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	-
Core Size	-
Speed	-
Connectivity	-
Peripherals	-
Number of I/O	-
Program Memory Size	-
Program Memory Type	-
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	-
Data Converters	-
Oscillator Type	-
Operating Temperature	-
Mounting Type	-
Package / Case	-
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd789488gk-524-9eu-a

LIST OF TABLES (3/3)

1.4 Pin Configuration (Top View)

(1) μPD789488, 78F9488

80-pin plastic QFP (14 × 14)

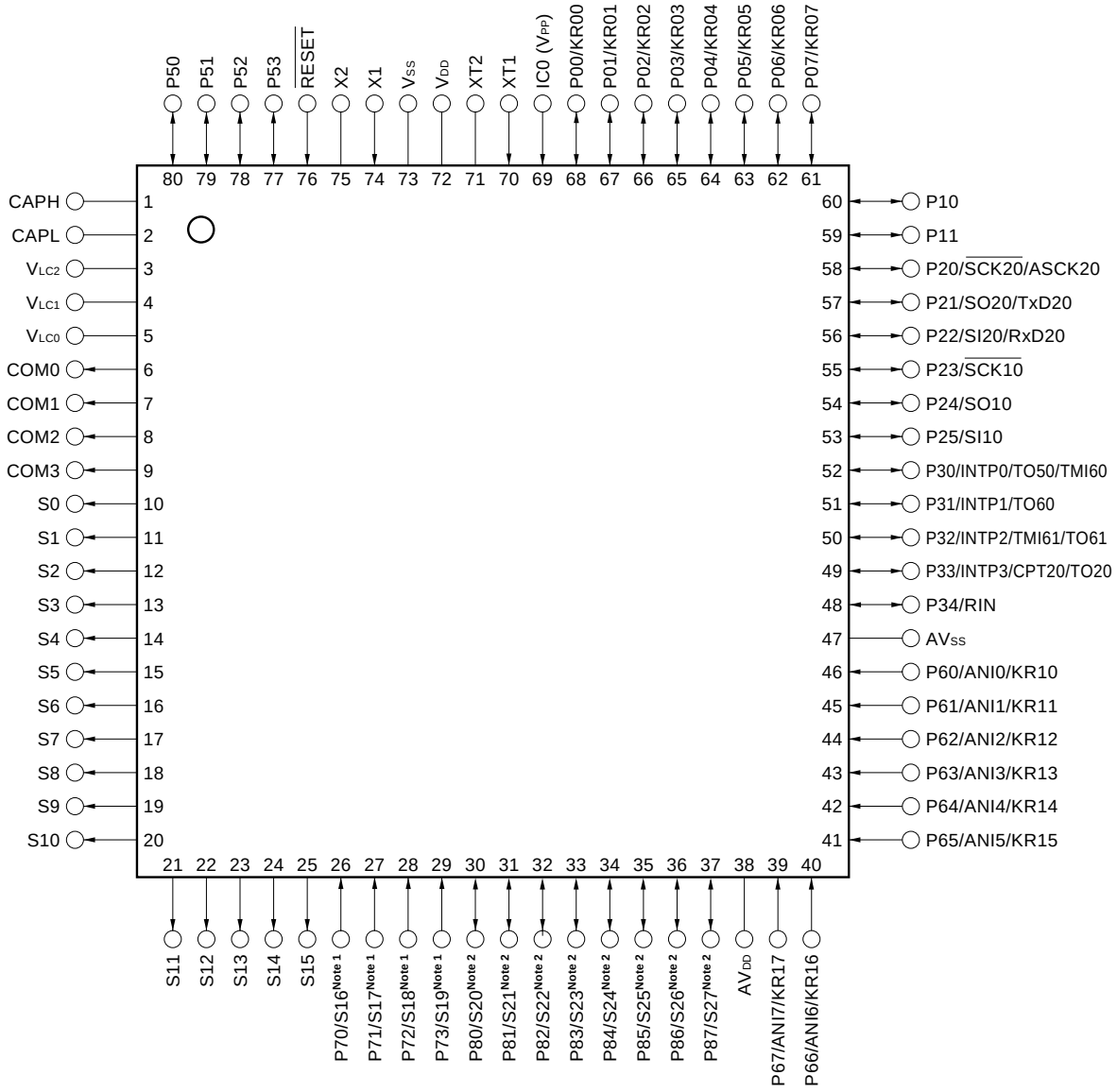
μPD789488GC-×××-8BT　　μPD78F9488GC-8BT

μPD789488GC-×××-8BT-A　　μPD78F9488GC-8BT-A

80-pin plastic TQFP (fine pitch) (12 × 12)

