0	0	:
0	1	0	1	:
0	1	1	0	:
0	1	1	1	:
1	0	0	0	:
1	0	0	1	:
1	0	1	0	:
1	0	1	1	:
1	1	0	0	:
1	1	0	1	:
1	1	1	0	:
1	1	1	1	: Dark

Note: This function is not available in the OPTION A and OPTION C.

(4) LP0 control register (LP0CON)

LP0CON is a 4-bit special function register (SFR) used to determine the output mode when pins L32 to L35 are selected as output port pins by the mask option.

At system reset LP0CON is initialized to "0".



13.4 LCD Driver Operation

The LCD driver outputs LCD driving waveforms based on the data written to the segment register.

Use the mask option to specify the assignment of segment register address and bit and to select the segment driver and common driver.

One segment driver is assigned to four segments of the segment register. For 1/4 duty, all the four segments are used; for 1/3 duty, three segments are used; and for 1/2 duty, two segments are used.

The display duty is selected from 1/1 to 1/4 using DSPCON1. The frame frequency for each duty ratio is indicated in Table 13-1. Depending on the duty selected, the common signal (COM1 to COM4) is generated, and data written in synchronization with that common signal to the segment register is output to the segment driver. The segment driver uses bit 0 (ALLON) and bit 1 (LCDON) of the display control register 0 (DSPCON0) to control all OFF and all ON modes.

Table 13-1 Frame Frequency for Each Duty

DSPCON1			Duty	Frame frequency
DT1–0	DT1	DT0		
0H	0	0	1/4	64 Hz
1H	0	1	1/1	64 Hz
2H	1	0	1/2	64 Hz
3H	1	1	1/3	85.3 Hz

When PDWN (bit 2 of DSPCON0) is set to "1", the LCD power down mode is enabled. In the LCD power-down mode the bias generation circuit operation stops, and the L0 to L63 pins are all output at the V_{SS} level to reduce supply current.

BISEL (bit 3 of DSPCON0) selects 1/2 or 1/3 bias. When 1/2 duty is selected, select 1/2 bias with the display control register 0.

DSPCNT controls the LCD contrast of 16 tones. However, the adjustable range of the contrast varies depending on the battery voltage.

The contrast adjustment is disabled in the OPTION A and OPTION C.

Note:

When the LCD driver is not used, select the power-down mode and set all the bits of the display control register (DSPCNT) to "0" to save the supply current.

13.5 Output Port Selection by Mask Option

The four pins L32 to L35 of the LCD driver can be selected by mask option setting to form an output port in 4-bit unit.

When these pins are specified as an output port, one bit of the segment register will be allocated to one pin of the port.

Figure 13-7 shows an example of the equivalent circuit when the pins L32 to L35 are allocated collectively as an output port. In this case, the output voltage level will be V_{DD} for the "H" level output and V_{SS} for the "L" level output.

The output mode can be selected to be either N-channel open drain output or CMOS output using the register LP0CON.

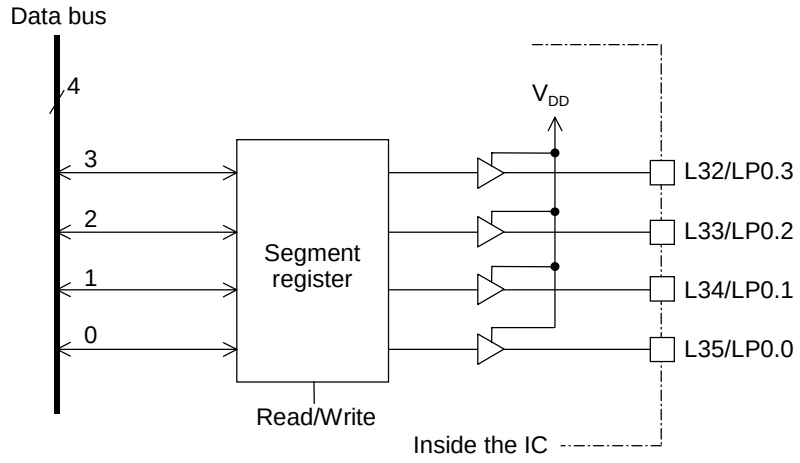


Figure 13-7 Equivalent Circuit When the LCD Driver Pins L32 to L35 Are Assigned as Output Port Pins

For details of the method of assigning the bits of the segment register for the output pins, and of specifying the output port, refer to the "MOGTOOL Mask Option Generator User's Manual".

13.6 Bias Generator Circuit (BIAS)

The bias generator circuit (BIAS) for the LCD driver in the OPTION B and OPTION D generates the different bias voltages, when external capacitors are connected, by multiplying the constant voltage output (V_{DD1} = 0.95 V) of the voltage regulator circuit for the LCD.

Further, in the OPTION A and OPTION C, the bias generator circuit (BIAS) generates the different bias voltages, when external capacitors are connected and the pin V_{DD2} (V_{DD1} for the OPTION A) is shorted to V_{DD}, by multiplying the power supply voltage V_{DD}.

(1) The configuration of the bias generator circuit of the OPTION B and OPTION D is shown in Figure 13-8.

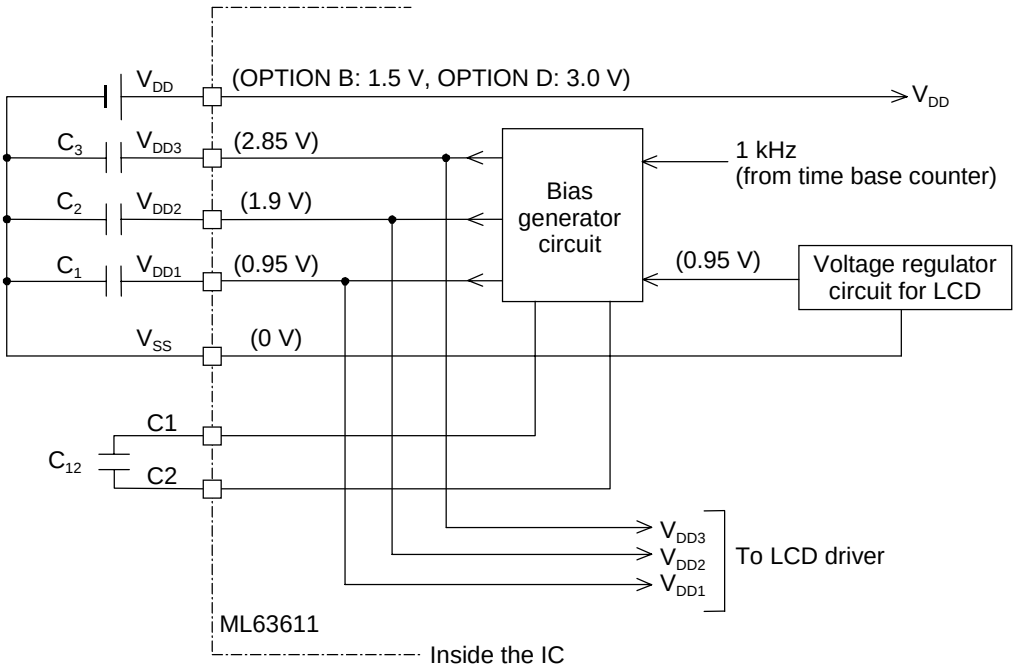


Figure 13-8 Configuration of the Bias Generator Circuit of the OPTION B and OPTION D (C_1, C_2, C_3, C_{12} = 0.1 µF)

Note:

When not using the LCD driver, the four capacitors C_1, C_2, C_3, and C_{12} are not necessary. Leave the pins V_{DD1}, V_{DD2}, V_{DD3}, C1, and C2 open.
Further, when using the LCD driver at 1/2 duty, the capacitor C_2 is not necessary. Leave the pin V_{DD2} open in this case.

(2) Figure 13-9 shows the OPTION A bias generator configuration.

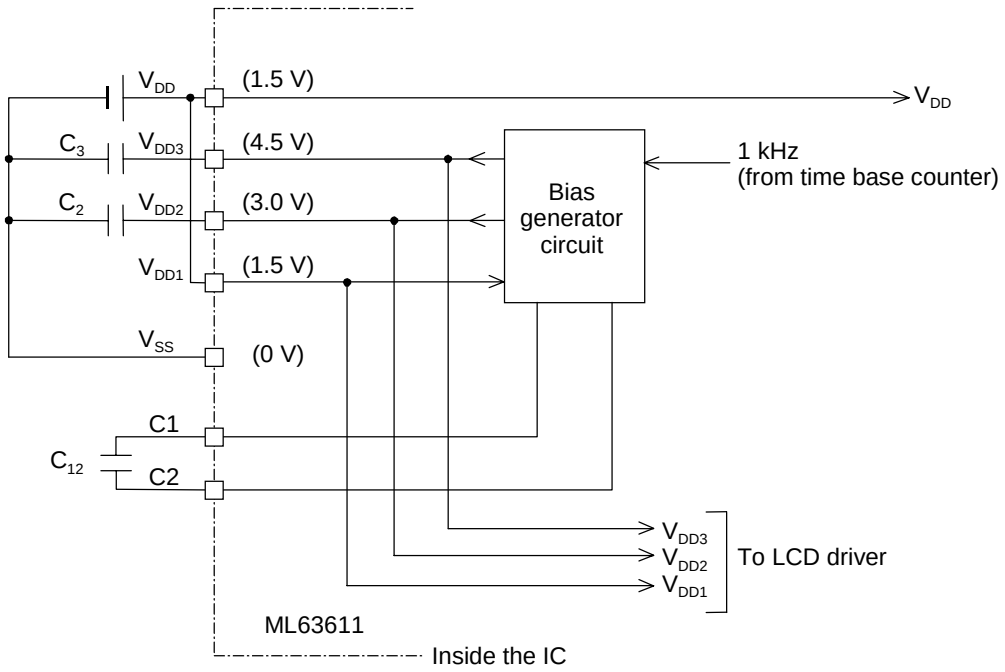


Figure 13-9 Configuration of the Bias Generator Circuit of the OPTION A (C_2, C_3, C_{12} = 0.1 µF)

Note:

If the LCD driver is not used, capacitors C_2, C_3, and C_{12} are not necessary. Leave each of the V_{DD1}, V_{DD2}, V_{DD3}, C1, and C2 pins open.

(3) Figure 13-10 shows the OPTION C bias generator configuration.

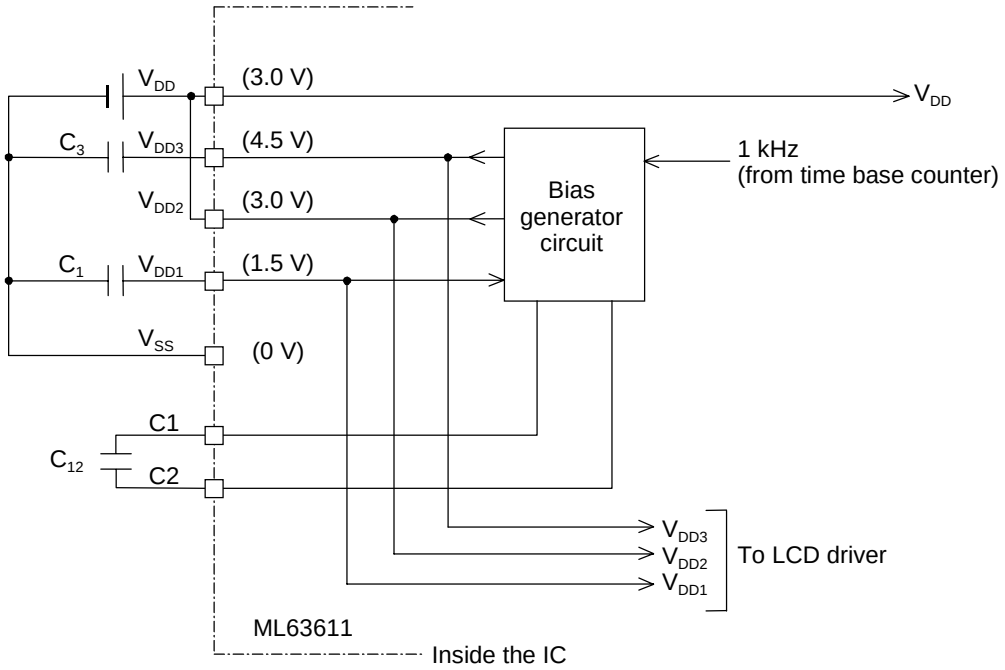


Figure 13-10 Configuration of the Bias Generator Circuit of the OPTION C (C_1, C_3, C_{12} = 0.1 µF)

Note:

If the LCD driver is not used, capacitors C_1, C_3, and C_{12} are not necessary. Leave each of the V_{DD1}, V_{DD2}, V_{DD3}, C1, and C2 pins open.

Table 13-2 OPTION B Display Contrast Adjusting Voltages (V_{DD1})

(Ta = 25°C, V_{SS} = 0 V)

DSPCNT					V_{DD1} Voltage (V)			Contrast
CN3–0	CN3	CN2	CN1	CN0	Min.	Typ.	Max.	
0H	0	0	0	0	0.90	0.95	1.00	Light
1H	0	0	0	1	0.93	0.98	1.03	
2H	0	0	1	0	0.96	1.01	1.06	
3H	0	0	1	1	0.99	1.04	1.09	
4H	0	1	0	0	1.02	1.07	1.12	
5H	0	1	0	1	1.05	1.10	1.15	
6H	0	1	1	0	1.08	1.13	1.18	
7H	0	1	1	1	1.11	1.16	1.21	
8H	1	0	0	0	1.14	1.19	1.24	Dark

Table 13-3 OPTION D Display Contrast Adjusting Voltages (V_{DD1})

(Ta = 25°C, V_{SS} = 0 V)

DSPCNT					V_{DD1} Voltage (V)			Contrast
CN3–0	CN3	CN2	CN1	CN0	Min.	Typ.	Max.	
0H	0	0	0	0	0.90	0.95	1.00	Light
1H	0	0	0	1	0.93	0.98	1.03	
2H	0	0	1	0	0.96	1.01	1.06	
3H	0	0	1	1	0.99	1.04	1.09	
4H	0	1	0	0	1.02	1.07	1.12	
5H	0	1	0	1	1.05	1.10	1.15	
6H	0	1	1	0	1.08	1.13	1.18	
7H	0	1	1	1	1.11	1.16	1.21	
8H	1	0	0	0	1.14	1.19	1.24	
9H	1	0	0	1	1.17	1.22	1.27	
0AH	1	0	1	0	1.20	1.25	1.30	
0BH	1	0	1	1	1.23	1.28	1.33	
0CH	1	1	0	0	1.26	1.31	1.36	
0DH	1	1	0	1	1.29	1.34	1.39	
0EH	1	1	1	0	1.32	1.37	1.42	
0FH	1	1	1	1	1.35	1.40	1.45	Dark

Table 13-4 OPTION B and OPTION D Display Contrast Adjusting Voltages (V_{DD2}, V_{DD3})

(V_{SS} = 0 V)

BISEL	Mode	Power supply	Voltage (V)		
			Min.	Typ.	Max.
0	1/3 bias	V_{DD2}	Typ. – 0.3	$2 \times V_{DD1}$*	Typ. + 0.3
		V_{DD3}	Typ. – 0.4	$3 \times V_{DD1}$*	Typ. + 0.4
1	1/2 bias	V_{DD2}	Typ. – 0.2	V_{DD1}*	Typ. + 0.2
		V_{DD3}	Typ. – 0.3	$2 \times V_{DD1}$*	Typ. + 0.3

*: Indicates the typical V_{DD1} value in Table 13-2.

Table 13-5 OPTION A Display Contrast Adjusting Voltages (V_{DD1}, V_{DD2}, V_{DD3})

(V_{SS} = 0 V)

BISEL	Mode	Power supply	Voltage (V)		
			Min.	Typ.	Max.
0	1/3 bias	V_{DD1}	Typ. – 0.1	V_{DD}	Typ. + 0.1
		V_{DD2}	Typ. – 0.2	$2 \times V_{DD}$	Typ. + 0.2
		V_{DD3}	Typ. – 0.3	$3 \times V_{DD}$	Typ. + 0.3
1	1/2 bias	V_{DD1}	Typ. – 0.1	V_{DD}	Typ. + 0.1
		V_{DD2}	Typ. – 0.1	V_{DD}	Typ. + 0.1
		V_{DD3}	Typ. – 0.2	$2 \times V_{DD}$	Typ. + 0.2

Table 13-6 OPTION C Display Contrast Adjusting Voltages (V_{DD1}, V_{DD2}, V_{DD3})

(V_{SS} = 0 V)

BISEL	Mode	Power supply	Voltage (V)		
			Min.	Typ.	Max.
0	1/3 bias	V_{DD1}	Typ. – 0.1	$1/2 \times V_{DD}$	Typ. + 0.1
		V_{DD2}	Typ. – 0.1	V_{DD}	Typ. + 0.1
		V_{DD3}	Typ. – 0.2	$3/2 \times V_{DD}$	Typ. + 0.2
1	1/2 bias	V_{DD1}	Typ. – 0.1	$1/2 \times V_{DD}$	Typ. + 0.1
		V_{DD2}	Typ. – 0.1	V_{DD}	Typ. + 0.1
		V_{DD3}	Typ. – 0.1	V_{DD}	Typ. + 0.1

Notes:

The contrast adjusting voltages in the OPTION B and OPTION D are in the following range:

- OPTION B
  (Example) Battery voltage = 1.3 to 1.7 V ..... 0.95 V min. to 1.19 V max.
- OPTION D
  When the halver circuit is used:
  (Example) Battery voltage = 2.4 V ..... 0.95 V min. to 1.00 V max.
  Battery voltage = 3.0 V ..... 0.95 V min. to 1.30 V max.
  Battery voltage = 3.2 V ..... 0.95 V min. to 1.40 V max.
  When the halver circuit is not used (battery voltage = 1.8 to 2.4 V):
  ..... 0.95 V min. to 1.40 V max.
- Contrast adjusting is not available in the OPTION A and OPTION C.

13.7 LCD Driver Output Waveform

Figures 13-11 (a) and 13-11 (b) show the output waveforms for 1/4 duty and 1/3 bias, and Figures 13-12 (a) and 13-12 (b) show the output waveforms for 1/3 duty and 1/3 bias.

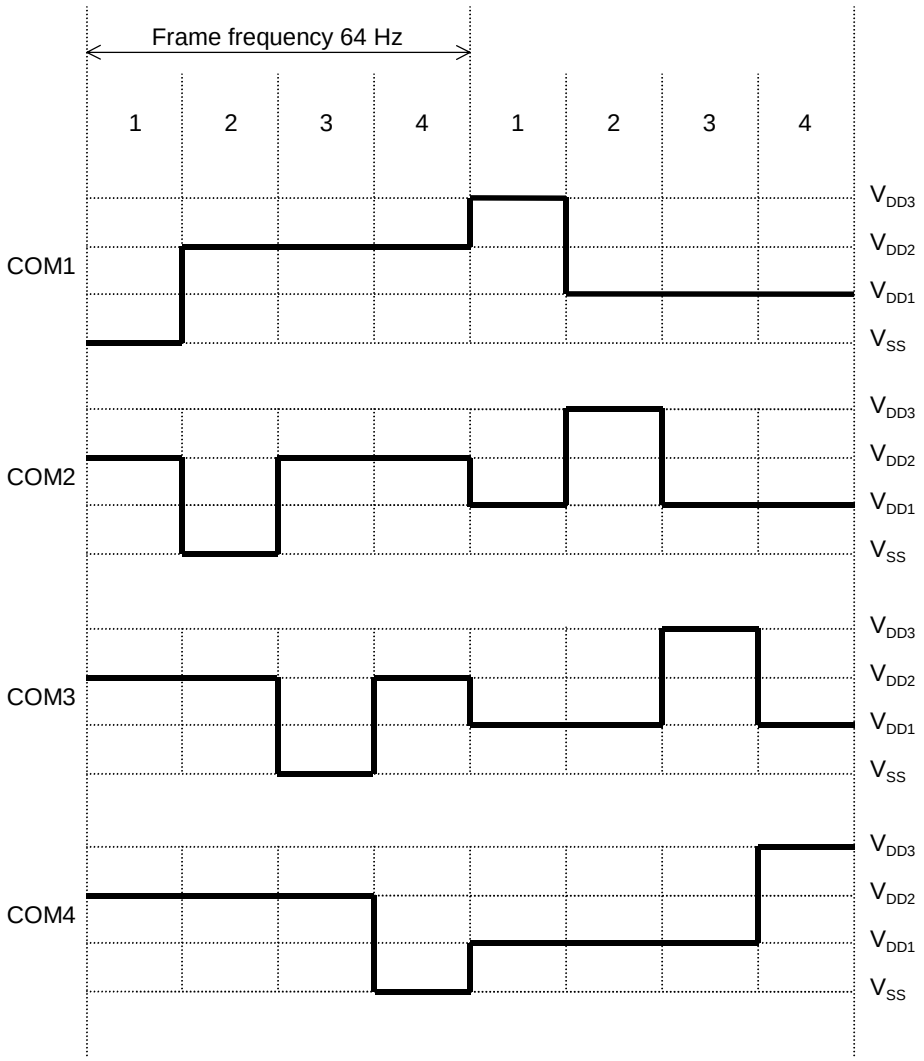


Figure 13-11 (a) 1/4 Duty, 1/3 Bias Common Output Waveform

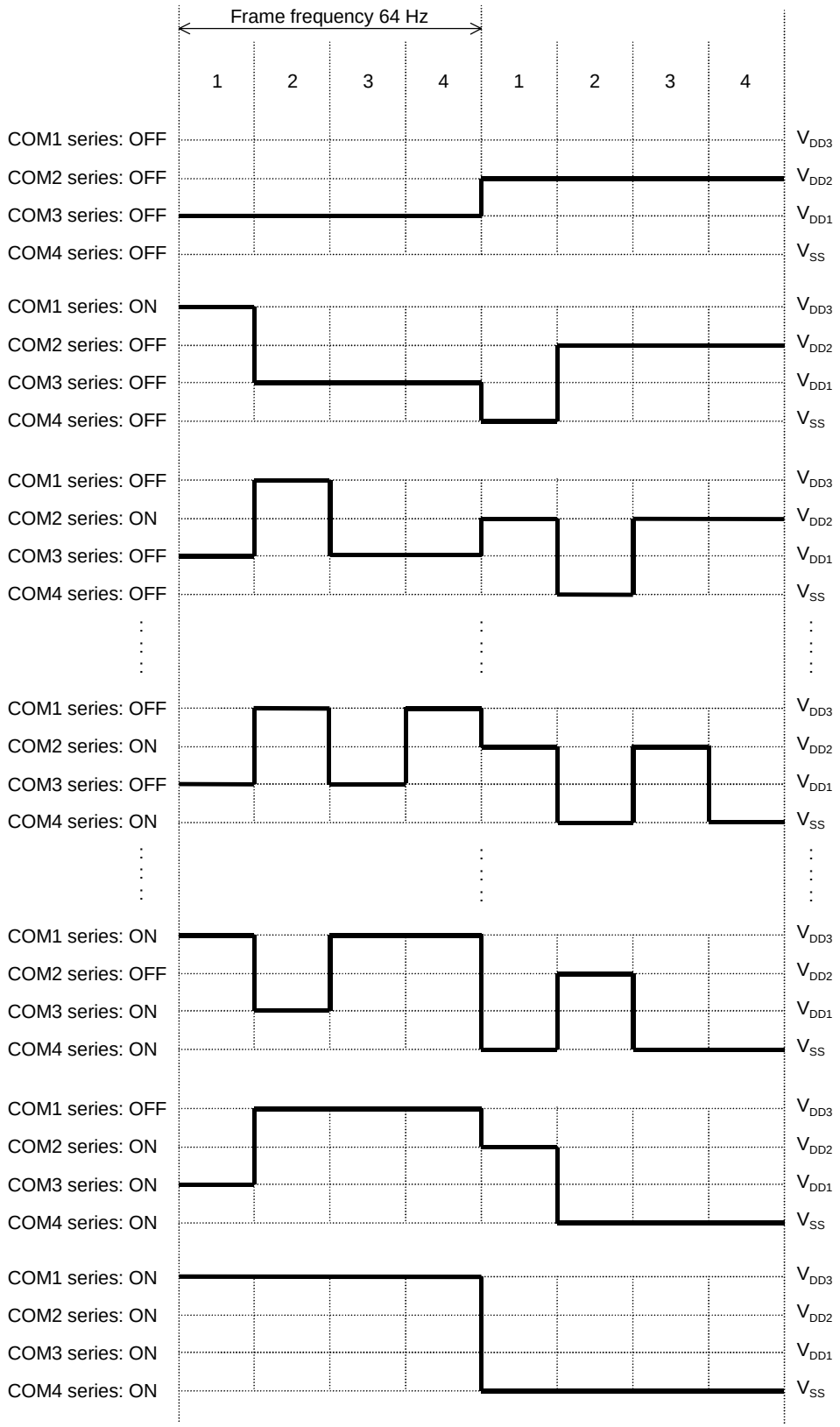


Figure 13-11 (b) 1/4 Duty, 1/3 Bias Segment Output Waveform

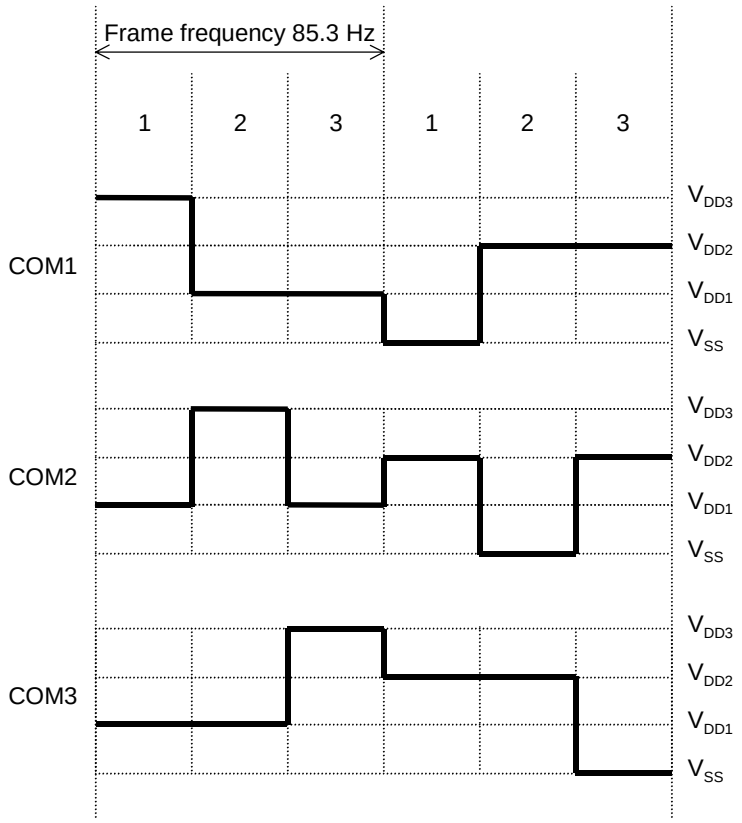


Figure 13-12 (a) 1/3 Duty, 1/3 Bias Common Output Waveform

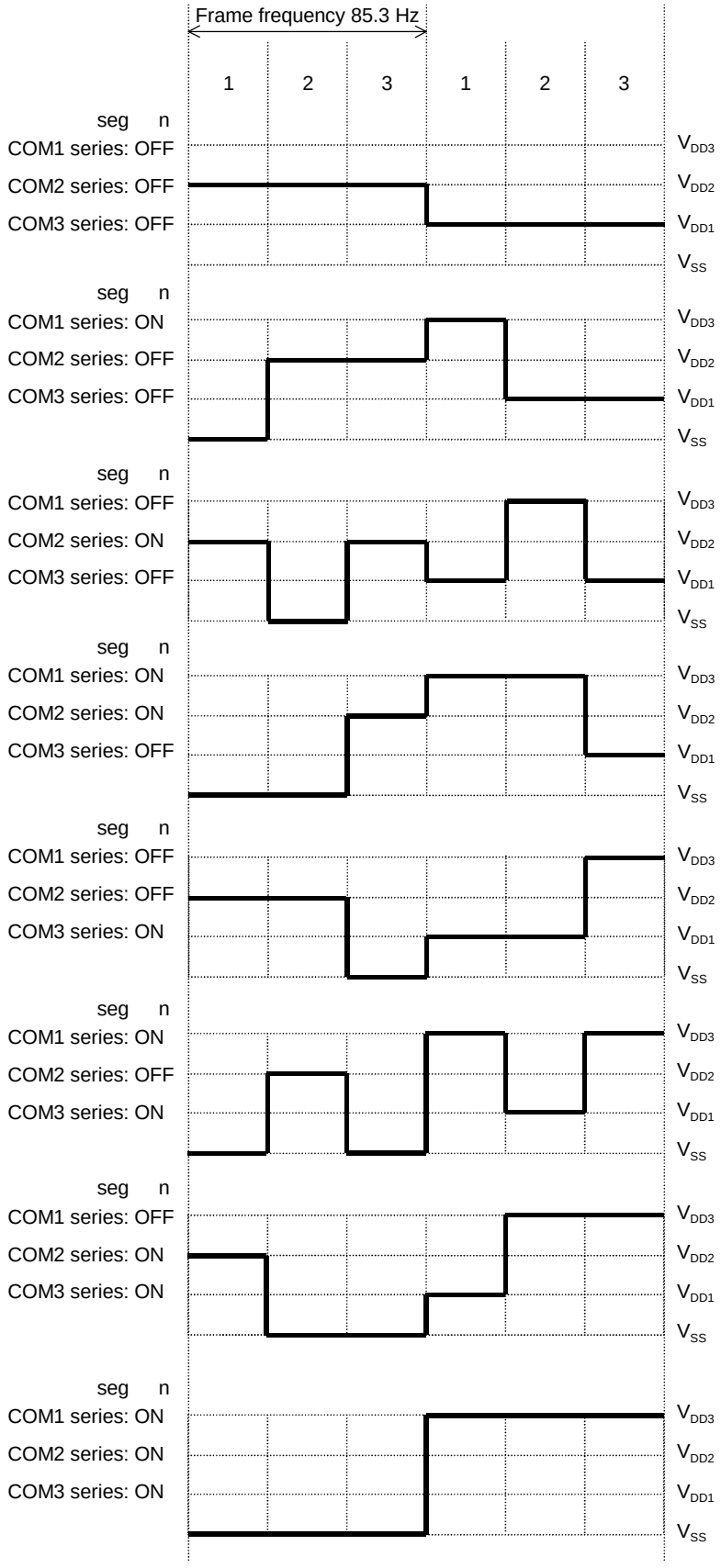


Figure 13-12 (b) 1/3 Duty, 1/3 Bias Common Output Waveform

Chapter 14

Battery Low Detect Circuit (BLD)

14. Battery Low Detect Circuit (BLD)

14.1 Overview

The OPTION C and OPTION D have an internal battery low detect circuit (BLD).

