Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Not For New Designs
Core Processor	H8/300H
Core Size	16-Bit
Speed	16MHz
Connectivity	I²C, SCI
Peripherals	PWM, WDT
Number of I/O	29
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LFQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/hd64f3664fpv

Preface

The H8/3664 Group are single-chip microcomputers made up of the high-speed H8/300H CPU employing Renesas Technology original architecture as their cores, and the peripheral functions required to configure a system. The H8/300H CPU has an instruction set that is compatible with the H8/300 CPU.

Target Users: This manual was written for users who will be using the H8/3664 Group in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8/3664 Group to the target users.
Refer to the H8/300H Series Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions and electrical characteristics.
- In order to understand the details of the CPU's functions
  Read the H8/300H Series Software Manual.
- In order to understand the details of a register when its name is known
  Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bits, and initial values of the registers are summarized in section 19, List of Registers.
  Example: Bit order: The MSB is on the left and the LSB is on the right.

Notes:

When using the on-chip emulator (E7, E8) for H8/3664 program development and debugging, the following restrictions must be noted.

1. The $\overline{\text{NMI}}$ pin is reserved for the E7 or E8, and cannot be used.
2. Pins P85, P86, and P87 cannot be used. In order to use these pins, additional hardware must be provided on the user board.
3. Area H'7000 to H'7FFF is used by the E7 or E8, and is not available to the user.
4. Area H'F780 to H'FB7F must on no account be accessed.

6.1.3 Module Standby Control Register 1 (MSTCR1)

MSTCR1 allows the on-chip peripheral modules to enter a standby state in module units.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0.
6	MSTIIC	0	R/W	IIC Module Standby IIC enters standby mode when this bit is set to 1
5	MSTS3	0	R/W	SCI3 Module Standby SCI3 enters standby mode when this bit is set to 1
4	MSTAD	0	R/W	A/D Converter Module Standby A/D converter enters standby mode when this bit is set to 1
3	MSTWD	0	R/W	Watchdog Timer Module Standby Watchdog timer enters standby mode when this bit is set to 1.When the internal oscillator is selected for the watchdog timer clock, the watchdog timer operates regardless of the setting of this bit
2	MSTTW	0	R/W	Timer W Module Standby Timer W enters standby mode when this bit is set to 1
1	MSTTV	0	R/W	Timer V Module Standby Timer V enters standby mode when this bit is set to 1
0	MSTTA	0	R/W	Timer A Module Standby Timer A enters standby mode when this bit is set to 1

7.1 Block Configuration

Figure 7.1 shows the block configuration of 32-kbyte flash memory. The thick lines indicate erasing units, the narrow lines indicate programming units, and the values are addresses. The flash memory is divided into 1 kbyte × 4 blocks and 28 kbytes × 1 block. Erasing is performed in these units. Programming is performed in 128-byte units starting from an address with lower eight bits H'00 or H'80.

