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	Obsolete
Core Processor	H8/300H
Core Size	16-Bit
Speed	20MHz
Connectivity	CANbus, SCI, SSU
Peripherals	PWM, WDT
Number of I/O	45
Program Memory Size	56KB (56K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3K 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-LQFP
Supplier Device Package	64-LFQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df36037fzv

Contents

Tables

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 states
  - 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

Table 2.6 Bit Manipulation Instructions (2)

Instruction	Size*	Function
BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C XORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIXOR	B	C ⊕ ¬ (<bit-No.> of <EAd>) → C XORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BLD	B	(<bit-No.> of <EAd>) → C Transfers a specified bit in a general register or memory operand to the carry flag.
BILD	B	¬ (<bit-No.> of <EAd>) → C Transfers the inverse of a specified bit in a general register or memory operand to the carry flag. The bit number is specified by 3-bit immediate data.
BST	B	C → (<bit-No.> of <EAd>) Transfers the carry flag value to a specified bit in a general register or memory operand.
BIST	B	¬ C → (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data.

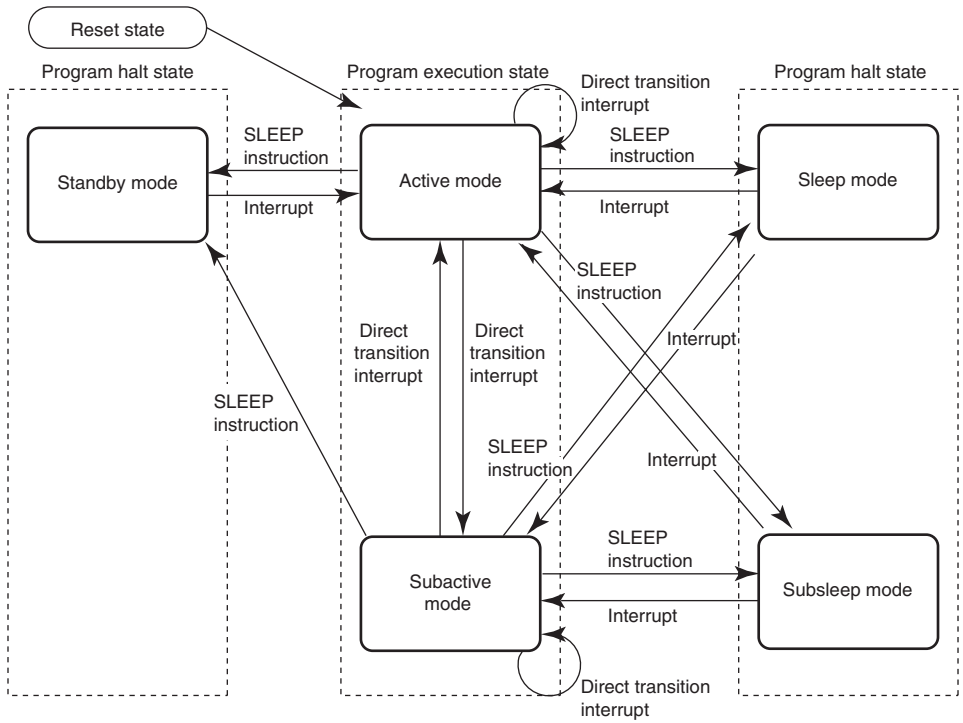
[Legend]

B: Byte

Note: * Refers to the operand size.

6.2 Mode Transitions and States of LSI

Figure 6.1 shows the possible transitions among these operating modes. A transition is made from the program execution state to the program halt state by executing a SLEEP instruction. Interrupts allow for returning from the program halt state to the program execution state. A direct transition between active mode and subactive mode, which are both program execution states, can be made without halting the program. The operating frequency can also be changed in the same modes by making a transition directly from active mode to active mode, and from subactive mode to subactive mode. $\overline{\text{RES}}$ input enables transitions from a mode to the reset state. Table 6.2 shows the transition conditions of each mode after the SLEEP instruction is executed and a mode to return by an interrupt. Table 6.3 shows the internal states of the LSI in each mode.



Notes: 1. To make a transition to another mode by an interrupt, make sure interrupt handling is after the interrupt is accepted.
2. Details on the mode transition conditions are given in table 6.2.