The battery low detect circuit detects when the battery voltage (supply voltage V_{DD}) falls below the judgment voltage value. Four levels of judgment voltage can be selected by the BLDCON bits.

Judgment voltage values (Ta = 25°C): 1.8 ±0.1 V, 2.0 ±0.1 V, 2.4 ±0.1 V, 2.6 ±0.1 V

Note:

- Does not apply to the OPTION A and OPTION B.
- When verifying BLD operation, the operation must be verified with an evaluation sample device.

14.2 Battery Low Detect Circuit Configuration

The battery low detect circuit consists of a judgment circuit and a judgment voltage select circuit. Figure 14-1 shows the circuit configuration.

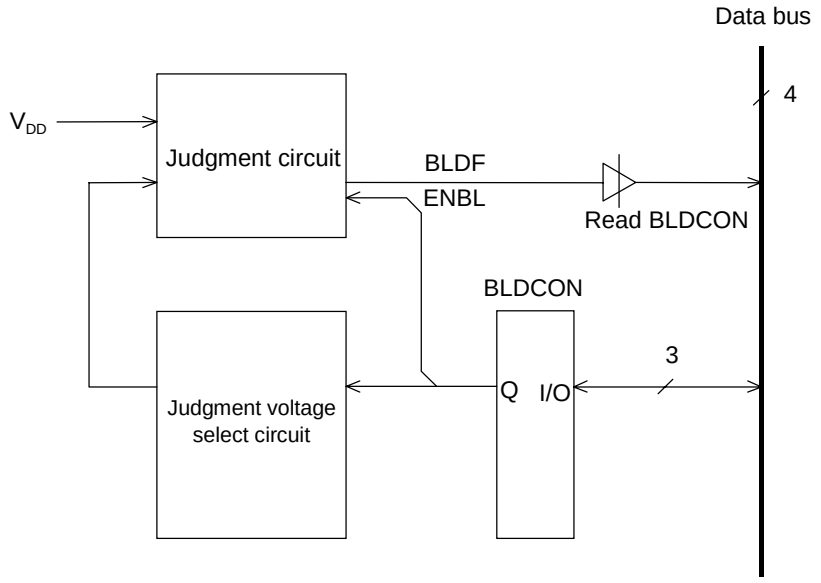


Figure 14-1 Battery Low Detect Circuit

14.3 Judgment Voltage

The value of the judgment voltage is selected by the software by setting the LD1 (bit 1 of BLDCON) and LD0 (bit 0 of BLDCON) bits.

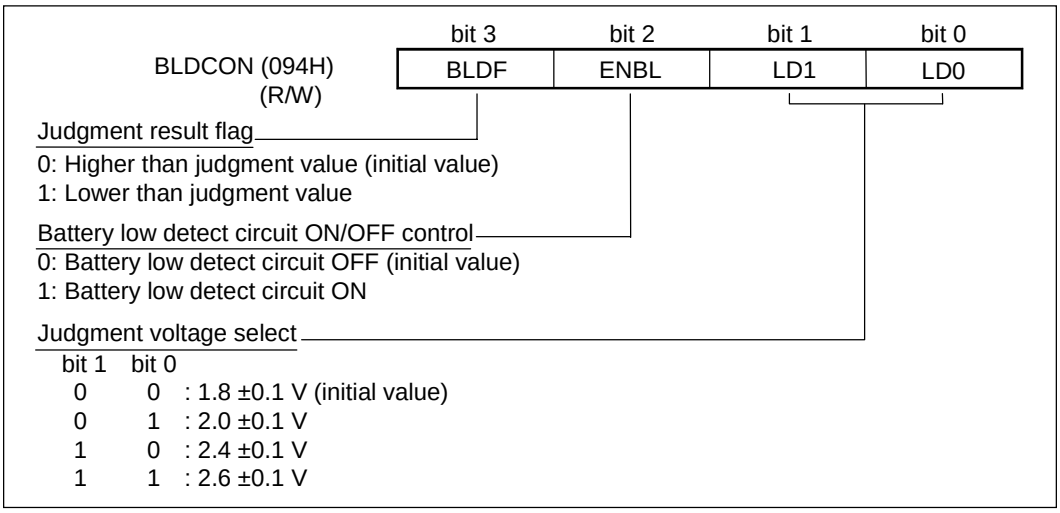
Table 14-1 lists judgment voltage and precision values.

Table 14-1 Judgment Voltage

LD1	LD0	Judgment voltage (V)	Precision (V)	Remarks
0	0	1.8	±0.1	Ta = 25°C
0	1	2.0	±0.1	Ta = 25°C
1	0	2.4	±0.1	Ta = 25°C
1	1	2.6	±0.1	Ta = 25°C

14.4 Battery Low Detect Circuit Register

- Battery low detect control register (BLDCON)
  BLDCON is a 4-bit special function register (SFR) that controls the battery low detect circuit.



bit 3: BLDF

This flag indicates the judgment result of the battery low detect circuit.
This bit is set to "1" when V_{DD} is lower than the judgment voltage selected by LD0 and LD1, and is set to "0" when V_{DD} is higher. This bit is "0" when the BLD circuit stops operation.
This bit is read-only and writes are invalid.

bit 2: ENBL

This bit turns the battery low detect circuit ON or OFF.
When ENBL is set to "1", the battery low detect circuit is ON. When ENBL is set to "0", the battery low detect circuit is OFF.
At system reset, this bit is cleared to "0".

bit 1, 0: LD1, LD0

These bits select the judgment voltage.
At system reset, these bits are cleared to "0".

14.5 Battery Low Detect Circuit Operation

The battery low circuit is turned ON or OFF by ENBL (bit 2 of BLDCON), and outputs to BLDF (bit 3 of BLDCON) the result of a comparison with the judgment voltage.

ENBL is the enable control bit for the battery low detect circuit. Setting ENBL to "1" turns ON the battery low detect circuit. Setting ENBL to "0" turns OFF the battery low detect circuit and BLD current supply drops to zero.

BLDF is the judgment result flag. If BLDF is "1", the power supply voltage is lower than the judgment voltage. If BLDF is "0", the power supply voltage is higher than the judgment voltage. BLDF is valid when ENBL is "1".

The judgment circuit of the battery low detect circuit requires time to become stable. Therefore, after setting ENBL to "1", wait at least 1 ms before reading BLDF. No load should be applied to the power supply voltage during the detection.

Figure 14-2 shows an example operation timing.

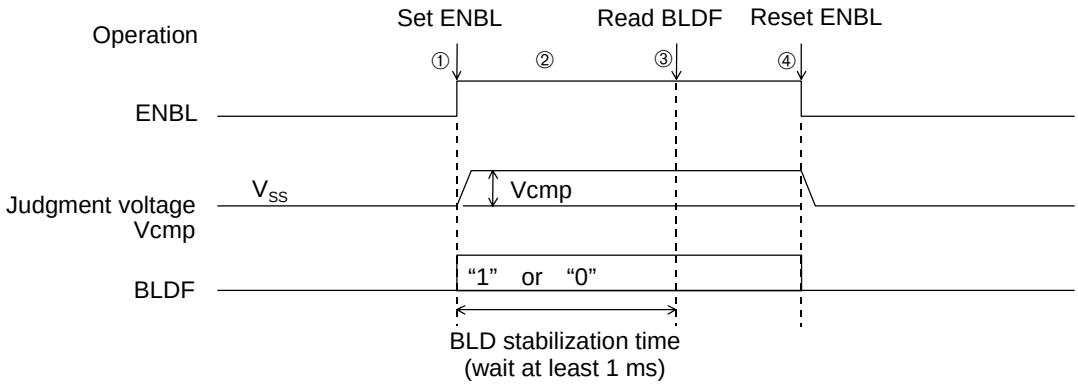


Figure 14-2 Operation Timing Example

Figure 14-2 shows the following operations.

① ENBL is set to "1" to turn ON the BLD.

② Operation starts with no load applied to power supply system and waits for BLD stabilization time interval (at least 1 ms).

③ Judgment result flag (BLDF) is read.

④ ENBL is cleared to "0".

Chapter 15

Power Supply Circuit (POWER)

15. Power Supply Circuit (POWER)

15.1 Overview

In the different voltage regulator circuits of the OPTION C and OPTION D, the battery voltage (V_{DD})*1 is halved to generate the voltage V_{HF}, which is then used to generate internally the different driving voltages. By using the output voltage of this halver circuit*2, it is possible to drive the internal circuit operations at half the battery voltage, thereby suppressing the supply current. However, the battery voltage is connected directly to the power supply of the high-speed clock generator circuit.

In the different voltage regulator circuits of the OPTION A and OPTION B, the battery voltage is connected directly to the power supply line and the different driving voltages are generated internally from this voltage.

The following three voltage regulator circuits are incorporated in the OPTION B and OPTION D:

(1) V/R1: Output voltage = 1.15 V
… The voltage regulator circuit generating the power supply voltage (V_{CH}) for the CPU, ROM, RAM, and other internal logic circuits.
(2) V/R2: Output voltage = 0.7 V
… The voltage regulator circuit generating the power supply voltage (V_{XT}) for the low-speed clock generator circuit.
(3) V/R3: Output voltage = 0.95 V
… The voltage regulator circuit generating the reference voltage (V_{DD1}) for the LCD bias generator circuit.

The following two voltage regulator circuits are incorporated in the OPTION A and OPTION C:

(1) V/R1: Output voltage = 1.15 V
… The voltage regulator circuit generating the power supply voltage (V_{CH}) for the CPU, ROM, RAM, and other internal logic circuits.
(2) V/R2: Output voltage = 0.7 V
… The voltage regulator circuit generating the power supply voltage (V_{XT}) for the low-speed clock generator circuit.

Note:

*1: The power supply voltage (V_{DD}) supplied from outside should be within the following recommended ranges:
OPTION A and OPTION B: V_{DD} = 1.3 to 1.7 V
OPTION C and OPTION D: V_{DD} = 1.8 to 3.6 V

*2: The halver circuit can be used in OPTION C and OPTION D (but cannot be used in OPTION A and OPTION B).

15.1.1 Precautions in Using the Halver Circuit

The halver circuit can be used in OPTION C and OPTION D. The halver circuit can operate when V_{DD} is 2.4 to 3.6 V. Depending on the battery load of the application system being used, it is possible to select by software to supply either the output voltage of the halver circuit or the battery voltage to the different voltage regulator circuits. It is possible to suppress the supply current during operation by efficiently using the halver circuit output voltage.

The software setting should be made so that the output voltage of the halver circuit is supplied to the different voltage regulator circuits during normal load, and the battery voltage is supplied to the different voltage regulator circuits during heavy load (such as when the backlight is lit).

Normal load is when V_{DD} is 2.4 to 3.6 V, and heavy load is when V_{DD} is less than 2.4 V.

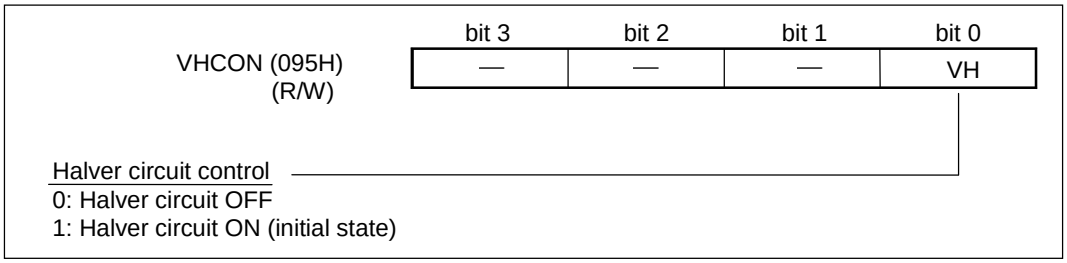
Further, it is possible to select by software a judgment threshold voltage of 2.6 V, 2.4 V, 2.0 V, or 1.8 V for the BDL (battery low detection) function.

15.2 Power Supply Circuit Related Register

The halver control register (VHCON) is used to control the power supply circuit.

- Halver control register (VHCON)
  VHCON is a 4-bit special function register (SFR) that specifies whether the output voltage of the halver circuit is used or whether the battery voltage is used as the voltage supplied to the internal different voltage regulator circuits.

  This register is available in the OPTION C and OPTION D (but not in the OPTION A and OPTION B, where the halver circuit cannot be used).



bit 0: VH
At system reset, VH is set to "1". At this time, the halver circuit enters an operating state and the output voltage (half the battery voltage) of the halver circuit will be supplied to the internal different voltage regulator circuits.

When VH is set to "0", the halver circuit stops operating and the battery voltage will be supplied to the internal different voltage regulator circuits.

15.3 Power Supply Circuit Configuration

15.3.1 OPTION A Power Supply Circuit Configuration

The power supply section of the OPTION A consists of the voltage regulator circuit (V/R1) for internal logic power supply and the voltage regulator circuit (V/R2) for the low-speed clock generator circuit.

The OPTION A does not have the voltage regulator circuit (V/R3) for the LCD bias generation reference voltage. Since the power supply voltage is used as the LCD bias generation reference voltage, V_{DD1} has to be shorted to V_{DD}.

The OPTION A power supply circuit configuration is shown in Figure 15-1.

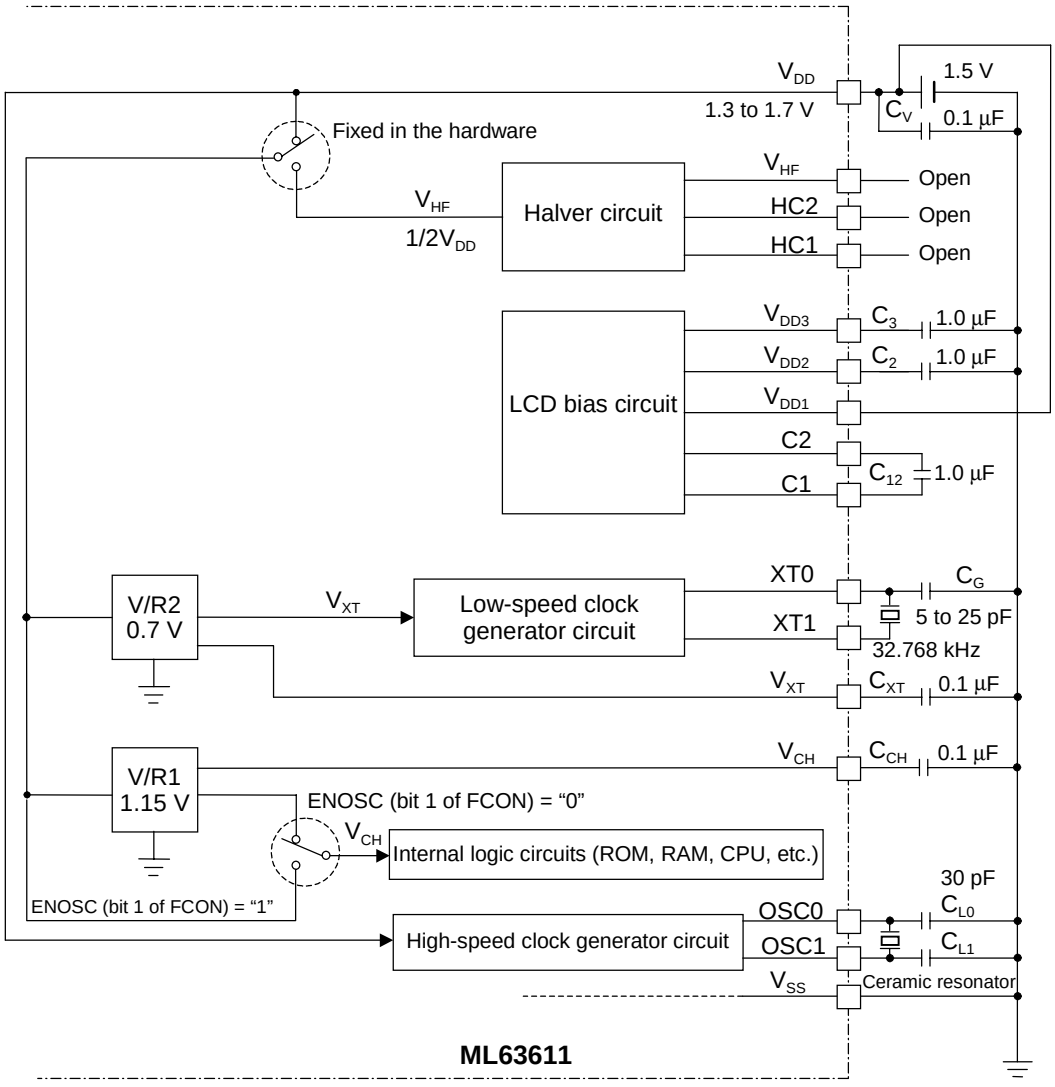


Figure 15-1 OPTION A Power Supply Circuit Configuration

15.3.2 Operation of the OPTION A Power Supply Circuit

The battery voltage (V_{DD}) is supplied to the different voltage regulator circuits (V/R1 and V/R2) of the OPTION A. The output voltages of the voltage regulator circuits are used as the internal logic power supply (V_{CH}) and the power supply voltage (V_{XT}) for the low-speed clock generator circuit. Further, the battery voltage is connected directly to the power supply of the high-speed clock generator circuit.

The output voltage of the voltage regulator V/R1 is supplied as the power supply (V_{CH}) for the internal logic circuits during low-speed clock operation.

The output voltage of the voltage regulator circuits V/R2 and V/R3 are always supplied respectively as the power supply voltage (V_{XT}) for the low-speed clock generator circuit and the reference voltage (V_{DD1}) for the LCD bias generator circuit.

15.3.3 OPTION B Power Supply Circuit Configuration

The power supply section of the OPTION B consists of the voltage regulator circuit (V/R1) for internal logic power supply, the voltage regulator circuit (V/R2) for the power supply of the low-speed clock generator circuit, and the voltage regulator circuit (V/R3) for the LCD bias generation reference voltage.

The OPTION B power supply circuit configuration is shown in Figure 15-2.

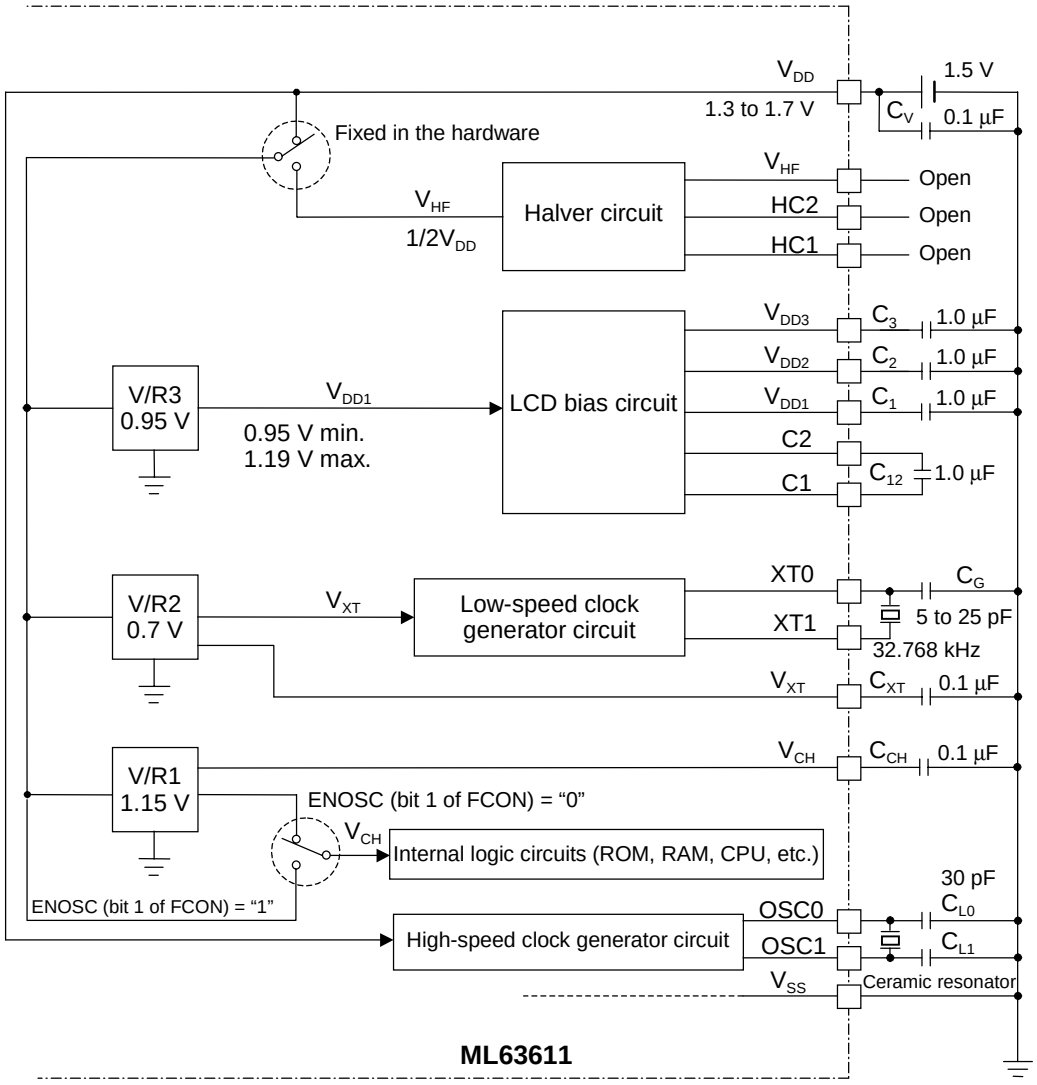


Figure 15-2 OPTION B Power Supply Circuit Configuration

15.3.4 Operation of the OPTION B Power Supply Circuit

The battery voltage (V_{DD}) is supplied to the different voltage regulator circuits (V/R1, V/R2, and V/R3) of the OPTION B. The output voltages of the voltage regulator circuits are used as the internal logic power supply (V_{CH}), the power supply voltage (V_{XT}) for the low-speed generator circuit, and the reference voltage (V_{DD1}) for the LCD bias generator circuit. Further, the battery voltage is connected directly to the power supply of the high-speed clock generator circuit.

The output voltage of the voltage regulator V/R1 is supplied as the power supply (V_{CH}) for the internal logic circuits during low-speed clock operation.

The output voltage of the voltage regulator circuits V/R2 and V/R3 are always supplied respectively as the power supply voltage (V_{XT}) for the low-speed clock generator circuit and the reference voltage (V_{DD1}) for the LCD bias generator circuit.

15.3.5 OPTION C Power Supply Circuit Configuration

The power supply section of the OPTION C consists of the halver circuit, the voltage regulator circuit (V/R1) for internal logic power supply and the voltage regulator circuit (V/R2) for the power supply of the low-speed clock generator circuit.

The OPTION C does not have the voltage regulator circuit (V/R3) for the LCD bias generation reference voltage. Since the power supply voltage is used as the LCD bias generation reference voltage, the pin V_{DD2} has to be shorted to V_{DD}.

The OPTION C power supply circuit configuration is shown in Figure 15-3.