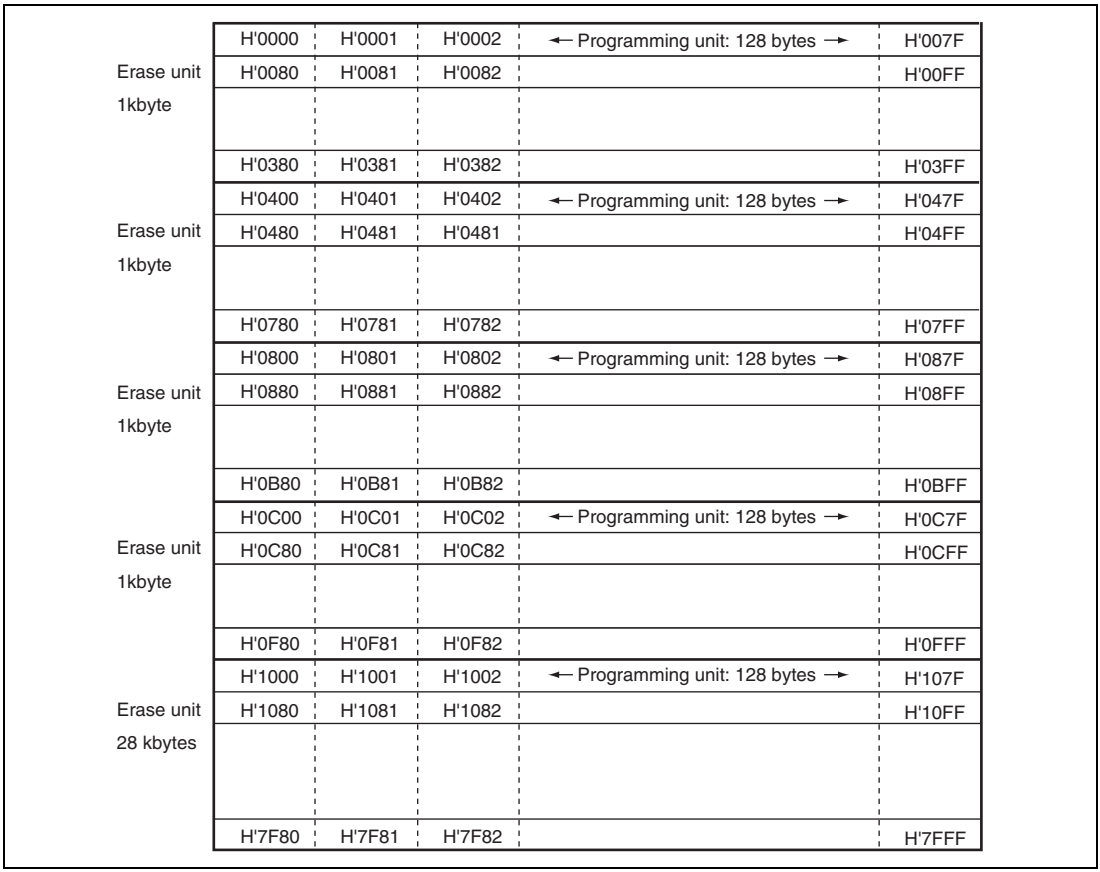

Figure 7.1 Flash Memory Block Configuration

9.3.5 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P57/SCL Pin

Register	ICCR	PCR5	
Bit Name	**ICE**	**PCR57**	**Pin Function**
Setting Value	0	0	P57 input pin
		1	P57 output pin
	1	X	SCL I/O pin

[Legend]
X: Don't care.

SCL performs the NMOS open-drain output, that enables a direct bus drive.

- P56/SDA Pin

Register	ICCR	PCR5	
Bit Name	**ICE**	**PCR56**	**Pin Function**
Setting Value	0	0	P56 input pin
		1	P56 output pin
	1	X	SDA I/O pin

[Legend]
X: Don't care.

SDA performs the NMOS open-drain output, that enables a direct bus drive.

- P55/$\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ Pin

Register	PMR5	PCR5	
Bit Name	**WKP5**	**PCR55**	**Pin Function**
Setting Value	0	0	P55 input pin
		1	P55 output pin
	1	X	$\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ input pin

[Legend]
X: Don't care.

9.4.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P76/TMOV Pin

Register	TCSRV	PCR7	
Bit Name	**OS3 to OS0**	**PCR76**	**Pin Function**
Setting Value	0000	0	P76 input pin
		1	P76 output pin
	Other than the above values	X	TMOV output pin

[Legend]
X: Don't care.

- P75/TMCIV Pin

Register	PCR7	
Bit Name	**PCR75**	**Pin Function**
Setting Value	0	P75 input/TMCIV input pin
	1	P75 output/TMCIV input pin

- P74/TMRIV Pin

Register	PCR7	
Bit Name	**PCR74**	**Pin Function**
Setting Value	0	P74 input/TMRIV input pin
	1	P74 output/TMRIV input pin

10.4 Operation

10.4.1 Interval Timer Operation

When bit TMA3 in TMA is cleared to 0, timer A functions as an 8-bit interval timer.

Upon reset, TCA is cleared to H'00 and bit TMA3 is cleared to 0, so up-counting of timer A resume immediately as an interval timer. The clock input to timer A is selected by bits TMA2 to TMA0 in TMA; any of eight internal clock signals output by prescaler S can be selected.