Figure 6.1 Mode Transition Diagram

- P96/HRXD pin

Register	TCMR	PCR9	
Bit Name	**PMR96**	**PCR96**	**Pin Function**
Setting Value	0	0	P96 input pin
		1	P96 output pin
	1	X	HRXD output pin

[Legend]
X: Don't care.

- P95 pin

Register	PCR9	
Bit Name	**PCR95**	**Pin Function**
Setting Value	0	P95 input pin
	1	P95 output pin

- P94 pin

Register	PCR9	
Bit Name	**PCR94**	**Pin Function**
Setting Value	0	P94 input pin
	1	P94 output pin

- P93/SSI pin

Register	PCR9	
Bit Name	**PCR93**	**Pin Function**
Setting Value	0	P93 input pin
	1	P93 output pin
	X	SSI input/SSI output pin

[Legend]
X: Don't care.

Note: When this pin is used as the SSI pin, register settings of the SSU are required. For details, see section 16.4.4, Communication Modes and Pin Functions.

12.3 Register Descriptions

The timer Z has the following registers.

Common

- Timer start register (TSTR)
- Timer mode register (TMDR)
- Timer PWM mode register (TPMR)
- Timer function control register (TFCR)
- Timer output master enable register (TOER)
- Timer output control register (TOCR)

Channel 0

- Timer control register_0 (TCR_0)
- Timer I/O control register A_0 (TIORA_0)
- Timer I/O control register C_0 (TIORC_0)
- Timer status register_0 (TSR_0)
- Timer interrupt enable register_0 (TIER_0)
- PWM mode output level control register_0 (POCR_0)
- Timer counter_0 (TCNT_0)
- General register A_0 (GRA_0)
- General register B_0 (GRB_0)
- General register C_0 (GRC_0)
- General register D_0 (GRD_0)

Channel 1

- Timer control register_1 (TCR_1)
- Timer I/O control register A_1 (TIORA_1)
- Timer I/O control register C_1 (TIORC_1)
- Timer status register_1 (TSR_1)
- Timer interrupt enable register_1 (TIER_1)
- PWM mode output level control register_1 (POCR_1)
- Timer counter_1 (TCNT_1)
- General register A_1 (GRA_1)
- General register B_1 (GRB_1)

12.3.14 Interface with CPU

16-Bit Register: TCNT and GR are 16-bit registers. Reading/writing in a 16-bit unit is enabled but disabled in an 8-bit unit since the data bus with the CPU is 16-bit width. These registers must always be accessed in a 16-bit unit. Figure 12.5 shows an example of accessing the 16-bit registers.

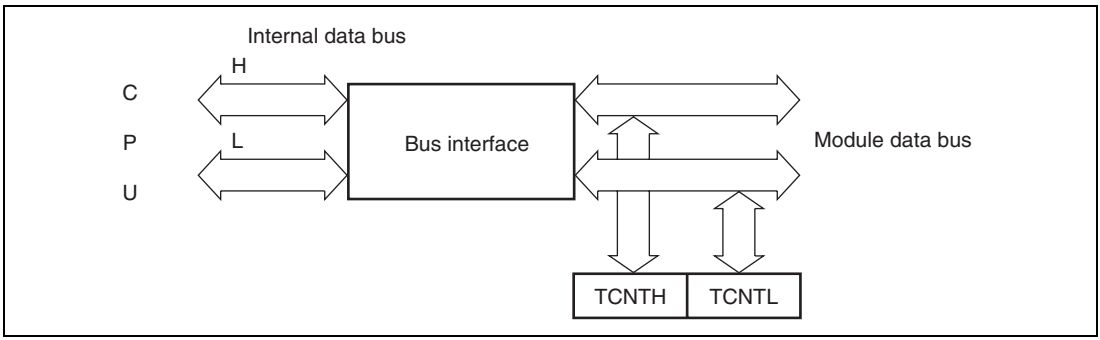


Figure 12.5 Accessing Operation of 16-Bit Register (between CPU and TCNT (16 Bits))

8-Bit Register: Registers other than TCNT and GR are 8-bit registers that are connected internally with the CPU in an 8-bit width. Figure 12.6 shows an example of accessing the 8-bit registers.

