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	Active
Core Processor	H8/300H
Core Size	16-Bit
Speed	16MHz
Connectivity	SCI
Peripherals	PWM, WDT
Number of I/O	26
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 4x10b
Oscillator Type	Internal
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/hd64f3672fpv

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

Table 2.8 System Control Instructions

Instruction	Size*	Function
TRAPA	—	Starts trap-instruction exception handling.
RTE	—	Returns from an exception-handling routine.
SLEEP	—	Causes a transition to a power-down state.
LDC	B/W	(EAs) → CCR Moves the source operand contents to the CCR. The CCR size is one byte, but in transfer from memory, data is read by word access.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers the CCR contents to a destination location. The condition code register size is one byte, but in transfer to memory, data is written by word access.
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR with immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR with immediate data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically XORs the CCR with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

Note: * Refers to the operand size.
B: Byte
W: Word

2.6 Basic Bus Cycle

CPU operation is synchronized by a system clock (ϕ) or a subclock (ϕ_{SUB}). The period from a rising edge of ϕ or ϕ_{SUB} to the next rising edge is called one state. A bus cycle consists of two states or three states. The cycle differs depending on whether access is to on-chip memory or to on-chip peripheral modules.

2.6.1 Access to On-Chip Memory (RAM, ROM)

Access to on-chip memory takes place in two states. The data bus width is 16 bits, allowing access in byte or word size. Figure 2.9 shows the on-chip memory access cycle.

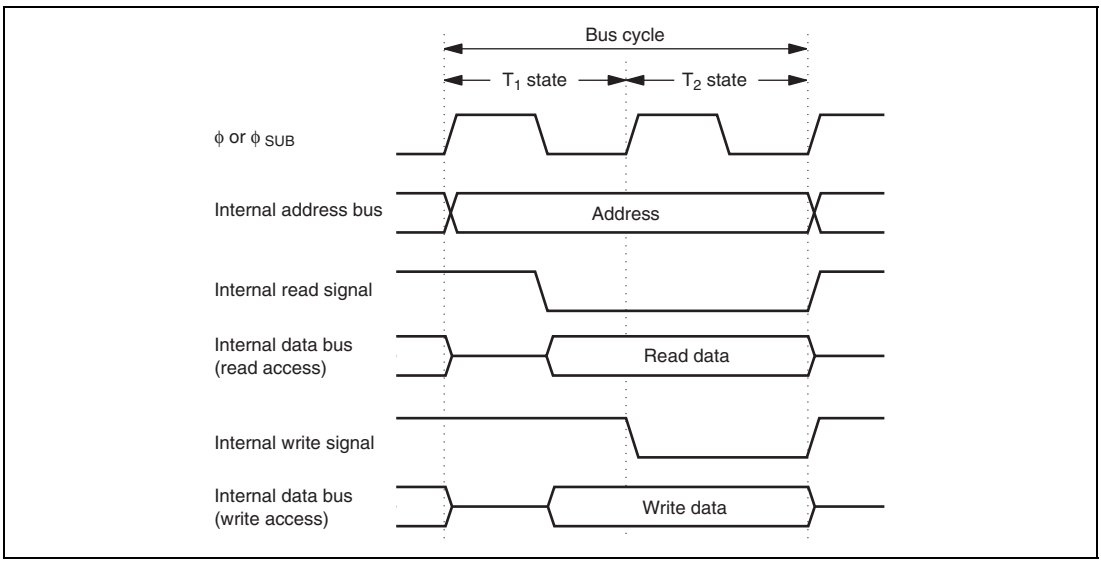


Figure 2.9 On-Chip Memory Access Cycle

3.2 Register Descriptions

Interrupts are controlled by the following registers.

- Interrupt edge select register 1 (IEGR1)
- Interrupt edge select register 2 (IEGR2)
- Interrupt enable register 1 (IENR1)
- Interrupt flag register 1 (IRR1)
- Wakeup interrupt flag register (IWPR)

3.2.1 Interrupt Edge Select Register 1 (IEGR1)

IEGR1 selects the direction of an edge that generates interrupt requests of pins and $\overline{\text{IRQ3}}$ and $\overline{\text{IRQ0}}$.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0.
6	—	1	—	Reserved
5	—	1	—	These bits are always read as 1.
4	—	1	—	
3	IEG3	0	R/W	IRQ3 Edge Select 0: Falling edge of $\overline{\text{IRQ3}}$ pin input is detected 1: Rising edge of $\overline{\text{IRQ3}}$ pin input is detected
2	—	0	—	Reserved
1	—	0	—	These bits are always read as 0.
0	IEG0	0	R/W	IRQ0 Edge Select 0: Falling edge of $\overline{\text{IRQ0}}$ pin input is detected 1: Rising edge of $\overline{\text{IRQ0}}$ pin input is detected