After the count value in TCA reaches H'FF, the next clock signal input causes timer A to overflow, setting bit IRRTA to 1 in interrupt Flag Register 1 (IRR1). If IENTA = 1 in interrupt enable register 1 (IENR1), a CPU interrupt is requested. At overflow, TCA returns to H'00 and starts counting up again. In this mode timer A functions as an interval timer that generates an overflow output at intervals of 256 input clock pulses.

10.4.2 Clock Time Base Operation

When bit TMA3 in TMA is set to 1, timer A functions as a clock-timer base by counting clock signals output by prescaler W. When a clock signal is input after the TCA counter value has become H'FF, timer A overflows and IRRTA in IRR1 is set to 1. At that time, an interrupt request is generated to the CPU if IENTA in the interrupt enable register 1 (IENR1) is 1. The overflow period of timer A is set by bits TMA1 and TMA0 in TMA. A choice of four periods is available. In clock time base operation (TMA3 = 1), setting bit TMA2 to 1 clears both TCA and prescaler W to H'00.

10.4.3 Clock Output

Setting bit TMOW in port mode register 1 (PMR1) to 1 causes a clock signal to be output at pin TMOW. Eight different clock output signals can be selected by means of bits TMA7 to TMA5 in TMA. The system clock divided by 32, 16, 8, or 4 can be output in active mode and sleep mode. A 32.768 kHz signal divided by 32, 16, 8, or 4 can be output in active mode, sleep mode, and subactive mode.

12.3.6 Timer I/O Control Register 1 (TIOR1)

TIOR1 selects the functions of GRC and GRD, and specifies the functions of the FTIOC and FTIOD pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1.
6	IOD2	0	R/W	I/O Control D2 Selects the GRD function. 0: GRD functions as an output compare register 1: GRD functions as an input capture register
5	IOD1	0	R/W	I/O Control D1 and D0
4	IOD0	0	R/W	When IOD2 = 0, 00: No output at compare match 01: 0 output to the FTIOD pin at GRD compare match 10: 1 output to the FTIOD pin at GRD compare match 11: Output toggles to the FTIOD pin at GRD compare match When IOD2 = 1, 00: Input capture at rising edge at the FTIOD pin 01: Input capture at falling edge at the FTIOD pin 1X: Input capture at rising and falling edges at the FTIOD pin
3	—	1	—	Reserved This bit is always read as 1.
2	IOC2	0	R/W	I/O Control C2 Selects the GRC function. 0: GRC functions as an output compare register 1: GRC functions as an input capture register

14.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI3 and multiprocessor bits for transfer. 1 cannot be written to flags TDRE, RDRF, OER, PER, and FER; they can only be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/W	Transmit Data Register Empty Displays whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR3 is 0 • When data is transferred from TDR to TSR [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the transmit data is written to TDR
6	RDRF	0	R/W	Receive Data Register Full Indicates that the received data is stored in RDR. [Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When data is read from RDR
5	OER	0	R/W	Overrun Error [Setting condition] When an overrun error occurs in reception [Clearing condition] When 0 is written to OER after reading OER = 1
4	FER	0	R/W	Framing Error [Setting condition] When a framing error occurs in reception [Clearing condition] When 0 is written to FER after reading FER = 1

Table 14.2 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (3)

	Operating Frequency ϕ (MHz)								
	14			14.7456			16		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	64	0.70	3	70	0.03
150	2	181	0.16	2	191	0.00	2	207	0.16
300	2	90	0.16	2	95	0.00	2	103	0.16
600	1	181	0.16	1	191	0.00	1	207	0.16
1200	1	90	0.16	1	95	0.00	1	103	0.16
2400	0	181	0.16	0	191	0.00	0	207	0.16
4800	0	90	0.16	0	95	0.00	0	103	0.16
9600	0	45	–0.93	0	47	0.00	0	51	0.16
19200	0	22	–0.93	0	23	0.00	0	25	0.16
31250	0	13	0.00	0	14	–1.70	0	15	0.00
38400	—	—	—	0	11	0.00	0	12	0.16

[Legend]

—: A setting is available but error occurs.