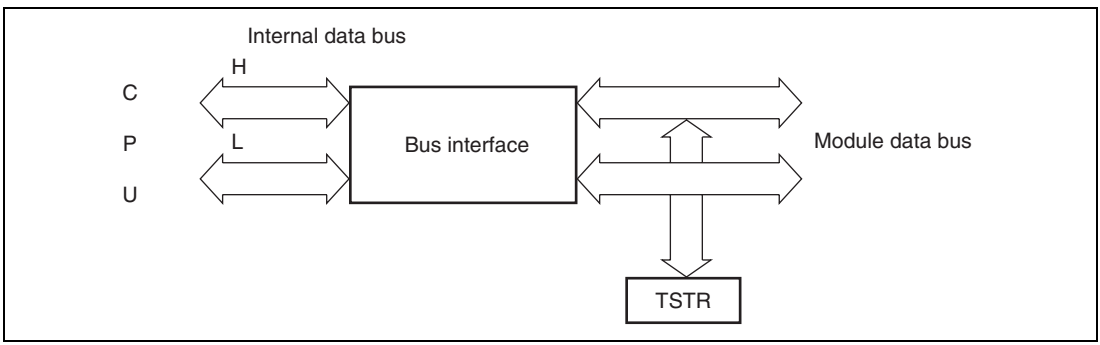


Figure 12.6 Accessing Operation of 8-Bit Register (between CPU and TSTR (8 Bits))

12.4.8 Buffer Operation

Buffer operation differs depending on whether GR has been designated for an input capture register or an output compare register, or in reset synchronous PWM mode or complementary PWM mode.

Table 12.8 shows the register combinations used in buffer operation.

Table 12.8 Register Combinations in Buffer Operation

General Register	Buffer Register
GRA	GRC
GRB	GRD

When GR is Output Compare Register: When a compare match occurs, the value in the buffer register of the corresponding channel is transferred to the general register.

This operation is illustrated in figure 12.35.

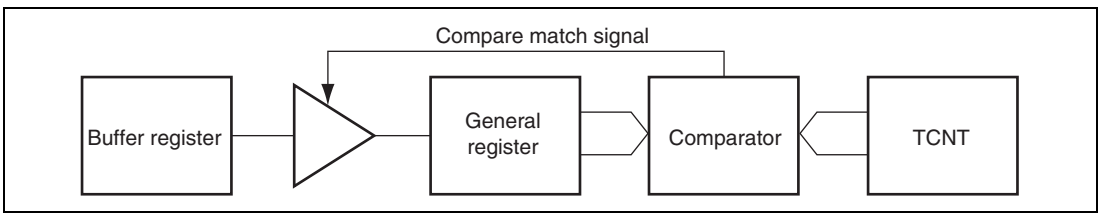


Figure 12.35 Compare Match Buffer Operation

When GR is Input Capture Register: When an input capture occurs, the value in TCNT is transferred to the general register and the value previously stored in the general register is transferred to the buffer register.

This operation is illustrated in figure 12.36.

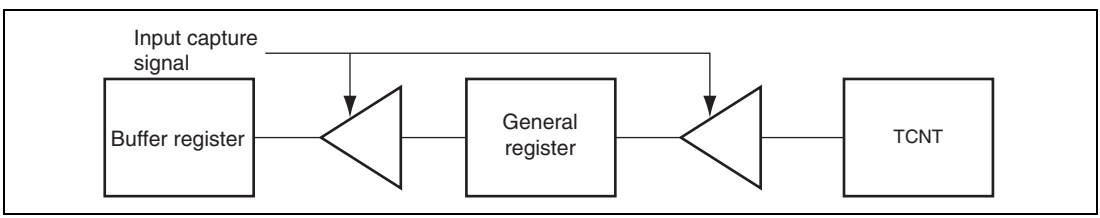


Figure 12.36 Input Capture Buffer Operation

13.3 Operation

The watchdog timer is provided with an 8-bit counter. If 1 is written to WDON while writing 0 to B2WI when the TCSRWE bit in TCSRWD is set to 1, TCWD begins counting up. (To operate the watchdog timer, two write accesses to TCSRWD are required.) When a clock pulse is input after the TCWD count value has reached H'FF, the watchdog timer overflows and an internal reset signal is generated. The internal reset signal is output for a period of 256 ϕ_{osc} clock cycles. TCWD is a writable counter, and when a value is set in TCWD, the count-up starts from that value. An overflow period in the range of 1 to 256 input clock cycles can therefore be set, according to the TCWD set value.

