Renesas Electronics America Inc - UPD78F0455GK-GAJ-AX Datasheet

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	78K/0
Core Size	8-Bit
Speed	10MHz
Connectivity	3-Wire SIO, LINbus, UART/USART
Peripherals	LCD, LVD, POR, PWM, WDT
Number of I/O	46
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f0455gk-gaj-ax

An outline of the timer is shown below.

		16-Bit Timer/ Event Counters 00	8-Bit Timer/ Event Counters 50, 51, and 52			8-Bit Timers H0, H1, and H2			Real-time Counter	Watchdog Timer
		TM00	TM50	TM51	TM52	TMH0	TMH1	TMH2		
Function	Interval timer	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel Note 1	–
	External event counter	1 channel Note 2	1 channel	1 channel	1 channel Note 2	–	–	– Note 2	–	–
	PPG output	1 output	–	–	–	–	–	–	–	–
	PWM output	–	1 output	1 output	–	1 output	1 output	–	–	–
	Pulse width measurement	2 inputs	–	–	–	–	–	–	–	–
	Square-wave output	1 output	1 output	1 output	–	1 output	1 output	–	–	–
	Carrier generator	–	–	– Note 3	–	–	1 output Note 3	–	–	–
	Calendar function	–	–	–	–	–	–	–	1 channel Note 1	
	RTC output	–	–	–	–	–	–	–	2 outputs Note 4	–
	Watchdog timer	–	–	–	–	–	–	–	–	1 channel
Interrupt source		2	1	1	1	1	1	1	1	–

Notes 1. In the real-time counter, the Interval timer function and calendar function can be used simultaneously.

2. TM52 and TM00 can be connected in cascade to be used as a 24-bit counter. Also, the external event input of TM52 can be input enable-controlled via TMH2.

3. TM51 and TMH1 can be used in combination as a carrier generator mode.

4. A 1 Hz output can be used as one output and a 512 Hz, 16.384 kHz, or 32.768 kHz output can be used as one output.

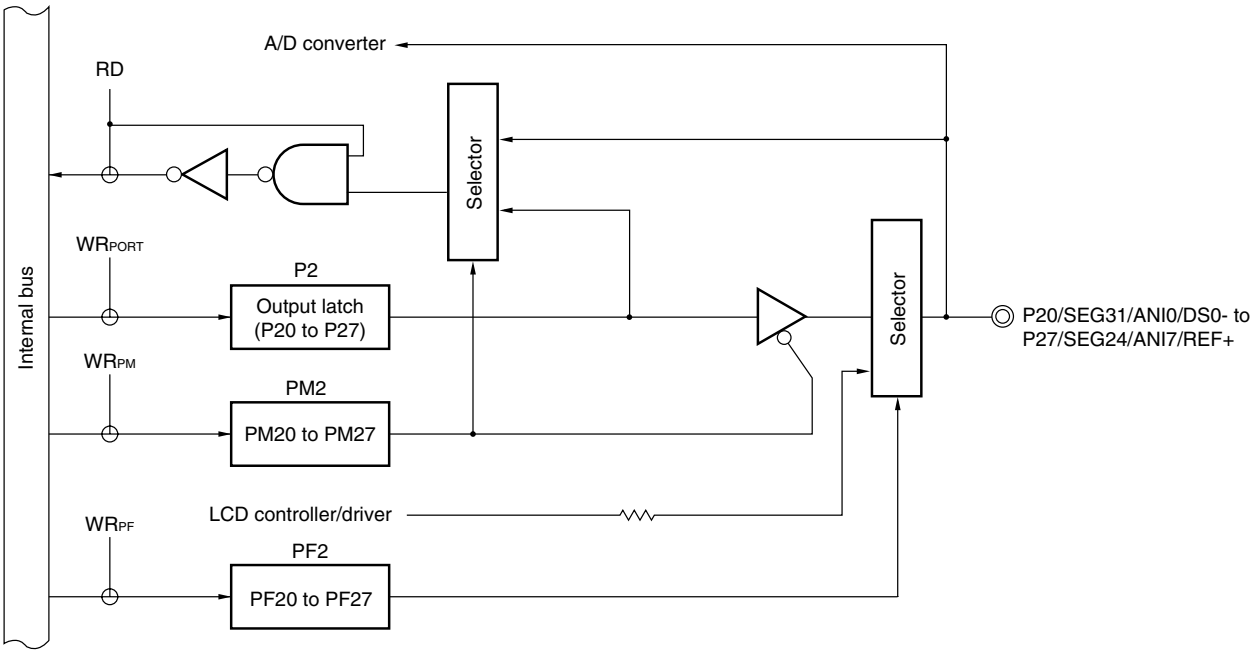
1.9 Outline of Functions (μPD78F046x)

(1/2)

Item		μPD78F0461	μPD78F0462	μPD78F0463	μPD78F0464	μPD78F0465
Internal memory	Flash memory (self-programming supported)[Note]	16 KB	24 KB	32 KB	48 KB	60 KB
	High-speed RAM[Note]	768 bytes	1 KB			
	Expansion RAM[Note]	–			1 KB	
	LCD display RAM	24 × 4 bits (with 4 com) or 20 × 8 bits (with 8 com)				
Memory space		64 KB				
Main system clock (oscillation frequency)	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) 2 to 10 MHz: V_{DD} = 2.7 to 5.5 V, 2 to 5 MHz: V_{DD} = 1.8 to 5.5 V				
	Internal high-speed oscillation clock	Internal oscillation 8 MHz (TYP.): V_{DD} = 1.8 to 5.5 V				
Subsystem clock (oscillation frequency)		XT1 (crystal) oscillation 32.768 kHz (TYP.): V_{DD} = 1.8 to 5.5 V				
Internal low-speed oscillation clock (for TMH1, WDT)		Internal oscillation 240 kHz (TYP.): V_{DD} = 1.8 to 5.5 V				
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Minimum instruction execution time		0.2 μs (high-speed system clock: @ f_{XH} = 10 MHz operation)				
		0.25 μs (internal high-speed oscillation clock: @ f_{RH} = 8 MHz (TYP.) operation)				
		122 μs (subsystem clock: @ f_{SUB} = 32.768 kHz operation)				
Instruction set		• 8-bit operation and 16-bit operation • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.				
I/O ports		Total: 46 CMOS I/O: 42 CMOS input: 4				
Timers		• 16-bit timer/event counter: 1 channel • 8-bit timer/event counter: 3 channels (out of which 2 channels can perform PWM output) • 8-bit timer: 3 channels (out of which 2 channels can perform PWM output) • Real-time counter: 1 channel • Watchdog timer: 1 channel				
	Timer outputs	5 (PWM output: 4 and PPG output: 1)				
	RTC outputs	2 • 1 Hz (Subsystem clock: f_{SUB} = 32.768 kHz) • 512 Hz or 16.384 kHz or 32.768 kHz (Subsystem clock: f_{SUB} = 32.768 kHz)				
Buzzer output		• 1.22 kHz, 2.44 kHz, 4.88 kHz, 9.77 MHz (peripheral hardware clock: @ f_{PRS} = 10 MHz operation)				