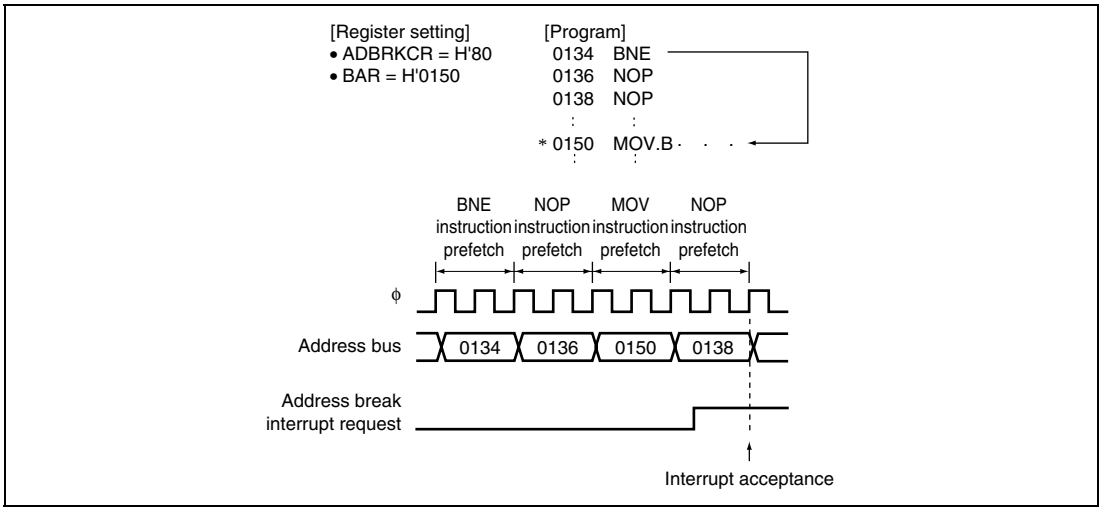


Figure 4.5 Operation when the Instruction Set is not Executed and does not Branch due to Conditions not Being Satisfied

5. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower two bits are B'00. Verify data can be read in longwords from the address to which a dummy write was performed.
6. If the read data is not erased successfully, set erase mode again, and repeat the erase/erase-verify sequence as before. The maximum number of repetitions of the erase/erase-verify sequence is 100.

7.4.3 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including the $\overline{\text{NMI}}$ interrupt, are disabled while flash memory is being programmed or erased, or while the boot program is executing, for the following three reasons:

1. Interrupt during programming/erasing may cause a violation of the programming or erasing algorithm, with the result that normal operation cannot be assured.
2. If interrupt exception handling starts before the vector address is written or during programming/erasing, a correct vector cannot be fetched and the CPU malfunctions.
3. If an interrupt occurs during boot program execution, normal boot mode sequence cannot be carried out.

10.5.2 Pulse Output with Arbitrary Pulse Width and Delay from TRGV Input

The trigger function can be used to output a pulse with an arbitrary pulse width at an arbitrary delay from the TRGV input, as shown in figure 10.10. To set up this output:

1. Set bits CCLR1 and CCLR0 in TCRV0 so that TCNTV will be cleared by compare match with TCORB.
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 TVEG1 and TVEG0 in TCRV1 and set TRGE to select the falling edge of the TRGV input.
4. Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired clock source.
5. After these settings, a pulse waveform will be output without further software intervention, with a delay determined by TCORA from the TRGV input, and a pulse width determined by (TCORB – TCORA).