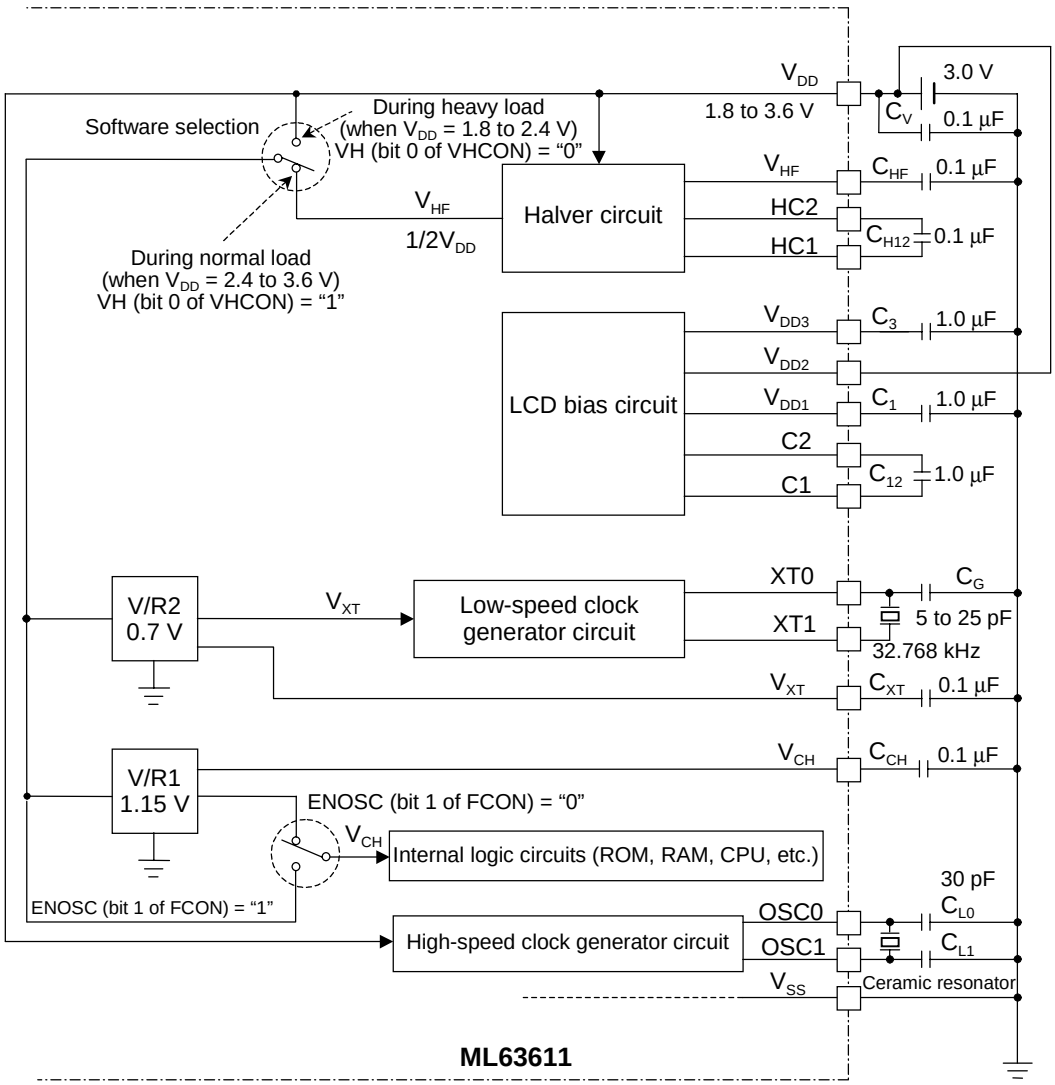


Figure 15-3 OPTION C Power Supply Circuit Configuration

Note:

When operating with the high-speed clock, first set VH (bit 0 of VHCON) to "0" and then set ENOSC (bit 1 of FCON) to "1".

If VH is the "1" state, the halver circuit output voltage (V_{HF}) becomes the power supply (V_{CH}) for the internal logic circuits and the operation will not be made properly.

15.3.6 Operation of the OPTION C Power Supply Circuit

The output voltage (1/2 V_{DD}) of the halver circuit is supplied to the different voltage regulator circuits (V/R1 and V/R2) of the OPTION C. The output voltages of the voltage regulator circuits are used as the internal logic power supply (V_{CH}) and the power supply voltage (V_{XT}) for the low-speed generator circuit. Further, the battery voltage is connected directly to the power supply of the high-speed clock generator circuit.

The software setting should be made so that the output voltage of the halver circuit is supplied to the different voltage regulator circuits during normal load, and the battery voltage is supplied directly to the different voltage regulator circuits during heavy load.

During normal load (V_{DD} = 2.4 to 3.6 V), use the halver circuit by setting VH (bit 0 of VHCON) to "1", and during heavy load (V_{DD} = less than 2.4 V), disable the halver circuit by setting VH to "0". When the halver circuit is not being used, the battery voltage will be supplied directly to the different voltage regulator circuits. By setting the judgment threshold voltage to 2.4 V using the BLD function, it is possible to detect the battery voltage during heavy load.

The output voltage of the voltage regulator V/R1 is supplied as the power supply (V_{CH}) for the internal logic circuits during low-speed clock operation. Further, during high-speed clock operation, it is possible to directly connect the battery voltage in the hardware to the power supply by setting ENOSC (bit 1 of FCON) to "1".

The output voltage of the voltage regulator circuit V/R2 is always supplied as the power supply voltage (V_{XT}) for the low-speed clock generator circuit.

15.3.7 OPTION D Power Supply Circuit Configuration

The power supply section of the OPTION D consists of the halver circuit, the voltage regulator circuit (V/R1) for internal logic power supply, the voltage regulator circuit (V/R2) for the power supply of the low-speed clock generator circuit, and the voltage regulator circuit (V/R3) for the LCD bias generation reference voltage.

The OPTION D power supply circuit configuration is shown in Figure 15-4.

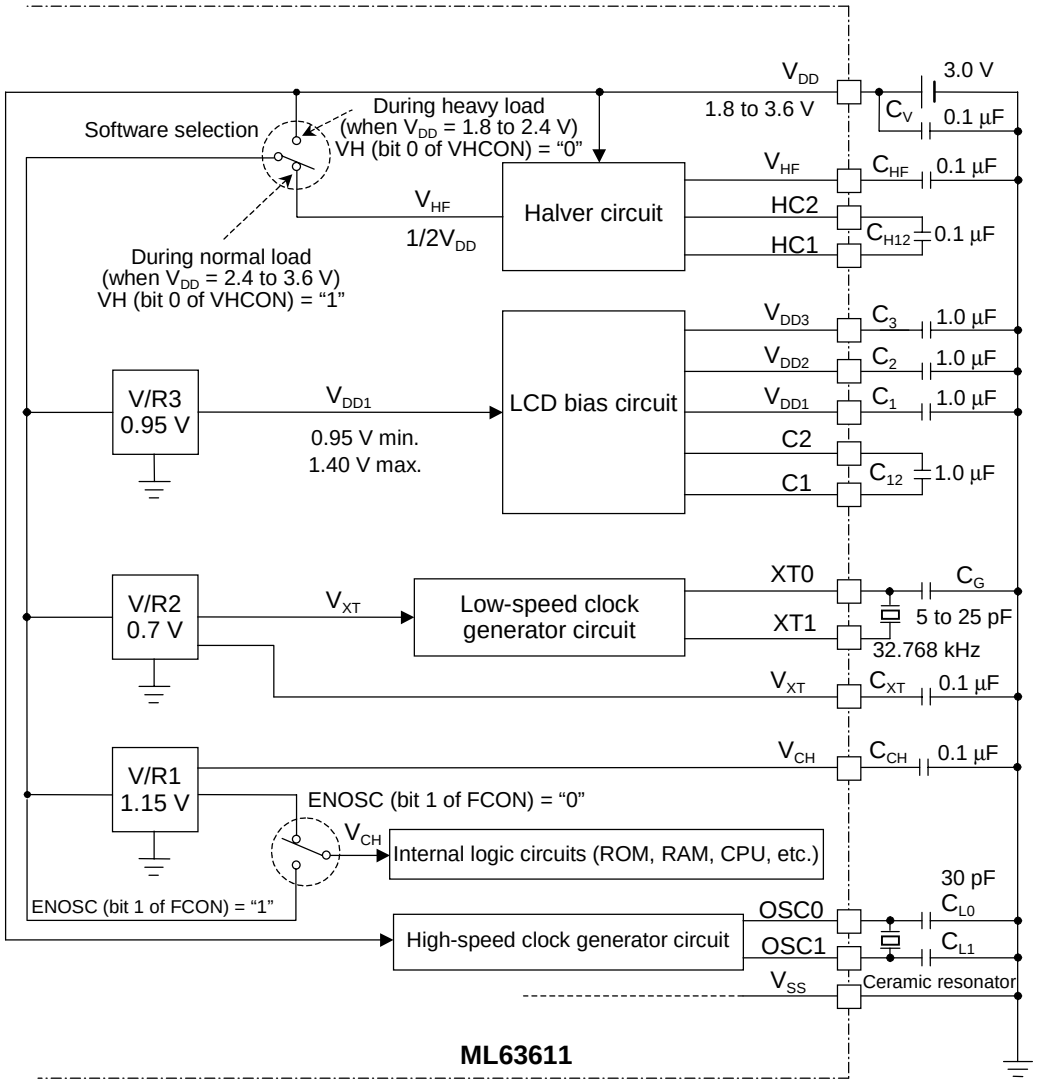


Figure 15-4 OPTION D Power Supply Circuit Configuration

Note:

When operating with the high-speed clock, first set VH (bit 0 of VHCON) to "0" and then set ENOSC (bit 1 of FCON) to "1".

If VH is the "1" state, the halver circuit output voltage (V_{HF}) becomes the power supply (V_{CH}) for the internal logic circuits and the operation will not be made properly.

15.3.8 Operation of the OPTION D Power Supply Circuit

The output voltage (1/2 V_{DD}) of the halver circuit is supplied to the different voltage regulator circuits (V/R1, V/R2, and V/R3) of the OPTION D. The output voltages of the voltage regulator circuits are used as the internal logic power supply (V_{CH}), the power supply voltage (V_{XT}) for the low-speed clock generator circuit, and the reference voltage (V_{DD1}) for LCD bias generator circuit. Further, the battery voltage is connected directly to the power supply of the high-speed clock generator circuit.

The software setting should be made so that the output voltage of the halver circuit is supplied to the different voltage regulator circuits during normal load, and the battery voltage is directly supplied to the different voltage regulator circuits during heavy load.

During normal load (V_{DD} = 2.4 to 3.6 V), use the halver circuit by setting VH (bit 0 of VHCON) to "1", and during heavy load (V_{DD} = less than 2.4 V), disable the halver circuit by setting VH to "0". When the halver circuit is not being used, the battery voltage will be supplied directly to the different voltage regulator circuits.
By setting the judgment threshold voltage of the BLD function to 2.4 V, it is possible to detect the battery voltage during heavy load.

The output voltage of the voltage regulator V/R1 is supplied as the power supply (VH) for the internal logic circuits during low-speed clock operation. Further, during high-speed clock operation, it is possible to directly connect the battery voltage in the hardware to the power supply by first setting VH to "0" and then setting ENOSC (bit 1 of FCON) to "1".

The output voltages of the voltage regulator circuits V/R2 and V/R3 are always supplied respectively as the power supply voltage (V_{XT}) for the low-speed clock generator circuit and the reference voltage (V_{DD1}) for the LCD bias generator circuit.

Chapter 16

A/D Converter (ADC)

16. A/D Converter (ADC)

16.1 Overview

The ML63611 has a built-in 2-channel RC oscillation method A/D converter. The A/D converter is composed of a 2-channel oscillation circuit, Counter A (CNTA0 to 4) which is a 4.8-digit decade counter, Counter B (CNTB0 to 3) which is a 14-bit binary counter and A/D converter control registers 0 and 1 (ADCON0, ADCON1).

By counting oscillation frequencies due to resistance or capacitance connected to the RC oscillation circuit, the A/D converter converts resistance values or capacitance values to corresponding digital values. By using a thermistor or a humidity sensor as a resistance, a thermometer or a hygrometer can be constructed. By applying sensors to the 2-channel RC oscillation circuit, it is also possible to extend measurement ranges or measurement at two places.

16.2 A/D Converter Configuration

Figure 16-1 shows the A/D converter configuration.

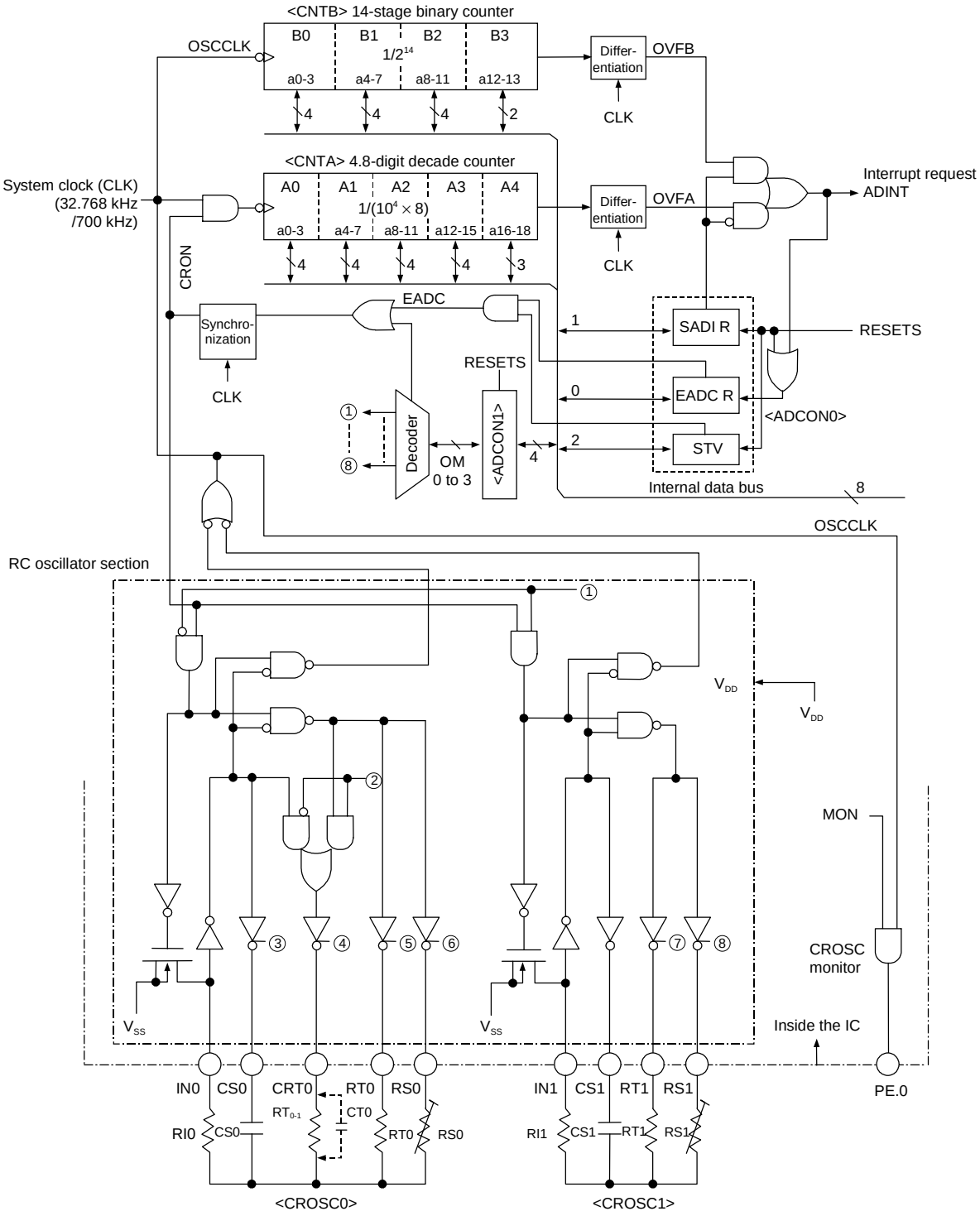


Figure 16-1 Layout of A/D Converter

16.3 Operation of A/D Converter

As shown in Figure 16-1, the RC oscillation circuit can be made by connecting resistors and capacitors to each pin.

Counter A (CNTA0 to 4) is a 4.8-digit decade counter ($1/(10^4 \times 8)$) to count the system clock (CLK) which is the time reference and can count up to a maximum of 79,999.

Counter B (CNTB0 to 3) is a 14-stage binary counter to count the oscillation clock (OSCCLK) of the RC oscillation circuit and can count up to a maximum of 16,383.

Both Counter A and Counter B have the overflow flags (OVFA and OVFB) and overflow output generates the A/D converter interrupt request (ADINT). ADINT due to overflow of either Counter A or Counter B is selected by Bit 1 (SADI) of the A/D converter control register 0 (ADCON0). By resetting SADI to "0", overflow of Counter A is selected and by setting SADI to "1", overflow of Counter B is selected. The vector address of ADINT is at address 02FH.

Bit 0 (EADC) of ADCON0 is a bit to select operation/halt of the A/D conversion. By resetting EADC to "0", the RC oscillation is halted and no counting is performed. By setting EADC to "1", the RC oscillation is begun and counting of the RC oscillation clock and the system clock is started.

Bit 2 (STV) is used for setting the internal power supply voltage (V_{CH}) as the power supply voltage in order to enable the operation of Counter A and Counter B. However, carry out the setting of this bit only after setting the halver circuit to "disabled".

Set EADC to "1" after waiting for about 120 μs (2 clock periods of the low-speed clock) after setting STV to "1". Note that Counter A is likely to operate improperly if the RC oscillator circuit is made to operate when the internal power supply has not become stable.

Various oscillation mode of the RC oscillation part is performed by the A/D converter control register 1 (ADCON1). The RC oscillation clock can be monitored by outputting to PE.0 in secondary function.

16.3.1 RC Oscillation Circuit

The A/D converter of the RC oscillation method performs A/D conversion by digitizing the oscillation frequency ratio of a reference resistance (or capacitance) to a resistance sensor, such as thermistor sensor (or capacitance sensor).

By taking the ratio of oscillation frequencies of the reference to the sensor, it is possible to A/D convert the characteristics of a sensor itself by canceling error factors intrinsic to the RC oscillation circuit. Consequently, it is necessary to oscillate the reference side and the sensor side with the same oscillation circuit and a pair of the reference side and the sensor side is usually used.

Table 16-1 shows the oscillation mode by Bits 3 to 0 (OM3 to OM0) of the A/D converter control register (ADCON1). Figures 16-2 to 16-5 show the layout and values of the OM3 to OM0 bits.

Table 16-1 Oscillation Mode by OM3 to OM0 Bits

Mode No.	ADCON1				CROSC0 output pins				CROSC1 output pins		Mode	
	OM3	OM2	OM1	OM0	RS0	RT0	CRT0	CS0	RS1	RT1		
0	0	0	0	0	Z	Z	Z	Z	Z	Z	IN0 external clock input mode	
1	0	0	0	1	1/0	Z	Z	0/1	Z	Z	RS0-CS0 oscillation	CROSC0 oscillation mode
2	0	0	1	0	Z	1/0	Z	0/1	Z	Z	RT0-CS0 oscillation	
3	0	0	1	1	Z	Z	1/0	0/1	Z	Z	$RT_{0\text{-}1}$-CS0 oscillation	
4	0	1	0	0	1/0	Z	0/1	Z	Z	Z	RS0-CT0 oscillation	
5	0	1	0	1	Z	Z	Z	Z	1/0	Z	RS1-CS1 oscillation	CROSC1 oscillation mode
6	0	1	1	0	Z	Z	Z	Z	Z	1/0	RT1-CS1 oscillation	
7	0	1	1	1	Z	Z	Z	Z	Z	Z	IN1 external clock input mode	
8	1	—	—	—	Z	Z	Z	Z	Z	Z	(unavailable)	

Note: Z: High-impedance output
1/0, 0/1: Active output
—: Arbitrary

Modes No.0 and No.7 in Table 16-1 measure the external clock which is input to the IN0 pin or the IN1 pin by halting the operation of the RC oscillation circuit.

As shown in Table 16-1, no two oscillation circuits can operate simultaneously. This prevents interference to the oscillation operation when two are operated simultaneously.

The relation between oscillation frequency (f_{OSCCLK}), capacitance value (C), and resistance value (R) is expressed by the following:

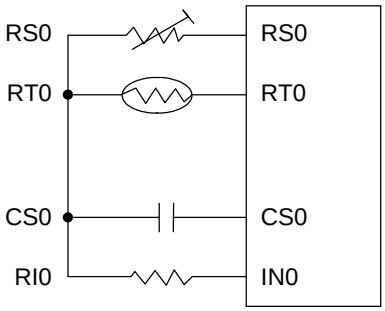
$$\frac{1}{f_{OSCCLK}} = t_{OSCCLK} = k_{OSCCLK} \bullet C \bullet R$$

where t_{OSCCLK} is the period of the oscillation frequency, k_{OSCCLK} is a proportionality constant and C•R is product of C_S or C_T and R_S or R_T. The value of k_{OSCCLK} varies slightly depending on V_{DD} (power supply voltage), RI, C and R and its standard values are listed in Table 16-2.

Table 16-2 Standard Value of k_{OSCCLK} of RC Oscillation Circuit

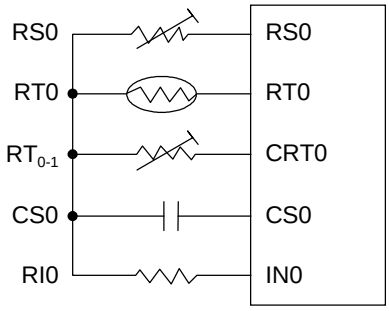
V_{DD} (V)	Rln (kΩ)	CSn, CTn (pF)	RSn, RTn (kΩ)	k_{OSCCLK} (Typ.)
3	10	820	100	2.1
		820	10	2.4
1.5	10	820	100	2.1
		820	10	2.3

Note: n = 0, 1, 0-1



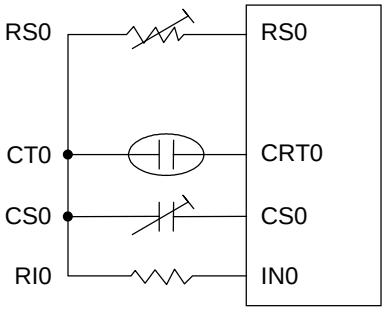
OM3	OM2	OM1	OM0	Oscillation mode
0	0	0	1	Oscillation with reference resistance RS0 and CS0
0	0	1	0	Oscillation with sensor RT0 and CS0

Figure 16-2 Measurement of CROSC0 by a Resistance Sensor



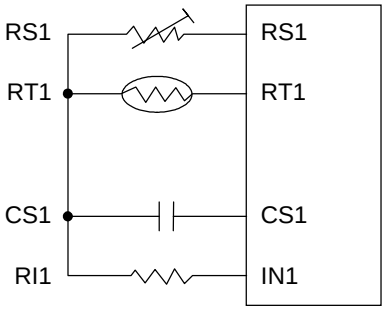
OM3	OM2	OM1	OM0	Oscillation mode
0	0	0	1	Oscillation with reference resistance RS0 and CS0
0	0	1	0	Oscillation with sensor RT0 and CS0
0	0	1	1	Oscillation with reference resistance RT_{0-1} and CS0

Figure 16-3 Measurement of CROSC0 by a Resistance Sensor (when two-point adjustment with two reference resistances)



OM3	OM2	OM1	OM0	Oscillation mode
0	0	0	1	Oscillation with RS0 and reference capacitance CS0
0	1	0	0	Oscillation with RS0 and sensor CT0

Figure 16-4 Measurement of CROSC0 by a Capacitance Sensor



OM3	OM2	OM1	OM0	Oscillation mode
0	1	0	1	Oscillation with reference resistance RS1 and CS1
0	1	1	0	Oscillation with sensor RT1 and CS1

Figure 16-5 Measurement of CROSC1 by a Resistance Sensor

Note:

Unused pins should be left open.

16.3.2 Counter A/B Reference Mode

The conversion operation of the A/D converter is performed by one of the following two modes.

- Counter A Reference Mode (SADI bit of ADCON0 = 0)
  This is the mode to set gate time by the system clock (CLK) and Counter A, to count the RC oscillation clock (OSCCLK) by Counter B with the gate time and to output contents of Counter B as a digital value.
  The digital value is proportional to the RC oscillation frequency.

- Counter B Reference Mode (SADI bit of ADCON0 = 1)
  This is the mode to set gate time by the RC oscillation clock (OSCCLK) and Counter B, to count the system clock (CLK) by Counter A with the gate time and to output contents of Counter A as a digital value.
  The digital value is inversely proportional to the RC oscillation frequency.

(1) Operation of Counter A Reference Mode
Figure 16-6 shows the operating timing of Counter A reference mode.

Counter A reference mode is performed by the following procedure: (refer to Figure 16-6)

① Subtract "nA0" (the count value) from the maximum value +1 (80,000) and set that value to Counter A (CNTA4 to 0). Here, the product of the count value, "nA0", and the period of CLK indicates the gate time.
Counter A ← (80,000 – nA0)

② Clear Counter B (CNTB3 to 0) to 0000H.
Counter B ← 0000H

③ Set the bits OM3 to OM0 of ADCON1 to a necessary oscillation mode (refer to Table 16-1).

④ Set the internal power supply as the power supply voltage by first setting the halver circuit to "turned OFF" (set bit 0 of VHCON to "0") and then writing "4H" to ADCON0 (STV = 1). In the OPTION A and OPTION B, the setting of VHCON is not required.

⑤ Write ADCON0 to "5H" (STV =1, SADI = 0, EADC = 1) after waiting for about 120 μs.

Note:

The order of ① to ③ is arbitrary.

After setting STV (bit 2 of ADCON0) to "1", wait for about 120 μs and then set EADC to "1". In the OPTION A and OPTION B, the setting of VHCON is not required.

By ⑤, A/D conversion starts.

Counter A starts counting the system clock (CLK) when EADC is set to "1" and the CRON signal that synchronizes with the falling of the system clock is set to "1". When Counter A overflows, the EADC bit is automatically reset (⑥) and the counting is finished. At the same time, the A/D converter interrupt request signal (ADINT) becomes "1" to generate the A/D converter interrupt request (⑦).

When the CRON signal is set to "1", the RC oscillation is started and Counter B starts counting the RC oscillation clock (OSCCLK). When Counter A overflows and the EADC bit is automatically reset, the counting of counter B is finished.

The last count value of "nB0" at Counter B is the count value of OSCCLK during the gate time "$nA0 \cdot t_{SYSCLK}$" and is expressed by

$$nB0 \cong nA0 \cdot \frac{t_{SYSCLK}}{t_{OSCCLK}} \propto f_{OSCCLK}$$

where t_{SYSCLK} is the period of CLK and t_{OSCCLK} is the period of OSCCLK.

In other words, "nB0" is directly proportional to the RC oscillation frequency (f_{OSCCLK}).

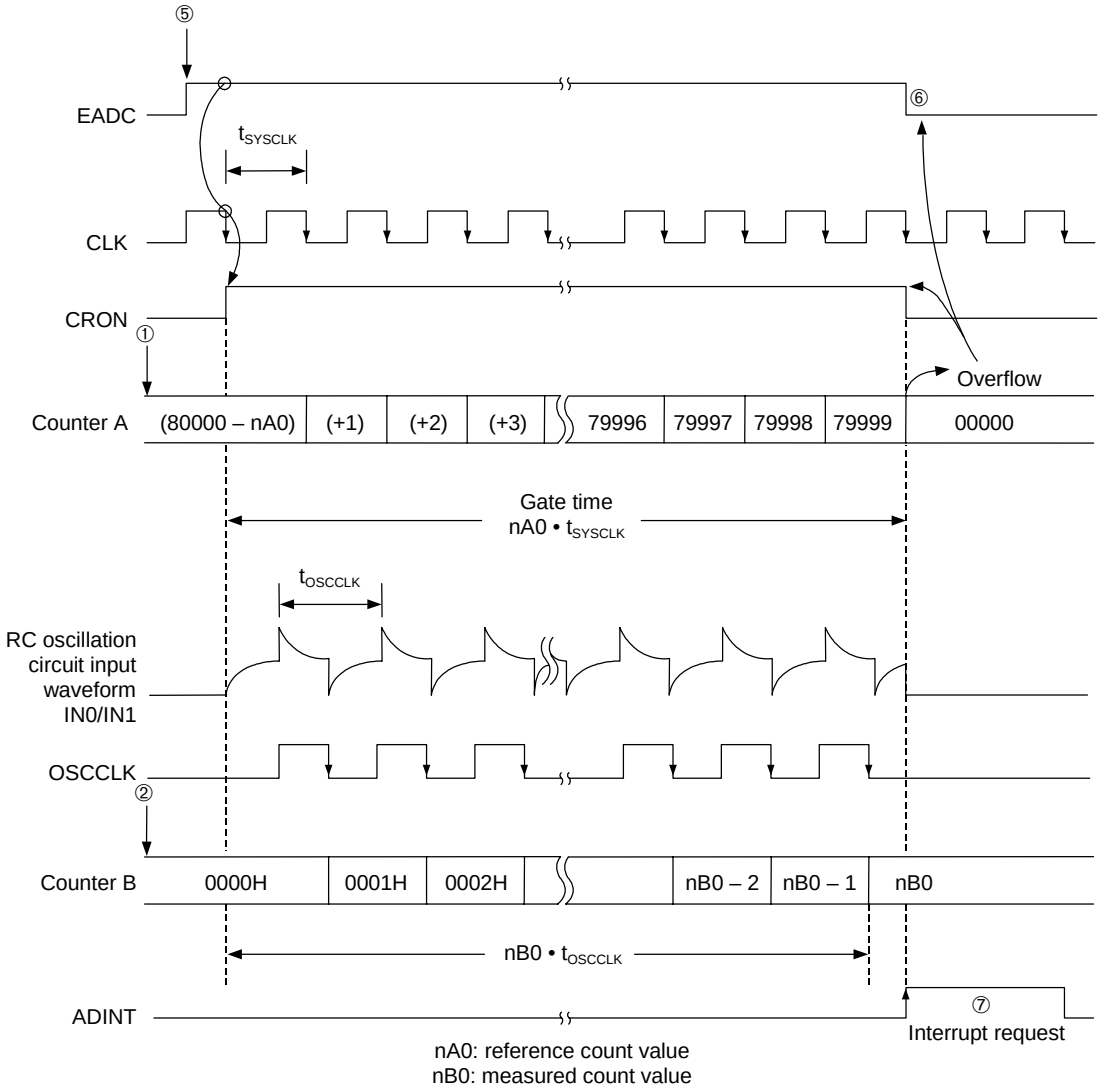


Figure 16-6 Operating Timing of Counter A Reference Mode

(2) Operation of Counter B Reference Mode
Figure 16-7 shows the operating timing of Counter B reference mode.