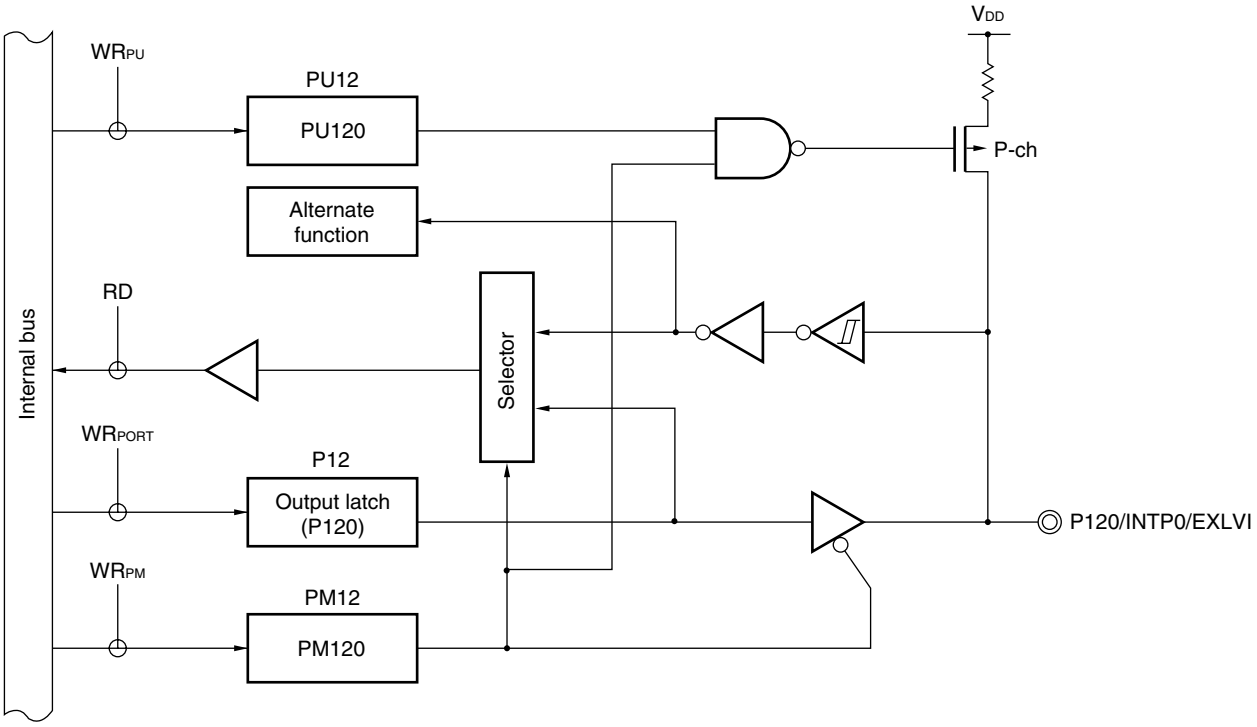
Note The internal flash memory capacity, internal high-speed RAM capacity, and internal expansion RAM capacity can be changed using the internal memory size switching register (IMS) and the internal expansion RAM size switching register (IXS).

<R> **Figure 4-5. Block Diagram of P20 to P27**



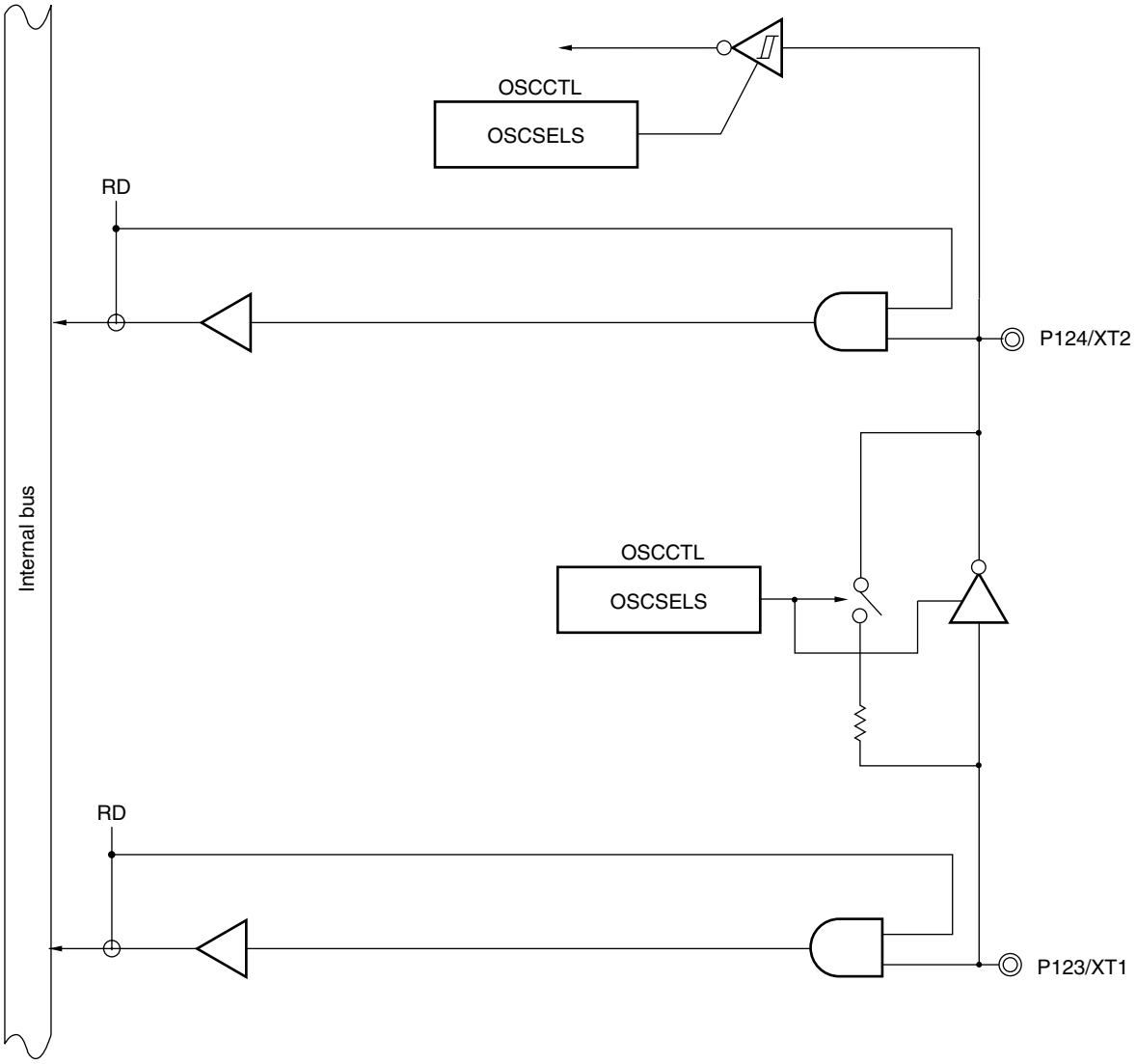
P2: Port register 2
PM2: Port mode register 2
PF2: Port function register 2
RD: Read signal
WR××: Write signal

Figure 4-16. Block Diagram of P120



P12: Port register 12
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
RD: Read signal
WR××: Write signal

Figure 4-18. Block Diagram of P123 and P124



OSCCTL: Clock operation mode select register
RD: Read signal

(4) Prescaler mode register 00 (PRM00)

PRM00 is the register that sets the TM00 count clock and TI000 and TI010 pin input valid edges.