Table 14.3 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N	ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N
2	62500	0	0	7.3728	230400	0	0
2.097152	65536	0	0	8	250000	0	0
2.4576	76800	0	0	9.8304	307200	0	0
3	93750	0	0	10	312500	0	0
3.6864	115200	0	0	12	375000	0	0
4	125000	0	0	12.288	384000	0	0
4.9152	153600	0	0	14	437500	0	0
5	156250	0	0	14.7456	460800	0	0
6	187500	0	0	16	500000	0	0
6.144	192000	0	0				

Assuming values of F (absolute value of clock rate deviation) = 0 and D (clock duty) = 0.5 in formula (1), the reception margin can be given by the formula.

$M = \{0.5 - 1/(2 \times 16)\} \times 100\ [\%] = 46.875\%$

However, this is only the computed value, and a margin of 20% to 30% should be allowed for in system design.

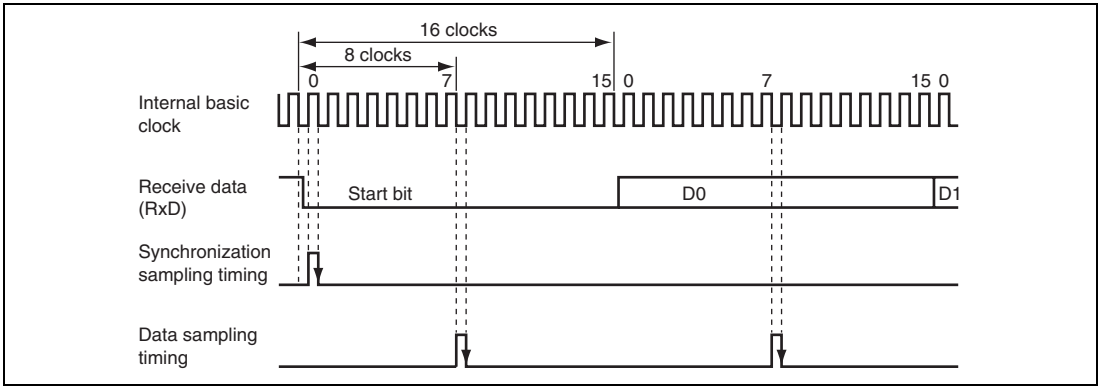

Figure 14.19 Receive Data Sampling Timing in Asynchronous Mode

15.4.5 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR to 1. Set the MLS bit in ICMR and the MST and TRS bits in ICCR according to the operating mode.
2. When the slave address matches in the first frame following detection of the start condition, the slave device drives SDA low at the 9th clock pulse and returns an acknowledge signal. At the same time, the IRIC flag in ICCR is set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. If the 8th data bit (R/$\overline{W}$) is 1, the TRS bit in ICCR is set to 1, and the mode changes to slave transmit mode automatically. The TDRE internal flag is set to 1. The slave device drives SCL low from the fall of the transmit clock until ICDR data is written.
3. After clearing the IRIC flag to 0, write data to ICDR. The TDRE internal flag is cleared to 0. The written data is transferred to ICDRS, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again. After clearing the IRIC flag to 0, write the next data to ICDR. The slave device sequentially sends the data written into ICDR in accordance with the clock output by the master device at the timing shown in figure 15.9.
4. When one frame of data has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th transmit clock pulse. If the TDRE internal flag has been set to 1, this slave device drives SCL low from the fall of the transmit clock until data is written to ICDR. The master device drives SDA low at the 9th clock pulse, and returns an acknowledge signal. As this acknowledge signal is stored in the ACKB bit in ICSR, this bit can be used to determine whether the transfer operation was performed normally. When the TDRE internal flag is 0, the data written into ICDR is transferred to ICDRS, transmission is started, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again.
5. To continue transmission, clear the IRIC flag to 0, then write the next data to be transmitted into ICDR. The TDRE flag is cleared to 0.

Transmit operations can be performed continuously by repeating steps [4] and [5]. To end transmission, write H'FF to ICDR. When SDA is changed from low to high when SCL is high, and the stop condition is detected, the BBSY flag in ICCR is cleared to 0.

