# Old Company Name in Catalogs and Other Documents

On April 1<sup>st</sup>, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1<sup>st</sup>, 2010 Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

#### Notice

- 1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
- Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
- 3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
- 4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
- 5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
- 6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
- 7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
  - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
  - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anticrime systems; safety equipment; and medical equipment not specifically designed for life support.
  - "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
- 8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
- 9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
- 10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
- 12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.
- (Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majorityowned subsidiaries.
- (Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

# RENESAS

# mos integrated circuit $\mu$ PD17933, 17934

# 4-BIT SINGLE-CHIP MICROCONTROLLERS WITH DIGITAL TUNING SYSTEM HARDWARE

The  $\mu$ PD17933 and 17934 are 4-bit single-chip CMOS microcontrollers with hardware for digital tuning systems (DTSs).

These microcontrollers integrate a prescaler that operates at a voltage as low as 1.05 V and up to 220 MHz, a PLL frequency synthesizer, an intermediate frequency (IF) counter, and an LCD controller/driver on a single chip.

Therefore, a high-performance digital tuning system for a portable set can be organized with a single chip.

#### **FEATURES**

\*

- General-purpose data memory (RAM) )  $448 \times 4$  bits
- Instruction execution time
   53.3 μs (with 75-kHz crystal resonator)
- PLL frequency synthesizer Dual modulus prescaler (220 MHz MAX.), programmable divider, phase comparator, charge pump
- Peripheral hardware General-purpose I/O ports, LCD controller/driver, serial interface, A/D converter, BEEP output, frequency counter
- Interrupt
   External: 1 source
   Internal: 3 sources
- Reset by RESET pin
- Low power consumption
- Supply voltage: VDD = 1.05 to 1.8 V

#### **ORDERING INFORMATION**

Part Number	Package
μPD17933GK-×××-BE9	80-pin plastic TQFP (12 $ imes$ 12 mm, 0.5 mm pitch)
$\mu$ PD17934GK- $\times$ ×-BE9	80-pin plastic TQFP (12 $ imes$ 12 mm, 0.5 mm pitch)

**Remark** ××× indicates ROM code suffix.

Unless otherwise specified, the  $\mu$ PD17934 is explained as the representative model in this document.

The information in this document is subject to change without notice.

# FUNCTIONAL OUTLINE

	Part Number		μPD17933	μPD17934			
	Program memory (ROM)		12K bytes (6144 x 16 bits) 16K bytes (8192 x 16 bits)				
	General-purpose data memory (RAM)		448 × 4 bits				
	Instruction execution time		53.3 $\mu$ s (with 75-MHz crystal oscillator)				
	General-purpo	ose port	37 pins • I/O port : 20 pins • Input port : 11 pins (of which 3 are muxed with LCD segment pins) • Output port : 6 pins				
	Stack level		<ul> <li>Address stack : 15 levels (stack can b</li> <li>Interrupt stack: 4 levels (stack can be</li> </ul>	e manipulated) manipulated)			
	Vector interru (maskable inte	ot errupt)	<ul> <li>External : 1 source (INT)</li> <li>Internal : 3 sources (basic timer 0, 8-</li> </ul>	bit timer, serial interface)			
	Timer		3 channels • Basic timer 0 (125 ms) • Basic timer 1 (8 ms, 32 ms) • 8-bit timer (with event counter)				
	A/D converter		8-bits resolution $\times$ 3 channels				
	LCD controller/driver		<ul> <li>20 segments, 4 commons</li> <li>1/4 duty, 1/2 bias, frame frequency: 62.5 Hz, drive voltage VLCD1=3.0 V TYP.</li> </ul>				
			Muxed segment pins. 3 (Each can be used as general-purpose input port pin.)				
	Serial interface		1 channel (3-wire/2-wire modes selectable)				
	PLL frequency synthesizer	Division mode	2 types { • Direct division mode (VCOL pin) • Pulse swallow mode (VCOL pin/VCOH pin)				
		Reference frequency	6 types selectable (1, 3, 5, 6.25, 12.5, 25 kHz)				
		Charge pump	Error out output: 2 pins (EO0 and EO1 p	ins)			
		Phase comparator	Unlock detectable by program				
*	Intermediate f counter	requency (IF)	Frequency measurement { • AMIFC pin: 400 to 500 kHz • FMIFC pin: 10 to 11 MHz				
	BEEP output		1 pin (1.5 kHz, 3 kHz)				
	Reset		Reset by RESET pin				
			<ul> <li>Watchdog timer reset</li> <li>Can be set only once on power application: 4096 or 8192 instructions selectable</li> </ul>				
			<ul> <li>Stack pointger overflow/underflow reset</li> <li>Can be set only once on power application: Interrupt stack and address stack selectable</li> </ul>				
*	Supply voltage	9	VDD = 1.05 to 1.8 V				
	Package		80-pin plastic TQFP (12 $\times$ 12 mm, 0.5 mm pitch)				

# NEC

#### **PIN CONFIGURATION (Top View)**

80-pin plastic TQFP (12  $\times$  12 mm, 0.5 mm pitch)  $\mu$ PD17933GK- $\times$ -BE9  $\mu$ PD17934GK- $\times$ -BE9



# PIN NAME

	AD0-AD2	: A/D converter inputs	P0D0-P0D3	: Port 0D
	AMIFC	: Intermediate frequency counter input	P1A0-P1A3	: Port 1A
		for AM	P1C0-P1C3	: Port 1C
	BEEP	: BEEP output	P1D0-P1D3	: Port 1D
	CAPLCD0-CAPLCD3	: Capacitor connection for LCD drive	P2A0-P2A2	: Port 2A
		voltage	P2B0-P2B3	: Port 2B
	COM0-COM3	: LCD common output	P2C0-P2C3	: Port 2C
	EO0, EO1	: Error output	REGLCD0-REGLCD2	: Regulator output for LCD drive
	FMIFC	: Intermediate frequency counter input	RESET	: Reset input
		for FM	SCK	: 3-wire serial clock I/O
	GND	: Ground	SI1	: 3-wire serial data input
	INT	: External interrupt input	SO1, SO2	: 3-wire serial data outputs
	LCD0-LCD19	: LCD segment outputs	TEST	: Test input
$\star$	NC	: No connection	TM0, TM1	: Timer event inputs
	P0A0, P0A1	: Port 0A	VCOH, VCOL	: Local oscillation inputs for PLL
	P0B0-P0B3	: Port 0B	Vdd0-Vdd2	: Power supply
	P0C0-P0C3	: Port 0C	Xin, Xout	: Crystal resonator connection

#### **BLOCK DIAGRAM**



## CONTENTS

1.	PIN F	UNCTIONS	11
	1.1	Pin Function List	11
	1.2	Equivalent Circuits of Pins	14
	1.3	Recommended Connections of Unused Pins	18
	1.4	Cautions on Using TEST Pin	19
2.	PRO	GRAM MEMORY (ROM)	20
	2.1	Outline of Program Memory	20
	2.2	Program Memory	21
	2.3	Program Counter	22
	2.4	Flow of Program	22
	2.5	Cautions on Using Program Memory	25
3.	ADD	RESS STACK (ASK)	26
	3.1	Outline of Address Stack	26
	3.2	Address Stack Register (ASR)	26
	3.3	Stack Pointer (SP)	28
	3.4	Operation of Address Stack	29
	3.5	Cautions on Using Address Stack	30
4.	DATA	A MEMORY (RAM)	31
	4.1	Outline of Data Memory	31
	4.2	Configuration and Function of Data Memory	33
	4.3	Data Memory Addressing	35
	4.4	Cautions on Using Data Memory	36
5.	SYST	EM REGISTERS (SYSREG)	37
	5.1	Outline of System Registers	37
	5.2	System Register List	38
	5.3	Address Register (AR)	39
	5.4	Window Register (WR)	41
	5.5	Bank Register (BANK)	42
	5.6	Index Register (IX) and Data Memory Row Address Pointer	
		(MP: memory pointer)	43
	5.7	General Register Pointer (RP)	45
	5.8	Program Status Word (PSWORD)	47
6.	GEN	ERAL REGISTER (GR)	49
	6.1	Outline of General Register	49
	6.2	General Register	49
	6.3	Generating Address of General Register by Each Instruction	50
	6.4	Cautions on Using General Register	50

# NEC

7.	ALU	.U (Arithmetic Logic Unit) BLOCK				
	7.1	Outline of ALU Block	51			
	7.2	Configuration and Function of Each Block	52			
	7.3	ALU Processing Instruction List	52			
	7.4	Cautions on Using ALU	56			
8.	REGI	STER FILE (RF) AND CONTROL REGISTERS	57			
	8.1	Outline of Register File	57			
	8.2	Configuration and Function of Register File	58			
	8.3	Control Registers and Input/Output Selection Registers	59			
	8.4	LCD Segment Registers	69			
	8.5	autions on Using Control Register	69			
•						
9.	DATA		70			
	9.1	Outline of Data Buffer	70			
	9.2	Data Buffer	71			
	9.3	Relationships between Peripheral Hardware and Data Buffer	72			
	9.4	Cautions on Using Data Buffer	76			
10.	DATA	BUFFER STACK	77			
	10.1	Outline of Data Buffer Stack	77			
	10.2	Data Buffer Stack Register	77			
	10.3	Data Buffer Stack Pointer	79			
	10.4	Operation of Data Buffer Stack	80			
	10.5	Using Data Buffer Stack	81			
	10.6	Cautions on Using Data Buffer Stack	81			
	10.0		01			
11.	GENE	ERAL-PURPOSE PORT	82			
	11.1	Outline of General-purpose Port	82			
	11.2	General-Purpose I/O Port (P0B, P1A, P1D, P2B, P2C)	84			
	11.3	General-Purpose Input Port (P0D, P1C, P2A)	93			
	11.4	General-Purpose Output Port (P0A, P0C)	96			
12.	INTE	RRUPT	98			
	12.1	Outline of Interrupt Block	98			
	12.2	Interrupt Control Block	100			
	12.3	Interrupt Stack Register	106			
	12.4	Stack Pointer, Address Stack Registers, and Program Counter	110			
	12.5	Interrupt Enable Flip-Flop (INTE)	110			
	12.6	Accepting Interrupt	111			
	12.7	Operations after Interrupt Has Been Accepted	116			
	12.8	Returning from Interrupt Routine	116			
	12.9	External Interrupts (INT pin)	117			
	12.10	Internal Interrupts	119			

13. TIMERS 12					
	13.1	Outline of Timers	120		
	13.2	Basic Timer 0	121		
	13.3	Basic Timer 1	125		
	13.4	Timer 0	131		
14.	A/D C	ONVERTER	138		
	14.1	Outline of A/D Converter	138		
	14.2	Input Selection Block	139		
	14.3	Compare Voltage Generation and Compare Blocks	141		
	14.4	Comparison Timing Chart	143		
	14.5	Using A/D Converter	144		
	14.6	Cautions on Using A/D Converter	148		
	14.7	Status at Reset	148		
. –					
15.	SERI		149		
	15.1	General	149		
	15.2	Clock Input/Output Control Block and Data Input/Output Control Block	150		
	15.3	Clock Control Block	153		
	15.4	Clock Counter	153		
	15.5	Presettable Shift Register	154		
	15.6	Wait Control Block	154		
	15.7	Serial Interface Operation	155		
	15.8	Notes on Setting and Reading Data	159		
	15.9	Operation Mode and Operational Outline of Each Blocks	160		
	15.10	Status on Reset	162		
16			162		
10.			103		
	10.1	Junut Selection Block and Brogrammable Divider	103		
	10.2	Input Selection Block and Programmable Divider	104		
	16.3	Reference Frequency Generator	168		
	16.4	Phase Comparator (FDET), Charge Pump, and Unlock FF	170		
	16.5	PLL Disabled Status	1/4		
	16.6	Using PLL Frequency Synthesizer	1/5		
	16.7	Status at Reset	179		
17.	INTE	RMEDIATE FREQUENCY (IF) COUNTER	180		
	17.1	Outline of Frequency Counter	180		
	17.2	IF Counter Input Selection Block and Gate Time Control Block	181		
	17.3	Start/Stop Control Block and IF Counter	183		
	17 4	Using IF Counter	189		
	17 5	Status at Reset	191		
	17.5		191		
18.	BEEP	)	192		
	18.1	Outlines of BEEP	192		
	18.2	Output Wave Form of BEEP	194		
	18.3	Status at Reset	195		

 $\star$ 

19. LCD	CONTROLLER/DRIVER	196
19.1	Outline of LCD Controller/Driver	196
19.2	LCD Drive Voltage Generation Block	197
19.3	LCD Segment Register	198
19.4	Segment Signal/General-Purpose Input Port Select Block	200
19.5	Common Signal Output and Segment Signal Output Timing Control Blocks	202
19.6	Common Signal and Segment Signal Output Waves	203
19.7	Using LCD Controller/Driver	205
19.8	Status at Reset	207
20. STAN	IDBY	208
20.1	Outline of Standby Function	208
20.2	Halt Function	209
20.3	Clock Stop Function	215
20.4	Device Operation in Halt and Clock Stop Status	217
20.5	Cautions on Processing of Each Pin in Halt and Clock Stop Status	217
24 DESI	- <b>T</b>	220
21. RESE		220
21.1		220
21.2		221
21.3	WDI&SP Reset	222
22. INST	RUCTION SET	228
22.1	Outline of Instruction Set	228
22.2	Legend	229
22.3	Instruction List	230
22.4	Assembler (RA17K) Embedded Macro Instruction	232
23. RESE	ERVED SYMBOLS	233
23.1	Data Buffer (DBF)	233
23.2	System Registers (SYSREG)	233
23.3	LCD Segment Register	234
23.4	Port Register	235
23.5	Register File (Control Register)	236
23.6	Peripheral Hardware Register	238
23.7	Others	238
24. ELE	CTRICAL CHARACTERISTICS	239
25. PAC	KAGE DRAWING	243
26. REC	OMMENDED SOLDERING CONDITIONS	244
APPEND	IX A. CAUTIONS ON CONNECTING CRYSTAL RESONATOR	245
APPEND	IX B. DEVELOPMENT TOOLS	246

MEMO

## 1. PIN FUNCTIONS

## 1.1 Pin Function List

Pin No.	Symbol		Function		Output Form	
1 2 3 4	PIC0/TM0 P1C1/TM1 P1C2/AMIFC P1C3/FMIFC/ AMIFC	<ul> <li>Port 1C, timer event input the AM/FM.</li> <li>PIC0 through P1C3 <ul> <li>4-bit input port</li> </ul> </li> <li>TM0 and TM1 <ul> <li>Timer event input pins</li> </ul> </li> <li>AMIFC <ul> <li>IF counter input pin for</li> </ul> </li> </ul>				
		At r	At reset On clock stop			
		Input (P1C0-P1C3)	Input (P1C0-P1C3)	Input (P1C0-P1C3)		
80	Vdd2	Power supply pins. Supp	bly the same voltage to thes	se pins.	_	
5 32	Vdd1 Vdd0	VDD2: Supplies power to t and to the IF count VDD1: Supplies power to VDD0: Supplies power to				
6	NC	No connection	_			
7 8	EO0 EO1	Output from the charge pump of the PLL frequency synthesizer. These pins output the result of comparing the phases of the divided frequency of local oscillation with the reference frequency			CMOS 3-state	
		At r	eset	On clock stop		
		Reset by RESET pin	WDT&SP reset			
		High-impedance output	High-impedance output	High-impedance output		
9 10	VCOL VCOH	<ul> <li>Input of the local oscillation (VCO) frequency of the PLL.</li> <li>VCOL <ul> <li>Active when HF or MF mode selected by program; otherwise; pulled down.</li> </ul> </li> <li>VCOH <ul> <li>Active when VHF mode selected by program; otherwise, pulled down.</li> </ul> </li> <li>Because these input pins are connected to an internal AC amplifier, cut the DC component of the input signal with a capacitor.</li> </ul>			_	
11	GND	Ground			_	
12	TEST	Test input pin Bo sure t				
13	RESET	Reset input			_	

 $\star$ 

Pin No.	Symbol	Function					
14 	P2B0	These pins form a 4-bit I/O port which can be set in input or output mode in 1-bit units.			These pins form a 4-bit I/O port which can be set in input or output mode in 1-bit units.		CMOS push-pull
17	P2B3	Ati	At reset				
		Reset by RESET pin	WDT&SP reset				
		Input	Input	Retained			
18	P0A0	These pins form a 2-bit o	utput port.		N-ch open-		
19	P0A1	Atı	reset	On clock stop	drain		
		Reset by RESET pin	Reset by RESET pin WDT&SP reset				
		Output low level	Output low level	Retained			
20 	P1A0	These pins form a 4-bit l/ 1-bit units.	O port which can be set in	input or output mode in	N-ch open- drain		
23	P1A3	At	reset	On clock stop			
		Reset by RESET pin	WDT&SP reset				
		Input	Input	Retained			
24 	P1D0	These pins form a 4-bit l, 1-bit units.	O port which can be set in	input or output mode in	N-ch open- drain		
27	P1D3	Atı	reset	On clock stop			
		Reset by RESET pin	WDT&SP reset		-		
		Input	Input	Retained			
28	P0B0/SCK	Port P0B, the I/O pins of	Port P0B, the I/O pins of the serial interface and BEEP output pin.		CMOS		
I	P0B1/SI1/SO2	<ul> <li>P0B3 through P0B0</li> </ul>			push-pull		
31	P0B2/SO1	<ul> <li>4-bit I/O port</li> </ul>					
	P0B3/BEEP	<ul> <li>Can be set in input of</li> </ul>	or output mode in 1-bit units	5.			
		• BEEP					
		<ul> <li>BEEP output</li> </ul>					
		• SO1, SO2, SI1					
		Serial data output pins serial I/O mode of seria	and serial data input pin w al interface 1 is selected.	hen the 3-wire or 2-wire			
		• SCK					
		Serial clock I/O	Serial clock I/O				
		At i	reset	On clock stop			
		Reset by RESET pin	WDT&SP reset		-		
		Input (P0B3-P0B0)	Input (P0B3-P0B0)	Retained			
33	CAPLCD0	These pins can be used	to connect capacitors for a	double circuit to generate	-		
34	CAPLCD1	power to drive an LCD. Connect 0.33 $\mu F$ capacitors between CAP_LCD0 and		between CAPLCD0 and			
37	CAPLCD2	CAPLCD1 pins and betwee	en CAPLCD2 and CAPLCD3 pi	ns.			
38	CAPLCD3						
35	REGLCD0	Regulator output pins of	power supply for LCD drive		-		
36	REGLCD1	Connect to GND with a 0	.1 μF capacitor.				
39	REGLCD2						

Pin No.	Symbol		Output Form		
40	COM0	These pins output the cor	nmon signals of the LCD c	ontroller/driver.	CMOS
1	1	At reset		On clock stop	3-state
43	COM3	Reset by RESET pin	WDT&SP reset		
		Output low level	Output low level	Output low level	
44	LCD0	These pins are the segme	ent signal output pins of the	e LCD controller/driver.	CMOS
1	I				push-pull
60	LCD16				
61	P2A0/LCD17	Port 2A input port, the se	gment signal output pins of	f the LCD controller/driver.	-
I I	P2A1/LCD18	P2A0 through P2A2			
63	P2A2/LCD19	3-bit input port			
		LCD17 through LCD19			
		LCD segment output		1	
		At r	eset	On clock stop	
		Reset by RESET pin	WDT&SP reset		
		Input (P2A2-P2A0)	Input (P2A2-P2A0)	Retained	
64	P0C0	4-bit output			CMOS
I.	I	At r	eset	On clock stop	push-pull
67	P0C3	Reset by RESET pin	WDT&SP reset		
		Output low level	Output low level	Retained	
68	INT	Edge-detected vector inte	_		
		Rising or falling edge is s			
69	Хоит	Crystal resonator connect	tion pin.		_
70	Xin				
71	P2C0	These pins form a 4-bit I/	O port which can be set in	the input or output mode	CMOS
1	I	in 1-bit units.			push-pull
74	P2C3	At r	eset	On clock stop	
		Reset by RESET pin	WDT&SP reset		
		Input	Input	Retained	
75	P0D0	Port 0D input port, A/D co	onverter input pins and to re	elease the HALT	-
1	P0D1/AD0	and STOP modes.			
78	P0D2/AD1	<ul> <li>P0D0 through P0D3</li> </ul>			
	P0D3/AD2	4-bit input port			
		AD0 through AD2			
		Analog input of A/D cor			
		HALT and STOP mode releasing			
		Releases HALT or STO	P mode when a high level	is input.	
		At r	eset	On clock stop	
		Reset by RESET pin	WDT&SP reset		
		Input with pull-down resistor (P0D3-P0D0)	Input with pull-down resistor (P0D3-P0D0)	Retained	

#### **1.2 Equivalent Circuits of Pins**

(1) P0B (P0B3/BEEP, P0B2/SO1, P0B1/SI1/SO2, P0B0/SCK) P2B (P2B3, P2B2, P2B1, P2B0) P2C (P2C3, P2C2, P2C1, P2C0) (I/O)





(3) P0A (P0A1, P0A0) (Output)



 $\star$ 

(4) P0C (P0C3, P0C2, P0C1, P0C0) (Output)



(5) POD (P0D3/AD2, P0D2/AD1, P0D1/AD0, P0D0) P1C (P1C3/FMIFC/AMIFC, P1C2/AMIFC, P1C1/TM1, PIC0/TM0) P2A (P2A2/LCD19, P2A1/LCD18, P2A0/LCD17)



(6)  $\frac{INT}{RESET}$  (Schmitt trigger input)



(7) XOUT (Output), XIN (Input)



(8) EO1, EO0 (Output)



(9) COM3 through COM0 (Output)



(10) VCOH (Input)



(11)VCOL (Input)



#### 1.3 Recommended Connections of Unused Pins

It is recommended to connect the unused pins as follows:

Table 1-1.	Recommended	Connections	of	Unused	Pins
------------	-------------	-------------	----	--------	------

	Pin Name		I/O Form	Recommended Connection
	Port pins	P0D3/AD2-P0D1/AD0,P0D0	Input	Connect each pin to GND via resistorNote 1.
		P1C3/FMIFC/AMIFC <sup>Note 2</sup>		
*		P1C2/AMIFCNote 2		
		P1C1/TM1		Set each pin in port mode and connect it to VDD or GND
		P1C0/TM0		via resistor.
		P0A1, P0A0	Output	Set in low-level output mode via software and leave
		P0C3-P0C0		unconnected.
		P0B3/BEEP	I/ONote 3	Set in general-purpose input port mode via software, and
		P0B2/SO1		connect each pin to VDD or GND via resistor.
		P0B1/SI1/SO2		
		P0B0/SCK		
		P1A3-P1A0		
		P1D3-P1D0		
		P2A2/LCD19		
		P2A1/LCD18		
		P2A0/LCD17		
		P2B3-P2B0		
		P2C3-P2C0		
	Pins other	EO1	Output	Leave unconnected.
	than port	EO0		
	pins	INT	Input	Connect to GND via resistor <sup>Note 1</sup> .
		NC	_	Leave unconnected.
		TEST	_	Directly connect to GND.
		VCOH	Input	Set to PLL disable via software and leave unconnected.
		VCOL		

- Notes 1. If these pins are externally pulled up (connected to V<sub>DD</sub> via resistor) or down (connected to GND via resistor) with a high resistor, the pins almost go into a high-impedance state, increasing the current consumption (through current) of the port. Although the value of the pull-up or pull-down resistor varies depending on the application circuit, it generally is several 10 kΩ.
  - 2. The circuit of the general-purpose input port is designed not to increase the current consumption even in the high-impedance state. Do not set these pins in the AMIFC and FMIFC modes; otherwise, the current consumption will increase.
  - **3.** The I/O ports are set in the general-purpose input port mode at reset by the RESET pin and reset due to overflow/underflow of the watchdog timer or stack, and on execution of the clock stop instruction.

\*

#### 1.4 Cautions on Using TEST Pin

When VDD is applied to the TEST pin, the device is set in the test mode. Therefore, be sure to keep the wiring length of this pin as short as possible, and directly connect it to the GND pin.

If the wiring length between the TEST pin and GND pin is too long, or if external noise is superimposed on the TEST pin, generating a potential difference between the TEST pin and GND pin, your program may not run normally.



#### 2. PROGRAM MEMORY (ROM)

#### 2.1 Outline of Program Memory

Figure 2-1 outlines the program memory.

As shown in this figure, the addresses of the program memory are specified by the program counter. The program memory has the following two major functions.

- To store programs
- To store constant data





#### 2.2 Program Memory

Figure 2-2 shows the configuration of the program memory.

As shown in this figure, the program memory consists of:

 $\mu$ PD17933: 6144 × 16 bits (0000H-17FFH)

 $\mu$ PD17934: 8192 × 16 bits (0000H-1FFFH)

Because all "instructions" are "one-word instructions", one instruction can be stored to one address of the program memory.

As constant data, the contents of the program memory are read to the data buffer by using a table reference instruction.



Figure 2-2. Configuration of Program Memory

#### 2.3 Program Counter

#### 2.3.1 Configuration of program counter

Figure 2-3 shows the configuration of the program counter.

As shown in this figure, the program counter consists of a 13-bit binary counter. Bits 11 and 12 of the program counter indicate a page.

The program counter specifies an address of the program memory.

#### Figure 2-3. Configuration of Program Counter

PC <sub>12</sub>	PC <sub>11</sub>	PC <sub>10</sub>	PC <sub>9</sub>	PC <sub>8</sub>	PC7	PC <sub>6</sub>	PC₅	PC <sub>4</sub>	PC₃	PC <sub>2</sub>	PC1	PC <sub>0</sub>
Pa	ge											
PC												

#### 2.4 Flow of Program

The flow of the program is controlled by the program counter that specifies an address of the program memory.

The program flow when each instruction is executed is described below.

Figure 2-4 shows the value that is set to the program counter when each instruction is executed.

Table 2-4 shows the vector address when an interrupt is accepted.

#### 2.4.1 Branch instruction

#### (1) Direct branch ("BR addr")

The branch destination addresses of the direct branch instruction are all the addressess of the program memory.

#### (2) Indirect branch ("BR @AR")

The branch destination addresses of the indirect branch instruction are all the addresses of the program memory. The addresses are 0000H through 17FFH in the  $\mu$ PD17933, and 0000H through 1FFFH in the  $\mu$ PD17934.

Refer to 5.3 Address Register (AR).

#### 2.4.2 Subroutine

#### (1) Direct subroutine call ("CALL addr")

The first address of a subroutine that can be called by the direct subroutine instruction is in page 0 (addresses 0000H through 07FFH).

#### (2) Indirect subroutine call (CALL @AR)

The first addresses of a subroutine that can be called by the indirect subroutine call instruction are all the addresses of the program memory. The addresses are 0000H through 17FFH in the  $\mu$ PD17933, and 0000H through 1FFFH in the  $\mu$ PD17934.

Refer to 5.3 Address Register (AR).

#### 2.4.3 Table reference

The addresses that can be referenced by the table reference instruction ("MOVT DBF, @AR") are all the addresses of the program memory. The addresses are 0000H through 17FFH in the  $\mu$ PD17933, and 0000H through 1FFFH in the  $\mu$ PD17934.

Refer to 5.3 Address Register (AR) and 9.2.2 Table reference instruction (MOVT, DBF, @AR).

Pro	gram counter	r Contents of Program Counter (PC)												
Instruction		<b>b</b> 12	<b>b</b> 11	<b>b</b> 10	b۹	b8	b7	b6	b₅	b4	bз	b2	b1	bo
BR addr	Page 0	0	0											
	Page 1	0	1											
	Page 2	1	0				~		. ,		,			
	Page 3	1	1	Operand of instruction (addr)										
CALL addr	0	0	Operand of instruction (addr)											
BR @AR														
CALL @AR														
MOVT DBF, @AR	1	Contents of address register												
RET														
RETSK		Contents of address stack register (ASR) (return address)												
RETI		specified by stack pointer (SP)												
Other instructions														
(including skip instru	uction)	Increment												
When interrupt is ac	Vector address of each interrupt													
Watchdog timer rese	ət,													
reset by RESET pin	0	0	0	0	0	0	0	0	0	0	0	0	0	

## Figure 2-4. Value of Program Counter Upon Execution of Instruction

#### Table 2-1. Interrupt Vector Address

Order	Internal/External	Interrupt Source	Vector Address
1	External	INT pin	0004H
2	Internal	Timer 0	0003H
3	Internal	Basic timer 1	0002H
4	Internal	Serial interface 1	0001H

#### 2.5 Cautions on Using Program Memory

#### (1) µPD17933

The program memory of the  $\mu$ PD17933 is assigned to addresses 0000H through 17FFH. However, addresses that can be specified by the program counter (PC) are 0000H through 1FFFH, therefore, be careful on the following points when specifying the program memory addresses.

- Write a branch instruction when writing an instruction to address 17FFH.
- Do not write anything to addresses 1800H through 1FFFH.
- Do not branch to addresses 1800H through 1FFFH.

#### **(2)** μ**PD17934**

The program memory of the  $\mu$ PD17934 is assigned to addresses 0000H through 1FFFH. Be careful on the following points.

• Write a branch instruction when writing an instruction to address 1FFFH.

#### 3. ADDRESS STACK (ASK)

#### 3.1 Outline of Address Stack

Figure 3-1 outlines the address stack.

The address stack consists of a stack pointer and address stack registers.

The address of an address stack register is specified by the stack pointer.

The address stack saves a return address when a subroutine call instruction is executed or when an interrupt is accepted.

The address stack is also used when the table reference instruction is executed.

Figure 3-1. Outline of Address Stack



#### 3.2 Address Stack Register (ASR)

Figure 3-2 shows the configuration of the address stack register.

The address stack register consists of sixteen 13-bit registers ASR0 through ASR15. Actually, however, it consists of fifteen 13-bit registers (ASR0 through ASR14) because no register is allocated to ASR15.

The address stack saves a return address when a subroutine is called, when an interrupt is accepted, and when the table reference instruction is executed.

S	tack   (S	point P)	er	Address stack register (ASR)														
	В	it			Address							Bit						
bз	b2	b1	bo			<b>b</b> 12	<b>b</b> 11	<b>b</b> 10	b۹	b8	b7	b6	b₅	b4	bз	b2	b1	bo
SP3	SP2	SP1	SP0	┍╴╸	ОН							ASRO	)					
					 1H				+								†	
							I	L	1	1	, 	ASR1		1	I	L	1	
				-	2H								 >					
									+		, 		<u>-</u> 				+	
				-	3H		I	I		I	1	ASR3	3	I	I	I	I	I
					 4H				+								+	
								<b></b>	+	4	ر 	ASR4	<b>↓</b> ↓	4		<b>⊢</b>	+	
				-	5H													
					ец				+				, 				+	
				-			I	I	I	I	1	ASR	6	I	I	I	I	I
					7H				+								+	1
								<b> </b> ·	+		ر 	ASR7	<b>7</b> 			+	+	
				-	8H							ASR	3					
					ан				+								+	
				-					, I			ASRS	)				'	
				-	AH					1							1	1
								·	+		A 	\SR1	U 			<b>⊦</b>	+	
				-	BH						 A	SR1	1					I
					СН				+								+	
							I	L	1	1	A	SR1	2	1	I	L	1	
				-	DH								) 					
									+		F 		ວ 				+	
					EH		I	I		I	A	SR1	4	I	I	I		
					FH												+	
							I	1	I	AS	R15	(Unc	lefine	ed)	1	1	1	I

## Figure 3-2. Configuration of Address Stack Register

#### 3.3 Stack Pointer (SP)

#### 3.3.1 Configuration and function of stack pointer

Figure 3-3 shows the configuration and functions of the stack pointer.

The stack pointer consists of a 4-bit binary counter.

It specifies the address of an address stack register.

A value can be directly read from or written to the stack pointer by using a register manipulation instruction.

Figure 3-3. Configuration and Function of Stack Pointer

Name	FI	ag s	ymb	ol	Address	Read/Write								
	bз	b2	b1	bo										
Stack pointer					01H	R/W								
(SP)	s	s	s	s										
	Р	P	P	Р										
	3	2	¦ ¦ 1	0										
	_													
					S	Specifies address of address stack register (ASR)								
	0	0	0	0	Address 0 (AS	Address 0 (ASR0)								
	0	0	0	1	Address 1 (AS	Address 1 (ASR1)								
	0	0	1	0	Address 2 (ASR2)									
	0	0	1	1	Address 3 (ASR3)									
	0	1	0	0	Address 4 (AS	R4)								
	0	1	0	1	Address 5 (AS	R5)								
	0	1	1	0	Address 6 (AS	R6)								
	0	1	1	1	Address 7 (ASR7)									
	1	0	0	0	Address 8 (ASR8)									
	1	0	0	1	Address 9 (ASR9)									
	1	0	1	0	Address 10 (ASR10)									
	1	0	1	1	Address 11 (ASR11)									
	1	1	0	0	Address 12 (ASR12)									
	1	1	1	0	Address 14 (A	SR14)								
	1	1	1	1	Setting prohibit	ted								

eset	Reset by RESET pin	1	1	1	1
At re	WDT&SP reset	1	1	1	1
Clo	ock stop	Re	tain	ed	

Reset by RESET pin : Reset by RESET pin

WDT&SP reset : Reset by watchdog timer and stack pointer

Clock stop : Upon execution of clock stop instruction

#### 3.4 Operation of Address Stack

#### 3.4.1 Subroutine call instruction ("CALL addr", "CALL @AR") and return instruction ("RET", "RETSK")

When a subroutine call instruction is executed, the value of the stack pointer is decremented by one, and the return address is stored to an address stack register specified by the stack pointer.

When the return instruction is executed, the contents of the address stack register (return address) specified by the stack pointer are restored to the program counter, and the value of the stack pointer is incremented by one.

#### 3.4.2 Table reference instruction ("MOVT DBF, @AR")

When the table reference instruction is executed, the value of the stack pointer is incremented by one, and the return address is stored to an address stack register specified by the stack pointer.

Next, the contents of the program memory specified by the address register are read to the data buffer, the contents of the address stack register (return value) specified by the stack pointer are restored to the program counter, and the value of the stack pointer is incremented by one.

#### 3.4.3 When interrupt is accepted and on execution of return instruction ("RETI")

When an interrupt is accepted, the value of the stack pointer is decremented by one, and the return address is stored to an address stack register specified by the stack pointer.

When the return instruction is executed, the contents of an address stack register (return value) specified by the stack pointer are restored to the program counter, and the value of the stack pointer is incremented by one.

#### 3.4.4 Address stack manipulation instruction ("PUSH AR", "POP AR")

When the "PUSH" instruction is executed, the value of the stack pointer is decremented by one, and the contents of the address register are transferred to an address stack register specified by the stack pointer.

When the "POP" instruction is executed, the contents of an address stack register specified by the stack pointer are transferred to the address register, and the value of the stack pointer is incremented by one.

#### 3.5 Cautions on Using Address Stack

#### 3.5.1 Nesting level and operation on overflow

The value of address stack register (ASR15) is "undefined" when the value of the stack pointer is 0FH. Accordingly, if a subroutine call exceeding 15 levels, or an interrupt is used without manipulating the stack, execution returns to an "undefined" address.

#### 3.5.2 Reset on detection of overflow or underflow of address stack

Whether the device is reset on detection of overflow or underflow of the address stack can be specified by program. At reset, the program is started from address 0, and some control registers are initialized.

This reset function is valid at reset by the RESET pin. For details, refer to 21. RESET.

#### 4. DATA MEMORY (RAM)

### 4.1 Outline of Data Memory

Figure 4-1 outlines the data memory.

As shown in the figure, system registers, a data buffer, port registers, LCD segment registers, and control registers are located on the data memory.

The data memory stores data, transfers data with the peripheral hardware or ports, and controls the CPU.



#### Figure 4-1. Outline of Data Memory

- Notes 1. LCD segment registers are allocated to addresses 5CH through 6FH of BANK14.
  - 2. Control registers are allocated to addresses 00H through 6FH of BANK15. Port input/output select registers are allocated to 60H through 6FH.
- Cautions 1. Never write anything to address 31H of BANK15 because this address is a test mode area.2. Address 5BH is not provided to BANK4 through BANK14.

#### 4.2 Configuration and Function of Data Memory

Figure 4-2 shows the configuration of the data memory.

As shown in this figure, the data memory is divided into several banks with each bank made up of a total of 128 nibbles with 7H row addresses and 0FH column addresses.

The data memory can be divided into five functional blocks. Each block is described in 4.2.1 through 4.2.6 below.

The contents of the data memory can be operated on, compared, judged, and transferred in 4-bit units with a single data memory manipulation instruction.