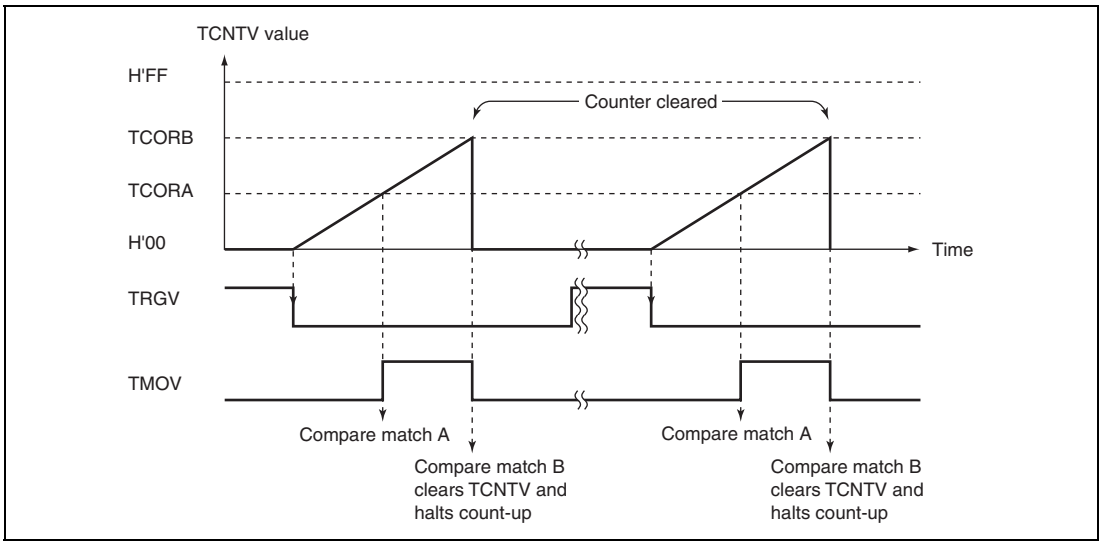


Figure 10.10 Example of Pulse Output Synchronized to TRGV Input

10.6 Usage Notes

The following types of contention or operation can occur in timer V operation.

1. Writing to registers is performed in the T3 state of a TCNTV write cycle. If a TCNTV clear signal is generated in the T3 state of a TCNTV write cycle, as shown in figure 10.11, clearing takes precedence and the write to the counter is not carried out. If counting-up is generated in the T3 state of a TCNTV write cycle, writing takes precedence.
2. If a compare match is generated in the T3 state of a TCORA or TCORB write cycle, the write to TCORA or TCORB takes precedence and the compare match signal is inhibited. Figure 10.12 shows the timing.
3. If compare matches A and B occur simultaneously, any conflict between the output selections for compare match A and compare match B is resolved by the following priority: toggle output > output 1 > output 0.
4. Depending on the timing, TCNTV may be incremented by a switch between different internal clock sources. When TCNTV is internally clocked, an increment pulse is generated from the falling edge of an internal clock signal, that is divided system clock (ϕ). Therefore, as shown in figure 10.3 the switch is from a high clock signal to a low clock signal, the switchover is seen as a falling edge, causing TCNTV to increment. TCNTV can also be incremented by a switch between internal and external clocks.

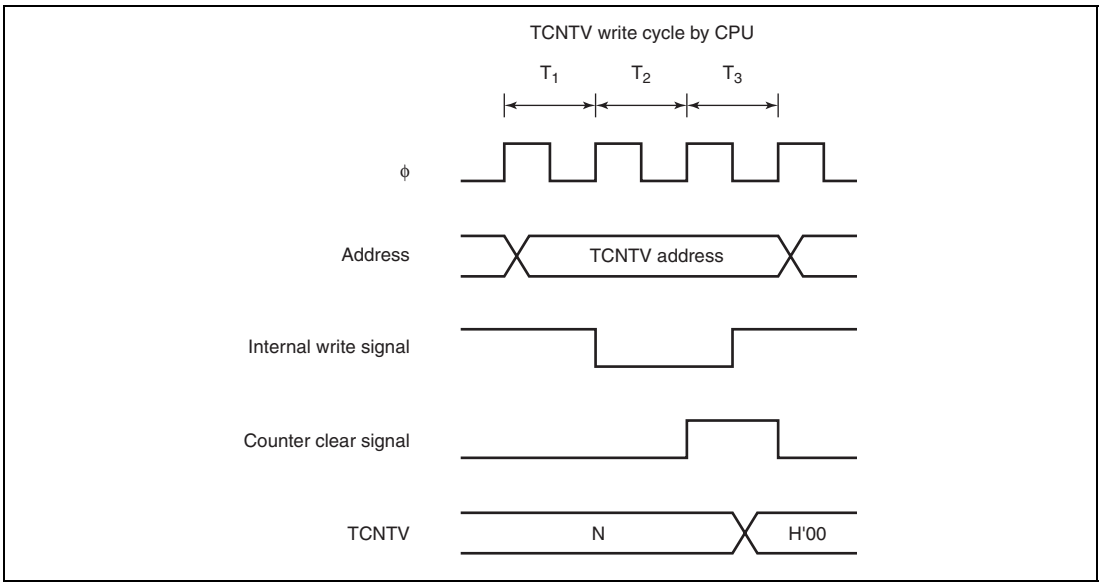


Figure 10.11 Contention between TCNTV Write and Clear

Section 11 Timer W

The timer W has a 16-bit timer having output compare and input capture functions. The timer W can count external events and output pulses with an arbitrary duty cycle by compare match between the timer counter and four general registers. Thus, it can be applied to various systems.