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
DEC	DEC.L #1, ERd	L		2								ERd32–1 → ERd32	—	—	↕	↕	↕	—	2	
	DEC.L #2, ERd	L		2								ERd32–2 → ERd32	—	—	↕	↕	↕	—	2	
DAS	DAS.Rd	B		2								Rd8 decimal adjust → Rd8	—	*	↕	↕	*	—	2	
MULXU	MULXU. B Rs, Rd	B		2								Rd8 × Rs8 → Rd16 (unsigned multiplication)	—	—	—	—	—	—	14	
	MULXU. W Rs, ERd	W		2								Rd16 × Rs16 → ERd32 (unsigned multiplication)	—	—	—	—	—	—	22	
MULXS	MULXS. B Rs, Rd	B		4								Rd8 × Rs8 → Rd16 (signed multiplication)	—	—	↕	↕	—	—	16	
	MULXS. W Rs, ERd	W		4								Rd16 × Rs16 → ERd32 (signed multiplication)	—	—	↕	↕	—	—	24	
DIVXU	DIVXU. B Rs, Rd	B		2								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	(6)	(7)	—	—	14	
	DIVXU. W Rs, ERd	W		2								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	(6)	(7)	—	—	22	
DIVXS	DIVXS. B Rs, Rd	B		4								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	(8)	(7)	—	—	16	
	DIVXS. W Rs, ERd	W		4								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	(8)	(7)	—	—	24	
CMP	CMP.B #xx:8, Rd	B	2									Rd8–#xx:8	—	↕	↕	↕	↕	↕	2	
	CMP.B Rs, Rd	B		2								Rd8–Rs8	—	↕	↕	↕	↕	↕	2	
	CMP.W #xx:16, Rd	W	4									Rd16–#xx:16	—	(1)	↕	↕	↕	↕	4	
	CMP.W Rs, Rd	W		2								Rd16–Rs16	—	(1)	↕	↕	↕	↕	2	
	CMP.L #xx:32, ERd	L	6									ERd32–#xx:32	—	(2)	↕	↕	↕	↕	4	
	CMP.L ERs, ERd	L		2								ERd32–ERs32	—	(2)	↕	↕	↕	↕	2	

7. System control instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
TRAPA	TRAPA #x:2	—									2	PC → @–SP CCR → @–SP <vector> → PC	1	—	—	—	—	—	14	16
RTE	RTE	—										CCR ← @SP+ PC ← @SP+	↕	↕	↕	↕	↕	↕	10	
SLEEP	SLEEP	—										Transition to power-down state	—	—	—	—	—	—	2	
LDC	LDC #xx:8, CCR	B	2									#xx:8 → CCR	↕	↕	↕	↕	↕	↕	2	
	LDC Rs, CCR	B		2								Rs8 → CCR	↕	↕	↕	↕	↕	↕	2	
	LDC @ERs, CCR	W			4							@ERs → CCR	↕	↕	↕	↕	↕	↕	6	
	LDC @(d:16, ERs), CCR	W				6						@(d:16, ERs) → CCR	↕	↕	↕	↕	↕	↕	8	
	LDC @(d:24, ERs), CCR	W				10						@(d:24, ERs) → CCR	↕	↕	↕	↕	↕	↕	12	
	LDC @ERs+, CCR	W					4					@ERs → CCR ERs32+2 → ERs32	↕	↕	↕	↕	↕	↕	8	
	LDC @aa:16, CCR	W						6				@aa:16 → CCR	↕	↕	↕	↕	↕	↕	8	
	LDC @aa:24, CCR	W						8				@aa:24 → CCR	↕	↕	↕	↕	↕	↕	10	
STC	STC CCR, Rd	B		2								CCR → Rd8	—	—	—	—	—	—	2	
	STC CCR, @ERd	W			4							CCR → @ERd	—	—	—	—	—	—	6	
	STC CCR, @(d:16, ERd)	W				6						CCR → @(d:16, ERd)	—	—	—	—	—	—	8	
	STC CCR, @(d:24, ERd)	W				10						CCR → @(d:24, ERd)	—	—	—	—	—	—	12	
	STC CCR, @–ERd	W					4					ERd32–2 → ERd32 CCR → @ERd	—	—	—	—	—	—	8	
	STC CCR, @aa:16	W						6				CCR → @aa:16	—	—	—	—	—	—	8	
	STC CCR, @aa:24	W						8				CCR → @aa:24	—	—	—	—	—	—	10	
ANDC	ANDC #xx:8, CCR	B	2									CCR∧#xx:8 → CCR	↕	↕	↕	↕	↕	↕	2	
ORC	ORC #xx:8, CCR	B	2									CCR∨#xx:8 → CCR	↕	↕	↕	↕	↕	↕	2	
XORC	XORC #xx:8, CCR	B	2									CCR⊕#xx:8 → CCR	↕	↕	↕	↕	↕	↕	2	
NOP	NOP	—									2	PC ← PC+2	—	—	—	—	—	—	2	