Table 4-1 lists the data memory manipulation instructions.

#### 4.2.1 System registers (SYSREG)

The system registers are allocated to addresses 74H through 7FH.

Because the system registers are allocated to all banks, the same system registers exist at addresses 74H through 7FH of any bank.

For details, refer to 5. SYSTEM REGISTER (SYSREG).

#### 4.2.2 Data buffer (DBF)

The data buffer is allocated to addresses 0CH through 0FH of BANK 0. For details, refer to **9. DATA BUFFER (DBF)**.

#### 4.2.3 Port registers

The port registers are allocated to addresses 70H through 73H of BANKs 0 through 2. For details, refer to **11. GENERAL-PURPOSE PORTS**.

#### 4.2.4 Control registers and port input/output select registers

The control registers are allocated to addresses 00H through 6FH of BANK 15. Of these registers, the port input/output select registers are allocated to addresses 60H through 6FH of BANK15. Addresses 00H through 3FH of BANK15, to which control registers are allocated, also overlap addresses 00H through 3FH of the register file.

For details, refer to 8. REGISTER FILE (RF) AND CONTROL REGISTERS.

#### 4.2.5 LCD segment registers

The LCD segment registers consists of a total of 20 nibbles at addresses 5CH through 6FH of BANK15 of the data memory.

For details, refer to 8.4 LCD Segment Registers and 19. LCD CONTROLLER/DRIVER.

#### 4.2.6 General-purpose data memory

The general-purpose data memory is allocated to the addresses of the data memory excluding those of the system registers, control registers, port registers, port input/output selection registers, and LCD segment register.

The general-purpose data memory consists of a total of 448 nibbles of the 112 nibbles each of BANKs 0 through 3.

Row address



Figure 4-2. Configuration of Data Memory

Notes 1. The high-order 2 bits of 70H are fixed to 0.

- 2. The same system register exists.
- 3. Address 71H of BANK1, and the high-order 1 bit and address 73H of BANK2 are fixed to 0.

Cautions 1. Never write anything to address 31H of BANK15 because this address is a test mode area.

2. Address 5BH is not provided to BANK4 through BANK14.
| Fund      | ction    | Instruction                 |  |  |  |
|-----------|----------|-----------------------------|--|--|--|
| Operation | Add      | ADD<br>ADDC                 |  |  |  |
|           | Subtract | SUB<br>SUBC                 |  |  |  |
|           | Logic    | AND<br>OR<br>XOR            |  |  |  |
| Compare   |          | SKE<br>SKGE<br>SKLT<br>SKNE |  |  |  |
| Transfer  |          | MOV<br>LD<br>ST             |  |  |  |
| Judge     |          | SKT<br>SKF                  |  |  |  |

#### Table 4-1. Data Memory Manipulation Instructions

## 4.3 Data Memory Addressing

Figure 4-3 shows address specification of the data memory.

An address of the data memory is specified by a bank, row address, and column address.

A row address and a column address are directly specified by a data memory manipulation instruction. However, a bank is specified by the contents of a bank register.

For the details of the bank register, refer to 5. SYSTEM REGISTER (SYSREG).

Figure 4-3.	Address	Specification	of Data	Memory
-------------	---------	---------------	---------	--------

		Ва	nk		Rov	v addı	ess	Co	lumn	addre	SS
	bз	b2	b₁	bo	b2	b1	bo	b₃	b2	b₁	bo
Data memory address	E	Bank r	egiste	er		lı	nstruc	tion o	peran	d	

## 4.4 Cautions on Using Data Memory

## 4.4.1 At reset by RESET pin

The contents of the general-purpose data memory are "undefined" at reset by  $\overrightarrow{\mathsf{RESET}}$  pin. Initialize the data memory as necessary.

## 4.4.2 Cautions on data memory not provided

If a data memory manipulation instruction that reads the data memory is executed to a data memory address not provided, undefined data is read.

Nothing is changed even if data is written to such an address.

## 5. SYSTEM REGISTERS (SYSREG)

## 5.1 Outline of System Registers

Figure 5-1 shows the location of the system registers on the data memory and their outline.

As shown in the figure, the system registers are allocated to addresses 74H through 7FH of all the banks of the data memory. Therefore, identical system registers exist at addresses 74H through 7FH of any bank. Because the system registers are located on the data memory, they can be manipulated by all data memory manipulation instructions.

Seven types of system registers are available depending on function.





Address	74H	75H	76H	77H	78H	79H	7AH	7BH	7CH	7DH	7EH	7FH
Name		Address	register		Window	Bank	Ir	ndex regist	er	General r	egister	Program
		(A	R)		register	register		(IX)		pointer (F	RP)	status
					(WR)	(BANK)	Data mem	ory row				word
							address p	ointer (MP)				(PSWORD)
Function	Contro	ls program	memory a	address	Transfers	Specifies	Modifies a	ddress of da	ata memory	Specifies		Controls
					data with	bank of				address o	of	operation
					register	data				general re	egister	
					file	memory						

# 5.2 System Register List

Figure 5-2 shows the configurations of the system registers.

Figure 5-2.	Configuration	of	System	Registers
-------------	---------------	----	--------	-----------

Address	74H			75ł	-		76	н		77	Н			78	'8H 79		9Н	1		7AF	+		7	BH			7C	H		7	Έ	1		78	7EH			7FH				
Name																		Sys	ste	m	regi	iste	rs		_																	
			A	dd	res	s re	gis	ter				,	W	inc	lo	w	E	3ar	۱k				I	Ind	lex	reç	gist	er			0	Ger	ner	ralı	reç	giste	er	Pr	œ	ırar	n	
					(/	AR)	Į.						re	gis	ste	r	r	egi	iste	ər					(	IX)					þ	oir	nte	er (F	٢P	)		st	status word			
							(WR)					(	ΒA	۱N۲	(ک	Dat	a m	iem	nor	y ro	w									(PSWORD			RD)									
																				address pointer (MP)																						
Symbol	AR3			AR	2		AR	1		AR	20			W	R			ΒA	١N	K		IXH	ł		D	٢M			IX	L		R	۱۹	4		R	PL			P۶	w	
																					MPH		MPL		1																	
Bit	b3 b2 b1	bol	b₃k	)2b	1 b	0 <b>b</b> 3	b2 ł	<b>)</b> 1 b0	bз	b2	bık	20 k	)3 l	b2	b1	bo	bз	b2	2 <b>b</b> 1	bo	bзk	22b	1 b	ob	3 <b>b</b> 2	2 <b>b</b> 1	bo	bз	b2	b1 b	юb	зb	2 b	1 <b>b</b> c	ba	3 b2	b1	bo	bз	b2	<b>b</b> 1 <b>b</b> 0	
Data																	   		М			: (	(IX)	)			T	-	-	-		   	   	в	С	С	ZI					
						+				_	_	-	•	_		-	-	P		-► P					-	<u>_</u>		(RF	') ⊤			С	м	Y	x							
																		E		E (MP)		-								-	D	Р		E								

## 5.3 Address Register (AR)

## 5.3.1 Configuration of address register

Figure 5-3 shows the configuration of the address register.

As shown in the figure, the address register consists of 16 bits of system register addresses 74H through 77H (AR3 through AR0).

	Address		74H				75	БH			70	6H			7	7H	
	Name							Addre	ess re	gister	(AR)						
	Symbol		AF	۲3			AF	R2			A	R1			A	R0	
	Bit	bз	b2	b1	bo	b₃	b2	b1	bo	bз	b2	b1	bo	b₃	b2	b1	bo
	Data	́М															∧ L
		S															S
		B ~			       								       			       	B ∽
eset	Reset by RESET pin		0				(	)		0					(	C	
At re	WDT&SP reset		0			0				0				0			
	Clock stop		Retained				Reta	ined		Retained				Retained			

## Figure 5-3. Configuration of Address Register

Reset by RESET pin : Reset by RESET pin

WDT&SP reset : Reset by watchdog timer and stack pointer

Clock stop

: On execution of clock stop instruction

## 5.3.2 Function of address register

The address register specifies a program memory address when the table reference instruction ("MOVT DBF, @AR"), stack manipulation instruction ("PUSH AR", "POP AR"), indirect branch instruction ("BR @AR"), or indirect subroutine call instruction ("CALL @AR") is executed.

A dedicated instruction ("INC AR") is available that can increment the contents of the address instruction by one.

The following paragraphs (1) through (5) describe the operation of the address register when the respective instructions are executed.

## (1) Table reference instruction ("MOVT DBF, @AR")

When the table reference instruction is executed, the constant data (16 bits) of a program memory address specified by the contents of the address register are read to the data buffer.

The constant data that can be specified by the address register is stored to address 0000H to 17FFH in the case of the  $\mu$ PD17933, and address 0000H to 1FFFH in the case of the  $\mu$ PD17934.

## (2) Stack manipulation instruction ("PUSH AR", "POP AR")

When the "PUSH AR" instruction is executed, the value of the stack pointer is decremented by one, and the contents of the address register (AR) are transferred to an address stack register specified by the stack pointer whose value has been decremented by one.

When the "POP AR" instruction is executed, the contents of an address stack register specified by the stack pointer are transferred to the address register, and the value of the stack pointer is incremented by one.

## (3) Indirect branch instruction ("BR @AR")

When this instruction is executed, the program branches to a program memory address specified by the contents of the address register.

The branch address that can be specified by the address register is 0000H to 17FFH in the case of the  $\mu$ PD17933, and 0000H to 1FFFH in the case of the  $\mu$ PD17934.

## (4) Indirect subroutine call instruction ("CALL @AR")

The subroutine at a program memory address specified by the contents of the address register can be called.

The first address of the subroutine that can be specified by the address register is 0000H to 17FFH in the case of the  $\mu$ PD17933, and 0000H to 1FFFH in the case of the  $\mu$ PD17934.

## (5) Address register increment instruction ("INC AR")

This instruction increments the contents of the address register by one.

## 5.3.3 Address register and data buffer

The address register can transfer data as part of the peripheral hardware via the data buffer. For details, refer to **9. DATA BUFFER (DBF)**.

## 5.3.4 Cautions on Using Address Register

Because the address register is configured in 16 bits, it can specify an address up to FFFFH.

However, the program memory exists at addresses 0000H through 17FFH in the case of the  $\mu$ PD17933 and addresses 0000H through 1FFFH in the case of the  $\mu$ PD17934.

Therefore, the maximum value that can be set to the address register of the  $\mu$ PD17933 is address 17FFH. In the case of the  $\mu$ PD17934, it is address 1FFFH.

## 5.4 Window Register (WR)

## 5.4.1 Configuration of window register

Figure 5-4 shows the configuration of the window register.

As shown in the figure, the window register consists of 4 bits of system register address 78H (WR).

	Address		78	ЗН	
	Name	W	indow	regis	ster
			(V)	/R)	
	Symbol		N	/R	
	Bit	b₃	b2	b1	bo
	Data	Â			Ĺ
		S			S
		B ∽			B ~
eset	Reset by RESET pin		Unde	efined	
At re	WDT&SP reset		Reta	ained	
	Clock stop				

Figure 5-4.	Configuration	of	Window	Register
-------------	---------------	----	--------	----------

## 5.4.2 Function of window register

The window register is used to transfer data with the register file (RF) to be described later.

Data transfer between the window register and register file is manipulated by using dedicated instructions "PEEK WR, rf" and "POKE, rf WR" (rf: address of register file).

The following paragraphs (1) and (2) describe the operation of the window register when these instructions are executed.

For further information, also refer to 8. REGISTER FILE (RF) AND CONTROL REGISTERS.

## (1) "PEEK WR, rf" instruction

When this instruction is executed, the contents of the register file addressed by "rf" are transferred to the window register.

## (2) "POKE rf, WR" instruction

When this instruction is executed, the contents of the window register are transferred to the register file addressed by "rf".

## 5.5 Bank Register (BANK)

## 5.5.1 Configuration of bank register

Figure 5-5 shows the configuration of the bank register.

As shown in the figure, the bank register consists of 4 bits of system register address 79H (BANK).

	Address		79	ЭH					
	Name	E	Bank i	regist	er				
			(BA	NK)					
	Symbol		BA	NK					
	Bit	bз	b2	b1	bo				
	Data	́М			r >				
		S	, , , ,	S					
		B		       	B				
eset	Reset by RESET pin			0					
At re	WDT&SP reset	0							
	Clock stop	Retained							

Figure 5-5. Configuration of Bank Register

#### 5.5.2 Function of bank register

The bank register specifies a bank of the data memory.

Table 5-1 shows the relationships between the value of the bank register and a bank of the data memory that is specified.

Because the bank register is one of the system registers, its contents can be rewritten regardless of the bank currently specified.

When manipulating a bank register, therefore, the status of the bank at that time is irrelevant.

B	ank F	Regis	ter	Bank of Data
	(BA	NK)		Memory
b₃	b2	b1	bo	
0	0	0	0	BANK0
0	0	0	1	BANK1
0	0	1	0	BANK2
0	0	1	1	BANK3
1	1	1	0	BANK14
1	¦ 1	1	1	BANK15

Table 5-1. Data Memory Bank Specification

**Remark** Addresses 00H through 5BH are not provided to BANK4 through BANK14.

Caution The area to which the data memory is allocated differs depending on the model. For details, refer to Figure 4-2 Configuration of Data Memory.

## 5.6 Index Register (IX) and Data Memory Row Address Pointer (MP: memory pointer)

#### 5.6.1 Configuration of index register and data memory row address pointer

Figure 5-6 shows the configuration of the index register and data memory row address pointer.

As shown in the figure, the index register consists of an index register (IX) made up of 11 bits (the low-order 3 bits (IXH) of system register address 7AH, and 7BH and 7CH (IXM, IXL)) and an index enable flag (IXE) at the lowest bit position of 7FH (PSW).

The data memory row address pointer (memory pointer) consists of a data memory row address pointer (MP) that is made up of 7 bits of the low-order 3 bits of 7AH (MPH) and 7BH (MPL), and a data memory row address pointer enable flag (memory pointer enable flag: MPE) at the lowest bit position of 7AH (MPH).

In other words, the high-order 7 bits of the index register are shared with the data memory row address pointer

	Address		7/	٩H			71	BH			70	СН			7	EH		7FH				
	Name					Ind	ex re	gister	(IX)								Pi	ograr	n stat	us wo	ord	
				Mem	ory po	ointer	(MP)			, , ,							(P	SWO	RD)			
	Symbol		D	кн			D	KM			D	٢L							PS	SW		
			М	PH			М	PL														
	Bit	bз	b2	b1	bo	b₃	b2	b1	bo	bз	b2	b1	bo	bз	b2	b1	bo	bз	b2	b1	bo	
	Data	м	Â										$\langle -$				- - - -				-	
			M S													1					ı V	
		Р	5	   				1	1		1	   	5		   	, , ,	1 1 1				X	
		E	B				IX						B								Е	
			-						~													
			М						L													
			S						S						   							
			B MP			Ρ			B													
set	Reset by RESET pin		0				0					0									0	
Atre	WDT&SP reset		0				0				0										0	
	Clock stop		Retained				Retained				Retained										R	

#### Figure 5-6. Configuration of Index Register and Data Memory Row Address Pointer

R: retained

## 5.6.2 Functions of index register and data memory row address pointer

The index register and data memory row address pointer modify the addresses of the data memory. The following paragraphs (1) and (2) describe their functions.

A dedicated instruction ("INC IX") that increments the value of the index register by one is available. For the details of address modification, refer to **7. ALU (Arithmetic Logic Unit) BLOCK**.

## (1) Index register (IX)

When a data memory manipulation instruction is executed, the data memory address is modified by the contents of the index register.

This modification, however, is valid only when the IXE flag is set to 1.

To modify the address, the bank, row address, and column address of the data memory are ORed with the contents of the index register, and the instruction is executed to a data memory address (called real address) specified by the result of this OR operation.

All data memory manipulation instructions are subject to address modification by the index register. The following instructions, however, are not subject to address modification by the index register.

INC	AR	RORC	r
INX	IX	CALL	addr
MOVT	DBF, @AR	CALL	@AR
PUSH	AR	RET	
POP	AR	RETSK	
PEEK	WR,rf	RETI	
POKE	rf,WR	EI	
GET	DBF,p	DI	
PUT	p, DBF	STOP s	
BR	addr	HALT h	
BR	@AR	NOP	

## (2) Data memory row address pointer (MP)

When the general register indirect transfer instruction ("MOV @r,m" or "MOV m,@r") is executed, the indirect transfer destination address is modified.

This modification, however, is valid only when the MPE flag is set to 1.

To modify the address, the bank and row address at the indirect transfer destination are replaced by the contents of the data memory row address pointer.

Instructions other than the general register indirect transfer instruction are not subject to address modification.

## (3) Index register increment instruction ("INC IX")

This instruction increments the contents of the index register by one.

Because the index register is configured of 10 bits, its contents are incremented to "000H" if the "INC IX" instruction is executed when the contents of the index register are "3FFH".

## 5.7 General Register Pointer (RP)

## 5.7.1 Configuration of General Register Pointer

Figure 5-7 shows the configuration of the general register pointer.

As shown in the figure, the general register pointer consists of 7 bits including 4 bits of system register address 7DH (RPH) and the high-order 3 bits of address 7EH (RPL).

	Address		7[	ЭН		7EH						
	Name	Ger	neral i	registe	ər poi	nter						
		(RP	)									
	Symbol		R	PH			RI	PL				
	Bit	bз	b2	b1	bo	b₃	b2	b1	bo			
	Data	Â						Ê	В			
		S					1	S	С			
		B						B	D >			
eset	Reset by RESET pin			0			(	C				
At re	WDT&SP reset			0	0							
	Clock stop		Reta	ained		Retained						

#### Figure 5-7. Configuration of General Register Pointer

## 5.7.2 Function of general register pointer

The general register pointer specifies a general register on the data memory.

Figure 5-8 shows the addresses of the general registers specified by the general register pointer.

As shown in the figure, a bank is specified by the high-order 4 bits (RPH: address 7DH) of the general register pointer, and a row address is specified by the low-order 3 bits (RPL: address 7EH).

Because the valid number of bits of the general register pointer is 7, all the row addresses (0H through 7FH) of all the banks can be specified as general registers.

For the details of the operation of the general register, refer to 6. GENERAL REGISTER (GR).

Figure 5-8. Address of General Register Specified by General Register Pointer





## 5.7.3 Cautions on using general register pointer

The lowest bit of address 7EH (RPL) of the general register pointer is allocated as the BCD flag of the program status word.

When rewriting RPL, therefore, pay attention to the value of the BCD flag.

## 5.8 Program Status Word (PSWORD)

#### 5.8.1 Configuration of program status word

Figure 5-9 shows the configuration of the program status word.

As shown in the figure, th program status word consists of a total of 5 bits including the lowest bit of system register address 7EH (RPL) and 4 bits of address 7FH (PSW).

Each bit of the program status word has its own function. The 5 bits of the program status word are BCD flag (BCD), compare flag (CMP), carry flag (CY), zero flag (Z), and index enable flag (IXE).

	Address		71	ΞH		7FH							
	Name				Prog	gram status word							
					(PS\	WORD)							
	Symbol		R	PL		PSW							
	Bit	b₃	b2	b1	bo	b₃	b2	b1	bo				
	Data				В	С	С	Z	Ι				
					с	М	Y		х				
					D	Ρ			Е				
		1											
eset	Reset by RESET pin			0				0					
At re	WDT&SP reset			0		0							
	Clock stop		Reta	ained			Reta	ained					

## Figure 5-9. Configuration of Program Status Word

## 5.8.2 Function of program status word

The program status word is a register that sets the conditions under which the ALU (Arithmetic Logic Unit) executes an operation or data transfer, or indicates the result of an operation.

Table 5-2 outlines the function of each flag of the program status word.

For details, refer to 7. ALU (Arithmetic Logic Unit) BLOCK.

## Table 5-2. Outline of Function of Each Flag of Program Status Word

	(RP)		F	Prog	am S	Statu	JS			
			N	/ord	(PSV	VOR	D)			
	RF	۶L			PS	SW				
bз	b <sub>2</sub>	b1	bo	bз	b <sub>2</sub>	b1	bo			
			В	С	С	7	1			
			-		v	-	v			
						1				
									Flag Name	Function
								-	Index enable flag	Modifies address of data memory when data memory
									(IXE)	manipulation instruction is exeuted.
										0 : Does not modify
										1 : Modifies
								-	Zero flag	Indicates result of arithmetic operation is zero.
									(Z)	Status of this flag differs depending on contents of compare
										flag.
								•	Carry flag	Indicates occurrence of carry or borrow as result of execution
									(CY)	of addition or subtraction instruction.
										This flag is reset to 0 if no carry or borrow occurs.
										It is set to 1 if carry or borrow occurs.
										This flag is also used as shift bit of "RORC r" instruction.
								-	Compare flag	Indicates whether result of arithmetic operation is stored to
									(CMP)	data memory or general register.
										0 : Stores result.
										1 : Does not store result.
								•	BCD flag	Indicates whether arithmetic operation is performed in decimal
									(BCD)	or binary.
										0 : Binary operation
										1 : Decimla operation

## 5.8.3 Cautions on using program status word

When an arithmetic operation (addition or subtraction) is executed to the program status word, the "result" of the arithmetic operation is stored.

For example, even if an operation that generates a carry is executed, if the result of the operation is 0000B, 0000B is stored to the PSW.

## 6. GENERAL REGISTER (GR)

## 6.1 Outline of General Register

Figure 6-1 outlines the general register.

As shown in the figure, the general register is specified in the data memory by the general register pointer. The bank and row address of the general register are specified by the general register pointer. The general register is used to transfer or operate data between data memory addresses.

Figure 6-1. Outline of General Register



Remark Address 5BH is not provided to BANK4 through BANK14.

## 6.2 General Register

The general register consists of 16 nibbles (16  $\times$  4 bis) of the same row address on the data memory.

For the range of the banks and row addresses that can be specified by the general register pointer as a general register, refer to **5.7 General Register Pointer (RP)**.

The 16 nibbles of the same row address specified as a general register operate or transfer data with the data memory by a single instruction.

In other words, operation or data transfer between data memory addresses can be executed by a single instruction.

The general register can be controlled by the data memory manipulation instruction, like the other data memory areas.

## 6.3 Generating Address of General Register by Each Instruction

The following sections 6.3.1 and 6.3.2 explain how the address of the general register is generated when each instruction is executed.

For the details of the operation of each instruction, refer to 7. ALU (Arithmetic Logic Unit) BLOCK.

```
6.3.1 Add ("ADD r, m", "ADDC r, m"),
subtract ("SUB r, m", "SUBC r, m"),
logical operation ("AND r, m", "OR r, m", "XOR r, m"),
direct transfer ("LD r, m", "ST m, r"), and
rotation ("RORC r") instructions
```

Table 6-1 shows the address of the general register specified by operand "r" of an instruction. Operand "r" of an instruction specifies only a column address.

Table 6-1.	Generating	Address of	f General	Register
------------	------------	------------	-----------	----------

		Ba	ink		Rov	v addr	ess	Column address					
	bз	b2	b₁	bo	b2	b₁	b٥	b₃	b2	b1	bo		
General register address	Cor	ntents	of ge	neral	regist	er poii	nter	+	I	r	*		

## 6.3.2 Indirect transfer ("MOV @r, m", "MOV m, @r") instruction

Table 6-2 shows a general register address specified by instruction operand "r" and an indirect transfer address specified by "@r".

 Table 6-2. Generating Address of General Register

	Bank				Rov	v addı	ess	Column address					
	bз	b2	b1	bo	b <sub>2</sub>	b1	bo	bз	b <sub>2</sub>	b1	bo		
General register address	Contents of general register pointer						I	-	•				
Indirect transfer address	-	s data	memo	ory		¢	onten	ts of "	r"				

## 6.4 Cautions on Using General Register

## 6.4.1 Row address of general register

Because the row address of the general register is specified by the general register pointer, the currently specified bank may differ from the bank of the general register.

## 6.4.2 Operation between general register and immediate data

No instruction is available that executes an operation between the general register and immediate data.

To execute an operation between the general register and immediate data, the general register must be treated as a data memory area.

# 7. ALU (Arithmetic Logic Unit) BLOCK

## 7.1 Outline of ALU Block

Figure 7-1 outlines the ALU block.

As shown in the figure, the ALU block consists of an ALU, temporary registers A and B, program status word, decimal adjustment circuit, and memory address control circuit.

The ALU operates on, judges, compares, rotates, and transfers 4-bit data in the data memory.

Data bus Address Temporary Temporary Program status word register B control register A Carry/borrow/zero Index modification detection/decimal/storage memory pointer specification ALU • Arithmetic operation Logical operation • Bit judgment Comparison Rotation Transfer Data memory Decimal adjustment

Figure 7-1. Outline of ALU Block

## 7.2 Configuration and Function of Each Block

## 7.2.1 ALU

The ALU performs arithmetic operation, logical operation, bit judgment, comparison, rotation, and transfer of 4-bit data according to instructions specified by the program.

## 7.2.2 Temporay registers A and B

Temporary registers A and B temporarily store 4-bit data.

These registers are automatically used when an instruction is executed, and cannot be controlled by program.

## 7.2.3 Program status word

The program status word controls the operation of and stores the status of the ALU. For further information on the program status word, also refer to **5.8 Program Status Word (PSWORD)**.

## 7.2.4 Decimal adjustment circuit

The decimal adjustment circuit converts the result of an arithmetic operation into a decimal number if the BCD flag of the program status word is set to "1" during arthmetic operations.

## 7.2.5 Address control circuit

The address control circuit specifies an address of the data memory.

At this time, address modification by the index register and data memory row address pointer is also controlled.

## 7.3 ALU Processing Instruction List

Table 7-1 lists the ALU operations when each instruction is executed.

Table 7-2 shows how data memory addresses are modified by the index register and data memory row address pointer.

Table 7-3 shows decimal adjustment data when a decimal operation is performed.

ALU	Instruction Differe			Difference	in Operation Depen	ding on Program S	Status Word (PSWORD)	Address N	lodification
Function			Value of	Value of	Operation	Operation of	Operation of Z flag	Index	Memory
			BCD flag	CMP flag		CY flag			pointer
Add	ADD	r, m	0	0	Stores result of	Set if carry or	Set if result of operation	Modifies	Does not
		m, #n4		   	binary operation	borrow occurs;	is 0000B; otherwise, reset		modify
	ADDC	r, m	0	, 1	Does not store result	otherwise, reset	Retains status if result of operation		
		m, #n4		   	of binary operation		is 0000B; otherwise, reset		
Subtract	SUB	r, m	1	0	Stores result of		Set if result of operation		
		m, #n4		 	decimal operation		is 0000B; otherwise, reset		
	SUBC	r, m	1	1	Does not store result		Retains status if result of operation		
		m, #n4		   	of decimal operation		is 0000B; otherwise, reset		
Logical	OR	r, m	Don't care	Don't care	Not affected	Retains previous	Retains previous status	Modifies	Does not
operation		m, #n4	(retained)	(retained)		status			modify
	AND	r, m		 					
		m, #n4		 					
	XOR	r, m		 					
		m, #n4		 					
Judge	SKT	m, #n	Don't care	Don't care	Not affected	Retains previous	Retains previous status	Modifies	Does not
	SKF	m, #n	(retained)	(reset)		status			modify
Compare	SKE	m, #n4	Don't care	Don't care	Not affected	Retains previous	Retains previous status	Modifies	Does not
	SKNE	m, #n4	(retained)	(retained)		status			modify
	SKGE	m, #n4		1					
	SKLT	m, #n4		 					
Transfer	LD	r, m	Don't care	Don't care	Not affected	Retains previous	Retains previous status	Modifies	Does not
	ST	m, r	(retained)	(retained)		status			modify
	MOV	m, #n4		1					
		@r, m		 					Modifies
		m, @r		1					
Rotate	RORC	r	Don't care	Don't care	Not affected	Value of bo of	Retains previous status	Does not	Does not
			(retained)	(retained)		general register		modify	modify

Table 7-1.	List of ALU	Processing	Instruction	Operations
------------	-------------	------------	-------------	------------

# Table 7-2. Modification of Data Memory Address and Indirect Transfer Address by IndexRegister and Data Memory Row Address Pointer

IXE	MPE	E General Register Address Specified b							d by	"r"	Data Memory Address Specified by "m"							Indirect Transfer Address Specified by							oy "@	⊉r"								
			Ba	ank			Row	v		Col	umn	ı		Ba	ank		   	Row			Column				Ва	nk		: :	Row	/	1 1 1	Col	umr	1
						ac	ddre	SS	i i	add	ress	6					a	ddre	ess	     	add	ress						ac	dre	ss	1 1 1 1	add	ress	3
		b	3 <b>b</b> 2	b1	bo	b2	b1	b <sub>0</sub>	bз	b <sub>2</sub>	b1	bo	b₃	b2	b1	bo	b2	2 <b>b</b> 1	bo	bз	b <sub>2</sub>	b1	bo	bз	b2	b1	bo	b2	b1	b٥	bз	b2	b1	bo
0	0																1			1											1			
					RP			1		ŗ	r			ΒA	NK		   			m				BANK					mr	i I I	(r)			
									-							-	-			1 1 1			•							-			-	
0	1																   			   										   				
							dittc	2									, , , ,	ditto	)	- - - -						I	MP	, , , ,		1		(	r)	
																								-				     			·			
1	0					1								ΒA	NK					m				_	BAI	NK	_		МR		   			
							dittc	)								Log	gica	al	(	OR			-	•			Log	jical		Ċ	OR			
													-					IX					-	-		IXH	H, I.	XM		<b>_</b>	-	(	r)	
1	1					1											     			1								1			1			
							dittc	<b>)</b>										ditte	С							I	MP	1 1 1				(	r)	
																	   			1								1						

BANK : bank register

IX : index register

IXE : index enable flag

IXH : bits 10 through 8 of index register

IXM : bits 7 through 4 of index register

- IXL : bits 3 through 0 of index register
- m : data memory address indicated by mR, mc
- m<sub>R</sub> : data memory row address (high-order)
- mc : data memory column address (low-order)
- MP : data memory row address pointer
- MPE : memory pointer enable flag
- r : general register column address

RP : general register pointer

( $\times$ ) : contents addressed by  $\times$ 

 $\times\!\!:\,$  direct address such as "m" and "r"

Operation	Hexa	decimal Addition	De	Оре	
Result	CY	Operation result	CY	Operation result	Re
0	0	0000B	0	0000B	
1	0	0001B	0	0001B	
2	0	0010B	0	0010B	
3	0	0011B	0	0011B	
4	0	0100B	0	0100B	
5	0	0101B	0	0101B	
6	0	0110B	0	0110B	
7	0	0111B	0	0111B	
8	0	1000B	0	1000B	
9	0	1001B	0	1001B	
10	0	1010B	1	0000B	
11	0	1011B	1	0001B	
12	0	1100B	1	0010B	
13	0	1101B	1	0011B	
14	0	1110B	1	0100B	
15	0	1111B	1	0101B	
16	1	0000B	1	0110B	-
17	1	0001B	1	0111B	-
18	1	0010B	1	1000B	-
19	1	0011B	1	1001B	-
20	1	0100B	1	1110B	-
21	1	0101B	1	1111B	-
22	1	0110B	1	1100B	-
23	1	0111B	1	1101B	
24	1	1000B	1	1110B	
25	1	1001B	1	1111B	
26	1	1010B	1	1100B	
27	1	1011B	1	1101B	
28	1	1100B	1	1010B	
29	1	1101B	1	1011B	
30	1	1110B	1	1100B	
31	1	1111B	1	1101B	

Table 7-3.	Decimal	Ad	iustment	Data

Operation	Hexa	decimal Addition	De	cimal Addition
Result	CY	Operation result	CY	Operation result
0	0	0000B	0	0000B
1	0	0001B	0	0001B
2	0	0010B	0	0010B
3	0	0011B	0	0011B
4	0	0100B	0	0100B
5	0	0101B	0	0101B
6	0	0110B	0	0110B
7	0	0111B	0	0111B
8	0	1000B	0	1000B
9	0	1001B	0	1001B
10	0	1010B	1	1100B
11	0	1011B	1	1101B
12	0	1100B	1	1110B
13	0	1101B	1	1111B
14	0	1110B	1	1100B
15	0	1111B	1	1101B
-16	1	0000B	1	1110B
-15	1	0001B	1	1111B
-14	1	0010B	1	1100B
-13	1	0011B	1	1101B
-12	1	0100B	1	1110B
-11	1	0101B	1	1111B
-10	1	0110B	1	0000B
-9	1	0111B	1	0001B
-8	1	1000B	1	0010B
-7	1	1001B	1	0011B
-6	1	1010B	1	0100B
-5	1	1011B	1	0101B
-4	1	1100B	1	0110B
-3	1	1101B	1	0111B
-2	1	1110B	1	1000B
-1	1	1111B	1	1001B

**Remark** Decimal adjustment is not correctly carried out in the shaded area in the above table.

## 7.4 Cautions on Using ALU

#### 7.4.1 Cautions on execution operation to program status word

If an arithmetic operation is executed to the program status word, the result of the operation is stored to the program status word.

The CY and Z flags in the program status word are usually set or reset by the result of the arithmetic operation. If an arithmetic operation is executed to the program status word itself, the result of the operation is stored to the program status word, and consequently, it cannot be judged whether a carry or borrow occurs or whether the result of the operation is zero.

If the CMP flag is set, however, the result of the operation is not stored to the program status word. Therefore, the CY and Z flags are set or reset normally.

#### 7.4.2 Cautions on executing decimal operation

The decimal operation can be executed only when the result of the operation falls within the following ranges:

- (1) Result of addition : 0 to 19 in decimal
- (2) Result of subtraction: 0 to 9 or -10 to -1 in decimal

If a decimal operation is executed exceeding or falling below the above ranges, the result is a value greater than 1010B (0AH).

## 8. REGISTER FILE (RF) AND CONTROL REGISTERS

## 8.1 Outline of Register File

Figure 8-1 outlines the register file.

As shown in the figure, the rgister file consists of the control registers existing on addresses 00H through 3FH of BANK15 in the data memory, and a portion overlapping the data memory specified by the BANK register. The control registers set conditions of the peripheral hardware units.

The data on the register file can be read or written via window register.





Window register

## 8.2 Configuration and Function of Register File

Figure 8-2 shows the configuration of the register file and the relationships between the register file and data memory.

The register file is assigned addresses in 4-bit units, like the data memory, and consists of a total of 128 nibbles with row addresses 0H through 7FH and column addresses 0H through 0FH.

Addresses 00H through 3FH are overlapping addresses 00H through 3FH of BANK15 and called control registers that sets the conditions of the peripheral hardware units.

Addresses 40H through 7FH overlap the data memory specified by the BANK register.

In other words, addresses 40H through 7FH of the register file are addresses 40H through 7FH of the currently-selected bank of data memory.

Addresses 40H through 7FH of the register file can be manipulated in the same manner as the data memory, except that the addresses of the register file can also be manipulated by using register file manipulation instructions ("PEEK WR, rf" and "POKE rf, WR"). Note, however, that addresses 40H through 6FH of BANK15 are assigned control register including port input/output selection registers (for details refer to **8.3 Control Registers and Input/Output Selection Registers**).

Addresses 5CH through 6FH of BANK14 are assigned LCD segment register (for details, refer to **8.4 LCD** Segment Register).







Remark Address 5BH is not provided to BANK4 through BANK14.

## 8.2.1 Register file manipulation instructions ("PEEK WR, rf", "POKE rf, WR")

Data is read from or written to the register file via the window register of the system registers, by using the following instructions.

## (1) "PEEK WR, rf"

Reads data of the register file addressed by "rf" to the window register.

## (2) "POKE rf, WR"

Writes the data of the window register to the register file addressed by "rf".

## 8.3 Control Registers and Input/Output Selection Registers

Figure 8-3 shows the configuration of the control registers.

The control registers, including the input/output select registers, consist of a total of 112 nibbles (112 x 4 bits) at addresses 00H through 6FH of BANK15 of the data memory. Of these addresses, 00H through 3FH overlap addresses 00H through 3FH of the register file. Addresses 60H through 6FH are assigned the input/output select registers.

However, only 38 nibbles of the control registers are actually used. The remaining 14 nibbles are unused registers and prohibited from being written or read.

Each control register has an attribute of 1 nibble that identifies four types of registers: read/write (R/W), read-only (R), write-only (W), and read-and-reset (R&Reset) registers.

Nothing is changed even if data is written to a read-only (R and R&Reset) register.

An "undefined" value is read if a write-only (W) register is read.