11.1 Features

- Selection of five counter clock sources: four internal clocks (ϕ,ϕ/2, ϕ/4, and ϕ/8) and an external clock (external events can be counted)
- Capability to process up to four pulse outputs or four pulse inputs
- Four general registers:
  - — Independently assignable output compare or input capture functions
  - — Usable as two pairs of registers; one register of each pair operates as a buffer for the output compare or input capture register
- Four selectable operating modes:
  - — Waveform output by compare match
    Selection of 0 output, 1 output, or toggle output
  - — Input capture function
    Rising edge, falling edge, or both edges
  - — Counter clearing function
    Counters can be cleared by compare match
  - — PWM mode
    Up to three-phase PWM output can be provided with desired duty ratio.
- Any initial timer output value can be set
- Five interrupt sources
  Four compare match/input capture interrupts and an overflow interrupt.

Table 11.1 summarizes the timer W functions, and figure 11.1 shows a block diagram of the timer W.

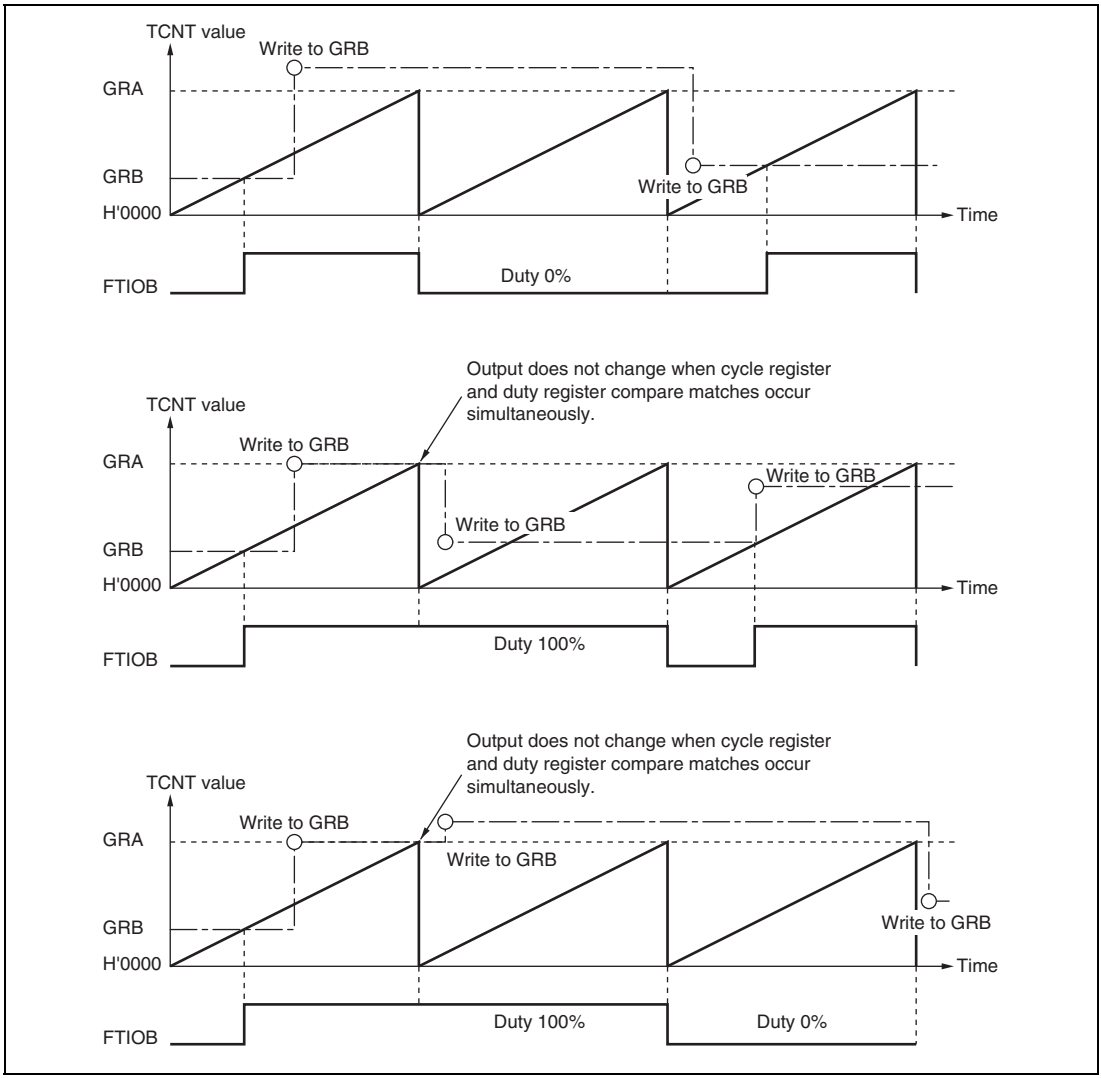


Figure 11.12 PWM Mode Example
(TOB, TOC, and TOD = 0: initial output values are set to 0)