μPD789488GK-×××-9EU　　μPD78F9488GK-9EU

μPD789488GK-×××-9EU-A　　μPD78F9488GK-9EU-A



Notes 1. Whether to use these pins as input port pins (P70 to P73) or segment outputs (S16 to S19) can be selected in 1-bit units by means of a mask option or port function register (refer to **4.3 (3) Port function registers** and **CHAPTER 20 MASK OPTIONS**).

2. Whether to use these pins as I/O port pins (P80 to P87) or segment outputs (S20 to S27) can be selected in 1-bit units by means of a mask option or port function register (refer to **4.3 (3) Port function registers** and **CHAPTER 20 MASK OPTIONS**).

3.2.2 General-purpose registers

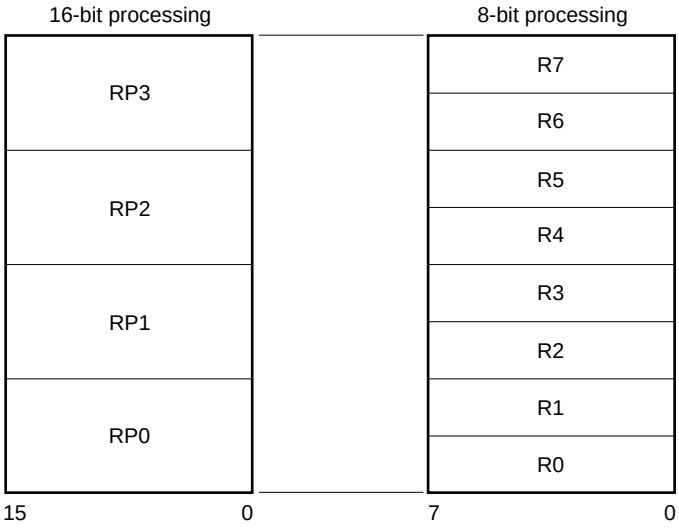
The general-purpose registers consist of eight 8-bit registers (X, A, C, B, E, D, L, and H).

Each register can be used as an 8-bit register, or two 8-bit registers in pairs can be used as a 16-bit register (AX, BC, DE, and HL).

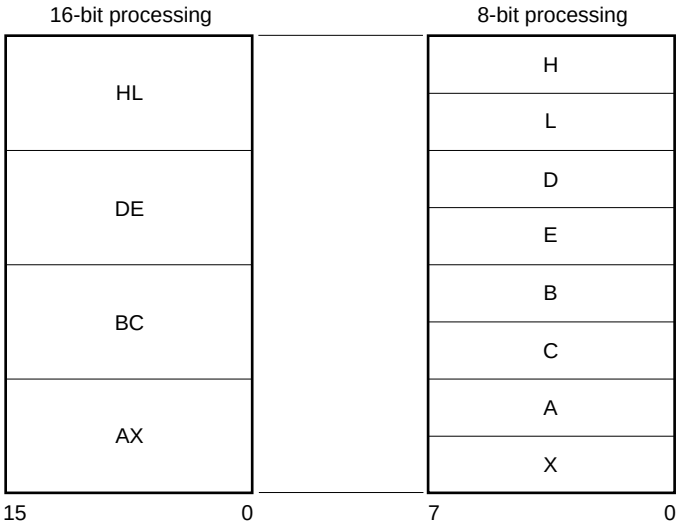
General-purpose registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, or HL) or absolute names (R0 to R7 and RP0 to RP3).