Among the 4-bit data in 1 nibble, the bit fixed to "0" is always "0" when it is read, and is also "0" when it is written.

The 74 nibbles of unused registers are undefined when their contents are read, and nothing changes even when they are written.

Never write anything to address 31H of BANK15 because this address is a test mode area.

Table 8-1 lists the peripheral hardware control functions of the control registers.

(BANK	15)								
Colum	n Address								
Row Addres	s Item	0	1	2	3	4	5	6	7
0	Name		Stack	Watchdog	Watchdog	Data buffer	Stack overflow/		MOVT bit
(8) <sup>Note1</sup>			pointer	timer clock	timer counter	stack pointer	underflow reset		selection
				selection	reset		selection		
	Symbol		(の	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	W 0 0 0 D T R E S	0 (DBFSP0) 0	0 0 SPRSEL PRSEL 1		20 ►ГТ#04-<0 ОГТ#04-<0
	Read/		R/W	R/W	W & Reset	R	R/W		R/W
	Write								
1	Name	PLL mode	PLL reference	PLL unlock		BEEP clock		Watchdog	Basic timer
(9) <sup>Note1</sup>		selection	frequency selection	FF		selection		timer/stack pointer reset status detection	0 carry
	Symbol	PLLLMD SCN F	PLLRFCK0 PLLRFCK1 PLLRFCK2	0 0 0 P L U U		0 B B B E E E P P O O C S C C K L 1 0		0 0 0 WDTSPRES	0 0 0 B T M 0 C Y
	Read/	R/W	R/W	R & Reset		R/W		R & Reset	R & Reset
	Write								
2	Name		IF counter	IF counter	IF counter	A/D converter	A/D converter		
(A) <sup>Note1</sup>			gate status	mode	control	channel	mode		
			detection	selection		selection	selection		
	Symbol	0 0 0 F G C H H 0 Note 2	0 0 0	IFCCK0 IFCCK1 IFCMD0 1	0 0 I FFC STR T	0 0 A A C C C C H H 1 0	0 0 A A DCC ST P T		
	Read/ Write		R	R/W	W	R/W	R	R/W	R/W
3	Name								
(B) <sup>Note1</sup>			Note 3						
	Symbol								
	Read/ Write								

Figure 8-3. Configuration of Control Registers (Addresses 00H-3FH) (1/4)

Notes 1. ( ) indicates an address that is used when the assembler is used.

2. Never write anything to address 20H of BANK15.

3. Never write anything to address 31H of BANK15 because this address is a test mode area.

×

8	9	А	В	С	D	Е	F
System register							
interrupt stack							
pointer							
(SYSRSP0) (SYSRSP1) (SYSRSP1) (SYSRSP2) (SYSRSP2)							
K							
Basic timer 1				Serial I/O1	Serial I/O1		Interrupt
clock				clock	mode		edge
selection					selection	,	selection
0 0 0 B T 1 K 0				0 0 S S O 1 C K 1 O 1 C K 1 O	0 S S S 0 I 0 0 1 I 1 0 I 5 D Z		0 0 0 I E G 0
R/W				R/W	R/W		R/W
			Timer 0	Timer 0	Interrupt	Interrupt	Interrupt
			counter clock	mode	enable 1	enable 2	enable 3
			selection	selection			
			T M 00C K 0 R E S N S	T 0 0 0 0 0 V F	I I I I I P P P P S S T T I I M M O O 3 2 1 0	I I I I P P P P T T 4 3 M M 1 0	I I I F P P C 2 1 0 C E
			R/W	R/W	R/W	R/W	R/W
				Serial interface 1	Basic interval	Timer 0	INT0 pin
				interrupt	timer 1 interrupt	interrupt	interrupt
			 	request	request	request	request
				0 0 0 0 I R Q S I O 1	0 0 0 0 R Q T M 1	0 0 0 0 R Q T M 0	I 0 0 I N R T Q 0 1
				R/W	R/W	R/W	R/W

Figure 8-3. Configuration of Control Register (Address 00H-3FH) (2/4)

**Note** () indicates an address that is used when the assembler is used.

(BANK Colum	15) n Address								
Row Addres	s Item	0	1	2	3	4	5	6	7
4	Name	LCD driver							
		display start							
	Symbol	L 0 0 C D D D E B N C K							
	Read/	R/W							
	Write								
5	Nathe								
	Symbol								
	Read/								
	Write								
б	Name								
	Symbol								
	Read/								
	vvrite								

# Figure 8-3. Configuration of Control Register (Addresses 40H-6FH) (3/4)

★ Note Bit 3 of address 40H of BANK15 is a test mode area, therefore, do not write "1" to bit 3.

	8	0		P				
	0	9	A	Ь	U		E	Г
		LCD port	Port 0D	Port 2C bit	Port 2B bit	Port 1D bit	Port 1A bit	Port 0B bit
		selection	pull-down resistor	I/O selection	I/O selection	I/O selection	I/O selection	I/O selection
		0 L L L C C C D D 1 1 1 9 8 7 S S S E E E L L L	P P P P P 0 0 0 0 0 D D D D P P P P L L L L L D D D D 3 2 1 0	P P P P 2 2 2 2 2 C C C C C B B B B B I I I I O O O O 3 2 1 0	P P P P 2 2 2 2 2 B B B B B B B B I I I I I I O O O O 3 2 1 0	P P P P 1 1 1 1 D D D B B B I I I I I O O O O 3 2 1 0	P P P P P 1 1 1 1 1 A A A A B B B B I I I I I O O O O 3 2 1 0	P P P P 0 0 0 0 B B B B B B B B I I I I I 0 0 0 0 3 2 1 0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W
4		1	1	1	1	1	1	1

Figure 8-3. Configuration of Control Register (Addresses 40H-6FH) (4/4)

Peripheral	Co	Control Register Peripheral Hardware Control Function								Clock
Hardware	Name	Address	Read/	bз	Function	Set	value	Reset by	WDT	Stop
		(BANK15)	Write	b <sub>2</sub> by Symbol				RESET	& SP	
				b <sub>0</sub>		0	1	pin	reset	
Stack	Stack pointer	01H	R/W	(SP3)				F	F	Retained
				(SP2)						
				(SP1)						
				(SP0)						
	Interrupt stack	08H	R	0				5	5	Retained
	pointer of			(SYSRSP2)						
	system register			(SYSRSP1)						
				(SYSRSP0)						
	Data buffer	04H	R	0	Fixed to "0"			0	0	Retained
	stack pointer			0						
				(DBFSP1)	Detects nesting level	0 0	1 1			
					of data buffer stack	Level 0 Level 1 0 1	Level 2 Level 3 0 1			
	Stack overflow/	05H	R/W	0	Fixed to "0"			3	Retained	Retained
	underflow reset			0						
	selection			SPRSEL1	Selects interrupt stack	Reset	Reset valid			
					overflow/underflow reset	prohibited				
					(can be set only once					
					following power application)					
				SPRSEL0	Selects address stack	-				
					overflow/underflow reset					
					(can be set only once					
					following power application)					
Watchdog	Watchdog timer	02H	R/W	0	Fixed to "0"			3	Retained	Retained
timer	clock selection			0						
				WDTCK1	Selects clock of watchdog timer (can be	0 0 1 Not 4096 S	1 etting 8192			
				WDTCK0	set only once following power application)	used instruction pr 0 1 0	rohibited instruction 1			
	Watchdog timer	03H	W &	WDTRES	Resets watchdog timer counter	Invalid	Reset if written	Undefined	Undefined	Undefined
	counter reset		Reset	0	Fixed to "0"					
				0						
				0						
	WDT&SP reset	16H	R &	0				0	1	Retained
	status detection		Reset	0						
				0						
				WDTSPRES	Detects resetting of watchdog	No reset	Reset request			
					timer/stack pointer	request				

# Table 8-1. Peripheral Hardware Control Functions of Control Registers (1/5)

Peripheral	Co	ntrol Re	gister		Peripheral Hardwar	e Control Function	At	Reset	Clock
Hardware	Name	Address (BANK15)	Read/ Write	b3 b2 b1 Symbol	Function	Set value	Reset by RESET	WDT & SP	Stop
				bo		0 1	pin	reset	
MOVT	MOVT bit selection	07H	R/W	0 0 MOVTSEL1 MOVTSEL0	Fixed to "0" Sets bit transferred by MOVT (transferred to 0EH and 0FH only during 8-bit transfer)	00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	0	Retained
Serial interface	Serial I/O1 clock selection	1CH	R/W	0 0 SIO0CK1 SIO0CK0	Fixed to "0" Sets internal clck of serial interface 1	0 0 1 1 External 125 18.75 37.5 clock kHz kHz kHz (75 kHz) 0 1 0 1	0	0	0
	Serial I/O1 mode selection	1DH	R/W	0 SIO1MOD SIO1HIZ SIO1TS	Fixed to "0" Sets SI1/SO2 pin switching Sets P0B2/SO1 pin status Starts or stops operation	SI1 SO2 General I/O port Serial data output pin Stops operation Starts operation	0	0	0
PLL	PLL mode	10H	R/W	PLLSCNF	Sets low-order bits of swallow counter	LSB is 0 LSB is 1	Undefined	Undefined	Retained
frequency synthesizer	selection			0 PLLMD1 PLLMD0	Fixed to "0"	0 0 1 1 Disabled MF VHF HF 0 1 0 1	0	0	0
	PLL reference frequency selection	11H	R/W	0 PLLRFCK2 PLLRFCK1 PLLRFCK0	Fixed to "0" Sets reference frequency of PLL	0: 1 kHz 1: 3 kHz 2: 5 kHz 3: 6.25 kHz 4: 12.5 kHz 5: 25 kHz 6: Setting prohibited 7: Setting prohibited	7	7	7
	PLL unlcok FF	12H	R & Reset	0  0  0 PLLUL	Fixed to "0"	Locked Unclocked	Undefined	Undefined	Retained
BEEP	BEEP/general- purpose port pin function selection	14H	R/W	0 BEEP0SEL  BEEP0CK1 BEEP0CK0	Fixed to "0" Selects function of P0B3/BEEP pin Sets BEEP pin	General-purpose BEEP I/O port 0 - 0 - 1 1 Low level High level 1.5 kHz 3 kHz output output 0 1 0 1	0	0	0
Timer	Basic timer 0 carry	17H	R & Reset	0 0  0 BTM0CY	Fixed to "0" Detects basic timer 0 carry FF	FF reset FF set	0	Retained	Retained

Table 8-1.	Peripheral	Hardware	Control	Functions of	f Control	Registers (2/5	)
------------	------------	----------	---------	--------------	-----------	----------------	---

Peripheral	Co	ontrol Re	egister		Peripheral Hardwar	e Control Functi	on	At	Reset	Clock
Hardware	Name	Address	Read/	b <sub>3</sub>	Function	Set v	value	Reset by	WDT	Stop
		(BANK15)	Write	b <sub>2</sub> b <sub>1</sub> Symbol				RESET	& SP	
				bo		0	1	pin	reset	
Timer	Basic timer 1	18H	R/W	0	Fixed to "0"			0	0	Retained
	clock selection			0						
				0						
				BTM1CK0	Selects clock of basic timer 0	31.25 Hz (32 ms)	125 Hz (8 ms)			
	Timer 0 counter	2BH	R/W	TMOEN	Starts or stops timer 0 counter	Stops	Starts	0	0	0
	clock selection			TMORES	Resets timer 0 counter	Not affected	Reset			
				TM0CK1	Sets basic clock of timer	0 0 TM0 TM1	1 1 75 kHz 25 kHz			
				TM0CK0	0 counter	0 1	(13.3 μs) (40 μs) 0 1			
	Timer 0 mode	2CH	R/W	TM0OVF	Detects timer 0 overflow	No overflow	Overflow	0	0	0
	selection			TM0GCEG	Sets edge of gate close input	Rising edge	Falling edge			
					signal					
				TM0GOEG	Sets edge of gate open input					
					signal					
				TM0MD	Selects modulo counter/gate	Modulo counter	Gate counter			
					counter of timer 0					
Interrupt	Interrupt edge	1FH	R/W	0	Fixed to "0"			0	0	Retained
	selection			0	-					
				0						
				IEG0	Sets interrupt issuance edge	Rising edge	Falling edge			
					(INT pin)					
	Interrupt enable	2FH	R/W	IPSIO1	Enables serial interface 1	Disables	Enables	0	0	Retained
					interrupt	interrupt	interrupt			
				IPBTM1	Enables basic interface timer 1					
					interrupt 	-				
				IPTM0	Enables timer 0 interrupt	-				
				IP0	Enables INT pin interrupt					
	Serial interface 1	3CH	R/W	0	Fixed to "0"			0	0	Retained
	interrupt request			0	-					
				0						
				IRQSI01	Detects serial interface 1	No interrupt	Interrupt			
					interrupt request	request	request			

# Table 8-1. Peripheral Hardware Control Functions of Control Registers (3/5)

Peripheral	Co	ntrol Re	gister		Peripheral Hardwar	e Control Functi	on	At	Reset	Clock
Hardware	Name	Address	Read/	b <sub>3</sub>	Function	Set v	value	Reset by	WDT	Stop
		(BANK15)	Write	b <sub>2</sub> Symbol				RESET	& SP	
				bo		0	1	pin	reset	
Interrupt	Basic interval	3DH	R/W	0	Fixed to "0"			0	0	Retained
	timer 1			0						
	interrupt request			0						
				IRQBTM1	Detects basic interval timer 1	No interrupt request	Interrupt request			
					interrupt request					
	Timer 0 interrupt	3EH	R/W	0	Fixed to "0"			0	0	Retained
	request			0						
				0						
				IRQTM0	Detects timer 0 interrupt request	No interrupt request	Interrupt request			
	INT0 pin interrupt	3FH	R/W	INT0	Detects INT0 pin status	_ Low level	High level	Undefined	Undefined	Undefined
	request			0	Fixed to "0"			0	0	Retained
				0						
				IRQ0	Detects INT0 pin interrupt request	No interrupt request	Interrupt request			
IF		20H	R/W	0	Fixed to "0"			0	0	0
counter				0						
				0						
				FCGCH0 <sup>Note 1</sup>		Note 1				
	IF counter gate	21H	R	0	Fixed to "0"			0	0	0
	status detection			0						
				0						
				IFCGOSTT	Detects IF counter gate status	Closed	Open			
	IF counter	22H	R/W	IFCMD1	Sets IF counter mode	0 0 1 general-AMIFC F	1 MIFC AMIFC2	0	0	0
	mode selection			IFCMD0		purpose input port	1			
				– – – – – – – – – – – – – – – – – – –	Sets IF counter gate time	$\begin{vmatrix} \frac{0}{2} & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 1 \end{vmatrix}$	1			
				– – – – – ІFCCK0		1ms 4ms 8 0 1 0	ms Open 1			
	IF counter	23H	W	0	Fixed to "0"			0	0	0
	control			0						
				IFCSTRT	Starts or stops IF counter	Nothing affected	Starts counter			
				IFCRES	Resets IF counter data	Nothing affected	Resets counter			
A/D	A/D converter	24H	R/W	0	Fixed to "0"		<u>I</u>	0	0	Retained
converter	channel			0						
	selection			ADCCH1	Selects pin used for A/D converter	0: A/D converter no	ot used			
				ADCCH0		2: P0D2/AD1pin				
						3: P0D3/AD2 pin				

# Table 8-1. Peripheral Hardware Control Functions of Control Registers (4/5)

\*

Peripheral	Cc	ontrol Re	gister		Peripheral Hardwar	e Control Functi	on	At	Reset	Clock
Hardware	Name	Address (BANK15)	Read/ Write	b3 b2 b1 b0	Function	Set v	/alue 1	Power- ON reset	WDT & SP reset	Stop
A/D converter	A/D converter mode selection	25H	R	0	Fixed to "0"			0	0	0
				ADCSTRT	Detects operating status of A/D converter	Conversion ends	Converting			
				ADCCMP	Detects comparison result of A/D converter	Vadcref > Vadcin	Vadcref < Vadcin			Retained
LCD driver	LCD driver display start	40H	R/W	LCDDBCK <sup>Note 2</sup>  0  0	Fixed to "0"	Note 2		0	0	0
				LCDEN	Sets ON/OFF of all the LCD display	Display OFF	Display ON			
	LCD port	69H	R/W	0	Fixed to "0"			0	0	0
	selection			LCD19SEL LCD18SEL	Selects function of P2A2/LCD19 pin Selects function of P2A2/LCD18 pin	General- purpose	LCD Segment	0	0	Retaine
				LCD17SEL	Selects function of P2A2/LCD17 pin	input port				
Input/ output port	Port 0D pull- down resistor selection	6AH	R/W	P0DPLD3 P0DPLD2 P0DPLD1	Selects pull-down resistor of P0D3 pin Selects pull-down resistor of P0D2 pin Selects pull-down resistor of P0D1 pin	Pull-down resistor used	Pull-down resistor not used	0	0	Retaine
	Port 2C bit I/O selection	6BH	R/W	P2CBIO3 P2CBIO2 P2CBIO1 P2CBIO1 P2CBIO0	Selects pull-down resistor of P0D0 pin Selects I/O of port P2C3 Selects I/O of port P2C2 Selects I/O of port P2C1 Selects I/O of port P2C0	Input	Output	0	0	Retaine
	Port 2B bit I/O selection	6CH	R/W	P2BBIO3 P2BBIO2 P2BBIO1 P2BBIO0	Selects I/O of port P2B3 Selects I/O of port P2B2 Selects I/O of port P2B1 Selects I/O of port P2B0	Input	Output	0	0	Retaine
	Port 1D bit I/O selection	6DH	R/W	P1DBIO3 P1DBIO2 P1DBIO1 P1DBIO0	Selects I/O of port P1D3 Selects I/O of port P1D2 Selects I/O of port P1D1 Selects I/O of port P1D0	Input	Output	0	0	Retaine
	Port 1A bit I/O selection	6EH	R/W	P1ABIO3 P1ABIO2 P1ABIO1 P1ABIO1 P1ABIO0	Selects I/O of port P1A3 Selects I/O of port P1A2 Selects I/O of port P1A1 Selects I/O of port P1A0	Input	Output	0	0	Retaine
	Port 0B bit I/O selection	6FH	R/W	P0BBIO3 P0BBIO2 P0BBIO1 P0BBIO0	Selects I/O of port P0B3 Selects I/O of port P0B2 Selects I/O of port P0B1 Selects I/O of port P0B0	Input	Output	0	0	Retaine

## Table 8-1. Peripheral Hardware Control Functions of Control Registers (5/5)

★ Notes 1. Never write anything to address 20H of BANK15.

2. Bit 3 of address 40H of BANK15 is a test mode area, therefore, do not write "1" to bit 3.

\*

## 8.4 LCD Segment Registers

The LCD segment registers consist of a total of 20 nibbles (20 x 4 bits) of addresses 5CH through 6FH of BANK14 of the data memory. For details, refer to **19. LCD CONTROLLER/DRIVER**.

## 8.5 Cautions on Using Control Register

Keep in mind the following points (1) through (4) when using the write-only (W), read-only (R), and unused registers of the control registers (addresses 00H through 6FH of BANK15.

- (1) An "undefined value" is read if a write-only register is read.
- (2) Nothing is affected even if a read-only register is written.
- (3) An "undefined value" is read if an unused register is read. Nor is anything affected if this register is written.
- (4) Never write anything to address 31H of BANK15 because this address is a test mode area.

# 9. DATA BUFFER (DBF)

## 9.1 Outline of Data Buffer

Figure 9-1 outlines the data buffer. The data buffer is located on the data memory and has the following two functions.

- Reads constant data on the program memory (table reference)
- Transfers data with the peripheral hardware units



Figure 9-1. Outline of Data Buffer
# 9.2 Data Buffer

## 9.2.1 Configuration of data buffer

Figure 9-2 shows the configuration of the data buffer.

As shown in the figure, the data buffer consists of a total of 16 bits of addresses 0CH through 0FH of BANK 0 on the data memory.

The 16-bit data is configured with bit 3 of address 0CH as the MSB and bit 0 of address 0FH as the LSB. Because the data buffer is located on the data memory, it can be manipulated by all data memory manipulation instructions.

Figure 9-2. Configuration of Data Buffer



## 9.2.2 Table reference instruction ("MOVT DBF, @AR")

This instruction moves the contents of the program memory addressed by the contents of the address register to the data buffer.

The number of bits transferred by the table reference instruction can be specified by MOVT selection register (address 07H) of the control registers.

When 8-bit data is transferred, it is read to DBF1 and 0.

When the table reference instruction is used, one stack level is used.

All the addresses of the program memory can be referenced by the table reference instruction.

## 9.2.3 Peripheral hardware control instructions ("PUT" and "GET")

The operations of the "PUT" and "GET" instructions are as follows:

## (1) GET DBF, p

Reads the data of a peripheral register addressed by "p" to the data buffer.

#### (2) PUT p, DBF

Sets the data of the data buffer to a peripheral register addressed by "p".

## 9.3 Relationships between Peripheral Hardware and Data Buffer

Table 9-1 shows the relationships between the peripheral hardware and the data buffer.

MEMO

Peripheral Hardware	Peripheral Regis	ster Transfer	ring Data wit	h Data Buffer		
	Name	Symbol	Peripheral	Execution of	I/O	Actual
			address	PUT/GET	bit	bit
				instruction		
A/D converter	A/D converter data register	ADCR	02H	PUT/GET	8	8
Serial interface 1	Presettable shift register 1	SIO1SFR	04H	PUT/GET	8	8
Timer 0	Timer 0 modulo register	тмом	1AH	PUT/GET	8	8
	Timer 0 counter	TM0C	1BH	GET	8	8
Address register	Address register	AR	40H	PUT/GET	16	16
Data buffer stack	DBF stack	DBFSTK	41H	PUT/GET	16	16
PLL frequency synthesizerNote	PLL data register	PLLR	42H	PUT/GET	16	16
Frequency counter	IFC data register	IFC	43H	GET	16	16

## Table 9-1. Relationships between Peripheral Hardware and Data Buffer (1/2)

**Note** The programmable counter of the PLL frequency synthesizer is configured of 17 bits, of which the highorder 16 bits indicate the PLL data register (PLLR) and the low-order bits are allocated to the PLLSCNF flag (the third bit of address 10H).

For details, refer to 16. PLL FREQUENCY SYNTHESIZER.

At R	eset	Clock	Function
Reset by RESET pin	WDT&SP reset	Stop	
0	0	Retained	Sets compare voltage VADCREF of A/D converter
Undefined	Undefined	Retained	Sets serial-out data and reads serial-in data
FF	FF	FF	Sets modulo register value of timer 0
0	0	0	Reads count value of timer 0 counter
0	0	Retained	Transfers data with address register
Undefined	Undefined	Retained	Saves data of data buffer
Undefined	Undefined	Retained	Sets division value (N value) of PLL
0	0	0	Reads count value of frequency counter

Table 9-1.	Relationships	between	Peripheral	Hardware	and	Data	Buffer	(2/2)
------------	---------------	---------	------------	----------	-----	------	--------	-------

# 9.4 Cautions on Using Data Buffer

Keep the following points in mind concerning the unused peripheral addresses, write-only peripheral register (PUT only), and read-only peripheral register (GET only) when transferring data with the peripheral hardware via data buffer.

- An "undefined value" is read if a write-only register is read.
- Nothing is affected even if a read-only register is written.
- An "undefined value" is read if an unused address is read. Nor is anything affected if this address is written.

## **10. DATA BUFFER STACK**

## 10.1 Outline of Data Buffer Stack

Figure 10-1 outlines the data buffer stack.

As shown in the figure, the data buffer stack consists of a data buffer stack pointer and data buffer stack registers.

The data buffer stack saves or restores the contents of the data buffer when the "PUT" or "GET" instruction is executed.

Therefore, the contents of the data buffer can be saved by one instruction when an interrupt is accepted.

Figure 10-1. Outline of Data Buffer Stack



## 10.2 Data Buffer Stack Register

Figure 10-2 shows the configuration of the data buffer stack registers.

As shown in the figure, the data buffer stack registers consist of four 16-bit registers.

The contents of the data buffer are saved by executing the "PUT" instruction, and the saved data is restored by executing the "GET" instruction.

The data buffer contents can be successively saved up to 4 levels.

						C	Data	buff	er							
	C	BF3			DE	BF2			DE	3F1			DE	3F0		
	-					Tr	ansf	er d	ata							- -
	1					$\leq$	16	bits		~					PU	T
Name	Data	u buffe	er sta	ack I	regis	ster										
Symbol	DBF	sтк														
Address	41H															
Bit	b15 b1	4 b13	b12	<b>b</b> 11	<b>b</b> 10	b9	b8	b7	b6	b₅	b4	b₃	b <sub>2</sub>	b1	bo	
Data																-
			1		I I I I		Valid	dat	a	       	     	1	     	     		
																Saves contents of d buffer up to 4 levels

# Figure 10-2. Configuration of Data Buffer Stack Register

## 10.3 Data Buffer Stack Pointer

The data buffer stack pointer detects the multiplexing level of the data buffer stack registers.

When the "PUT" instruction is executed to the data buffer stack, the value of the data buffer stack pointer is incremented by one; when the "GET" instruction is executed, the value of the pointer is decremented by one. The data buffer stack pointer can be only read and cannot be written.

The configuration and function of the data buffer stack pointer are illustrated below.

Name	FI	ag s	symb	ool	Address	Read/Write	
	b3	b <sub>2</sub>	b1	bo			
Data buffer stack pointer	0	0	D	D	04H	R	
		1 1 1	В	в			
		   	F	F			
		1	s	S			
		1	Р	Ρ			
		1	1	<b>0</b>			
				-			
				_		Detects multipl	exing level of data buffer stack
			0	0	Level 0		
			0	1	Level 1		
			1	0	Level 2		
			1	1	Level 3		
					Fixed to "0"		
Reset by RESET pin	0	0	0	0			
₩ ₩DT&SP reset			0	0			
Clock stop			Reta	ained			

## 10.4 Operation of Data Buffer Stack

Figure 10-3 shows the operation of the data buffer stack.

As shown in the figure, when the PUT instruction is executed, the contents of the data buffer are transferred to a data buffer stack register specified by the stack pointer, and the stack pointer is incremented by one.

When the GET instruction is executed, the contents of a data buffer stack register specified by the stack pointer are transferred to the data buffer, and the stack pointer is decremented by one.

Therefore, note that the value of the stack pointer is set to 1 if data has been written once because its initial value is 0, and that the stack pointer is set to 0 when data has been written four times.

Note that when writing (PUT) exceeding four levels, the first data are discarded.

Figure 10-3. Operation of Data Buffer Stack

#### (a) If writing does not exceed level 4



#### (b) If writing exceeds level 4

0	А		А		А		А		E		E		E
1	Undefined		В		В		В		В		В		В
2	Undefined		Undefined		С		С		С		С		С
3	Undefined		Undefined		Undefined		D		D		D		D
P	∱ UT I	ן 901	г	† PUT	г	∳ PUT	г	∱ PU1	г	∳ GE1	Г	GE	Г

## 10.5 Using Data Buffer Stack

A program example is shown below.

**Example** To save the contents of the data buffer and address register by using INT0 interrupt routine (the contents of the data buffer and address register are not automatically saved when an interrupt occurs).

START:

	BR ; Inter NOP NOP NOP	INITIAL rupt vector addre	; Reset address ess ; SI0 ; Basic timer 1 ; TM0
INTINT:	PUT GET PUT	DBFSTK, DBF DBF, AR DBFSTK, DBF	; INT pin interrupt vector address (0004H) ; Saves contents of DBF to first level of data buffer ; stack (DBFSTK) ; Transfers contents of address register (AR) to DBF ; Saves contents of AR to second level of data buffer ; stack
	Pro	cessing B	; INT interrupt processing
	GET PUT GET EI RETI	DBF, DBFSTK AR, DBF DBF, DBFSTK	; Restores second level of data buffer stack to data buffer, ; and restores contents of data buffer to address register ; Restores first level of data buffer stack to data buffer
INITIAL:	SET1 EI	IP0	
LOOP:			
	Pro	cessing A	
END	BR	LOOP	

#### 10.6 Cautions on Using Data Buffer Stack

The contents of the data buffer stack are not automatically saved when an interrupt is accepted, and therefore, must be saved by software.

Even when a bank of the data memory other than BANK0 is specified, the contents of the data buffer (existing in BANK0) can be saved or restored by using the "PUT" and "GET" instructions.

# 11. GENERAL-PURPOSE PORT

The general-purpose ports output high-level, low-level, or floating signals to external circuits, and read high-level or low-level signals from external circuits.

## 11.1 Outline of General-purpose Port

Table 11-1 shows the relationships between each port and port register.

The general-prupose ports are classified into I/O, input, and output ports.

The I/O ports can be set in the input or output mode in 1-bit (1-pin) units. The inut or output mode of each I/O port is specified by the port input/output selection registers (addresses 60H through 6FH) of BANK15.

Port		Pin			Data S	etting Metho	d		
	No.	Symbol	I/O	Port register (data memory)					
				Bank	Address	Symbol	Bi (rese	t symbol erved word)	
Port 0A	N	o pin	Output	BANK0	70H	P0A	bз	-	
	N	o pin					b2		
	19	P0A1					b1	P0A1	
	18	P0A0					bo	P0A0	
Port 0B	31	P0B3	I/O (bit I/O)		71H	P0B	bз	P0B3	
	30	P0B2					b2	P0B2	
	29	P0B1					b1	P0B1	
	28	P0B0					bo	P0B0	
Port 0C	67	P0C3	Output		72H	P0C	bз	P0C3	
	66	P0C2					b <sub>2</sub>	P0C2	
	65	P0C1					b1	P0C1	
	64	P0C0					bo	P0C0	
Port 0D	78	P0D3	Input		73H	P0D	bз	P0D3	
	77	P0D2					b2	P0D2	
	76	P0D1					b1	P0D1	
	75	P0D0					bo	P0D0	

Table 11-1. Relationships between Port (Pin) and Port Register (1/2)

Port		Pin			Data S	etting Metho	d	
	No.	Symbol	I/O		Port regist	er (data men	nory)	
				Bank	Address	Symbol	Bi (rese	t symbol erved word)
Port 1A	23	P1A3	I/O (bit I/O)	BANK1	70H	P1A	bз	P1A3
	22	P1A2					b2	P1A2
	21	P1A1					b1	P1A1
	20	P1A0					bo	P1A0
Port 1C	4	P1C3	Input		72H	P1C	bз	P1C3
	3	P1C2					b2	P1C2
	2	P1C1					b1	P1C1
	1	P1C0					bo	P1C0
Port 1D	27	P1D3	I/O (bit I/O)		73H	P1D	bз	P1D3
	26	P1D2					b2	P1D2
	25	P1D1					b1	P1D1
	24	P1D0					b <sub>0</sub>	P1D0
Port 2A	No	pin	Input	BANK2	70H	P2A	bз	-
	63	P2A2					b2	P2A2
	62	P2A1					b1	P2A1
	61	P2A0					bo	P2A0
Port 2B	17	P2B3	I/O (bit I/O)		71H	P2B	bз	P2B3
	16	P2B2					b <sub>2</sub>	P2B2
	15	P2B1					b1	P2B1
	14	P2B0					bo	P2B0
Port 2C	74	P2C3	I/O		72H	P2C	bз	P2C3
	73	P2C2	(bit I/O)				b2	P2C2
	72	P2C1					b1	P2C1
	71	P2C0					bo	P2C0
_	No	pin	_	BANK3	70H-73H	-	Fixed	d to 0
				BANKISNOte				
				DAINK 15				

Table 11-1.	Relationships	between	Port	(Pin) and	Port	Register	(2/2)
-------------	---------------	---------	------	-----------	------	----------	-------

Note Address 5BH is not provided to BANK4 through BANK14.

# 11.2 General-Purpose I/O Port (P0B, P1A, P1D, P2B, P2C)

## 11.2.1 Configuration of I/O port

The following paragraphs (1) and (2) show the configuration of the I/O ports.

- (1) P0B (P0B3, P0B2, P0B1, P0B0)
  - P2B (P2B3, P2B2, P2B1, P2B0)

P2C (P2C3, P2C2, P2C1, P2C0)



\*

(2) P1A (P1A3, P1A2, P1A1, P1A0) P1D (P1D3, P1D2, P1D1, P1D0)



#### 11.2.2 Using I/O port

The input or output mode of the I/O ports is set by I/O selection register P0B, P1A, P1D, P2B or P2C of the control registers.

Because these are bit I/O ports, they can be set in the input or output mode in 1-bit units.

Setting the output data of or reading the input data of a port is carried out by executing an instruction that writes data to or reads data from the port.

11.2.3 shows the configuration of the I/O selection register of each port.

11.2.4 and 11.2.5 describe how each port is used as an input or output port.

11.2.6 describes the reset status of the I/O ports.

## 11.2.3 I/O port I/O selection register

The following I/O selection registers of the I/O ports are available.

- Port 0B bit I/O selection register
- Port 1A bit I/O selection register
- Port 1D bit I/O selection register
- Port 2B bit I/O selection register
- Port 2C bit I/O selection register

Each I/O selection register sets the input or output mode of the corresponding port pin.

The following paragraphs (1) through (5) describe the configuration and functions of the above I/O selection registers.

## (1) Port 0B bit I/O selection register

Name	Fl	ag s	ymb	ol	Address	Read/Write
	b₃	b <sub>2</sub>	b1	bo		
Port 0B bit I/O selection	Р	Р	Р	Р	(BANK15)	R/W
	0	0	0	0	6FH	
	в	в	В	В		
	в	в	В	В		
	Т	1	I	Т		
	0	0	0	0		
	3	2	1	0		



At r	WDT&SP reset	0	0	0	0
Clo	ock stop		Reta	aineo	b

# (2) Port 1A bit I/O selection register

Name	FI	ag s	ymb	ol	Address	Read/Write				
	bз	b <sub>2</sub>	b1	bo						
Port 1A bit I/O selection	Р	Р	Р	Р	(BANK15)	R/W				
	1	1	1	1	6EH					
	Α	A	A	A						
	В	В	В	В						
	I	1	I	I						
	0	0	0	0						
	3	2	1	0						
				-		Sets in	put/output mode of port			
				0	Sets P1A0 pin	Sets P1A0 pin in input mode				
				1	Sets P1A0 pin in output mode					
						Sets in	out/output mode of port			
			0	1	Sets P1A1 pin	in input mode				
			1	4 1 1 1	Sets P1A1 pin	in output mode				
						Sets in	out/output mode of port			
		0	1   		Sets P1A2 pin in input mode					
		1			Sets P1A2 pin	Sets P1A2 pin in output mode				
				-		Sets in	out/output mode of port			
	0				Sets P1A3 pin	in input mode				
	1				Sets P1A3 pin in output mode					
Reset by RESET pin	0	0	0	0						

eset	Reset by RESET pin	0	0	0	0
At re	WDT&SP reset	0	0	0	0
Clo	ock stop		Reta	ainec	ł

# (3) Port 1D bit I/O selection register

Name	FI	ag s	ymb	ol	Address	Read/Write					
	b₃	b <sub>2</sub>	b1	bo							
Port 1D bit I/O selection	Р	Р	Р	Р	(BANK15)	R/W					
	1	1	1	1	6DH						
	D	D	D	D							
	В	В	В	В							
	Ι										
	0	0	0	0							
	3	2									
				_ <b>_</b>		Sets in	put/output mode of port				
				0	Sets P1D0 pin	in input mode					
				1	Sets P1D0 pin	in output mode					
						Sets in	put/output mode of port				
			0	ן ו ו	Sets P1D1 pin	in input mode					
			1	4       	Sets P1D1 pin	in output mode					
						Sets in	put/output mode of port				
		0	     		Sets P1D2 pin in input mode						
		1	4 1 1 1 1		Sets P1D2 pin	in output mode					
					Sets input/output mode of port						
	0				Sets P1D3 pin	in input mode					
	1				Sets P1D3 pin	in output mode					

eset	Reset by RESET pin	0	0	0	0
At re	WDT&SP reset	0	0	0	0
Clo	ock stop		Reta	aineo	d

# (4) Port 2B bit I/O selection register

Name	Flag symbol		ol	Address	Read/Write				
	bз	b <sub>2</sub>	b1	bo					
Port 2B bit I/O selection	Ρ	Р	Р	Р	(BANK15)	R/W			
	2	2	2	2	6CH				
	В	в	в	в					
	В	В	В	В					
	I	1	1	I					
	0	0	0	0					
	3	2	1	0					
				-		Sets in	put/output mode of port		
				0	Sets P2B0 pin	in input mode			
				1	Sets P2B0 pin	in output mode			
				_		Sets in	put/output mode of port		
			0	1	Sets P2B1 pin	in input mode			
			1	- - - - -	Sets P2B1 pin	in output mode			
						Sets in	out/output mode of port		
		0	1		Sate D2D2 nin in input made				
		1	J		Sets P2B2 pin	In output mode			
				•		Sets in	put/output mode of port		
	0				Sets P2B3 pin	in input mode			
	1				Sets P2B3 pin	in output mode			
	0	0	0	0					

eset	Reset by RESET pin	0	0	0	0
At re	WDT&SP reset	0	0	0	0
Clo	ock stop	l	Reta	inec	ł

## (5) Port 2C bit I/O selection register

Name	FI	ag s	ymb	ol	Address	Read/Write
	bз	b2	b1	bo		
Port 2C bit I/O selection	Р	Р	Ρ	Р	(BANK15)	R/W
	2	2	2	2	6BH	
	С	С	С	С		
	в	в	В	в		
	Т	1	I	1		
	0	0	0	0		
	3	2	1	0		
		·		·		



eset	Reset by RESET pin	0	0	0	0
At re	WDT&SP reset	0	0	0	0
CI	ock stop	I	Reta	inec	ł

## 11.2.4 When using I/O port as input port

The port pin to be set in the input mode is selected by the I/O selection register corresponding to the port. Ports P0B, P1A, P1D, P2B and P2C can be set in the input or output mode in 1-bit units.