Counter B reference mode is performed by the following procedure: (refer to Figure 16-7)

① Subtract "nB1" (the count value) from the maximum value +1 (4000H) and set the result to Counter B (CNTB3 to 0). Here, the product of the count value, "nB1", and the period of OSCCLK denotes the gate time.
Counter B ← (4000 – nB1)

② Clear Counter A (CNTA4 to 0) to 0000H.
Counter A ← 0000H

③ Set the bits OM3 to 0 of ADCON1 to a necessary oscillation mode (refer to Table 16-1).

④ Set the internal power supply as the power supply voltage by first setting the halver circuit to "turned OFF" (set bit 0 of VHCON to "0") and then writing "4H" to ADCON0 (STV = 1). In the OPTION A and OPTION B, the setting of VHCON is not required.

⑤ Write ADCON0 to "5H" (STV =1, SADI = 0, EADC = 1) after waiting for about 120 µs.

Note:

The order of ① to ③ is arbitrary.

After setting STV (bit 2 of ADCON0) to "1", wait for about 120 µs and then set EADC to "1". In the OPTION A and OPTION B, the setting of VHCON is not required.

By ⑤, A/D conversion starts.

Counter B starts counting the RC oscillation clock (OSCCLK) when the EADC bit is set to "1" and the CRON signal (signal that synchronizes with the falling of the system clock) is set to "1". When Counter B overflows, the EADC bit is automatically reset (⑥) and the conversion is finished. At the same time, the A/D converter interrupt request signal (ADINT) becomes "1" to generate the A/D converter interrupt request (⑦).

When the CRON signal is set to "1", Counter A starts counting the system clock (CLK). When Counter B overflows and the EADC bit is automatically reset, the counting of counter A is finished.

The last count value of "nA1" at Counter A is the count value of SYSCLK during the gate time "$nB1 \bullet t_{OSCCLK}$" and is expressed by

$$nA1 \cong nB1 \bullet \frac{t_{OSCCLK}}{t_{SYSCLK}} \propto \frac{1}{f_{OSCCLK}}$$

In other words, "nA1" is inversely proportional to the RC oscillation frequency (f_{OSC}).

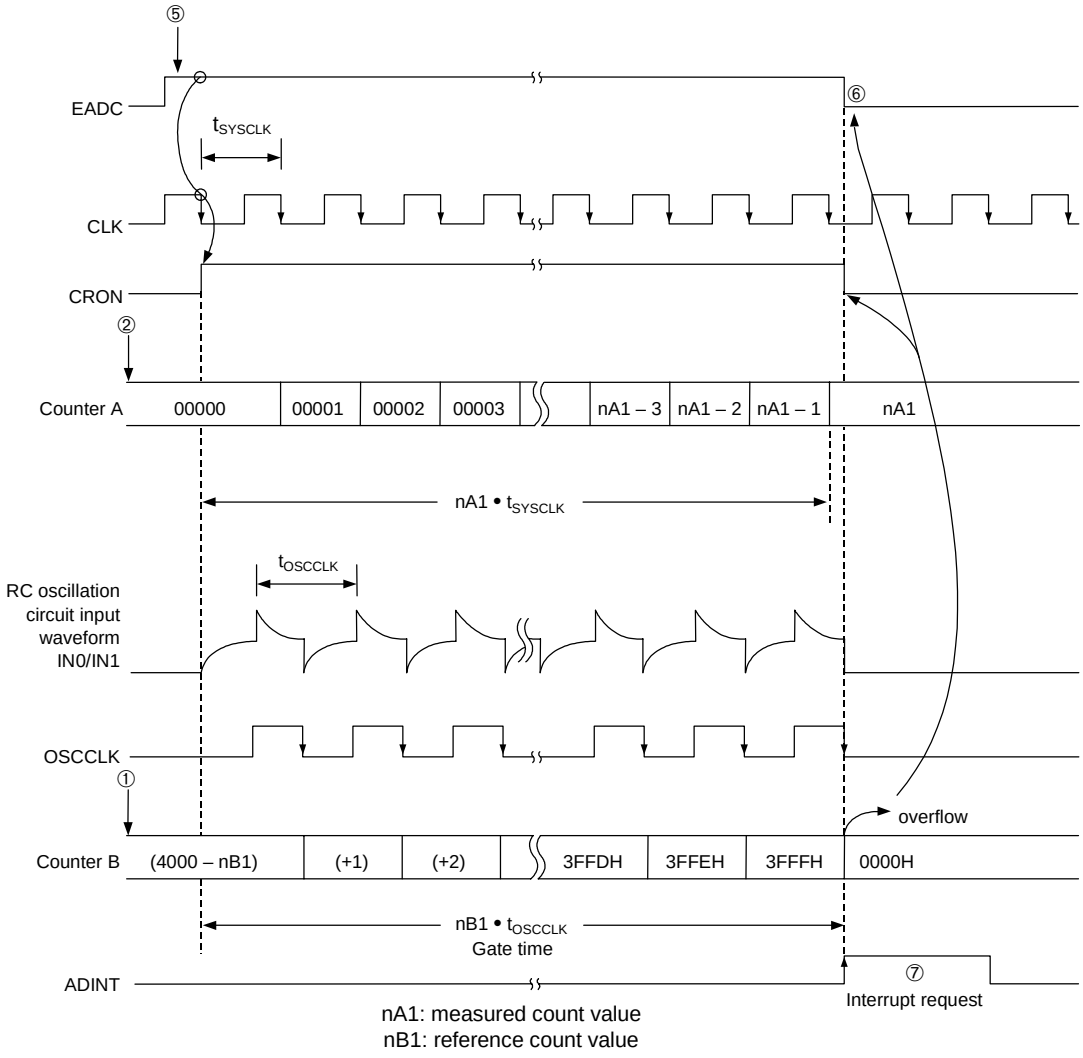


Figure 16-7 Operating Timing of Counter B Reference Mode

16.3.3 Example of Usage of A/D Converter

The method to perform A/D conversion of sensor values by using Counter A reference mode and Counter B reference mode is explained by taking temperature measurement with a thermistor as an example.

Figure 16-8 shows the layout of RC oscillation circuit.

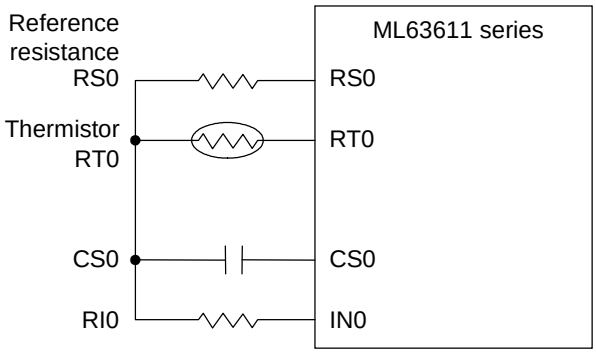


Figure 16-8 Layout of RC Oscillation Circuit of a Thermistor Using CROSC0

Figure 16-9 shows the temperature characteristics of the resistance value, RT0, of the thermistor.

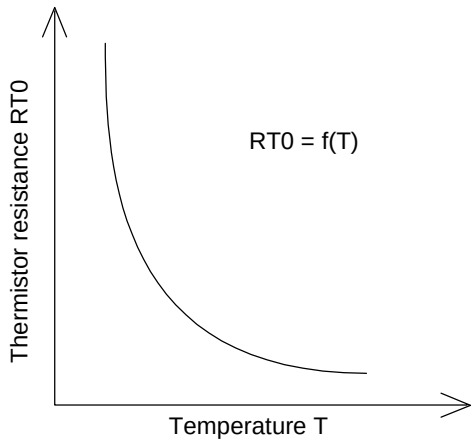


Figure 16-9 Temperature Characteristics of Thermistor

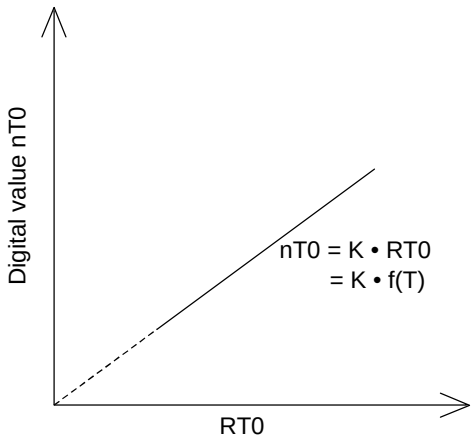


Figure 16-10 A/D Conversion Characteristics (Ideal characteristics when nT0 is proportional to RT0)

RT0 is expressed as a function of temperature T as

$$RT0 = f(T)$$

Figure 16-10 shows the ideal characteristics of A/D conversion taking RT0 as an analog quantity and it is ideal that the A/D conversion value nT0 depends solely on RT0. Assuming that nT0 is proportional to RT0, the value of nT0 is expressed by temperature T and the proportionality constant K as

$$nT0 = K \bullet RT0 = K \bullet f(T) \quad \text{- - - - -} \quad \text{equation (a)}$$

Consequently, by performing conversion processing corresponding to the characteristics shown by Figure 16-9 to nT0, it is possible to express temperatures by digital values.

The conversion method from an analog value of RT0 to a digital value of nT0 is now explained.

To convert RT0 to a digital value, the ratio of oscillation frequencies of RT0 to RS0 (ideal if independent of temperature) is used. This is to cancel the error factors of the oscillation characteristics.

As shown in Figures 16-9 and 16-11, RT0 depends on temperature T and RS0 is always constant regardless of temperature T. The oscillation characteristics, f_{OSC}-T, using these resistances is ideal if the solid lines of Figures 16-12 and 16-13 can be realized. However, in reality, the dotted lines are obtained due to error factors of the temperature characteristics of the IC and others. Since the conditions of f_{OSC}(RT0) and f_{OSC}(RS0) are about the same except the resistance, their error should be similar with each other and consequently, if the ratio of one to the other is taken, the error should be canceled.

The ratio of f_{OSC}(RT0) to f_{OSC}(RS0) corresponds to the A/D conversion value of nT0 which, ideally, depends solely on RT0.

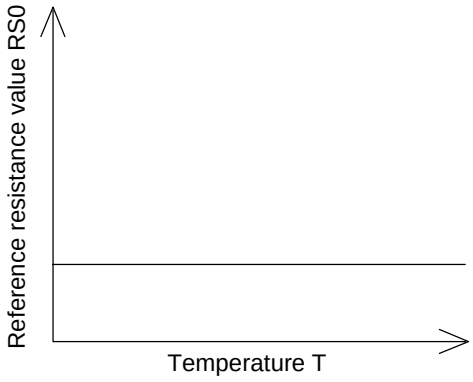


Figure 16-11 Temperature Characteristics of Thermistor

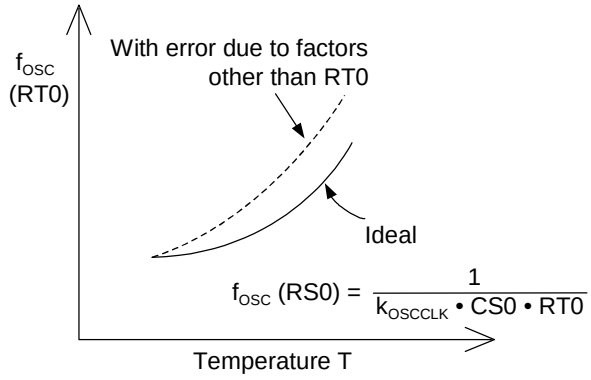


Figure 16-12 Oscillation Characteristics of Thermistor

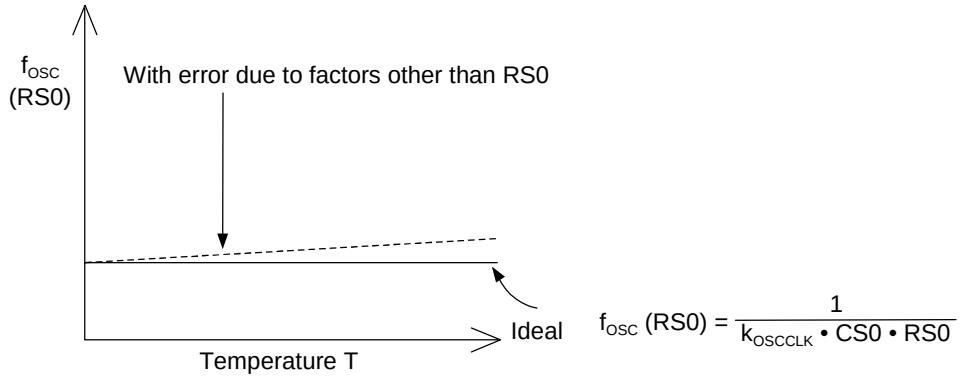


Figure 16-13 Oscillation Characteristics of Reference Resistance

Figure 16-14 shows the timing chart for one cycle of conversion to digital values from the RT0 values, i.e. one cycle of A/D conversion.

One cycle of A/D conversion needs to be composed of two steps shown in Figure 16-14 because the reference resistance and the thermistor must be oscillated independently when taking the ratio of them.

In this example, those two steps are taken by the following combination:

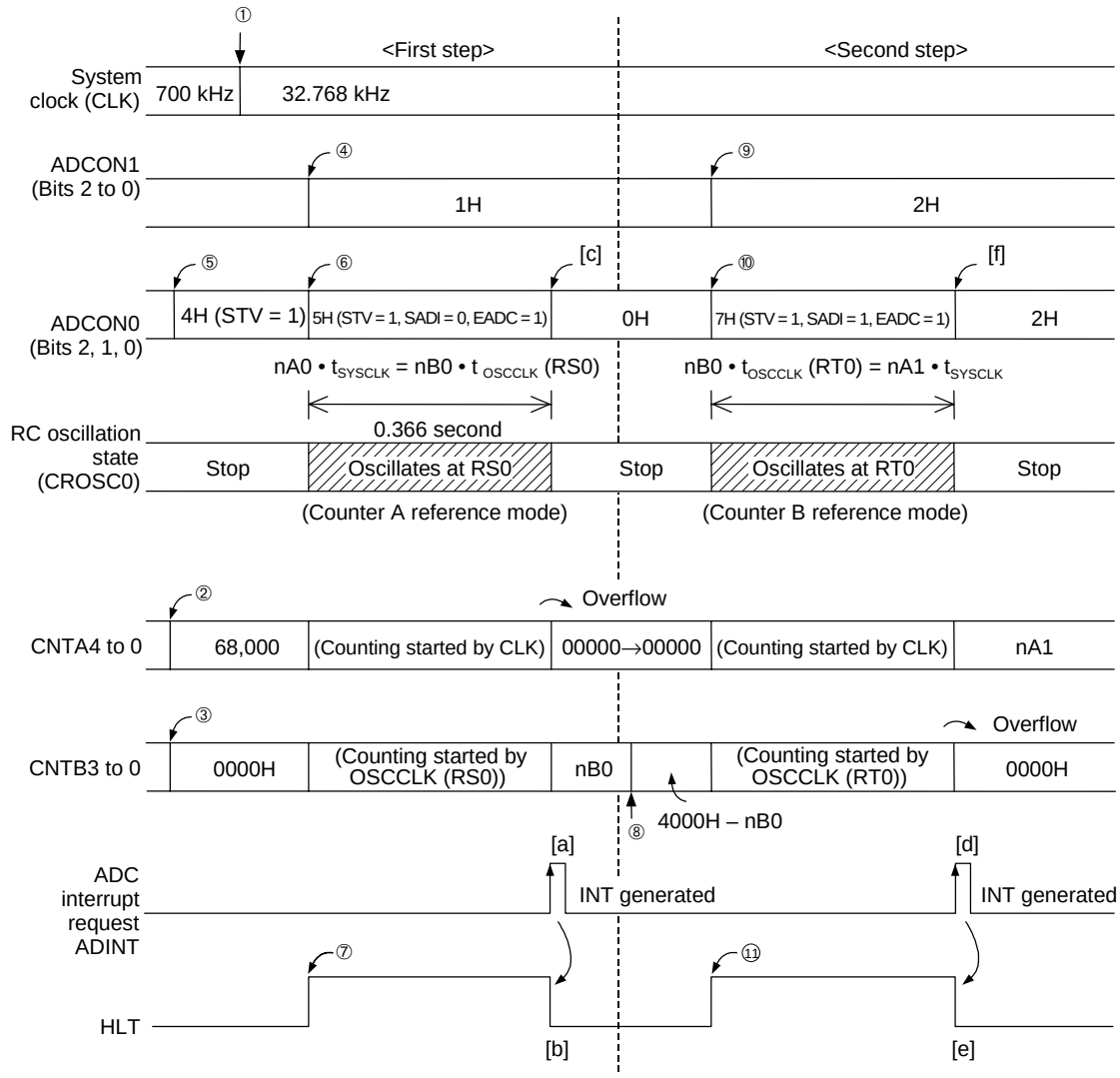
- First step = RC oscillation by RS0 with Counter A reference mode
- Second step = RC oscillation by RT0 with Counter B reference mode

Various other methods are possible besides the one above.

In the above method, the operating time by the second step varies by the value of thermistor RT0. However, if it is necessary to avoid such variation, the following combination is recommended:

- First step = RC oscillation by RS0 with Counter B reference mode
- Second step = RC oscillation by RT0 with Counter A reference mode

In the following, the procedure of A/D conversion will be explained taking Figure 16-14 as an example.



Notes: nA0 = 12,000, t_{SYSCLK} = 1/32768 Hz, ① to ⑪: processing by software, [a] to [f]: processing by hardware

Figure 16-14 Timing Chart of One Cycle of A/D Conversion

<First Step>

① Set the system clock to 32.768 kHz (write 0H to FCON), if using High-speed clock as system clock.

② Set "80,000 – nA0" to Counter A.

Note:

nA0 is taken as 12,000 in order to set the gate time nA0 • t_{SYSCLK} of oscillation mode of the reference resistance RS0 as 0.366 second. The value of nA0 depends on the size of quantum error of A/D conversion and the larger nA0, the smaller the error.

③ Clear Counter B to "0000H".

④ Write "1H" to ADCON1 and set it to oscillation mode with reference resistance RS0.

⑤ Write "4H" to ADCON0 to set the internal power supply voltage as the power supply voltage.

⑥ Wait for about 120 μs, then write "5H" to ADCON0 to start operation of Counter A reference mode.

Note:

The order of ① to ④ is arbitrary.

For step ⑤, set STV (bit 2 of ADCON0) to "1" after setting the halver circuit to "turned OFF" (set bit 0 of VHCON to "0"). Next, wait for about 120 μs and then set EADC to "1". In the OPTION A and OPTION B, the setting of VHCON is not required.

⑦ Execute the HALT instruction to enter the halt mode.

Note:

By selecting halt mode, noise to the RC oscillation circuit may be reduced. In regular usage, halt mode is recommended during RC oscillation operation.

The RC oscillation circuit (CROSC0) continues oscillation with reference resistance RS0 for about 0.366 second at this time and when Counter A overflows, the ADINT signal is set to "1" and the A/D converter interrupt request is generated ([a]). By the generation of the interrupt request, halt mode is released ([b]) and the A/D conversion operation is stopped ([c]; the EADC bit = 0). At this moment, Counter A is in 00000 state.

The contents of Counter B are expressed as

$$nB0 = nA0 \bullet \frac{t_{SYSCLK}}{t_{OSCCLK}\,(RS0)} \quad \text{- - - - -} \quad \text{equation (b)}$$

<Second Step>

⑧ Calculate "4000H – nB0" by the contents "nB0" of Counter B and set that value to Counter B.

Note:

Although clearing of Counter A is needed, additional processing is not necessary as it is already in "00000" state.

⑨ Write "2H" to ADCON1 and start oscillation mode with thermistor RT0.

⑩ Write "7H" to ADCON0 and start A/D conversion in Counter B reference mode.

Note:

Since the setting of STV = 1 has already been made in the preprocessing, there is no need to wait for 120 μs.

⑪ Execute the HALT instruction to enter the halt mode.

The RC oscillation circuit (CROSC0) oscillates with thermistor RT0 from this time until overflow of Counter B. This period is equivalent to product of "nB0" from first step and t_{OSCCLK} (RT0) due to RT0.

When Counter B overflows, the ADINT signal is set to "1" and the A/D converter interrupt request is generated ([d]). By the generation of the interrupt request, halt mode is released ([e]) and the A/D conversion operation is stopped ([f]; the EADC bit = "0"). The contents of Counter A becomes the A/D conversion value of nA1 and is expressed by the following:

$$nA1 = nB0 \bullet \frac{t_{OSCCLK}\,(RT0)}{t_{SYSCLK}} \quad \text{- - - - -} \quad \text{equation (c)}$$

By equations (b) and (c), nA1 is expressed as

$$nA1 = nA0 \bullet \frac{t_{OSCCLK}\,(RT0)}{t_{OSCCLK}\,(RS0)} \quad \text{- - - - -} \quad \text{equation (d)}$$

where t_{OSCCLK} (RS0) is the period of the oscillation clock by reference resistance RS0 and t_{OSCCLK} (RT0) is the period of the oscillation clock by thermistor RT0.

The oscillation period is expressed ideally as

$$t_{OSCCLK}\,(RS0) = k_{OSCCLK} \bullet CS0 \bullet RS0$$
$$t_{OSCCLK}\,(RT0) = k_{OSCCLK} \bullet CS0 \bullet RT0 \quad \text{- - - - -} \quad \text{equation (e)}$$

By substituting equation (e) to equation (d), nA1 is expressed as

$$nA1 = nA0 \bullet \frac{RT0}{RS0}$$

As "nA0" (12,000 in this example) and RS0 are fixed constants, "nA1" becomes a digital value proportional to RT0. This "nA1" is equivalent to "nT0" in equation (a).

The obtained "nA1" must be further converted to a temperature display value depending on the temperature-resistance characteristics of the thermistor in a program.

16.3.4 RC Oscillation Monitor

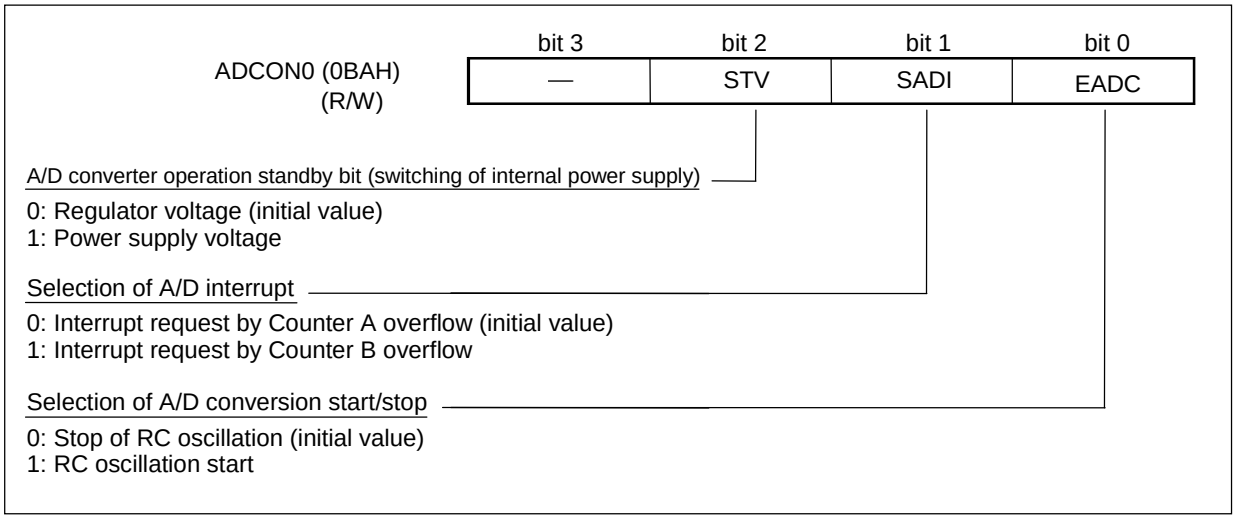
By setting Bit 0 (MON) of the Port E mode register (PEMOD) to "1", the RC oscillation clock (OSCCLK) can be output to PE.0.

The RC oscillation monitor is useful when checking the characteristics of the RC oscillation circuit. For instance, it is possible to measure the relationship between sensors such as a thermistor and an oscillation frequency. For example, by examining the relationship between ambient temperature of a thermistor built-in RC oscillation circuit and oscillation frequencies of the thermistor RT0 and the reference resistance RS0, it is possible to obtain the conversion coefficient from the value of nA1 to the temperature display values.

16.4 Registers Related to A/D Converter

(1) A/D converter control register 0 (ADCON0)

The A/D converter control register 0 (ADCON0) is a 4-bit special function register (SFR) that selects start/stop of RC oscillation of the A/D converter and the A/D converter interrupt by Counter A or Counter B.



bit 2: STV

This bit is used for setting the internal power supply as the power supply voltage in order to enable the operation of Counter A and Counter B. However, carry out the setting of this bit only after setting the halver circuit to "turned OFF".

