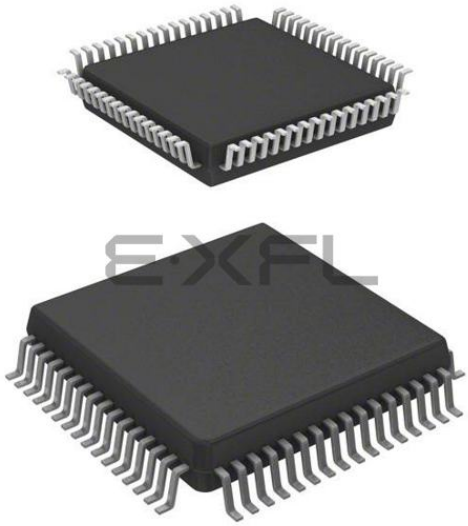


Welcome to E-XFL.COM

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	20MHz
Connectivity	I²C, SCI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	45
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	64-BQFP
Supplier Device Package	64-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/hd64f3684ghv

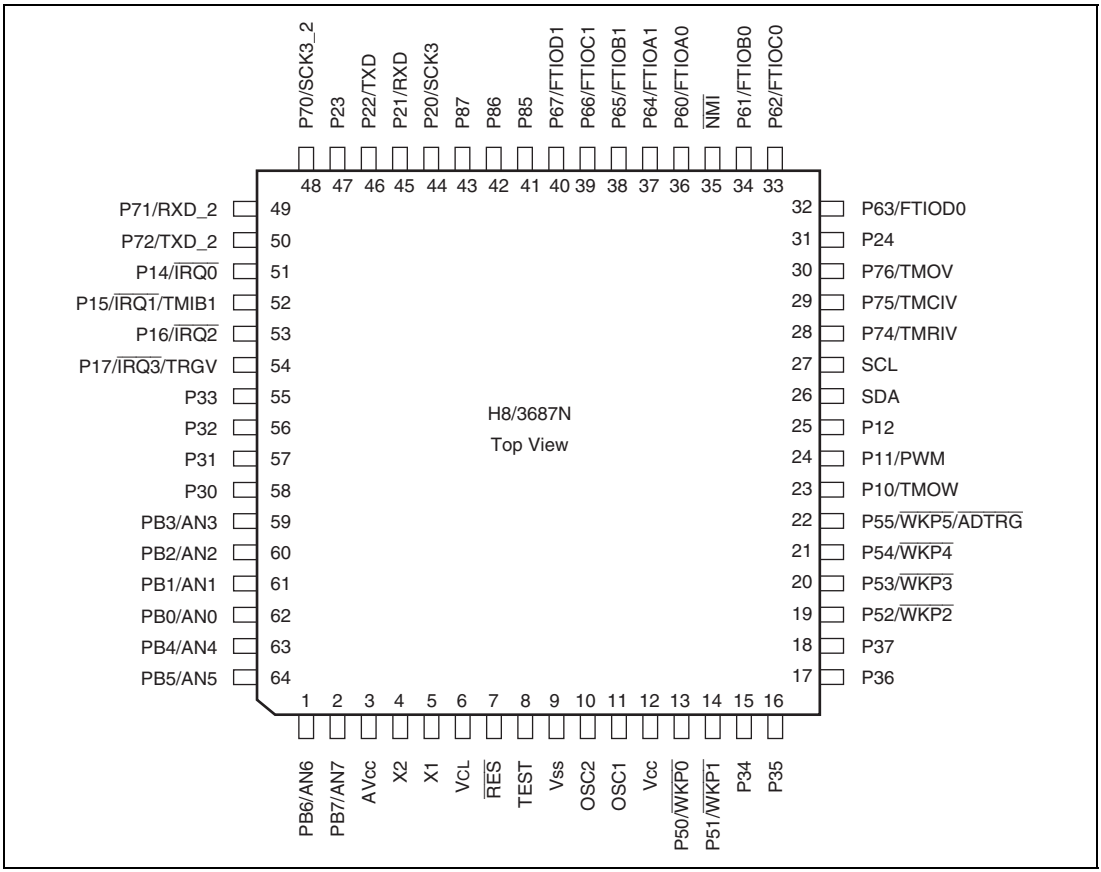


Figure 1.4 Pin Arrangement of H8/3687N (EEPROM Stacked Version) (FP-64E)

Section 2 CPU

This LSI has an H8/300H CPU with an internal 32-bit architecture that is upward-compatible with the H8/300CPU, and supports only normal mode, which has a 64-kbyte address space.