Rewriting PRM00 is prohibited during operation (when TMC003 and TMC002 = other than 00).

PRM00 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PRM00 to 00H.

Cautions 1. Do not apply the following setting when setting the PRM001 and PRM000 bits to 11 (to specify the valid edge of the TI000 pin as a count clock).

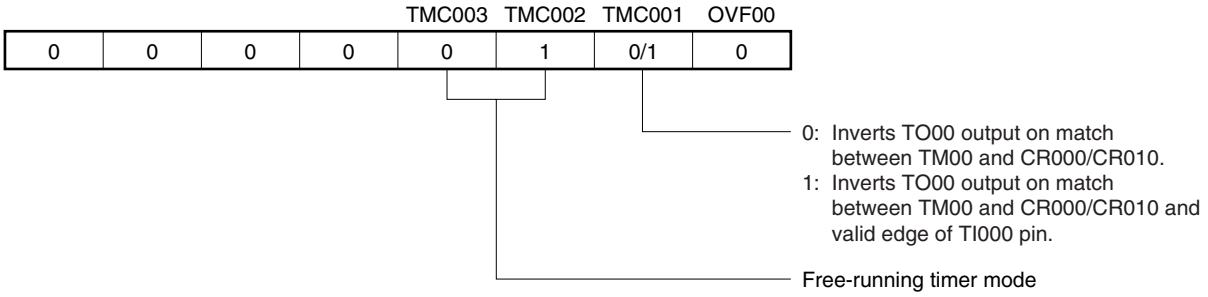
- **Clear & start mode entered by the TI000 pin valid edge**
- **Setting the TI000 pin as a capture trigger**

2. If the operation of the 16-bit timer/event counter 00 is enabled when the TI000 or TI010 pin is at high level and when the valid edge of the TI000 or TI010 pin is specified to be the rising edge or both edges, the high level of the TI000 or TI010 pin is detected as a rising edge. Note this when the TI000 or TI010 pin is pulled up. However, the rising edge is not detected when the timer operation has been once stopped and then is enabled again.

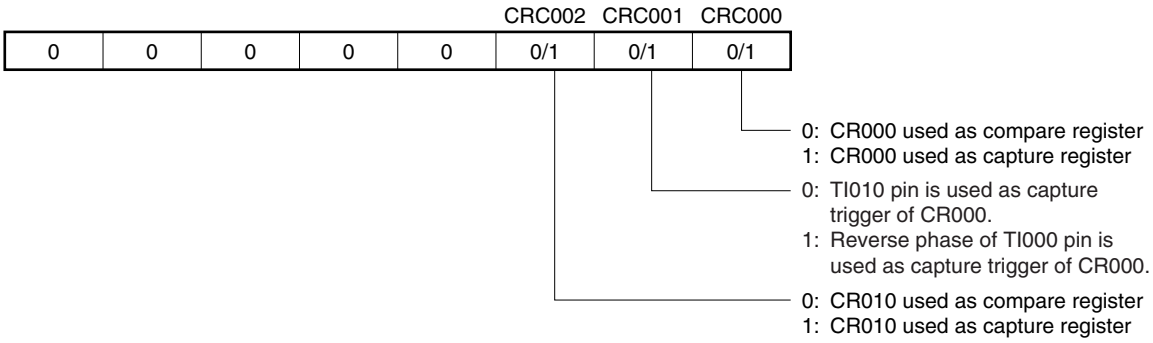
3. The valid edge of TI010 and timer output (TO00) cannot be used for the P34 pin at the same time. Select either of the functions.

Figure 6-39. Example of Register Settings in Free-Running Timer Mode (1/2)

(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)

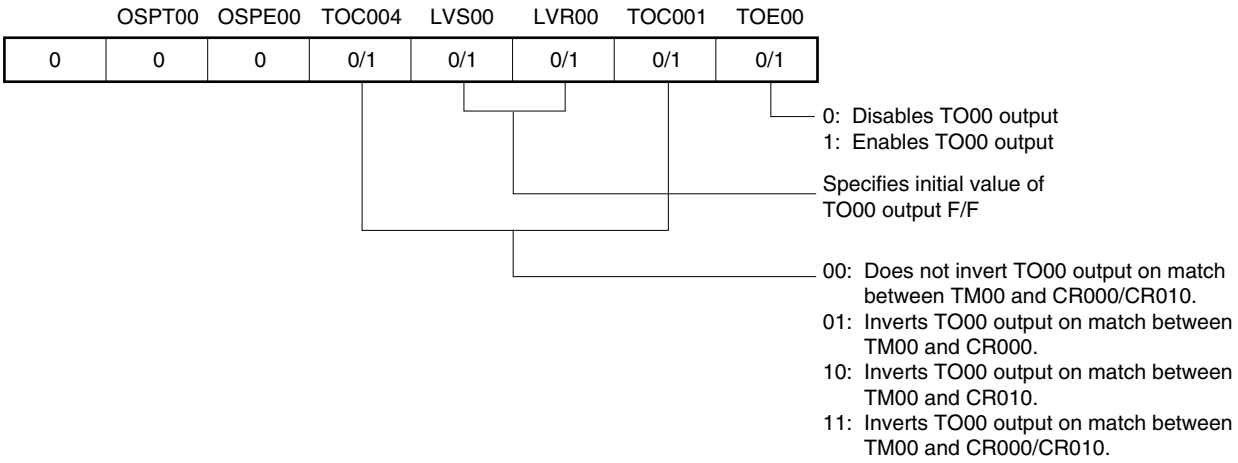
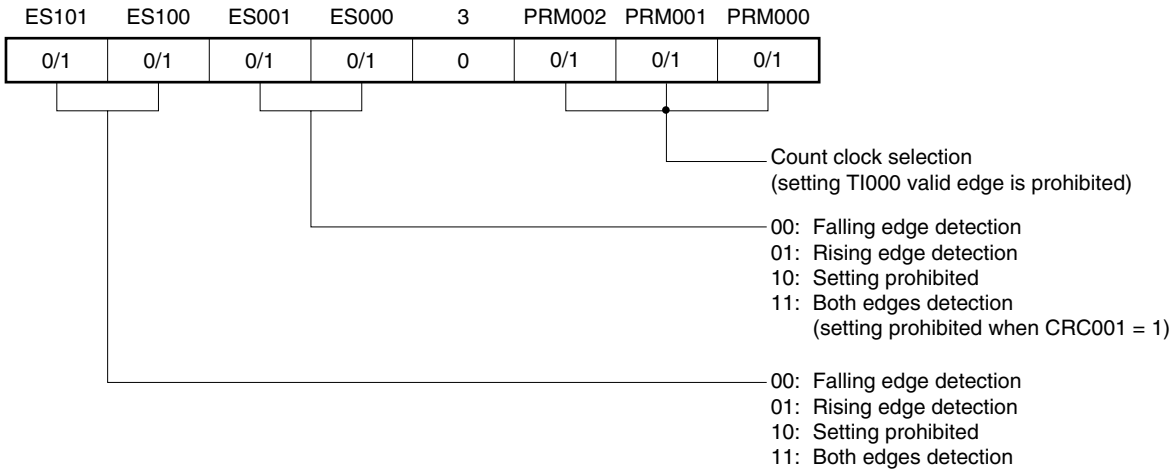


Figure 6-39. Example of Register Settings in Free-Running Timer Mode (2/2)

(d) Prescaler mode register 00 (PRM00)



(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM000) is generated. The count value of TM00 is not cleared.