The pin set in the input mode is floated (Hi-Z) and waits for input of an external signal.

The input data is read by executing a read instruction (such as SKT) to the port register corresponding to the port pin.

"1" is read from the port register when a high level is input to the corresponding port pin; when a low level is input to the port pin, "0" is read from the register.

When a write instruction (such as MOV) is executed to the port register corresponding to the pin set in the input mode, the contents of the output latch are rewritten.

## 11.2.5 When using I/O port as output port

The port pin to be set in the output mode is selected by the I/O selection register corresponding to the port. Ports P0B, P1A, P1D, P2B and P2C can be set in the input or output mode in 1-bit units.

The pin set in the output mode outputs the contents of the output latch.

The output data is set by executing a write instruction (such as MOV) to the port register corresponding to the port pin.

Write "1" to the port register to output a high level to the port pin; write "0" to output a low level. The port pin can be also floated (Hi-Z) if it is set in the input mode.

If a read instruction (such as SKT) is executed to the port register corresponding to a port pin set in the output mode, the contents of the output latch are read.

#### 11.2.6 Status of I/O port at reset

#### (1) At reset by RESET pin

All the I/O ports are set in the input mode. The contents of the output latch are reset to "0".

#### (2) At WDT&SP reset

All the I/O ports are set in the input mode. The contents of the output latch are reset to "0".

#### (3) On execution of clock stop instruction

The setting of the input or output mode is retained. The contents of the output latch are also retained.

## (4) In halt status

The previous status is retained.

# 11.3 General-Purpose Input Port (P0D, P1C, P2A)

## 11.3.1 Configuration of input port

The following paragraphs (1) and (2) show the configuration of the input port.

## (1) POD (POD3, POD2, POD1, POD0)



(2) P1C (P1C3, P1C2, P1C1, P1C0) P2A (P2A2, P2A1, P2A0)



## 11.3.2 Using input port

The input data is read by executing a read instruction (such as SKT) to the port register corresponding to the port pin.

"1" is read from the port register when a high level is input to the corresponding port pin; when a low level is input to the port pin, "0" is read from the register.

Nothing is affected even if a write instruction (such as MOV) is executed to the port register.

P0D has a pull-down resistor that can be connected or disconnected by software in 1-bit units. The pull-down resistor is connected when "0" is written to the corresponding bit of the port 0D pull-down resistor selection register. When "1" is written to the corresponding bit of this register, the pull-down resistor is disconnected.

## 11.3.3 Port 0D pull-down resistor selection register

• Port 0D pull-down resistor selection register

The port 0D pull-down resistor selection register specifies whether a pull-down resistor is connected to P0D3 through P0D0 pins. The configuration and function of this register are illustrated below.

Name	FI	Flag symbol			Address	Read/Write
	b₃	b2	b1	bo		
Port 0D pull-down resistor	Ρ	Р	Р	Р	(BANK15)	R/W
selection	0	0	0	0	6AH	
	D	D	D	D		
	P	Р	Р	Р		
	L	L	L	L		
	D	D	D	D		
	3	2	1	0		

			Selects pull-down resistor of P0D0 pin								
		C	Connects pull-down resistor to P0D0 pin								
			Disconnects pull-down resistor from P0D0 pin								
			Selects pull-down resistor of P0D1 pin								
	(	) ¦	Connects pull-down resistor to P0D1 pin								
			Disconnects pull-down resistor from P0D1 pin								
			Selects pull-down resistor of P0D2 pin								
	0		Connects pull-down resistor to P0D2 pin								
	1		Disconnects pull-down resistor from P0D2 pin								
			Selects pull-down resistor of P0D3 pin								
			Connects pull-down resistor to P0D3 pin								
0			Disconnects pull-down resistor from P0D3 pin								
0			Disconnects pull-down resistor from P0D3 pin								

eset	Reset by RESET pin	0	0	0	0
At re	WDT&SP reset	0	0	0	0
Clo	ock stop		Reta	aineo	ł

#### 11.3.4 Status of input port at reset

## (1) At reset by RESET pin

All the input ports are set in the input mode. All the pull-down resistors of P0D are connected.

## (2) At WDT&SP reset

All the input ports are set in the input mode. All the pull-down resistors of P0D are connected.

## (3) On execution of clock stop instruction

All the input ports are set in the input mode. The pull-down resistors of P0D retain the previous status.

## (4) In halt status

The previous status is retained.

## 11.4 General-Purpose Output Port (P0A, P0C)

## 11.4.1 Configuration of output port

The configuration of the output port is shown below.

#### (1) P0A (P0A1, P0A0)



(2) POC (POC3, POC2, POC1, POC0)



#### 11.4.2 Using output port

The output port outputs the contents of the output latch to each pin.

The output data is set by executing a write instruction (such as MOV) to the port register corresponding to the port pin.

Write "1" to the port register to output a high level to the port pin; write "0" to output a low level.

However, because P0A is an N-ch open-drain output port, it is floated when it outputs a high level. Therefore, an external pull-up resistor must be connected to this port.

If a read instruction (such as SKT) is executed to the port register, the contents of the output latch are read.

#### 11.4.3 Status of output port at reset

#### (1) At reset by RESET pin

The contents of the output latch are output.

The contents of the output latch are reset to "0".

## (2) At WDT&SP reset

The contents of the output latch are output. The contents of the output latch are reset to "0".

## (3) On execution of clock stop instruction

The contents of the output latch are output. The contents of the output latch are retained.

## (4) In halt status

The contents of the output latch are output. The contents of the output latch are retained.

# **12. INTERRUPT**

# 12.1 Outline of Interrupt Block

Figure 12-1 outlines the interrupt block.

As shown in the figure, the interrupt block temporarily stops the currently executed program and branches execution to a vector address in response to an interrupt request output by a peripheral hardware unit.

The interrupt block consists of an "interrupt request servicing block" corresponding to each peripheral hardware unit, "interrupt enable flip-flop" that enables all interrupts, "stack pointer" that is controlled when an interrupt is accepted, "address stack registers", "program counter", and "interrupt stack".

The "interrupt control block" of each peripheral hardware unit consists of an "interrupt request flag ( $IRQ \times \times \times$ )" that detects the corresponding interrupt request, "interrupt enable flag ( $IP \times \times \times$ )" that enables the interrupt, and "vector address generator (VAG)" that specifies a vector address when the interrupt is accepted.

The  $\mu$ PD17934 has the following three types of maskable interrupts.

- INT interrupts
- Timer 0 and basic interval timer 1 interrupts
- Serial interface 1 interrupts

When an interrupt is accepted, execution branches to a predetermined address, and the interrupt is serviced.



Figure 12-1. Outline of Interrupt Block

## 12.2 Interrupt Control Block

An interrupt control block is provided for each peripheral hardware unit. This block detects issuance of an interrupt request, enables the interrupt, and generates a vector address when the interrupt is accepted.

#### 12.2.1 Configuration and function of interrupt request flag (IRQ×××)

Each interrupt request flag is set to 1 when an interrupt request is issued by the corresponding peripheral hardware unit, and is reset to 0 when the interrupt is accepted.

Writing the interrupt request flag to "1" directly is equivalent to issuance of the interrupt request.

By detecting the interrupt request flag when an interrupt is not enabled, issuance status of each interrupt request can be detected.

Once the interrupt request flag has been set, it is not reset until the corresponding interrupt is accepted, or until "0" is written to the flag via a window register.

Even if two or more interrupt requests are issued at the same time, the interrupt request flag corresponding to the interrupt that has not been accepted is not reset.

Figures 12-2 through 12-15 show the configuration and function of the respective interrupt request registers.

#### Figure 12-2. Configuration of Serial Interface 1 Interrupt Request Register

Name	FI	Flag symbol		ool	Address	Read/Write	
	b3	b2	b1	bo			
Serial interface 1	0	0	0	I	(BANK15)	R/W	1
interrupt request		 	 	R	3CH		
				Q			
			1	S			
			1	1			
			     	0			
		1		1			
				-	Indicat	es interrupt requ	lest issuance status of serial interface 1
				0	Interrupt reque	est not issued	
				1	Interrupt reque	est issued	
				-	Fixed to "0"		
Reset by RESET pin	0	0	0	0			
문 국 WDT&SP reset				0			
Clock stop				R			

R: Retained

Name	Flag symbol			ol	Address	Read/Write		
	bз	b2	b1	bo				
Basic timer 1	0	0	0	I	(BANK15)	R/W		
interrupt request		1	   	R	3DH			
		1	   	Q				
		1	1	В				
		ļ	1	Т				
			   	М				
			<u> </u>	1				
					In	dicates interrupt	request issuance status o	of timer 2
				0	Interrupt reque	est not issued		
				1	Interrupt reque	est issued		
				-	Fixed to "0"			
<u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u></u>					1			
Keset by RESET pin	0	0	0	0				
WDT&SP reset				0				

R

# Figure 12-3. Configuration of Basic Timer 1 Interrupt Request Register

R: Retained

Clock stop

# Figure 12-4. Configuration of Timer 0 Interrupt Request Register

Name	FI	Flag symbol			Address	Read/Write					
	bз	b <sub>2</sub>	b1	bo							
Timer 0	0	0	0	I	(BANK15)	R/W					
interrupt request			1	R	3EH						
				Q							
		- - - - -	   	Т							
		1 1 1	   	М							
			     	0							
				-	In	dicates interrupt	equest issuance status of timer				
				0	Interrupt reque	est not issued					
				1	Interrupt reque	errupt request issued					
				•	Fixed to "0"						
	_			-							
	1 ()	· ()	· ()	· ()							

At reset	Reset by RESET pin	0		0	0	0	
	WDT&SP reset					0	
Clo	ock stop		1 1 1			R	

R: Retained



#### Figure 12-5. Configuration of INT Pin Interrupt Request Register

U: Undefined, R: Retained

## 12.2.2 Function and configuration of interrupt request flag (IP $\times \times \times$ )

Each interrupt request flag enables the interrupt of the corresponding peripheral hardware unit. In order for an interrupt to be accepted, all the following conditions must be satisfied.

- The interrupt must be enabled by the corresponding interrupt request flag.
- The interrupt request must be issued by the corresponding interrupt request flag.
- The EI instruction (which enables all interrupts) must be executed.

The interrupt enable flags are located on the interrupt enable register on the register file. Figures 12-6 shows the configuration of each interrupt enable register.



Figure 12-6. Configuration of Interrupt Enable Register

eset	Reset by RESET pin	0	0	0	0
At re	WDT&SP reset	0	0	0	0
Clo	ock stop	Re	tain	ed	

#### 12.2.3 Vector address generator (VAG)

The vector address generator generates a branch address (vector address) of the program memory corresponding to an interrupt source that has been accepted from the corresponding peripheral hardware. Table 12-1 shows the vector addresses of the respective interrupt sources.

Interrupt Source	Vector Address
INT pin	0004H
Timer 0	0003H
Basic timer 1	0002H
Serial interface 1	0001H

#### Table 12-1. Interrupt Sources and Vector Addresses

# 12.3 Interrupt Stack Register

## 12.3.1 Configuration and function of interrupt stack register

Figure 12-7 shows the configuration of the interrupt stack register.

The interrupt stack register saves the contents of the following system registers (except the address register (AR)) when an interrupt is accepted.

- Window register (WR)
- Bank register (BANK)
- Index register (IX)
- General pointer (RP)
- Program status word (PSWORD)

When an interrupt is accepted and the contents of the above system registers are saved to the interrupt stack, the contents of the above system registers, except the window register, are reset to "0".

The interrupt stack can save the contents of the above system registers at up to four levels.

Therefore, interrupts can be nested up to four levels.

The contents of the interrupt stack register are restored to the system registers when the interrupt return (RETI) instruction is executed.

The contents of the interrupt stack register are undefined at reset by  $\overline{\text{RESET}}$  pin.

The previous contents are retained on execution of the clock stop instruction.

Interrupt stack				Interrupt stack register (INTSK)									
pointer of system register				Name	Window stack WRSK	Bank stack BANKSK	Index stack H IXHSK	Index stack M IXHSK	Index stack L IXHSK	Pointer stack H RPHSK	ointer Pointer ack H stack L PHSK RPLSK		
	В	it			Address		Bit						
bз	b2	bı	bo			b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0
0	s	s	s		► он	Undefined							
	Y	Υ	Y										
	s	s	s		► 1H				INTS	S K 1			
	R	R	R										
	s	s	s		- 2H				INT	SK 2			
	Р	Ρ	Р										
	2	1	0		► 3H					SK3			
					► 4H				INT:	SK4			
L_					► 5H				Unde	efined			

# Figure 12-7. Configuration of Interrupt Stack Register
### 12.3.2 Interrupt stack pointer of system register

The interrupt stack pointer of the system register detects the nesting level of interrupts. The interrupt stack pointer can be only read and cannot be written.

The configuration and function of the interrupt stack pointer are illustrated below.

Name	FI	Flag symbol		Address	Read/Write		
	bз	b <sub>2</sub>	b1	bo			
Interrupt stack pointer of system registers	0	(SYSRSP 2	С S Y S R S P 1	(SYSRSP0	08H	R	
		000011	0 0 1 1 0 0	0 1 0 1 1 0	Use prohibited 4 levels (INTS) 3 levels (INTS) 2 levels (INTS) 1 level (INTSK) 0 level	Detects level of in K1) K2) K3)	nterrupt stack of system registers
	L				Fixed to "0"		

eset	Reset by RESET pin	0		1	0	1
At re	WDT&SP reset			1	0	1
Clo	ock stop		1	Re	taine	ed

### 12.3.3 Interrupt stack operation

Figure 12-8 shows the operation of the interrupt stack.

When nested interrupts exceeding four levels are accepted, since the contents saved first are discarded they therefore must be saved by the program.

#### Figure 12-8. Operation of Interrupt Stack (1/2)



#### (a) Where interrupt nesting level is 4 or less

Figure 12-8. Operation of Interrupt Stack (2/2)



#### (b) Where interrupt nesting level is 5 or more

Caution The system is reset when an interrupt of level 5 is accepted.

However, the ISPRES flag, which resets the non-maskable interrupt if the interrupt stack overflows or underflows, must be set to "1". This flag is "1" after system reset, and can then be written only once.

### 12.4 Stack Pointer, Address Stack Registers, and Program Counter

The address stack registers save a return address when execution returns from an interrupt routine.

The stack pointer specifies the address of an address stack register.

When an interrupt is accepted, the value of the stack pointer is decremented by one, and the value of the program counter at that time is saved to an address stack register specified by the stack pointer.

Next, the interrupt routine is executed. When the interrupt return (RETI) instruction is executed after that, the contents of an address stack register specified by the stack pointer are restored to the program counter, and the value of the stack pointer is incremented by one.

For further information, also refer to 3. ADDRESS STACK (ASK).

### 12.5 Interrupt Enable Flip-Flop (INTE)

The interrupt enable flip-flop enables or disables the four types of maskable interrupts.

When this flip-flop is set, all the interrupts are enabled. When it is reset, all the interrupts are disabled. This flip-flop is set or reset by dedicated instructions EI (to set) and DI (to reset).

The EI instruction sets this flip-flop when the instruction next to EI is executed, and the DI instruction resets the flip-flop while it is being executed.

When an interrupt is accepted, this flip-flop is automatically reset.

This flip-flop is also reset at power-ON reset, at a reset by the RESET pin, at a watchdog timer, overflow or underflow of the stack. The flip-flop retains the previous status on execution of the clock stop instruction.

### 12.6 Accepting Interrupt

#### 12.6.1 Accepting interrupt and priority

The following operations are performed before an interrupt is accepted.

- (1) Each peripheral hardware unit outputs an interrupt request signal to the corresponding interrupt request block if a given interrupt condition (for example, input of the falling signal to the INT pin) is satisfied.
- (2) When each interrupt request block accepts an interrupt request signal from the corresponding peripheral hardware unit, it sets the corresponding interrupt request flag (for example, IRQ0 flag if it is the INT pin that has issued the interrupt request) to "1".
- (3) The interrupt enable flag corresponding to each interrupt request flag (for example, IPO flag if the interrupt request flag is IRQ0) is set to "1" when each interrupt request flag is set to "1", and each interrupt request block outputs "1".
- (4) The signal output by the interrupt request block is ORed with the output of the interrupt enable flip-flop, and an interrupt accept signal is output.

This interrupt enable flip-flop is set to "1" by the EI instruction, and reset to "0" by the DI instruction. If "1" is output by each interrupt request processing block while the interrupt enable flip-flop is set to "1", the interrupt is accepted.

As shown in Figure 12-1, the output of the interrupt enable flip-flop is input to each interrupt request block via an AND circuit when an interrupt is accepted.

The signal input to each interrupt request block causes the interrupt request flag corresponding to each interrupt request flag to be reset to "0" and the vector address corresponding to each interrupt to be output.

If the interrupt request block outputs "1" at this time, the interrupt accept signal is not transferred to the next stage. If two or more interrupt requests are issued at the same time, therefore, the interrupts are accepted according to the priority shown in Table 12-2.

Unless the interrupt request enable flag is set to "1", the corresponding interrupt is not accepted.

Therefore, by resetting the interrupt enable flag to "0", the interrupt with a high hardware priority can be disabled.

Interrupt Source	Priority
INT pin	1
Timer 0	2
Basic timer 1	3
Serial interface 1	4

Table	12-2.	Interrupt	Priority
Tuble		menupe	1 1101109

#### 12.6.2 Timing chart when interrupt is accepted

The timing charts in Figure 12-9 illustrate the operations performed when an interrupt or interrupts are accepted.

Figure 12-9 (1) is the timing chart when one interrupt is accepted.

(a) in (1) is the timing chart where the interrupt request flag is set to "1" after all the others, and (b) is the timing chart where the interrupt enable flag is set to "1" after all the others.

In either case, the interrupt is accepted when the interrupt request flag, interrupt enable-flip flop, and interrupt enable flag all have been set to "1".

If the flag or flip-flop that has been set last is set in the first instruction cycle of the "MOVT DBF, @AR" instruction or by an instruction that satisfies a given skip condition, the interrupt is accepted in the second instruction cycle of the "MOVT DBF, @AR" instruction or after the instruction that is skipped (this instruction is treated as NOP) has been executed.

The interrupt enable flip-flop is set in the instruction cycle next to that in which the El instruction is executed. Therefore, the interrupt is accepted after the instruction next to the El instruction has been executed even when the interrupt request flag is set in the execution cycle of the El instruction.

(2) in Figure 12-9 is the timing chart where two or more interrupts are used.

When two or more interrupts are used, the interrupts are accepted according to the hardware priority if all the interrupt enable flags are set. However, the hardware priority can be changed by setting the interrupt enable flags by the program.

"Instruction cycle" shown in Figure 12-9 is a special cycle in which the interrupt request flag is reset, a vector address is specified, and the contents of the program counter are saved after an interrupt has been accepted. It takes 53.3  $\mu$ s, which is equivalent to one instruction execution time, to be completed.

For details, refer to 12.7 Operations after Interrupt Has Been Accepted.

Figure 12-9. Timing Charts When Interrupt Is Accepted (1/3)

- (1) When one interrupt (e.g., rising of INT pin) is used
  - (a) If there is no interrupt mask time by the interrupt flag (IP $\times\times\times$ )
    - <1> If a normal instruction which is not "MOVT" or an instruction that satisfies a skip condition is executed when interrupt is accepted



<2> If "MOVT" or an instruction that satisfies a skip condition is executed when interrupt is accepted





Figure 12-9. Timing Charts When Interrupt Is Accepted (2/3)

# (b) If interrupt is kept pending by the interrupt enable flag





#### (a) Hardware priority



Figure 12-9. Timing Charts When Interrupt Is Accepted (3/3)

### (b) Software priority

### 12.7 Operations after Interrupt Has Been Accepted

When an interrupt is accepted, the following operations are sequentially performed automatically.

- (1) The interrupt enable flip-flop and the interrupt request flag corresponding to the accepted interrupt request are reset to "0". As a result, the other interrupts are disabled.
- (2) The contents of the stack pointer are decremented by one.
- (3) The contents of the program counter are saved to an address stack register specified by the stack pointer. At this time, the contents of the program counter are the program memory address after the address at which the interrupt has been accepted.

For example, if a branch instruction is executed when the interrupt has been accepted, the contents of the program counter are the branch destination address. If a subroutine call instruction is executed, the contents of the program counter are the call destination address. If the skip condition of a skip instruction is satisfied, the next instruction is executed as NOP and then the interrupt is accepted. Consequently, the contents of the program counter are the address after that of the instruction that is skipped.

- (4) The contents of the system registers (except the address register) are saved to the interrupt stack.
- (5) The contents of the vector address generator corresponding to the interrupt that has been accepted are transferred to the program counter. In other words, execution branches to the interrupt routine.

The operations (1) through (5) above require the time of one special instruction cycle (53.3  $\mu$ s) in which normal instruction execution is not performed.

This instruction cycle is called an "interrupt cycle".

In other words, the time of one instruction cycle (53.3  $\mu$ s) is required after an interrupt has been accepted until execution branches to the corresponding vector address.

#### 12.8 Returning from Interrupt Routine

The interrupt return (RETI) instruction is used to return from an interrupt routine to the processing during which an interrupt was accepted.

When the RETI instruction is executed, the following operations are sequentially performed automatically.

- (1) The contents of an address stack register specified by the stack pointer are restored to the program counter.
- (2) The contents of the interrupt stack are restored to the system registers.
- (3) The contents of the stack pointer are incremented by one.

The operations (1) through (3) above require one instruction cycle (53.3  $\mu$ s) in which the RETI instruction is executed.

The only difference between the RETI instruction and the RET and RETSK instructions, which are subroutine return instructions, is the restoration of the bank register and index register in step (2) above.

### 12.9 External Interrupts (INT pin)

#### 12.9.1 Outline of external interrupts

Figure 12-10 outlines the external interrupts.

As shown in the figure, external interrupt requests are issued at the rising or falling edges of signals input to the INT pin.

Whether an interrupt request is issued at the rising or falling edge of an INT pin is independently specified by the program.

The INT pin is a Schmitt trigger input pin to prevent malfunctioning due to noise. This pin does not accept a pulse input of less than 100 ns.



Figure 12-10. Outline of External Interrupts

#### 12.9.2 Edge detection block

The edge detection block specifies the valid edge (rising or falling edge) of an input signal that issues the interrupt request of INT pin, by using an interrupt edge selection register.

Figure 12-11 shows the configuration and function of the interrupt edge selection register.

### Figure 12-11. Configuration of Interrupt Edge Selection Register



eset	Reset by RESET pin	0	(	)	(	)	0
At re	WDT&SP reset						0
Clo	ock stop					,	R

R: Retained

Caution The external input delays about 100 ns.

Changes in IEG0 Flag	Status of INT Pin	Issuance of Interrupt Request	Status of Interrupt Request Flag
$1 \rightarrow 0$	Low level	Not issued	Retains previous status
(Falling) (Rising)	High level	Issued	Set to "1"
$0 \rightarrow 1$	Low level	Issued	Set to "1"
(Rising) (Falling)	High level	Not issued	Retains previous status

#### Table 12-3. Issuance of Interrupt Request by Changing IEG Flag

#### 12.9.3 Interrupt control block

The signal levels that are input to the INT pin can be detected by using the INT0 flag.

Because INT0 flag is reset independently of interrupts, when the interrupt function is not used the INT pin can be used as a 1-bit input port.

If the interrupts are not enabled, these ports can be used as general-purpose port pins whose rising or falling edge can be detected by reading the corresponding interrupt request flags.

At this time, however, the interrupt request flags are not automatically reset and must be reset by the program. For further information, also refer to **12.2.1 Configuration and function of interrupt request flag (IRQ**×××).

#### 12.10 Internal Interrupts

The following three internal interrupts are available.

- Timer 0
- Basic timer 1
- Serial interface 1

#### 12.10.1 Timer 0 and basic timer 1 interrupts

Interrupt requests are issued at fixed intervals. For details, refer to **13. TIMER**.

#### 12.10.2 Serial interface 1 interrupts

Interrupt requests can be issued at the end of a serial output or serial input operation. For details, refer to **15. SERIAL INTERFACE**.

### 13. TIMERS

Timers are used to manage the program execution time.

### 13.1 Outline of Timers

Figure 13-1 outlines the timers. The following three timers are available.

- Basic timer 0, 1
- Timer 0

Basic timer 0 and 1 detect the status of a flip-flop that is set at fixed time intervals in software. Timer 0 is a modulo timer and can use interrupts.

The clock of each timer is created by dividing the system clock (75 kHz).



### (1) Basic timer 0



(2) Basic timer 0



#### (3) Timer 0



\*

### 13.2 Basic Timer 0

#### 13.2.1 Outline of basic timer 0

Figure 13-2 outlines basic timer 0.

Basic timer 0 is used as a timer by detecting in software the BTM0CY flag that is set at fixed intervals (125 ms).

If the BTM0CY flag is read first after reset by RESET pin, "1" is always read. After that, the flag is set to "0" at 125 ms intervals.

#### Figure 13-2. Outline of Basic Timer 0



**Remark** BTM0CY (bit 0 of basic timer 0 carry register: refer to **Figure 13-3**) detects the status of the flip-flop.

# NEC

### 13.2.2 Flip-flop and BTM0CY flag

The flip-flop is set at fixed intervals (125 ms) and its status is detected by the BTM0CY flag of the basic timer 0 carry register.

When the BTM0CY flag is read, it is reset to "0" (Read & Reset).

The BTM0CY flag is always set to "1" at reset by  $\overline{\text{RESET}}$  pin instruction.

Figure 13-3 shows the configuration of the basic timer 0 carry register.

### Figure 13-3. Configuration of Basic Timer 0 Carry Register

Name	FI	ag s	ymb	ol	Address	Read/Write	]			
	b₃	b <sub>2</sub>	b1	bo						
Basic timer 0 carry	0	0	0	В	(BANK15)	R & Reset	1			
			     	т	17H					
				м						
		1	   	0						
		1	   	С						
				Y						
							-			
				-		Dete	ects stat	us of flip-fl	ор	
				0	Flip-flop is not	set				
				1	Flip-flop is set					
					Fixed to "0"					

eset	Reset by RESET pin	0	0	0	1
At re	WDT&SP reset				R
Clo	ock stop				R

R: Retained

### 13.2.3 Example of using basic timer 0

An example of a program using basic timer 0 is shown below. This program executes processing A every 1 second.

#### Example

	LOOP:		
	SKT1	BTM0CY	; Branches to NEXT if BTM0CY flag is "0"
	BR	NEXT	
	ADD	M1, #1	; Adds 1 to M1
	SKE	M1, #08H	; Executes processing A if M1 is "8" (1 second has elapsed)
	BR	NEXT	
	MOV	M1, #0	
	Proc	essing A	
NEX	T:		
	Proc	essing B	; Executes processing B and branches to LOOP
	BR	LOOP	

#### 13.2.4 Errors of basic timer 0

Errors of basic timer 0 include an error due to the detection time of the BTM0CY flag.

#### Error due to detection time of BTM0CY flag

The time to detect the BTM0CY flag must be shorter than the time at which the BTM0CY flag is set. Where the time interval at which the BTM0CY flag is detected is tCHECK and the time interval at which the flag is set is tset (125 ms), tCHECK and tset must relate as follows.

tcheck < tset

At this time, the error of the timer when the BTM0CY flag is detected is as follows, as shown in Figure 13-4.

0 < Error < tSET





As shown in Figure 13-4, the timer is updated because BTM0CY flag is "1" when it is detected in step <2>.

When the flag is detected next in step <3>, it is "0". Therefore, the timer is not updated until the flag is detected again in <4>.

This means that the timer is extended by the time of tCHECK3.

### 13.3 Basic Timer 1

#### 13.3.1 General

Figure 13-5 outlines the basic timer 1.

The basic timer 1 issues an interrupt request at fixed time interval and sets the IRQBTM1 flag to 1.

The time interval of the IRQBTM1 flag is set by the BTM1CK0 flag of the basic timer 1 clock select register. Figure 13-6 shows the configuration of the basic timer 1 clock select register.

The interrupt generated by the basic timer 1 is accepted when the IRQBTM1 flag is set, if the EI instruction has been issued and the IPBTM1 flag has been set (refer to **12. INTERRUPT**).

Figure 13-5. Outline of Basic Timer 1



**Remark** BTM1CK0 (bit 1 of interrupt edge select register. Refer to **Figure 13-6**) set the time interval at which the IRQBTM1 flag is set.

#### 13.3.2 Clock selection block

The clock selection block divides the system clock (75 kHz) and sets the time interval at which the IRQBTM1 flag is to be set, by using the BTM1CK0 flags.

Figure 13-6 shows the configuration of the basic timer 0 clock selection register.

### Figure 13-6. Configuration of Basic Timer 1 Clock Selection Register

b1	b			
0	В	(BANK15)	R/W	
	т	18H		
	М			
   	1			
   	С			
	к			
	0			
			Sets time interv	I at which IRQBTM1 flag is set
	0	32 ms (31.25	Hz)	
	1	8 ms (125 Hz)		
		Fixed to "0"		
		С К 0 1	C K 0 32 ms (31.25 1 8 ms (125 Hz) Fixed to "0"	C K 0 Sets time interva 0 32 ms (31.25 Hz) 1 8 ms (125 Hz) Fixed to "0"

eset	Reset by RESET pin	0	0	0	0
At re	WDT&SP reset				0
Clo	ock stop	•		 ,	R

R : Retained

### 13.3.3 Application example of basic timer 1

A program example is shown below.

### Example

M1	MEM	0.10H	; 80-ms counter
BTIME	R1 DAT	0002H	; Symbol definition of basic timer 1 interrupt vector address
	BR	START	; Branches to START
ORG	BTIMER1		; Program address (0002H)
	ADD	M1, #0001B	; Adds 1 to M1
	SKT1	CY	; Tests CY flag
	BR	EI_RETI	; Returns if no carry
	MOV	M1, #0110B	
	Process	sing A	
EI_RE	TI:		
	EI		
	RETI		
START	Γ:		
	MOV	M1, #0110B	; Initializes contents of M1 to 6
	BANK1		
	SET1	BTM1CK	; Embedded macro
			; Sets basic timer 1 interrupt pulse to 8 ms
	SET1	IPBTM1	: Enables basic timer 1 interrupt
	EI		: Enables all interrupts
LOOP:			· •
	BANK0		
	Process	sing B	
	BR	LOOP	

This program executes processing A every 80 ms.

The points to be noted in this case are that the DI status is automatically set when an interrupt has been accepted, and that the IRQBTM1 flag is set to 1 even in the DI status.

This means that the interrupt is accepted even if execution exits from an interrupt service routine by execution of the "RETI" instruction, if processing A takes longer than 8 ms.

Consequently, processing B is not executed.

#### 13.3.4 Error of basic timer 1

As described in 13.3.3, the interrupt generated by basic timer 1 is accepted each time the basic timer 1 interrupt pulse falls, if the EI instruction has been executed, and if the interrupt has been enabled.

Therefore, an error of basic timer 1 occurs only when any of the following operations are performed:

- · When the first interrupt after basic timer 1 interrupt has been enabled has been accepted
- When the time interval at which the IRQBTM1 flag is to be set is changed, i.e., when the first interrupt is accepted after the interrupt pulse has been changed
- When data has been written to the IRQBTM1 flag

Figure 13-7 shows an error in each of the above operations.

#### Figure 13-7. Error of Basic Timer 1 (1/2)



#### (a) When interrupt by basic timer 1 is enabled

At point <1> in the above figure, the interrupt by basic timer 1 is accepted as soon as the interrupt is enabled.

At this time, the error is -tset.

If an interrupt is enabled by the "EI" instruction at the next point <3>, the interrupt occurs at the falling edge of the basic timer 1 interrupt pulse.

At this time, the error is:

 $-t_{\text{SET}} < error < 0$ 

#### Figure 13-7. Error of Basic Timer 1 (2/2)



#### (b) When basic timer 1 interrupt pulse is changed

Even if the basic timer 1 interrupt pulse is changed to B at point <1> in the above figure, the interrupt is accepted at the next point <2> because the basic timer 1 interrupt pulse does not fall. If the basic timer 1 interrupt pulse is changed to A at <3>, the interrupt is immediately accepted because the basic timer 1 interrupt pulse falls.



#### (c) When IRQBTM1 flag is manipulated

The interrupt is immediately accepted if the IRQBTM1 flag is set to 1 at <1>. If clearing the IRQBTM1 flag to 0 overlaps with the falling of the basic timer 1 interrupt pulse at <2>, the interrupt is not accepted.

#### 13.3.5 Notes on using basic timer 1

When creating a program, such as a program for watch, in which processing is always performed at fixed time intervals by using the basic timer 1 after the supply voltage has been once applied (power-ON reset), the basic timer 1 interrupt service must be completed in a fixed time.

Let's take the following example:

#### Example

M1	MEM	0.10H	; 80-ms counter
BTIMER1	DAT	0002H	; Symbol definition of interrupt vector address of basic timer 1
	BR	START	; Branches to START
ORG	BTIMER1		; Program address (0002H)
	ADD	M1, #0001B	; Adds 1 to M1
	SKT1	CY	; Watch processing if carry occurs
	BR	EI_RETI	; Restores if no carry occurs
	MOV	M1, #0110B	
; <1	>		
	Process	ing B	
EI_RETI:			
	EI		
	RETI		
START:			
	MOV BANK1	M1, #0110B	; Initializes contetns of M1 to 6
	SET1	BTM1CK	
			; Embedded macro
			; Sets time of interrupt by basic timer 1 to 8 ms
	SET1	IPBTM1	; Embedded macro
			; Enables interrupt by basic timer 1
	EI		; Enables all interrupts
LOOP:			
	Process	ing A	
	BR	LOOP	

In this example, processing B is executed every 80 ms while processing A is executed.

### 13.4 Timer 0

#### 13.4.1 Outline of timer 0

Figure 13-8 outlines timer 0.

Timer 0 counts the basic clock (75, 25, or external clock (TM0, TM1)) with an 8-bit counter, and compares the count value with a value set in advance.

#### Figure 13-8. Outline of Timer 0



- **Remarks 1.** TM0CK1 and TM0CK0 (bits 1 and 0 of timer 0 counter clock selection register: refer to **Figure 13-9**) set the basic clock frequency.
  - 2. TM0EN (bit 3 of timer 0 counter clock selection register: refer to Figure 13-9) starts or stops timer 0.
  - 3. TMORES (bit 2 of timer 0 counter clock selection register: refer to Figure 13-9) resets timer 0 counter.

### 13.4.2 Clock selection and start/stop control blocks

The clock selection block selects a basic clock to operate timer 0 counter. Four types of basic clocks can be selected by using the TM0CK1 and TM0CK0 flags. The start/stop block starts or stops the basic clock input to timer 0 by using the TM0EN flag. Figure 13-9 shows the configuration and function of each flag.

#### 13.4.3 Count block

The count block counts the basic clock with timer 0 counter, reads the count value, and issues an interrupt request when its count value coincides with the value of the timer 0 modulo register.

The timer 0 counter can be reset by the TM0RES flag.

The timer 0 counter is automatically reset when its value coincides with the value of the timer 0 modulo register.

The value of the timer 0 counter can be read via data buffer.

Data can be written to the value of the timer 0 modulo register via data buffer.

Figure 13-9 shows the configuration of timer 0 counter clock selection register.

