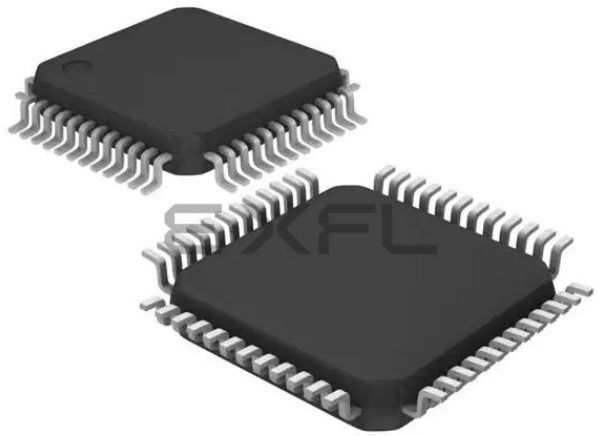


Welcome to E-XFL.COM

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	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	48-LQFP
Supplier Device Package	48-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/hd64f3664fxv

Table 2.7 Branch Instructions

Instruction	Size	Function		
Bcc*	—	Branches to a specified address if a specified condition is true. The branching conditions are listed below.		
		Mnemonic	**Description**	**Condition**
		BRA (BT)	Always (true)	Always
		BRN (BF)	Never (false)	Never
		BHI	High	$C \vee Z = 0$
		BLS	Low or same	$C \vee Z = 1$
		BCC (BHS)	Carry clear (high or same)	$C = 0$
		BCS (BLO)	Carry set (low)	$C = 1$
		BNE	Not equal	$Z = 0$
		BEQ	Equal	$Z = 1$
		BVC	Overflow clear	$V = 0$
		BVS	Overflow set	$V = 1$
		BPL	Plus	$N = 0$
		BMI	Minus	$N = 1$
		BGE	Greater or equal	$N \oplus V = 0$
		BLT	Less than	$N \oplus V = 1$
		BGT	Greater than	$Z \vee (N \oplus V) = 0$
		BLE	Less or equal	$Z \vee (N \oplus V) = 1$
JMP	—	Branches unconditionally to a specified address.		
BSR	—	Branches to a subroutine at a specified address.		
JSR	—	Branches to a subroutine at a specified address.		
RTS	—	Returns from a subroutine		

Note: * Bcc is the general name for conditional branch instructions.

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)

Example
Direct transition time = (2 + 1) × 8 tw + (8192 + 14) × tosc = 24tw + 8206 tosc
(when the CPU operating clock of $\phi_w/8 \rightarrow \phi_{osc}$ and a waiting time of 8192 states are selected)

[Legend]
tosc: OSC clock cycle time
tw: watch clock cycle time
tcyc: system clock (ϕ) cycle time
tsubcyc: subclock (ϕ_{SUB}) cycle time