Figure 3-14. General-Purpose Register Configuration

(a) Absolute names



(b) Function names



3.4.2 Short direct addressing

[Function]

The memory to be manipulated in the fixed space is directly addressed with 8-bit data in an instruction word.

The fixed space is the 256-byte space FE20H to FF1FH where the addressing is applied. Internal high-speed RAM and special function registers (SFRs) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

The SFR area (FF00H to FF1FH) where short direct addressing is applied is a part of the whole SFR area. Ports that are frequently accessed in a program and the compare register of the timer/event counter are mapped in this area, and these SFRs can be manipulated with a small number of bytes and clocks.

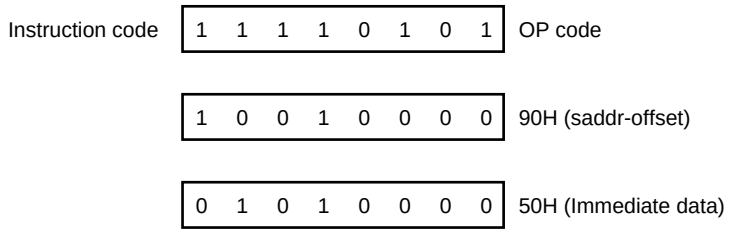
When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1. See **[Illustration]** below.

[Operand format]

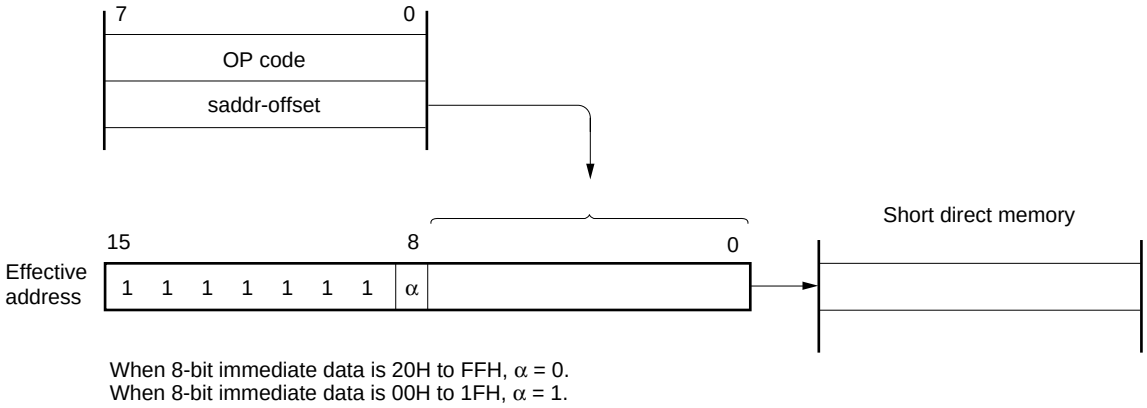
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[Description example]

MOV FE90H, #50H; When setting saddr to FE90H and the immediate data to 50H



[Illustration]



When 8-bit immediate data is 20H to FFH, $\alpha = 0$.
When 8-bit immediate data is 00H to 1FH, $\alpha = 1$.