Figure 13-10 shows the configuration of the timer 0 counter.

Figure 13-11 shows the configuration of the timer 0 modulo register.

Name	FI	ag s	ymb	ol	Address	Read/Write	
	bз	b2	b1	bo			
Timer 0 counter clock selection	т	т	т	Т	(BANK15)	R/W	
	м	М	М	М	2BH		
	0	0	0	0			
	E	R	с	с			
	N	Е	к	к			
		s	1	0			
							1
				-		Sets bas	ic clock of timer 0 counter
			0	0	TM0		
			0	1	TM1		
			1	0	75 kHz (13.3 μ	es)	
			1	1	25 kHz (40 <i>μ</i> s)	)	
				-			
				->		Resets time	er 0 counter (valid on writing)
		0	ר י י		Does not chan	ge	
		1	4 1 1		Resets counte	r	
			J				
						Sta	arts or stops timer 0
	0	ר י י			Stops		
	1	4 1 1			Starts		
		,					
g Reset by RESET pin	0	0	0	0			

### Figure 13-9. Configuration of Timer 0 Counter Clock Selection Register

ese	Reset by RESET pin	0	0	¦ C	)	0	
At re	WDT&SP reset						
Clo	ock stop						

Caution When the TM0RES flag is read, 0 is always read.

	Data	buffer		
DBF3	DBF2	DBF1	DBF0	
Don't care	Don't care	Transf	er data	
		81	pits	GET PUT must not be executed
	Name	Timer 0 counte	er	
	Symbol	TM0C		
	Address	1BH		
	Bit	b7 b6 b5 b4	b3 b2 b1 b0	
	Data	Valic	I data	
				Reads count value of timer 0
			n	
		`	- 	
			×	Count value
		I I I		
		0F	FH	
Reset by	RESET pin	0 0 0 0	0 0 0 0	

# Figure 13-10. Configuration of Timer 0 Counter

eset	Reset by RESET pin	0	0	0	0	0	0	0	0
At re	WDT&SP reset	0	0	0	0	0	0	0	0
Cl	ock stop	0	0	0	0	0	0	0	0

	Data	buffer				
DBF3	DBF2	DBF1		DBF0		
Don't care	Don't care	- Tra	ansfer d	ata	-	
			8 bits			GET PUT
	Name	Timer 0 mo	odulo re	egister		
	Symbol	тмом				
	Address	1AH				
	Bit	b7 b6 b5	b4 b3	b2 b	bo	
	Data	V	/alid dat	ta		
						1
					•	Sets modulo data of timer 0
			0			Setting prohibited
			1			
			x			Modulo counter value
			0FFH			

Figure 13-11.	Configuration	of Timer	0 Modulo	Register
---------------	---------------	----------	----------	----------

eset	Reset by RESET pin	1	1	1	1	1	1	1	1
At r	WDT&SP reset	1	1	1	1	1	1	1	1
Clo	ock stop	1	1	1	1	1	1	1	¦ 1

### 13.4.4 Example of using timer 0

### (1) Modulo timer

The modulo timer is used for time management by generating timer 0 interrupt at fixed intervals. An example of a program is shown below.

This program executes processing B every 400  $\mu$ s.

TMODATA	DAT	0009H	; Count data = 10
START:	BR ; Interrupt v NOP NOP BR NOP	INITIAL vector address INT_TM0	; Reset address ; SIO1 ; BTM1 ; TM0 ; INT
INITIAL:	INITFLG ; MOV MOV PUT SET1 SET1 EI	NOT TM0EN, TM0RES (Stop) , (Reset DBF0, #TM0DATA DBF1, #TM0DATA SH TM0, DBF TM0EN IPTM0	S, TM0CK1, TM0CK0 ), (Basic clock = 40 μs) R4 AND 0FH ; START ; Enables timer 0 interrupt
LOOP:	Processi	ng A	
	BR	LOOP	
INT_TM0:	PUT Processi GET FI	DBFSTK, DBF ng B DBF, DBFSTK	; Saves data buffer
	RETI		; Return

END

#### 13.4.5 Error of timer 0

Timer 0 has an error of up to 1 basic clock in the following cases.

### (1) On starting/stopping counter

The counter is started or stopped by setting the TM0EN flag. Therefore, an error of 0 to +1 clocks occurs when the TM0EN flag is set, and an error of -1 to 0 clocks occurs when the flag is reset.

In all, an error of  $\pm 1$  count occurs.

#### (2) On resetting counter operation

An error of 0 to +1 clocks occurs when the counter is reset.

### (3) On selecting basic clock during counter operation

An error of 0 to +1 clocks of the newly selected clock occurs.

### 14. A/D CONVERTER

### 14.1 Outline of A/D Converter

Figure 14-1 outlines the A/D converter.

The A/D converter compares the analog voltages input to the AD2 through AD0 pins with the internal compare voltage and converts them into 4-bit digital signals by judging the result of the comparison by software. The result of the comparison is detected by the ADCCMP flag.

The result of the comparison is detected by the ADCCMP ha

This converter is of successive approximation type.



Figure 14-1. Outline of A/D Converter

- Remarks 1. ADCCH1 and ADCCH0 (bits 1 and 0 of A/D converter channel selection register: refer to Figure 14-3) select pins used for the A/D converter.
  - ADCCMP (bit 0 of A/D converter mode selection register: refer to Figure 14-5) detects the result of comparison.
  - **3.** ADCSTRT (bit 1 of A/D converter mode selection register: refer to **Figure 14-5**) detects the operating status.

### 14.2 Input Selection Block

Figure 14-2 shows the configuration of the input selection block.

The input selection block selects a pin to be used by using the ADCCH1 and ADCCH0 flags. Only one pin can be used for the A/D converter. When one of the P0D1/AD0 through P0D3/AD2 pins is selected, the other two pins are forcibly set in the input port mode.

The P0D1/AD0 through P0D3/AD2 pins can be connected to a pull-down resistor if so specified by the P0DPL1 through P0DPLD3 flags. To use the P0D1/AD0 through P0D3/AD2 pins for the A/D converter, therefore, disconnect their pull-down resistors to correctly detect an external input analog voltage. (For details, refer to **11.3.3 Port 0D pull-down resistor selection register.**)

Figure 14-3 shows the configuration of the A/D converter channel selection register.





# Figure 14-3. Configuration of A/D Converter Channel Selection Register

Name	FI	ag s	ymb	ol	Address	Read/Write				
	bз	b <sub>2</sub>	b1	bo						
A/D converter channel selection	0	0	Α	Α	(BANK15)	R/W				
			D	D	24H					
			С	С						
			с	С						
			н	н						
			1	0						
				-		Selects p	pin used for A/D converter			
			0	0	A/D converter not used (general-purpose input port)					
			0	1	P0D1/AD0 pin	P0D1/AD0 pin				
			1	0	P0D2/AD1 pin					
			1	1	P0D3/AD2 pin					
				-	Fixed to "0"					

eset	Reset by RESET pin	(	)	0	0	0
At re	WDT&SP reset			0	0	0
Clo	ock stop	,	,	Re	etain	ed

### 14.3 Compare Voltage Generation and Compare Blocks

Figure 14-4 shows the configuration of the compare voltage generation block and compare block.

The compare voltage generation block switches a tap decoder according to the 8-bit data set to the A/D converter reference voltage setting register and generates 256 different of compare voltages VADCREF. In other words, this block is an R-string D/A converter.

The supply voltage to this R-string D/A converter is the same as the supply voltage V<sub>DD</sub> of the device. The compare block compares voltage V<sub>ADCIN</sub> input from a pin with compare voltage V<sub>ADCREF</sub>.

Comparison by the comparator is performed as soon as data has been written to the ADCSTRT flag. It takes the A/D converter the time of executing two instructions (106.6  $\mu$ s) to perform comparison once.

The current operating status of the comparator can be checked by reading the content of the ADCSTRT flag. The result of the comparison is detected by the ADCCMP flag.

Figure 14-5 shows the configuration of A/D converter mode selection register.

#### Figure 14-4. Configuration of Compare Voltage Generation and Compare Blocks



#### Name Flag symbol Address Read/Write b3 b2 b1 b0 A/D converter mode selection 0 0 A A (BANK15) R/W D D 25H с с s c т М R P Т Detects result of comparison by A/D converter 0 VADCIN < VADCREF 1 $V_{ADCIN} > V_{ADCREF}$ Detects operating status of A/D converter 0 End of conversion 1 Conversion in progress Fixed to "0"

Figure 14-5. Configuration of A/D Converter Mode Selection Regist	Figure 14-5.	/D Converter Mode Selection Register
---	--------------	--------------------------------------

At reset	Reset by RESET pin	0	0	0	0
	WDT&SP reser			0	0
Clock stop				0	R

R:Retained
## 14.4 Comparison Timing Chart

The ADCSTRT flag is reset to 0 two instructions after the ADCSTRT flag has been set. At this point, the compare result (ADCCMP flag) can be read.

Figure 14-6 shows the timing chart.





## 14.5 Using A/D Converter

## 14.5.1 Comparing with one compare voltage

An example of a program in this mode is shown below.

**Example** To compare input voltage VADCIN of AD0 pin with compare voltage VADCREF (127.5/256 VDD), and branch to AAA if VADCIN < VADCREF, or to BBB if VADCIN > VADCREF

ADCR7 ADCR6 ADCR5 ADCR4 ADCR3 ADCR2 ADCR1	FLG FLG FLG FLG FLG FLG FLG	0.0EH.3 ; Defines each bit of DBF as ADCR 0.0EH.2 0.0EH.1 0.0EH.0 0.0FH.3 0.0FH.2 0.0FH.1	data setting flag.
ADCRU	FLG	0.0FH.0	
BANK15 INITFLG	NOT P0DPLD3, NOT	P0DPLD2, P0DPLD1, NOT P0DPLD0	; Disconnects pull-down resistor of P0D1 pin.
INITFLG INITFLG INITFLG	NOT ADCCH1, ADCO ADCR7, NOT ADCR6 NOT ADCR3, NOT A	CH0 5, NOT ADCR5, NOT ADCR4 DCR2, NOT ADCR1, NOT ADCR0	; Selects AD0 pin for A/D converter. ;
PUT SET1 NOP NOP	ADCR, DBF ADCSTRT		; Sets compare voltage VADCREF. ; Starts A/D conversion. ; Waits for duration of two instructions.
SKT1 BR BR	ADCCMP AAA BBB		; Judges result of comparison.

### 14.5.2 Successive comparison by means of binary search

The A/D converter can compare only one compare voltage at a time.

Consequently, successive comparison must be executed through program in order to convert input voltages into digital signals.

If the processing time of the successive comparison program is different depending on the input voltage, it is not desirable because of the relations with the other programs.

Therefore, the binary search method described in (1) through (3) below is useful.

### (1) Concept of binary search

The following figure illustrates the concept of binary search.

First, the compare voltage is set to 1/2Vbb. If the result of comparison is True (high level), a voltage of 1/4Vbb is applied; if the result is False (low level), a voltage of 1/4Vbb is subtracted for comparison. Similarly, comparison is performed in sequence from 1/8Vbb to 1/16Vbb. If the result is False after comparison has been executed four times, 1/16Vbb is subtracted, and the comparison ends.



(2) Flowchart of binary search



# (3) Program example of binary search

STAF	RT:		
	BANK1		
	INITFLG	NOT ADCCH1, ADCCH0	; Selects AD0 pin.
	INITFLG	P0DPLD1	; Sets pull-down resistor of AD0 pin OFF.
	INITFLG	NOT ADCRFSEL3, ADCRFSEL2, ADCRFSEL1, ADRFSEL0	; Sets compare voltage to 7.5/16 $V_{\text{DD}}.$
	SET1	ADCSTRT	; A/D comparator starts operating.
	NOP		; 2 wait
	NOP		;
	SKF1	ADCCMP	; Detects ADCCMP.
	SET1	ADCRFSEL3	; If 0, adds 7.5/16 $V_{\text{DD}}$ and,
	CLR1	ADCRFSEL2	; subtracts 3.5/16 VDD.
	SET1	ADCSTRT	; A/D comparator starts operating.
	NOP		; 2 wait
	NOP		;
	SKF1	ADCCMP	; Detects ADCCMP.
	SET1	ADCRFSEL2	; If 0, adds 3.5/16 $V_{\text{DD}}$ and,
	CLR1	ADCRFSEL1	; subtracts 1.5/16 VDD.
	SET1	ADCSTRT	; A/D comparator starts operating.
	NOP		; 2 wait
	NOP		;
	SKF1	ADCCMP	; Detects ADCCMP.
	SET1	ADCRFSEL1	; If 0, adds 1.5/16 $V_{\text{DD}}$ and,
	CLR1	ADCRFSEL0	; subtracts 0.5/16 V <sub>DD</sub> .
	SET1	ADCSTRT	; A/D comparator starts operating.
	NOP		; 2 wait
	NOP		;
	SKF1	ADCCMP	; Detects ADCCMP.
	SET1	ADCRFSEL0	; If 0, adds 0.5/16 V <sub>DD</sub> .

END:

## 14.6 Cautions on Using A/D Converter

## 14.6.1 Cautions on selecting A/D converter pin

When one of the P0D1/AD0 through P0D3/AD2 pins is selected, the other two pins are forcibly set in the input port mode. The P0D1/AD0 through P0D3/AD2 pins can be connected to a pull-down resistor if so specified by the P0DPL1 through P0DPLD3 flags in bank 15. To use the P0D1/AD0 through P0D3/AD2 pins for the A/D converter, therefore, disconnect their pull-down resistors to correctly detect an external input analog voltage.

## 14.7 Status at Reset

## 14.7.1 At reset by RESET pin

All the P0D1/AD0 through P0D3/AD2 pins are set in the general-purpose input port mode. The P0D1 through P0D3 pins are connected with a pull-down resistor.

### 14.7.2 At WDT&SP reset

All the P0D1/AD0 through P0D3/AD2 pins are set in the general-purpose input port mode. The P0D1 through P0D3 pins are connected with a pull-down resistor.

## 14.7.3 On execution of clock stop instruction

The status of the pin selected for the A/D converter is retained as is. The previous status of the pull-down resistor of the P0D1 through P0D3 pins is retained.

## 14.7.4 In halt status

The status of the pin selected for the A/D converter is retained as is. The previous status of the pull-down resistor of the P0D1 through P0D3 pins is retained.

## **15. SERIAL INTERFACE**

### 15.1 General

Figure 15-1 shows the outline of the serial interface.

This serial interface is of two-wire/three-wire serial I/O type. The former type uses SCK and SO2/SI1 pins. The latter uses SCK, SI1, and SO1 pins.

#### Figure 15-1. Outline of Serial Interface



**Remarks 1.** SIO1CK1 and 0 (bits 0 and 1 of serial I/O clock select register. Refer to **Figure 15-2**) set a shift clock.

- 2. SIO1TS (bit 0 of serial I/O mode select register. Refer to Figure 15-3) starts/stops communication.
- SIO1HIZ (bit 1 of serial I/O mode select register. Refer to Figure 15-3) sets the function of the SO1/P0B2 pin.
- SIO1MOD (bit 3 of serial I/O mode select register. Refer to Figure 15-3) selects I/O of SO2/ SI1/P0B1 pin.

## 15.2 Clock Input/Output Control Block and Data Input/Output Control Block

The clock input/output control block and data input/output control block select the operation mode of the serial interface (2-wire or 3-wire mode), control the transmit/receive operation, and select a shift clock.

The flags that control these blocks are allocated to the serial I/O clock select register and serial I/O mode select register.

Figure 15-2 shows the configuration and function of the serial I/O clock select register.

Figure 15-3 shows the configuration and function of the serial I/O mode select register.

Table 15-1 shows the setting status of each pin by the corresponding control flags. As shown in this table, the input/output setting flag of each pin must be also manipulated in addition to the control flag of the serial interface, to set each pin.

The SIO1CK1 and 0 flags select the internal clock (master) or external clock (slave) operation.

The SIO1HIZ flag selects whether the SO1/P0B2 pin is used as a serial data output pin.

The SIO1MOD flag selects whether the SO1/SI1/P0B1 pin is used as a serial data input (SI1 pin) or serial data output (SO2) pin.

	F	lag s	ymbo	ol		Read/					
Name	bз	b2	b1	bo	Address	Write					
			S	S	(BANK15)						
		   	1	I	1CH						
Serial I/O		1	0	0							
clock select	0	0	1	1		R/W					
register			С	С							
		1	К	К							
			1	0							
						Sets	shift clock of serial interface				
			0	0	External clock						
			0	1	12.5 kHz						
			1	0	18.75 kHz		Internal clock				
			1	1	37.5 kHz						
				•	Fixed to "0"						

## Figure 15-2. Configuration of Serial I/O Clock Select Register

eset	Reset by RESET pin	C	)	С	)	0	0
At re	WDT&SP reset					0	0
Clo	ock stop	,	, ;	,		0	0

	F	lag s	ymbo	ol		Read/				
Name	b₃	b2	b1	bo	Address	Write				
		S	s	S	(BANK15)					
		I	Т	Ι	1DH					
Serial I/O		0	0	0						
mode select	0	1	1	1		R/W				
register		М	н	т						
		0	I	S						
		D	Z							
				-		Starts	s/stops serial communication			
				0	Stops (wait st	tatus)				
				1	Starts					
				•		Sets	s function of PDB2/SO1 pin			
			0	1	General-purp	ose input/outp	ut port			
			4							
				1	Senal data ot	uput pin				
			,	Þ		Selects	function of P0B1/SI1/SO2 pin			
		0 As serial data input (SI1) pin								
		1			As serial data	a output (SO2)	pin			
			,							
				-	Fixed to "0"					

# Figure 15-3. Configuration of Serial I/O Mode Select Register

eset	Reset by RESET pin	0	0	0	0
At re	WDT&SP reset			0	0
Clo	ock stop	•	,	0	0

 $\star$ 

		Control flag	gs of	serial interf	ace			I/O setting flag of each pin				ing flag of each pin
Communi- cation mode	S   O 1 M O D	Serial I/O select	S I 0 1 H I Z	Serial interface pin setting	S I 0 1 C K 1	S I O 1 C K 0	Clock setting	Pin name	P 0 B 0 2	P 0 B 1 0	P 0 B 0 0	Set status of pin
3-wire serial					0	0	External	P0B0/SCK			0	During wait: general-purpose input port
I/O <sup>Note 1</sup>							clock					Wait released: external clock input
and											1	General-purpose output port
2-wire serial					0	1	Internal				0	General-purpose input port
I/O <sup>Note 2</sup>					1	0	clock				1	During wait: waits for internal clock output
					1	1						Wait released: internal clock output
	0	Internal clock						P0B1/SI1/ SO2		0		During wait: general-purpose input port Wait released: serial input
		(reception)								1		General-purpose output port
	1	Output								0		During wait: waits for serial output
		(trans- mission)								1		Wait released: serial output
			0	General-				P0B2/SO1	0			General-purpose input port
				I/O					1			General-purpose output port
			1	Serial					0			During wait: waits for serial output
				output					1			Wait released: serial output

## Table 15-1. Set Status of Each Pin By Control Flags

Notes 1. To set the 3-wire serial I/O mode, be sure to reset SIO1MOD to 0 and set SIO1HIZ to 1.

2. To use the 2-wire serial I/O mode, be sure to reset SIO1HIZ to 0.

### 15.2.1 Setting 2-/3-wire mode

The serial interface uses two pins in the two-wire mode: SCK/P0B0 and SO2/SI1/P0B1.

The SCK/P0B0 pin is used as a shift clock input/output pin, and the SO2/SI1/P0B1 pin is used as a serial data input/output pin. The SO1/P0B2 pin is not used for the serial interface and is set in the general-purpose output port mode by the SIO1HIZ flag. In this way, the serial interface operates in the two-wire mode.

In the three-wire mode, three pins, SCK/P0B0, SO1/P0B2, and SO2/SI1/P0B1 are used.

The SCK/P0B0 is used as a shift clock input/output pin, the SO1/P0B2 pin is used as a serial data output pin, and the SO2/SI1/P0B1 pin is used as a serial data input pin.

Unlike in the two-wire mode, the SO1/P0B2 pin is used as a serial data output pin according to the setting of the SIO1HIZ flag. The SO2/SI1/P0B1 pin is used as a serial data input pin according to the setting of the SIO1MOD flag. In this way, the serial interface operates in the three-wire mode.

## 15.2.2 Selecting data input/output using 2-wire serial interface

In the two-wire mode, the SO2/SI1/P0B1 pin is used as an input/output pin for serial data.

Whether the SO2/SI1/P0B1 pin is used as a serial data input pin (SI1 pin) or serial data output pin (SO2 pin) is specified by the SIO1MOD flag (refer to **Figure 15-3 Configuration of Serial I/O Mode Select Register**).

## 15.3 Clock Control Block

The clock control block generates a clock when the internal clock is used (master operation), and controls clock output timing.

The frequency fsc of the internal clock is set by the SIO1CK0 and SIO1CK1 flags of the serial I/O clock select register.

Figure 15-2 shows the configuration and function of the serial I/O clock select register.

For the clock generation timing, refer to 15.7 Operation of Serial Interface.

### 15.4 Clock Counter

The clock counter counts the shift clock output or input from the shift clock pin (SCK/P0B0 pin).

Because the clock counter directly reads the status of the clock pin, it cannot identify whether the clock is an internal clock or an external clock.

The contents of the clock counter cannot be directly read by software.

For the operation and timing chart of the clock counter, refer to 15.7 Operation of Serial Interface.

## 15.5 Presettable Shift Register

The presettable shift register is an 8-bit shift register that writes serial-out data and reads serial-in data. Writing/reading data to/from the presettable shift register is performed by PUT and GET instructions via data buffer.

The presettable shift register outputs (transmits) the content of its most significant bit (MSB) from the serial data I/O pin in synchronization with the falling edge of the shift clock, and reads data to its least significant bit (LSB) in synchronization with the rising edge of the shift clock.

Figure 15-4 shows the configuration and function of the presettable shift register.





**Note** If the PUT or GET instruction is executed during serial communication, the data may be lost. For details, refer to **15.8 Notes on Setting and Reading Data**.

### 15.6 Wait Control Block

The wait control block performs wait (pause) control of communication.

By releasing the wait status by using the SIO1TS flag of the serial I/O mode select register, serial communication is started.

After the wait status has been released, and communication has been started, the wait status is resumed if shift clock rises at clock counter "8".

The communication status can be detected by the SIO1TS flag.

That is, the communication status can be detected by detecting the status of the SIO1TS flag after setting "1" to the SIO1TS flag.

If "0" is written to the SIO1TS flag while the wait status is released, the wait status is set. This is called a forced wait status.

For the configuration and function of the serial I/O mode select register, refer to Figure 15-3.

## 15.7 Serial Interface Operation

The timing of each operation of the serial interface is described below. This timing is applicable to both 2-wire and 3-wire modes.

### 15.7.1 Timing chart

Figure 15-5 shows a timing chart.



Figure 15-5. Timing Chart of Serial Interface

Remark <1> Initial status (general-purpose input port)

- <2> Start condition satisfied by general-purpose I/O port
- <3> Wait released
- <4> Wait timing
- <5> General-purpose input port mode set
- <6> Stop condition satisfied by general-purpose I/O port

#### 15.7.2 Operation of clock counter

The initial value of the clock counter is "0". The value of the clock counter is incremented each time the falling edge of the clock pin has been detected. When the value of the clock counter reaches "8", it is reset to "0" at the next rising edge of the clock pin. After the clock counter has been reset to "0", the serial communication is placed in the wait status.

The conditions under which the clock counter is reset are as follows:

- (1) At power-ON reset
- (2) When clock stop instruction is executed
- (3) When "0" is written to SIO1TS flag
- (4) If shift clock rises while wait status is released and present value of clock counter is "8"

### 15.7.3 Wait operation and note

When the wait status has been released, serial data is output at the next falling edge of the clock (transmission operation), and the wait released status continues until eight clocks are counted.

After the eight clocks have been output, make the shift clock pin high and stop the operations of the clock counter and presettable shift register.

Note that, if data is written to or read from the presettable shift register while the wait status is released and the shift clock pin is high, the correct data is not set.

If data is written to the presettable shift register while the wait status is released and the shift clock pin is low, the content of the MSB of the data is output to the serial data output pin when the "PUT" instruction is executed.

If the forced wait status is set while the wait status is released, the wait status is immediately set when "0" is written to the SIO1TS flag.

#### 15.7.4 Interrupt request issuance timing

An interrupt request is issued when eight clocks have been transmitted (received).

#### 15.7.5 Shift clock generation timing

#### (1) When wait status is released from initial status

The "initial status" means the point at which the P0B0/SCK pin has been made high with the internal clock operation selected.

During the wait status, a high level is output to the shift clock pin.

The wait status can be released and a clock can be selected at the same time.



### Figure 15-6. Shift Clock Generation Timing of Serial Interface (1/4)

## (2) When wait operation is performed

(a) When wait status is set at the 8th clock (normal operation)

Figure 15-6. Shift Clock Generation Timing of Serial Interface (2/4)



## (b) When forced wait status is set during wait status

### Figure 15-6. Shift Clock Generation Timing of Serial Interface (3/4)



### (d) When wait status is released while wait status is released

The clock output waveform does not change. Neither is the counter reset. However, do not change the clock frequency while the wait status is released.

## 15.8 Notes on Setting and Reading Data

Use the "PUT SIO1SFR, DBF" instruction to set data to the presettable shift register. Use the "GET DBF, SIO1SFR" instruction to read data.

Set or read the data in the wait status. While the wait status is released, the data may not be correctly set or read depending on the status of the shift clock pin.

The following table describes the points to be noted in setting and reading data.

### Table 15-2. Data Read and Write Operations of Presettable Shift Register and Notes

Status o of P	on execution UT/GET	Status of shift clock pin	Operation of presettable shift register
	Read (GET)		Normal read
Wait status	Write (PUT)	<ul> <li>Floating with external clock</li> <li>Value of output latch with internal clock. Normally, used with high level</li> </ul>	Normal write         Content of MSB is output as data at falling edge of shift clock         after wait status is released next (transmission operation).         Clock         Data       X MSB
		Low level	Normal read
	Read (GET)	High level	Cannot be read normally. Contents of SIO1SFR are lost.
		Low level	Cannot be written normally. Contents of SIO1SFR are lost.
Wait release status	Write (PUT)	High level	Normal write Content of MSB is output as data when PUT instruction is executed. Clock counter is not reset. Clock Data MSB † PUT SIO1SFR, DBF

# 15.9 Operation Mode and Operational Outline of Each Blocks

Tables 15-3 and 15-4 outline the operations of the serial interface.

0	peration mode	Slave operation (SIO1	CK1 = SIO1CK0 = 0)	Master operation (SIO1CK1 = SIO1CK0 = other than 0)								
Item		During wait (SIO1TS = 0)	Wait released (SIO1TS = 1)	During wait (SIO1TS = 0)	Wait released (SIO1TS = 1)							
Status of each pin	SCK/P0B0	<ul> <li>When P0BBIO0 = 0 General-purpose input port</li> <li>When P0BBIO0 = 1 General-purpose output port</li> </ul>	<ul> <li>When P0BBIO0 = 0 External clock input port</li> <li>When P0BBIO0 = 1 General-purpose output port</li> </ul>	<ul> <li>When P0BBIO0 = 0 General-purpose input port</li> <li>When P0BBIO0 = 1 Waits for internal clock output</li> </ul>	<ul> <li>When P0BBIO0 = 0 General-purpose input port</li> <li>When P0BBIO0 = 1 Internal clock output</li> </ul>							
	SI1/SO2/P0B1		SIO1M	OD = 0								
		<ul> <li>When P0BBIO1 = 0 General-purpose input port</li> <li>When P0BBIO1 = 1 General-purpose output port</li> </ul>	<ul> <li>When P0BBIO1 = 0 Serial input</li> <li>When P0BBIO1 = 1 General-purpose output port</li> </ul>	<ul> <li>When P0BBIO1 = 0 General-purpose input port</li> <li>When P0BBIO1 = 1 General-purpose output port</li> </ul>	<ul> <li>When P0BBIO1 = 0 Serial input</li> <li>When P0BBIO1 = 1 General-purpose output port</li> </ul>							
	SO1/P0B2	SIO1HIZ = 1										
		Waits for serial output	Serial output	Waits for serial output	Serial output							
Program count	er	Incremented at falling edge of SCK pin										
Operation of presettable shift register	Output	<ul> <li>When SIO1HIZ = 0 Not output</li> <li>When SIO1HIZ = 1 Shifted from MSB and output</li> </ul>	When SIO1HIZ = 0     Not output     When SIO1HIZ = 1     Shifted from MSB and output from SO1 pin at falling edge of SCK pin									
	Input	When SIO1MOD = 0 Shifted from LSB and status If SI1 pin is set in output mod	of SI1 pin is input at rising edg de, however, contents of outpu	e of SCK pin. t latch are input.								

Table 15-3. Operation in 3-wire Serial I/O Mode

0	peration mode	Slave operation (SIO	ICK1 = SIO1CK0 = 0)	Master operation (SIO1CK1 = SIO1CK0 = other than 0)									
Item		During wait (SIO1TS = 0)	Wait released (SIO1TS = 1)	During wait (SIO1TS = 0)	Wait released (SIO1TS = 1)								
Status of each pin	SCK/P0B0	<ul> <li>When P0BBIO0 = 0 General-purpose input port</li> <li>When P0BBIO0 = 1 General-purpose output port</li> </ul>	<ul> <li>When P0BBIO0 = 0 External clock input port</li> <li>When P0BBIO0 = 1 General-purpose output port</li> </ul>	<ul> <li>When P0BBIO0 = 0 General-purpose input port</li> <li>When P0BBIO0 = 1 Waits for internal clock output</li> </ul>	<ul> <li>When P0BBIO0 = 0 General-purpose input port</li> <li>When P0BBIO0 = 1 Internal clock output</li> </ul>								
	SI1/SO2/P0B1	SIO1MOD = 0											
		<ul> <li>When P0BBIO1 = 0 General-purpose input port</li> <li>When P0BBIO1 = 1 General-purpose output port</li> </ul>	<ul> <li>When P0BBIO1 = 0 Serial input</li> <li>When P0BBIO1 = 1 General-purpose output port</li> </ul>	<ul> <li>When P0BBIO1 = 0 General-purpose input port</li> <li>When P0BBIO1 = 1 General-purpose output port</li> </ul>	<ul> <li>When P0BBIO1 = 0 Serial input</li> <li>When P0BBIO1 = 1 General-purpose output port</li> </ul>								
		SIO1MOD = 1											
		Waits for serial output regardless of P0BBIO1	Serial output regardless of P0BBIO1	Waits for serial output regardless of P0BBIO1	Serial output regardless of P0BBIO1								
	SO1/P0B2	SIO1HIZ = 0											
		<ul> <li>When P0BBIO2 = 0 General-purpose input port</li> <li>When P0BBIO2 = 1 General-purpose output port</li> </ul>	<ul> <li>When P0BBIO2 = 0 Serial input</li> <li>When P0BBIO2 = 1 General-purpose output port</li> </ul>	<ul> <li>When P0BBIO2 = 0 General-purpose input port</li> <li>When P0BBIO2 = 1 General-purpose output port</li> </ul>	<ul> <li>When P0BBIO2 = 0 Serial input</li> <li>When P0BBIO2 = 1 General-purpose output port</li> </ul>								
		SIO1MOD = 1											
		Waits for serial output regardless of P0BBIO2	Serial output regardless of P0BBIO2	Waits for serial output regardless of P0BBIO2	Serial output regardless of P0BBIO2								
Program coun	ter	Incremented at falling edge of	SCK pin		•								
Operation of presettable	Output	When SIO1SEL = 1     Shifted from MSB and output	It from SO2 pin at falling edge	of SCK pin									
shift register	Input	• When SIO1SEL = 0 Shifted from LSB and status If SI1 pin is set in output por	of SI1 pin is input at rising edg t mode, however, contents of c	ge of SCK pin. butput latch are input.									

# Table 15-4. Operation in Two-Wire Serial I/O Mode

## 15.10 Status on Reset

## 15.10.1 At reset by RESET pin

P0B0/SCK, P0B1/SI1/SO2, and P0B2/SO1 pins are set in the general-purpose input port mode. The contents of the presettable shift register are undefined.

### 15.10.2 WDT&SP reset

P0B0/SCK pin, P0B1/SI1/SO2, and P0B2/SO1 pins are set in the general-purpose input port. The previous contents of the presettable shift register are retained.

### 15.10.3 At clock stop

All the pins hold the current status.

The previous contents of the presettable shift register are retained.

## 15.10.4 In halt status

All the pins hold the current status.

The internal clock stops output in the status in which the HALT instruction is executed.

When the external clock is used, the operation continued even if the HALT instruction is executed.

## **16. PLL FREQUENCY SYNTHESIZER**

The PLL (Phase Locked Loop) frequency synthesizer is used to lock a frequency in the MF (Medium Frequency), HF (High Frequency), and VHF (Very High Frequency) to a constant frequency by means of phase difference comparison.

## 16.1 Outline of PLL Frequency Synthesizer

Figure 16-1 outlines the PLL frequency synthesizer. A PLL frequency synthesizer can be configured by connecting an external lowpass filter (LPF) and voltage controlled oscillator (VCO).

The PLL frequency synthesizer divides a signal input from the VCOH or VCOL pin by using a programmable divider and outputs a phase difference between this signal and a reference frequency from the EO0 and EO1 pins.



Figure 16-1. Outline of PLL Frequency Synthesizer

Note External circuit

- **Remarks 1.** PLLMD1 and PLLMD0 (bits 1 and 0 of PLL mode selection register: refer to **Figure 16-3**) selects a division mode of the PLL frequency synthesizer.
  - 2. PLLSCNF (bit 3 of PLL mode selection register: refer to **Figure 16-3**) selects the least significant bit of the swallow counter.
  - **3.** PLLRFCK2 through PLLRFCK0 (bits 3 through 0 of PLL reference frequency selection register: refer to **Figure 16-6**) selects a reference frequency fr of the PLL frequency synthesizer.
  - 4. PLLUL (bit 0 of PLL unlock FF register: refer to Figure 16-9) detects the PLL unlock FF status.

## 16.2 Input Selection Block and Programmable Divider

### 16.2.1 Configuration and function of input selection block and programmable divider

Figure 16-2 shows the configuration of the input selection block and programmable divider.

The input selection block selects an input pin and division mode of the PLL frequency synthesizer.

The VCOH or VCOL pin can be selected as the input pin.

The voltage on the selected pin is at the intermediate level (approx. 1/2 VDD). The pin not selected is internally pulled down.

Because these pins are connected to an internal AC amplifier, cut the DC component of the input signal by connecting a capacitor in series to the pin.

Direct division mode and pulse swallow mode can be selected as division modes.

The programmable divider divides the frequency of the input signal according to the value set to the swallow counter and programmable counter.

The pin and division mode to be used are selected by the PLL mode selection register.

Figure 16-3 shows the configuration of the PLL mode selection register.

The value of the programmable divider is set by using the PLL data register via data buffer.





Note PLLSCNF flag

Name	FI										
	bз	b2	b1	bo							
PLL mode selection	Ρ	0	Ρ	Р	(BANK15)	R/W					
	L	1 1 1	L	L	10H						
	L	1 1 1	L	L							
	s	- - - -	М	м							
	с	- - - -	D	D							
	N	     	1	0							
	F	1 1 1	1	   							
					S	Selects division mode of PLL frequency synthesizer					
			0	0	Disables VCO	L and VCOH pin	s				
			0	1	Direct division	(VCOL pin, MF	mode)				
			1	0	Pulse swallow	(VCOH pin, VH	<sup>=</sup> mode)				
			1	1	Pulse swallow	(VCOL pin, HF	node)				
				->	Fixed to "0"						
				-		Selects least si	gnificant bit of swallow counter				
	0				Clears least si	gnificant bit to 0					
	1	-			Sets least significant bit to 1						
		L									

#### Figure 16-3. Configuration of PLL Mode Selection Register

eset	Reset by RESET pin	U	0	0	0
At re	WDT&SP reset	U		0	0
Clo	ock stop	R		0	0

U: Undefined R: Retained

#### 16.2.2 Outline of each division mode

### (1) Direct division mode (MF)

In this mode, the VCOL pin is used. The VCOH pin is pulled down. In this mode, only the programmable counter is used for frequency division.

## (2) Pulse swallow mode (HF)

In this mode, the VCOL pin is used. The VCOH pin is pulled down. In this mode, the swallow counter and programmable counter are used for frequency division.