5. The TOA to TOD bits in TCRW decide the value of the FTIO pin, which is output until the first compare match occurs. Once a compare match occurs and this compare match changes the values of FTIOA to FTIOD output, the values of the FTIOA to FTIOD pin output and the values read from the TOA to TOD bits may differ. Moreover, when the writing to TCRW and the generation of the compare match A to D occur at the same timing, the writing to TCRW has the priority. Thus, output change due to the compare match is not reflected to the FTIOA to FTIOD pins. Therefore, when bit manipulation instruction is used to write to TCRW, the values of the FTIOA to FTIOD pin output may result in an unexpected result. When TCRW is to be written to while compare match is operating, stop the counter once before accessing to TCRW, read the port 8 state to reflect the values of FTIOA to FTIOD output, to TOA to TOD, and then restart the counter. Figure 11.26 shows an example when the compare match and the bit manipulation instruction to TCRW occur at the same timing.

TCRW has been set to H'06. Compare match B and compare match C are used. The FTIOB pin is in the 1 output state, and is set to the toggle output or the 0 output by compare match B.
When BCLR#2, @TCRW is executed to clear the TOC bit (the FTIOC signal is low) and compare match B occurs at the same timing as shown below, the H'02 writing to TCRW has priority and compare match B does not drive the FTIOB signal low; the FTIOB signal remains high.

Bit	7	6	5	4	3	2	1	0
TCRW	CCLR	CKS2	CKS1	CKS0	TOD	TOC	TOB	TOA
Set value	0	0	0	0	0	1	1	0

BCLR#2, @TCRW
(1) TCRW read operation: Read H'06
(2) Modify operation: Modify H'06 to H'02
(3) Write operation to TCRW: Write H'02

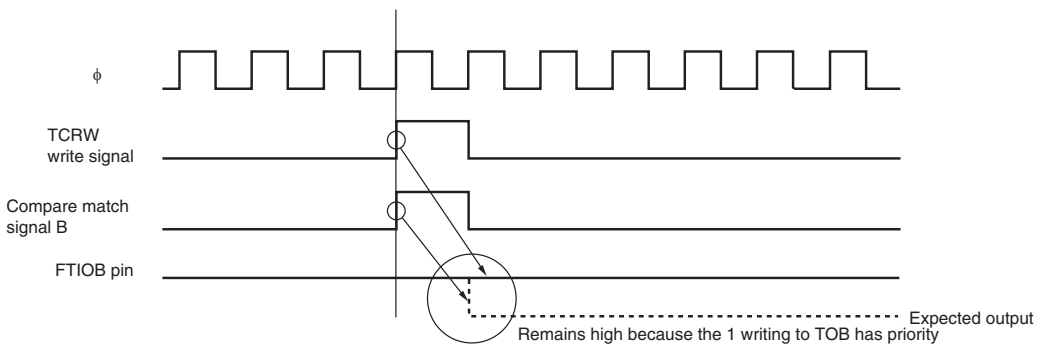


Figure 11.26 When Compare Match and Bit Manipulation Instruction to TCRW Occur at the Same Timing

13.3.7 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

13.4.4 Serial Data Reception

Figure 13.7 shows an example of operation for reception in asynchronous mode. In serial reception, the SCI operates as described below.

1. The SCI3 monitors the communication line. If a start bit is detected, the SCI3 performs internal synchronization, receives data in RSR, and checks the parity bit and stop bit.
2. If an overrun error occurs (when reception of the next data is completed while the RDRF flag is still set to 1), the OER bit in SSR is set to 1. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.
3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated.
4. If a framing error is detected (when the stop bit is 0), the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated.
5. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt request is generated. Continuous reception is possible because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has been completed.