To use this register as a capture register, select either the TI000 or TI010 pin input as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR000.

(g) 16-bit capture/compare register 010 (CR010)

When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM010) is generated. The count value of TM00 is not cleared.

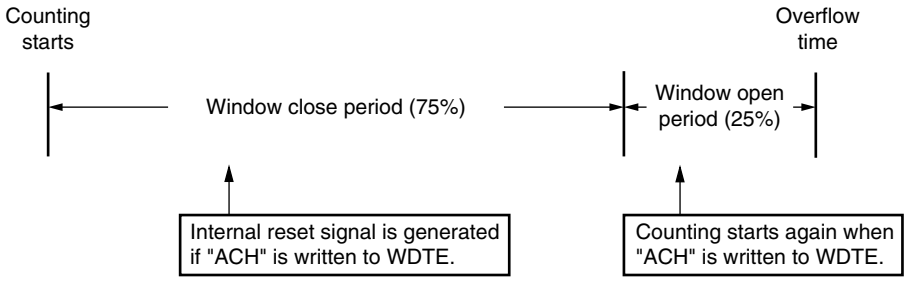
When this register is used as a capture register, the TI000 pin input is used as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR010.

10.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (0080H). The outline of the window is as follows.

- If “ACH” is written to WDTE during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to WDTE during the window close period, an abnormality is detected and an internal reset signal is generated.

Example: If the window open period is 25%



Caution The first writing to WDTE after a reset release clears the watchdog timer, if it is made before the overflow time regardless of the timing of the writing, and the watchdog timer starts counting again.

The window open period to be set is as follows.

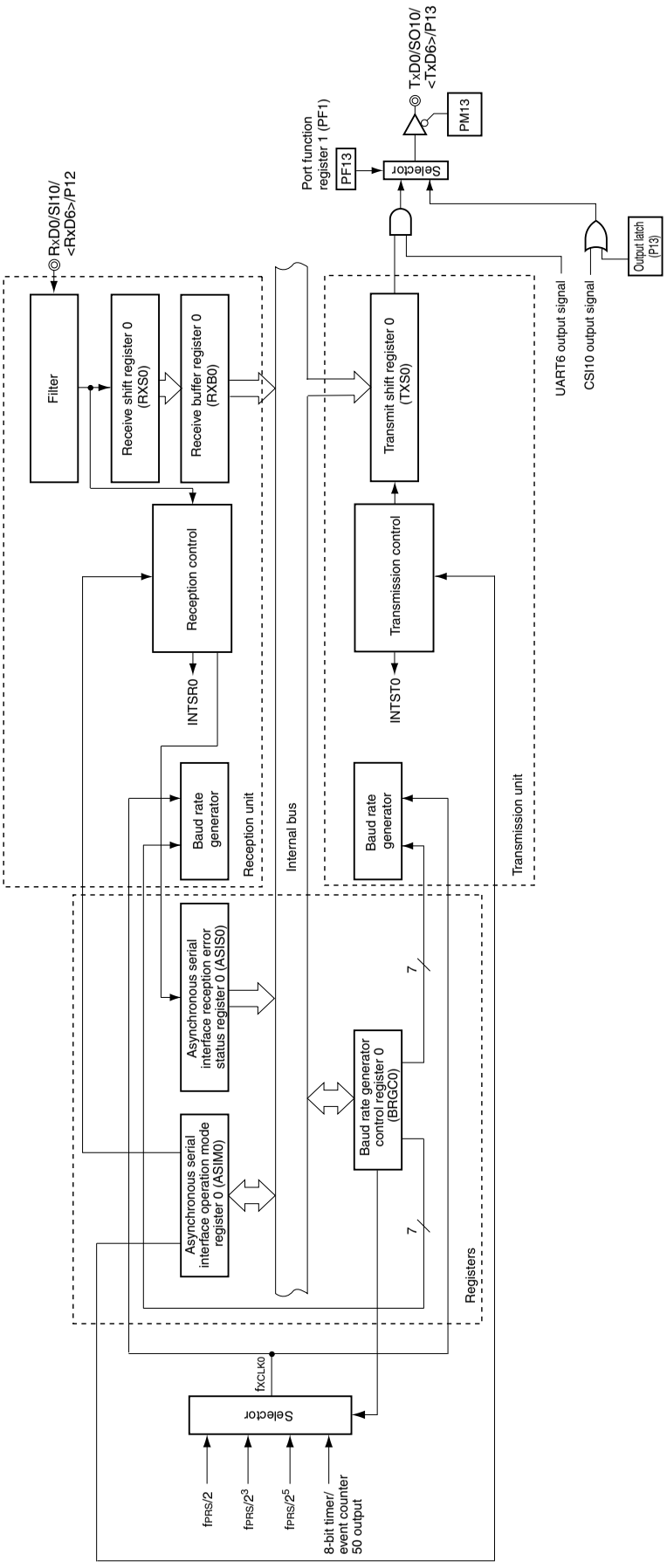
Table 10-4. Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	25%
0	1	50%
1	0	75%
1	1	100%

Cautions 1. The combination of WDCS2 = WDCS1 = WDCS0 = 0 and WINDOW1 = WINDOW0 = 0 is prohibited.

2. The watchdog timer continues its operation during self-programming and EEPROM emulation of the flash memory. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