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
MOV	MOV.B Rs, @aa:16	2			1		
	MOV.B Rs, @aa:24	3			1		
	MOV.W #xx:16, Rd	2					
	MOV.W Rs, Rd	1					
	MOV.W @ERs, Rd	1				1	
	MOV.W @(d:16,ERs), Rd	2				1	
	MOV.W @(d:24,ERs), Rd	4				1	
	MOV.W @ERs+, Rd	1				1	2
	MOV.W @aa:16, Rd	2				1	
	MOV.W @aa:24, Rd	3				1	
	MOV.W Rs, @ERd	1				1	
	MOV.W Rs, @(d:16,ERd)	2				1	
	MOV.W Rs, @(d:24,ERd)	4				1	
MOV	MOV.W Rs, @-ERd	1				1	2
	MOV.W Rs, @aa:16	2				1	
	MOV.W Rs, @aa:24	3				1	
	MOV.L #xx:32, ERd	3					
	MOV.L ERs, ERd	1					
	MOV.L @ERs, ERd	2				2	
	MOV.L @(d:16,ERs), ERd	3				2	
	MOV.L @(d:24,ERs), ERd	5				2	
	MOV.L @ERs+, ERd	2				2	2
	MOV.L @aa:16, ERd	3				2	
	MOV.L @aa:24, ERd	4				2	
	MOV.L ERs,@ERd	2				2	
	MOV.L ERs, @(d:16,ERd)	3				2	
	MOV.L ERs, @(d:24,ERd)	5				2	
	MOV.L ERs, @-ERd	2				2	2
	MOV.L ERs, @aa:16	3				2	
	MOV.L ERs, @aa:24	4				2	
MOVFPE	MOVFPE @aa:16, Rd*[2]	2			1		
MOVTPE	MOVTPE Rs,@aa:16*[2]	2			1		

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ROTXR	ROTXR.B Rd	1					
	ROTXR.W Rd	1					
	ROTXR.L ERd	1					
RTE	RTE	2		2			2
RTS	RTS	2		1			2
SHAL	SHAL.B Rd	1					
	SHAL.W Rd	1					
	SHAL.L ERd	1					
SHAR	SHAR.B Rd	1					
	SHAR.W Rd	1					
	SHAR.L ERd	1					
SHLL	SHLL.B Rd	1					
	SHLL.W Rd	1					
	SHLL.L ERd	1					
SHLR	SHLR.B Rd	1					
	SHLR.W Rd	1					
	SHLR.L ERd	1					
SLEEP	SLEEP	1					
STC	STC CCR, Rd	1					
	STC CCR, @ERd	2				1	
	STC CCR, @(d:16,ERd)	3				1	
	STC CCR, @(d:24,ERd)	5				1	
	STC CCR,@-ERd	2				1	2
	STC CCR, @aa:16	3				1	
	STC CCR, @aa:24	4				1	
SUB	SUB.B Rs, Rd	1					
	SUB.W #xx:16, Rd	2					
	SUB.W Rs, Rd	1					
	SUB.L #xx:32, ERd	3					
	SUB.L ERs, ERd	1					
SUBS	SUBS #1/2/4, ERd	1					

