

Welcome to [E-XFL.COM](https://www.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	78K/0
Core Size	8-Bit
Speed	10MHz
Connectivity	LINbus, UART/USART
Peripherals	LCD, LVD, POR, PWM, WDT
Number of I/O	30
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f0400ga-gam-ax

13.4.4 Calculation of baud rate	331
CHAPTER 14 SERIAL INTERFACE UART6.....	335
14.1 Functions of Serial Interface UART6.....	335
14.2 Configuration of Serial Interface UART6	339
14.3 Registers Controlling Serial Interface UART6.....	342
14.4 Operation of Serial Interface UART6.....	353
14.4.1 Operation stop mode	353
14.4.2 Asynchronous serial interface (UART) mode.....	354
14.4.3 Dedicated baud rate generator	368
14.4.4 Calculation of baud rate	370
CHAPTER 15 LCD CONTROLLER/DRIVER.....	376
15.1 Functions of LCD Controller/Driver	376
15.2 Configuration of LCD Controller/Driver	378
15.3 Registers Controlling LCD Controller/Driver	380
15.4 Setting LCD Controller/Driver	385
15.5 LCD Display Data Memory	386
15.6 Common and Segment Signals	387
15.7 Display Modes	393
15.7.1 Static display example	393
15.7.2 Two-time-slice display example	396
15.7.3 Three-time-slice display example.....	399
15.7.4 Four-time-slice display example	403
15.8 Supplying LCD Drive Voltages V_{LC0} , V_{LC1} , V_{LC2} and V_{LC3}	406
15.8.1 Internal resistance division method.....	406
15.8.2 External resistance division method.....	408
CHAPTER 16 MANCHESTER CODE GENERATOR.....	410
16.1 Functions of Manchester Code Generator	410
16.2 Configuration of Manchester Code Generator.....	410
16.3 Registers Controlling Manchester Code Generator	413
16.4 Operation of Manchester Code Generator	416
16.4.1 Operation stop mode	416
16.4.2 Manchester code generator mode	417
16.4.3 Bit sequential buffer mode	426
CHAPTER 17 INTERRUPT FUNCTIONS	435
17.1 Interrupt Function Types.....	435
17.2 Interrupt Sources and Configuration	435
17.3 Registers Controlling Interrupt Functions	440
17.4 Interrupt Servicing Operations	447
17.4.1 Maskable interrupt acknowledgment.....	447
17.4.2 Software interrupt request acknowledgment.....	449
17.4.3 Multiple interrupt servicing	450
17.4.4 Interrupt request hold.....	453

3.3.2 Immediate addressing

[Function]

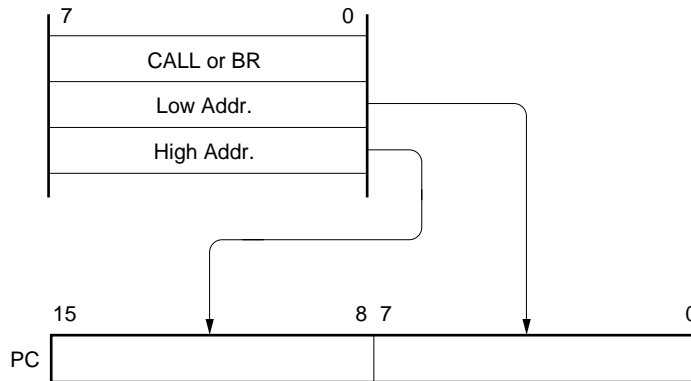
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed.

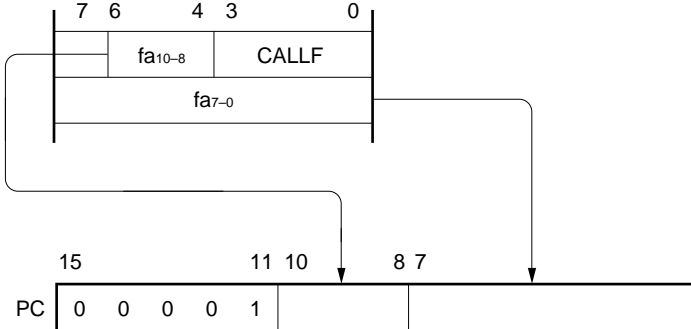
CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

In the case of CALL !addr16 and BR !addr16 instructions



In the case of CALLF !addr11 instruction



(7) A/D port configuration register 0 (ADPC0) (μ PD78F041x only)

This register switches the P20/ANI0 to P25/ANI5 pins to analog input of A/D converter or digital I/O of port.

ADPC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 08H.

Caution Set the values shown in Figure 4-22 after the reset is released.

Figure 4-22. Format of A/D Port Configuration Register 0 (ADPC0)

Address: FF8FH After reset: 08H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC0	0	0	0	0	0	ADPC02	ADPC01	ADPC00

ADPC02	ADPC01	ADPC00	Digital I/O (D)/analog input (A) switching					
			P25 /ANI5	P24 /ANI4	P23 /ANI3	P22 /AN2	P21 /ANI1	P20 /ANI0
0	0	0	A	A	A	A	A	A
0	0	1	A	A	A	A	A	D
0	1	0	A	A	A	A	D	D
0	1	1	A	A	A	D	D	D
1	0	0	A	A	D	D	D	D
1	0	1	A	D	D	D	D	D
1	1	0	D	D	D	D	D	D
Other than above			Setting prohibited					

- Cautions**
1. Set the channel used for A/D conversion to the input mode by using port mode register 2 (PM2).
 2. The pin to be set as a digital I/O via ADPC, must not be set via ADS, ADDS1 or ADDS0.
 3. If data is written to ADPC0, a wait cycle is generated. Do not write data to ADPC0 when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 29 CAUTIONS FOR WAIT.
 4. If pins ANI0/P20/SEG21 to ANI5/P25/SEG16 are set to segment output via the PF2 register, output is set to segment output, regardless of the ADPC0 setting (for μ PD78F041x only).