Figure 14-1. Block Diagram of Serial Interface UART0

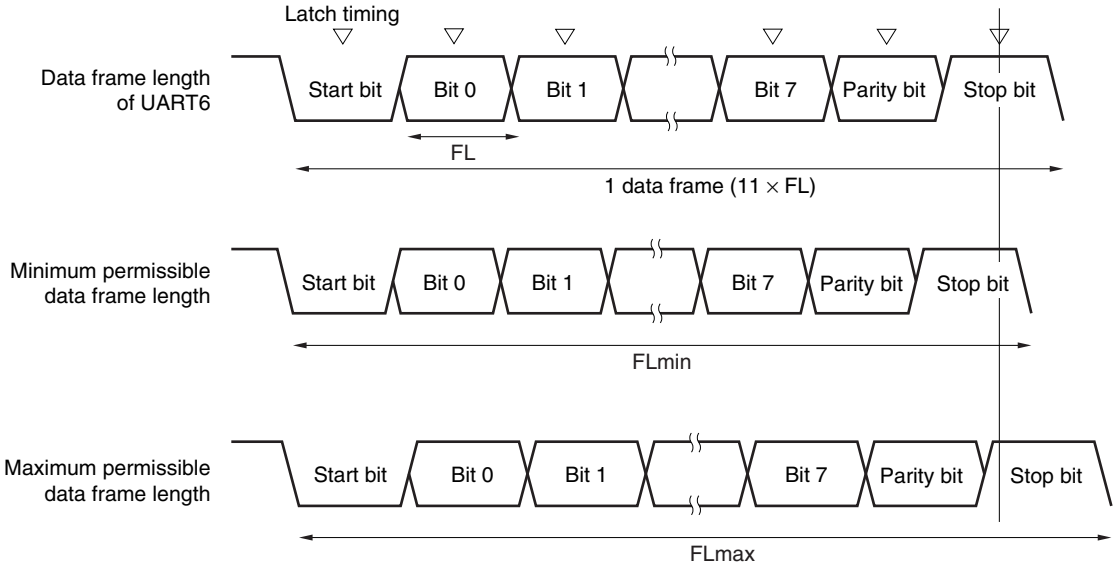


(4) Permissible baud rate range during reception

The permissible error from the baud rate at the transmission destination during reception is shown below.

Caution Make sure that the baud rate error during reception is within the permissible error range, by using the calculation expression shown below.

Figure 15-27. Permissible Baud Rate Range During Reception



As shown in Figure 15-27, the latch timing of the receive data is determined by the counter set by baud rate generator control register 6 (BRGC6) after the start bit has been detected. If the last data (stop bit) meets this latch timing, the data can be correctly received.

Assuming that 11-bit data is received, the theoretical values can be calculated as follows.

$FL = (Brate)^{-1}$

Brate: Baud rate of UART6
k: Set value of BRGC6
FL: 1-bit data length
Margin of latch timing: 2 clocks

Figure 17-1. Block Diagram of LCD Controller/Driver

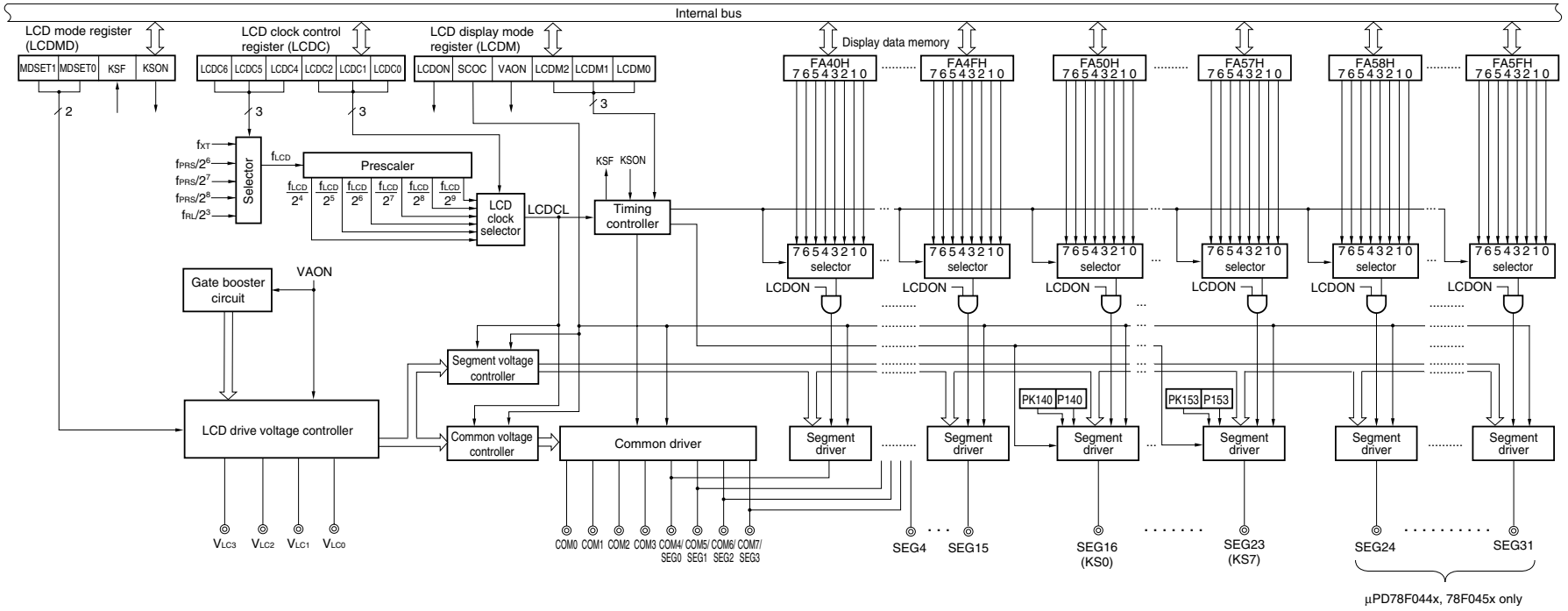
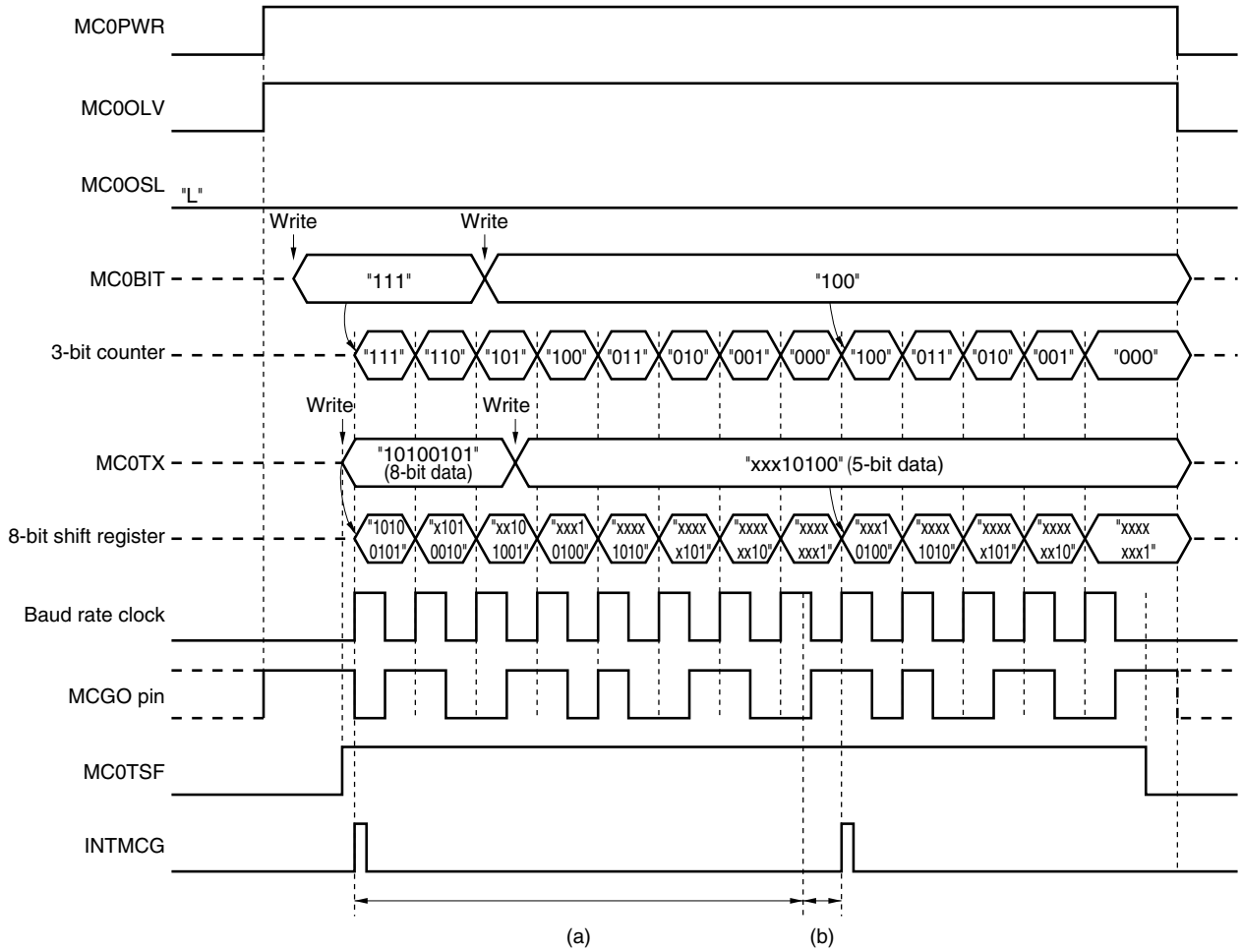