- Upward-compatible with H8/300 CPUs
  - Can execute H8/300 CPUs object programs
  - Additional eight 16-bit extended registers
  - 32-bit transfer and arithmetic and logic instructions are added
  - Signed multiply and divide instructions are added.
- General-register architecture
  - Sixteen 16-bit general registers also usable as sixteen 8-bit registers and eight 16-bit registers, or eight 32-bit registers
- Sixty-two basic instructions
  - 8/16/32-bit data transfer and arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16,ERn) or @(d:24,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 64-kbyte address space
- High-speed operation
  - All frequently-used instructions execute in one or two states
  - 8/16/32-bit register-register add/subtract : 2 state
  - 8 × 8-bit register-register multiply : 14 states
  - 16 ÷ 8-bit register-register divide : 14 states
  - 16 × 16-bit register-register multiply : 22 states
  - 32 ÷ 16-bit register-register divide : 22 states
- Power-down state
  - Transition to power-down state by SLEEP instruction

Bit	Bit Name	Initial Value	R/W	Description
7	I	1	R/W	Interrupt Mask Bit Masks interrupts other than NMI when set to 1. NMI is accepted regardless of the I bit setting. The I bit is set to 1 at the start of an exception-handling sequence.
6	UI	Undefined	R/W	User Bit Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.
5	H	Undefined	R/W	Half-Carry Flag When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.
4	U	Undefined	R/W	User Bit Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit of data as a sign bit.
2	Z	Undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.
1	V	Undefined	R/W	Overflow Flag Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times.
0	C	Undefined	R/W	Carry Flag Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by: • Add instructions, to indicate a carry • Subtract instructions, to indicate a borrow • Shift and rotate instructions, to indicate a carry The carry flag is also used as a bit accumulator by bit manipulation instructions.

- Prior to executing BCLR instruction

```
MOV.B   #3F,  R0L
MOV.B   R0L,  @RAM0
MOV.B   R0L,  @PCR5
```

The PCR5 value (H'3F) is written to a work area in memory (RAM0) as well as to PCR5.

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1	1

- BCLR instruction executed

```
BCLR    #0,     @RAM0
```

The BCLR instructions executed for the PCR5 work area (RAM0).

- After executing BCLR instruction

```
MOV.B   @RAM0, R0L
MOV.B   R0L,  @PCR5
```

The work area (RAM0) value is written to PCR5.

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR5	0	0	1	1	1	1	1	0
PDR5	1	0	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1	0

3.3 Reset Exception Handling

When the $\overline{\text{RES}}$ pin goes low, all processing halts and this LSI enters the reset. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized by the reset. To ensure that this LSI is reset at power-up, hold the $\overline{\text{RES}}$ pin low until the clock pulse generator output stabilizes. To reset the chip during operation, hold the $\overline{\text{RES}}$ pin low for at least 10 system clock cycles. When the $\overline{\text{RES}}$ pin goes high after being held low for the necessary time, this LSI starts reset exception handling. The reset exception handling sequence is shown in figure 3.1. However, for the reset exception handling sequence of the product with on-chip power-on reset circuit, refer to section 20, Power-On Reset and Low-Voltage Detection Circuits (Optional).

The reset exception handling sequence is as follows:

1. Set the I bit in the condition code register (CCR) to 1.
2. The CPU generates a reset exception handling vector address (from H'0000 to H'0001), the data in that address is sent to the program counter (PC) as the start address, and program execution starts from that address.

3.4 Interrupt Exception Handling

3.4.1 External Interrupts

As the external interrupts, there are NMI, IRQ3 to IRQ0, and WKP5 to WKP0 interrupts.

NMI Interrupt

NMI interrupt is requested by input signal edge to pin $\overline{\text{NMI}}$. This interrupt is detected by either rising edge sensing or falling edge sensing, depending on the setting of bit NMIEG in IEGR1.

NMI is the highest-priority interrupt, and can always be accepted without depending on the I bit value in CCR.

IRQ3 to IRQ0 Interrupts

IRQ3 to IRQ0 interrupts are requested by input signals to pins $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$. These four interrupts are given different vector addresses, and are detected individually by either rising edge sensing or falling edge sensing, depending on the settings of bits IEG3 to IEG0 in IEGR1.

When pins $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$ are designated for interrupt input in PMR1 and the designated signal edge is input, the corresponding bit in IRR1 is set to 1, requesting the CPU of an interrupt. These interrupts can be masked by setting bits IEN3 to IEN0 in IENR1.

3.5 Usage Notes

3.5.1 Interrupts after Reset

If an interrupt is accepted after a reset and before the stack pointer (SP) is initialized, the PC and CCR will not be saved correctly, leading to a program crash. To prevent this, all interrupt requests, including NMI, are disabled immediately after a reset. Since the first instruction of a program is always executed immediately after the reset state ends, make sure that this instruction initializes the stack pointer (example: MOV.W #xx: 16, SP).