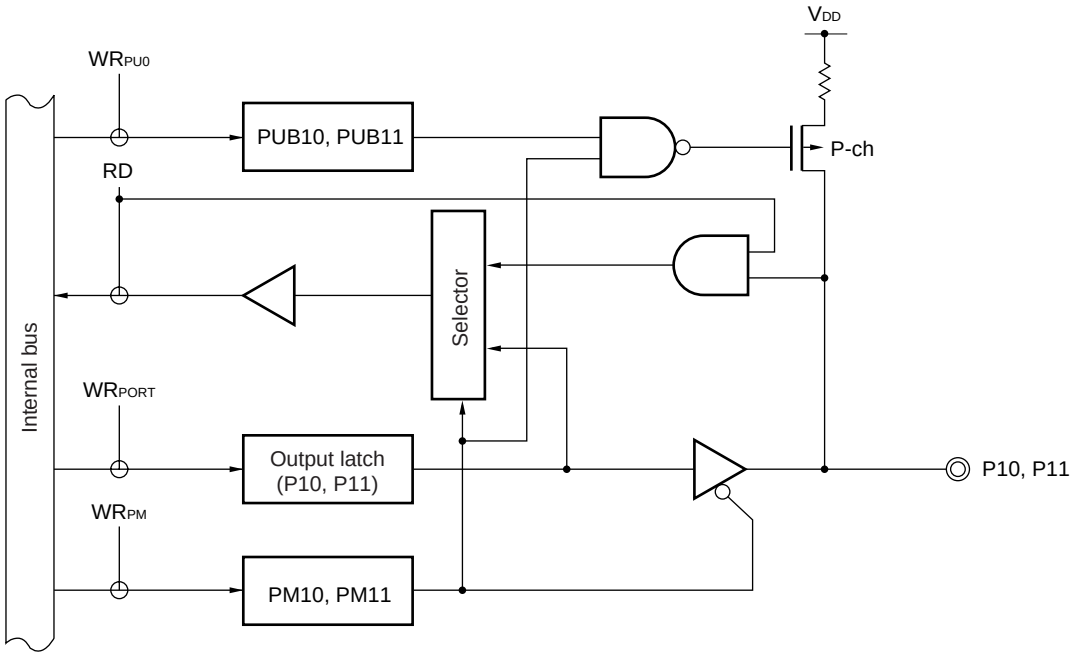
4.2.2 Port 1

This is a 2-bit I/O port with an output latch. Port 1 can be specified in the input or output mode in 1-bit units by using port mode register 1 (PM1). When using the P10 and P11 pins as input port pins, on-chip pull-up resistors can be connected in 1-bit units by using pull-up resistor option register B1 (PUB1).

$\overline{\text{RESET}}$ input sets this port to input mode.

Figure 4-3 shows a block diagram of port 1.

Figure 4-3. Block Diagram of P10 and P11



PUB1: Pull-up resistor option register B1
PM: Port mode register
RD: Port 1 read signal
WR: Port 1 write signal

6.4 16-Bit Timer 20 Operation

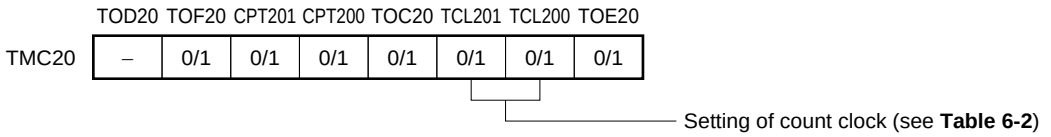
6.4.1 Operation as timer interrupt

16-bit timer 20 can generate interrupts repeatedly each time the free-running counter value reaches the value set to CR20. Since this counter is not cleared and holds the count even after an interrupt is generated, the interval time is equal to one cycle of the count clock set in TCL201 and TCL200.

To operate 16-bit timer 20 as a timer interrupt, the following settings are required.

- Set count values in CR20
- Set 16-bit timer mode control register 20 (TMC20) as shown in Figure 6-4.

Figure 6-4. Settings of 16-Bit Timer Mode Control Register 20 for Timer Interrupt Operation



Caution If both the CPT201 and CPT200 flags are set to 0, the capture edge operation is prohibited.

When the count value of 16-bit timer counter 20 (TM20) matches the value set in CR20, counting of TM20 continues and an interrupt request signal (INTTM20) is generated.

Table 6-2 shows interval time, and Figure 6-5 shows timing of timer interrupt operation.

Caution When rewriting the value in CR20 during a count operation, be sure to execute the following processing.

<1> Disable interrupts (set TMMK20 (bit 2 of interrupt mask flag register 1 (MK1)) to 1).
<2> Disable inversion control of timer output data (set TOC20 to 0)

If the value in CR20 is rewritten in the interrupt-enabled state, an interrupt request may occur at the moment of rewrite.