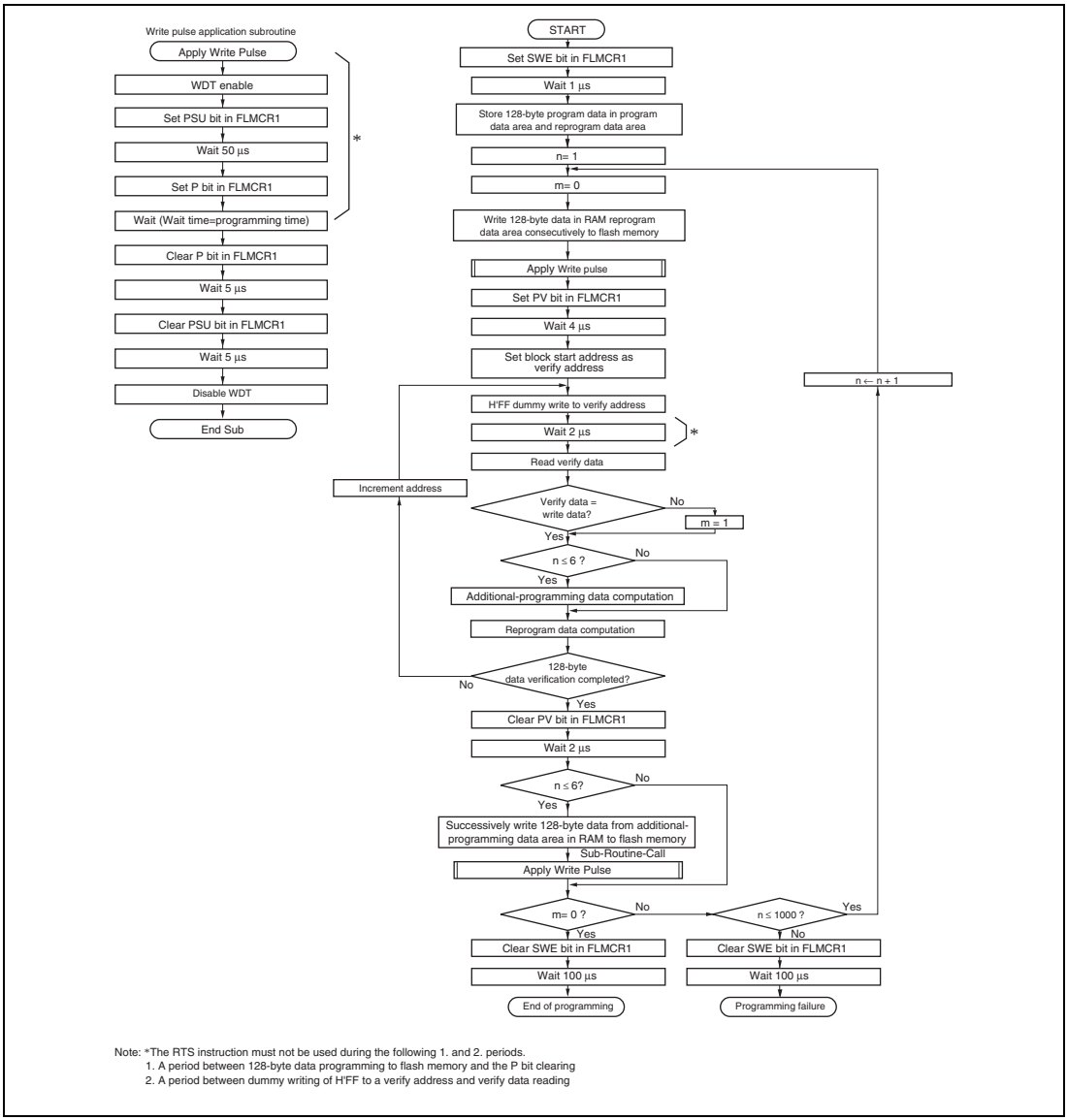
6.5 Module Standby Function

The module-standby function can be set to any peripheral module. In module standby mode, the clock supply to modules stops to enter the power-down mode. Module standby mode enables each on-chip peripheral module to enter the standby state by setting a bit that corresponds to each module to 1 and cancels the mode by clearing the bit to 0.

6.6 Usage Note

When subsleep mode is entered by setting the SMSEL bit to 1 while the subclock is not used (the X_1 pin is fixed), note that active mode cannot be re-entered by using an interrupt. To use a power-down mode while a port is retained, connect the subclock to the X_1 and X_2 pins.

8. The maximum number of repetitions of the program/program-verify sequence of the same bit is 1,000.



Note: *The RTS instruction must not be used during the following 1. and 2. periods.
1. A period between 128-byte data programming to flash memory and the P bit clearing
2. A period between dummy writing of H'FF to a verify address and verify data reading

Figure 7.3 Program/Program-Verify Flowchart

Bit	Bit Name	Initial Value	R/W	Description
0	TMOW	0	R/W	P10/TMOW Pin Function Switch This bit selects whether pin P10/TMOW is used as P10 or as TMOW. 0: General I/O port 1: TMOW output pin

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. Bit 3 is a reserved bit.
6	PCR16	0	W	
5	PCR15	0	W	
4	PCR14	0	W	
3	—	—	—	
2	PCR12	0	W	
1	PCR11	0	W	
0	PCR10	0	W	

- P14/$\overline{\text{IRQ0}}$ Pin

Register	PMR1	PCR1	
Bit Name	**IRQ0**	**PCR14**	**Pin Function**
Setting value	0	0	P14 input pin
		1	P14 output pin
	1	X	$\overline{\text{IRQ0}}$ input pin

[Legend]
X: Don't care.