3.5.2 Notes on Stack Area Use

When word data is accessed, the least significant bit of the address is regarded as 0. Access to the stack always takes place in word size, so the stack pointer (SP: R7) should never indicate an odd address. Use PUSH Rn (MOV.W Rn, @–SP) or POP Rn (MOV.W @SP+, Rn) to save or restore register values.

3.5.3 Notes on Rewriting Port Mode Registers

When a port mode register is rewritten to switch the functions of external interrupt pins, $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$, and $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$, the interrupt request flag may be set to 1.

When switching a pin function, mask the interrupt before setting the bit in the port mode register. After accessing the port mode register, execute at least one instruction (e.g., NOP), then clear the interrupt request flag from 1 to 0.

Figure 3.4 shows a port mode register setting and interrupt request flag clearing procedure.

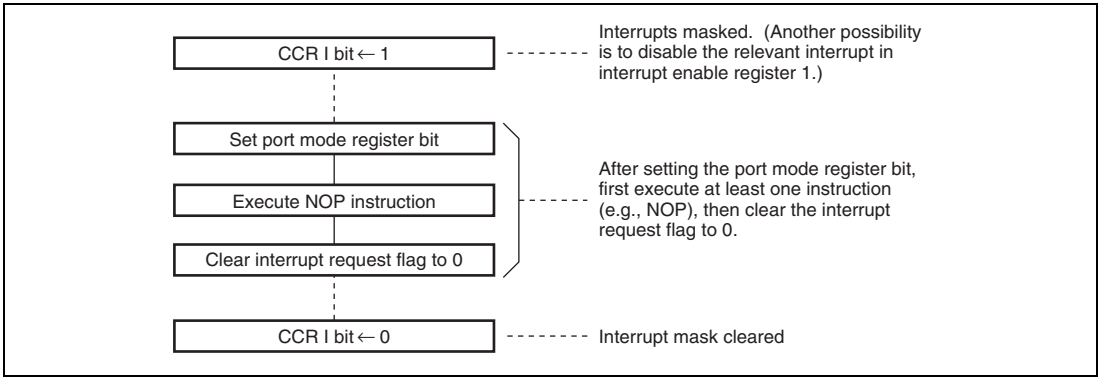


Figure 3.4 Port Mode Register Setting and Interrupt Request Flag Clearing Procedure

6.4 Direct Transition

The CPU can execute programs in two modes: active and subactive modes. A direct transition is a transition between these two modes without stopping program execution. A direct transition can be made by executing a SLEEP instruction while the DTON bit in SYSCR2 is set to 1. The direct transition also enables operating frequency modification in active or subactive mode. After the mode transition, direct transition interrupt exception handling starts.

If the direct transition interrupt is disabled in interrupt enable register 1, a transition is made instead to sleep or subsleep mode. Note that if a direct transition is attempted while the I bit in CCR is set to 1, sleep or subsleep mode will be entered, and the resulting mode cannot be cleared by means of an interrupt.

6.4.1 Direct Transition from Active Mode to Subactive Mode

The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (1).

Direct transition time = {(number of SLEEP instruction execution states) + (number of internal processing states)}× (tcyc before transition) + (number of interrupt exception handling states) × (tsubcyc after transition) (1)

Example
Direct transition time = (2 + 1) × tosc + 14 × 8tw = 3tosc + 112tw
(when the CPU operating clock of $\phi_{osc} \rightarrow \phi_w/8$ is selected)

Legend
tosc: OSC clock cycle time
tw: Watch clock cycle time
tcyc: System clock (ϕ) cycle time
tsubcyc: Subclock (ϕ_{SUB}) cycle time

6.4.2 Direct Transition from Subactive Mode to Active Mode

The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (2).

Direct transition time = {(number of SLEEP instruction execution states) + (number of internal processing states)} × (tsubcyc before transition) + {(waiting time set in bits STS2 to STS0) + (number of interrupt exception handling states)} × (tcyc after transition) (2)

Section 7 ROM

The features of the 56-kbyte or 32-kbyte flash memories built into the flash memory (F-ZTAT) version are summarized below.