Figure 18-8. Timing of Manchester Code Generator Mode (LSB First) (3/4)

(3) Transmit timing (MC0OLV = 1, total transmit bit length = 13 bits)



(a): "8-bit transfer period" – (b)

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

f_{XCLK}: Frequency of the 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_{XCLK}) after the last bit has been transmitted. Rewrite the MC0BIT before writing to MC0TX during continuous transmission.

(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 18-9. Rewrite MC0BIT before writing to MC0TX during continuous transmission.

Figure 18-9. Timing of Bit Sequential Buffer Mode (LSB First) (1/4)

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

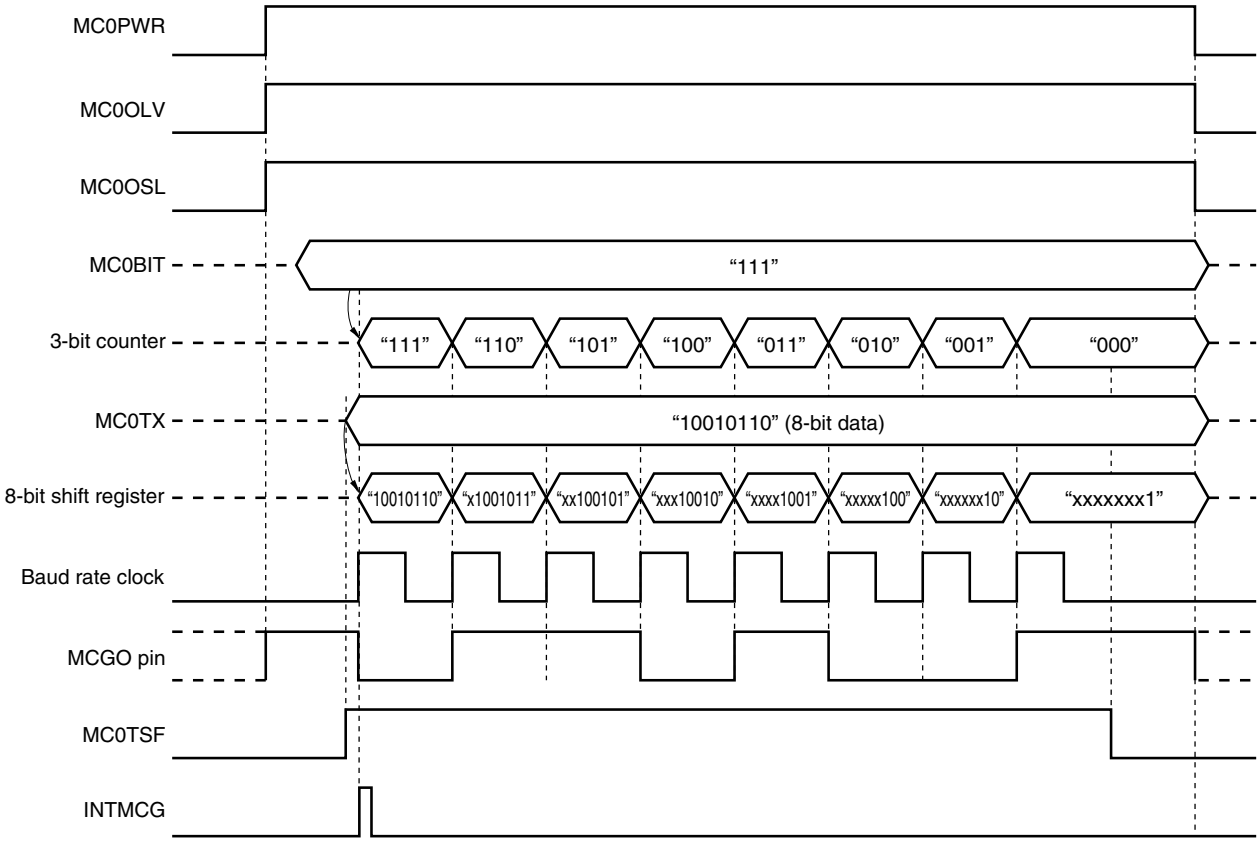
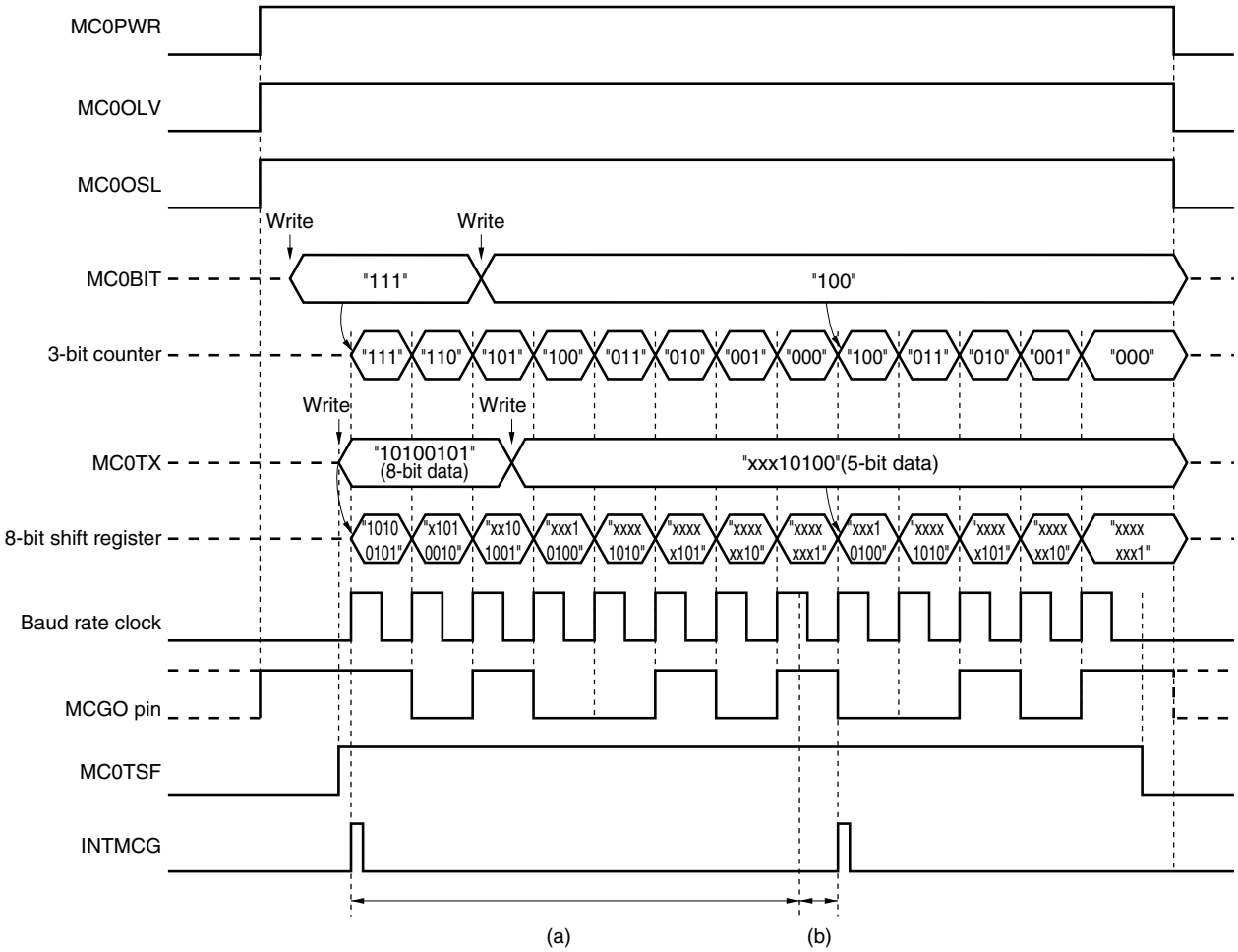