Wait for about 120 µs before starting RC oscillation. Note that Counter A or Counter B may operate improperly if the RC oscillator circuit is made to operate when the internal power supply has not become stable. When this bit is "0", the RC oscillator circuit does not operate.

bit 1: SADI

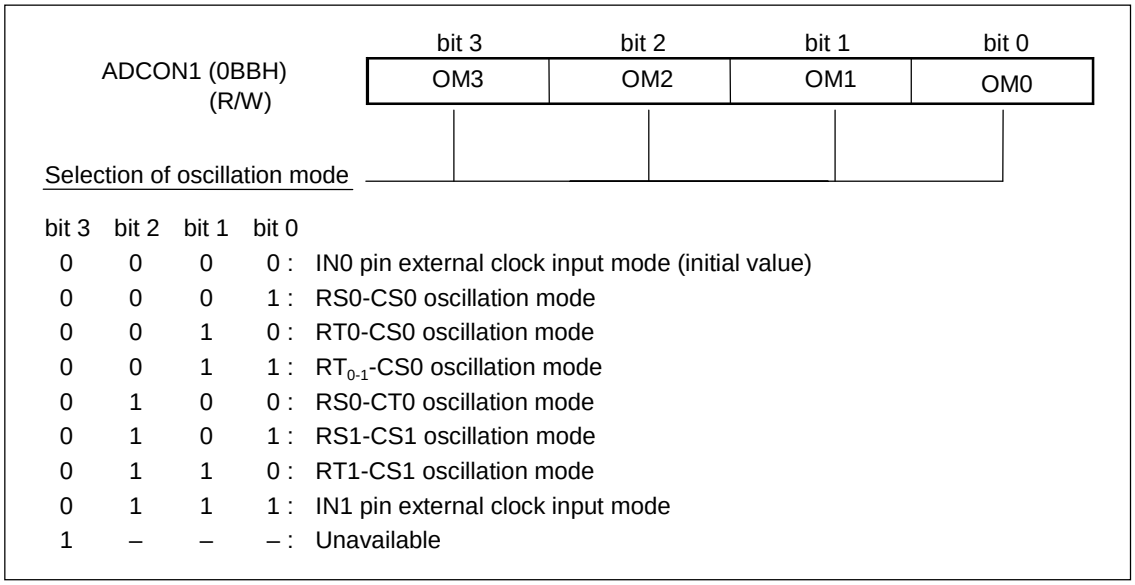
This bit selects the A/D converter interrupt request (ADINT) by overflow of either Counter A or Counter B. By resetting SADI to "0", the interrupt request by overflow of Counter A is selected and by setting SADI to "1", the interrupt request by overflow of Counter B is selected. At system reset, SADI is reset to "0".

bit 0: EADC

This bit selects start/stop of conversion of the A/D converter. When set to "1", A/D conversion is started and when reset to "0", A/D conversion is stopped. When either Counter A or Counter B overflows while EADC is set to "1" to start counting, the EADC bit is set to "0" automaticaly. Consequently, EADC indicates that the measurement is in progress. At system reset, the EADC bit is reset to "0" and the system is in stop state.

When the STV bit is "0", A/D conversion is not started.

(2) A/D converter control register 1 (ADCON1)
The A/D converter control register 1 (ADCON1) is a 4-bit special function register (SFR) to select oscillation mode of the RC oscillation circuit.

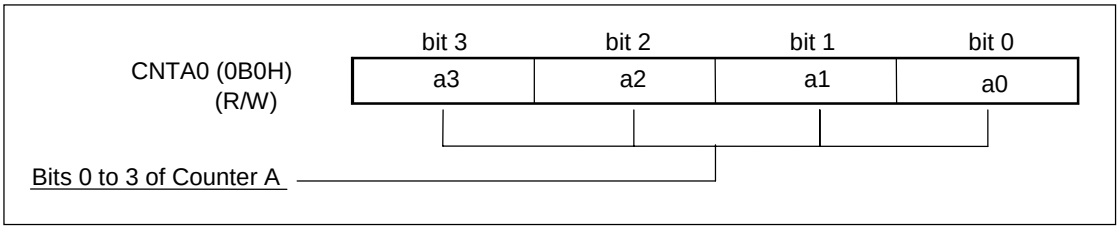


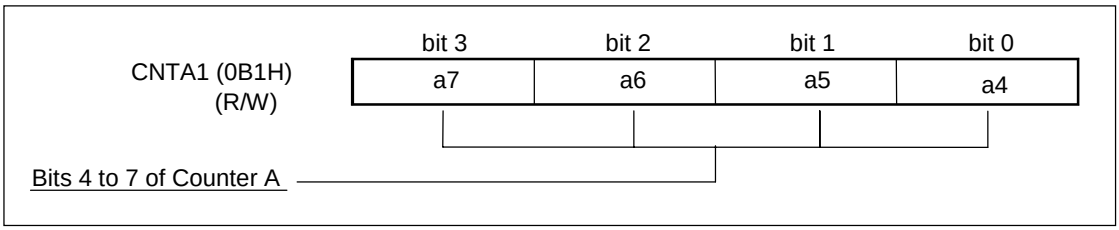
bit 3	bit 2	bit 1	bit 0	
0	0	0	0 :	IN0 pin external clock input mode (initial value)
0	0	0	1 :	RS0-CS0 oscillation mode
0	0	1	0 :	RT0-CS0 oscillation mode
0	0	1	1 :	RT_{0-1}-CS0 oscillation mode
0	1	0	0 :	RS0-CT0 oscillation mode
0	1	0	1 :	RS1-CS1 oscillation mode
0	1	1	0 :	RT1-CS1 oscillation mode
0	1	1	1 :	IN1 pin external clock input mode
1	–	–	– :	Unavailable

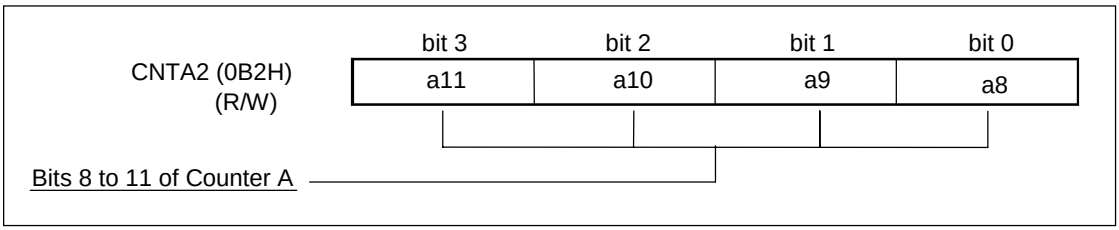
(3) A/D converter counter A registers (CNTA0 to 4)
The A/D converter counter A registers (CNTA0 to 4) are 4-bit special function registers (SFRs) to read/write the Counter A.

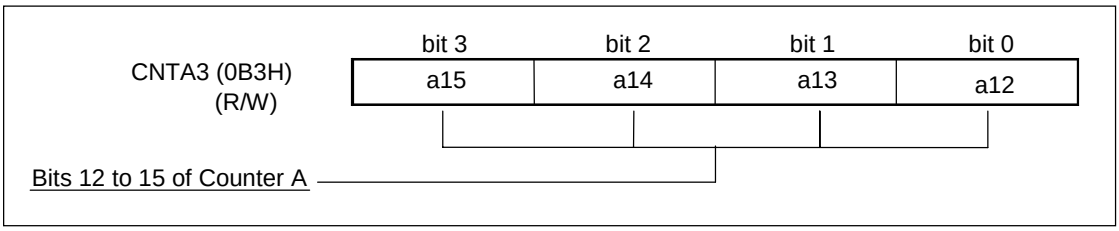
Note:

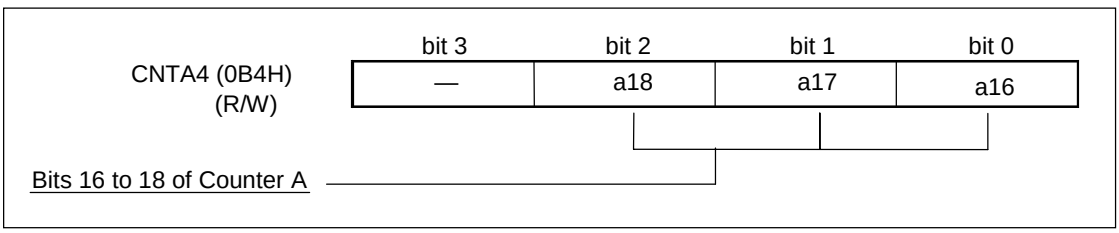
CNTA0 to CNTA3 are decimal counters and can handle only data from 0H to 9H.



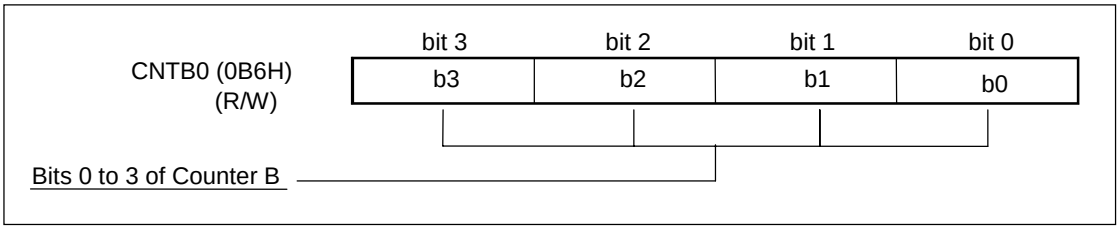


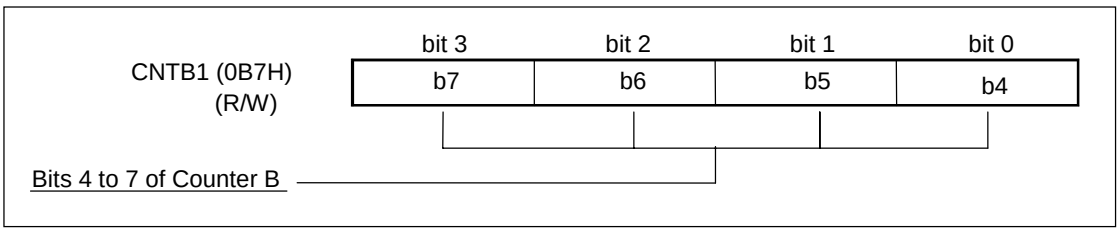


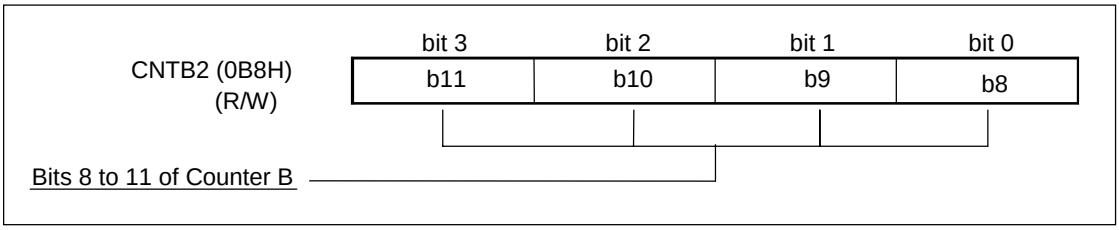


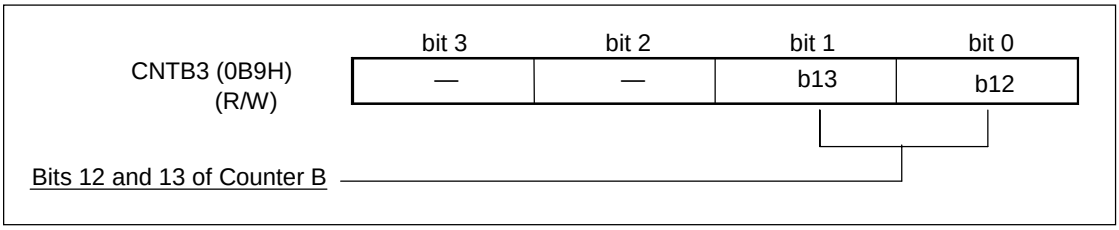


(4) A/D converter counter B registers (CNTB0 to 3)
The A/D converter counter B registers (CNTB0 to 3) are 4-bit special function registers (SFRs) to read/write the Counter B.









Tables 16-3 and 16-4 list A/D converter-related registers and pins.

Table 16-3 List of A/D Converter-Related Registers

Register name	Symbol	Address	Read/Write	Value at system reset
A/D converter control register 0	ADCON0	0BAH	R/W	8H
A/D converter control register 1	ADCON1	0BBH	R/W	0H
A/D converter counter A register 0	CNTA0	0B0H	R/W	0H
A/D converter counter A register 1	CNTA1	0B1H	R/W	0H
A/D converter counter A register 2	CNTA2	0B2H	R/W	0H
A/D converter counter A register 3	CNTA3	0B3H	R/W	0H
A/D converter counter A register 4	CNTA4	0B4H	R/W	8H
A/D converter counter B register 0	CNTB0	0B6H	R/W	0H
A/D converter counter B register 1	CNTB1	0B7H	R/W	0H
A/D converter counter B register 2	CNTB2	0B8H	R/W	0H
A/D converter counter B register 3	CNTB3	0B9H	R/W	0CH

Table 16-4 List of A/D Converter-Related Pins

Pin name	Pad No.	I/O	Function
RT0	29	O	Resistance sensor connection pin to measure Channel 0
CRT0	30	O	Resistance/capacitance sensor connection pin to measure Channel 0
RS0	31	O	Reference resistance connection pin for Channel 0
CS0	32	O	Reference capacitance connection pin for Channel 0
IN0	33	I	Oscillation input pin for Channel 0
RT1	37	O	Resistance sensor connection pin to measure Channel 1
RS1	36	O	Reference resistance connection pin for Channel 1
CS1	35	O	Reference capacitance connection pin for Channel 1
IN1	34	I	Oscillation input pin for Channel 1

Appendixes

Appendix A List of Special Function Registers

The Special Function Registers of the ML63611 are listed in Table A.

"—" indicates an invalid bit.

Table A Special Function Register List

Register name	Symbol	Address	bit 3	bit 2	bit 1	bit 0	R/W	Initial value at system reset
Port 0 data register	P0D	000H	P03	P02	P01	P00	R	Undefined
Reserved		001H to 009H						
Port A data register	PAD	00AH	PA3	PA2	PA1	PA0	R/W	0H
Port B data register	PBD	00BH	PB3	PB2	PB1	PB0	R/W	0H
Port C data register	PCD	00CH	PC3	PC2	PC1	PC0	R/W	0H
Reserved		00DH						
Port E data register	PED	00EH	PE3	PE2	PE1	PE0	R/W	0H
Reserved		00FH						
Port 0 control register 0	P0CON0	010H	P03MD	P02MD	P01MD	P00MD	R/W	0H
Port 0 control register 1	P0CON1	011H	—	—	P0PUD	P0F	R/W	0CH
Port 0 interrupt enable register	P0IE	012H	P03IE	P02IE	P01IE	P00IE	R/W	0H
Reserved		013H to 029H						
Port A control register 0	PACON0	02AH	PA1MD1	PA1MD0	PA0MD1	PA0MD0	R/W	0H
Port A control register 1	PACON1	02BH	PA3MD1	PA3MD0	PA2MD1	PA2MD0	R/W	0H
Port A direction register	PADIR	02CH	PA3DIR	PA2DIR	PA1DIR	PA0DIR	R/W	0H
Reserved		02DH						
Port B control register 0	PBCON0	02EH	PB1MD1	PB1MD0	PB0MD1	PB0MD0	R/W	0H
Port B control register 1	PBCON1	02FH	PB3MD1	PB3MD0	PB2MD1	PB2MD0	R/W	0H
Port B direction register	PBDIR	030H	PB3DIR	PB2DIR	PB1DIR	PB0DIR	R/W	0H
Port B interrupt enable register	PBIE	031H	PB3IE	PB2IE	PB1IE	PB0IE	R/W	0H
Port B mode register	PBMOD	032H	PBF	—	PB1MOD	PB0MOD	R/W	4H
Port C control register 0	PCCON0	033H	PC1MD1	PC1MD0	PC0MD1	PC0MD0	R/W	0H
Port C control register 1	PCCON1	034H	PC3MD1	PC3MD0	PC2MD1	PC2MD0	R/W	0H
Port C direction register	PCDIR	035H	PC3DIR	PC2DIR	PC1DIR	PC0DIR	R/W	0H
Port C interrupt enable register	PCIE	036H	PC3IE	PC2IE	PC1IE	PC0IE	R/W	0H
Port C mode register 0	PCMOD0	037H	—	—	—	PCF	R/W	0EH
Port C mode register 1	PCMOD1	038H	PC3MOD	PC2MOD	PC1MOD	PC0MOD	R/W	0H

Table A Special Function Register List (continued)

Register name	Symbol	Address	bit 3	bit 2	bit 1	bit 0	R/W	Initial value at system reset
Reserved		039H to 03CH						
Port E control register 0	PECON0	03DH	PE1MD1	PE1MD0	PE0MD1	PE0MD0	R/W	0H
Port E control register 1	PECON1	03EH	PE3MD1	PE3MD0	PE2MD1	PE2MD0	R/W	0H
Port B direction register	PEDIR	03FH	PE3DIR	PE2DIR	PE1DIR	PE0DIR	R/W	0H
Port E mode register	PEMOD	040H	PEF	PE2MOD	PE1MOD	PE0MOD	R/W	0H
Reserved		041H to 045H						
LP0 control register	LP0CON	046H	LP0MD3	LP0MD2	LP0MD1	LP0MD0	R/W	0H
Reserved		047H to 04FH						
Interrupt enable register 0	IE0	050H	EXI1	EXI0	EMD	—	R/W	1H
Interrupt enable register 1	IE1	051H	EXI5	—	EAD	EXI2	R/W	4H
Interrupt enable register 2	IE2	052H	ETM3	ETM2	ETM1	ETM0	R/W	0H
Interrupt enable register 3	IE3	053H	E10Hz	—	EST	ESR	R/W	4H
Interrupt enable register 4	IE4	054H	E2Hz	E4Hz	E16Hz	E32Hz	R/W	0H
Interrupt request register 0	IRQ0	055H	QXI1	QXI0	QMD	QWDT	R/W	0H
Interrupt request register 1	IRQ1	056H	QXI5	—	QAD	QXI2	R/W	4H
Interrupt request register 2	IRQ2	057H	QTM3	QTM2	QTM1	QTM0	R/W	0H
Interrupt request register 3	IRQ3	058H	Q10Hz	—	QST	QSR	R/W	4H
Interrupt request register 4	IRQ4	059H	Q2Hz	Q4Hz	Q16Hz	Q32Hz	R/W	0H
Reserved		05AH to 05FH						
Time base counter register 0	TBCR0	060H	16Hz	32Hz	64Hz	128Hz	R/W	0H
Time base counter register 1	TBCR1	061H	1Hz	2Hz	4Hz	8Hz	R/W	0H
Frequency control register	FCON	062H	—	OSCSEL	ENOSC	CPUCLK	R/W	8H
Reserved		063H						
100 Hz timer counter register	T100CR	064H	100C3	100C2	100C1	100C0	R/W	Undefined
10 Hz timer counter register	T10CR	065H	10C3	10C2	10C1	10C0	R/W	0H
100 Hz timer counter control register	T100CON	066H	—	—	—	ECNT	R/W	0EH
Reserved		067H						
Timer 0 data register L	TM0DL	068H	T0D3	T0D2	T0D1	T0D0	R/W	0H
Timer 0 data register H	TM0DH	069H	T0D7	T0D6	T0D5	T0D4	R/W	0H
Timer 1 data register L	TM1DL	06AH	T1D3	T1D2	T1D1	T1D0	R/W	0H
Timer 1 data register H	TM1DH	06BH	T1D7	T1D6	T1D5	T1D4	R/W	0H

Table A Special Function Register List (continued)

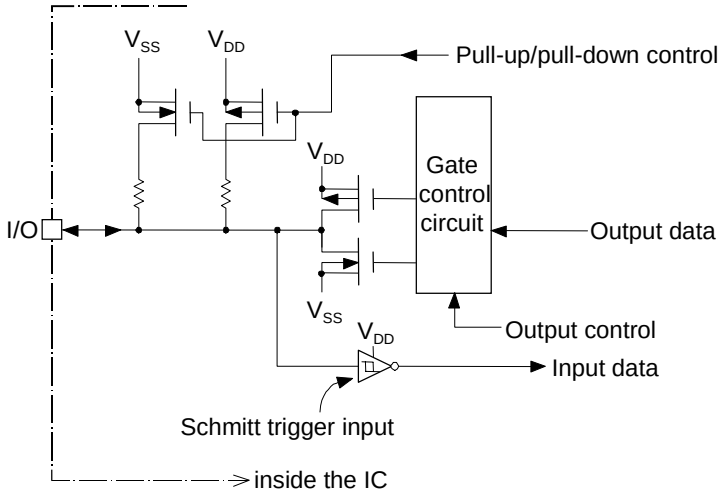
Register name	Symbol	Address	bit 3	bit 2	bit 1	bit 0	R/W	Initial value at system reset
Timer 0 counter register L	TM0CL	06CH	T0C3	T0C2	T0C1	T0C0	R/W	0H
Timer 0 counter register H	TM0CH	06DH	T0C7	T0C6	T0C5	T0C4	R/W	0H
Timer 1 counter register L	TM1CL	06EH	T1C3	T1C2	T1C1	T1C0	R/W	0H
Timer 1 counter register H	TM1CH	06FH	T1C7	T1C6	T1C5	T1C4	R/W	0H
Timer 0 control register 0	TM0CON0	070H	—	FMEAS0	TM0ECAP	TM0RUN	R/W	8H
Timer 0 control register 1	TM0CON1	071H	—	—	TM0CL1	TM0CL0	R/W	0CH
Timer 1 control register 0	TM1CON0	072H	—	—	TM1ECAP	TM1RUN	R/W	0CH
Timer 1 control register 1	TM1CON1	073H	—	—	TM1CL1	TM1CL0	R/W	0CH
Timer 0 status register	TM0STAT	074H	—	—	TM0CAP	TM0OVF	R	0CH
Timer 1 status register	TM1STAT	075H	—	—	TM1CAP	TM1OVF	R	0CH
Timer 2 data register L	TM2DL	076H	T2D3	T2D2	T2D1	T2D0	R/W	0H
Timer 2 data register H	TM2DH	077H	T2D7	T2D6	T2D5	T2D4	R/W	0H
Timer 3 data register L	TM3DL	078H	T3D3	T3D2	T3D1	T3D0	R/W	0H
Timer 3 data register H	TM3DH	079H	T3D7	T3D6	T3D5	T3D4	R/W	0H
Timer 2 counter register L	TM2CL	07AH	T2C3	T2C2	T2C1	T2C0	R/W	0H
Timer 2 counter register H	TM2CH	07BH	T2C7	T2C6	T2C5	T2C4	R/W	0H
Timer 3 counter register L	TM3CL	07CH	T3C3	T3C2	T3C1	T3C0	R/W	0H
Timer 3 counter register H	TM3CH	07DH	T3C7	T3C6	T3C5	T3C4	R/W	0H
Timer 2 control register 0	TM2CON0	07EH	—	FMEAS2	—	TM2RUN	R/W	0AH
Timer 2 control register 1	TM2CON1	07FH	—	—	TM2CL1	TM2CL0	R/W	0CH
Timer 3 control register 0	TM3CON0	080H	—	—	—	TM3RUN	R/W	0EH
Timer 3 control register 1	TM3CON1	081H	—	—	TM3CL1	TM3CL0	R/W	0CH
Timer 2 status register	TM2STAT	082H	—	—	—	TM2OVF	R	0EH
Timer 3 status register	TM3STAT	083H	—	—	—	TM3OVF	R	0EH
Reserved		084H to 08FH						
Display control register 0	DSPCON0	090H	BISEL	PDWN	ALLON	LCDON	R/W	0H
Display control register 1	DSPCON1	091H	—	—	DT1	DT0	R/W	0CH
Display contrast register	DSPCNT	092H	CN3	CN2	CN1	CN0	R/W	0H
Reserved		093H						
Battery low detect control register	BLDCON	094H	BLDF	ENBL	LD1	LD0	R/W	0H
Halver control register	VHCON	095H	—	—	—	VH	R/W	0FH
Tempo register	TEMPO	096H	TMP3	TMP2	TMP1	TMP0	R/W	0H
Melody driver control register	MDCON	097H	MSF	EMBD	MBM1	MBM0	R/W	0H
Reserved		098H to 09EH						
Watchdog timer control register	WDTCON	09FH	d3	d2	d1	d0	W	—
Reserved		0A0H to 0A3H						

Table A Special Function Register List (continued)

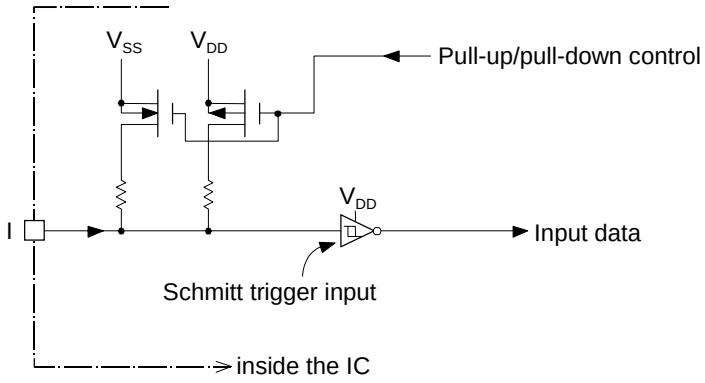
Register name	Symbol	Address	bit 3	bit 2	bit 1	bit 0	R/W	Initial value at system reset
Serial port send buffer L	STBUFL	0A4H	TB3	TB2	TB1	TB0	R/W	0H
Serial port send buffer H	STBUFH	0A5H	TB7	TB6	TB5	TB4	R/W	0H
Serial port send control register 0	STCON0	0A6H	STSTB	STL1	STL0	STMOD	R/W	0H
Serial port send control register 1	STCON1	0A7H	STLMB	STPOE	STPEN	STCLK	R/W	0H
Serial port receive buffer L	SRBUFL	0A8H	RB3	RB2	RB1	RB0	R	0H
Serial port receive buffer H	SRBUFH	0A9H	RB7	RB6	RB5	RB4	R	0H
Serial port receive control register 0	SRCON0	0AAH	SREN	SRL1	SRL0	SRMOD	R/W	0H
Serial port receive control register 1	SRCON1	0ABH	SRLMB	SRPOE	SRPEN	SRCLK	R/W	0H
Serial port receive baud rate setting register	SRBRT	0ACH	—	—	BRT1	BRT0	R/W	0CH
Serial port status register	SSTAT	0ADH	BFULL	PERR	OERR	FERR	R	0H
Reserved		0AEH and 0AFH						
A/D converter counter A register 0	CNTA0	0B0H	CA3	CA2	CA1	CA0	R/W	0H
A/D converter counter A register 1	CNTA1	0B1H	CA7	CA6	CA5	CA4	R/W	0H
A/D converter counter A register 2	CNTA2	0B2H	CA11	CA10	CA9	CA8	R/W	0H
A/D converter counter A register 3	CNTA3	0B3H	CA15	CA14	CA13	CA12	R/W	0H
A/D converter counter A register 4	CNTA4	0B4H	—	CA18	CA17	CA16	R/W	8H
Reserved		0B5H						
A/D converter counter B register 0	CNTB0	0B6H	CB3	CB2	CB1	CB0	R/W	0H
A/D converter counter B register 1	CNTB1	0B7H	CB7	CB6	CB5	CB4	R/W	0H
A/D converter counter B register 2	CNTB2	0B8H	CB11	CB10	CB9	CB8	R/W	0H
A/D converter counter B register 3	CNTB3	0B9H	—	—	CB13	CB12	R/W	0CH
A/D converter control register 0	ADCON0	0BAH	—	STV	SADI	EADC	R/W	8H
A/D converter control register 1	ADCON1	0BBH	OM3	OM2	OM1	OM0	R/W	0H
Reserved		0BCH to 0F1H						
RA register 0	RA0	0F2H	a3	a2	a1	a0	R/W	0H
RA register 1	RA1	0F3H	a7	a6	a5	a4	R/W	0H
RA register 2	RA2	0F4H	a11	a10	a9	a8	R/W	0H
RA register 3	RA3	0F5H	a15	a14	a13	a12	R/W	0H
Register stack pointer	RSP	0F6H	rsp3	rsp2	rsp1	rsp0	R/W	0H
Stack pointer	SP	0F7H	sp3	sp2	sp1	sp0	R	0H
Reserved		0F8H						
Y register	Y	0F9H	y3	y2	y1	y0	R/W	0H
X register	X	0FAH	x3	x2	x1	x0	R/W	0H
L register	L	0FBH	l3	l2	l1	l0	R/W	0H
H register	H	0FCH	h3	h2	h1	h0	R/W	0H
Current bank register	CBR	0FDH	c3	c2	c1	c0	R/W	0H
Extra bank register	EBR	0FEH	e3	e2	e1	e0	R/W	0H
Master interrupt enable flag register	MIEF	0FFH	—	—	—	MIE	R	0EH