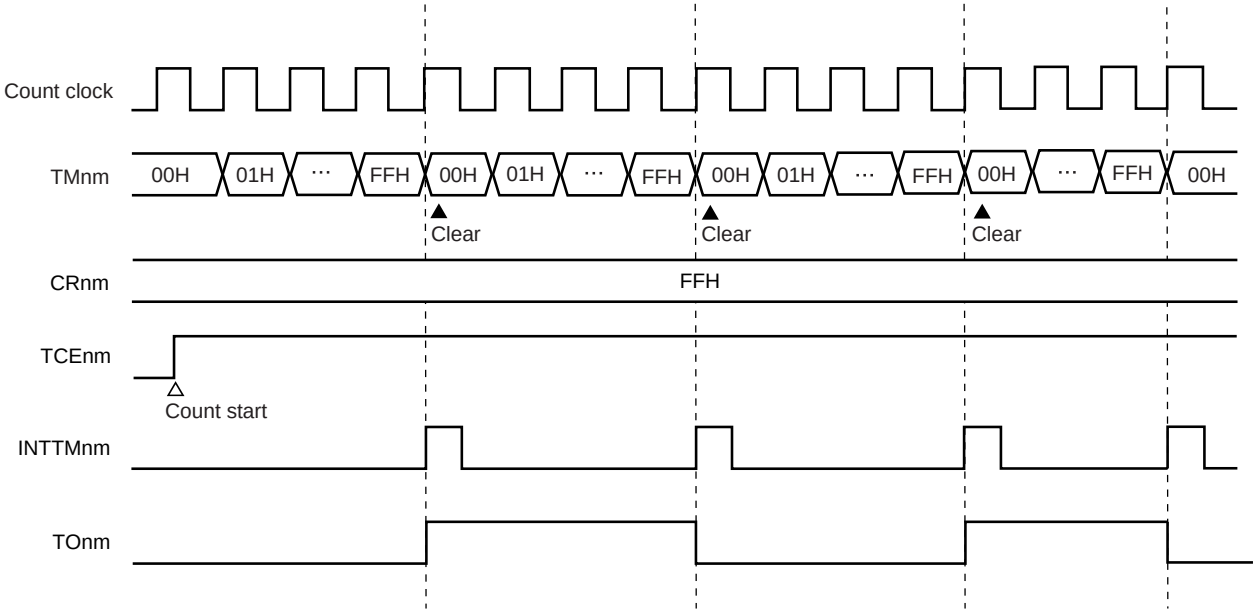
Table 6-2. Interval Time of 16-Bit Timer 20

TCL201	TCL200	Count Clock	Interval Time
0	0	Timer 61 interrupt signal	Cycle of timer 61 interrupt signal $\times 2^{16}$
0	1	$1/f_X$ (0.2 μs)	$2^{16}/f_X$ (13.1 ms)
1	0	$2^2/f_X$ (0.8 μs)	$2^{18}/f_X$ (52.4 ms)
1	1	$2^5/f_X$ (6.4 μs)	$2^{21}/f_{XT}$ (419 ms)