- P12 Pin

Register	PCR1	
Bit Name	**PCR12**	**Pin Function**
Setting value	0	P12 input pin
	1	P12 output pin

- P11 Pin

Register	PCR1	
Bit Name	**PCR11**	**Pin Function**
Setting value	0	P11 input pin
	1	P11 output pin

- P10/TMOW Pin

Register	PMR1	PCR1	
Bit Name	**TMOW**	**PCR10**	**Pin Function**
Setting value	0	0	P10 input pin
		1	P10 output pin
	1	X	TMOW output pin

[Legend]
X: Don't care.

- P21/RXD Pin

Register	SCR3	PCR2	
Bit Name	**RE**	**PCR21**	**Pin Function**
Setting Value	0	0	P21 input pin
		1	P21 output pin
	1	X	RXD input pin

[Legend]
X: Don't care.

- P20/SCK3 Pin

Register	SCR3		SMR	PCR2	
Bit Name	**CKE1**	**CKE0**	**COM**	**PCR20**	**Pin Function**
Setting Value	0	0	0	0	P20 input pin
				1	P20 output pin
	0	0	1	X	SCK3 output pin
	0	1	X	X	SCK3 output pin
	1	X	X	X	SCK3 input pin

[Legend]
X: Don't care.

Bit	Bit Name	Initial Value	R/W	Description
2	TMA2	0	R/W	Internal Clock Select 2 to 0
1	TMA1	0	R/W	These bits select the clock input to TCA when TMA3 = 0.
0	TMA0	0	R/W	000: ϕ/8192
				001: ϕ/4096
				010: ϕ/2048
				011: ϕ/512
				100: ϕ/256
				101: ϕ/128
				110: ϕ/32
				111: ϕ/8
				These bits select the overflow period when TMA3 = 1 (when a 32.768 kHz crystal oscillator with is used as ϕW).
				000: 1s
				001: 0.5 s
				010: 0.25 s
				011: 0.03125 s
				1XX: Both PSW and TCA are reset

[Legend]

X: Don't care.

10.3.2 Timer Counter A (TCA)

TCA is an 8-bit readable up-counter, which is incremented by internal clock input. The clock source for input to this counter is selected by bits TMA3 to TMA0 in TMA. TCA values can be read by the CPU in active mode, but cannot be read in subactive mode. When TCA overflows, the IRRTA bit in interrupt request register 1 (IRR1) is set to 1. TCA is cleared by setting bits TMA3 and TMA2 in TMA to B’11. TCA is initialized to H'00.