(4) Internal oscillation mode register (RCM)

This register sets the operation mode of internal oscillator.

RCM can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 80H^{Note 1}.

Figure 5-4. Format of Internal Oscillation Mode Register (RCM)

Address: FFA0H After reset: 80H^{Note 1} R/W^{Note 2}

Symbol	<7>	6	5	4	3	2	<1>	<0>
RCM	RSTS	0	0	0	0	0	LSRSTOP	RSTOP

RSTS	Status of internal high-speed oscillator
0	Waiting for accuracy stabilization of internal high-speed oscillator
1	Stability operating of internal high-speed oscillator

LSRSTOP	Internal low-speed oscillator oscillating/stopped
0	Internal low-speed oscillator oscillating
1	Internal low-speed oscillator stopped

RSTOP	Internal high-speed oscillator oscillating/stopped
0	Internal high-speed oscillator oscillating
1	Internal high-speed oscillator stopped

Notes 1. The value of this register is 00H immediately after a reset release but automatically changes to 80H after internal high-speed oscillator has been stabilized.

2. Bit 7 is read-only.

Caution When setting RSTOP to 1, be sure to confirm that the CPU operates with a clock other than the internal high-speed oscillation clock. Specifically, set under either of the following conditions.

- When MCS = 1 (when CPU operates with the high-speed system clock)
- When CLS = 1 (when CPU operates with the subsystem clock)

In addition, stop peripheral hardware that is operating on the internal high-speed oscillation clock before setting RSTOP to 1.

5.4.3 When subsystem clock is not used

If it is not necessary to use the subsystem clock for low power consumption operations, or if not using the subsystem clock as an I/O port, set the XT1 and XT2 pins to Input port mode (OSCSELS = 0) and independently connect to V_{DD} or V_{SS} via a resistor.

Remark OSCSELS: Bit 4 of clock operation mode select register (OSCCTL)

5.4.4 Internal high-speed oscillator

The internal high-speed oscillator is incorporated in the 78K0/LC3. Oscillation can be controlled by the internal oscillation mode register (RCM).

After a reset release, the internal high-speed oscillator automatically starts oscillation (8 MHz (TYP.)).

5.4.5 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the 78K0/LC3.

The internal low-speed oscillation clock is only used as the clock of the watchdog timer, 8-bit timer H1, and LCD controller/driver. The internal low-speed oscillation clock cannot be used as the CPU clock.

“Can be stopped by software” or “Cannot be stopped” can be selected by the option byte. When “Can be stopped by software” is set, oscillation can be controlled by the internal oscillation mode register (RCM).

After a reset release, the internal low-speed oscillator automatically starts oscillation, and the watchdog timer is driven (240 kHz (TYP.)) if the watchdog timer operation is enabled using the option byte.

5.4.6 Prescaler

The prescaler generates various clocks by dividing the main system clock when the main system clock is selected as the clock to be supplied to the CPU.

Table 5-5. CPU Clock Transition and SFR Register Setting Examples (3/4)

(6) CPU clock changing from high-speed system clock (C) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register	RSTOP	RSTS	MCM0
Status Transition			
(C) → (B)	0	Confirm this flag is 1.	0

Unnecessary if the CPU is operating with the internal high-speed oscillation clock

(7) CPU clock changing from high-speed system clock (C) to subsystem clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register	OSCSELS	Waiting for Oscillation Stabilization	CSS
Status Transition			
(C) → (D)	1	Necessary	1

Unnecessary if the CPU is operating with the subsystem clock

(8) CPU clock changing from subsystem clock (D) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register	RSTOP	RSTS	MCM0	CSS
Status Transition				
(D) → (B)	0	Confirm this flag is 1.	0	0

Unnecessary if the CPU is operating with the internal high-speed oscillation clock

Unnecessary if XSEL is 0

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figure 5-15.

- 2.** MCM0: Bit 0 of the main clock mode register (MCM)
 OSCSELS: Bit 4 of the clock operation mode select register (OSCCTL)
 RSTS, RSTOP: Bits 7 and 0 of the internal oscillation mode register (RCM)
 CSS: Bit 4 of the processor clock control register (PCC)

Table 5-5. CPU Clock Transition and SFR Register Setting Examples (4/4)

(9) CPU clock changing from subsystem clock (D) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register Status Transition	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL ^{Note}	MCM0	CSS
(D) → (C) (X1 clock)	0	1	0	Must be checked	1	1	0
(D) → (C) (external main clock)	1	1	0	Must not be checked	1	1	0

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the high-speed system clock

Note The value of this flag can be changed only once after a reset release. This setting is not necessary if it has already been set.

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 27 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)).

(10) • HALT mode (E) set while CPU is operating with internal high-speed oscillation clock (B)

- HALT mode (F) set while CPU is operating with high-speed system clock (C)
- HALT mode (G) set while CPU is operating with subsystem clock (D)

Status Transition	Setting
(B) → (E) (C) → (F) (D) → (G)	Executing HALT instruction

(11) • STOP mode (H) set while CPU is operating with internal high-speed oscillation clock (B)

- STOP mode (I) set while CPU is operating with high-speed system clock (C)

(Setting sequence) →

Status Transition	Setting	
(B) → (H) (C) → (I)	Stopping peripheral functions that cannot operate in STOP mode	Executing STOP instruction

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figure 5-15.

- 2.** EXCLK, OSCSEL: Bits 7 and 6 of the clock operation mode select register (OSCCTL)
MSTOP: Bit 7 of the main OSC control register (MOC)
XSEL, MCM0: Bits 2 and 0 of the main clock mode register (MCM)
CSS: Bit 4 of the processor clock control register (PCC)

8.3 Registers Controlling 8-Bit Timers H0, H1, and H2

The following four registers are used to control 8-bit timers H0, H1, and H2.