## (3) Pulse swallow mode (VHF)

In this mode, the VCOH pin is used. The VCOL pin is pulled down. In this mode, the swallow counter and programmable counter are used for frequency division.

## (4) VCOL and VCOH pin disabled

In this mode, only the VCOL and VCOH pins are internally pulled down, but the other blocks operate.

## 16.2.3 Programmable divider and PLL data register

The programmable divider consists of a 5-bit swallow counter and a 12-bit programmable counter. Each counter is a 17-bit binary down counter.

The programmable counter is allocated to the high-order 12 bits of the PLL data register, and the swallow counter is allocated to the low-order 4 bits. Data are set to these counters via data buffer.

The least significant bit of the swallow counter sets data to the PLLSCNF flag of the control register.

The value by which the input signal frequency is to be divided is called "N value".

For how to set a division value (N value) in each division mode, refer to 16.6 Using PLL Frequency Synthesizer.

## (1) PLL data register and data buffer

Figure 16-4 shows the relationships between the PLL data register and data buffer.

In the direct division mode, the high-order 12 bits of the PLL data register are valid, and all 17 bits of the register are valid in the pulse swallow mode.

In the direct division mode, all 12 bits are used as a programmable counter.

In the pulse swallow mode, the high-order 12 bits are used as a programmable counter, and the low-order 5 bits are used as a swallow counter.

(2) Relationship between division value N of programmable divider and divided output frequency

The relationship between the value "N" set to the PLL data register and the signal frequency "fN" divided and output by the programmable divider is as shown below.

For details, refer to 16.6 Using PLL Frequency Synthesizer.

## (a) Direct division mode (MF)

$$f_{\rm IN} = \frac{f_{\rm IN}}{N}$$
 N: 12 bits

(b) Pulse swallow mode (HF, VHF)

$$f_{IN} = \frac{f_{IN}}{N}$$
 N: 17 bits



#### Figure 16-4. Setting Division Value (N Value) of PLL Frequency Synthesizer

**Note** The value of PLLSCNF flag is transferred when a write (PUT) instruction is executed to the PLL data register (PLLR). Therefore, data must be set to the PLLSCNF flag before executing the write instruction to the PLL data register.

## 16.3 Reference Frequency Generator

Figure 16-5 shows the configuration of the reference frequency generator.

The reference frequency generator generates the reference frequency "fr" of the PLL frequency synthesizer by dividing the 75 kHz output of a crystal oscillator.

Six frequencies can be selected as reference frequency fr: 1, 3, 5, 6.25, 12.5, and 25 kHz.

The reference frequency fr is selected by the PLL reference frequency selection register.

Figure 16-6 shows the configuration and function of the PLL reference frequency selection registerion.





Name	Flag symbol		ol	Address	Read/Write		
	bз	b2	b1	bo			
PLL reference	0	Р	Р	Р	(BANK15)	R/W	
frequency selection		L	L	L	11H		
		L	L	L			
		R	R	R			
		F	F	F			
		с	с	с			
		к	к	к			
		2	1	0			
							1
				-	Sets	reference frequ	ency fr of PLL frequency synthesizer
		0	0	0	1 kHz		
		0	0	1	3 kHz		
		0	1	0	5 kHz		
		0	1	1	6.25 kHz		
		1	0	0	12.5 kHz		
		1	0	1	25 kHz		
		1	1	0	PLL disable		
		1	1	1	PLL disable		
			_				
	L			Fix	ked to "0"		

## Figure 16-6. Configuration of PLL Reference Frequency Selection Register

eset	Reset by RESET Pin	0	1	1	1
At re	WDT&SP reset	0	1	1	1
Clo	ock stop	0	1	1	1

**Remark** When the PLL frequency synthesizer is disabled by the PLL reference frequency selection register, the VCOH and VCOL pins are internally pulled down. The EO1 and EO0 pins are floated.

## 16.4 Phase Comparator (*\phi*-DET), Charge Pump, and Unlock FF

### 16.4.1 Configuration of phase comparator, charge pump, and unlock FF

Figure 16-7 shows the configuration of the phase comparator, charge pump, and unlock FF.

The phase comparator compares the phase of the divided frequency "fn" output by the programmable divider with the phase of the reference frequency "fr" output by the reference frequency generator, and outputs an up  $(\overline{UP})$  or down  $(\overline{DW})$  request signal.

The charge pump outputs the output of the phase comparator from an error out pin (EO1 and EO0 pins). The unlock FF detects the unlock status of the PLL frequency synthesizer.

16.4.2 through 16.4.4 describe the operations of the phase comparator, charge pump, and unlock FF.

Figure 16-7. Configuration of Phase Comparator, Charge Pump, and Unlock FF



### 16.4.2 Function of phase comparator

As shown in Figure 16-7, the phase comparator compares the phases of the divided frequency "f<sub>N</sub>" output by the programmable divider and the reference frequency "fr", and outputs an up or down request signal.

If the divided frequency  $f_N$  is lower than reference frequency fr, the up request signal is output. If  $f_N$  is higher than fr, the down request signal is output.

Figure 16-8 shows the relationship between reference frequency fr, divided frequency fN, up request signal, and down request signal.

When the PLL frequency synthesizer is disabled, neither the up request nor the down request signal is output. The up and down request signals are input to the charge pump and unlock FF, respectively.

## Figure 16-8. Relationship between fr, $f_N$ , $\overline{UP}$ , and $\overline{DW}$

#### (a) If fN lags behind fr



### 16.4.3 Charge pump

As shown in Figure 16-7, the charge pump outputs the up request and down request signals output by the phase comparator, from the error out pins (EO1 and EO0 pins).

Therefore, the relationship between the output of the error out pins, divided frequency  $f_N$  and reference frequency fr is as follows:

Where reference frequency fr > divided frequency fN: Low-level output Where reference frequency fr < divided frequency fN: High-level output Where reference frequency fr = divided frequency fN: Floating

## 16.4.4 Unlock FF

As shown in Figure 16-7, the unlock FF detects the unlock status of the PLL frequency synthesizer from the up request and down request signals of the phase comparator.

Because either the up request or down request signal is low in the unlock status, the unlock status is detected by this low-level signal.

In the unlock status, the unlock FF is set to 1.

The unlock FF is set in the cycle of the reference frequency fr selected at that time. When the contents of the PLL unlock FF register are read (by the PEEK instruction), the unlock FF is reset (Read & Reset).

Therefore, the unlock FF must be detected in a cycle longer than cycle 1/fr of the reference frequency fr.

The status of the unlock FF is detected by the PLL unlock FF register. Figure 16-9 shows the configuration of the PLL unlock FF register.

Because this register is a read-only register, its contents can be read to the window register by the "PEEK" instruction.

Because the unlock FF is set in a cycle of the reference frequency fr, the contents of the PLL unlock FF register are read to the window register in a cycle longer than cycle 1/fr of the reference frequency.

The delay time of the up and down request signals of the phase comparator are fixed to 0.8 to 1.0  $\mu$ s.

Name	Flag symbol		ol	Address	Read/Write		
	bз	b2	b1	bo			
PLL unlock FF	0	0	0	P	(BANK15)	R & Reset	
			1	L	12H		
				L			
		 	   	υ			
				L			
						Detec	status of unlock FF
				0	Unlock FF = 0:	PLL locked stat	i
				1	Unlock FF = 1: PLL unlocked status		
				•	Fixed to "0"		
	0						

# Figure 16-9. Configuration of PLL Unlock FF Register

eset	Reset by RESET pin	0	0	0	U
At re	WDT&SP reset				υ
Clo	ock stop				R

U: Undefined R: Retained

## 16.5 PLL Disabled Status

The PLL frequency synthesizer stops when PLL disabled status is selected by the PLL reference frequency register (RF address 11H).

Table 16-1 shows the operation of each block in the PLL disabled status.

When the VCOL and VCOH pins are disabled by the PLL mode selection register, only the VCOL and VCOH pins are internally pulled down, and the other blocks operate.

At reset by RESET pin, the PLL frequency synthesizer is disabled.

## Table 16-1. Operation of Each Block under Each PLL Disable Condition

Condition Each Block	PLL reference frequency selection register = 1111B (PLL disabled)	PLL mode selection register = 0000B (VCOH and VCOL disabled)		
VCOL, VCOH pins	Internally pulled down	Internally pulled down		
Programmable divider	Division stopped	Operates		
Reference frequency generator	Output stopped	Operates		
Phase comparator	Output stopped	Operates		
Charge pump	Error out pins are floated	Operates. However, usually outputs low level because no signal is input		

## 16.6 Using PLL Frequency Synthesizer

To control the PLL frequency synthesizer, the following data is necessary.

- (1) Division mode : Direct division (MF), pulse swallow (HF, VHF)
- (2) Pins used : VCOL and VCOH pins
- (3) Reference frequency : fr
- (4) Division value : N

16.6.1 through 16.6.3 below describe how to set PLL data in each division mode (MF, HF, and VHF).

#### 16.6.1 Direct division mode (MF)

#### (1) Selecting division mode

Select the direct division mode by using the PLL mode selection register.

(2) Pins used

The VCOL pin is enabled to operate when the direct division mode is selected.

#### (3) Selecting reference frequency fr

Select the reference frequency by using the PLL reference frequency selection register.

(4) Calculation of division value N

Calculate N as follows:

$$N = \frac{f_{VCOL}}{fr}$$

fvcoL : Input frequency of VCOL pin

fr : Reference frequency

### (5) Example of setting PLL data

How to set data to receive broadcasting in the following MW band is described below.

Reception frequency : 1422 kHz (MW band)

Reference frequency : 3 kHz

Intermediate frequency : +450 kHz

Division value N is calculated as follows:

$$N = \frac{f_{VCOL}}{fr} = \frac{1422 + 450}{3} = 624 \text{ (decimal)}$$
$$= 270 \text{H (hexadecimal)}$$

Set data to the PLL data register, PLL mode selection register, and PLL reference frequency selection register as follows:



Notes 1. PLLSCNF flag

2. don't care

### 16.6.2 Pulse swallow mode (HF)

#### (1) Selecting division mode

Select the pulse swallow mode by using the PLL mode selection register.

#### (2) Pins used

The VCOL pin is enabled to operate when the pulse swallow mode is selected.

#### (3) Selecting reference frequency fr

Select the reference frequency by using the PLL reference frequency selection register.

#### (4) Calculation of division value N

Calculate N as follows:

 $N = \frac{f_{VCOL}}{fr}$ 

fvcoL : Input frequency of VCOL pin

fr : Reference frequency

#### (5) Example of setting PLL data

How to set data to receive broadcasting in the following SW band is described below.

Reception frequency : 25.50 MHz (SW band)

Reference frequency : 5 kHz

Intermediate frequency: +450 kHz

Division value N is calculated as follows:

 $N = \frac{f_{VCOL}}{fr} = \frac{25500 + 450}{5} = 5190 \text{ (decimal)}$ = 1446H (hexadecimal)

Set data to the PLL data register, PLL mode selection register, and PLL reference frequency selection register as follows:

Caution The division value N is 17 bits long when the pulse swallow mode is selected, and the least significant bit of the swallow counter is the bit 3 of the PLL mode selection register (PLLSCNF). To set "1446H" as the division value N, the value to be actually set to the PLL data register is "0A23H".



Note PLLSCNF flag

## 16.6.3 Pulse swallow mode (VHF)

## (1) Selecting division mode

Select the pulse swallow mode by using the PLL mode selection register.

## (2) Pins used

The VCOH pin is enabled to operate when the pulse swallow mode is selected.

## (3) Selecting reference frequency fr

Select the reference frequency by using the PLL reference frequency selection register.

## (4) Calculation of division value N

Calculate N as follows:

 $N = \frac{f_{VCOH}}{f_{T}}$ 

fvcoн: Input frequency of VCOH pin

fr : Reference frequency

## (5) Example of setting PLL data

How to set data to receive broadcasting in the following FM band is described below.

Reception frequency : 98.15 MHz (FM band)

Reference frequency : 25 kHz

Intermediate frequency : +10.7 MHz

Division value N is calculated as follows:

 $N = \frac{f_{VCOH}}{fr} = \frac{98.15 + 10.7}{0.025} = 4354 \text{ (decimal)}$ = 1102H (hexadecimal)

Set data to the PLL data register, PLL mode selection register, and PLL reference frequency selection register as follows:

Caution The division value N is 17 bits long when the pulse swallow mode is selected, and the least significant bit of the swallow counter is the bit 3 of the PLL mode selection register (PLLSCNF). To set "1102H" as the division value N, the value to be actually set to the PLL data register is "0881H".



Note PLLSCNF flag
# NEC

Note that data must be set to the PLLSCNF flag before a write (PUT) instruction is executed to the PLL data register (PLLR).

#### Example

SET1	PLLSCNF
MOV	DBF0, #0
MOV	DBF1, #4
MOV	DBF2, #4
PUT	PLLR, DBF

# 16.7 Status at Reset

## 16.7.1 At reset by RESET pin

The PLL frequency synthesizer is disabled because the PLL reference frequency selection register is initialized to 0111B.

#### 16.7.2 At WDT&SP reset

The PLL frequency synthesizer is disabled because the PLL reference frequency selection register is initialized to 0111B.

### 16.7.3 On execution of clock stop instruction

The PLL frequency synthesizer is disabled because the PLL reference frequency selection register is initialized to 0111B.

## 16.7.4 In halt status

The set status is retained.

# 17. INTERMEDIATE FREQUENCY (IF) COUNTER

# ★ 17.1 Outline of Frequency Counter

Figure 17-1 outlines the frequency counter.

The IF counter is mainly used to count the intermediate frequency (IF) output from a tuner for detecting broadcasting stations.

The IF counter counts the frequency input to the P1C2/AMIFC or P1C3/FMIFC/AMIFC pin at fixed intervals (1 ms, 4 ms, 8 ms, or open) by using a 16-bit counter.



Figure 17-1. Outline of IF Counter



- **Remarks 1.** IFCMD1 and IFCMD0 (bits 3 and 2 of IF counter mode selection register: refer to **Figure 17-3**) select the IF counter function.
  - 2. IFCCK1 and IFCCK0 (bits 1 and 0 of IF counter mode selection register: refer to Figure 17-3) select the gate time of the IF counter.
  - 3. IFCSTRT (bit 1 of IF counter control register: refer to Figure 17-5) control starting of the IF counter.
  - 4. IFCGOSTT (bit 0 of IF counter gate status detection register: refer to Figure 17-6) detects opening/ closing the gate of the IF counter function.
  - 5. IFCRES (bit 0 of IF counter control register: refer to Figure 17-5) reset the count value of the IF counter.

★

# **\*** 17.2 IF Counter Input Selection Block and Gate Time Control Block

Figure 17-2 shows the configuration of the IF counter input selection block and gate time control block. The IF counter selection block selects whether the P1C2/AMIFC or P1C3/FMIFC/AMIFC pin is used as an IF counter function or a general-purpose input port, by using the IF counter mode selection register.

The gate time control block selects gate time by using the IF counter mode selection register when the frequency counter is used as the IF counter.

Figure 17-3 shows the configuration of the IF counter mode selection register.

#### Figure 17-2. Configuration of IF Counter Input Selection Block and Gate Time Control Block



# $\star$

# Figure 17-3. Configuration of IF Counter Mode Selection Register

Fl	Flag symbol			Address	Read/Write									
bз	b2	b1	bo											
I	I	Ι	I	(BANK15)	R/W									
F	F	F	F	22H										
с	с	С	С											
м	М	С	С											
D	D	к	к											
1	0	1	0											
<u> </u>						·								
			-	Selects gate time of IF counter										
		0	0	1 ms	ms									
		0	1	4 ms										
		1	0	8 ms										
		1	1	Open										
				Selects function of IF counter										
0	0			IF counter OFF	- mode (general	-purpose input port)								
0	1			AMIFC pin, AM	IIF count mode									
1	0			FMIFC pin, FM	IIF count mode,	1/2 division								
1	1			FMIFC pin, AM	1IF count mode									
	FI. b3 I F C M D 1 1	Flags         b3       b2         I       I         F       F         C       C         M       M         D       D         1       0	Flag symbol         b3       b2       b1         I       I       I         F       F       F         C       C       C         M       M       C         D       D       K         1       0       1         I       0       1         I       0       1         I       0       1         I       0       1         I       0       1         I       0       1         I       0       1         I       0       1         I       0       1	Flag symbol         b3       b2       b1       b0         I       I       I       I         F       F       F       F         C       C       C       C         M       M       C       C         D       D       K       K         1       0       1       0         0       0       1       1         0       0       1       1         0       0       1       1         1       0       1       1         1       0       1       1         1       0       1       1         1       0       1       1	Haddress         b3       b2       b1       b0         I       I       I       I       (BANK15)         F       F       F       F       22H         C       C       C       C         M       M       C       C       C         D       D       K       K       1         1       0       1       0       1         V       V       V       V       V         I       0       1       0       1       R         0       0       1       Ams       1       Image: Colstance of the	Haddress         Read/Write           b3         b2         b1         b0         Address         Read/Write           b3         b2         b1         b0         Colspan="5">Colspan=55         R/W           I         I         I         I         (BANK15)         R/W           F         F         F         F         P         22H         Image: Colspan=15           M         M         C         C         C         C         C         Image: Colspan=15         Image: Colspan=15								

eset	Reset by RESET pin	0	0	0	0
At re	WDT&SP reset	0	0	0	0
Clo	ock stop	0	0	0	0

+

## 17.3 Start/Stop Control Block and IF Counter

#### 17.3.1 Configuration of start/stop control block and IF counter

Figure 17-4 shows the configuration of the start/stop control block and IF counter.

The start/stop control block starts the frequency counter or detects the end of counting.

The counter is started by the IF counter control register.

The end of counting is detected by the IF counter gate status detection register.

Figure 17-6 shows the configuration of the IF counter control register.

Figure 17-7 shows the configuration of the IF counter gate status detection register.

17.3.2 describes the gate operation when the IF counter function is selected.

The IF counter is a 16-bit binary counter that counts up the input frequency when the IF counter function is selected. When the IF counter function is selected, the frequency input to a selected pin is counted while the gate is opened by an internal gate signal. The frequency count is counted without alteration in the AMIF count mode. In the FMIF counter mode, however, the frequency input to the pin is halved and counted.

When the IF counter counts up to FFFFH, it remains at FFFFH until reset.

The count value is read by the IF counter data register (IFC) via data buffer.

The count value is reset by the IF counter control register.

Figure 17-7 shows the configuration of the IF counter data register.

#### Figure 17-4. Configuration of Start/Stop Control Block and IF Counter



Name	FI	ag s	ymb	ol	Address	Read/Write						
	bз	b2	b1	bo								
IF counter control	0	0	Ι	I	(BANK15)	W						
			F	F	23H							
			C C S R									
		1										
		1	т	Е								
			R	s								
			т									
							•					
				-		Rese	ets data of IF counter					
				0	Nothing is affected							
				1	Resets counte	r						
			'									
				•	- Start IF counter							
			0		Nothing is affected							
	1				Resets counter							
			L									
					Fixed to "0"							
				I								

			-	_	_
Figure 17-5.	Configuration	of IF	Counter	Control	Register

eset	Reset by RESET pin	0	 0	0	0
At re	WDT&SP reset			0	0
Clo	ock stop	,		0	0

\*



#### Figure 17-6. Configuration of IF Counter Gate Status Detection Register

eset	Reset by RESET pin	0	- (	0	C	)	0
At re	WDT&SP reset						0
Clo	ock stop			,		,	0

Caution Do not read the contents of the IF counter data register (IFC) to the data buffer while the IFCGOSTT flag is set to 1.

## 17.3.2 Operation of gate when IF counter function is selected

#### (1) When gate time of 1, 4, or 8 ms is selected

The gate is opened for 1, 4, or 8 ms from the rising of the internal 1-kHz signal after the IFCSTRT flag has been set to 1, as illustrated below.

While this gate is open, the frequency input from a selected pin is counted by a 16-bit counter.

When the gate is closed, the IFCGOSTT flag is cleared to 0.

The IFCGOSTT flag is automatically set to 1 when the IFCSTRT flag is set.



#### (2) When gate is open

If opening of the gate is selected by the IFCCK1 and IFCCK0 flags, the gate is opened as soon as its opening has been selected, as illustrated below.

If the counter is started by using the IFCSTRT flag while the gate is open, the gate is closed after undefined time.

To open the gate, therefore, do not set the IFCSTRT flag to 1.

However, the counter can be reset by the IFCRES flag.



The gate is opened or closed in the following two ways when opening the gate is selected as the gate time.

#### (a) Resetting the gate to other than open by using IFCCK1 and IFCCK0 flags



#### (b) Unselect pin used by using IFCMD1 and IFCMD0 flags

In this way, the gate remains open, and counting is stopped by disabling input from the pin.



#### 17.3.3 Function and operation of 16-bit counter

The 16-bit counter counts up the frequency input within selected gate time.

The 16-bit counter can be reset by writing "1" to the IFCRES flag of the IF counter control register.

Once the 16-bit counter has counted up to FFFFH, it remains at FFFFH until it is reset.

The IF counter counts the frequency input to the P1C2/AMIFC or P1C3/FMIFC/AMIFC pin while the gate is open. Note, however, in the FMIF count mode input to the P1C3/FMIFC/AMIFC is divided by two and counted.

The relationship between count value "x (decimal)" and input frequencies (fFMIFC and fAMIFC) is shown below.

• FMIFC

 $f_{FMIFC} = \frac{x}{t_{GATE}} \times 2 \text{ (kHz)} \qquad t_{GATE} \text{ gate time (1 ms, 4 ms, 8 ms)}$  AMIFC  $f_{AMIFC} = \frac{x}{t_{GATE}} \text{ (kHz)} \qquad t_{GATE} \text{ gate time (1 ms, 4 ms, 8 ms)}$ 

The value of the IF counter data register is read via data buffer.

Figure 17-7 shows the configuration and function of the IF counter data register.



## Figure 17-7. Configuration of IF Counter Data Register

Once the IF counter data register has counted up to FFFFH, it remains at FFFFH until the counter is reset.

## 17.4 Using IF Counter

The following sections 17.4.1 through 17.4.3 describe how to use the hardware of the IF counter, a program example, and count error.

\*

#### 17.4.1 Using hardware of IF counter

Figure 17-8 shows the block diagram when the P1C2/AMIFC and P1C3/FMIFC/AMIFC pins.

As shown in the figure, the IF counter uses an input pin with an AC amplifier, the DC component of the input signal must be cut with a capacitor.

When the P1C2/AMIFC or P1C3/FMIFC/AMIFC pin is selected for the IF counter function, switch SW turns ON, and the voltage level on each pin reaches about 1/2Vpp.

If the voltage has not risen to a sufficient intermediate level at this time, the IF counter does not operate normally because the AC amplifier is not in the normal operating range.

Therefore, make sure that a sufficient wait time elapses after each pin has been specified to be used for the IF counter until counting is started.





## ★ 17.4.2 Program example of IF counter

A program example of the IF counter is shown below.

As shown in this example, make sure that a wait time elapses after an instruction that selects the P1C2/AMIFC or P1C3/FMIFC/AMIFC pin for the IF counter function has been executed until counting is started.

This is because, as described in 17.4.1, the internal AC amplifier does not operate normally immediately after a pin has been selected for the IF counter.

Example To count the frequency input to the P1C3/FMIFC pin (FMIF count mode) (gate time: 8 ms)

INITFLG IFCMD1, NOT IFCMD0, IFCCK1, NOT IFCCK0

			; Selects FMIFC pin (FMIF count mode), and sets gate time to 8 ms
	\ \	Wait	; Internal AC amplifier stabilization time
	SET1	IFCRES	; Resets counter
	SET1	IFCSTRT	; Starts counting
LOC	P:		
	SKT1	IFCG0STT	; Detects opening or closing of gate
	BR	READ	; Branches to READ: if gate is closed
	Proc	essing A	
	BR	LOOP	; Do not read data of IF counter with this processing A
REA			Deede velve of IC equation data as sister to data huffer
	GEL	DBF. IFC	: Reads value of it counter data redister to data butter

#### 17.4.3 Error of IF counter

The errors of the IF counter include a gate time error and a count error. The following paragraphs (1) and (2) describe each of these errors.

#### (1) Gate time error

The gate time of the IF counter is created by dividing the 75-kHz clock. Therefore, if the system clock is shifted from 75 kHz by "+x" ppm, the gate time is shifted by "-x" ppm.

#### (2) Count error

The IF counter counts frequency by the rising edge of the input signal. If a high level is input to the pin when the gate is open, therefore, one excess pulse is counted. If the gate is closed, however, a count error due to the status of the pin does not occur. Therefore, the count error is "+1, -0". \*

# 17.5 Status at Reset

## 17.5.1 At reset by RESET pin

The P1C2/AMIFC and P1C3/FMIFC/AMIFC pins are set in the general-purpose input port mode.

## 17.5.2 At WDT&SP reset

The P1C2/AMIFC and P1C3/FMIFC/AMIFC pins are set in the general-purpose input port mode.

#### 17.5.3 On execution of clock stop instruction

The P1C2/AMIFC and P1C3/FMIFC/AMIFC pins are set in the general-purpose input port mode.

#### 17.5.4 In halt status

The P1C2/AMIFC and P1C3/FMIFC/AMIFC pins retain the status immediately before the halt mode is set.

# 18. BEEP

## 18.1 Outlines of BEEP

Figure 18-1 outlines BEEP.

BEEP outputs a clock of 1.5 kHz or 3 kHz from the BEEP pin.

The output select block selects, by using the BEEP0SEL flag, whether the P0B3/BEEP pin is used as a generalpurpose I/O port pin or BEEP output pin. The BEEP0CK0 and BEEP0CK1 flags of the BEEP clock select register are used to select whether the output frequency of the P0B3/BEEP pin is 1.5 kHz or 3 kHz, or the output level of the BEEP pin. The clock generation block generates the 1.5-kHz or 3-kHz clock to be output to the P0B3/BEEP pin.

Figure 18-2 shows the configuration and function of the BEEP clock select register.



Figure 18-1. Outline of BEEP

Namo	F	lag	symbo	ol	Addroop	Read/						
Name	bз	b <sub>2</sub>	b1	bo	Address	Write						
		В	В	В	(BANK15)							
		E	E	Е	14H							
BEEP		Е	E	Е								
clock select	0	P	P	P		R/W						
register												
		E K	ĸ									
		L	1	0								
		<u> </u>										
				-		Setting o	f BEEP pin (when BEEP0SEL=1)					
			0	0	Outputs low level							
			0	1	Outputs high	Dutputs high level						
			1	0	Outputs 1.5 k	Outputs 1.5 kHz clock						
			1	1	Outputs 3 kH	z clock						
			_	•		Selecting g	eneral-purpose I/O port and BEEP					
		0			Uses P0B3/B	EEP pin as ge	neral-purpose I/O port					
					Uses P0B3/BEEP pin as BEEP							
					L							
					Fixed to 0							

## Figure 18-2. Configuration and Function of BEEP Clock Select Register

eset	Reset by RESET pin	0	0	0	0
At re	WDT&SP reset	0	0	0	0
Clo	ock stop	0	0	0	0

# NEC

# 18.2 Output Wave Form of BEEP

(1) Output wave of f = 1.5 kHz and f = 3 kHz



#### Example Program to output 3-kHz clock from P0B3/BEEP pin

BANK1		; Same as MOV BANK, #0001B
MOV	14H, #0011B	; Writes 0011B to data memory address 14H
		; Outputs 3 kHz from BEEP pin

(2) Maximum time until clock is output from P0B3/BEEP pin after instruction execution



## (3) Minimum time until clock is output from P0B3/BEEP pin after instruction execution



## 18.3 Status at Reset

# 18.3.1 At reset by RESET pin

The P0B3/BEEP pin is set in the general-purpose input port mode.

## 18.3.2 WDT&SP reset

P0B3/BEEP pin is set in the general-purpose input port mode.

#### 18.3.3 On execution of clock stop instruction

The P0B3/BEEP pin is set in the general-purpose input/output port mode.

#### 18.3.4 In halt status

The P0B3/BEEP output pin retains the previous status.

# 19. LCD CONTROLLER/DRIVER

The LCD (Liquid Crystal Display) controller/driver can display an LCD of up to 60 dots by a combination of command signal and segment signal outputs.

# 19.1 Outline of LCD Controller/Driver

Figure 19-1 outlines the LCD controller/driver.

The LCD controller/driver can be used to display up to 60 dots by using a combination of common signal output pins (COM0 through COM3) and segment signal output pins (LCD0 through LCD19).

The drive mode is 1/4 duty, 1/2 bias, the frame frequency is 62.5 Hz, and drive voltage is VLCD.

Among the segment signal output pins, LCD17 through LCD19 can be used as a general-purpose input port. For details of the general-purpose input port, refer to **11.3 GENERAL-PURPOSE INPUT PORT (P0D, P1C, P2A)**.



Figure 19-1. Outline of LCD Controller/Driver

Remarks 1. LCDEN (bit 0 of LCD driver display start register: refer to Figure 19-8) turns ON/OFF all LCD display.

2. LCD19SEL-LCD17SEL (bits 2-0 of LCD port select register: refer to Figure 19-6)

\*

#### **19.2 LCD Drive Voltage Generation Block**

The LCD drive voltage generation block generates a voltage to drive the LCD.

The µPD17934 supplies the LCD drive voltage from an external doubler circuit. To configure a doubler circuit, connect a capacitor to the CAPLCD0, CAPLCD1, CAPLCD2, CAPLCD3, REGLCD0, REGLCD1, and REGLCD2 pins.

Figure 19-2 shows an example of configuration of the doubler circuit. To use a voltage of 3.0 V (TYP.), connect as shown in Figure 19-2.

To operate the doubler circuit, the LCDEN flag of the LCD display start register must be set to "1". Unless this flag is set to "1", the LCD drive voltage generation block does not operate. For the LCDEN flag, refer to **19.5 Common Signal Output and Segment Signal Output Timing Control Blocks**.

#### Figure 19-2. Configuration of Doubler Circuit



Remark (): pin number

Note that, because of the configuration of the doubler circuit, the values of the LCD drive voltages (VLCD1 and VLCD0) differ if the values of C1, C2, C3, C4, and C5 are changed.

# 19.3 LCD Segment Register

The LCD segment register sets dot data to turn on or turn off dots on the LCD.

Figure 19-3 shows the location in the data memory and configuration of the LCD segment register.

Because the LCD segment register is located in data memory, it can be controlled by all the data memory manipulation instructions.

One nibble of the LCD segment register can set display data of 4 dots (data to turn dots on or off). If the LCD segment register is set to "1" at this time, the LCD display dot is on; the dot goes off if the register is set to "0".

Figure 19-4 shows the relation between the LCD segment register and LCD display dot.

## Figure 19-3. Location on Data Memory and Configuration of LCD Segment Register



	LCD segment register (BANK 14)																			
Address	5CH	5DH	5EH	5FH	60H	61H	62H	63H	64H	65H	66H	67H	68H	69H	6AH	6BH	6CH	6DH	6EH	6FH
Symbol	LCDD19	LCDD18	LCDD17	LCDD16	LCDD15	LCDD14	LCDD13	LCDD12	LCDD11	LCDD10	LCDD9	LCDD8	LCDD7	LCDD6	LCDD5	LCDD4	LCDD3	LCDD2	LCDD1	LCDD0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

		· · · · · ·	
	LCD	D19	
b₃	b2	b1	bo
	R/	Ŵ	

	LCD segment register																																			
Address		5C	Н			5E	ЭН			5E	Н		5FH 60H			61H 62H (			(	$\left( \right)$	6E	H			6F	Ή										
Symbol		LCD	D19	)		LCD	D18	3	I	LCD	D17	,		LCD	D16	5	LCDD15			LCDD14			LCDD13			$\sum_{i=1}^{n}$	LCDD1			LCDD0						
Bit	bз	b2	b1	bo	bз	b2	b1	bo	bз	b2	b1	bo	bз	b2	b1	bo	bз	b <sub>2</sub>	b1	bo	bз	b2	b1	bo	bз	b2	b1	bo	ba	b2	b1	bo	bз	b2	b1	bo
Display dot	A	в	С	D	A	в	С	D	Е	F	G	Н	А	в	С	D	Е	F	G	н	А	В	С	D	Е	F	G	H)	A	В	С	D	Е	F	G	н
COM3 pin			A	1	1	(	A				E			(	A		I		E				A			(	E	$\left\{ \right\}$	$\left\{ \right.$	(	A				E	
		)	_				-				_				_				_				_				_	$\left\{ \right.$	$\left\{ \right.$		_			)	_	
COM2 pin			B			-(	В			-(	F			-(	В			-(	F				<u>В</u>			-(	F	$\neg$	$\left\{ \right.$		В			-(	→ F	
COM1 pin			c			-(	c			-0	G			-(	c				G				)c			-(	G	$\left  \right\rangle$	$\left\langle \right\rangle$	-(	c			-(	G	
COM0 pin			D				P			-0	H )			-(	D				н Э				D			-(	н		$\left\{ \right\}$		D				→ H	
	LCD	19/P2	 2A2 p	vin	LC	D18/	  P2A1	l pin	LCI	 017/P	2A0	pin	L	.CD1	  6 pi	n	L	CD1	5 pi	n	L	.CD1	4 pi	n	L	CD	 13 pi	in )	5	LCD	 1 pir	۱	L	.CD	0 pin	۱

# Figure 19-4. Relation between LCD Segment Register and LCD Display Dot

199

# 19.4 Segment Signal/General-Purpose Input Port Select Block

Figure 19-5 shows the configuration of the segment signal/general-purpose input port select block.

This block specifies, by using the LCD19SEL through LCD17SEL flags of the LCD port select register, whether each pin is used as a segment signal output pin or a general-purpose input port pin. When each flag is "1", the corresponding pin is used as a segment signal output pin; when it is "0", the pin is used as a general-purpose input port pin.

Figure 19-6 shows the configuration of the LCD port select register.





**Note** The LCD19/P2A2 pin corresponds to the LCD19SEL flag and bit 2 of P2A, the LCD18/P2A1 pin, to the LCD18SEL flag and bit 1 of P2A, and the LCD17/P2A0 pin, to the LCD17SEL flag and bit 0 of P2A, respectively.

	F	lag s	ymbo	ol	Address	Read/	
Name	bз	b2	b1	bo	Address	Write	
LCD port select register	0	L C D 1 9 S E L	L C D 1 S E L	L C D 1 7 S E L	69H	R/W	
							1
					Selects LC	D segment s	signal output pin or general-purpose input port pin
				0	Uses LCD <sup>2</sup>	17/P2A0 pin	as general-purpose input port pin.
				1	Uses LCD <sup>2</sup>	17/P2A0 pin	as LCD segment pin.
				-	Selects LC	D segment s	ignal output pin or general-purpose input port pin
			0		Uses LCD <sup>2</sup>	18/P2A1 pin	as general-purpose input port pin.
			1		Uses LCD <sup>2</sup>	18/P2A1 pin	as LCD segment pin.
			L	ı			
				-	Selects LC	D segment s	signal output pin or general-purpose input port pin
		0	1 1 1		Uses LCD <sup>2</sup>	19/P2A2 pin	as general-purpose input port pin.
		1	     		Uses LCD	19/P2A2 pir	n as LCD segment pin.
			,				
				•	Fixed to "0	".	

# Figure 19-6. Configuration of LCD Port Select Register

eset	Reset by $\overline{\text{RESET}}$ pin	(	)	0	0	0
At re	WDT & SP reset			0	0	0
Clo	ck stop	,		R	R	R

R: Retained

# 19.5 Common Signal Output and Segment Signal Output Timing Control Blocks

Figure 19-7 shows the common signal output and segment signal output timing control blocks.

The common signal output timing control block controls the common signal output timing of the COM0 through COM3 pins.

The segment signal output timing control block controls the segment signal output timing of the LCD0 through LCD19 pins.

The common and segment signals are output when the LCDEN flag of the LCD driver display start register is set to "1".

When this flag is reset to "0", all the LCD display dots can be extinguished (refer to Figure 19-8).

When LCD display is not carried out, the COM0 through COM3 and LCD0 through LCD19 pins output low level.

## Figure 19-7. Configuration of Common Signal Output and Segment Signal Output Timing Control Blocks



#### ★

Figure 19-8. Configuration of LCD Driver Display Start Register



∢	WD1&SI leset	
Clo	ock stop	0 0

Remark R: Retained

Note Bit 3 of the LCD display start register is a test mode area. Therefore, do not write "1" to this bit.

#### 19.6 Common Signal and Segment Signal Output Waves

Figure 19-9 shows an example of the common signal and segment signal output waves.