- Programming/erase methods
  - The flash memory is programmed 128 bytes at a time. Erase is performed in single-block units. The flash memory is configured as follows: 1 kbyte × 4 blocks, 28 kbytes × 1 block, 16 kbytes × 1 block, and 8 kbytes × 1 block for H8/3687F and 1 kbyte × 4 blocks and 28 kbytes × 1 block for H8/3684F. To erase the entire flash memory, each block must be erased in turn.
- Reprogramming capability
  - The flash memory can be reprogrammed up to 1,000 times.
- On-board programming
  - On-board programming/erasing can be done in boot mode, in which the boot program built into the chip is started to erase or program of the entire flash memory. In normal user program mode, individual blocks can be erased or programmed.
- Programmer mode
  - Flash memory can be programmed/erased in programmer mode using a PROM programmer, as well as in on-board programming mode.
- Automatic bit rate adjustment
  - For data transfer in boot mode, this LSI's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Programming/erasing protection
  - Sets software protection against flash memory programming/erasing.
- Power-down mode
  - Operation of the power supply circuit can be partly halted in subactive mode. As a result, flash memory can be read with low power consumption.

7.1 Block Configuration

Figure 7.1 shows the block configuration of flash memory. The thick lines indicate erasing units, the narrow lines indicate programming units, and the values are addresses. The 56-kbyte flash memory is divided into 1 kbyte × 4 blocks, 28 kbytes × 1 block, 16 kbytes × 1 block, and 8 kbytes × 1 block. The 32-kbyte flash memory is divided into 1 kbyte × 4 blocks and 28 kbytes × 1 blocks. 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.

9.1.2 Port Control Register 1 (PCR1)

PCR1 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 1.

Bit	Bit Name	Initial Value	R/W	Description
7	PCR17	0	W	When the corresponding pin is designated in PMR1 as a general I/O pin, setting a PCR1 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.
6	PCR16	0	W	
5	PCR15	0	W	
4	PCR14	0	W	Bit 3 is a reserved bit.
3	—	—	—	
2	PCR12	0	W	
1	PCR11	0	W	
0	PCR10	0	W	

9.1.3 Port Data Register 1 (PDR1)

PDR1 is a general I/O port data register of port 1.

Bit	Bit Name	Initial Value	R/W	Description
7	P17	0	R/W	PDR1 stores output data for port 1 pins.
6	P16	0	R/W	If PDR1 is read while PCR1 bits are set to 1, the value stored in PDR1 are read. If PDR1 is read while PCR1 bits are cleared to 0, the pin states are read regardless of the value stored in PDR1.
5	P15	0	R/W	
4	P14	0	R/W	
3	—	1	—	Bit 3 is a reserved bit. This bit is always read as 1.
2	P12	0	R/W	
1	P11	0	R/W	
0	P10	0	R/W	

9.5 Port 6

Port 6 is a general I/O port also functioning as a timer Z I/O pin. Each pin of the port 6 is shown in figure 9.5. The register setting of the timer Z has priority for functions of the pins for both uses.

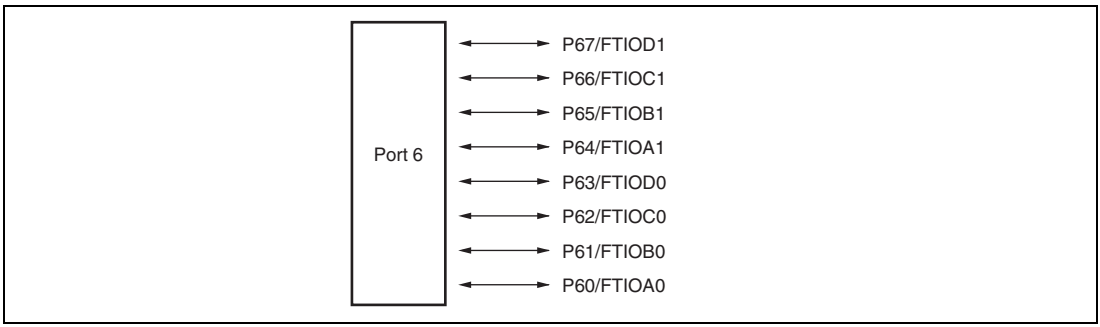


Figure 9.5 Port 6 Pin Configuration

Port 6 has the following registers.

- Port control register 6 (PCR6)
- Port data register 6 (PDR6)

9.5.1 Port Control Register 6 (PCR6)

PCR6 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 6.

Bit	Bit Name	Initial Value	R/W	Description
7	PCR67	0	W	When each of the port 6 pins P67 to P60 functions as a general I/O port, setting a PCR6 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.
6	PCR66	0	W	
5	PCR65	0	W	
4	PCR64	0	W	
3	PCR63	0	W	
2	PCR62	0	W	
1	PCR61	0	W	
0	PCR60	0	W	

12.5 Timer V Application Examples

12.5.1 Pulse Output with Arbitrary Duty Cycle

Figure 12.9 shows an example of output of pulses with an arbitrary duty cycle.