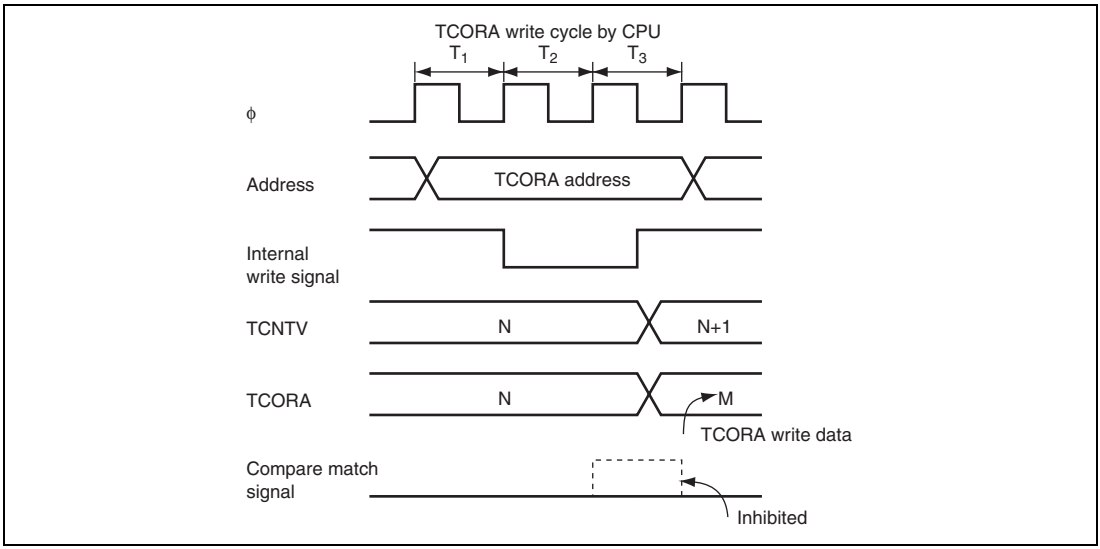


Figure 11.12 Contention between TCORA Write and Compare Match

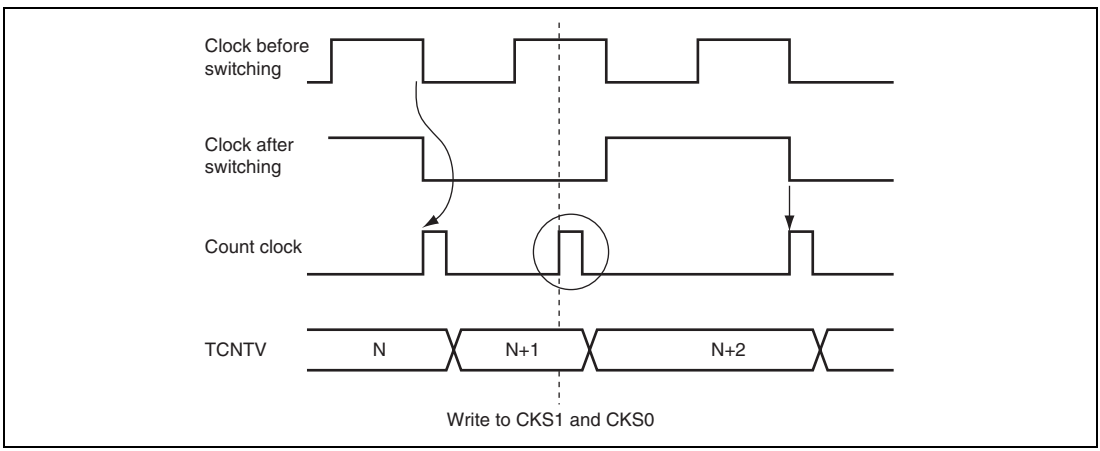


Figure 11.13 Internal Clock Switching and TCNTV Operation

Bit	Bit Name	Initial Value	R/W	Description
5	IOB1	0	R/W	I/O Control B1 and B0
4	IOB0	0	R/W	When IOB2 = 0, 00: No output at compare match 01: 0 output to the FTIOB pin at GRB compare match 10: 1 output to the FTIOB pin at GRB compare match 11: Output toggles to the FTIOB pin at GRB compare match When IOB2 = 1, 00: Input capture at rising edge at the FTIOB pin 01: Input capture at falling edge at the FTIOB pin 1X: Input capture at rising and falling edges of the FTIOB pin
3	—	1	—	Reserved This bit is always read as 1.
2	IOA2	0	R/W	I/O Control A2 Selects the GRA function. 0: GRA functions as an output compare register 1: GRA functions as an input capture register
1	IOA1	0	R/W	I/O Control A1 and A0
0	IOA0	0	R/W	When IOA2 = 0, 00: No output at compare match 01: 0 output to the FTIOA pin at GRA compare match 10: 1 output to the FTIOA pin at GRA compare match 11: Output toggles to the FTIOA pin at GRA compare match When IOA2 = 1, 00: Input capture at rising edge of the FTIOA pin 01: Input capture at falling edge of the FTIOA pin 1X: Input capture at rising and falling edges of the FTIOA pin

[Legend]

X: Don't care.

12.3.8 General Registers A to D (GRA to GRD)

Each general register is a 16-bit readable/writable register that can function as either an output-compare register or an input-capture register. The function is selected by settings in TIOR0 and TIOR1.

When a general register is used as an input-compare register, its value is constantly compared with the TCNT value. When the two values match (a compare match), the corresponding flag (IMFA, IMFB, IMFC, or IMFD) in TSRW is set to 1. An interrupt request is generated at this time, when IMIEA, IMIEB, IMIEC, or IMIED is set to 1. Compare match output can be selected in TIOR.

When a general register is used as an input-capture register, an external input-capture signal is detected and the current TCNT value is stored in the general register. The corresponding flag (IMFA, IMFB, IMFC, or IMFD) in TSRW is set to 1. If the corresponding interrupt-enable bit (IMIEA, IMIEB, IMIEC, or IMIED) in TSRW is set to 1 at this time, an interrupt request is generated. The edge of the input-capture signal is selected in TIOR.