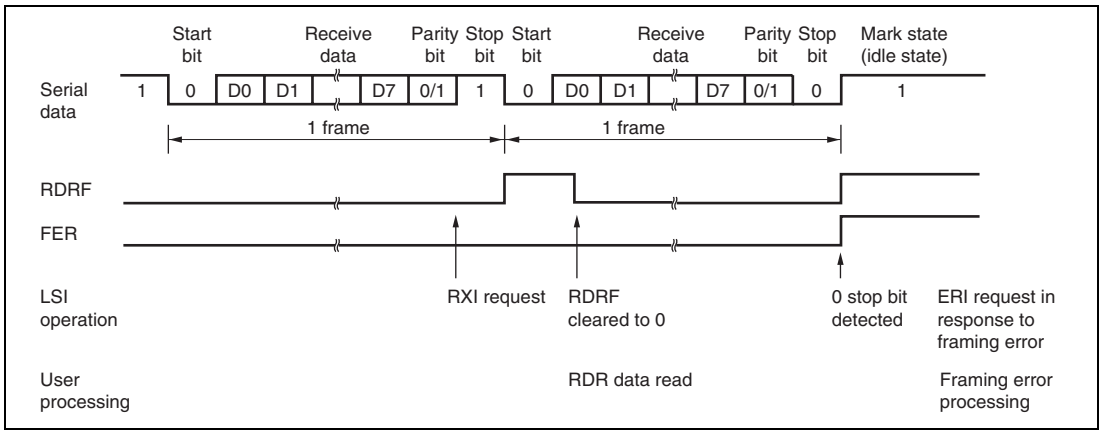
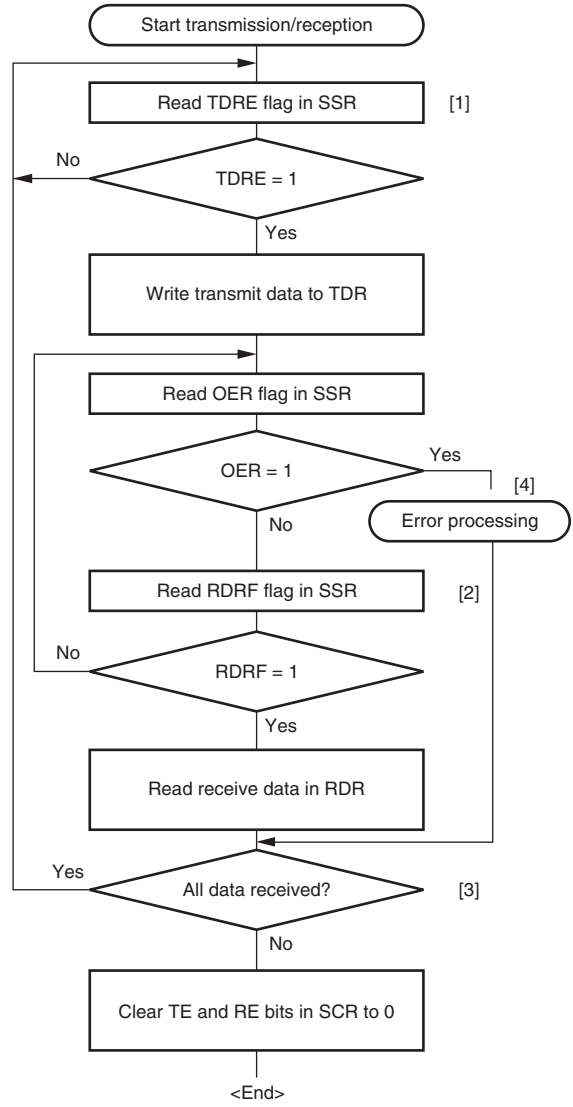


Figure 13.7 Example SCI3 Operation in Reception in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)



[1] Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR.
When data is written to TDR, the TDRE flag is automatically cleared to 0.

[2] Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR.
When data is read from RDR, the RDRF flag is automatically cleared to 0.

[3] To continue serial transmission/ reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR.
When data is written to TDR, the TDRE flag is automatically cleared to 0. When data is read from RDR, the RDRF flag is automatically cleared to 0.

[4] If an overrun error occurs, read the OER flag in SSR, and after performing the appropriate error processing, clear the OER flag to 0. Transmission/reception cannot be resumed if the OER flag is set to 1. For overrun error processing, see figure 13.13.

Figure 13.14 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations (Clocked Synchronous Mode)

13.6.1 Multiprocessor Serial Data Transmission

Figure 13.16 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmission cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI3 operations are the same as those in asynchronous mode.