1. Set bits CCLR1 and CCLR0 in TCRV0 so that TCNTV will be cleared by compare match with TCORA.
2. Set bits OS3 to OS0 in TCSRV so that the output will go to 1 at compare match with TCORA and to 0 at compare match with TCORB.
3. Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired clock source.
4. With these settings, a waveform is output without further software intervention, with a period determined by TCORA and a pulse width determined by TCORB.

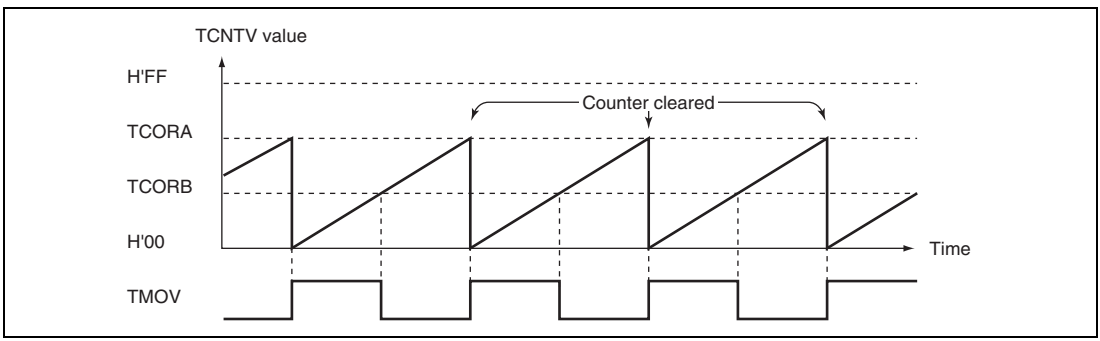


Figure 12.9 Pulse Output Example

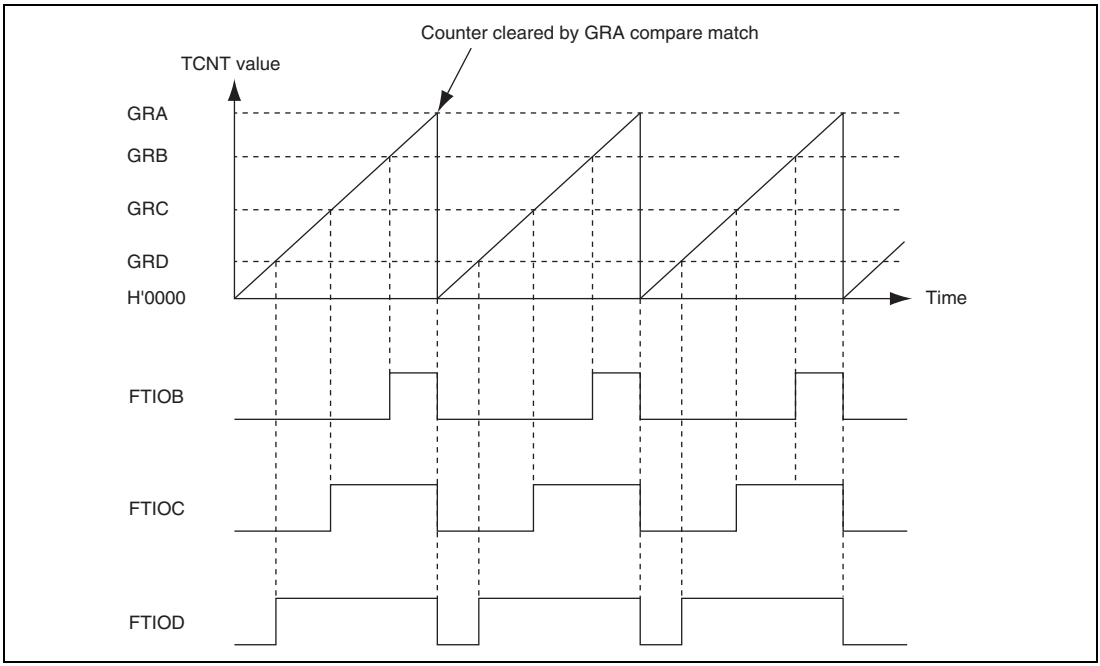


Figure 13.23 Example of PWM Mode Operation (2)

Figures 13.24 (when TOB, TOC, and TOD = 1, POLB, POLC, and POLD = 0) and 13.25 (when TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 1) show examples of the output of PWM waveforms with duty cycles of 0% and 100% in PWM mode.