Remarks 1. f_X: Main system clock oscillation frequency

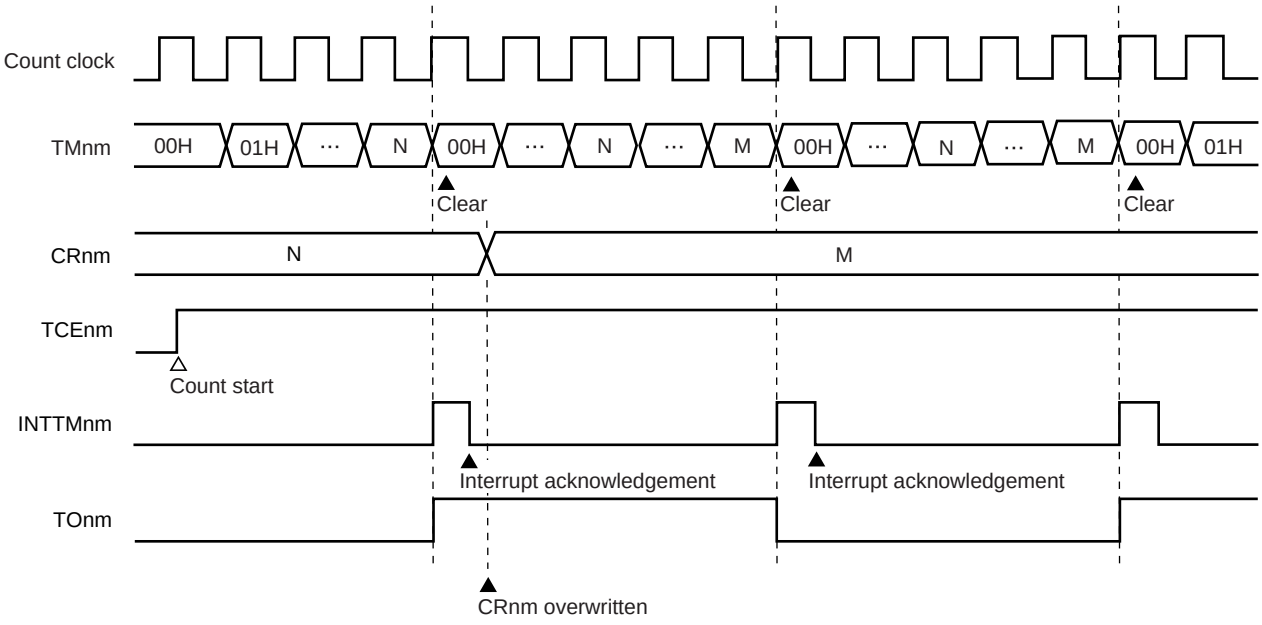
2. The parenthesized values apply to operation at f_X = 5.0 MHz.

★ **Figure 7-13. Timing of Interval Timer Operation with 8-Bit Resolution (When CRnm Is Set to FFH)**



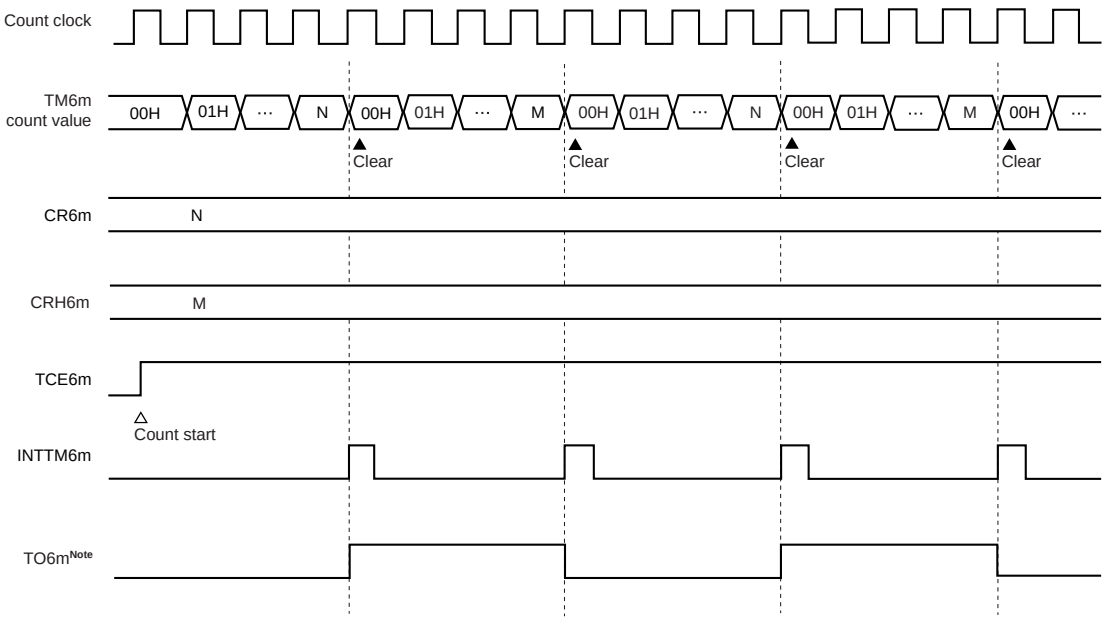
Remark nm = 50, 60, 61

Figure 7-14. Timing of Interval Timer Operation with 8-Bit Resolution (When CRnm Changes from N to M (N < M))