- 8-bit timer H mode register n (TMHMDn)
- 8-bit timer H carrier control register 1 (TMCYC1)^{Note}
- Port mode register 3 (PM3)
- Port register 3 (P3)

Note 8-bit timer H1 only

(1) 8-bit timer H mode register n (TMHMDn)

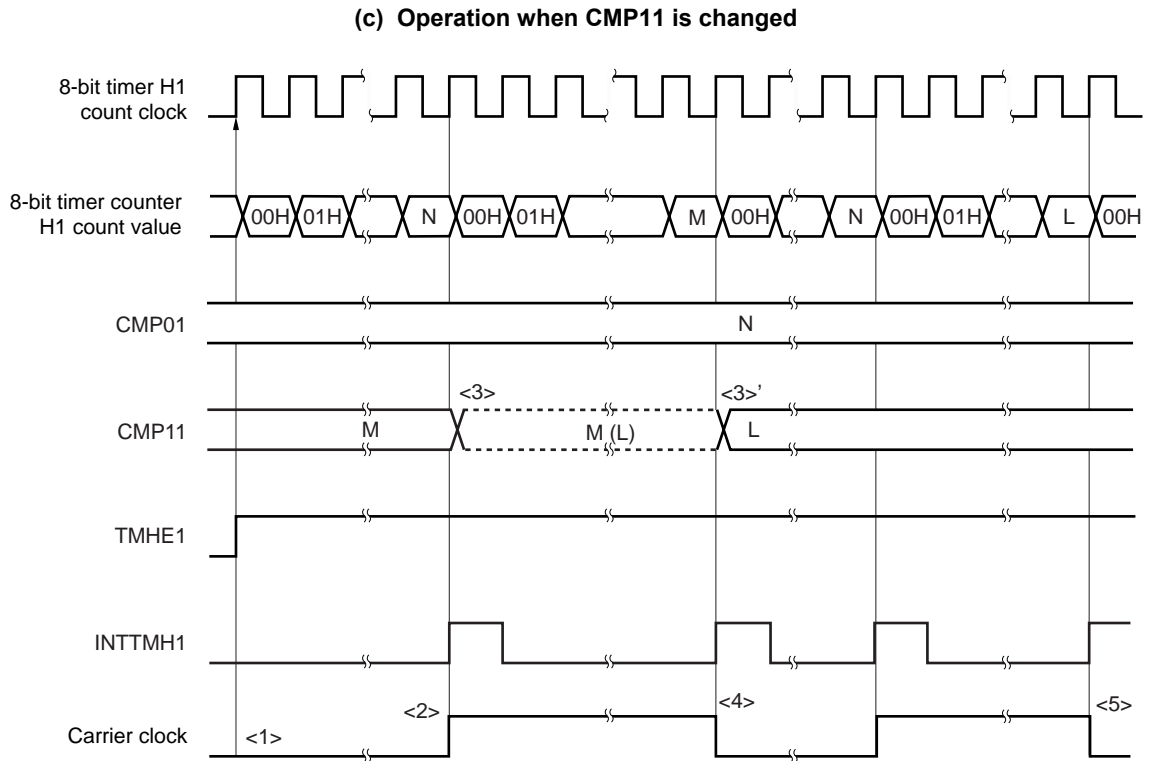
This register controls the mode of timer H.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Remark n = 0 to 2

Figure 8-17. Carrier Generator Mode Operation Timing (3/3)



- <1> When TMHE1 = 1 is set, the 8-bit timer H1 starts a count operation. At that time, the carrier clock remains default.
- <2> When the count value of the 8-bit timer counter H1 matches the value of the CMP01 register, the INTTMH1 signal is output, the carrier signal is inverted, and the timer counter is cleared to 00H. At the same time, the compare register whose value is to be compared with that of the 8-bit timer counter H1 is changed from the CMP01 register to the CMP11 register.
- <3> The CMP11 register is asynchronous to the count clock, and its value can be changed while the 8-bit timer H1 is operating. The new value (L) to which the value of the register is to be changed is latched. When the count value of the 8-bit timer counter H1 matches the value (M) of the CMP11 register before the change, the CMP11 register is changed (<3>'). However, it takes three count clocks or more since the value of the CMP11 register has been changed until the value is transferred to the register. Even if a match signal is generated before the duration of three count clocks elapses, the new value is not transferred to the register.
- <4> When the count value of 8-bit timer counter H1 matches the value (M) of the CMP1 register before the change, the INTTMH1 signal is output, the carrier signal is inverted, and the timer counter is cleared to 00H. At the same time, the compare register whose value is to be compared with that of the 8-bit timer counter H1 is changed from the CMP11 register to the CMP01 register.
- <5> The timing at which the count value of the 8-bit timer counter H1 and the CMP11 register value match again is indicated by the value after the change (L).

Remark LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network.

LIN communication is single-master communication, and up to 15 slaves can be connected to one master.

The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.

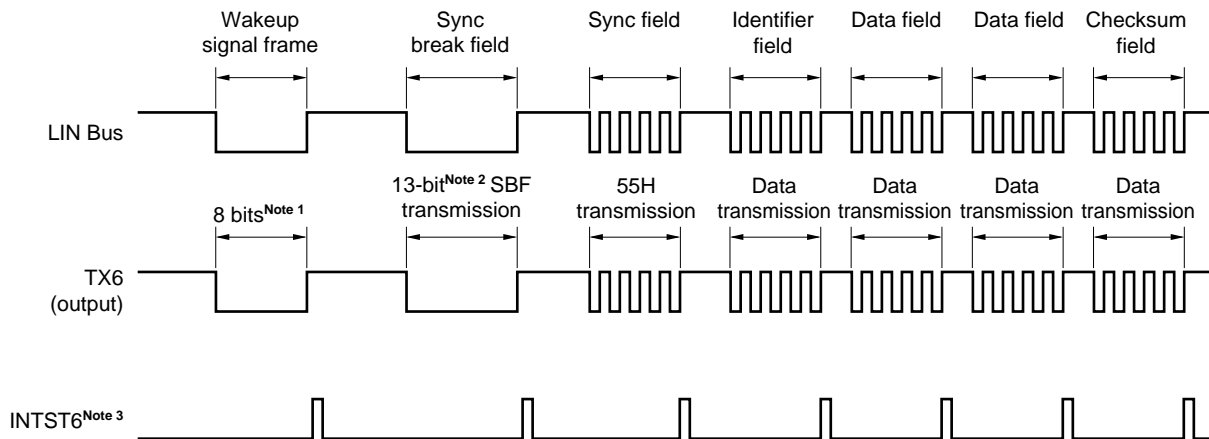
Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.