Appendix B Input/Output Circuit Configuration

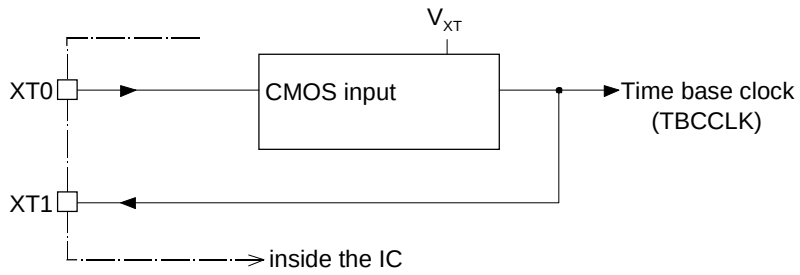
(1) I/O Port (PA.0–PA.3, PB.0–PB.3, PC.0–PC.3, PE.0–PE.3)



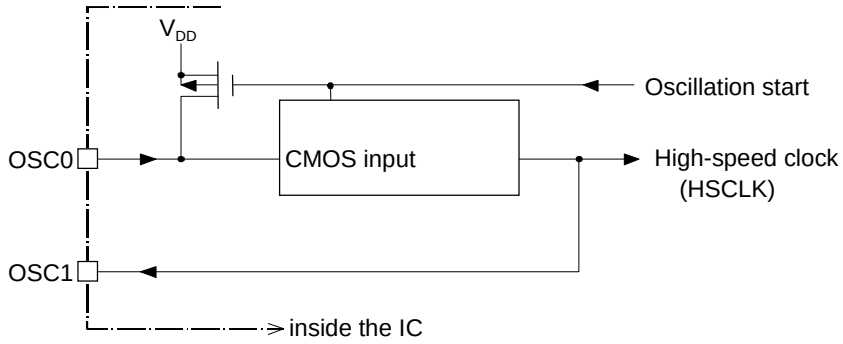
(2) Input Port (P0.0–P0.3)



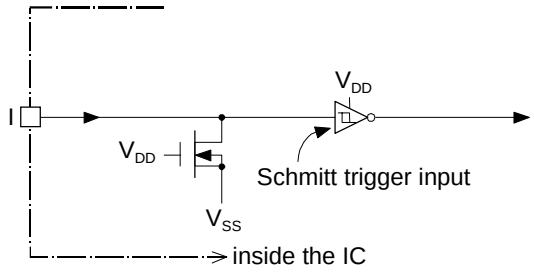
(3) Low-Speed Oscillation Circuit



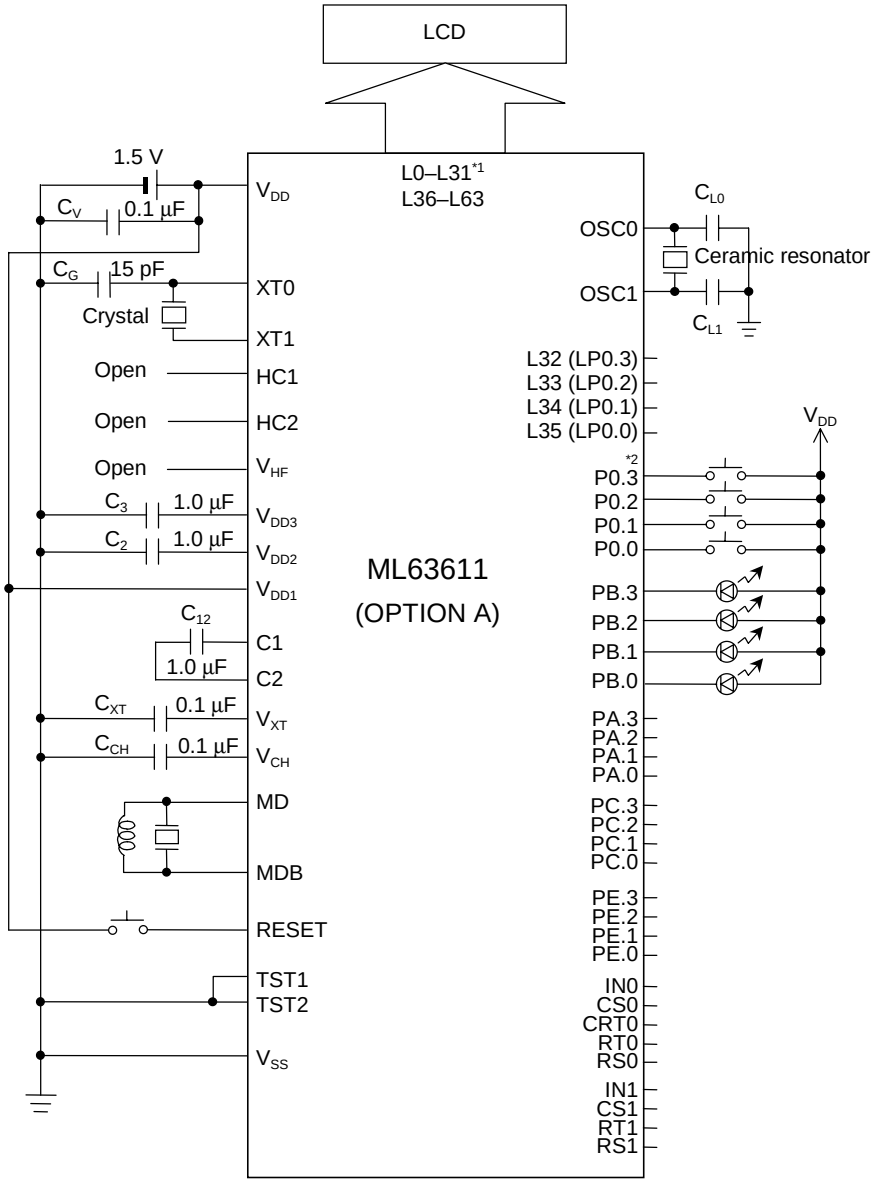
(4) High-Speed Oscillation Circuit



(5) RESET, TST1, and TST2 Inputs



Appendix C Peripheral Circuit Examples

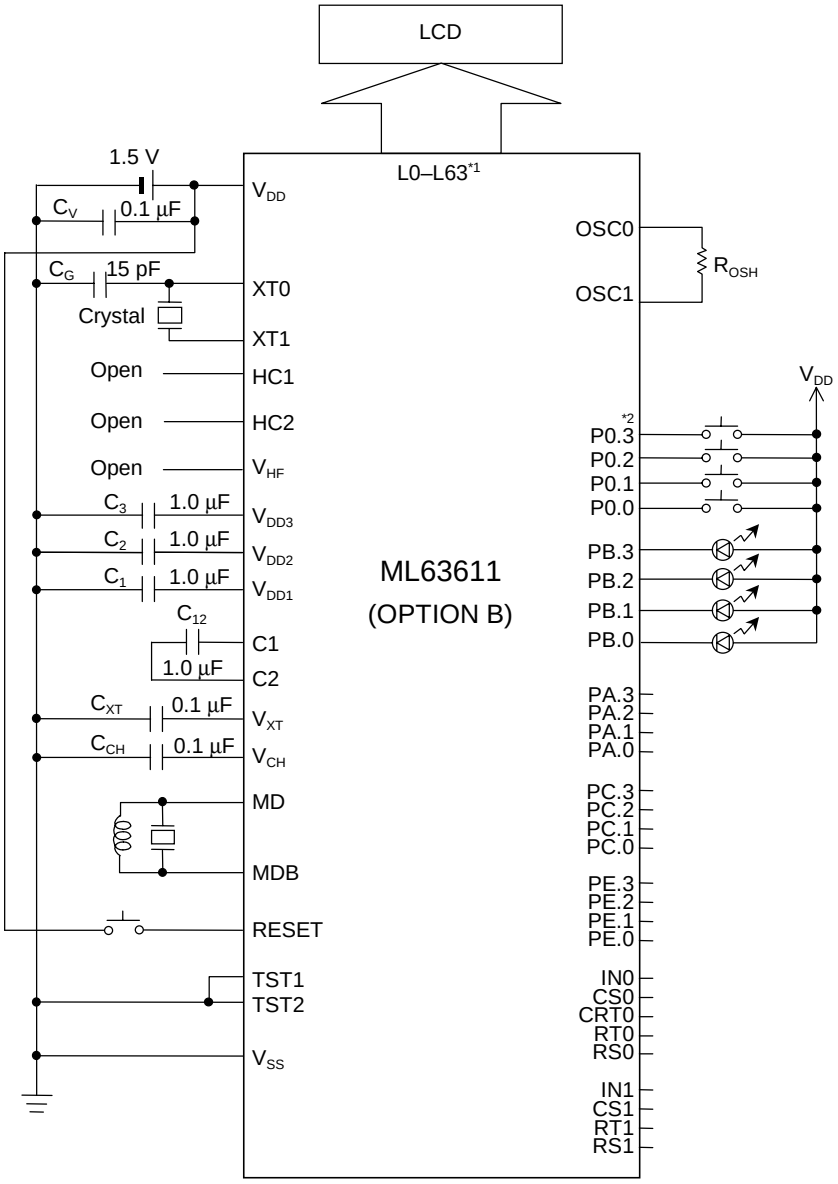


*1 When L0 to L3 are made COM pins, L4 to L31 and L36 to L63 are made SEG pins, and L32 to L35 are made port pins

*2 When P0PUD = "0" (input with pull-down resistor) and the function of a reset due to simultaneous key depression of four bits are specified using the mask option

- Ceramic oscillation is selected for high-speed oscillation.
- C_V is an IC power supply bypass capacitor.
- Capacitance values for C_2, C_3, C_{12}, C_{XT}, C_{CH}, and C_G are only for reference.

Figure C-1 OPTION A Peripheral Circuit Example

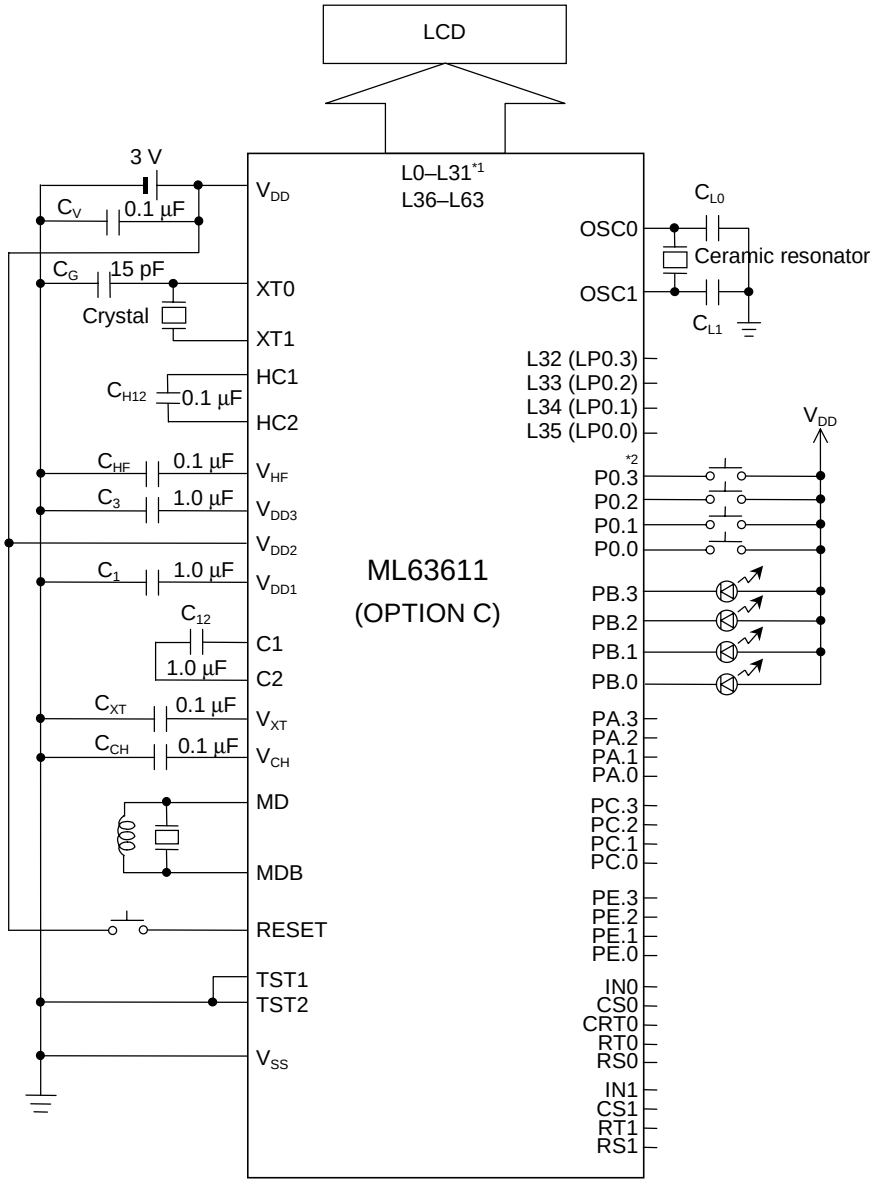


*1 When L0 to L3 are made COM pins and L4 to L63 are made SEG pins

*2 When P0PUD = "0" (input with pull-down resistor) and the function of a reset due to simultaneous key depression of four bits are specified using the mask option

- Ceramic oscillation is selected for high-speed oscillation.
- C_V is an IC power supply bypass capacitor.
- Capacitance values for C_1, C_2, C_3, C_{12}, C_{XT}, C_{CH}, and C_G are only for reference.

Figure C-2 OPTION B Peripheral Circuit Example

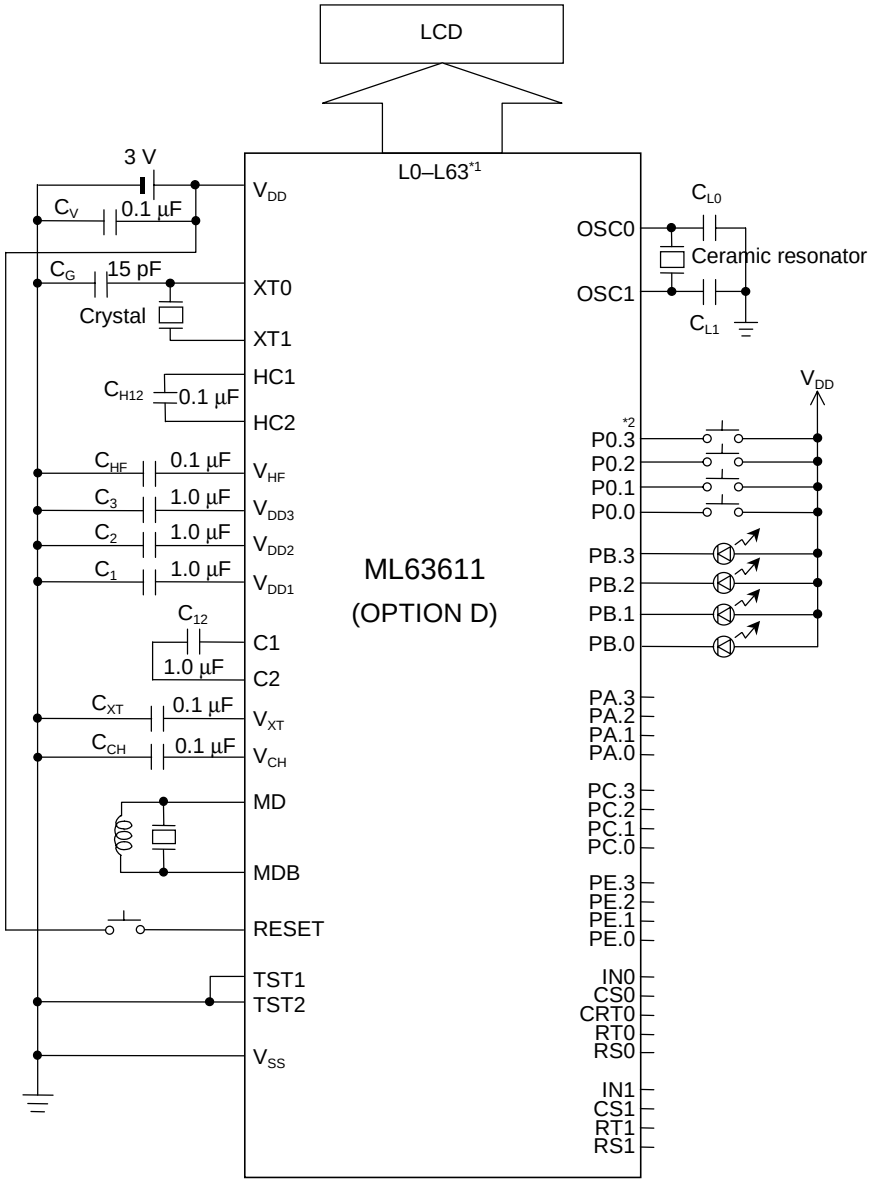


*1 When L0 to L3 are made COM pins, L4 to L31 and L36 to L63 are made SEG pins, and L32 to L35 are made port pins

*2 When P0PUD = "0" (input with pull-down resistor) and the function of a reset due to simultaneous key depression of four bits are specified using the mask option

- Ceramic oscillation is selected for high-speed oscillation.
- C_V is an IC power supply bypass capacitor.
- Capacitance values for C_1, C_3, C_{12}, C_{XT}, C_{CH}, C_{H12}, C_{HF}, and C_G are only for reference.

Figure C-3 OPTION C Peripheral Circuit Example



*1 When L0 to L3 are made COM pins and L4 to L63 are made SEG pins

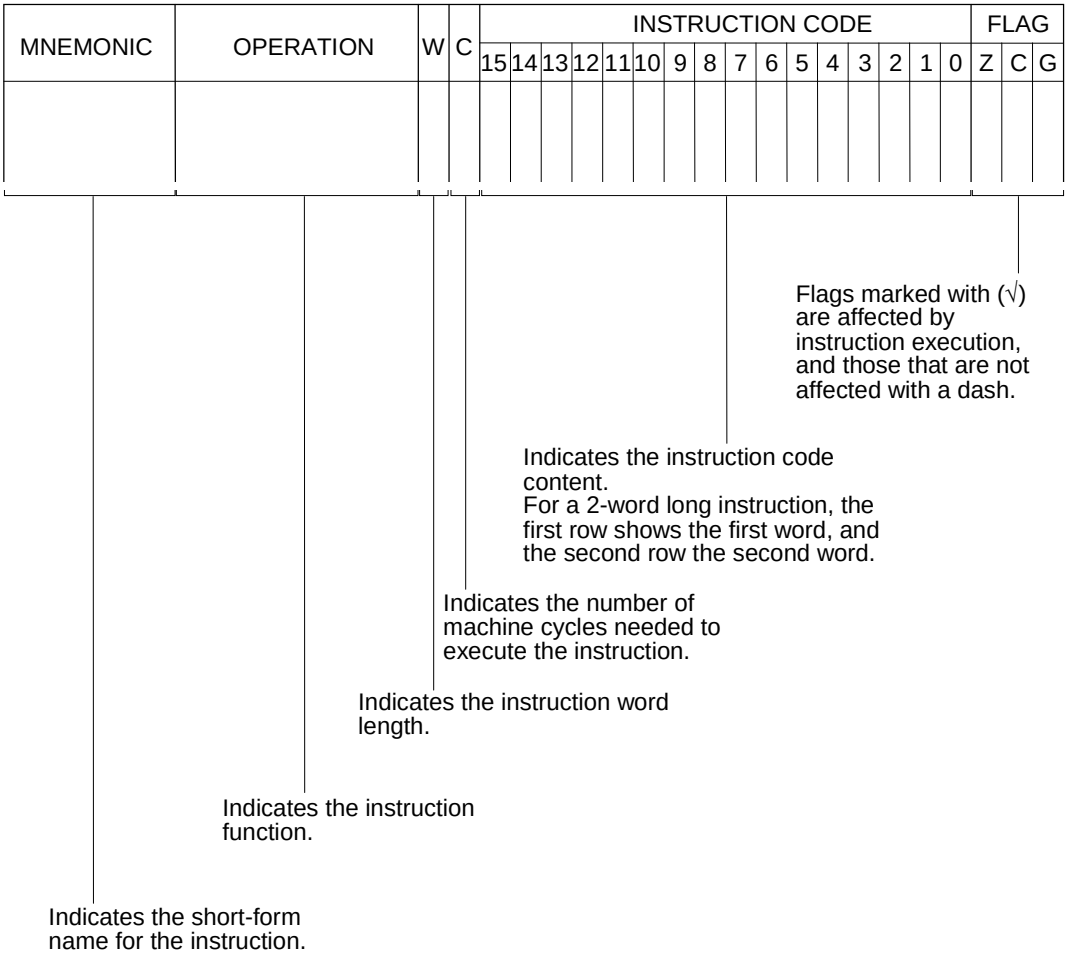
*2 When P0PUD = "0" (input with pull-down resistor) and the function of a reset due to simultaneous key depression of four bits are specified using the mask option

- Ceramic oscillation is selected for high-speed oscillation.
- C_V is an IC power supply bypass capacitor.
- Capacitance values for C_1, C_2, C_3, C_{12}, C_{XT}, C_{CH}, C_{H12}, C_{HF}, and C_G are only for reference.

Figure C-4 OPTION D Peripheral Circuit Example

Appendix D Instruction List

The format used in the list of instructions is indicated below.



Transfer Instructions

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
MOV direct,A	direct ← A	1	1	1	1	0	0	r_{11}	r_{10}	r_9	r_8	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	—	—	—
MOV [HL],A	[HL] ← A	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	—	—	—
MOV [XY],A	[XY] ← A	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	—	—	—
MOV E:[HL],A	E:[HL] ← A	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	—	—	—
MOV E:[XY],A	E:[XY] ← A	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	—	—	—
MOV [HL+],A	[HL] ← A, HL ← HL + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	—	—	√
MOV [XY+],A	[XY] ← A, XY ← XY + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	—	—	√
MOV E:[HL+],A	E:[HL] ← A, HL ← HL + 1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	—	—	√
MOV E:[XY+],A	E:[XY] ← A, XY ← XY + 1	1	1	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	—	—	√
MOV \cur,#i4	cur,A ← i4	1	1	0	1	0	0	i_3	i_2	i_1	i_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
MOV [HL],#i4	[HL],A ← i4	1	1	0	0	0	0	0	1	1	0	0	1	1	0	i_3	i_2	i_1	i_0	√	—	—
MOV [XY],#i4	[XY],A ← i4	1	1	0	0	0	0	0	1	1	0	0	1	1	1	i_3	i_2	i_1	i_0	√	—	—
MOV E:[HL],#i4	E:[HL],A ← i4	1	1	0	0	0	0	0	1	1	0	0	1	0	0	i_3	i_2	i_1	i_0	√	—	—
MOV E:[XY],#i4	E:[XY],A ← i4	1	1	0	0	0	0	0	1	1	0	0	1	0	1	i_3	i_2	i_1	i_0	√	—	—
MOV [HL+],#i4	[HL],A ← i4, HL ← HL + 1	1	1	0	0	0	0	0	1	1	1	0	1	1	0	i_3	i_2	i_1	i_0	√	—	√
MOV [XY+],#i4	[XY],A ← i4, XY ← XY + 1	1	1	0	0	0	0	0	1	1	1	0	1	1	1	i_3	i_2	i_1	i_0	√	—	√
MOV E:[HL+],#i4	E:[HL],A ← i4, HL ← HL + 1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	i_3	i_2	i_1	i_0	√	—	√
MOV E:[XY+],#i4	E:[XY],A ← i4, XY ← XY + 1	1	1	0	0	0	0	0	1	1	1	0	1	0	1	i_3	i_2	i_1	i_0	√	—	√
MOV A,#i4	A ← i4	1	1	0	0	0	0	0	0	0	1	1	1	0	0	i_3	i_2	i_1	i_0	√	—	—
MOV A,direct	A ← direct	1	1	1	1	0	1	r_{11}	r_{10}	r_9	r_8	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
MOV A,[HL]	A ← [HL]	1	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	√	—	—
MOV A,[XY]	A ← [XY]	1	1	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	√	—	—
MOV A,E:[HL]	A ← E:[HL]	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	√	—	—
MOV A,E:[XY]	A ← E:[XY]	1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	√	—	—
MOV A,[HL+]	A ← [HL], HL ← HL + 1	1	1	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	√	—	√
MOV A,[XY+]	A ← [XY], XY ← XY + 1	1	1	0	0	0	0	0	0	1	1	0	0	1	1	1	0	0	0	√	—	√
MOV A,E:[HL+]	A ← E:[HL], HL ← HL + 1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	√	—	√
MOV A,E:[XY+]	A ← E:[XY], XY ← XY + 1	1	1	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	√	—	√
XCH A,sfr	A ↔ sfr	1	1	0	0	1	0	1	1	1	0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	—	—	—
XCH A,\cur	A ↔ cur	1	1	0	0	1	1	1	1	1	0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	—	—	—
XCH A,[HL]	A ↔ [HL]	1	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	1	—	—	—
XCH A,[XY]	A ↔ [XY]	1	1	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	1	—	—	—
XCH A,E:[HL]	A ↔ E:[HL]	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	—	—	—
XCH A,E:[XY]	A ↔ E:[XY]	1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	1	—	—	—
XCH A,[HL+]	A ↔ [HL], HL ← HL + 1	1	1	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	1	—	—	√
XCH A,[XY+]	A ↔ [XY], XY ← XY + 1	1	1	0	0	0	0	0	0	1	1	0	0	1	1	1	0	0	1	—	—	√
XCH A,E:[HL+]	A ↔ E:[HL], HL ← HL + 1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	1	—	—	√
XCH A,E:[XY+]	A ↔ E:[XY], XY ← XY + 1	1	1	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	1	—	—	√

Rotate Instructions

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
ROL sfr	C ← {$_3$sfr$_0$} ← C, A ← sfr	1	1	0	0	1	0	0	0	1	0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
ROL \cur	C ← {$_3$cur$_0$} ← C, A ← cur	1	1	0	0	1	1	0	0	1	0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
ROL [HL]	C ← {$_3$[HL]$_0$} ← C, A ← [HL]	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	√	√	—
ROL [XY]	C ← {$_3$[XY]$_0$} ← C, A ← [XY]	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	1	1	0	√	√	—
ROL E:[HL]	C ← {$_3$E:[HL]$_0$} ← C, A ← E:[HL]	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	√	√	—
ROL E:[XY]	C ← {$_3$E:[XY]$_0$} ← C, A ← E:[XY]	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1	0	√	√	—
ROL [HL+]	C ← {$_3$[HL]$_0$} ← C, A ← [HL], HL ← HL + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	0	1	1	1	0	√	√	√
ROL [XY+]	C ← {$_3$[XY]$_0$} ← C, A ← [XY], XY ← XY + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	0	√	√	√
ROL E:[HL+]	C ← {$_3$E:[HL]$_0$} ← C, A ← E:[HL], HL ← HL + 1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	√	√	√
ROL E:[XY+]	C ← {$_3$E:[XY]$_0$} ← C, A ← E:[XY], XY ← XY + 1	1	1	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	0	√	√	√
ROR sfr	C → {$_3$sfr$_0$} → C, A ← sfr	1	1	0	0	1	0	0	0	1	1	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
ROR \cur	C → {$_3$cur$_0$} → C, A ← cur	1	1	0	0	1	1	0	0	1	1	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
ROR [HL]	C → {$_3$[HL]$_0$} → C, A ← [HL]	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	√	√	—
ROR [XY]	C → {$_3$[XY]$_0$} → C, A ← [XY]	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	1	1	1	√	√	—
ROR E:[HL]	C → {$_3$E:[HL]$_0$} → C, A ← E:[HL]	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	√	√	—
ROR E:[XY]	C → {$_3$E:[XY]$_0$} → C, A ← E:[XY]	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1	1	√	√	—
ROR [HL+]	C → {$_3$[HL]$_0$} → C, A ← [HL], HL ← HL + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	0	1	1	1	1	√	√	√
ROR [XY+]	C → {$_3$[XY]$_0$} → C, A ← [XY], XY ← XY + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	√	√	√
ROR E:[HL+]	C → {$_3$E:[HL]$_0$} → C, A ← E:[HL], HL ← HL + 1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	√	√	√
ROR E:[XY+]	C → {$_3$E:[XY]$_0$} → C, A ← E:[XY], XY ← XY + 1	1	1	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	√	√	√