A.4 Combinations of Instructions and Addressing Modes

Table A.5 Combinations of Instructions and Addressing Modes

Functions	Instructions	Addressing Mode												
		#xx	Rn	@ERn	@(d:16.ERn)	@(d:24.ERn)	@ERn+/@ERn	@aa:8	@aa:16	@aa:24	@(d:8.PC)	@(d:16.PC)	@@aa:8	|
Data transfer instructions	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL	—	—	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—	—	WL
	MOVFPE, MOVTPE	—	—	—	—	—	—	—	—	—	—	—	—	—
Arithmetic operations	ADD, CMP	BWL	BWL	—	—	—	—	—	—	—	—	—	—	—
	SUB	WL	BWL	—	—	—	—	—	—	—	—	—	—	—
	ADDX, SUBX	B	B	—	—	—	—	—	—	—	—	—	—	—
	ADDS, SUBS	—	L	—	—	—	—	—	—	—	—	—	—	—
	INC, DEC	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	DAA, DAS	—	B	—	—	—	—	—	—	—	—	—	—	—
	MULXU, MULXS, DIVXU, DIVXS	—	BW	—	—	—	—	—	—	—	—	—	—	—
	NEG	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	EXTU, EXTS	—	WL	—	—	—	—	—	—	—	—	—	—	—
Logical operations	AND, OR, XOR	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	NOT	—	BWL	—	—	—	—	—	—	—	—	—	—	—
Shift operations		—	BWL	—	—	—	—	—	—	—	—	—	—	—
Bit manipulations		—	B	B	—	—	—	B	—	—	—	—	—	—
Branching instructions	BCC, BSR	—	—	—	—	—	—	—	—	—	—	—	—	—
	JMP, JSR	—	—	◯	—	—	—	—	—	—	◯	◯	—	—
	RTS	—	—	—	—	—	—	—	—	◯	—	—	◯	—
System control instructions	TRAPA	—	—	—	—	—	—	—	—	—	—	—	—	◯
	RTE	—	—	—	—	—	—	—	—	—	—	—	—	◯
	SLEEP	—	—	—	—	—	—	—	—	—	—	—	—	◯
	LDC	B	B	W	W	W	W	—	W	W	—	—	—	◯
	STC	—	B	W	W	W	W	—	W	W	—	—	—	—
	ANDC, ORC, XORC	B	—	—	—	—	—	—	—	—	—	—	—	—
	NOP	—	—	—	—	—	—	—	—	—	—	—	—	◯
Block data transfer instructions		—	—	—	—	—	—	—	—	—	—	—	—	BW