GRC and GRD can be used as buffer registers of GRA and GRB, respectively, by setting BUFEA and BUFEB in TMRW.

For example, when GRA is set as an output-compare register and GRC is set as the buffer register for GRA, the value in the buffer register GRC is sent to GRA whenever compare match A is generated.

When GRA is set as an input-capture register and GRC is set as the buffer register for GRA, the value in TCNT is transferred to GRA and the value in the buffer register GRC is transferred to GRA whenever an input capture is generated.

GRA to GRD must be written or read in 16-bit units; 8-bit access is not allowed. GRA to GRD are initialized to H'FFFF by a reset.

14.3 Register Descriptions

The SCI3 has the following registers.

- Receive shift register (RSR)
- Receive data register (RDR)
- Transmit shift register (TSR)
- Transmit data register (TDR)
- Serial mode register (SMR)
- Serial control register 3 (SCR3)
- Serial status register (SSR)
- Bit rate register (BRR)

14.3.1 Receive Shift Register (RSR)

RSR is a shift register that is used to receive serial data input from the RxD pin and convert it into parallel data. When one byte of data has been received, it is transferred to RDR automatically. RSR cannot be directly accessed by the CPU.

14.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores received data. When the SCI3 has received one byte of serial data, it transfers the received serial data from RSR to RDR, where it is stored. After this, RSR is receive-enabled. As RSR and RDR function as a double buffer in this way, continuous receive operations are possible. After confirming that the RDRF bit in SSR is set to 1, read RDR only once. RDR cannot be written to by the CPU. RDR is initialized to H'00.

14.3.3 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI3 first transfers transmit data from TDR to TSR automatically, then sends the data that starts from the LSB to the TXD pin. TSR cannot be directly accessed by the CPU.

Table 14.5 shows the states of the SSR status flags and receive data handling when a receive error is detected. If a receive error is detected, the RDRF flag retains its state before receiving data. Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the OER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 14.8 shows a sample flowchart for serial data reception.

Table 14.5 SSR Status Flags and Receive Data Handling

	SSR Status Flag				
RDRF*	**OER**	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	1	0	0	Lost	Overrun error
0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overrun error + framing error
1	1	0	1	Lost	Overrun error + parity error
0	0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	1	Lost	Overrun error + framing error + parity error

Note: * The RDRF flag retains the state it had before data reception.

Section 15 I²C Bus Interface (IIC)

The I²C bus interface conforms to and provides a subset of the Philips I²C bus (inter-IC bus) interface functions. The register configuration that controls the I²C bus differs partly from the Philips configuration, however.

15.1 Features

- Selection of I²C format or clocked synchronous serial format
  - I²C bus format: addressing format with acknowledge bit, for master/slave operation
  - Clocked synchronous serial format: non-addressing format without acknowledge bit, for master operation only
- I²C bus format
- Two ways of setting slave address
- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Wait function in master mode

  A wait can be inserted by driving the SCL pin low after data transfer, excluding acknowledgement. The wait can be cleared by clearing the interrupt flag.
- Wait function in slave mode

  A wait request can be generated by driving the SCL pin low after data transfer, excluding acknowledgement. The wait request is cleared when the next transfer becomes possible.
- Three interrupt sources
  - Data transfer end (including transmission mode transition with I²C bus format and address reception after loss of master arbitration)
  - Address match: when any slave address matches or the general call address is received in slave receive mode
  - Stop condition detection
- Selection of 16 internal clocks (in master mode)
- Direct bus drive
  - Two pins, SCL and SDA pins function as NMOS open-drain outputs when the bus drive function is selected.

17.4.7 Write Operations

There are two types write operations; byte write operation and page write operation. To initiate the write operation, input 0 to R/$\overline{W}$ code following the slave address.

1. Byte Write

   A write operation requires an 8-bit data of a 7-bit slave address with R/$\overline{W}$ code = "0". Then the EEPROM sends acknowledgement "0" at the ninth bit. This enters the write mode. Then, two bytes of the memory address are received from the MSB side in the order of upper and lower. Upon receipt of one-byte memory address, the EEPROM sends acknowledgement "0" and receives a following a one-byte write data. After receipt of write data, the EEPROM sends acknowledgement "0". If the EEPROM receives a stop condition, the EEPROM enters an internally controlled write cycle and terminates receipt of SCL and SDA inputs until completion of the write cycle. The EEPROM returns to a standby mode after completion of the write cycle.