Increment/Decrement Instructions

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
INC sfr	sfr,A ← sfr + 1	1	1	0	0	1	0	0	0	0	0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
INC \cur	cur,A ← cur + 1	1	1	0	0	1	1	0	0	0	0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
INC [HL]	[HL],A ← [HL] + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	√	√	—
INC [XY]	[XY],A ← [XY] + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	1	0	0	0	0	√	√	—
INC E:[HL]	E:[HL],A ← E:[HL] + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	√	√	—
INC E:[XY]	E:[XY],A ← E:[XY] + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	√	√	—
INC [HL+]	[HL],A ← [HL] + 1, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	0	1	0	0	0	√	√	√
INC [XY+]	[XY],A ← [XY] + 1, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	1	1	0	0	0	√	√	√
INC E:[HL+]	E:[HL],A ← E:[HL] + 1, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	√	√	√
INC E:[XY+]	E:[XY],A ← E:[XY] + 1, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	1	0	0	0	√	√	√
DEC sfr	sfr,A ← sfr – 1	1	1	0	0	1	0	0	0	0	1	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
DEC \cur	cur,A ← cur – 1	1	1	0	0	1	1	0	0	0	1	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
DEC [HL]	[HL],A ← [HL] – 1	1	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	√	√	—
DEC [XY]	[XY],A ← [XY] – 1	1	1	0	0	0	0	0	1	0	1	0	0	1	1	0	0	0	1	√	√	—
DEC E:[HL]	E:[HL],A ← E:[HL] – 1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	√	√	—
DEC E:[XY]	E:[XY],A ← E:[XY] – 1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	√	√	—
DEC [HL+]	[HL],A ← [HL] – 1, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	0	1	0	0	1	√	√	√
DEC [XY+]	[XY],A ← [XY] – 1, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	1	1	0	0	1	√	√	√
DEC E:[HL+]	E:[HL],A ← E:[HL] – 1, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	√	√	√
DEC E:[XY+]	E:[XY],A ← E:[XY] – 1, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	1	0	0	1	√	√	√

Arithmetic Instructions

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
ADD sfr,A	sfr,A ← sfr + A	1	1	0	0	1	0	0	1	0	0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
ADD \cur,A	cur,A ← cur + A	1	1	0	0	1	1	0	1	0	0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
ADD [HL],A	[HL],A ← [HL] + A	1	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	1	0	√	√	—
ADD [XY],A	[XY],A ← [XY] + A	1	1	0	0	0	0	0	1	0	1	0	0	1	1	0	0	1	0	√	√	—
ADD E:[HL],A	E:[HL],A ← E:[HL] + A	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	√	√	—
ADD E:[XY],A	E:[XY],A ← E:[XY] + A	1	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0	√	√	—
ADD [HL+],A	[HL],A ← [HL] + A, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	0	1	0	1	0	√	√	√
ADD [XY+],A	[XY],A ← [XY] + A, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	1	1	0	1	0	√	√	√
ADD E:[HL+],A	E:[HL],A ← E:[HL] + A, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	√	√	√
ADD E:[XY+],A	E:[XY],A ← E:[XY] + A, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	1	0	1	0	√	√	√
ADD \cur,#i4	cur,A ← cur + i4	1	1	1	0	0	0	i_3	i_2	i_1	i_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
ADD [HL],#i4	[HL],A ← [HL] + i4	1	1	0	0	0	0	0	0	0	0	1	0	1	0	i_3	i_2	i_1	i_0	√	√	—
ADD [XY],#i4	[XY],A ← [XY] + i4	1	1	0	0	0	0	0	0	0	0	1	0	1	1	i_3	i_2	i_1	i_0	√	√	—
ADD E:[HL],#i4	E:[HL],A ← E:[HL] + i4	1	1	0	0	0	0	0	0	0	0	1	0	0	0	i_3	i_2	i_1	i_0	√	√	—
ADD E:[XY],#i4	E:[XY],A ← E:[XY] + i4	1	1	0	0	0	0	0	0	0	0	1	0	0	1	i_3	i_2	i_1	i_0	√	√	—
ADD [HL+],#i4	[HL],A ← [HL] + i4, HL ← HL + 1	1	1	0	0	0	0	0	0	0	1	1	0	1	0	i_3	i_2	i_1	i_0	√	√	√
ADD [XY+],#i4	[XY],A ← [XY] + i4, XY ← XY + 1	1	1	0	0	0	0	0	0	0	1	1	0	1	1	i_3	i_2	i_1	i_0	√	√	√
ADD E:[HL+],#i4	E:[HL],A ← E:[HL] + i4, HL ← HL + 1	1	1	0	0	0	0	0	0	0	1	1	0	0	0	i_3	i_2	i_1	i_0	√	√	√
ADD E:[XY+],#i4	E:[XY],A ← E:[XY] + i4, XY ← XY + 1	1	1	0	0	0	0	0	0	0	1	1	0	0	1	i_3	i_2	i_1	i_0	√	√	√
ADC sfr,A	sfr,A ← sfr + A + C	1	1	0	0	1	0	0	1	0	1	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
ADC \cur,A	cur,A ← cur + A + C	1	1	0	0	1	1	0	1	0	1	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
ADC [HL],A	[HL],A ← [HL] + A + C	1	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	1	1	√	√	—
ADC [XY],A	[XY],A ← [XY] + A + C	1	1	0	0	0	0	0	1	0	1	0	0	1	1	0	0	1	1	√	√	—
ADC E:[HL],A	E:[HL],A ← E:[HL] + A + C	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	√	√	—
ADC E:[XY],A	E:[XY],A ← E:[XY] + A + C	1	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	1	√	√	—
ADC [HL+],A	[HL],A ← [HL] + A + C, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	0	1	0	1	1	√	√	√
ADC [XY+],A	[XY],A ← [XY] + A + C, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	1	1	0	1	1	√	√	√
ADC E:[HL+],A	E:[HL],A ← E:[HL] + A + C, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	1	√	√	√
ADC E:[XY+],A	E:[XY],A ← E:[XY] + A + C, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	1	0	1	1	√	√	√

Arithmetic Instructions (continued)

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
ADCD sfr,A	sfr,A ← decimal adjustment {sfr + A + C}	1	1	0	0	1	0	0	1	1	0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
ADCD \cur,A	cur,A ← decimal adjustment {cur + A + C}	1	1	0	0	1	1	0	1	1	0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
ADCD [HL],A	[HL],A ← decimal adjustment {[HL] + A + C}	1	1	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	√	√	—
ADCD [XY],A	[XY],A ← decimal adjustment {[XY] + A + C}	1	1	0	0	0	0	0	1	0	1	0	0	1	1	0	1	0	0	√	√	—
ADCD E:[HL],A	E:[HL],A ← decimal adjustment {E:[HL] + A + C}	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	√	√	—
ADCD E:[XY],A	E:[XY],A ← decimal adjustment {E:[XY] + A + C}	1	1	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0	√	√	—
ADCD [HL+],A	[HL],A ← decimal adjustment {[HL] + A + C}, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	0	1	1	0	0	√	√	√
ADCD [XY+],A	[XY],A ← decimal adjustment {[XY] + A + C}, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	1	1	1	0	0	√	√	√
ADCD E:[HL+],A	E:[HL],A ← decimal adjustment {E:[HL] + A + C}, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	1	0	0	√	√	√
ADCD E:[XY+],A	E:[XY],A ← decimal adjustment {E:[XY] + A + C}, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	1	1	0	0	√	√	√
ADCJ \cur,n	cur,A ← n-ary adjustment {cur + C}	1	1	0	0	0	1	0	n_2	n_1	n_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
ADCJ [HL],n	[HL],A ← n-ary adjustment {[HL] + C}	1	1	0	0	0	0	0	1	1	0	0	0	1	0	0	n_2	n_1	n_0	√	√	—
ADCJ [XY],n	[XY],A ← n-ary adjustment {[XY] + C}	1	1	0	0	0	0	0	1	1	0	0	0	1	1	0	n_2	n_1	n_0	√	√	—
ADCJ E:[HL],n	E:[HL],A ← n-ary adjustment {E:[HL] + C}	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	n_2	n_1	n_0	√	√	—
ADCJ E:[XY],n	E:[XY],A ← n-ary adjustment {E:[XY] + C}	1	1	0	0	0	0	0	1	1	0	0	0	0	1	0	n_2	n_1	n_0	√	√	—
ADCJ [HL+],n	[HL],A ← n-ary adjustment {[HL] + C}, HL ← HL + 1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	0	n_2	n_1	n_0	√	√	√
ADCJ [XY+],n	[XY],A ← n-ary adjustment {[XY] + C}, XY ← XY + 1	1	1	0	0	0	0	0	1	1	1	0	0	1	1	0	n_2	n_1	n_0	√	√	√
ADCJ E:[HL+],n	E:[HL],A ← n-ary adjustment {E:[HL] + C}, HL ← HL + 1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	n_2	n_1	n_0	√	√	√
ADCJ E:[XY+],n	E:[XY],A ← n-ary adjustment {E:[XY] + C}, XY ← XY + 1	1	1	0	0	0	0	0	1	1	1	0	0	0	1	0	n_2	n_1	n_0	√	√	√

Arithmetic Instructions (continued)

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
SUB sfr,A	sfr,A ← sfr – A	1	1	0	0	1	0	0	1	1	1	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
SUB \cur,A	cur,A ← cur – A	1	1	0	0	1	1	0	1	1	1	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
SUB [HL],A	[HL],A ← [HL] – A	1	1	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	1	√	√	—
SUB [XY],A	[XY],A ← [XY] – A	1	1	0	0	0	0	0	1	0	1	0	0	1	1	0	1	0	1	√	√	—
SUB E:[HL],A	E:[HL],A ← E:[HL] – A	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	√	√	—
SUB E:[XY],A	E:[XY],A ← E:[XY] – A	1	1	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	1	√	√	—
SUB [HL+],A	[HL],A ← [HL] – A, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	0	1	1	0	1	√	√	√
SUB [XY+],A	[XY],A ← [XY] – A, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	1	1	1	0	1	√	√	√
SUB E:[HL+],A	E:[HL],A ← E:[HL] – A, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	1	0	1	√	√	√
SUB E:[XY+],A	E:[XY],A ← E:[XY] – A, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	1	1	0	1	√	√	√
SUB \cur,#i4	cur,A ← cur – i4	1	1	1	0	0	1	i_3	i_2	i_1	i_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
SUB [HL],#i4	[HL],A ← [HL] – i4	1	1	0	0	0	0	0	0	1	0	1	0	1	0	i_3	i_2	i_1	i_0	√	√	—
SUB [XY],#i4	[XY],A ← [XY] – i4	1	1	0	0	0	0	0	0	1	0	1	0	1	1	i_3	i_2	i_1	i_0	√	√	—
SUB E:[HL],#i4	E:[HL],A ← E:[HL] – i4	1	1	0	0	0	0	0	0	1	0	1	0	0	0	i_3	i_2	i_1	i_0	√	√	—
SUB E:[XY],#i4	E:[XY],A ← E:[XY] – i4	1	1	0	0	0	0	0	0	1	0	1	0	0	1	i_3	i_2	i_1	i_0	√	√	—
SUB [HL+],#i4	[HL],A ← [HL] – i4, HL ← HL + 1	1	1	0	0	0	0	0	0	1	1	1	0	1	0	i_3	i_2	i_1	i_0	√	√	√
SUB [XY+],#i4	[XY],A ← [XY] – i4, XY ← XY + 1	1	1	0	0	0	0	0	0	1	1	1	0	1	1	i_3	i_2	i_1	i_0	√	√	√
SUB E:[HL+],#i4	E:[HL],A ← E:[HL] – i4, HL ← HL + 1	1	1	0	0	0	0	0	0	1	1	1	0	0	0	i_3	i_2	i_1	i_0	√	√	√
SUB E:[XY+],#i4	E:[XY],A ← E:[XY] – i4, XY ← XY + 1	1	1	0	0	0	0	0	0	1	1	1	0	0	1	i_3	i_2	i_1	i_0	√	√	√
SBC sfr,A	sfr,A ← sfr – A – C	1	1	0	0	1	0	1	0	0	0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
SBC \cur,A	cur,A ← cur – A – C	1	1	0	0	1	1	1	0	0	0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
SBC [HL],A	[HL],A ← [HL] – A – C	1	1	0	0	0	0	0	1	0	1	0	0	1	0	0	1	1	0	√	√	—
SBC [XY],A	[XY],A ← [XY] – A – C	1	1	0	0	0	0	0	1	0	1	0	0	1	1	0	1	1	0	√	√	—
SBC E:[HL],A	E:[HL],A ← E:[HL] – A – C	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	1	0	√	√	—
SBC E:[XY],A	E:[XY],A ← E:[XY] – A – C	1	1	0	0	0	0	0	1	0	1	0	0	0	1	0	1	1	0	√	√	—
SBC [HL+],A	[HL],A ← [HL] – A – C, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	0	1	1	1	0	√	√	√
SBC [XY+],A	[XY],A ← [XY] – A – C, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	1	1	1	1	0	√	√	√
SBC E:[HL+],A	E:[HL],A ← E:[HL] – A – C, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	√	√	√
SBC E:[XY+],A	E:[XY],A ← E:[XY] – A – C, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	√	√	√

Arithmetic Instructions (continued)

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
SBCD sfr,A	sfr,A ← decimal adjustment {sfr – A – C}	1	1	0	0	1	0	1	0	0	1	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
SBCD \cur,A	cur,A ← decimal adjustment {cur – A – C}	1	1	0	0	1	1	1	0	0	1	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
SBCD [HL],A	[HL],A ← decimal adjustment {[HL] – A – C}	1	1	0	0	0	0	0	1	0	1	0	0	1	0	0	1	1	1	√	√	—
SBCD [XY],A	[XY],A ← decimal adjustment {[XY] – A – C}	1	1	0	0	0	0	0	1	0	1	0	0	1	1	0	1	1	1	√	√	—
SBCD E:[HL],A	E:[HL],A ← decimal adjustment {E:[HL] – A – C}	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	1	1	√	√	—
SBCD E:[XY],A	E:[XY],A ← decimal adjustment {E:[XY] – A – C}	1	1	0	0	0	0	0	1	0	1	0	0	0	1	0	1	1	1	√	√	—
SBCD [HL+],A	[HL],A ← decimal adjustment {[HL] – A – C}, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	0	1	1	1	1	√	√	√
SBCD [XY+],A	[XY],A ← decimal adjustment {[XY] – A – C}, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	0	1	1	1	1	1	1	√	√	√
SBCD E:[HL+],A	E:[HL],A ← decimal adjustment {E:[HL] – A – C}, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	1	√	√	√
SBCD E:[XY+],A	E:[XY],A ← decimal adjustment {E:[XY] – A – C}, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	1	√	√	√
SBCJ \cur,n	cur,A ← n-ary adjustment {cur – C}	1	1	0	0	0	1	1	n_2	n_1	n_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
SBCJ [HL],n	[HL],A ← n-ary adjustment {[HL] – C}	1	1	0	0	0	0	0	1	1	0	0	0	1	0	1	n_2	n_1	n_0	√	√	—
SBCJ [XY],n	[XY],A ← n-ary adjustment {[XY] – C}	1	1	0	0	0	0	0	1	1	0	0	0	1	1	1	n_2	n_1	n_0	√	√	—
SBCJ E:[HL],n	E:[HL],A ← n-ary adjustment {E:[HL] – C}	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	n_2	n_1	n_0	√	√	—
SBCJ E:[XY],n	E:[XY],A ← n-ary adjustment {E:[XY] – C}	1	1	0	0	0	0	0	1	1	0	0	0	0	1	1	n_2	n_1	n_0	√	√	—
SBCJ [HL+],n	[HL],A ← n-ary adjustment {[HL] – C}, HL ← HL + 1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	1	n_2	n_1	n_0	√	√	√
SBCJ [XY+],n	[XY],A ← n-ary adjustment {[XY] – C}, XY ← XY + 1	1	1	0	0	0	0	0	1	1	1	0	0	1	1	1	n_2	n_1	n_0	√	√	√
SBCJ E:[HL+],n	E:[HL],A ← n-ary adjustment {E:[HL] – C}, HL ← HL + 1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	1	n_2	n_1	n_0	√	√	√
SBCJ E:[XY+],n	E:[XY],A ← n-ary adjustment {E:[XY] – C}, XY ← XY + 1	1	1	0	0	0	0	0	1	1	1	0	0	0	1	1	n_2	n_1	n_0	√	√	√

Compare Instructions

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
CMP sfr,A	sfr – A	1	1	0	0	1	0	1	0	1	0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
CMP \cur,A	cur – A	1	1	0	0	1	1	1	0	1	0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
CMP [HL],A	[HL] – A	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	√	√	—
CMP [XY],A	[XY] – A	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	√	√	—
CMP E:[HL],A	E:[HL] – A	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	√	√	—
CMP E:[XY],A	E:[XY] – A	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	√	√	—
CMP [HL+],A	[XY] – A, HL ← HL + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0	√	√	√
CMP [XY+],A	[XY] – A, XY ← XY + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	1	1	0	0	√	√	√
CMP E:[HL+],A	E:[HL] – A, HL ← HL + 1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	√	√	√
CMP E:[XY+],A	E:[XY] – A, XY ← XY + 1	1	1	0	0	0	0	0	1	0	0	0	0	0	1	1	1	0	0	√	√	√
CMP \cur,#i4	cur – i4	1	1	1	0	1	0	i_3	i_2	i_1	i_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	√	—
CMP [HL],#i4	[HL] – i4	1	1	0	0	0	0	0	1	1	0	1	0	1	0	i_3	i_2	i_1	i_0	√	√	—
CMP [XY],#i4	[XY] – i4	1	1	0	0	0	0	0	1	1	0	1	0	1	1	i_3	i_2	i_1	i_0	√	√	—
CMP E:[HL],#i4	E:[HL] – i4	1	1	0	0	0	0	0	1	1	0	1	0	0	0	i_3	i_2	i_1	i_0	√	√	—
CMP E:[XY],#i4	E:[XY] – i4	1	1	0	0	0	0	0	1	1	0	1	0	0	1	i_3	i_2	i_1	i_0	√	√	—
CMP [HL+],#i4	[HL] – i4, HL ← HL + 1	1	1	0	0	0	0	0	1	1	1	1	0	1	0	i_3	i_2	i_1	i_0	√	√	√
CMP [XY+],#i4	[XY] – i4, XY ← XY + 1	1	1	0	0	0	0	0	1	1	1	1	0	1	1	i_3	i_2	i_1	i_0	√	√	√
CMP E:[HL+],#i4	E:[HL] – i4, HL ← HL + 1	1	1	0	0	0	0	0	1	1	1	1	0	0	0	i_3	i_2	i_1	i_0	√	√	√
CMP E:[XY+],#i4	E:[XY] – i4, XY ← XY + 1	1	1	0	0	0	0	0	1	1	1	1	0	0	1	i_3	i_2	i_1	i_0	√	√	√

Logic Instructions

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
AND sfr,A	sfr,A ← sfr ∧ A	1	1	0	0	1	0	1	0	1	1	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
AND \cur,A	cur,A ← cur ∧ A	1	1	0	0	1	1	1	0	1	1	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
AND [HL],A	[HL],A ← [HL] ∧ A	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	√	—	—
AND [XY],A	[XY],A ← [XY] ∧ A	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	1	√	—	—
AND E:[HL],A	E:[HL],A ← E:[HL] ∧ A	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	√	—	—
AND E:[XY],A	E:[XY],A ← E:[XY] ∧ A	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	√	—	—
AND [HL+],A	[HL],A ← [HL] ∧ A, HL ← HL + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	√	—	√
AND [XY+],A	[XY],A ← [XY] ∧ A, XY ← XY + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	1	√	—	√
AND E:[HL+],A	E:[HL],A ← E:[HL] ∧ A, HL ← HL + 1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	√	—	√
AND E:[XY+],A	E:[XY],A ← E:[XY] ∧ A, XY ← XY + 1	1	1	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	1	√	—	√
AND \cur,#i4	cur,A ← cur ∧ i4	1	1	0	1	0	1	i_3	i_2	i_1	i_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
AND [HL],#i4	[HL],A ← [HL] ∧ i4	1	1	0	0	0	0	0	1	0	0	0	1	1	0	i_3	i_2	i_1	i_0	√	—	—
AND [XY],#i4	[XY],A ← [XY] ∧ i4	1	1	0	0	0	0	0	1	0	0	0	1	1	1	i_3	i_2	i_1	i_0	√	—	—
AND E:[HL],#i4	E:[HL],A ← E:[HL] ∧ i4	1	1	0	0	0	0	0	1	0	0	0	1	0	0	i_3	i_2	i_1	i_0	√	—	—
AND E:[XY],#i4	E:[XY],A ← E:[XY] ∧ i4	1	1	0	0	0	0	0	1	0	0	0	1	0	1	i_3	i_2	i_1	i_0	√	—	—
AND [HL+],#i4	[HL],A ← [HL] ∧ i4, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	1	1	0	i_3	i_2	i_1	i_0	√	—	√
AND [XY+],#i4	[XY],A ← [XY] ∧ i4, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	1	1	1	i_3	i_2	i_1	i_0	√	—	√
AND E:[HL+],#i4	E:[HL],A ← E:[HL] ∧ i4, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	1	0	0	i_3	i_2	i_1	i_0	√	—	√
AND E:[XY+],#i4	E:[XY],A ← E:[XY] ∧ i4, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	1	0	1	i_3	i_2	i_1	i_0	√	—	√
OR sfr,A	sfr,A ← sfr ∨ A	1	1	0	0	1	0	1	1	0	0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
OR \cur,A	cur,A ← cur ∨ A	1	1	0	0	1	1	1	1	0	0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
OR [HL],A	[HL],A ← [HL] ∨ A	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	√	—	—
OR [XY],A	[XY],A ← [XY] ∨ A	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	0	√	—	—
OR E:[HL],A	E:[HL],A ← E:[HL] ∨ A	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	√	—	—
OR E:[XY],A	E:[XY],A ← E:[XY] ∨ A	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	√	—	—
OR [HL+],A	[HL],A ← [HL] ∨ A, HL ← HL + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0	√	—	√
OR [XY+],A	[XY],A ← [XY] ∨ A, XY ← XY + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	1	0	1	0	√	—	√
OR E:[HL+],A	E:[HL],A ← E:[HL] ∨ A, HL ← HL + 1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	√	—	√
OR E:[XY+],A	E:[XY],A ← E:[XY] ∨ A, XY ← XY + 1	1	1	0	0	0	0	0	1	0	0	0	0	0	1	1	0	1	0	√	—	√

Logic Instructions (continued)

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
OR \cur,#i4	cur,A ← cur ∨ i4	1	1	0	1	1	0	i_3	i_2	i_1	i_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
OR [HL],#i4	[HL],A ← [HL] ∨ i4	1	1	0	0	0	0	0	0	1	0	0	1	1	0	i_3	i_2	i_1	i_0	√	—	—
OR [XY],#i4	[XY],A ← [XY] ∨ i4	1	1	0	0	0	0	0	0	1	0	0	1	1	1	i_3	i_2	i_1	i_0	√	—	—
OR E:[HL],#i4	E:[HL],A ← E:[HL] ∨ i4	1	1	0	0	0	0	0	0	1	0	0	1	0	0	i_3	i_2	i_1	i_0	√	—	—
OR E:[XY],#i4	E:[XY],A ← E:[XY] ∨ i4	1	1	0	0	0	0	0	0	1	0	0	1	0	1	i_3	i_2	i_1	i_0	√	—	—
OR [HL+],#i4	[HL],A ← [HL] ∨ i4, HL ← HL + 1	1	1	0	0	0	0	0	0	1	1	0	1	1	0	i_3	i_2	i_1	i_0	√	—	√
OR [XY+],#i4	[XY],A ← [XY] ∨ i4, XY ← XY + 1	1	1	0	0	0	0	0	0	1	1	0	1	1	1	i_3	i_2	i_1	i_0	√	—	√
OR E:[HL+],#i4	E:[HL],A ← E:[HL] ∨ i4, HL ← HL + 1	1	1	0	0	0	0	0	0	1	1	0	1	0	0	i_3	i_2	i_1	i_0	√	—	√
OR E:[XY+],#i4	E:[XY],A ← E:[XY] ∨ i4, XY ← XY + 1	1	1	0	0	0	0	0	0	1	1	0	1	0	1	i_3	i_2	i_1	i_0	√	—	√
XOR sfr,A	sfr,A ← sfr ∀ A	1	1	0	0	1	0	1	1	0	1	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
XOR \cur,A	cur,A ← cur ∀ A	1	1	0	0	1	1	1	1	0	1	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
XOR [HL],A	[HL],A ← [HL] ∀ A	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1	√	—	—
XOR [XY],A	[XY],A ← [XY] ∀ A	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	1	√	—	—
XOR E:[HL],A	E:[HL],A ← E:[HL] ∀ A	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	√	—	—
XOR E:[XY],A	E:[XY],A ← E:[XY] ∀ A	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	1	√	—	—
XOR [HL+],A	[HL],A ← [HL] ∀ A, HL ← HL + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	1	1	√	—	√
XOR [XY+],A	[XY],A ← [XY] ∀ A, XY ← XY + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	1	0	1	1	√	—	√
XOR E:[HL+],A	E:[HL],A ← E:[HL] ∀ A, HL ← HL + 1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	√	—	√
XOR E:[XY+],A	E:[XY],A ← E:[XY] ∀ A, XY ← XY + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	1	1	1	√	—	√
XOR \cur,#i4	cur,A ← cur ∀ i4	1	1	0	1	1	1	i_3	i_2	i_1	i_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
XOR [HL],#i4	[HL],A ← [HL] ∀ i4	1	1	0	0	0	0	0	0	0	0	0	1	1	0	i_3	i_2	i_1	i_0	√	—	—
XOR [XY],#i4	[XY],A ← [XY] ∀ i4	1	1	0	0	0	0	0	0	0	0	0	1	1	1	i_3	i_2	i_1	i_0	√	—	—
XOR E:[HL],#i4	E:[HL],A ← E:[HL] ∀ i4	1	1	0	0	0	0	0	0	0	0	0	1	0	0	i_3	i_2	i_1	i_0	√	—	—
XOR E:[XY],#i4	E:[XY],A ← E:[XY] ∀ i4	1	1	0	0	0	0	0	0	0	0	0	1	0	1	i_3	i_2	i_1	i_0	√	—	—
XOR [HL+],#i4	[HL],A ← [HL] ∀ i4, HL ← HL + 1	1	1	0	0	0	0	0	0	0	1	0	1	1	0	i_3	i_2	i_1	i_0	√	—	√
XOR [XY+],#i4	[XY],A ← [XY] ∀ i4, XY ← XY + 1	1	1	0	0	0	0	0	0	0	1	0	1	1	1	i_3	i_2	i_1	i_0	√	—	√
XOR E:[HL+],#i4	E:[HL],A ← E:[HL] ∀ i4, HL ← HL + 1	1	1	0	0	0	0	0	0	0	1	0	1	0	0	i_3	i_2	i_1	i_0	√	—	√
XOR E:[XY+],#i4	E:[XY],A ← E:[XY] ∀ i4, XY ← XY + 1	1	1	0	0	0	0	0	0	0	1	0	1	0	1	i_3	i_2	i_1	i_0	√	—	√