0 ¦ 0

The  $\mu$ PD17934 outputs a signal with a frame frequency of 62.5 Hz using a 1/4 duty, 1/2 bias (voltage average method) drive mode.

As the common signals, the COM0 through COM3 pins output three levels of voltages (GND, VLCD0, and VLCD1) each having a phase difference of 1/8 from the others. In other words, voltages of  $\pm 1/2$ VDD are output with the VLCD0 as the reference. This display method is called the 1/2 bias drive method.

As the segment signals, the segment signal output pins output voltages of two levels (GND and V<sub>LCD1</sub>) having a phase corresponding to each display dot. Because one segment pin can turn on or off four display dots (A, B, C, and D) as shown in Figure 19-9, sixteen phases can be output by combining lighting and extinguishing of each dot.

Each display dot turnd on when the potential difference between a common signal and a segment signal is  $V_{LCD1}$ . In other words, the duty factor at which each display dot turns on is 1/4.

This display method is called the 1/4 duty display method, and the frame frequency is 62.5 Hz.



D

COM3 pin

# Figure 19-9. Common Signal and Segment Signal Output Waveforms

Each segment signal output pin (LCDn pin)



# 19.7 Using LCD Controller/Driver

Figure 19-10 shows an example of wiring of an LCD panel using the pins LCD0 through LCD14.

An example of a program that lights the 7 segments connected to LCD0 and LCD1 pins shown in Figure 19-10 is given below.

#### Example

PMN0	MEM	0.01H	; Preset number storage area
СН	FLG	LCDD0.3	; Defines symbol with high-order 1 bit of LCD0 register for 'CH' display
LCDDATA			; LCD segment table data
	DW	000000000000000B	; BLANK
	DW	000000000000110B	; 1
	DW	0000000010110101B	; 2
	DW	000000010100111B	; 3
	DW	000000001100110B	; 4
	DW	0000000011100011B	; 5
	DW	0000000011110011B	; 6
	DW	000000010000110B	; 7
	DW	0000000011110111B	; 8
	DW	0000000011100111B	; 9
	MOV	AR0, #.DL.LCDDATA SHR 12	AND 0FH
	MOV	AR1, #.DL.LCDDATA SHR 8	AND 0FH
	MOV	AR2, #.DL.LCDDATA SHR 4	AND 0FH
	MOV	AR3, #.DL.LCDDATA	AND 0FH
	LD	DBF0, AR0	
	LD	DBF1, AR1	
	LD	DBF2, AR2	
	LD	DBF3, AR3	
	ADD	DBF0, PMN0	
	ADDC	DBF1, #0	
	ADDC	DBF2, #0	
	ADDC	DBF3, #0	
	ST	AR0, DBF0	
	ST	AR1, DBF1	
	ST	AR2, DBF2	
	ST	AR3, DBF3	
	MOVT	DBF, @AR	; Table reference instruction
	BANK1		
	ST	LCDD0, DBF0	
	ST	LCDD1, DBF1	
	SET1	CH	
	SET1	LCDEN	; LCD ON



## Figure 19-10. Example of Wiring of LCD Panel (When LCD0-LCD14 Pins Are Used)

Correspondence of Segment and Common Pins, and LCD Panel Display (When LCD0-LCD14 Pins Are Used)

Segment Pin Common	L C D 14	L C D 13	L C D 12	L C D 11	L C D 10	L C D 9	L C D 8	L C D 7	L C D 6	L C D 5	L C D 4	L C D 3	L C D 2	L C D 1	L C D 0
СОМЗ	FM		В	1a		2a	:	За	С	4a	D	E	AM	5a	СН
COM2	MW		A	1f	1b	2f	2b	Зf	3b	4f	4b	F	PM	5f	5b
COM1	SW	[]	$\triangleright$	1g	1c	2g	2c	Зg	Зс	4g	4c	G	MHz	5g	5c
СОМ0	LW	·	$\triangleleft$	1e	1d	2e	2d	Зе	3d	4e	4d		kHz	5e	5d

## 19.8 Status at Reset

# 19.8.1 At reset by RESET pin

The LCD0 through LCD16 pins output a low level.

The LCD17/P2A0 through LCD19/P2A2 pins are set in the general purpose input port.

The COM0 through COM3 pins also output a low level.

Therefore, the LCD display is OFF.

The contents of the LCD segment register are undefined.

# 19.8.2 WDT&SP reset

The LCD0 through LCD16 pins output a low level. The LCD17/P2A0 through LCD19/P2A2 pins are set in the general-purpose input port. The COM0 through COM3 pins output a low level. Therefore, the LCD display is OFF. The contents of the LCD segment register are undefined.

# 19.8.3 On execution of clock stop instruction

The LCD0 through LCD16 pins output a low level. The LCD17/P2A0 through LCD19/P2A2 pins are set in the general purpose input port. The COM0 through COM3 pins also output a low level. Therefore, the LCD display is OFF. The LCD segment register retains the previous contents.

# 19.8.4 In halt status

The LCD0 through LCD19 pins output segment signals. The COM0 through COM3 pins output common signals. The LCD segment register retains the previous contents.

# 20. STANDBY

The standby function is used to reduce the current consumption of the device while the device is backed up.

## 20.1 Outline of Standby Function

Figure 20-1 outlines the standby block.

The standby function reduces the current consumption of the device by partly or totally stopping the device operation.

The following two types of standby functions are available for selection as the application requires.

- Halt function
- Clock stop function

The halt function reduces the current consumption of the device by stopping the CPU operation by using a dedicated instruction "HALT h".

The clock stop function reduces the current consumption of the device by stopping the oscillation of the oscillation circuit by using a dedicated instruction "STOP s".



Figure 20-1. Outline of Standby Block

## 20.2 Halt Function

#### 20.2.1 Outline of halt function

The halt function stops the operating clock of the CPU by executing the "HALT h" instruction.

When this instruction is executed, the program is stopped until the halt status is later released. Therefore, the current consumption of the device in the halt status is reduced by the operating current of the CPU.

The halt status is released by using basic timer 0 carry FF, interrupt, or port input (P0D).

The release condition is specified by operand "h" of the "HALT h" instruction.

#### 20.2.2 Halt status

In the halt status, all the operations of the CPU are stopped. In other words, execution of the program is stopped at the "HALT h" instruction. However, the peripheral hardware units continue the operation specified before execution of the "HALT h" instruction.

For the operation of each peripheral hardware unit, refer to 20.4 Device Operation in Halt and Clock Stop Status.

#### 20.2.3 Halt release condition

Figure 20-2 shows the halt release condition.

The halt release condition is specified by 4-bit data specified by operand "h" of the "HALT h" instruction.

The halt status is released when the condition specified by "1" in operand "h".

When the halt status is released, program execution is started from the instruction after the "HALT h" instruction. If the halt status is released by an interrupt, the operation to be performed after the halt status has been released differs depending on whether the interrupts are enabled (EI status) or disabled (DI status) when an interrupt source (IRQxxx = 1) is issued with the interrupt (IPxxx = 1) enabled.

If two or more releasing conditions are specified, the halt status is released when one of the specified condition is satisfied.

If 0000B is set as halt release condition "h", no releasing condition is set. If the device is reset at this time, the halt status is released.



#### Figure 20-2. Halt Release Condition

## 20.2.4 Releasing halt by input port (P0D)

The halt releasing condition using an input port is specified by the "HALT 0001B" instruction. When the halt releasing condition using an input port is specified, the halt status is released if a high level is input to one of the P0D0 through P0D3 pins.

The P0D0 through P0D3 pins are multiplexed with the A/D converter input pins AD0 through AD2 (except P0D0) and the halt status is not released when these pins are used as A/D converter input pins.

An example is given below.

## • To use as key matrix

The P0D0 through P0D3 pins are general-purpose input port pins which can be set in the input or output mode in 1-bit units and can be connected to an internal pull-down resistor. If connection of the internal pull-down resistor is specified by software, an external resistor can be eliminated as shown in this example (the internal-pull down resistor is connected at reset by the  $\overrightarrow{\text{RESET}}$  pin.)



The "HALT 0001B" instruction is executed after the general-purpose output ports for key source signal are made high. Note that if an alternate switch is used as shown by switch A in the above figure, the halt status is released immediately because a high level is input to the P0D0 pin while switch A is closed.

#### 20.2.5 Releasing halt status by basic timer 0 carry FF

Releasing the halt status by using the basic timer 0 carry FF is specified by the "HALT 0010B" instruction.

When releasing the halt status by the basic timer 0 carry FF is specified, the halt status is released as soon as the basic timer 0 carry FF has been set to 1.

The basic timer 0 carry FF corresponds to the BTM0CY flag on a one-to-one basis and is set at fixed time intervals (125 ms). Therefore, the halt status can be released at fixed time intervals.

Example To release halt status every 125 ms to execute processing A

HLTTMR LOOP:	DAT	0010B	; Symbol definition
	HALT	HLTTMR	; Specifies setting of basic timer 0 carry FF as halt releasing condition
	SKT1	BTMOCY	; Embedded macro
	BR	LOOP	; Branches to LOOP if BTM0CY flag is not set
	Process	ing A	; Executes processing A if carry occurs
	BR	LOOP	

#### 20.2.6 Releasing halt status by interrupt

Releasing the halt status by an interrupt is specified by the "HALT 1000B" instruction.

When releasing the halt status by an interrupt is specified, the halt status is released as soon as the interrupt has been accepted.

Many interrupt sources are available as described in 12. INTERRUPTS. Which interrupt source is used to release the halt status must be specified in advance in software.

To accept an interrupt, each interrupt request must be issued from each interrupt source and each interrupt must be enabled (by setting the corresponding interrupt enable flag).

Therefore, the interrupt is not accepted even if the interrupt request is issued, and the halt status is not released. When the halt status is released by accepting an interrupt, the program flow branches to the vector address of the

interrupt.

When the RETI instruction is executed after interrupt servicing, the program flow is restored to the instruction after the HALT instruction.

If all the interrupts are disabled (DI status), the halt status is released by enabling an interrupt (IPxxx = 1) and issuing an interrupt source (IRQxxx = 1), and the flow of the program goes to the instruction after the HALT instruction.

## Example Releasing halt status by timer 0 and INT pin interrupts

In this example, the halt status is released and processing B is executed when timer 0 interrupt is accepted. And processing A is executed when INT pin interrupt is accepted. Each time the halt status has been released, processing C is executed.

HLTINT	DAT	1000B	; Symbol definition
START:			; Address 0000H
	BR	MAIN	
;*** Interrup	ot vector ac	ldress ***	
	NOP		; SIO2
	NOP		; Basic timer 1
INTP:	BR	ΙΝΤΤΜΟ	; Branches to timer 0 interrupt processing ; Branches to INT pin interrupt processing ; INT pin interrupt vector address (0004H)
	Proces	ssing A	; INT pin interrupt processing
	EI		
	RETI		
INTTMO:			
	Proces	ssing B	; Timer 0 interrupt processing
	EI		
	RETI		
MAIN:			
	INITFLG	TMOCK1, TM0CK0	; Sets timer 0 count clock to 40 $\mu$ s
	MOV	DBF1, #0	
	MOV	DBF0, #32H	
	PUT	TM0M,DBF	; Sets time interval of timer 0 interrupt to 2 ms
	SET2	TMORES, TMOEN	; Resets and starts timer 0
	SET2	IPTM0, IP0	; Enables INT pin and timer 0 interrupts
LOOP:			
	Proce	essing C	; Main routine processing
	EI		; Enables all interrupts
	HALT	HLTINT	; Specifies releasing halt status by interrupt
;<1>	•		
	BR	LOOP	

If the INT pin interrupt request and timer 0 interrupt request are issued simultaneously in the halt status, processing A for the INT pin, which has the higher hardware priority, is executed.

After execution of processing A and when "RETI" is executed, the program branches to the "BR LOOP" instruction of <1>. However, the "BR LOOP" instruction is not executed, and timer 0 interrupt is immediately accepted.

When the "RETI" instruction is executed after processing B of timer 0 interrupt has been executed, the "BR LOOP" instruction is executed.

Caution To reset the interrupt request flag (IRQxxx) once before the halt instruction is executed, insert a NOP instruction (or one or more other instructions) between the HALT instruction and the instruction that resets the interrupt request flag (IRQxxx) as shown below. If a NOP instruction (or one or more other instructions) is not inserted, the interrupt request flag is not reset, and therefore, the halt status is released immediately.

#### Example

:		; IRQxxx is set at certain timing
CLR1 NOP	IRQxxx	<ul> <li>; Resets IRQxxx flag once</li> <li>; Resets IRQxxx flag at this timing</li> <li>; Unless this period is missing, the IRQxxx flag is not reset,</li> <li>; and the next HALT instruction is immediately released</li> </ul>
HALT	1000B	;

## 20.2.7 If two or more releasing conditions are specified at same time

If two or more halt releasing conditions are specified at same time, the halt status is released when one of the conditions is satisfied.

The following program example shows how the releasing conditions are identified if two or more conditions are satisfied at the same time.

#### Example

	HLTINTP HLTBTM HLTP0D	DAT DAT DAT	1000B 0010B 0001B	
START:			0.730	
·*** Interrun	DR t vector addr	WIAIN		
, monup	NOP	000		: SI0
	NOP			; Basic timer 1
	NOP			; TMO
	NOP			; INT
INTP:				; INT pin interrupt vector address (0004H)
	Process	sing A		; INT pin interrupt processing
	EI RETI			
BTMOUP:				; Timer carry FF processing
	Process	sing B		
	RET			
PODP:				; POD input processing
	Process	sing C		
	RET			
MAIN:	SET1	IP0		: Enables INT pin interrupt
	EI			
LOOP:				
				; Selects interrupt, timer carry FF (125 ms), and P0D input as halt releasing
	SKF1	BTMOCY		· Detects BTM0CY flag
	CALL	BTMOUP		; Timer carry FF processing if flag is set to 1
	SKF	P0D, 1111	В	; Detects P0D input
	CALL	PODP		; Port input processing if P0D is high
	BR	LOOP		
In the above example, three halt status releasing conditions, INT pin interrupt, 125-ms basic timer 0 carry FF, and port 0D input, are specified.

To identify which condition has released the halt status, a vector address (interrupt), BTM0CY flag (timer carry FF), and port register (port input) are detected.

To use two or more releasing conditions, the following two points must be noted.

- When the halt status is released, all the specified releasing conditions must be detected.
- The releasing condition with the higher priority must be detected first.

#### 20.3 Clock Stop Function

#### 20.3.1 Outline of clock stop function

The clock stop function stops the oscillation circuit of a 75-kHz crystal resonator by executing the "STOP s" instruction (clock stop status).

Therefore, the current consumption of the device is reduced to 10  $\mu$ A MAX (T<sub>A</sub> = -10 to +50 °C, V<sub>DD</sub> = 1.05 to.1.8 V)

#### 20.3.2 Clock stop status

In the clock stop status, all the device operations of the CPU and peripheral hardware units are stopped because the generation circuit of the crystal resonator is stopped.

For the operations of the CPU and peripheral hardware units, refer to **20.4 Device Operation in Halt and Clock Stop Status**.

#### 20.3.3 Releasing clock stop status

Figure 20-3 shows the stop status releasing conditions.

The stop status releasing condition is specified by 4-bit data specified by operand "s" of the "STOP s" instruction. The stop status is released when the condition specified by "1" in operand "s" is satisfied.

When the stop status has been released, a halt period which is half the time  $(t_{SET}/2)$  specified by the basic timer 0 clock selection register as oscillation circuit stabilization wait time has elapsed, and the program execution is started from the instruction next to the "STOP s" instruction. If releasing the stop status by an interrupt is specified, however, the program operation after the stop status has been released differs depending on whether the interrupt is enabled (EI status) or disabled (DI status) when an interrupt source is issued (IRQxxx = 1) with the interrupt enabled (IPxxx = 1).

If all the interrupts are enabled (EI status), the stop status is released when the interrupt is enabled (IPxxx = 1) and the interrupt source is issued (IRQxxx = 1), and the program flow returns to the instruction next to the STOP instruction.

If all the interrupts are disabled (DI status), the stop status is released when the interrupt is enabled (IPxxx = 1) and the interrupt resource is issued (IRQxxx = 1), and the program flow returns to the instruction next to the STOP instruction.

If two or more releasing conditions are specified at one time, and if one of the conditions is satisfied, the stop status is released.

If 0000B is specified as stop releasing condition "s", no releasing condition is satisfied. If the device is reset at this time the stop status is released.

\*

NEC



#### Figure 20-3. Stop Releasing Conditions

### 20.3.4 Releasing clock stop status by high level input of port 0D

Figure 20-4 illustrates how the clock stop status is released by the high level input to port 0D.





tset: basic timer 0 setting time

### 20.4 Device Operation in Halt and Clock Stop Status

Table 20-1 shows the operations of the CPU and peripheral hardware units in the halt and clock stop status.

In the halt status, all the peripheral hardware units continue the normal operation until instruction execution is stopped.

In the clock stop status, all the peripheral hardware units stop operation.

The control registers that control the operations of the peripheral hardware units operate normally (not initialized) in the halt status, but are initialized to specified values when the clock stop instruction is executed.

In other words, all peripheral hardware continues the operation specified by the control register in the halt status, and the operation is determined by the initialized value of the control register in the clock stop status.

For the values of the control registers in the clock stop status, refer to **8. REGISTER FILE (RF) AND CONTROL REGISTER**.

Peripheral Hardware	Status					
	Halt	Clock stop				
Program counter	Stops at address of HALT instruction	Stops at address of STOP instruction				
System register	Retained	Retained				
Peripheral register	Retained	Partly initialized <sup>Note</sup>				
Control register	Retained	Partly initialized <sup>Note</sup>				
Timer	Normal operation	Operation stops				
PLL frequency synthesizer	Normal operation	Operation stops				
A/D converter	Normal operation	Operation stops				
Serial interface	Stops operation when internal clock (master) is selected and continues operation when external clock (slave) is selected	Stops operation and used as general- purpose I/O port				
Frequency counter	Normal operation	Stops operation and used as general- purpose input port				
BEEP output	Normal operation	Stops operation and used as general- purpose I/O port				
LCD controller/driver	Normal operation	Stops operation				
General-purpose I/O port	Normal operation	Retained				
General-purpose input port	Normal operation	Input port				
General-purpose output port	Normal operation	Retains output latch				

Table 20-1. Device Operation in Halt and Clock Stop Status

Note For the value to which these registers are initialized, refer to 5. SYSTEM REGISTER (SYSREG) and 8. REGISTER FILE (RF) AND CONTROL REGISTER.

#### 20.5 Cautions on Processing of Each Pin in Halt and Clock Stop Status

The halt status is used to reduce the current consumption when, say, only the watch is used. The clock stop function is used to reduce the current consumption of the device to only use the data memory. Therefore, the current consumption must be reduced as much as possible in the halt status or clock stop status. At this time, the current consumption significantly varies depending on the status of each pin, and the points shown

in Table 20-2 must be noted.

Table 20-2. Status of I	Each Pin in Halt and	<b>Clock Stop Status and</b>	Cautions (1/2)
-------------------------	----------------------	------------------------------	----------------

Pin Function Pin Symbo		Pin Symbol	Status of Each Pin and Cautions on Processing			
			Halt status	Clock stop status		
General-	Port 0B	P0B3/BEEP	Retains status before halt	All port pins are set in general-purpose		
purpose		P0B2/SO1		port mode (except P0D3/AD2-P0D1/AD0,		
I/O port		P0B1/SI1/SO2	(1) When specified as output pin	P2A2/LCD19-P2A0/LCD17)		
		P0B0/SCK	Current consumption increases if pin	Input or output mode of general-purpose		
	Port 1A	P1A3	is externally pulled down while it	I/O port set before clock stop status is		
		P1A2	outputs high level, or externally pulled	retained.		
		P1A1	up while it outputs low level.			
		P1A0	Exercise care in using N-ch open-	(1) When specified as general-purpose		
	Port 1D	P1D3	drain output (P1A3-P1A0, P1D3-	output port		
		P1D2	P1D0)	Current consumption increases due		
		P1D1		to noise if pin is floated		
		P1D0	(2) When specified as input pin			
	Port 2B	P2B3-P2B0	Current consumption increases due	(2) When specified as general-purpose		
	Port 2C	P2C3-P2C0	to noise if pin is floated	input port		
General-	Port 0D	P0D3/AD2		Current consumption does not		
purpose		P0D2/AD1	(3) Port 0D (P0D3/AD1-P0D1/AD0,	increase due to noise even if pin is		
input port		P0D1/AD0	P0D0)	floated		
		P0D0	Current consumption increases if pin			
	Port 1C	P1C3/FMIFC/AMIFC	is externally pulled up because it is	(3) P0D3/AD2-P0D1/AD0		
		P1C2/AMIFC	provided with pull-down resistor	Pin used for A/D converter is retained		
		P1C1/TM1	selectable by software	as is.		
		P1C0/TM0		Pull-down resistor of P0D3- P0D0 pin		
	Port 2A	P2A2/LCD19	(4) Port 1C (P1C3/FMIFC/AMIFC,	retains previous status		
		P2A1/LCD18	P1C2/AMIFC, P1C1/TM1,			
		P2A0/LCD17	P1C0/TM0)	(4) P2A2/LCD19-P2A0/LCD17		
General-	Port 0A	P0A1	When P1C2/AMIFC or P1C3/FMIFC/	Pin used for LCD segment is		
purpose		P0A0	AMIFC pin is used for IF counter,	retained as is.		
output port	Port 0C	P0C3-P0C0	current consumption increases			
			because internal amplifier operates	Specified as general-purpose output port.		
				is externally nulled down while it output		
				high level or externelly pulled down while it outputs		
				nightevel or externally pulled up while it		
				increases		
				Increases		

 $\star$ 

Pin Function	Pin Symbol	Status of Each Pin and Cautions on Processing			
		Halt status	Clock stop status		
External interrupt	INT	Current consumption increases due to no	bise if pin is floated		
PLL frequency	VCOL	Current consumption increases during PLL	PLL is disabled		
synthesizer	VCOH	operation.			
	EO0	When PLL is disabled, pin is in following	VCOH, VCOL: internally pulled down		
	EO1	status:	EO1, EO0 : floated		
		VCOH, VCOL: internally pulled down			
		EO1, EO0 : floated			
Crystal oscillation	XIN	Current consumption changes due to	XIN pin is internally pulled down, and XOUT		
circuit	Хоит	oscillation waveform of crystal oscillation	pin outputs high level		
		circuit.			
		The higher oscillation amplitude, the lower			
		current consumption.			
		Oscillation amplitude must be evaluated			
		because it is influenced by crystal resonator			
		or load capacitor used			
LCD controller/driver	LCD19/P2A2	(1) When LCD19/P2A2-LCD17/P2A0 are	(1) When LCD19/P2A2-LCD17/P2A0 are		
	LCD18/P2A1	used as general-purpose input port	used as general-purpose input port		
	LCD17/P2A0	pins	pins		
	LCD16	When these pins are used as general-	When these pins are used as general-		
		purpose input port pins, the same	purpose input port pins, the same		
	LCD0	points as described above must be	points as described above must be		
	СОМЗ	noted.	noted.		
	COM0	(2) When LCD controller/driver is used	(2) When LCD controller/driver is used		
		(LCDEN = 1)	LCDEN = 0		
		LCD19-LCD0: Output segment	LCD16-LCD0: Output low level		
		signals	COM3-COM0: Output low level		
		COM3-COM0: Output common			
		signals			
		(3) LCD display off (LCDEN = 0)			
		LCD19-LCD0 : Output segment			
		signals			
		COM3-COM0: Output common			
		signals			

### Table 20-2. Status of Each Pin in Halt and Clock Stop Status and Cautions (2/2)

### 21. RESET

### 21.1 Outline of Reset

The reset function is used to initialize the device. The  $\mu$ PD17934 can be reset in the following ways:

- Reset by RESET pin
- WDT&SP reset

\*





# NEC

★

# ★ 21.2 Reset by RESET Pin

When a low level is input to the RESET pin, an internal reset signal is generated.

At this point, the program counter, stack, system registers, and control registers are initialized (for the initial value, refer to the description of each register).

When the  $\overline{\text{RESET}}$  pin is raised next time, the program starts from address 0 at the rising edge of the basic timer 0 carry FF setting signal 125 ms after a high level has been input to the  $\overline{\text{RESET}}$  pin.

If reset is executed by the RESET pin during program execution, the data in the data memory may be lost.

Figure 21-2. Reset Operation by RESET Pin



### 21.3 WDT&SP Reset

WDT&SP reset includes the following:

- Watchdog timer reset
- Stack pointer overflow/underflow reset





#### 21.3.1 Watchdog timer reset

The watchdog timer is a circuit that generates a reset signal when the execution sequence of the program is abnormal (hung-up).

Hanging-up means that the program jumps to an unexpected routine due to external noise, entering a specific infinite loop and causing the system to be deadlocked. By using the watchdog timer, the program can be restored from this hang-up status because a reset signal is generated from the watchdog timer at fixed time intervals and program execution is started from address 0.

The watchdog timer does not function in the clock stop mode and halt mode.

Resetting by the watchdog timer initializes all the registers except the stack overflow selection register, watchdog timer counter reset register, and basic timer 0 carry register.

The watchdog timer reset is detected by the WDTSPRES flag (R&Reset).

### 21.3.2 Watchdog timer setting flags

These flags can be set only once after power-ON reset on power application or reset by the RESET pin. The WDTCK0 and WDTCK1 flags select an interval at which the reset signal is output. The reference time can be selected to the following three conditions:

- 4096 instructions
- 8192 instructions
- Watchdog timer not set

On power application, 8192 instructions are selected.

If the reset signal generation interval is specified to be 8192 instructions, the watchdog timer FF must be reset at intervals not exceeding 8192 instructions. The valid reset period is from 1 to 8192 instructions.

If the reset signal generation interval is 4096 instructions, the watchdog timer FF must be reset at intervals not exceeding 4096 instructions. The valid reset period is from 1 to 4096 instructions.

#### Figure 21-4. Configuration of Watchdog Timer Clock Selection Register

Name	FI	ag s	symb	ol	Address Read/Write		
	bз	b2	b1	bo			
Watchdog timer	0	0	W	W	(BANK15)	R/W <sup>Note</sup>	
clock selection		1	D	D	02H		
			т	т			
			С	С			
		1 1 1	ĸ	к			
		1	1	0			
						Selects	clock of watchdog timer
			0	0	Does not set w	vatchdog timer	
			0	1	4096 instructio	ons	
			1	0	Setting prohibi	ited	
			1	1	8192 instructio	ons	
					Fixed to "0"		
Reset by RESET pin	0	0	1	1			
	$  ^{-}$	i [[					

eset	Reset by RESET pin	0	)	0	)	1	1
At re	WDT&SP reset					Reta	lined
Clo	ock stop		,			Reta	lined

Note Can be written only once.

The WDTRES flag is used to reset the watchdog timer counter.

U

When this flag is set to 1, the watchdog timer counter is automatically reset.

If the WDTRES flag is set to 1 once within a reference time in which the WDTCK0 and WDTCK1 flags are set, the reset signal is not output by the watchdog timer.

### Figure 21-5. Configuration of Watchdog Timer Counter Reset Register



Clock stop U: Undefined

### 21.3.3 Stack pointer overflow/underflow reset

A reset signal is generated if the address or interrupt stack overflows or underflows.

Stack pointer overflow/underflow reset can be used to detect a program hang-up in the same manner as watchdog timer reset.

The reset signal is generated under the following conditions:

- Interrupt due to overflow or underflow of interrupt stack (4 levels)
- Interrupt due to overflow or underflow of address stack (15 levels)

Reset by stack pointer overflow or underflow initializes all the registers, except the stack overflow selection register, watchdog timer counter reset register, and basic timer 0 carry register.

Generation of stack pointer overflow or underflow reset is detected by the WDTSPRES flag (R&Reset).

#### 21.3.4 Stack pointer setting flag

The stack overflow/underflow reset selection register can be set only once after reset by the RESET pin. This register specifies whether reset by address stack overflow or underflow and reset by interrupt stack overflow or underflow are enabled or disabled.

### Figure 21-6. Configuration of Stack Overflow/Underflow Reset Selection Register

Name	FI	ag s	symb	ol	Address	Read/Write	
	bз	b2	b1	bo			
Stack overflow/underflow	0	0	s	s	(RF)	R/W <sup>Note</sup>	
reset selection		   	P	P	05H		
			R	R			
			S	S			
			E	E			
			L	L			
			1	0			
				-		Selects address	s stack overflow/underflow reset
				0	Disables reset		
				1	Enables reset		
				•		Selects interrup	t stack overflow/underflow reset
				0	Disables reset		
				1	Enables reset		
					Fixed to "0"		

eset	Reset by RESET pin	0	0	1	1
At re	WDT&SP reset			Reta	ained
Clo	ock stop			Reta	ained

**Note** Can be written only once.

#### Flag symbol Name Address Read/Write bз b2 b1 b0 WDT&SP reset 0 0 0 W (BANK15) R&Reset D status detection 16H Т S Ρ R E S Detects occurrence of WDT&SP reset 0 No reset request Reset request 1 Fixed to "0"

Figure 21-7	Configuration	of WDT&SP Re	eset Selection	Register
Figure 21-7.	Connyuration	UI WDIQOF KE	esel Selection	register

eset	Reset by RESET pin	0	0	0	0
At r	WDT&SP reset				1
Clo	ock stop				R

R: Retained

# 22. INSTRUCTION SET

# 22.1 Outline of Instruction Set

b14-b11	b15		0		1	
BIN	HEX		0		I	
0000	0	ADD	r,m	ADD	m,#n4	
0001	1	SUB	r,m	SUB	m, #n4	
0010	2	ADDC	r,m	ADDC	m,#n4	
0011	3	SUBC	r,m	SUBC	m,#n4	
0100	4	AND	r,m	AND	m,#n4	
0101	5	XOR	r,m	XOR	m,#n4	
0110	6	OR	r,m	OR	m,#n4	
0111	7	INC INC RORC MOVT PUSH POP GET PUT PEEK POKE BR CALL RET RETSK RETI EI DI STOP HALT NOP	AR IX r DBF,@AR AR AR DBF,p p,DBF WR,rf rf,WR @AR @AR @AR			
1000	8	LD	r,m	ST	m,r	
1001	9	SKE	m,#n4	SKGE	m,#n4	
1010	А	MOV	@r,m	MOV	m,@r	
1011	В	SKNE	m,#n4	SKLT	m,#n4	
1100	С	BR	addr (page 0)	CALL	addr (page 0)	
1101	D	BR	addr (page 1)	MOV	m,#n4	
1110	E	BR	addr (page 2)	SKT	m,#n4	
1111	F	BR	addr (page 3)	SKF	m,#n	

# 22.2 Legend

AR	:	Address register
ASR	:	Address stack register indicated by stack pointer
addr	:	Program memory address (low-order 11 bits)
BANK	:	Bank register
CMP	:	Compare flag
CY	:	Carry flag
DBF	:	Data buffer
h	:	Halt release condition
INTEF	:	Interrupt enable flag
INTR	:	Register automatically saved to stack when interrupt occurs
INTSK	:	Interrupt stack register
IX	:	Index register
MP	:	Data memory row address pointer
MPE	:	Memory pointer enable flag
m	:	Data memory address indicated by mR, mc
mr	:	Data memory row address (high-order)
mc	:	Data memory column address (low-order)
n	:	Bit position (4 bits)
n4	:	Immediate data (4 bits)
PAGE	:	Page (bits 12 and 11 of program counter)
PC	:	Program counter
Р	:	Peripheral address
рн	:	Peripheral address (high-order 3 bits)
p∟	:	Peripheral address (low-order 4 bits)
r	:	General register column address
rf	:	Register file address
rfr	:	Register file row address (high-order 3 bits)
<b>r</b> fc	:	Register file column address (low-order 4 bits)
SP	:	Stack pointer
S	:	Stop release condition
WR	:	Window register
(x)	:	Contents addressed by x

# 22.3 Instruction List

Instructions	Mnemonic	Operand	Operation	Instruction Code				
				Op code		Operand	1	
Add	ADD	r,m	$(r) \leftarrow (r) + (m)$	00000	МR	mc	r	
		m,#n4	$(m) \leftarrow (m) + n4$	10000	mя	mc	n4	
	ADDC	r,m	$(r) \leftarrow (r) + (m) + CY$	00010	mя	mc	r	
		m,#n4	$(m) \leftarrow (m) + n4 + CY$	10010	mя	mc	n4	
	INC	AR	$AR \leftarrow AR + 1$	00111	000	1001	0000	
		IX	$IX \leftarrow IX + 1$	00111	000	1000	0000	
Subtract	SUB	r,m	$(r) \leftarrow (r) - (m)$	00001	МR	mc	r	
		m,#n4	$(m) \leftarrow (m) - n4$	10001	mя	mc	n4	
	SUBC	r,m	$(r) \leftarrow (r) - (m) - CY$	00011	mк	mc	r	
		m,#n4	$(m) \leftarrow (m) - n4 - CY$	10011	mя	mc	n4	
Logical	OR	r,m	$(r) \leftarrow (r) \vee (m)$	00110	МR	mc	r	
operation		m,#n4	$(m) \leftarrow (m) \vee n4$	10110	mк	mc	n4	
	AND	r,m	$(r) \leftarrow (r) \land (m)$	00100	mя	mc	r	
		m,#n4	$(m) \leftarrow (m) \land n4$	10100	mк	mc	n4	
XOR r,m		r,m	$(r) \leftarrow (r) \forall (m)$	00101	mя	mc	r	
		m,#n4	$(m) \leftarrow (m) \forall n4$	10101	mк	mc	n4	
Judge	SKT	m,#n	$CMP \leftarrow 0$ , if (m) $\land n = n$ , then skip	11110	МR	mc	n	
	SKF	m,#n	$CMP \leftarrow 0$ , if (m) $\land$ n = 0, then skip	11111	mя	mc	n	
Compare	SKE	m,#n4	(m) – n4, skip if zero	01001	mк	mc	n4	
	SKNE	m,#n4	(m) – n4, skip if not zero	01011	mк	mc	n4	
	SKGE	m,#n4	(m) – n4, skip if not borrow	11001	mк	mc	n4	
	SKLT	m,#n4	(m) – n4, skip if borrow	11011	mя	mc	n4	
Rotate	RORC	r	$\blacktriangleright CY \leftarrow (r) \ b_3 \leftarrow (r) \ b_2 \leftarrow (r) \ b_1 \leftarrow (r) \ b_0 \frown$	00111	000	0111	r	
Transfer	LD	r,m	$(r) \leftarrow (m)$	01000	mя	mc	r	
	ST	m,r	$(m) \leftarrow (r)$	11000	МR	mc	r	
	MOV	@r,m		01010	ΜR	mc	r	
		m, @r		11010	ШR	mc	r	
		m,#n4	(m) ← n4	11101	МR	mc	n4	
	MOVT	DBF,@AR	$SP \leftarrow SP - 1$ , $ASR \leftarrow PC$ , $PC \leftarrow AR$ , $DBF \leftarrow (PC)$ , $PC \leftarrow ASR$ , $SP \leftarrow SP + 1$	00111	000	0001	0000	
	PUSH	AR	$SP \leftarrow SP - 1$ , $ASR \leftarrow AR$	00111	000	1101	0000	
	POP	AR	$AR \leftarrow ASR, SP \leftarrow SP + 1$	00111	000	1100	0000	
	GET	DBF,p	$DBF \leftarrow (p)$	00111	рн	1011	p∟	
	PUT	p,DBF	$(p) \leftarrow DBF$	00111	рн	1010	p∟	
	PEEK	WR,rf	$WR \leftarrow (rf)$	00111	rfr	0011	rfc	
	POKE	rf,WR	$(rf) \leftarrow WR$	00111	rf <sub>R</sub>	0010	rfc	