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/W	Parity Error [Setting condition] • When a parity error is detected during reception [Clearing condition] • When 0 is written to PER after reading PER = 1
2	TEND	1	R	Transmit End [Setting conditions] • When the TE bit in SCR3 is 0 • When TDRE = 1 at transmission of the last bit of a 1-frame serial transmit character [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the transmit data is written to TDR
1	MPBR	0	R	Multiprocessor Bit Receive MPBR stores the multiprocessor bit in the receive character data. When the RE bit in SCR3 is cleared to 0, its state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT stores the multiprocessor bit to be added to the transmit character data.

Table 16.5 Examples of BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (1)

Bit Rate (bit/s)	Operating Frequency ϕ (MHz)									
	2		4		8		10		16	
	n	N	n	N	n	N	n	N	n	N
110	3	70	—	—	—	—	—	—		
250	2	124	2	249	3	124	—	—	3	249
500	1	249	2	124	2	249	—	—	3	124
1k	1	124	1	249	2	124	—	—	2	249
2.5k	0	199	1	99	1	199	1	249	2	99
5k	0	99	0	199	1	99	1	124	1	199
10k	0	49	0	99	0	199	0	249	1	99
25k	0	19	0	39	0	79	0	99	0	159
50k	0	9	0	19	0	39	0	49	0	79
100k	0	4	0	9	0	19	0	24	0	39
250k	0	1	0	3	0	7	0	9	0	15
500k	0	0*	0	1	0	3	0	4	0	7
1M			0	0*	0	1	—	—	0	3
2M					0	0*	—	—	0	1
2.5M							0	0*	—	—
4M									0	0*

Legend:

Blank : No setting is available.

— : A setting is available but error occurs.

* : Continuous transfer is not possible.

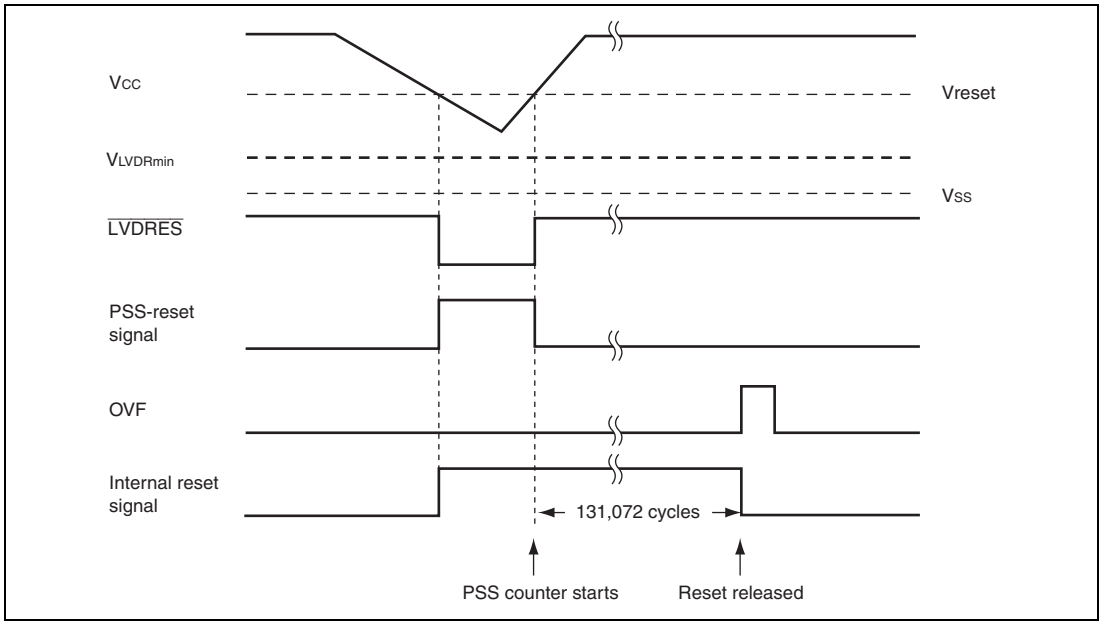


Figure 20.3 Operational Timing of LVDR Circuit

LVDI (Interrupt by Low Voltage Detect) Circuit:

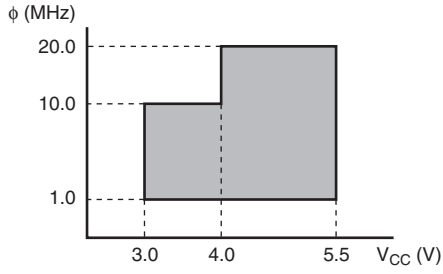
Figure 20.4 shows the timing of LVDI functions. The LVDI enters the module-standby state after a power-on reset is canceled. To operate the LVDI, set the LVDE bit in LVDCR to 1, wait for 50 µs (t_{LVDON}) until the reference voltage and the low-voltage-detection power supply have stabilized by a software timer, etc., then set the LVDDE and LVDUE bits in LVDCR to 1. After that, the output settings of ports must be made. To cancel the low-voltage detection circuit, first the LVDDE and LVDUE bits should all be cleared to 0 and then the LVDE bit should be cleared to 0. The LVDE bit must not be cleared to 0 at the same timing as the LVDDE and LVDUE bits because incorrect operation may occur.