Mask Operation Instructions

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
MTST sfr,A	Testing of all bits in sfr not masked by A	1	1	0	0	1	0	1	1	1	1	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
MTST \cur,A	Testing of all bits in cur not masked by A	1	1	0	0	1	1	1	1	1	1	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
MTST [HL],A	Testing of all bits in [HL] not masked by A	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	√	—	—
MTST [XY],A	Testing of all bits in [XY] not masked by A	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	1	√	—	—
MTST E:[HL],A	Testing of all bits in E:[HL] not masked by A	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	√	—	—
MTST E:[XY],A	Testing of all bits in E:[XY] not masked by A	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	1	√	—	—
MTST [HL+],A	Testing of all bits in [HL] not masked by A, HL ← HL + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	√	—	√
MTST [XY+],A	Testing of all bits in [XY] not masked by A, XY ← XY + 1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	1	1	0	1	√	—	√
MTST E:[HL+],A	Testing of all bits in E:[HL] not masked by A, HL ← HL + 1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	1	√	—	√
MTST E:[XY+],A	Testing of all bits in E:[XY] not masked by A, XY← XY + 1	1	1	0	0	0	0	0	1	0	0	0	0	0	1	1	1	0	1	√	—	√
MTST \cur,#m	Testing of all bits in cur not masked by #m	1	1	1	0	1	1	m_3	m_2	m_1	m_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
MTST [HL],#m	Testing of all bits in [HL] not masked by #m	1	1	0	0	0	0	0	1	0	0	1	0	1	0	m_3	m_2	m_1	m_0	√	—	—
MTST [XY],#m	Testing of all bits in [XY] not masked by #m	1	1	0	0	0	0	0	1	0	0	1	0	1	1	m_3	m_2	m_1	m_0	√	—	—
MTST E:[HL],#m	Testing of all bits in E:[HL] not masked by #m	1	1	0	0	0	0	0	1	0	0	1	0	0	0	m_3	m_2	m_1	m_0	√	—	—
MTST E:[XY],#m	Testing of all bits in E:[XY] not masked by #m	1	1	0	0	0	0	0	1	0	0	1	0	0	1	m_3	m_2	m_1	m_0	√	—	—
MTST [HL+],#m	Testing of all bits in [HL] not masked by #m, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	1	0	1	0	m_3	m_2	m_1	m_0	√	—	√
MTST [XY+],#m	Testing of all bits in [XY] not masked by #m, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	1	0	1	1	m_3	m_2	m_1	m_0	√	—	√
MTST E:[HL+],#m	Testing of all bits in E:[HL] not masked by #m, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	1	0	0	0	m_3	m_2	m_1	m_0	√	—	√
MTST E:[XY+],#m	Testing of all bits in E:[XY] not masked by #m, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	1	0	0	1	m_3	m_2	m_1	m_0	√	—	√

Mask Operation Instructions (continued)

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
MCLR \cur,#m	Clearing of all bits in cur not masked by #m, A ← cur	1	1	0	1	0	1	m_3	m_2	m_1	m_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
MCLR [HL],#m	Clearing of all bits in [HL] not masked by #m, A ← [HL]	1	1	0	0	0	0	0	1	0	0	0	1	1	0	m_3	m_2	m_1	m_0	√	—	—
MCLR [XY],#m	Clearing of all bits in [XY] not masked by #m, A ← [XY]	1	1	0	0	0	0	0	1	0	0	0	1	1	1	m_3	m_2	m_1	m_0	√	—	—
MCLR E:[HL],#m	Clearing of all bits in E:[HL] not masked by #m, A ← E:[HL]	1	1	0	0	0	0	0	1	0	0	0	1	0	0	m_3	m_2	m_1	m_0	√	—	—
MCLR E:[XY],#m	Clearing of all bits in E:[XY] not masked by #m, A ← E:[XY]	1	1	0	0	0	0	0	1	0	0	0	1	0	1	m_3	m_2	m_1	m_0	√	—	—
MCLR [HL+],#m	Clearing of all bits in [HL] not masked by #m, A ← [HL], HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	1	1	0	m_3	m_2	m_1	m_0	√	—	√
MCLR [XY+],#m	Clearing of all bits in [XY] not masked by #m, A ← [XY], XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	1	1	1	m_3	m_2	m_1	m_0	√	—	√
MCLR E:[HL+],#m	Clearing of all bits in E:[HL] not masked by #m, A ← E:[HL], HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	1	0	0	m_3	m_2	m_1	m_0	√	—	√
MCLR E:[XY+],#m	Clearing of all bits in E:[XY] not masked by #m, A ← E:[XY], XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	1	0	1	m_3	m_2	m_1	m_0	√	—	√

Mask Operation Instructions (continued)

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
MSET \cur,#m	Setting of all bits in cur not masked by #m, A ← cur	1	1	0	1	1	0	m_3	m_2	m_1	m_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
MSET [HL],#m	Setting of all bits in [HL] not masked by #m, A ← [HL]	1	1	0	0	0	0	0	0	1	0	0	1	1	0	m_3	m_2	m_1	m_0	√	—	—
MSET [XY],#m	Setting of all bits in [XY] not masked by #m, A ← [XY]	1	1	0	0	0	0	0	0	1	0	0	1	1	1	m_3	m_2	m_1	m_0	√	—	—
MSET E:[HL],#m	Setting of all bits in E:[HL] not masked by #m, A ← E:[HL]	1	1	0	0	0	0	0	0	1	0	0	1	0	0	m_3	m_2	m_1	m_0	√	—	—
MSET E:[XY],#m	Setting of all bits in E:[XY] not masked by #m, A ← E:[XY]	1	1	0	0	0	0	0	0	1	0	0	1	0	1	m_3	m_2	m_1	m_0	√	—	—
MSET [HL+],#m	Setting of all bits in [HL] not masked by #m, A ← [HL], HL ← HL + 1	1	1	0	0	0	0	0	0	1	1	0	1	1	0	m_3	m_2	m_1	m_0	√	—	√
MSET [XY+],#m	Setting of all bits in [XY] not masked by #m, A ← [XY], XY ← XY + 1	1	1	0	0	0	0	0	0	1	1	0	1	1	1	m_3	m_2	m_1	m_0	√	—	√
MSET E:[HL+],#m	Setting of all bits in E:[HL] not masked by #m, A ← E:[HL], HL ← HL + 1	1	1	0	0	0	0	0	0	1	1	0	1	0	0	m_3	m_2	m_1	m_0	√	—	√
MSET E:[XY+],#m	Setting of all bits in E:[XY] not masked by #m, A ← E:[XY], XY ← XY + 1	1	1	0	0	0	0	0	0	1	1	0	1	0	1	m_3	m_2	m_1	m_0	√	—	√

Mask Operation Instructions (continued)

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
MNOT \cur,#m	Inverting of all bits in cur not masked by #m, A ← cur	1	1	0	1	1	1	m_3	m_2	m_1	m_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
MNOT [HL],#m	Inverting of all bits in [HL] not masked by #m, A ← [HL]	1	1	0	0	0	0	0	0	0	0	0	1	1	0	m_3	m_2	m_1	m_0	√	—	—
MNOT [XY],#m	Inverting of all bits in [XY] not masked by #m, A ← [XY]	1	1	0	0	0	0	0	0	0	0	0	1	1	1	m_3	m_2	m_1	m_0	√	—	—
MNOT E:[HL],#m	Inverting of all bits in E:[HL] not masked by #m, A ← E:[HL]	1	1	0	0	0	0	0	0	0	0	0	1	0	0	m_3	m_2	m_1	m_0	√	—	—
MNOT E:[XY],#m	Inverting of all bits in E:[XY] not masked by #m, A ← E:[XY]	1	1	0	0	0	0	0	0	0	0	0	1	0	1	m_3	m_2	m_1	m_0	√	—	—
MNOT [HL+],#m	Inverting of all bits in [HL] not masked by #m, A ← [HL], HL ← HL + 1	1	1	0	0	0	0	0	0	0	1	0	1	1	0	m_3	m_2	m_1	m_0	√	—	√
MNOT [XY+],#m	Inverting of all bits in [XY] not masked by #m, A ← [XY], XY ← XY + 1	1	1	0	0	0	0	0	0	0	1	0	1	1	1	m_3	m_2	m_1	m_0	√	—	√
MNOT E:[HL+],#m	Inverting of all bits in E:[HL] not masked by #m, A ← E:[HL], HL ← HL + 1	1	1	0	0	0	0	0	0	0	1	0	1	0	0	m_3	m_2	m_1	m_0	√	—	√
MNOT E:[XY+],#m	Inverting of all bits in E:[XY] not masked by #m, A ← E:[XY], XY ← XY + 1	1	1	0	0	0	0	0	0	0	1	0	1	0	1	m_3	m_2	m_1	m_0	√	—	√

Bit Operation Instructions

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
BTST \cur.n	Bit testing of cur.n	1	1	1	0	1	1	n_3	n_2	n_1	n_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
BTST [HL].n	Bit testing of [HL].n	1	1	0	0	0	0	0	1	0	0	1	0	1	0	n_3	n_2	n_1	n_0	√	—	—
BTST [XY].n	Bit testing of [XY].n	1	1	0	0	0	0	0	1	0	0	1	0	1	1	n_3	n_2	n_1	n_0	√	—	—
BTST E:[HL].n	Bit testing of E:[HL].n	1	1	0	0	0	0	0	1	0	0	1	0	0	0	n_3	n_2	n_1	n_0	√	—	—
BTST E:[XY].n	Bit testing of E:[XY].n	1	1	0	0	0	0	0	1	0	0	1	0	0	1	n_3	n_2	n_1	n_0	√	—	—
BTST [HL+].n	Bit testing of [HL].n, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	1	0	1	0	n_3	n_2	n_1	n_0	√	—	√
BTST [XY+].n	Bit testing of [XY].n, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	1	0	1	1	n_3	n_2	n_1	n_0	√	—	√
BTST E:[HL+].n	Bit testing of E:[HL].n, HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	1	0	0	0	n_3	n_2	n_1	n_0	√	—	√
BTST E:[XY+].n	Bit testing of E:[XY].n, XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	1	0	0	1	n_3	n_2	n_1	n_0	√	—	√
BCLR \cur.n	cur.n ← 0, A ← cur	1	1	0	1	0	1	n_3	n_2	n_1	n_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
BCLR [HL].n	[HL].n ← 0, A ← [HL]	1	1	0	0	0	0	0	1	0	0	0	1	1	0	n_3	n_2	n_1	n_0	√	—	—
BCLR [XY].n	[XY].n ← 0, A ← [XY]	1	1	0	0	0	0	0	1	0	0	0	1	1	1	n_3	n_2	n_1	n_0	√	—	—
BCLR E:[HL].n	E:[HL].n ← 0, A ← E:[HL]	1	1	0	0	0	0	0	1	0	0	0	1	0	0	n_3	n_2	n_1	n_0	√	—	—
BCLR E:[XY].n	E:[XY].n ← 0, A ← E:[XY]	1	1	0	0	0	0	0	1	0	0	0	1	0	1	n_3	n_2	n_1	n_0	√	—	—
BCLR [HL+].n	[HL].n ← 0, A ← [HL], HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	1	1	0	n_3	n_2	n_1	n_0	√	—	√
BCLR [XY+].n	[XY].n ← 0, A ← [XY], XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	1	1	1	n_3	n_2	n_1	n_0	√	—	√
BCLR E:[HL+].n	E:[HL].n ← 0, A ← E:[HL], HL ← HL + 1	1	1	0	0	0	0	0	1	0	1	0	1	0	0	n_3	n_2	n_1	n_0	√	—	√
BCLR E:[XY+].n	E:[XY].n ← 0, A ← E:[XY], XY ← XY + 1	1	1	0	0	0	0	0	1	0	1	0	1	0	1	n_3	n_2	n_1	n_0	√	—	√
BSET \cur.n	cur.n ← 1, A ← cur	1	1	0	1	1	0	n_3	n_2	n_1	n_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
BSET [HL].n	[HL].n ← 1, A ← [HL]	1	1	0	0	0	0	0	0	1	0	0	1	1	0	n_3	n_2	n_1	n_0	√	—	—
BSET [XY].n	[XY].n ← 1, A ← [XY]	1	1	0	0	0	0	0	0	1	0	0	1	1	1	n_3	n_2	n_1	n_0	√	—	—
BSET E:[HL].n	E:[HL].n ← 1, A ← E:[HL]	1	1	0	0	0	0	0	0	1	0	0	1	0	0	n_3	n_2	n_1	n_0	√	—	—
BSET E:[XY].n	E:[XY].n ← 1, A ← E:[XY]	1	1	0	0	0	0	0	0	1	0	0	1	0	1	n_3	n_2	n_1	n_0	√	—	—
BSET [HL+].n	[HL].n ← 1, A ← [HL], HL ← HL + 1	1	1	0	0	0	0	0	0	1	1	0	1	1	0	n_3	n_2	n_1	n_0	√	—	√
BSET [XY+].n	[XY].n ← 1, A ← [XY], XY ← XY + 1	1	1	0	0	0	0	0	0	1	1	0	1	1	1	n_3	n_2	n_1	n_0	√	—	√
BSET E:[HL+].n	E:[HL].n ← 1, A ← E:[HL], HL ← HL + 1	1	1	0	0	0	0	0	0	1	1	0	1	0	0	n_3	n_2	n_1	n_0	√	—	√
BSET E:[XY+].n	E:[XY].n ← 1, A ← E:[XY], XY ← XY + 1	1	1	0	0	0	0	0	0	1	1	0	1	0	1	n_3	n_2	n_1	n_0	√	—	√

Bit Operation Instructions (continued)

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
BNOT \cur.n	cur.n ← $\overline{\text{cur.n}}$, A ← cur	1	1	0	1	1	1	n_3	n_2	n_1	n_0	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	√	—	—
BNOT [HL].n	[HL].n ← $\overline{\text{[HL].n}}$, A ← [HL]	1	1	0	0	0	0	0	0	0	0	0	1	1	0	n_3	n_2	n_1	n_0	√	—	—
BNOT [XY].n	[XY].n ← $\overline{\text{[XY].n}}$, A ← [XY]	1	1	0	0	0	0	0	0	0	0	0	1	1	1	n_3	n_2	n_1	n_0	√	—	—
BNOT E:[HL].n	E:[HL].n ← $\overline{\text{E:[HL].n}}$, A ← E:[HL]	1	1	0	0	0	0	0	0	0	0	0	1	0	0	n_3	n_2	n_1	n_0	√	—	—
BNOT E:[XY].n	E:[XY].n ← $\overline{\text{E:[XY].n}}$, A ← E:[XY]	1	1	0	0	0	0	0	0	0	0	0	1	0	1	n_3	n_2	n_1	n_0	√	—	—
BNOT [HL+].n	[HL].n ← $\overline{\text{[HL].n}}$, A ← [HL], HL ← HL + 1	1	1	0	0	0	0	0	0	0	1	0	1	1	0	n_3	n_2	n_1	n_0	√	—	√
BNOT [XY+].n	[XY].n ← $\overline{\text{[XY].n}}$, A ← [XY], XY ← XY + 1	1	1	0	0	0	0	0	0	0	1	0	1	1	1	n_3	n_2	n_1	n_0	√	—	√
BNOT E:[HL+].n	E:[HL].n ← $\overline{\text{E:[HL].n}}$, A ←E:[HL], HL ← HL + 1	1	1	0	0	0	0	0	0	0	1	0	1	0	0	n_3	n_2	n_1	n_0	√	—	√
BNOT E:[XY+].n	E:[XY].n ← $\overline{\text{E:[XY].n}}$, A ← E:[XY], XY ← XY + 1	1	1	0	0	0	0	0	0	0	1	0	1	0	1	n_3	n_2	n_1	n_0	√	—	√

ROM Table Reference Instructions

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
MOVHB [HL],[RA]	[HL],[HL + 1] ← $(RA)_{15-8}$	1	2	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	—	—	—
MOVHB [XY],[RA]	[XY],[XY + 1] ← $(RA)_{15-8}$	1	2	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	0	—	—	—
MOVHB E:[HL],[RA]	E:[HL],E:[HL + 1] ← $(RA)_{15-8}$	1	2	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	—	—	—
MOVHB E:[XY],[RA]	E:[XY],E:[XY + 1] ← $(RA)_{15-8}$	1	2	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1	0	—	—	—
MOVHB [HL+],[RA]	[HL],[HL + 1] ← $(RA)_{15-8}$, HL ← HL + 2	1	2	0	0	0	0	0	0	1	1	0	0	1	0	1	0	1	0	—	—	√
MOVHB [XY+],[RA]	[XY],[XY + 1] ← $(RA)_{15-8}$, XY ← XY + 2	1	2	0	0	0	0	0	0	1	1	0	0	1	1	1	0	1	0	—	—	√
MOVHB E:[HL+],[RA]	E:[HL],E:[HL + 1] ← $(RA)_{15-8}$, HL ← HL + 2	1	2	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	—	—	√
MOVHB E:[XY+],[RA]	E:[XY],E:[XY + 1] ← $(RA)_{15-8}$, XY ← XY + 2	1	2	0	0	0	0	0	0	1	1	0	0	0	1	1	0	1	0	—	—	√
MOVHB [HL],cadr16	[HL],[HL + 1] ← $(cadr16)_{15-8}$	2	3	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0	—	—	—
				a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOVHB [XY],cadr16	[XY],[XY + 1] ← $(cadr16)_{15-8}$	2	3	0	0	0	0	0	0	1	1	0	0	1	1	0	1	0	0	—	—	—
				a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOVHB E:[HL],cadr16	E:[HL],E:[HL + 1] ← $(cadr16)_{15-8}$	2	3	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	—	—	—
				a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOVHB E:[XY],cadr16	E:[XY],E:[XY + 1] ← $(cadr16)_{15-8}$	2	3	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	—	—	—
				a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOVHB [HL+],cadr16	[HL],[HL + 1] ← $(cadr16)_{15-8}$, HL ← HL + 2	2	3	0	0	0	0	0	0	1	1	0	0	1	0	1	1	0	0	—	—	√
				a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOVHB [XY+],cadr16	[XY],[XY + 1] ← $(cadr16)_{15-8}$, XY ← XY + 2	2	3	0	0	0	0	0	0	1	1	0	0	1	1	1	1	0	0	—	—	√
				a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOVHB E:[HL+],cadr16	E:[HL],E:[HL + 1] ← $(cadr16)_{15-8}$, HL ← HL + 2	2	3	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	—	—	√
				a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOVHB E:[XY+],cadr16	E:[XY],E:[XY + 1] ← $(cadr16)_{15-8}$, XY ← XY + 2	2	3	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	—	—	√
				a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			

ROM Table Reference Instructions (continued)

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
MOVLB [HL],[RA]	[HL],[HL + 1] ← $(RA)_{7-0}$	1	2	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	1	—	—	—
MOVLB [XY],[RA]	[XY],[XY + 1] ← $(RA)_{7-0}$	1	2	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	1	—	—	—
MOVLB E:[HL],[RA]	E:[HL],E:[HL + 1] ← $(RA)_{7-0}$	1	2	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	—	—	—
MOVLB E:[XY],[RA]	E:[XY],E:[XY + 1] ← $(RA)_{7-0}$	1	2	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1	1	—	—	—
MOVLB [HL+],[RA]	[HL],[HL + 1] ← $(RA)_{7-0}$, HL ← HL + 2	1	2	0	0	0	0	0	0	1	1	0	0	1	0	1	0	1	1	—	—	√
MOVLB [XY+],[RA]	[XY],[XY + 1] ← $(RA)_{7-0}$, XY ← XY + 2	1	2	0	0	0	0	0	0	1	1	0	0	1	1	1	0	1	1	—	—	√
MOVLB E:[HL+],[RA]	E:[HL],E:[HL + 1] ← $(RA)_{7-0}$, HL ← HL + 2	1	2	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	1	—	—	√
MOVLB E:[XY+],[RA]	E:[XY],E:[XY + 1] ← $(RA)_{7-0}$, XY ← XY + 2	1	2	0	0	0	0	0	0	1	1	0	0	0	1	1	0	1	1	—	—	√
MOVLB [HL],cadr16	[HL],[HL + 1] ← $(cadr16)_{7-0}$	2	3	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0	1	—	—	—
				a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOVLB [XY],cadr16	[XY],[XY + 1] ← $(cadr16)_{7-0}$	2	3	0	0	0	0	0	0	1	1	0	0	1	1	0	1	0	1	—	—	—
				a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOVLB E:[HL],cadr16	E:[HL],E:[HL + 1] ← $(cadr16)_{7-0}$	2	3	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	1	—	—	—
				a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOVLB E:[XY],cadr16	E:[XY],E:[XY + 1] ← $(cadr16)_{7-0}$	2	3	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	1	—	—	—
				a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOVLB [HL+],cadr16	[HL],[HL + 1] ← $(cadr16)_{7-0}$, HL ← HL + 2	2	3	0	0	0	0	0	0	1	1	0	0	1	0	1	1	0	1	—	—	√
				a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOVLB [XY+],cadr16	[XY],[XY + 1] ← $(cadr16)_{7-0}$, XY ← XY + 2	2	3	0	0	0	0	0	0	1	1	0	0	1	1	1	1	0	1	—	—	√
				a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOVLB E:[HL+],cadr16	E:[HL],E:[HL + 1] ← $(cadr16)_{7-0}$, HL ← HL + 2	2	3	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1	—	—	√
				a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOVLB E:[XY+],cadr16	E:[XY],E:[XY + 1] ← $(cadr16)_{7-0}$, XY ← XY + 2	2	3	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	1	—	—	√
				a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			

Stack Operation Instructions

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
PUSH HL	(RSP) ← {FLAG, A, HL}, RSP ← RSP + 1	1	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	—	—	—
PUSH XY	(RSP) ← {CBR, EBR, XY}, RSP ← RSP + 1	1	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	—	—	—
POP HL	RSP ← RSP – 1, {FLAG, A, HL} ← (RSP)	1	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	√	√	√
POP XY	RSP ← RSP – 1, {CBR, EBR, XY} ← (RSP)	1	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	—	—	—

Flag Operation Instructions

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
FCLR G	G ← 0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	—	—	√
FCLR C	C ← 0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	—	√	—
FCLR Z	Z ← 0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	√	—	—
FSET G	G ← 1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	—	—	√
FSET C	C ← 1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	—	√	—
FSET Z	Z ← 1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	√	—	—

Jump Instructions

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
LJMP cadr15	PC ← cadr15	2	2	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	—	—	—
				0	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
JMP cadr12	$PC_{11–0}$ ← cadr12	1	1	1	1	1	0	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	—	—	—
SJMP radr8	PC ← Next PC + radr8	1	1	0	0	0	0	1	0	0	a_7	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	—	—	—
JMP PC + A	PC ← PC + A + 1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	—	—	—

Conditional Branch Instructions

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
BC radr8 BLT radr8	if C = 1 then PC ← Next PC + radr8 (<)	1	1	0	0	0	0	1	0	1	a_7	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	—	—	—
BNC radr8 BGE radr8	if C = 0 then PC ← Next PC + radr8 (≥)	1	1	0	0	0	0	1	0	1	a_7	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	—	—	—
BZ radr8 BEQ radr8	If Z = 1 then PC ← Next PC + radr8 (=)	1	1	0	0	0	0	1	1	0	a_7	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	—	—	—
BNZ radr8 BNE radr8	If Z = 0 then PC ← Next PC + radr8 (≠)	1	1	0	0	0	0	1	1	0	a_7	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	—	—	—
BLE radr8	If (C = 1) ∨ (Z = 1) then PC ← Next PC + radr8 (≤)	1	1	0	0	0	0	1	1	1	a_7	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	—	—	—
BGT radr8	If (C = 0) ∧ (Z =0) then PC ← Next PC + radr8 (>)	1	1	0	0	0	0	1	1	1	a_7	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	—	—	—
BNG radr8	if G = 0 then PC ← Next PC + radr8	1	1	0	0	0	0	1	0	0	a_7	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	—	—	—

Call/Return Instructions

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
LCAL cadr15	(SP) ← PC, PC ← cadr15, SP ← SP + 1	2	2	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	—	—	—
				0	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
CAL cadr12	(SP) ← PC, $PC_{11–0}$ ← cadr12, SP ← SP + 1	1	1	1	1	1	1	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	—	—	—
RT	PC ← (SP) + 1, SP ← SP – 1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	—	—	—
RTI	PC ← (SP) + 1, SP ← SP – 1, MIE ← 1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	—	—	—
RTNMI	PC ← (SP) + 1, SP ← SP – 1 MIE ← status of MIE before an interrupt occurs	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	—	—	—

Control Instructions

MNEMONIC	OPERATION	W	C	INSTRUCTION CODE																FLAG		
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Z	C	G
NOP	NO OPERATION	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—	—	—
HALT	HALT CPU	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	—	—	—
EI	MIE ← 1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	—	—	—
DI	MIE ← 0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	—	—	—
INCB HL	HL ← HL + 1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	—	—	√
INCB XY	XY ← XY + 1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	—	—	√
INCW RA	RA ← RA + 1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	—	—	√
MOV CBR,#i4	CBR ← i4	1	1	0	0	0	0	0	0	0	0	0	0	1	1	i_3	i_2	i_1	i_0	—	—	—
MOV EBR,#i4	EBR ← i4	1	1	0	0	0	0	0	0	0	0	0	0	1	0	i_3	i_2	i_1	i_0	—	—	—
MOV RA0,#i4	RA0 ← i4	1	1	0	0	0	0	0	0	1	0	0	0	0	0	i_3	i_2	i_1	i_0	—	—	—
MOV RA1,#i4	RA1 ← i4	1	1	0	0	0	0	0	0	1	0	0	0	0	1	i_3	i_2	i_1	i_0	—	—	—
MOV RA2,#i4	RA2 ← i4	1	1	0	0	0	0	0	0	1	0	0	0	1	0	i_3	i_2	i_1	i_0	—	—	—
MOV RA3,#i4	RA3 ← i4	1	1	0	0	0	0	0	0	1	0	0	0	1	1	i_3	i_2	i_1	i_0	—	—	—
MOV H,#i4	H ← i4	1	1	0	0	0	0	0	0	0	1	0	0	1	1	i_3	i_2	i_1	i_0	—	—	—
MOV L,#i4	L ← i4	1	1	0	0	0	0	0	0	0	1	0	0	1	0	i_3	i_2	i_1	i_0	—	—	—
MOV X,#i4	X ← i4	1	1	0	0	0	0	0	0	0	1	0	0	0	1	i_3	i_2	i_1	i_0	—	—	—
MOV Y,#i4	Y ← i4	1	1	0	0	0	0	0	0	0	1	0	0	0	0	i_3	i_2	i_1	i_0	—	—	—
MSA cadr15	Melody output starts	2	3	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	—	—	—
				0	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			

ML63611

User's Manual

First Edition: May 2001
Second Edition: June 2001

PEUL63611-02