In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is $\pm 15\%$ or less.

Figures 14-1 and 14-2 outline the transmission and reception operations of LIN.

Figure 14-1. LIN Transmission Operation



- Notes**
1. The wakeup signal frame is substituted by 80H transmission in the 8-bit mode.
 2. The sync break field is output by hardware. The output width is the bit length set by bits 4 to 2 (SBL62 to SBL60) of asynchronous serial interface control register 6 (ASICL6) (see **14.4.2 (2) (h) SBF transmission**).
 3. INTST6 is output on completion of each transmission. It is also output when SBF is transmitted.

Remark The interval between each field is controlled by software.

(2) LCD display mode register (LCDM)

LCDM specifies whether to enable display operation. It also specifies whether to enable segment pin/common pin output, gate booster circuit control, and the display mode.

LCDM is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDM to 00H.

Figure 15-3. Format of LCD Display Mode Register

Address: FFB1H After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	3	2	1	0
LCDM	LCDON	SCOC	0	VAON	0	LCDM2	LCDM1	LCDM0

LCDON	LCD display enable/disable
0	Display off (all segment outputs are deselected.)
1	Display on

SCOC	Segment pin/common pin output control ^{Note 1}
0	Output ground level to segment/common pin
1	Output deselect level to segment pin and LCD waveform to common pin

VAON	Gate booster circuit control ^{Notes 1, 2}
0	No gate voltage boosting
1	Gate voltage boosting

LCDM2	LCDM1	LCDM0	LCD controller/driver display mode selection	
			Resistance division method	
			Number of time slices	Bias mode
1	1	1	8	1/4 ^{Note 3}
0	0	0	4	1/3
0	0	1	3	1/3
0	1	0	2	1/2
0	1	1	3	1/2
1	0	0	Static	
Other than above			Setting prohibited	

(Note and Caution are listed on the next page.)

16.3 Registers Controlling Manchester Code Generator

The following six types of registers are used to control the Manchester code generator.

- MCG control register 0 (MC0CTL0)
- MCG control register 1 (MC0CTL1)
- MCG control register 2 (MC0CTL2)
- MCG status register (MC0STR)
- Port mode register 3 (PM3)
- Port register 3 (P3)

(1) MCG control register 0 (MC0CTL0)

This register is used to set the operation mode and to enable/disable the operation.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 10H.

Figure 16-4. Format of MCG Control Register 0 (MC0CTL0)

Address: FF4CH After reset: 10H R/W

Symbol	<7>	6	5	<4>	3	2	<1>	<0>
MC0CTL0	MC0PWR	0	0	MC0DIR	0	0	MC0OSL	MC0OLV

MC0PWR	Operation control
0	Operation stopped
1	Operation enabled

MC0DIR	First bit specification
0	MSB
1	LSB

MC0OSL	Data format
0	Manchester code
1	Bit sequential data

MC0OLV	Output level when transmission suspended
0	Low level
1	High level

Caution Clear (0) the MC0PWR bit before rewriting the MC0DIR, MC0OSL, and MC0OLV bits (it is possible to rewrite these bits by an 8-bit memory manipulation instruction at the same time when the MC0PWR bit is set (1)).

<2> Error of baud rate

The baud rate error can be calculated by the following expression.

$$\bullet \text{ Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1 \right) \times 100 [\%]$$

Caution Keep the baud rate error during transmission to within the permissible error range at the reception destination.

Example: Frequency of base clock = 2.5 MHz = 2,500,000 Hz

Set value of MC0BRS4 to MC0BRS0 bits of MC0CTL2 register = 10000B (k = 16)

Target baud rate = 76,800 bps

$$\begin{aligned} \text{Baud rate} &= 2.5 \text{ M}/(2 \times 16) \\ &= 2,500,000/(2 \times 16) = 78125 [\text{bps}] \end{aligned}$$

$$\begin{aligned} \text{Error} &= (78,125/76,800 - 1) \times 100 \\ &= 1.725 [\%] \end{aligned}$$

<3> Example of setting baud rate

Baud Rate [bps]	f _{PRS} = 10.0 MHz				f _{PRS} = 8.38 MHz				f _{PRS} = 8.0 MHz				f _{PRS} = 6.0 MHz			
	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]
4800	—	—	—	—	5, 6, or 7	27	4850	1.03	5, 6, or 7	26	4808	0.16	5, 6, or 7	20	4688	−2.34
9600	5, 6, or 7	16	9766	1.73	4	27	9699	1.03	5, 6, or 7	13	9615	0.16	4	20	9375	−2.34
19200	5	8	19531	1.73	3	27	19398	1.03	4	13	19231	0.16	4	10	18750	−2.34
31250	4	10	31250	0	2	17	30809	−1.41	4	8	31250	0	2	24	31250	0
38400	4	8	39063	1.73	2	27	38796	1.03	3	13	38462	0.16	2	20	37500	−2.34
56000	3	11	56818	1.46	2	19	55132	−1.55	3	9	55556	−0.79	1	27	55556	−0.79
62500	2	20	62500	0	2	17	61618	−1.41	3	8	62500	0	2	12	62500	0
76800	2	16	78125	1.73	1	27	77592	1.03	2	13	76923	0.16	2	10	75000	−2.34
115200	1	22	113636	−1.36	2	9	116389	1.03	1	17	117647	2.12	1	13	115385	0.16
125000	1	20	125000	0	1	17	123235	−1.41	1	16	125000	0	1	12	125000	0
153600	1	16	156250	1.73	2	7	149643	−2.58	1	13	153846	0.16	1	10	150000	−2.34
250000	1	10	250000	0	1	8	261875	4.75	1	8	250000	0	1	6	250000	0
					0	17	246471	−1.41								

Remark MC0CKS2 to MC0CKS0: Bits 2 to 0 of MCG control register 1 (MC0CTL1) (setting of base clock (f_{CLK}))
 k: Value set by bits 4 to 0 (MC0BRS4 to MC0BRS0) of MCG control register 2 (MC0CTL2) (k = 4, 5, 6, ..., 31)
 f_{PRS}: Peripheral hardware clock frequency
 ERR: Baud rate error

(2) Transmit operation

In bit sequential buffer mode, data is transmitted in 1- to 8-bit units. Transmission is enabled if bit 7 (MC0PWR) of MCG control register 0 (MC0CTL0) is set to 1.