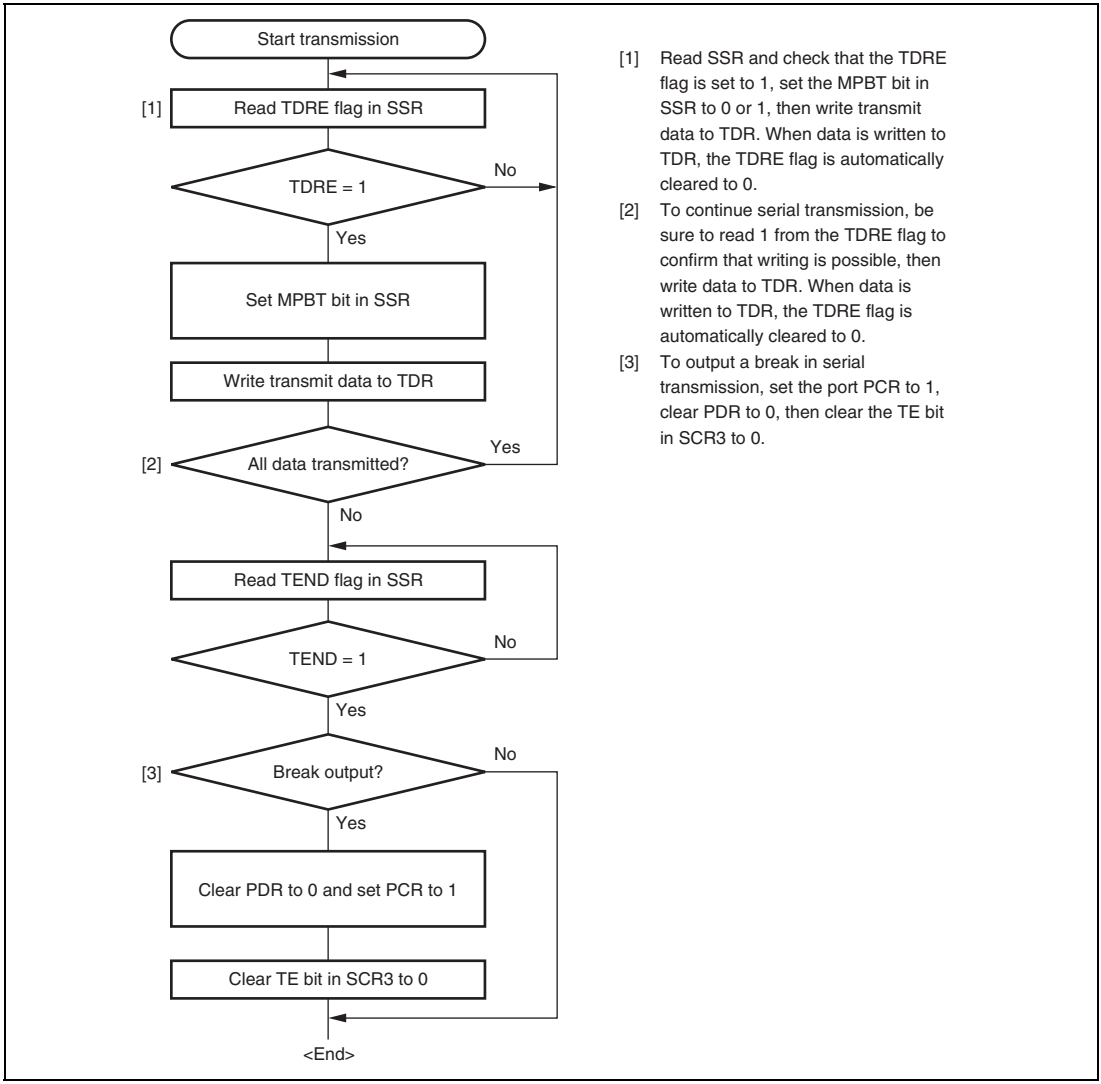


Figure 13.16 Sample Multiprocessor Serial Transmission Flowchart

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Notes
Output high voltage	V_{OH}	P12 to P10, P17 to P14, P22 to P20, P57 to P50, P76 to P74, P84 to P80	V_{CC} = 4.0 V to 5.5 V $-I_{OH}$ = 1.5 mA	V_{CC} – 1.0	—	—	V	
			$-I_{OH}$ = 0.1 mA	V_{CC} – 0.5	—	—		
Output low voltage	V_{OL}	P12 to P10, P17 to P14, P22 to P20, P55 to P50, P76 to P74	V_{CC} = 4.0 V to 5.5 V I_{OL} = 1.6 mA	—	—	0.6	V	
			I_{OL} = 0.4 mA	—	—	0.4		
		P84 to P80	V_{CC} = 4.0 V to 5.5 V I_{OL} = 20.0 mA	—	—	1.5	V	
			V_{CC} = 4.0 V to 5.5 V I_{OL} = 10.0 mA	—	—	1.0		
			V_{CC} = 4.0 V to 5.5 V I_{OL} = 1.6 mA	—	—	0.4		
			I_{OL} = 0.4 mA	—	—	0.4		