Figure 13.2 shows an example of watchdog timer operation.

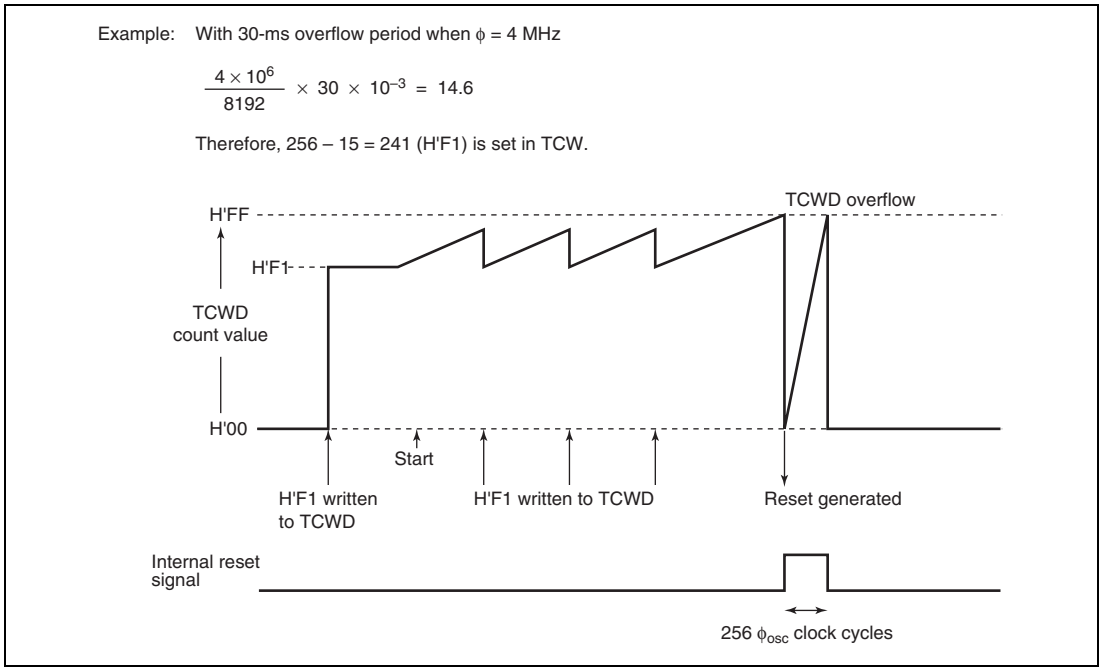


Figure 13.2 Watchdog Timer Operation Example

14.4 Operation in Asynchronous Mode

Figure 14.2 shows the general format for asynchronous serial communication. One character (or frame) consists of a start bit (low level), followed by data (in LSB-first order), a parity bit (high or low level), and finally stop bits (high level). Inside the SCI3, the transmitter and receiver are independent units, enabling full-duplex. Both the transmitter and the receiver also have a double-buffered structure, so data can be read or written during transmission or reception, enabling continuous data transfer.

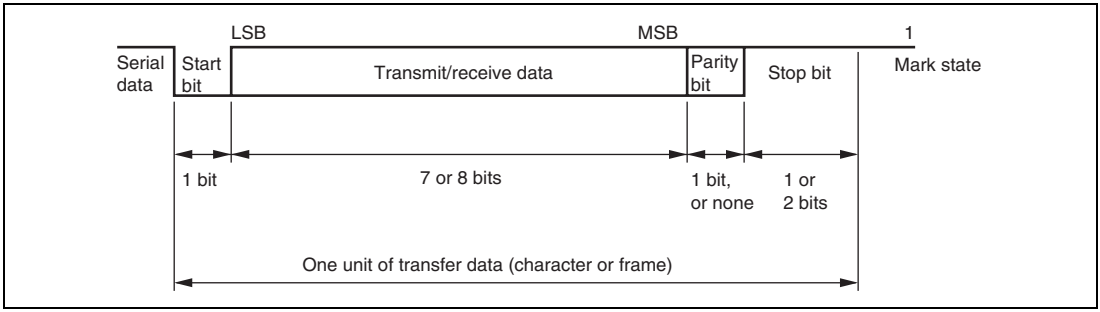


Figure 14.2 Data Format in Asynchronous Communication

14.4.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK3 pin can be selected as the SCI3's serial clock, according to the setting of the COM bit in SMR and the CKE0 and CKE1 bits in SCR3. When an external clock is input at the SCK3 pin, the clock frequency should be 14 times the bit rate used.