The output value while transmission is suspended can be set by using bit 0 (MC0OLV) of the MC0CTL0 register. A transmission starts by writing a value to the MCG transmit buffer register (MC0TX) after setting the transmit data bit length to the MCG transmit bit count specification register (MC0BIT). At the transmission start timing, the MC0BIT value is transferred to the 3-bit counter and data of MC0TX is transferred to the 8-bit shift register. An interrupt request signal (INTMCG) occurs at the timing that the MC0TX value is transferred to the 8-bit shift register. The 8-bit shift register is continuously shifted by the baud rate clock and is output from the MCGO pin. When continuous transmission is executed, the next data is set to MC0BIT and MC0TX during data transmission after INTMCG occurs.

To transmit continuously, writing the next transfer data to MC0TX must be complete within the period (3) and (4) in Figure 16-9. Rewrite MC0BIT before writing to MC0TX during continuous transmission.

Figure 16-9. Timing of Bit Sequential Buffer Mode (LSB First) (1/4)

(1) Transmit timing (MC0OLV = 1, total transmit bit length = 8 bits)

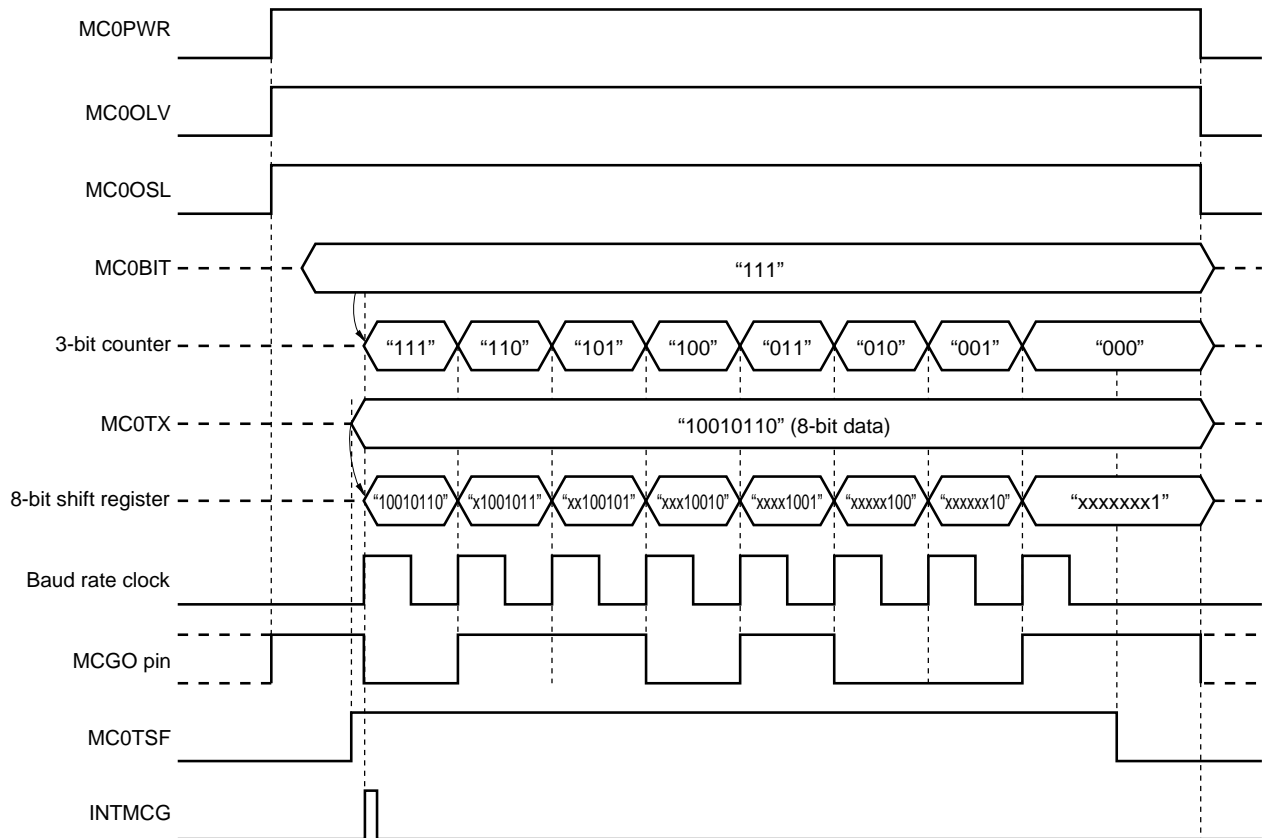
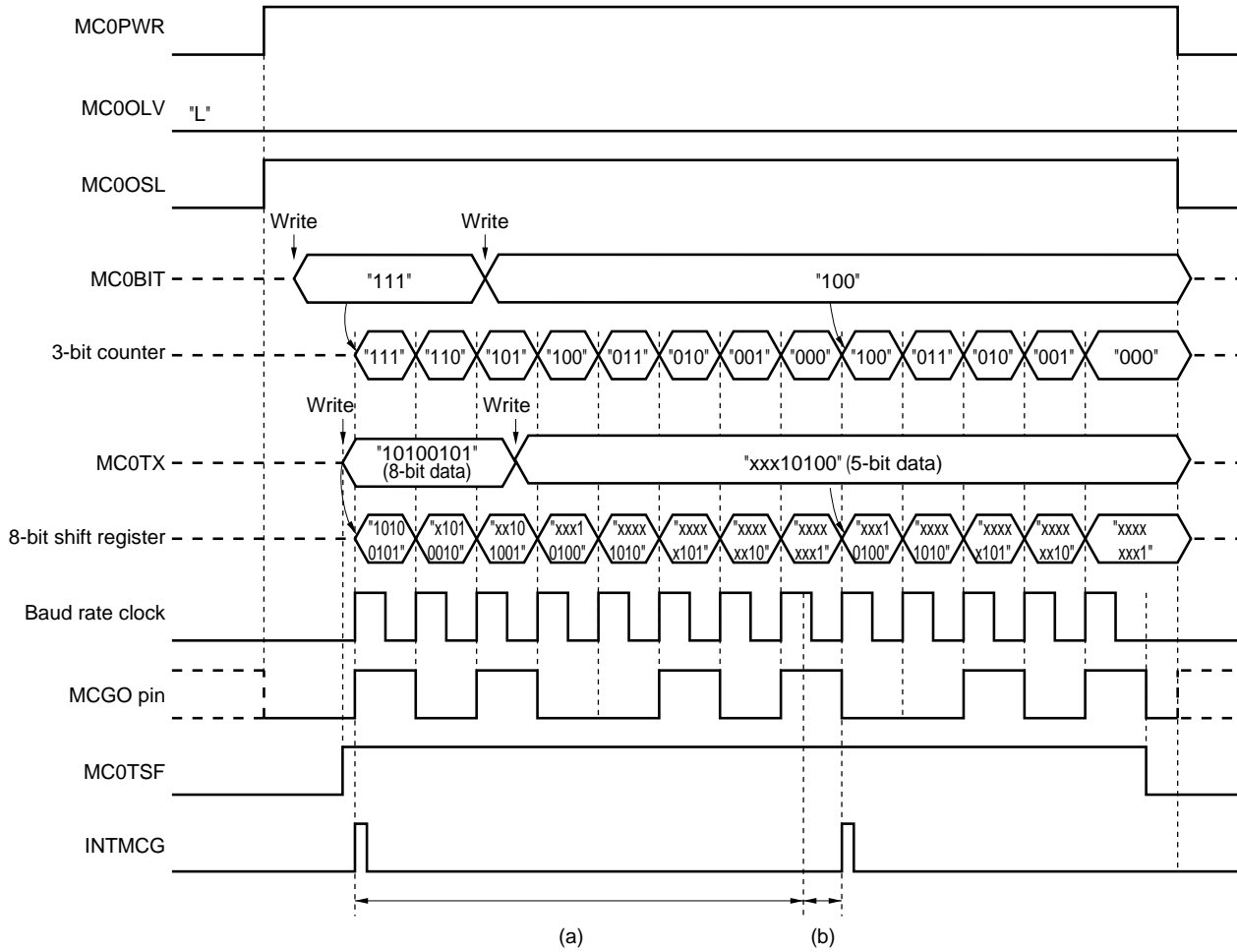