When the power-supply voltage falls below Vint (D) (typ. = 3.7 V) voltage, the LVDI clears the $\overline{\text{LVDINT}}$ signal to 0 and the LVDDF bit in LVDSR is set to 1. If the LVDDE bit is 1 at this time, an IRQ0 interrupt request is simultaneously generated. In this case, the necessary data must be saved in the external EEPROM, etc, and a transition must be made to standby mode or subsleep mode. Until this processing is completed, the power supply voltage must be higher than the lower limit of the guaranteed operating voltage.

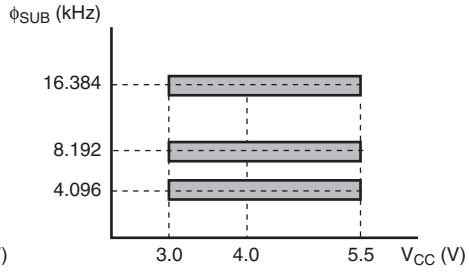
When the power-supply voltage does not fall below Vreset1 (typ. = 2.3 V) voltage but rises above Vint (U) (typ. = 4.0 V) voltage, the LVDI sets the $\overline{\text{LVDINT}}$ signal to 1. If the LVDUE bit is 1 at

Register Name	Reset	Active	Sleep	Subactive	Subsleep	Standby	Module
TCORA	Initialized	—	—	Initialized	Initialized	Initialized	Timer V
TCORB	Initialized	—	—	Initialized	Initialized	Initialized	
TCNTV	Initialized	—	—	Initialized	Initialized	Initialized	
TCRV1	Initialized	—	—	Initialized	Initialized	Initialized	
SMR	Initialized	—	—	Initialized	Initialized	Initialized	SCI3
BRR	Initialized	—	—	Initialized	Initialized	Initialized	
SCR3	Initialized	—	—	Initialized	Initialized	Initialized	
TDR	Initialized	—	—	Initialized	Initialized	Initialized	
SSR	Initialized	—	—	Initialized	Initialized	Initialized	
RDR	Initialized	—	—	Initialized	Initialized	Initialized	
ADDRA	Initialized	—	—	Initialized	Initialized	Initialized	A/D converter
ADDRB	Initialized	—	—	Initialized	Initialized	Initialized	
ADDRC	Initialized	—	—	Initialized	Initialized	Initialized	
ADDRD	Initialized	—	—	Initialized	Initialized	Initialized	
ADCSR	Initialized	—	—	Initialized	Initialized	Initialized	
ADCR	Initialized	—	—	Initialized	Initialized	Initialized	
PWDRL	Initialized	—	—	—	—	—	14bit PWM
PWDRU	Initialized	—	—	—	—	—	
PWCR	Initialized	—	—	—	—	—	
TCSRWD	Initialized	—	—	—	—	—	WDT*[2]
TCWD	Initialized	—	—	—	—	—	
TMWD	Initialized	—	—	—	—	—	
ABRKCR	Initialized	—	—	—	—	—	Address break
ABRKSR	Initialized	—	—	—	—	—	
BARH	Initialized	—	—	—	—	—	
BARL	Initialized	—	—	—	—	—	
BDRH	Initialized	—	—	—	—	—	
BDRL	Initialized	—	—	—	—	—	
PUCR1	Initialized	—	—	—	—	—	I/O port
PUCR5	Initialized	—	—	—	—	—	
PDR1	Initialized	—	—	—	—	—	

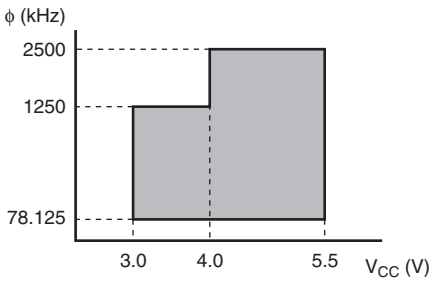
Power Supply Voltage and Operating Frequency Range



- AV_{CC} = 3.3 to 5.5 V
- Active mode
- Sleep mode
  (When MA2 in SYSCR2 = 0)



- AV_{CC} = 3.3 to 5.5 V
- Subactive mode
- Subsleep mode



- AV_{CC} = 3.3 to 5.5 V
- Active mode
- Sleep mode
  (When MA2 in SYSCR2 = 1)