When the SCI3 is operated on an internal clock, the clock can be output from the SCK3 pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 14.3.

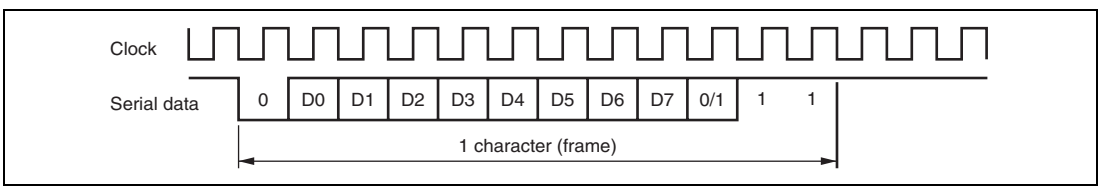


Figure 14.3 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode) (Example with 8-Bit Data, Parity, Two Stop Bits)

generated at this time. All other SCI3 operations are the same as those in asynchronous mode. Figure 14.18 shows an example of SCI3 operation for multiprocessor format reception.

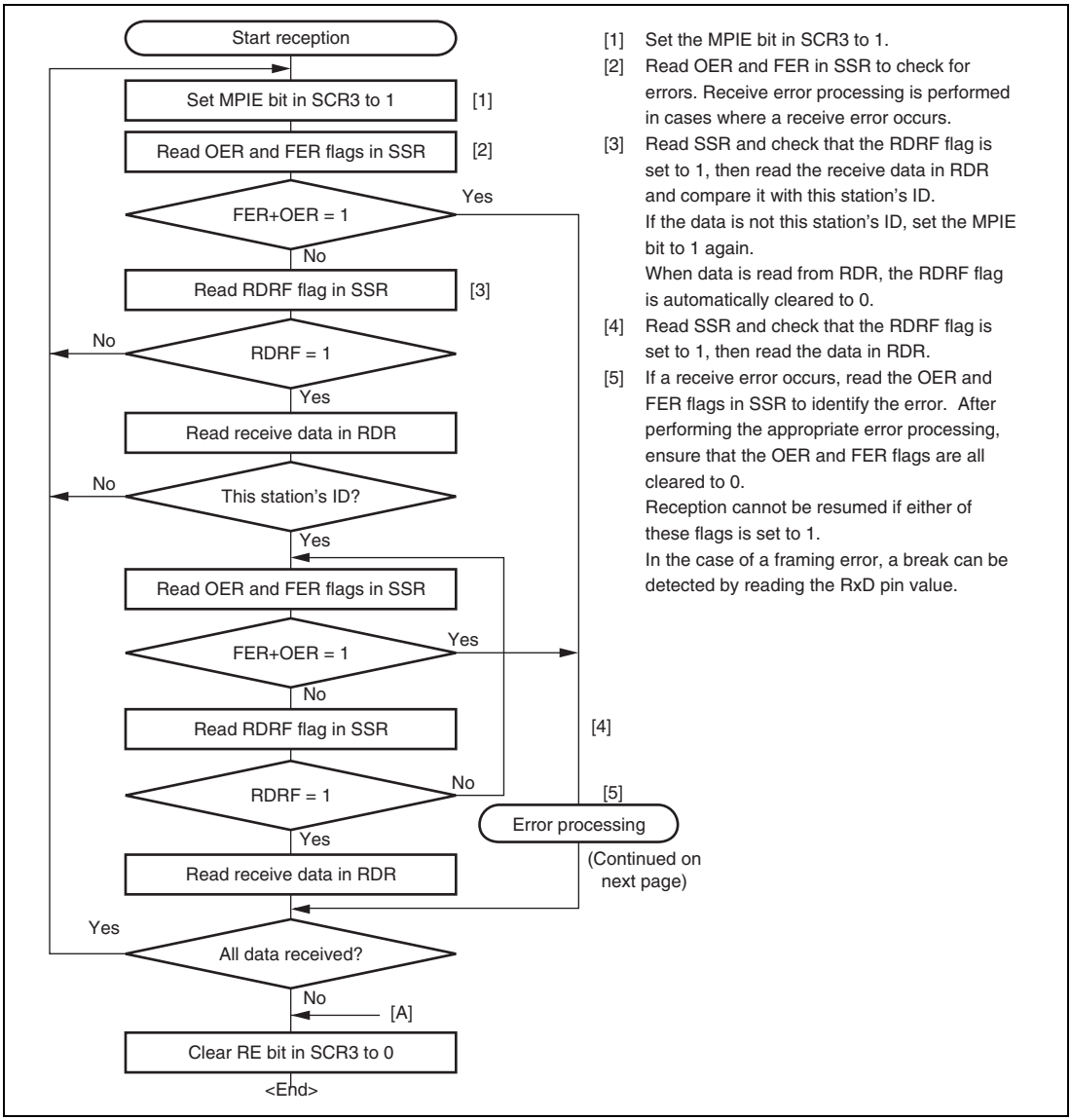


Figure 14.17 Sample Multiprocessor Serial Reception Flowchart (1)

Time Quantum (TQ) is an integer multiple of the number of system clocks, and is determined by the baud rate prescaler (BRP) as follows. ϕ means the system clock frequency.

$$TQ = (BRP + 1)/\phi$$

The following formula is used to calculate the 1-bit time and bit rate.

$$\text{1-bit time} = TQ \times \{1 + (1 + TSG1) + (1 + TSG2)\}$$

$$\text{Bit rate} = 1/\text{Bit time}$$
$$= \phi/\{(BRP + 1) \times \{1 + (1 + TSG1) + (1 + TSG2)\}\}$$

Values that can be set for TSG1 and TSG2 in BCR1 are listed in table 15.3.

Table 15.3 Settable Values for TSG1 and TSG2 in BCR1

		TSG2						
		001	010	011	100	101	110	111
TSG1	0011	No	Yes	No	No	No	No	No
	0100	Yes	Yes	Yes	No	No	No	No
	0101	Yes	Yes	Yes	Yes	No	No	No
	0110	Yes	Yes	Yes	Yes	Yes	No	No
	0111	Yes	Yes	Yes	Yes	Yes	Yes	No
	1000	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1001	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1010	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1011	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1100	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1101	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1110	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1111	Yes	Yes	Yes	Yes	Yes	Yes	Yes

[Legend] Yes: Setting is possible
No: Setting is prohibited

[Example]

To have a baud rate of 1 Mbps with ϕ = 16 MHz, BRP = 1, and (1 + TSG1) + (1 + TSG2) = 7. In this case, the settings are BCR1 = H′23 and BCR0 = H'01.

15.5.4 Message Reception

Figure 15.14 shows a message reception flowchart. Figure 15.15 shows the set timing for TXPR and TXCR during reception. A transmit request can be canceled by TXCR at any time during reception.