Figure 16-9. Timing of Bit Sequential Buffer Mode (LSB First) (4/4)

(4) Transmit timing (MC0OLV = 0, total transmit bit length = 13 bits)



(a): "8-bit transfer period" – (b)

(b): "1/2 cycle of baud rate" + 1 clock (f_{CLK}) before the last bit of transmit data

f_{CLK}: Frequency of operation base clock selected by using the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

Last bit: Transfer bit when 3-bit counter = 000

Caution Writing the next transmit data to MC0TX must be complete within the period (a) during continuous transmission. If writing the next transmit data to MC0TX is executed in the period (b), the next data transmission starts 2 clocks (f_{CLK}) after the last bit has been transmitted. Rewrite the MC0BIT before writing to MC0TX during continuous transmission.

CHAPTER 20 RESET FUNCTION

The following four operations are available to generate a reset signal.

- (1) External reset input via $\overline{\text{RESET}}$ pin
- (2) Internal reset by watchdog timer program loop detection
- (3) Internal reset by comparison of supply voltage and detection voltage of power-on-clear (POC) circuit
- (4) Internal reset by comparison of supply voltage and detection voltage of low-power-supply detector (LVI)

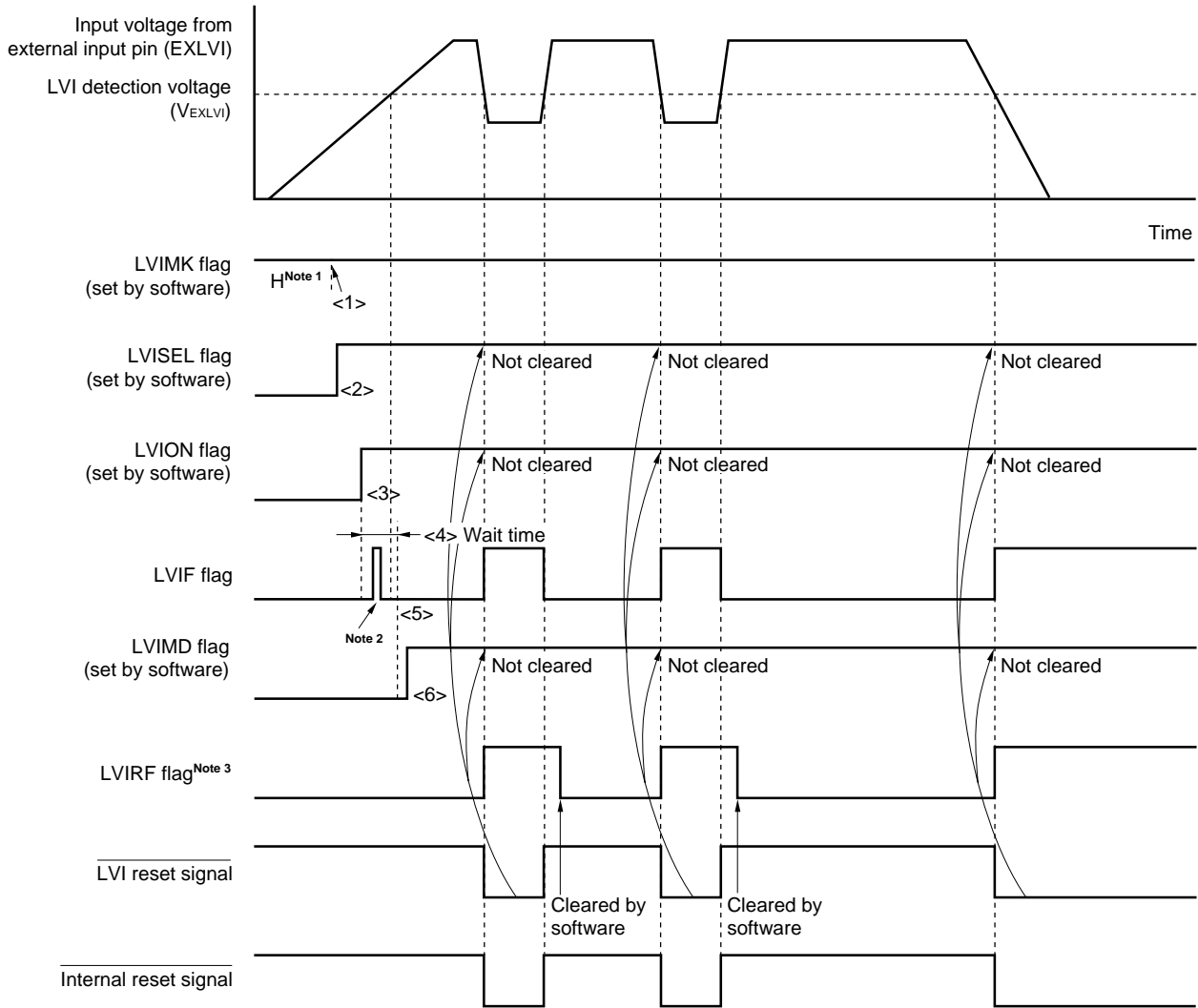
External and internal resets have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H when the reset signal is generated.

A reset is applied when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POC and LVI circuit voltage detection, and each item of hardware is set to the status shown in Tables 20-1 and 20-2. Each pin is high impedance during reset signal generation or during the oscillation stabilization time just after a reset release.

When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset. It is released from the reset status when a high level is input to the $\overline{\text{RESET}}$ pin and program execution is started with the internal high-speed oscillation clock after reset processing. A reset by the watchdog timer is automatically released, and program execution starts using the internal high-speed oscillation clock (see **Figures 20-2 to 20-4**) after reset processing. Reset by POC and LVI circuit power supply detection is automatically released when $V_{DD} \geq V_{POC}$ or $V_{DD} \geq V_{LVI}$ after the reset, and program execution starts using the internal high-speed oscillation clock (see **CHAPTER 21 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 22 LOW-VOLTAGE DETECTOR**) after reset processing.

- Cautions**
1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