Appendix A Instruction Set

A.1 Instruction List

Operand Notation

Symbol	Description
Rd	General (destination*) register
Rs	General (source*) register
Rn	General register*
ERd	General destination register (address register or 32-bit register)
ERs	General source register (address register or 32-bit register)
ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
PC	Program counter
SP	Stack pointer
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
disp	Displacement
→	Transfer from the operand on the left to the operand on the right, or transition from the state on the left to the state on the right
+	Addition of the operands on both sides
–	Subtraction of the operand on the right from the operand on the left
×	Multiplication of the operands on both sides
÷	Division of the operand on the left by the operand on the right
∧	Logical AND of the operands on both sides
∨	Logical OR of the operands on both sides
⊕	Logical exclusive OR of the operands on both sides
¬	NOT (logical complement)
( ), < >	Contents of operand

Note: General registers include 8-bit registers (R0H to R7H and R0L to R7L) and 16-bit registers (R0 to R7 and E0 to E7).

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	

Appendix B I/O Port Block Diagrams

B.1 I/O Port Block

$\overline{\text{RES}}$ goes low in a reset, and $\overline{\text{SBY}}$ goes low in a reset and in standby mode.

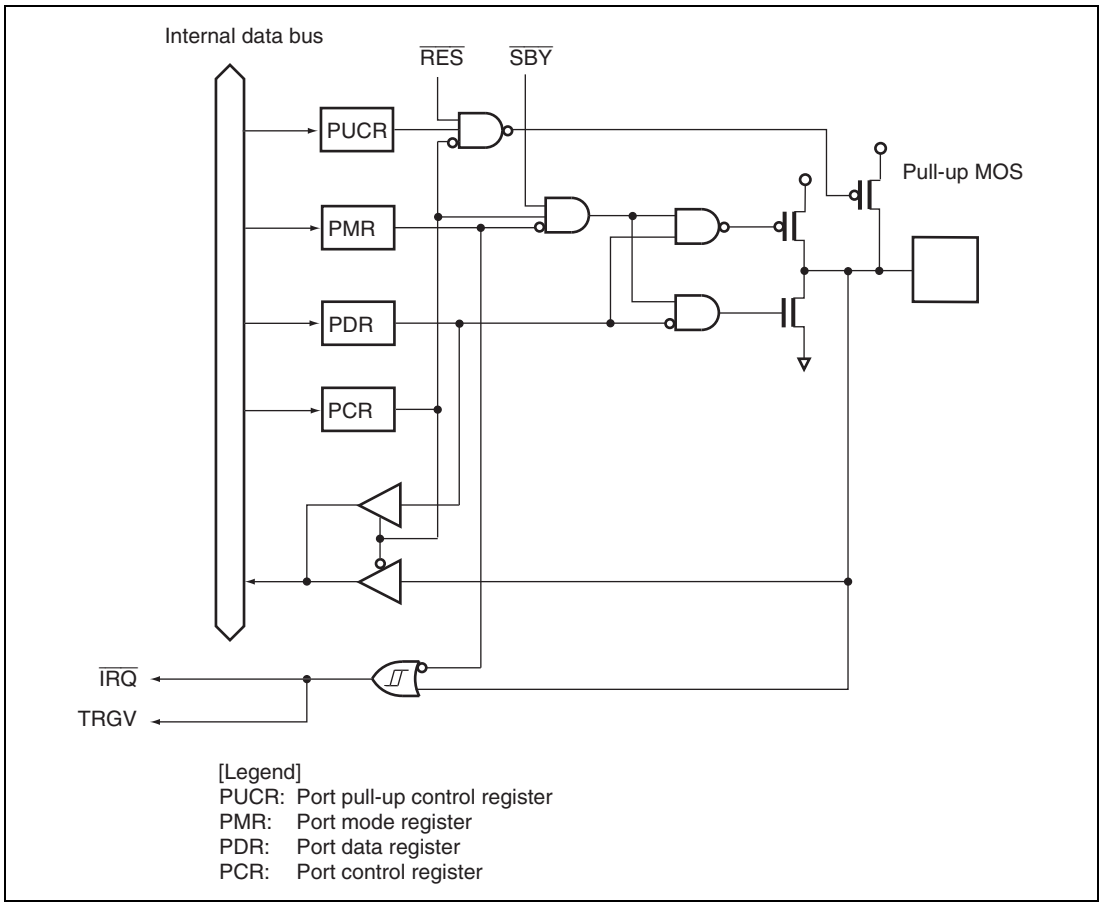


Figure B.1 Port 1 Block Diagram (P17)