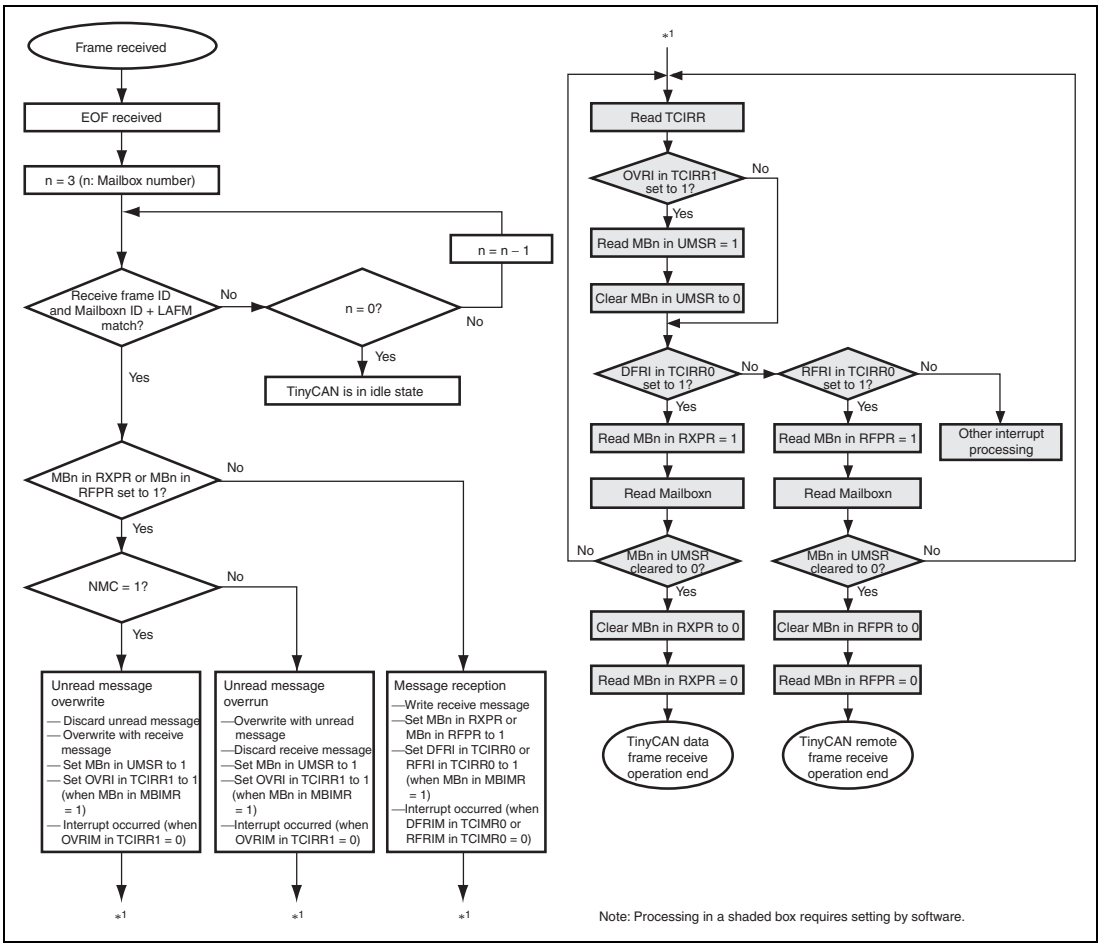


Figure 15.14 Message Reception Flowchart

16.4 Operation

16.4.1 Transfer Clock

Transfer clock can be selected from seven internal clocks and an external clock. When this module is used, the SSCK pin must be selected as a serial clock by setting the SCKS bit in SSCRH to 1. When the MSS bit in SSCRH is 1, an internal clock is selected and the SSCK pin is in the output state. If transfer is started, the SSCK pin outputs clocks of the transfer rate set in the CKS2 to CKS0 bits in SSMR. When the MSS bit is 0, an external clock is selected and the SSCK pin is in the input state.

16.4.2 Relationship between Clock Polarity and Phase, and Data

Relationship between clock polarity and phase, and transfer data changes according to a combination of the SSUMS bit in SSCRL and the CPOS and CPHS bits in SSMR. Figure 16.2 shows the relationship.

MSB-first transfer or LSB first transfer can be selected by the setting of the MLS bit in SSMR. When the MLS bit is 0, transfer is started from LSB to MSB. When the MLS bit is 1, transfer is started from MSB to LSB.