 2. During reset input, the X1 clock, XT1 clock, internal high-speed oscillation clock, and internal low-speed oscillation clock stop oscillating. External main system clock input becomes invalid.
 3. When the STOP mode is released by a reset, the STOP mode contents are held during reset input. However, the port pins become high-impedance.

**Figure 22-6. Timing of Low-Voltage Detector Internal Reset Signal Generation
(Detects Level of Input Voltage from External Input Pin (EXLVI))**



- Notes**
1. The LVIMK flag is set to "1" by reset signal generation.
 2. The LVIF flag may be set (1).
 3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 20 RESET FUNCTION**.

Remark <1> to <6> in Figure 22-6 above correspond to <1> to <6> in the description of "When starting operation" in **22.4.1 (2) When detecting level of input voltage from external input pin (EXLVI)**.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	OR	A, #byte	2	4	–	$A \leftarrow A \vee \text{byte}$	x		
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	x		
		A, r Note 3	2	4	–	$A \leftarrow A \vee r$	x		
		r, A	2	4	–	$r \leftarrow r \vee A$	x		
		A, saddr	2	4	5	$A \leftarrow A \vee (\text{saddr})$	x		
		A, !addr16	3	8	9	$A \leftarrow A \vee (\text{addr16})$	x		
		A, [HL]	1	4	5	$A \leftarrow A \vee (\text{HL})$	x		
		A, [HL + byte]	2	8	9	$A \leftarrow A \vee (\text{HL} + \text{byte})$	x		
		A, [HL + B]	2	8	9	$A \leftarrow A \vee (\text{HL} + B)$	x		
		A, [HL + C]	2	8	9	$A \leftarrow A \vee (\text{HL} + C)$	x		
	XOR	A, #byte	2	4	–	$A \leftarrow A \nabla \text{byte}$	x		
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \nabla \text{byte}$	x		
		A, r Note 3	2	4	–	$A \leftarrow A \nabla r$	x		
		r, A	2	4	–	$r \leftarrow r \nabla A$	x		
		A, saddr	2	4	5	$A \leftarrow A \nabla (\text{saddr})$	x		
		A, !addr16	3	8	9	$A \leftarrow A \nabla (\text{addr16})$	x		
		A, [HL]	1	4	5	$A \leftarrow A \nabla (\text{HL})$	x		
		A, [HL + byte]	2	8	9	$A \leftarrow A \nabla (\text{HL} + \text{byte})$	x		
		A, [HL + B]	2	8	9	$A \leftarrow A \nabla (\text{HL} + B)$	x		
		A, [HL + C]	2	8	9	$A \leftarrow A \nabla (\text{HL} + C)$	x		
	CMP	A, #byte	2	4	–	$A - \text{byte}$	x	x	x
		saddr, #byte	3	6	8	$(\text{saddr}) - \text{byte}$	x	x	x
		A, r Note 3	2	4	–	$A - r$	x	x	x
		r, A	2	4	–	$r - A$	x	x	x
		A, saddr	2	4	5	$A - (\text{saddr})$	x	x	x
		A, !addr16	3	8	9	$A - (\text{addr16})$	x	x	x
		A, [HL]	1	4	5	$A - (\text{HL})$	x	x	x
		A, [HL + byte]	2	8	9	$A - (\text{HL} + \text{byte})$	x	x	x
		A, [HL + B]	2	8	9	$A - (\text{HL} + B)$	x	x	x
		A, [HL + C]	2	8	9	$A - (\text{HL} + C)$	x	x	x