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BTST	BTST #xx:3, Rd	1					
	BTST #xx:3, @ERd	2			1		
	BTST #xx:3, @aa:8	2			1		
	BTST Rn, Rd	1					
	BTST Rn, @ERd	2			1		
	BTST Rn, @aa:8	2			1		
BXOR	BXOR #xx:3, Rd	1					
	BXOR #xx:3, @ERd	2			1		
	BXOR #xx:3, @aa:8	2			1		
CMP	CMP.B #xx:8, Rd	1					
	CMP.B Rs, Rd	1					
	CMP.W #xx:16, Rd	2					
	CMP.W Rs, Rd	1					
	CMP.L #xx:32, ERd	3					
	CMP.L ERs, ERd	1					
DAA	DAA Rd	1					
DAS	DAS Rd	1					
DEC	DEC.B Rd	1					
	DEC.W #1/2, Rd	1					
	DEC.L #1/2, ERd	1					
DUVXS	DIVXS.B Rs, Rd	2					12
	DIVXS.W Rs, ERd	2					20
DIVXU	DIVXU.B Rs, Rd	1					12
	DIVXU.W Rs, ERd	1					20
EEPMOV	EEPMOV.B	2			2n+2*[1]		
	EEPMOV.W	2			2n+2*[1]		
EXTS	EXTS.W Rd	1					
	EXTS.L ERd	1					
EXTU	EXTU.W Rd	1					
	EXTU.L ERd	1					

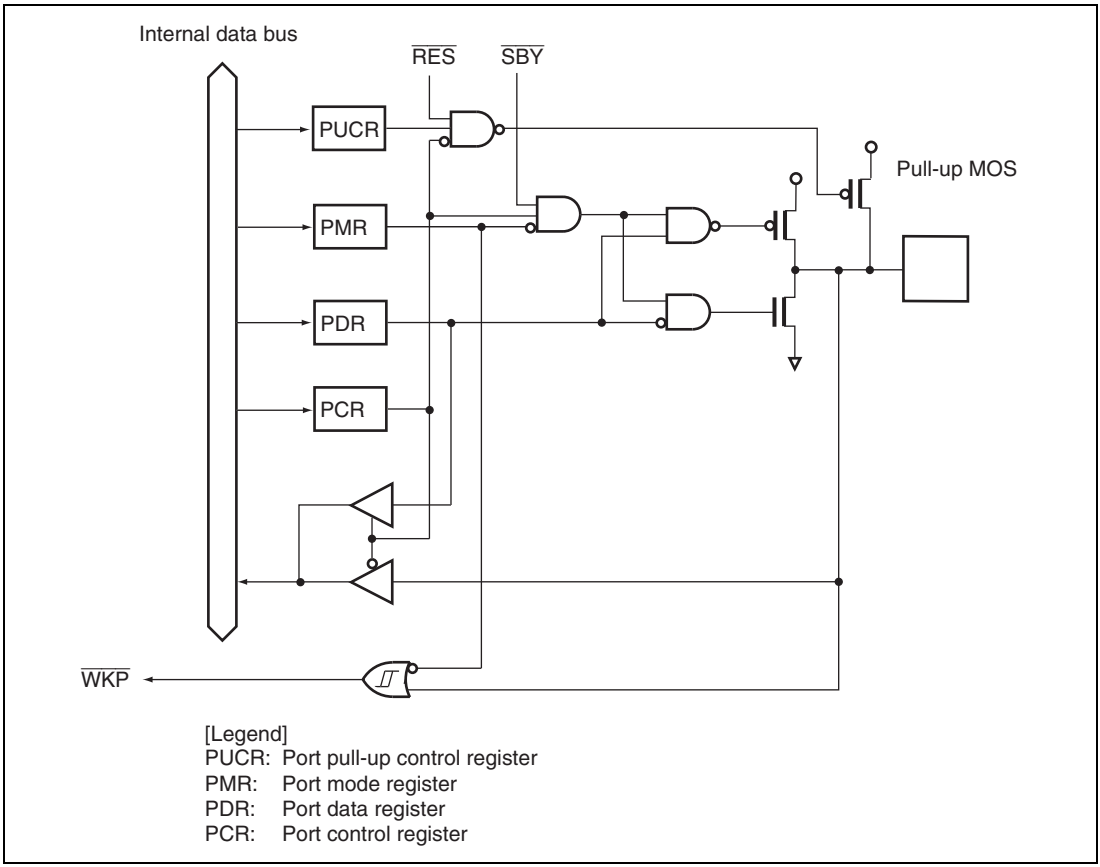


Figure B.14 Port 5 Block Diagram (P54 to P50)