Instructions	Mnemonic	Operand	Operation	In	struction	n Code		
				Op code	Operand			
Branch	BR	addr	$PC_{10-0} \leftarrow addr,  PAGE \leftarrow 0$	01100		addr		
			$PC_{10-0} \leftarrow addr, PAGE \leftarrow 1$	01101				
			$PC_{10-0} \leftarrow addr, PAGE \leftarrow 2$	01110				
			$PC_{10-0} \leftarrow addr,  PAGE \leftarrow 3$	01111				
		@AR	$PC \leftarrow AR$	00111	000	0100	0000	
Subroutine	CALL	addr	$SP \leftarrow SP - 1$ , $ASR \leftarrow PC$ $PC_{11} \leftarrow 0$ , $PC_{10-0} \leftarrow addr$	11100	addr			
		@AR	$SP \leftarrow SP - 1$ , $ASR \leftarrow PC$ $PC \leftarrow AR$	00111	000	0101	0000	
	RET		$PC \leftarrow ASR, SP \leftarrow SP + 1$	00111	000	1110	0000	
	RETSK		$PC \leftarrow ASR, SP \leftarrow SP + 1 and skip$	00111	001	1110	0000	
	RETI		$PC \leftarrow ASR, INTR \leftarrow INTSK, SP \leftarrow SP + 1$	00111	010	1110	0000	
Interrupt	EI		$INTEF \leftarrow 1$	00111	000	1111	0000	
	DI		$INTEF \leftarrow 0$	00111	001	1111	0000	
Others	STOP	s	STOP	00111	010	1111	s	
	HALT	h	HALT	00111	011	1111	h	
	NOP		No operation	00111	100	1111	0000	

# 22.4 Assembler (RA17K) Embedded Macro Instruction

### Legend

NEC

flag n : FLG symbol

n : Bit number

<> : Can be omitted

	Mnemonic	Operand	Operation	n
Embedded	SKTn	flag 1, … flag n	if (flag1) ~ (flag n) = all "1", then skip	$1 \le n \le 4$
macro	SKFn	flag 1, … flag n	if (flag 1) ~ (flag n) = all "0", then skip	$1 \le n \le 4$
	SETn	flag 1, … flag n	(flag 1) ~ (flag n) ← 1	$1 \le n \le 4$
	CLRn	flag 1, … flag n	(flag 1) ~ (flag n) $\leftarrow$ 0	$1 \le n \le 4$
	NOTn	flag 1, … flag n	if (flag n) = "0", then (flag n) $\leftarrow$ 1 if (flag n) = "1", then (flag n) $\leftarrow$ 0	1 ≤ n ≤ 4
	INITFLG	<not> flag 1, ··· &lt;<not> flag n&gt;</not></not>	if description = NOT flag n, then (flag n) $\leftarrow$ 0 if description = flag n, then (flag n) $\leftarrow$ 1	1 ≤ n ≤ 4
	BANKn		$(BANK) \leftarrow n$	0 ≤ n ≤ 15
Expanded	BRX	Label	Jump Label	—
instruction	CALLX	function-name	CALL sub-routine	_
	SYSCALX	function-name or expression	CALL system sub-routine	_
	INITFLGX	<not inv=""> flag 1, ··· <not inv=""> flag n</not></not>	$\begin{array}{l} \mbox{if description} = \mbox{NOT (or INV)} \\ \mbox{flag, (flag)} \leftarrow 0 \\ \mbox{if description} = \mbox{flag, (flag)} \leftarrow 1 \end{array}$	n ≤ 4

### 23. RESERVED SYMBOLS

# 23.1 Data Buffer (DBF)

Symbol Name	Attribute	Value	R/W	Description
DBF3	MEM	0.0CH	R/W	Bits 15 through 12 of data buffer
DBF2	MEM	0.0DH	R/W	Bits 11 through 8 of data buffer
DBF1	MEM	0.0EH	R/W	Bits 7 through 4 of data buffer
DBF0	MEM	0.0FH	R/W	Bits 3 through 0 of data buffer

### 23.2 System Registers (SYSREG)

Symbol Name	Attribute	Value	R/W	Description
AR3	MEM	0.74H	R/W	Bits 15 through 12 of address register
AR2	MEM	0.75H	R/W	Bits 11 through 8 of address register
AR1	MEM	0.76H	R/W	Bits 7 through 4 of address register
AR0	MEM	0.77H	R/W	Bits 3 through 0 of address register
WR	MEM	0.78H	R/W	Window register
BANK	MEM	0.79H	R/W	Bank register
ІХН	MEM	0.7AH	R/W	Bits 10 through 8 of index register
MPH	MEM	0.7AH	R/W	Bits 6 through 4 of memory pointer
MPE	FLG	0.7AH.3	R/W	Memory pointer enable flag
IXM	MEM	0.7BH	R/W	Bits 7 through 4 of index register
MPL	MEM	0.7BH	R/W	Bits 3 through 0 of memory pointer
IXL	MEM	0.7CH	R/W	Bits 3 through 0 of index register
RPH	MEM	0.7DH	R/W	Bits 6 through 3 of general register pointer
RPL	MEM	0.7EH	R/W	Bits 2 through 0 of general register pointer
BCD	FLG	0.7EH.0	R/W	BCD operation flag
PSW	MEM	0.7FH	R/W	Program status word
CMP	FLG	0.7FH.3	R/W	Compare flag
CY	FLG	0.7FH.2	R/W	Carry flag
Z	FLG	0.7FH.1	R/W	Zero flag
IXE	FLG	0.7FH.0	R/W	Index enable flag

# 23.3 LCD Segment Register

Symbol Name	Attribute	Value	R/W	Description
LCDD19	MEM	14.5CH	R/W	LCD segment register
LCDD18	MEM	14.5DH	R/W	LCD segment register
LCDD17	MEM	14.5EH	R/W	LCD segment register
LCDD16	MEM	14.5FH	R/W	LCD segment register
LCDD15	MEM	14.60H	R/W	LCD segment register
LCDD14	MEM	14.61H	R/W	LCD segment register
LCDD13	MEM	14.62H	R/W	LCD segment register
LCDD12	MEM	14.63H	R/W	LCD segment register
LCDD11	MEM	14.64H	R/W	LCD segment register
LCDD10	MEM	14.65H	R/W	LCD segment register
LCDD9	MEM	14.66H	R/W	LCD segment register
LCDD8	MEM	14.67H	R/W	LCD segment register
LCDD7	MEM	14.68H	R/W	LCD segment register
LCDD6	MEM	14.69H	R/W	LCD segment register
LCDD5	MEM	14.6AH	R/W	LCD segment register
LCDD4	MEM	14.6BH	R/W	LCD segment register
LCDD3	MEM	14.6CH	R/W	LCD segment register
LCDD2	MEM	14.6DH	R/W	LCD segment register
LCDD1	MEM	14.6EH	R/W	LCD segment register
LCDD0	MEM	14.6FH	R/W	LCD segment register

### 23.4 Port Register

Symbol Name	Attribute	Value	R/W	Description
P0A1	FLG	0.70H.1	R/W	Port 0A bit 1
P0A0	FLG	0.70H.0	R/W	Port 0A bit 0
P0B3	FLG	0.71H.3	R/W	Port 0B bit 3
P0B2	FLG	0.71H.2	R/W	Port 0B bit 2
P0B1	FLG	0.71H.1	R/W	Port 0B bit 1
P0B0	FLG	0.71H.0	R/W	Port 0B bit 0
P0C3	FLG	0.72H.3	R/W	Port 0C bit 3
P0C2	FLG	0.72H.2	 R/W	Port 0C bit 2
P0C1	FLG	0.72H.1	 R/W	Port 0C bit 1
P0C0	FLG	0.72H.0	R/W	Port 0C bit 0
P0D3	FLG	0.73H.3	R Note	Port 0D bit 3
P0D2	FLG	0.73H.2	R Note	Port 0D bit 2
P0D1	FLG	0.73H.1	R Note	Port 0D bit 1
P0D0	FLG	0.73H.0	R Note	Port 0D bit 0
P1A3	FLG	1.70H.3	R/W	Port 1A bit 3
P1A2	FLG	1.70H.2	R/W	Port 1A bit 2
P1A1	FLG	1.70H.1	R/W	Port 1A bit 1
P1A0	FLG	1.70H.0	R/W	Port 1A bit 0
P1C3	FLG	1.72H.3	R Note	Port 1C bit 3
P1C2	FLG	1.72H.2	R Note	Port 1C bit 2
P1C1	FLG	1.72H.1	R Note	Port 1C bit 1
P1C0	FLG	1.72H.0	R Note	Port 1C bit 0
P1D3	FLG	1.73H.3	R/W	Port 1D bit 3
P1D2	FLG	1.73H.2	R/W	Port 1D bit 2
P1D1	FLG	1.73H.1	R/W	Port 1D bit 1
P1D0	FLG	1.73H.0	R/W	Port 1D bit 0
P2A2	FLG	2.70H.2	R/W	Port 2A bit 2
P2A1	FLG	2.70H.1	R/W	Port 2A bit 1
P2A0	FLG	2.70H.0	R/W	Port 2A bit 0
P2B3	FLG	2.71H.3	R/W	Port 2B bit 3
P2B2	FLG	2.71H.2	R/W	Port 2B bit 2
P2B1	FLG	2.71H.1	R/W	Port 2B bit 1
P2B0	FLG	2.71H.0	R/W	Port 2B bit 0
P2C3	FLG	2.72H.3	R/W	Port 2C bit 3
P2C2	FLG	2.72H.2		Port 2C bit 2
P2C1	FLG	2.72H.1	R/W	Port 2C bit 1
P2C0	FLG	2.72H.0	R/W	Port 2C bit 0

**Note** These ports are input-only ports. The assembler and in-circuit emulator will not output error messages even if an instruction to output from these ports is written. Also, if the instruction is actually executed on a device, the operation will have no change.

# **\*** 23.5 Register File (Control Register)

Symbol Name	Attribute	Value	R/W	Description
SP	MEM	0.81H	R/W	Stack pointer
DBFSP	MEM	0.84H	R	DBF stack pointer
SPRSEL	MEM	0.85H	R/W	Stack overflow select flag (Settable only once after power application)
MOVTSEL1	FLG	0.87H.1	R/W	MOVT bit select flag
MOVTSEL0	FLG	0.87H.0	R/W	MOVT bit select flag
SYSRSP	MEM	0.88H	R	System register stack pointer
WDTCK	MEM	15.02H	R/W	Watchdog timer clock select flag
WDTRES	FLG	15.03H.3	R/W	Watchdog timer count reset
PLLSCNF	FLG	15.10H.3	R/W	Swallow counter MSB set flag
PLLMD1	FLG	15.10H.1	R/W	PLL mode select flag
PLLMD0	FLG	15.10H.0	R/W	PLL mode select flag
PLLRFCK3	FLG	15.11H.3	R/W	PLL reference frequency select flag
PLLRFCK2	FLG	15.11H.2	R/W	PLL reference frequency select flag
PLLRFCK1	FLG	15.11H.1	R/W	PLL reference frequency select flag
PLLRFCK0	FLG	15.11H.0	R/W	PLL reference frequency select flag
PLLUL	FLG	15.12H.0	R	PLL unlock FF flag
BEEP0SEL	FLG	15.14H.2	R/W	BEEP0 enable flag
BEEP0CK1	FLG	15.14H.1	R/W	BEEP0 clock select flag
BEEP0CK0	FLG	15.14H.0	R/W	BEEP0 clock select flag
WDTCY	FLG	15.16H.0	R	Watchdog timer/stack pointer reset status detection flag
BTM0CY	FLG	15.17H.0	R	Basic timer 0 carry flag
BTM1CK0	FLG	15.18H.0	R/W	Basic timer 1 clock select flag
SIO1CK1	FLG	15.1CH.1	R/W	Serial interface 1 I/O clock select flag
SIO1CK0	FLG	15.1CH.0	R/W	Serial interface 1 I/O clock select flag
SIO1MOD	FLG	15.1DH.2	R/W	Serial interface 1 SI1/SO2 select flag
SIO1HIZ	FLG	15.1DH.1	 R/W	Serial interface 1 general-purpose port select flag
SIO1TS	FLG	15.1DH.0	R/W	Serial interface 1 transmit/receive start flag
IEG0	FLG	15.1FH.0	R/W	INT0 pin interrupt request detection edge direction select flag
IFCG0STT	FLG	15.21H.0	R	IF counter gate status detection flag (1: open, 0: close)
IFCMD1	FLG	15.22H.3	R/W	IF counter mode select flag (10: FMIFC, 11: AMIFC2)
IFCMD0	FLG	15.22H.2	R/W	IF counter mode select flag (00: FCG, 01: AMIFC)
IFCCK1	FLG	15.22H.1	R/W	IF counter clock select flag
IFCCK0	FLG	15.22H.0	R/W	IF counter clock select flag
IFCSTRT	FLG	15.23H.1	W	IF counter count start
IFCRES	FLG	15.23H.0	R/W	IF counter reset

Symbol Name	Attribute	Value	R/W	Description
ADCCH3	FLG	15.24H.3	R/W	A/D converter channel select flag
ADCCH2	FLG	15.24H.2	R/W	A/D converter channel select flag
ADCCH1	FLG	15.24H.1	R/W	A/D converter channel select flag
ADCCH0	FLG	15.24H.0	R/W	A/D converter channel select flag
ADCSTRT	FLG	15.25H.1	R/W	A/D converter compare start flag
ADCCMP	FLG	15.25H.0	R	A/D converter compare result detection flag
TMOEN	FLG	15.2BH.3	R/W	Modulo timer 0 count start flag
TMORES	FLG	15.2BH.2	R/W	Modulo timer 0 reset flag (The value is 0 when read)
ТМ0СК1	FLG	15.2BH.1	R/W	Modulo timer 0 clock select flag (10: TM10, 11: TM11)
ТМОСКО	FLG	15.2BH.0	R/W	Modulo timer 0 clock select flag (00: 75 kHz, 01: 25 kHz)
TM00VF	FLG	15.2CH.3	R	Modulo timer 0 overflow detection flag
IPSIO1	FLG	15.2FH.3	R/W	Serial interface 1 interrupt enable flag
IPBTM1	FLG	15.2FH.2	R/W	Basic timer 1 interrupt enable flag
IPTM0	FLG	15.2FH.1	R/W	Modulo timer 0 interrupt enable flag
IP0	FLG	15.2FH.0	R/W	INT0 pin interrupt enable flag
IRQSIO1	FLG	15.3CH.0	R/W	Serial interface 1 interrupt request detection flag
IRQBTM1	FLG	15.3DH.0	R/W	Basic timer 1 interrupt request detection flag
IRQTM0	FLG	15.3EH.0	R/W	Modulo timer 0 interrupt request detection flag
INT0	FLG	15.3FH.3	R/W	INT0 pin status detection flag
IRQ0	FLG	15.3FH.0	R/W	INT0 pin interrupt request detection flag
LCDEN	FLG	15.40H.0	R/W	LCD driver display start flag
LCD19SEL	FLG	15.69H.2	R/W	P2A2/LCD19 switching flag
LCD18SEL	FLG	15.69H.1	R/W	P2A1/LCD18 switching flag
LCD17SEL	FLG	15.69H.0	R/W	P2A0/LCD17 switching flag
P0DPLD3	FLG	15.6AH.3	R/W	POD3 pin pull-down resistor switching flag
P0DPLD2	FLG	15.6AH.2	R/W	POD2 pin pull-down resistor switching flag
P0DPLD1	FLG	15.6AH.1	R/W	POD1 pin pull-down resistor switching flag
P0DPLD0	FLG	15.6AH.0	R/W	POD0 pin pull-down resistor switching flag
P2CBIO3	FLG	15.6BH.3	R/W	P2C3 I/O select flag
P2CBIO2	FLG	15.6BH.2	R/W	P2C2 I/O select flag
P2CBIO1	FLG	15.6BH.1	R/W	P2C1 I/O select flag
P2CBIO0	FLG	15.6BH.0	R/W	P2C0 I/O select flag
P2BBIO3	FLG	15.6CH.3	R/W	P2B3 I/O select flag
P2BBIO2	FLG	15.6CH.2	R/W	P2B2 I/O select flag
P2BBIO1	FLG	15.6CH.1	R/W	P2B1 I/O select flag
P2BBIO0	FLG	15.6CH.0	R/W	P2B0 I/O select flag

Symbol Name	Attribute	Value	R/W	Description
P1DBIO3	FLG	15.6DH.3	R/W	P1D3 I/O select flag
P1DBIO2	FLG	15.6DH.2	R/W	P1D2 I/O select flag
P1DBIO1	FLG	15.6DH.1	R/W	P1D1 I/O select flag
P1DBIO0	FLG	15.6DH.0	R/W	P1D0 I/O select flag
P1ABIO3	FLG	15.6EH.3	R/W	P1A3 I/O select flag
P1ABIO2	FLG	15.6EH.2	R/W	P1A2 I/O select flag
P1ABIO1	FLG	15.6EH.1	R/W	P1A1 I/O select flag
P1ABIO0	FLG	15.6EH.0	R/W	P1A0 I/O select flag
P0BBIO3	FLG	15.6FH.3	R/W	P0B3 I/O select flag
P0BBIO2	FLG	15.6FH.2	R/W	P0B2 I/O select flag
P0BBIO1	FLG	15.6FH.1	R/W	P0B1 I/O select flag
P0BBIO0	FLG	15.6FH.0	R/W	P0B0 I/O select flag

# 23.6 Peripheral Hardware Register

Symbol Name	Attribute	Value	R/W	Description
ADCR	DAT	02H	R/W	A/D converter reference voltage set register
SIO1SFR	DAT	04H	R/W	Serial interface 1 presettable shift register
тмом	DAT	1AH	R/W	Timer modulo 0 register
TM0C	DAT	1BH	R	Timer modulo 0 counter
AR	DAT	40H	R/W	Address register
DBFSTK	DAT	41H	R/W	DBF stack register
PLLR	DAT	42H	R/W	PLL data register
IFC	DAT	43H	R	IF counter data register

### 23.7 Others

Symbol Name	Attribute	Value	Description
DBF	DAT	0FH	Operand (DBF) for GET/PUT/MOVT/MOVTH/MOVTL instruction
IX	DAT	01H	Operand (IX) for INC instruction
AR_EPA1	DAT	8040H	Operand (EPA bit ON) for CALL/BR/MOVT/MOVTH/MOVTL instruction
AR_EPA0	DAT	4040H	Operand (EPA bit OFF) for CALL/BR/MOVT/MOVTH/MOVTL instruction

#### \*

### 24. ELECTRICAL CHARACTERISTICS

#### Absolute Maximum Ratings (T<sub>A</sub> = 25 $^{\circ}$ C)

Parameter	Symbol	Condition	Rating	Unit
Supply voltage	Vddo		-0.3 to +2.0	V
	Vdd1		-0.3 to +2.0	V
	Vdd2		-0.3 to +2.0	V
Input voltage	VI1	P0D0-P0D3, P1C0-P1C3 pins	-0.3 to VDD2+0.3	V
	V12	VCOH and VCOL pins	-0.3 to VDD1+0.3	V
	Vı3	P0B0-P0B3, P1A0-P1A3, P1D0-P1D3, P2A0-P2A2, P2B0-P2B3, P2C0-P2C3, RESET, and INT pins	–0.3 to Vdd0+0.3	V
Output voltage	Vo1 P0B0-P0B3, P0C0-P0C3, P2B0-P2B3, P2C0-P2C3		-0.3 to VDD0+0.3	V
	V <sub>O2</sub>	EO1 and EO2 pins	-0.3 to REGLCD1+0.3	V
	V <sub>O3</sub>	LCD0-LCD19, COM0-COM3 pins	-0.3 to REGLCD2+0.3	V
High-level output current	Іон	1 pin	-3.0	mA
		Total of P0B0-P0B3, P0C0-P0C3, P2B0-P2B3, and P2C0-P2C3	-30.0	mA
Low-level output current	IOL1	1 pin of P0A0, P0A1, and P1D0-P1D3	10.0	mA
	IOL2	1 pin other P0A0, P0A1, and P1D0-P1D3	3.0	mA
		Total of P0A0, P0A1, P0B0-P0B3, P0C0-P0C3, P1A0- P1A3, P1D0-P1D3, P2B0-P2B3, and P2C0-P2C3	45.0	mA
Output voltage	VBDS	P0A0, P0A1, P1A0-P1A3, P1D0-P1D3	-0.3 to +4.0	V
Operating ambient temperature	TA		-10 to +50	°C
Storage temperature	Tstg		-55 to +125	°C

Caution If the absolute maximum rating of even one of the above parameters is exceeded even momentarily, the quality of the product may be degraded. The absolute maximum ratings, therefore, specify the value exceeding which the product may be physically damaged. Be sure to use the product with these ratings never being exceeded.

# Recommended Supply Voltage Range (T<sub>A</sub> = -10 to +50 °C)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Supply voltage	Vddo		1.05		1.8	V
	Vdd1		1.05		1.8	V
	Vdd2		1.05		1.8	V

# Recommended Output Voltage ( $T_A = -10$ to +50 °C)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Output voltage	VBDS	P0A0, P0A1, P1A0-P1A3, P1D0-P1D3			+4.0	V

\*

DC Characteristics (TA = -10 to +50 °C, VDD = VDD0 = VDD1 = VDD2 = 1.05 to 1.8 V)

Parameter	Symbol	Condition	n	MIN.	TYP.	MAX.	Unit
Supply current	Idd11	CPU operation (VDD0 = 1.8 V, T	∝ = 25 °C)		120	320	μΑ
	DD12	HALT operation (VDD0 = 1.8 V, $T_A = 25 \ ^{\circ}C$ )			70	240	μΑ
	Idd2	PLL operation (VCOH pin, fı⊨ 130 MHz, Vı⊨	PLL operation (VCOH pin, fin = 130 MHz, Vin = 0.5 VP-P, VDD = 1.8 V)		7.0	10.0	mA
	Idd3	A/D converter, IF counter opera (f_{IN} = 10.7 MHz, V_{IN} = 0.3 V_{P-P},	ation V <sub>DD2</sub> = 1.8 V)		3.0	7.0	mA
High-level input voltage	VIH1	P0B0-P0B3, P1A0-P1A3, P1C0 P2A0-P2A2, P2B0-P2B3, P2C0	D-P1C3, P1D0-P1D3, D-P2C3, RESET, INT	0.8 Vdd		Vdd	V
	VIH2	P0D0-P0D3		0.8 Vdd		Vdd	V
Low-level input voltage	VIL1	P0B0-P0B3, P1A0-P1A3, P1C0 P2A0-P2A2, P2B0-P2B3, P2C0	D-P1C3, P1D0-P1D3, D-P2C3, RESET, INT	0		0.1 Vdd	V
	VIL2	P0D0-P0D3		0		0.1 Vdd	V
High-level output voltage	Vон1	P0B0-P0B3, P0C0-P0C3, P2B0	D-P2B3, P2C0-P2C3			VDD0	V
	Vон2	EO0, EO1				REGLCD1	V
	Vонз	LCD0-LCD19, COM0-COM3				REGLCD2	V
High-level input current	Іінт	With P0D0-P0D3 pulled down	VIH = VDD0 = 1.1 V	2		30	μΑ
			Vih = Vddo	2			μΑ
Low-level input current	lil1	With X <sub>IN</sub> pin pulled up	VIH = VDD0 = 1.1 V	-2		-30	μΑ
			Vih = Vddo	-2			μΑ
High-level output current	Іон1	Р0В0-Р0В3, Р0С0-Р0С3, Р2В0-Р2В3, Р2С0-Р2С3 (Voн = Vdd1 – 0.2 V)		-0.13			mA
	Іон2	EO0, EO1 (Vон = Vdd2 - 0.2 V)		-0.13			mA
	Іонз	LCD0-LCD19 (Vон = LCDreg2 -	– 0.2 V)	-1			μΑ
	Іон4	СОМ0-СОМ3 (Vон = LCD <sub>REG</sub> 2 - 0.2 V)		-10			μΑ
Low-level output current	IOL1	P0B0-P0B3, P0C0-P0C3, P2B0 (Vol = 0.2 V)	D-P2B3, P2C0-P2C3	0.2			mA
	IOL2	EO0, EO1 (Vol = 0.2 V)		0.2			mA
	Іоіз	P1A0-P1A3 (VoL = 0.2 V)		0.4			mA
	IOL4	P0A0, P1A1, P1D0-P1D3 (Vo∟	= 0.2 V)	6.0			mA
	IOL5	LCD0-LCD19 (VoL = 0.2 V)		1			μΑ
	IOL6	COM0-COM3 (VoL = 0.2 V)		10			μΑ
LCD drive voltage	VLCD0	LCD0-LCD19 output open, C1-	C5 = 0.1 μF	1.4	1.5	1.6	V
	VLCD1 TA = 25 °C		2.8	3.0	3.2	V	
Output off leakage current	IL1	Р0А0, Р0А1, Р1А0-Р1А3, Р1D0-Р1D3 (Vон = 1.8 V)				1	μΑ
	IL2 EO0, EO1 (Vон = 1.8 V, VоL = 0 V)				±1	μΑ	
COM intermediate potential output voltage	Vм	COM0-COM3 (output open)		VLCD0 -0.2	VLCD0	V <sub>LCD0</sub> +0.2	V
COM intermediate potential output current	Іом	СОМО-СОМЗ (Vом = Vм ± 0.2 V	V)	±1			μA

Remark Unless otherwise specified, the characteristics of muxed pins are the same as those of the port pins.

### AC Characteristics (TA = -10 to +50 $^{\circ}$ C, VDD = VDD0 = VDD1 = VDD2 = 1.05 to 1.8 V)

Parameter	Symbol	Condition MIN. TYP.		MAX.	Unit	
Operating frequency	fin1	VHF mode, sine wave input $V_{IN} = 0.3 V_{P-P}$	35		220	MHz
	fin2	HF mode, sine wave input $V_{IN} = 0.3 V_{P-P}$	5		40	MHz
	fіnз	MF mode, sine wave input V_{IN} = 0.3 V_{P-P}	0.8		3.5	MHz
	fin4	AMIFC pin, sine wave input V_IN = 0.3 V_{P-P}	0.4		2	MHz
	fin5	AMIFC pin, sine wave input V_IN = 0.3 V_{P-P}	0.4		2	MHz
	fin6	FMIFC pin, sine wave input V_IN = 0.3 V_{P-P}	10		11	MHz
SCK input frequency	fin7	External clock			75	kHz

#### A/D Converter Characteristics (TA = -10 to +50 °C, VDD = VDD0 = VDD1 = VDD2 = 1.05 to 1.8 V)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Total conversion error		V <sub>DD2</sub> = 1.05 to 1.8 V			±3	LSB

### 25. PACKAGE DRAWING

# 80 PIN PLASTIC TQFP (FINE PITCH) (□12)



#### NOTE

Each lead centerline is located within 0.10 mm (0.004 inch) of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS	INCHES
Α	14.0±0.2	0.551±0.008
В	12.0±0.2	$0.472^{+0.009}_{-0.008}$
С	12.0±0.2	$0.472\substack{+0.009\\-0.008}$
D	14.0±0.2	0.551±0.008
F	1.25	0.049
G	1.25	0.049
н	0.22±0.05	$0.009^{+0.002}_{-0.003}$
I	0.10	0.004
J	0.5 (T.P.)	0.020 (T.P.)
к	1.0±0.2	$0.039^{+0.009}_{-0.008}$
L	0.5±0.2	$0.020^{+0.008}_{-0.009}$
М	0.145±0.05	$0.006^{+0.002}_{-0.003}$
Ν	0.10	0.004
Ρ	1.0±0.05	$0.040^{+0.002}_{-0.003}$
Q	0.1±0.05	0.004±0.002
R	3° <sup>+7°</sup> -3°	3° <sup>+7°</sup> -3°
S	1.2 MAX.	0.048 MAX.
		S80GK-50-9EU

### ★ 26. RECOMMENDED SOLDERING CONDITIONS

Solder the  $\mu$ PD17933 and  $\mu$ PD17934 under the following recommended conditions.

For details of the recommended soldering conditions, refer to information document **Semiconductor Device Mounting Technology Manual (C10535E)**.

For soldering methods and conditions other than those recommended, consult NEC.

#### Table 26-1. Soldering conditions for surface-mount type

### $\mu$ PD17933GK-×××-BE9: 80-pin plastic TQFP (12 mm × 12 mm, 0.5-mm pitch) $\mu$ PD17934GK-×××-BE9: 80-pin plastic TQFP (12 mm × 12 mm, 0.5-mm pitch)

Soldering Method	Soldering Conditions	Recommended Conditions Symbol
Infrared reflow	Package peak temperature: 235 °C; Time: 30 secs. max. (210 °C min.); Number of times: twice max.; Number of days: 7 <sup>Note</sup> (after that, prebaking is necessary at 125 °C for 10 hours) <precaution> Products other than in heat-resistance trays (such as those packaged in a magazine, taping, or non-heat-resistance tray) cannot be baked while they are in their package.</precaution>	IR35-107-2
VPS	Package peak temperature: 215 °C; Time: 40 secs. max. (200 °C min.); Number of times: twice max.; Number of days: 7 <sup>Note</sup> (after that, prebaking is necessary at 125 °C for 10 hours) <precaution> Products other than in heat-resistance trays (such as those packaged in a magazine, taping, or non-heat-resistance tray) cannot be baked while they are in their package.</precaution>	VP15-107-2
Partial heating	Pin temperature: 300 °C max.; Time: 3 secs. max. (per device side)	_

**Note** Number of days in storage after the dry pack has been opened. The storage conditions are at 25 °C, 65 % RH max.

Caution Do not use two or more soldering methods in combination (except partial heating).

### APPENDIX A. CAUTIONS ON CONNECTING CRYSTAL RESONATOR

When using the system clock oscillation circuit, wire the portion enclosed by the dotted line in the figure below as follows to prevent adverse influence from wiring capacity.

- Keep the wiring length as short as possible.
- If capacitances C1 and C2 are too high, the oscillation start characteristics may be degraded or current consumption may increase.
- Generally, connect a trimmer capacitor for adjusting the oscillation frequency to the XIN pin. Depending on the crystal resonator to be used, however, the oscillation stability differs. Therefore, evaluate the crystal resonator actually used.
- The crystal oscillation frequency cannot be accurately adjusted when an emulation probe is connected to the Xout and XiN pin, because of the capacitance of the probe. Adjust the frequency while measuring the VCO oscillation frequency.



### APPENDIX B. DEVELOPMENT TOOLS

The following development tools are available for development of programs for the  $\mu$ PD17933 and 17934.

#### Hardware

Name	Outline
In-circuit emulator IE-17K IE-17K-ET <sup>Note 1</sup> EMU-17K <sup>Note 2</sup>	IE-17K, IE-17K-ET, and EMU-17K are in-circuit emulators that can be used with any model in the 17K series. IE-17K and IE-17K-ET are connected to a host machine, which is PC-9800 series or IBM PC/AT <sup>TM</sup> , with RS-232C. EMU-17K is mounted to the expansion slot of a host machine, PC-9800 series. By using these in-circuit emulators with a system evaluation board (SE board) corresponding to each model, these emulators operate dedicated to the model. When man-machine interface software <i>SIMPLEHOST</i> <sup>TM</sup> is used, a more sophisticated debugging environment can be created. EMU-17K also has a function to allow you to check the contents of the data memory real-time.
SE board (SE-17934)	SE-17934 is an SE board for the $\mu$ PD17933 and 17934. This board can be used alone to evaluate a system, or in combination with an in-circuit emulator for debugging.
Emulation probe (EP-17K80GK)	EP-17K80GK is an emulation probe for the $\mu$ PD17933 and 17934. By using this probe with TGK-080SDPNote <sup>3</sup> , the SE board and target system are connected.
Conversion socket (TGK-080SDP <sup>Note 3</sup> )	TGK-080SDP is a conversion socket for 80-pin plastic TQFP ( $12 \times 12$ mm). It is used to connect EP-17K80GK and target system.

- Notes 1. Low-price model: external power supply type
  - 2. This is a product of I.C Corp. For details, consult I.C Corp. (03-3447-3793).
  - **3.** A product of TOKYO Eletech Corp. (03-5295-1661). Consult your NEC distributor when purchasing the product.
- **Remark** Third party PROM programmers AF-9703, AF-9704, AF-9705, and AF-9706 are available from Ando Electric Co., Ltd. Use these programmers with programmer adapter PA-17P709GC. For details, consult Ando Electric Co., Ltd. (03-3733-1163).

#### $\star$

#### Software

Name	Outline	Host Machine	08	6	Parts Number	
17K series assembler (RA17K)	RA17K is a relocatable assembler that can be commonly used with 17K series. To develop programs for the $\mu$ PD17933 and 17934, this RA17K and a device file (AS17934) are used	PC-9800 series	MS-DOS™		μS5A13RA17K	
	in combination. The RA17K is provided with linker (LK17K) and document processor (DOC17K) utility.	IBM PC/AT	PC DOS	ГМ	μS7B13RA17K	
17K series	<i>emIC-17K</i> is a C-like compiler that can be used	PC-9800 series	MS-DOS		μS5A13CC17K	
( <i>emlC-17K</i> <sup>™</sup> )	It is used together with RA17K.	IBM PC/AT	PC DOS		μS7B13CC17K	
Device file (AS17934)	AS17934 is a device file for the $\mu$ PD17933 and 17934.	PC-9800 series	MS-DOS		μS5A13AS17707	
	It is used together with the assembler common to the 17K series (RA17K).	IBM PC/AT	PC DOS		μS7B13AS17707	
Support software	SIMPLEHOST is man-machine interface software that runs on Windows <sup>™</sup> when a program is	PC-9800 series	MS-DOS	Windows	μ\$5A13IE17K	
(SIMPLEHOST) developed by personal comp	developed by using an in-circuit emulator and personal computer.	IBM PC/AT	PC DOS		μS7B13IE17K	

### **Remark** The version of the supported OS is as follows:

OS	Version
MS-DOS	Ver.3.30 to Ver.5.00A <sup>Note</sup>
PC DOS	Ver.3.1 to Ver.5.0 <sup>Note</sup>
Windows	Ver.3.0 to Ver.3.1

**Note** MS-DOS Ver. 5.00/5.00A and PC DOS Ver. 5.0 have a task swap function, but this function cannot be used with this software.

[MEMO]

[MEMO]

# NOTES FOR CMOS DEVICES -

## **1** PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note: Strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

### **(2)** HANDLING OF UNUSED INPUT PINS FOR CMOS

Note: No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS device behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to VDD or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

### **③** STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note: Power-on does not necessarily define initial status of MOS device. Production process of MOS does not define the initial operation status of the device. Immediately after the power source is turned ON, the devices with reset function have not yet been initialized. Hence, power-on does not guarantee out-pin levels, I/O settings or contents of registers. Device is not initialized until the reset signal is received. Reset operation must be executed immediately after power-on for devices having reset function.

NEC
# **Regional Information**

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- · Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

### **NEC Electronics Inc. (U.S.)**

Santa Clara, California Tel: 408-588-6000 800-366-9782 Fax: 408-588-6130 800-729-9288

#### **NEC Electronics (Germany) GmbH**

Duesseldorf, Germany Tel: 0211-65 03 02 Fax: 0211-65 03 490

## NEC Electronics (UK) Ltd.

Milton Keynes, UK Tel: 01908-691-133 Fax: 01908-670-290

#### NEC Electronics Italiana s.r.1.

Milano, Italy Tel: 02-66 75 41 Fax: 02-66 75 42 99

### NEC Electronics (Germany) GmbH Benelux Office Eindhoven, The Netherlands Tel: 040-2445845 Fax: 040-2444580

NEC Electronics (France) S.A. Velizy-Villacoublay, France Tel: 01-30-67 58 00 Fax: 01-30-67 58 99

NEC Electronics (France) S.A. Spain Office Madrid, Spain Tel: 01-504-2787 Fax: 01-504-2860

#### NEC Electronics (Germany) GmbH Scandinavia Office

Taeby, Sweden Tel: 08-63 80 820 Fax: 08-63 80 388 NEC Electronics Hong Kong Ltd. Hong Kong Tel: 2886-9318 Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd. Seoul Branch Seoul, Korea Tel: 02-528-0303 Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd. United Square, Singapore 1130 Tel: 65-253-8311 Fax: 65-250-3583

NEC Electronics Taiwan Ltd. Taipei, Taiwan Tel: 02-719-2377 Fax: 02-719-5951

#### NEC do Brasil S.A.

Cumbica-Guarulhos-SP, Brasil Tel: 011-6465-6810 Fax: 011-6465-6829

J98. 2

#### emIC-17K and SIMPLEHOST are trademarks of NEC Corporation.

MS-DOS and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

PC/AT and PC DOS are trademarks of IBM Corporation.

The export of this product from Japan is regulated by the Japanese government. To export this product may be prohibited without governmental license, the need for which must be judged by the customer. The export or re-export of this product from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.

NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or others.

While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.

NEC devices are classified into the following three quality grades:

"Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device before using it in a particular application.

- Standard: Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots
- Special: Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)
- Specific: Aircrafts, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems or medical equipment for life support, etc.

The quality grade of NEC devices is "Standard" unless otherwise specified in NEC's Data Sheets or Data Books. If customers intend to use NEC devices for applications other than those specified for Standard quality grade, they should contact an NEC sales representative in advance.

Anti-radioactive design is not implemented in this product.