Figure 18-9. Timing of Bit Sequential Buffer Mode (LSB First) (3/4)

(3) Transmit timing (MC0OLV = 1, total transmit bit length = 13 bits)



(a): "8-bit transfer period" – (b)

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

f_{XCLK}: 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_{XCLK}) after the last bit has been transmitted. Rewrite the MC0BIT before writing to MC0TX during continuous transmission.

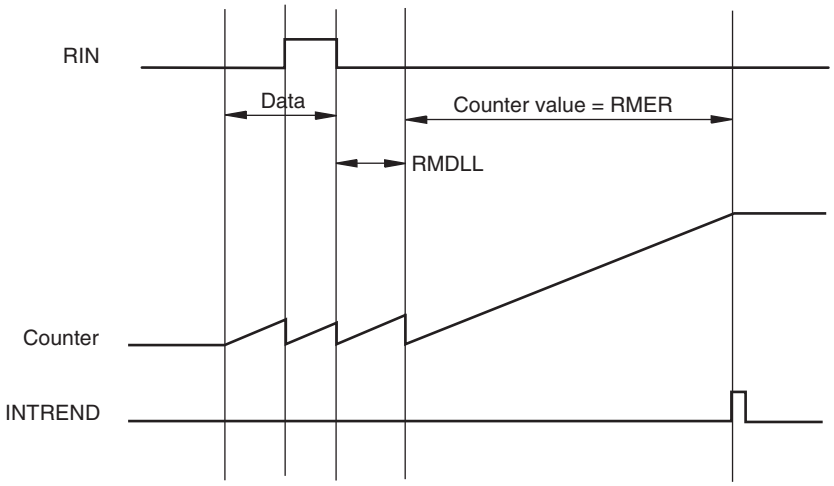
(14) Remote controller receive end-width select register (RMER)

This register determines the interval between the timing at which the INTREND signal is output.

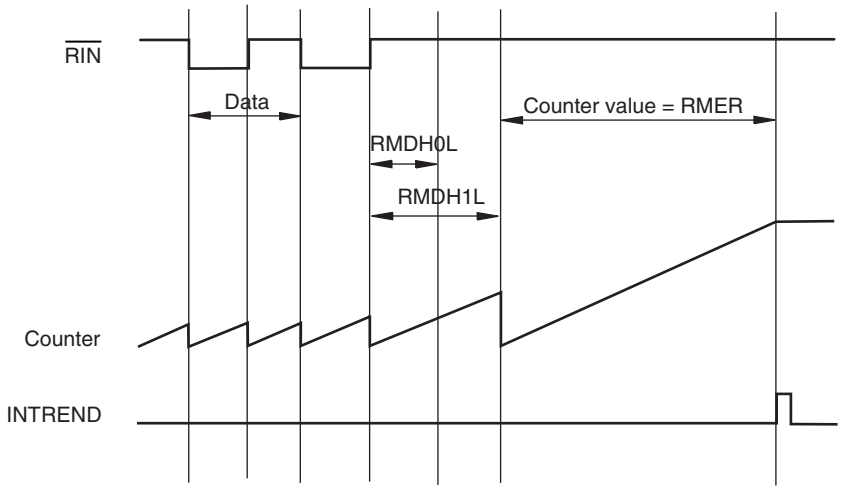
RMER is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMER to 00H.

(a) Type A reception mode

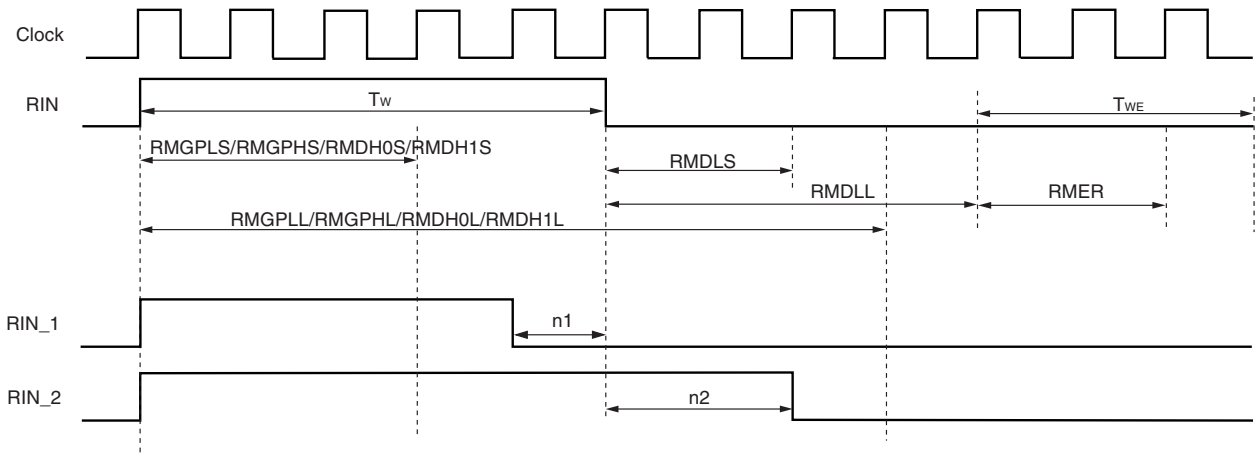


(b) Type B, Type C reception mode



Caution For RMER and all the remote controller receive compare registers (RMGPLS, RMGPLL, RMGPHS, RMGPHL, RMDLS, RMDLL, RMDH0S, RMDH0L, RMDH1S, and RMDH1L), disable remote controller reception (bit 7 (RMEN) of the remote controller receive control register (RMCN) = 0) first, and then change the value.