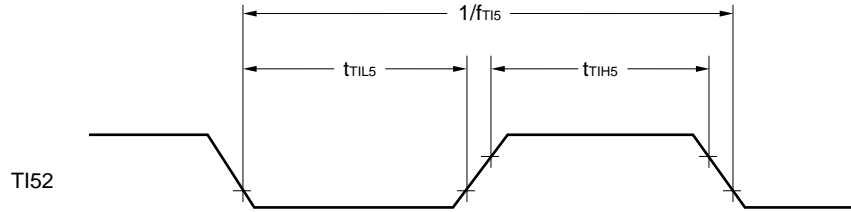
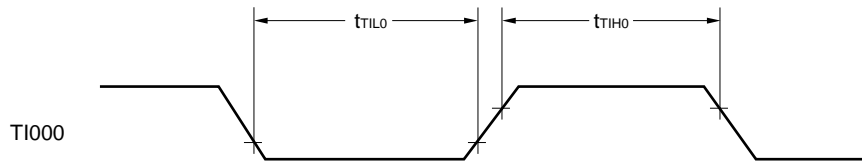
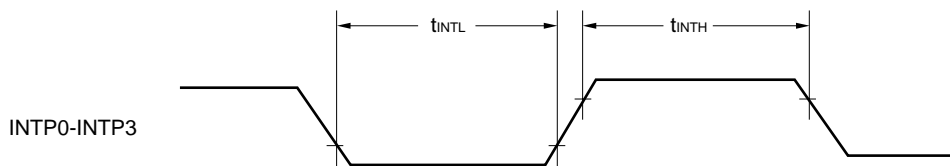
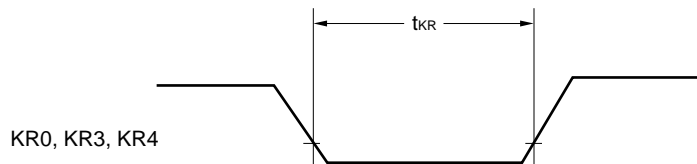
- Notes**
1. When the internal high-speed RAM area is accessed or for an instruction with no data access
 2. When an area except the internal high-speed RAM area is accessed
 3. Except "r = A"

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).
 2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	ADDW	AX, #word	3	6	–	$AX, CY \leftarrow AX + \text{word}$	×	×	×
	SUBW	AX, #word	3	6	–	$AX, CY \leftarrow AX - \text{word}$	×	×	×
	CMPW	AX, #word	3	6	–	$AX - \text{word}$	×	×	×
Multiply/divide	MULU	X	2	16	–	$AX \leftarrow A \times X$			
	DIVUW	C	2	25	–	$AX \text{ (Quotient)}, C \text{ (Remainder)} \leftarrow AX \div C$			
Increment/decrement	INC	r	1	2	–	$r \leftarrow r + 1$	×	×	
		saddr	2	4	6	$(saddr) \leftarrow (saddr) + 1$	×	×	
	DEC	r	1	2	–	$r \leftarrow r - 1$	×	×	
		saddr	2	4	6	$(saddr) \leftarrow (saddr) - 1$	×	×	
	INCW	rp	1	4	–	$rp \leftarrow rp + 1$			
	DECW	rp	1	4	–	$rp \leftarrow rp - 1$			
Rotate	ROR	A, 1	1	2	–	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1 \text{ time}$			×
	ROL	A, 1	1	2	–	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1 \text{ time}$			×
	RORC	A, 1	1	2	–	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1 \text{ time}$			×
	ROLC	A, 1	1	2	–	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1 \text{ time}$			×
	ROR4	[HL]	2	10	12	$A_{3-0} \leftarrow (HL)_{3-0}, (HL)_{7-4} \leftarrow A_{3-0}, (HL)_{3-0} \leftarrow (HL)_{7-4}$			
	ROL4	[HL]	2	10	12	$A_{3-0} \leftarrow (HL)_{7-4}, (HL)_{3-0} \leftarrow A_{3-0}, (HL)_{7-4} \leftarrow (HL)_{3-0}$			
BCD adjustment	ADJBA		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulate	MOV1	CY, saddr.bit	3	6	7	$CY \leftarrow (saddr.bit)$			×
		CY, sfr.bit	3	–	7	$CY \leftarrow sfr.bit$			×
		CY, A.bit	2	4	–	$CY \leftarrow A.bit$			×
		CY, PSW.bit	3	–	7	$CY \leftarrow PSW.bit$			×
		CY, [HL].bit	2	6	7	$CY \leftarrow (HL).bit$			×
		saddr.bit, CY	3	6	8	$(saddr.bit) \leftarrow CY$			
		sfr.bit, CY	3	–	8	$sfr.bit \leftarrow CY$			
		A.bit, CY	2	4	–	$A.bit \leftarrow CY$			
		PSW.bit, CY	3	–	8	$PSW.bit \leftarrow CY$	×	×	
		[HL].bit, CY	2	6	8	$(HL).bit \leftarrow CY$			

- Notes**
1. When the internal high-speed RAM area is accessed or for an instruction with no data access
 2. When an area except the internal high-speed RAM area is accessed

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).
 2. This clock cycle applies to the internal ROM program.

TI Timing**Interrupt Request Input Timing****Key Interrupt Input Timing****RESET Input Timing**