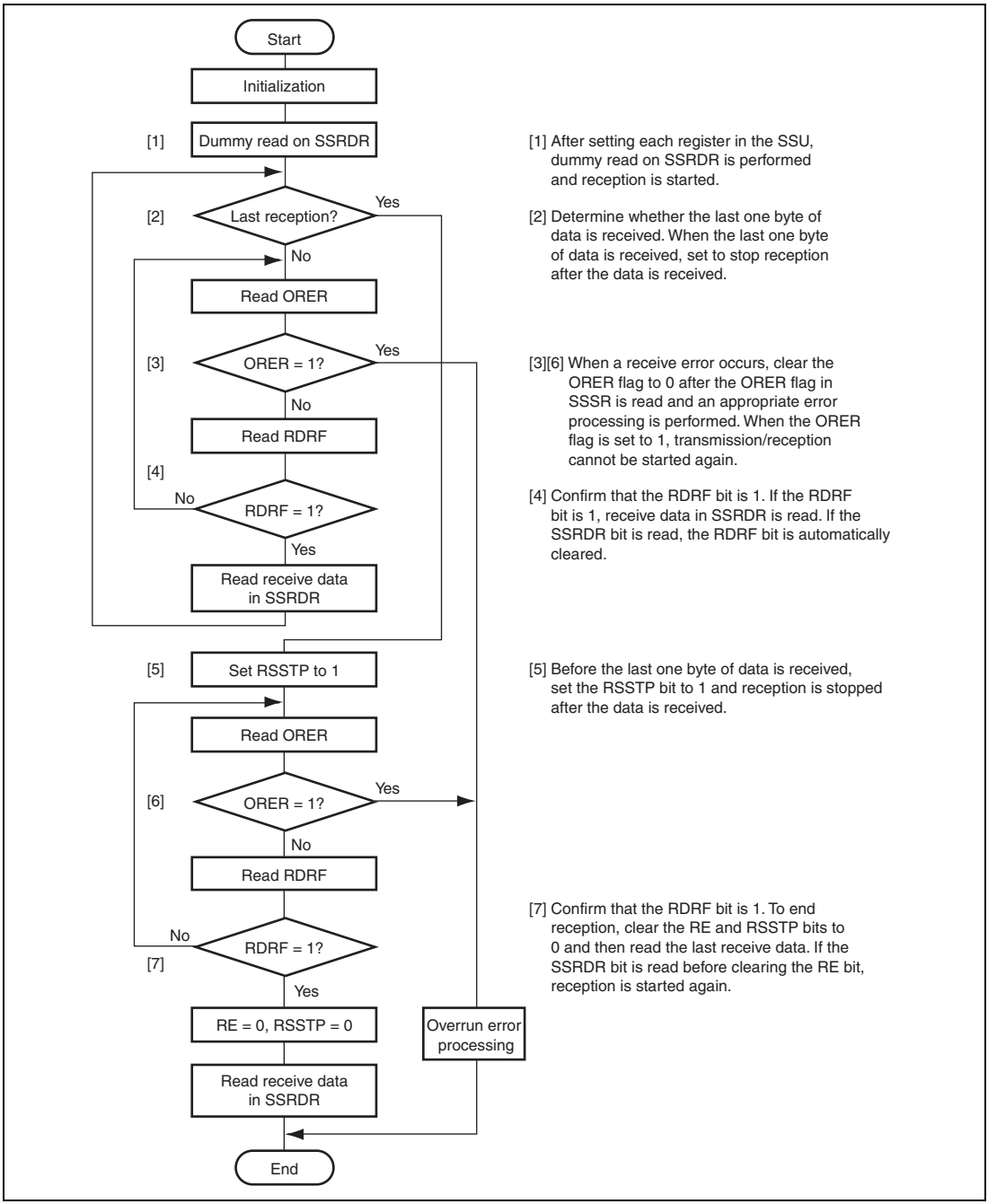


Figure 16.8 Sample Serial Reception Flowchart (MSS = 1)

Table 22.17 Synchronous Communication Unit (SSU) Timing

V_{CC} = 2.7 to 5.5 V, V_{SS} = 0.0 V, T_a = –20 to +75°C (regular specifications) or T_a = –40 to +85°C (wide-range specifications), C_L = 100 pF, unless otherwise indicated.

Item		Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
					Min.	Typ.	Max.		
Clock cycle		t_{SUCYC}	SSCK		4	—	—	t_{CYC}	Figures 22.7 to 22.11
Clock high pulse width		t_{HI}	SSCK		0.4	—	0.6	t_{SUCYC}	
Clock low pulse width		t_{LO}	SSCK		0.4	—	0.6	t_{SUCYC}	
Clock rise time	Master	t_{RISE}	SSCK		—	—	1	t_{CYC}	
	Slave				—	—	1.0	µs	
Clock fall time	Master	t_{FALL}	SSCK		—	—	1	t_{CYC}	
	Slave				—	—	1.0	µs	
Data input setup time		t_{SU}	SSO, SSI		1	—	—	t_{CYC}	
Data input hold time		t_H	SSO, SSI		1	—	—	t_{CYC}	
$\overline{SCS}$ setup time	Slave	t_{LEAD}	$\overline{SCS}$		1 t_{CYC} + 100	—	—	ns	
$\overline{SCS}$ hold time	Slave	t_{LAG}	$\overline{SCS}$		1 t_{CYC} + 100	—	—	ns	
Data output delay time		t_{OD}	SSO, SSI		—	—	1	t_{CYC}	
Slave access time		t_{SA}	SSI		—	—	1 t_{CYC} + 100	ns	
Slave out release time		t_{OR}	SSI		—	—	1 t_{CYC} + 100	ns	