Remark 00H ≤ N < M ≤ FFH
nm = 50, 60, 61

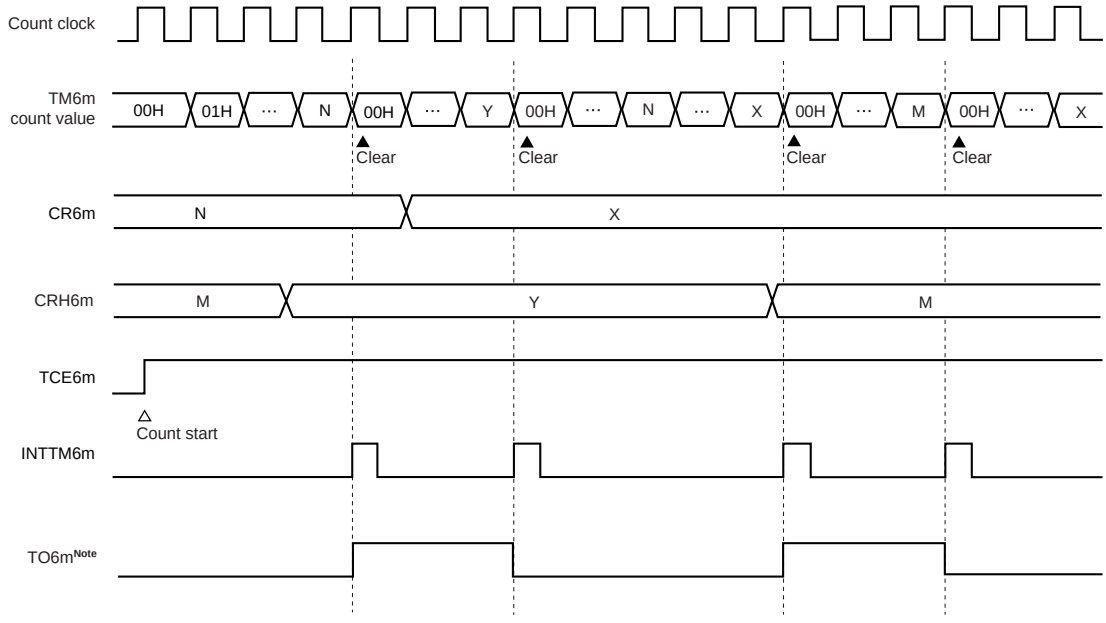
Figure 7-29. PPG Output Mode Timing (Basic Operation)



Note The initial value of TO6m is low level when output is enabled (TOE6m0 = 1).

Remark N, M = 00H to FFH
m = 0, 1

Figure 7-30. PPG Output Mode Timing (When CR6m and CRH6m Are Overwritten)



Note The initial value of TO6m is low level when output is enabled (TOE6m0 = 1).

Remark N, M, X, Y = 00H to FFH
m = 0, 1

8.4 Watch Timer Operation

8.4.1 Operation as watch timer

The main system clock (4.19 MHz) or subsystem clock (32.768 kHz) is used to enable the watch timer to operate at 0.5-second intervals.

Also, an interrupt request (INTWT) occurs at an interval of 1.0 seconds when using the 32.768 kHz subsystem clock via a setting in the watch timer interrupt time selection register (WTIM).

The watch timer is used to generate an interrupt request at specified intervals.

By setting bits 0 and 1 (WTM0 and WTM1) of the watch timer mode control register (WTM) to 1, the watch timer starts counting. By setting them to 0, the 5-bit counter is cleared and the watch timer stops counting.

It is possible to start the watch timer from zero seconds by clearing WTM1 to 0 when the interval timer and watch timer operate at the same time. In this case, however, an error of up to $2^9 \times 1/f_W$ seconds may occur in the overflow (INTWT) after the zero-second start of the watch timer because the 9-bit prescaler is not cleared to 0.

8.4.2 Operation as interval timer

The interval timer is used to repeatedly generate an interrupt request at the interval specified by a preset count value.

The interval can be selected by bits 4 to 6 (WTM4 to WTM6) of the watch timer mode control register (WTM).