Figure 19-12. Setting Example (Where n1 = 1, n2 = 2)



(1) Formula for RMGPLS, RMGPHS, RMDLS, RMDH0S, and RMDH1S

$$\left[\frac{T_W \times (1 - a/100)}{1/f_{REMPRS}}\right]_{INT} - 2 - n1$$

(2) Formula for RMGPLL, RMGPHL, RMDLL, RMDH0L, and RMDH1L

$$\left[\frac{T_W \times (1 + a/100)}{1/f_{REMPRS}}\right]_{INT} + 1 + n2$$

(3) Formula for RMER

$$\left[\frac{T_{WE} \times (1 - a/100)}{1/f_{REMPRS}}\right]_{INT} - 1$$

T_W: Width of RIN input waveform

$1/f_{REMPRS}$: Width of internal operation clock cycle after division control by PRSEN

a: Tolerance (%)

$[\ ]_{INT}$: Round down the fractional portion of the value produced by the formula in the brackets.

n1, n2: Variables of waveform change caused by noise[Note1]

T_{WE}: End width of RIN input[Note2]

Notes 1. Set the values of n1 and n2 as required to meet the user's system specification.

2. This end width is counted after RMDLL.
The low-level width actually required after the last data has been received is as follows:
(RMDLL + 1 + RMER + 1) × (width of internal operation clock cycle after division control by PRSEN)

Table 20-1. Interrupt Source List (1/2)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	0	INTLVI	Low-voltage detection[Note 3]	Internal	0004H	(A)
	1	INTP0	Pin input edge detection	External	0006H	(B)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTSRE6	UART6 reception error generation	Internal	0012H	(A)
	7	INTSR6	End of UART6 reception		0014H	
	8	INTST6	End of UART6 transmission		0016H	
	9	INTCSI10/ INTST0	End of CSI10 communication/end of UART0 transmission		0018H	
	10	INTTMH1	Match between TMH1 and CMP01 (when compare register is specified)		001AH	
	11	INTTMH0	Match between TMH0 and CMP00 (when compare register is specified)		001CH	
	12	INTTM50	Match between TM50 and CR50 (when compare register is specified)		001EH	
	13	INTTM000	Match between TM00 and CR000 (when compare register is specified), TI010 pin valid edge detection (when capture register is specified)		0020H	
	14	INTTM010	Match between TM00 and CR010 (when compare register is specified), TI000 pin valid edge detection (when capture register is specified)		0022H	
	15	INTAD[Note 5]	End of 10-bit successive approximation type A/D conversion		0024H	
	16	INTSR0	End of UART0 reception or reception error generation		0026H	
	17	INTRTC	Fixed-cycle signal of real-time counter/alarm match detection		0028H	
	18	INTTM51 [Note 4]	Match between TM51 and CR51 (when compare register is specified)		002AH	
	19	INTKR	Key interrupt detection	External	002CH	(C)
	20	INTRTCI	Interval signal detection of real-time counter	Internal	002EH	(A)

Notes 1. The default priority determines the sequence of processing vectored interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 26 indicates the lowest priority.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 20-1.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is cleared to 0.

4. When 8-bit timer/event counter 51 and 8-bit timer H1 are used in the carrier generator mode, an interrupt is generated upon the timing when the INTTM5H1 signal is generated (see **Figure 8-15 Transfer Timing**).

5. μPD78F045x and 78F046x only.

Table 22-1. Operating Statuses in HALT Mode (1/2)

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on Internal High-Speed Oscillation Clock (f_{RH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EXCLK})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{RH}	Operation continues (cannot be stopped)	Status before HALT mode was set is retained	
		f_X	Status before HALT mode was set is retained	Operation continues (cannot be stopped)	Status before HALT mode was set is retained
		f_{EXCLK}	Operates or stops by external clock input		Operation continues (cannot be stopped)
	Subsystem clock	f_{XT}	Status before HALT mode was set is retained		
	f_{RL}		Status before HALT mode was set is retained		
CPU			Operation stopped		
Flash memory			Operation stopped		
RAM			Status before HALT mode was set is retained		
Port (latch)			Status before HALT mode was set is retained		
16-bit timer/event counter 00			Operable		
8-bit timer/event counter		50			
		51			
		52			
8-bit timer		H0			
		H1			
		H2			
Real-time counter					
Watchdog timer			Operable. Clock supply to watchdog timer stops when “internal low-speed oscillator can be stopped by software” is set by option byte.		
Buzzer output			Operable		
10-bit successive approximation type A/D converter					
16-bit ΔΣ type A/D converter					
Serial interface		UART0			
		UART6			
		CSI10			
LCD controller/driver					
Manchester code generator					
Remote controller receiver					
Power-on-clear function					
Low-voltage detection function					
External interrupt					

Remark f_{RH}: Internal high-speed oscillation clock
f_X: X1 clock
f_{EXCLK}: External main system clock
f_{XT}: XT1 clock
f_{RL}: Internal low-speed oscillation clock

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 ← AX + word	× × ×
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	× × ×
	CMPW	AX, #word	3	6	–	AX – word	× × ×
Multiply/ divide	**MULU**	X	2	16	–	AX ← A × X	
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C	
Increment/ decrement	**INC**	r	1	2	–	r ← r + 1	× ×
		saddr	2	4	6	(saddr) ← (saddr) + 1	× ×
	DEC	r	1	2	–	r ← r – 1	× ×
		saddr	2	4	6	(saddr) ← (saddr) – 1	× ×
	INCW	rp	1	4	–	rp ← rp + 1	
	DECW	rp	1	4	–	rp ← rp – 1	
Rotate	**ROR**	A, 1	1	2	–	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1 time	×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1 time	×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1 time	×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1 time	×
	ROR4	[HL]	2	10	12	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$	
	ROL4	[HL]	2	10	12	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$	
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 ← (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← sfr.bit	×
		CY, A.bit	2	4	–	CY ← A.bit	×
		CY, PSW.bit	3	–	7	CY ← PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← (HL).bit	×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY	
		sfr.bit, CY	3	–	8	sfr.bit ← CY	
		A.bit, CY	2	4	–	A.bit ← CY	
		PSW.bit, CY	3	–	8	PSW.bit ← CY	× ×
		[HL].bit, CY	2	6	8	(HL).bit ← 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.