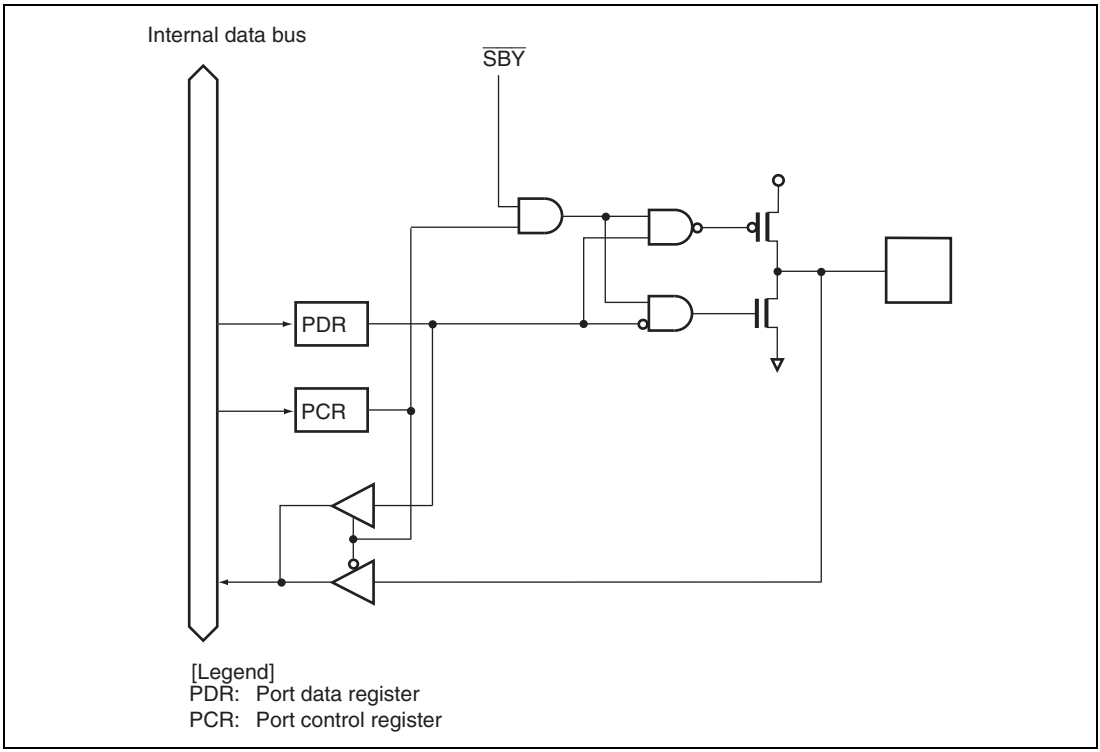

Figure B.14 Port 8 Block Diagram (P87 to P85)

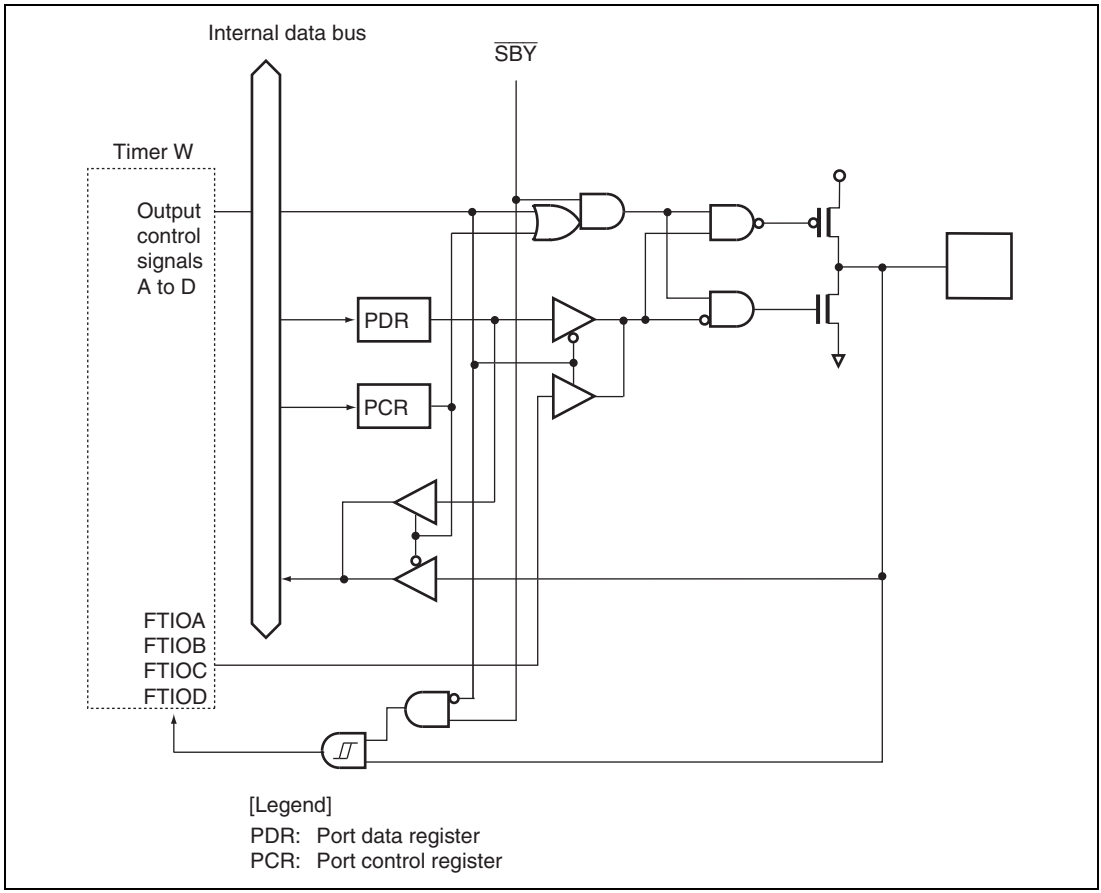

Figure B.15 Port 8 Block Diagram (P84 to P81)