   The byte write operation is shown in figure 17.3.

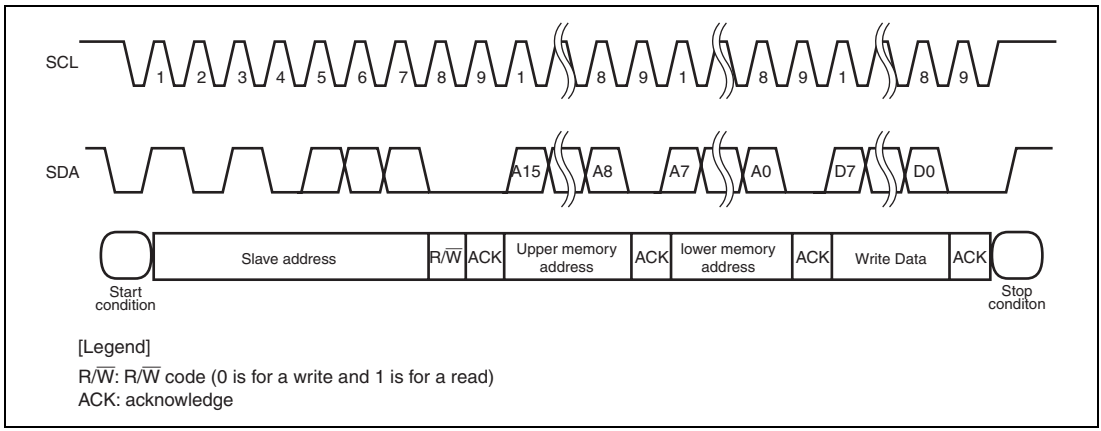


Figure 17.3 Byte Write Operation

2. Page Write

   This LSI is capable of the page write operation which allows any number of bytes up to 8 bytes to be written in a single write cycle. The write data is input in the same sequence as the byte write in the order of a start condition, slave address + R/$\overline{W}$ code, memory address (n), and write data (Dn) with every ninth bit acknowledgement "0" output. The EEPROM enters the page write operation if the EEPROM receives more write data (Dn+1) is input instead of receiving a stop condition after receiving the write data (Dn). LSB 3 bits (A2 to A0) in the EEPROM address are automatically incremented to be the (n+1) address upon receiving write data (Dn+1). Thus the write data can be received sequentially.

Section 19 List of Registers

The register list gives information on the on-chip I/O register addresses, how the register bits are configured, and the register states in each operating mode. The information is given as shown below.

1. Register addresses (address order)
- Registers are listed from the lower allocation addresses.
- Registers are classified by functional modules.
- The data bus width is indicated.
- The number of access states is indicated.

2. Register bits
- Bit configurations of the registers are described in the same order as the register addresses.
- Reserved bits are indicated by — in the bit name column.
- When registers consist of 16 bits, bits are described from the MSB side.

3. Register states in each operating mode
- Register states are described in the same order as the register addresses.
- The register states described here are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, refer to the section on that on-chip peripheral module.

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Reference Figure
Conversion time (single mode)			AV_{CC} = 4.0 V to 5.5 V	134	—	—	t_{cyc}	
Nonlinearity error				—	—	±3.5	LSB	
Offset error				—	—	±3.5	LSB	
Full-scale error				—	—	±3.5	LSB	
Quantization error				—	—	±0.5	LSB	
Absolute accuracy				—	—	±4.0	LSB	

Notes: 1. Set $AV_{CC} = V_{CC}$ when the A/D converter is not used.
2. AI_{STOP1} is the current in active and sleep modes while the A/D converter is idle.
3. AI_{STOP2} is the current at reset and in standby, subactive, and subsleep modes while the A/D converter is idle.

20.3.5 Watchdog Timer Characteristics

Table 20.15 Watchdog Timer Characteristics

V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Reference Figure
On-chip oscillator overflow time	t_{OVF}			0.2	0.4	—	s	*

Note: * Shows the time to count from 0 to 255, at which point an internal reset is generated, when the internal oscillator is selected.

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					