Table 8-3. Interval Time of Interval Timer

WTM6	WTM5	WTM4	Interval Time	At f_X = 5.0 MHz	At f_X = 4.19 MHz	At f_{XT} = 32.768 kHz	At f_{XT} = 16.384 kHz
0	0	0	$2^4 \times 1/f_W$	409.6 μs	488 μs	488 μs	976 μs
0	0	1	$2^5 \times 1/f_W$	819.2 μs	977 μs	977 μs	1.95 ms
0	1	0	$2^6 \times 1/f_W$	1.64 ms	1.95 ms	1.95 ms	3.90 ms
0	1	1	$2^7 \times 1/f_W$	3.28 ms	3.91 ms	3.91 ms	7.82 ms
1	0	0	$2^8 \times 1/f_W$	6.55 ms	7.81 ms	7.81 ms	15.6 ms
1	0	1	$2^9 \times 1/f_W$	13.1 ms	15.6 ms	15.6 ms	31.2 ms
Other than above			Setting prohibited				

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. f_W: Watch timer clock frequency

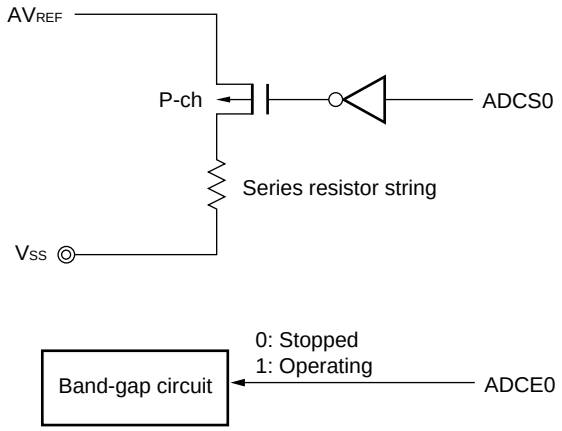
10.5 Cautions Related to 10-Bit A/D Converter

★ **(1) Current consumption in standby mode**

In standby mode, the A/D converter stops operation. Clearing bit 7 (ADCS0) and bit 0 (ADCE0) of A/D converter mode register 0 (ADML0) to 0 can reduce the current consumption.

Figure 10-7 shows how to reduce the current consumption in standby mode.

Figure 10-7. How to Reduce Current Consumption in Standby Mode



(2) Input range for pins ANI0 to ANI7

Be sure to keep the input voltage at ANI0 to ANI7 within the rating. If a voltage greater than or equal to AV_{DD} or less than or equal to AV_{SS} (even within the absolute maximum ratings) is input into a conversion channel, the conversion output of the channel becomes undefined, which may affect the conversion output of the other channels.

(3) Conflict

<1> Conflict between writing to A/D conversion result register 0 (ADCRL0) at the end of conversion and reading from ADCRL0 using instruction

Reading from ADCRL0 takes precedence. After reading, the new conversion result is written to ADCRL0.

<2> Conflict between writing to ADCRL0 at the end of conversion and writing to A/D converter mode register 0 (ADML0) or analog input channel specification register 0 (ADS0)

Writing to ADML0 or ADS0 takes precedence. ADCRL0 is not written to. No A/D conversion end interrupt request signal (INTAD0) is generated.

(4) Conversion result immediately after start of A/D conversion

If the band-gap circuit is not used (ADCE0 = 0) or conversion is started before 14 μs has elapsed following the setting of ADCE, only the first A/D conversion value immediately after A/D conversion has been started is undefined. Poll the A/D conversion end interrupt request (INTAD0), drop the first conversion result and use the second and subsequent conversion results. When 14 μs have elapsed following the activation of the band-gap circuit (ADCE0 = 1), the first conversion value is normal.

11.3 Serial Interface 20 Control Registers

Serial interface 20 is controlled by the following six registers.

- Serial operation mode register 20 (CSIM20)
- Asynchronous serial interface mode register 20 (ASIM20)
- Asynchronous serial interface status register 20 (ASIS20)
- Baud rate generator control register 20 (BRGC20)
- Port mode register 2 (PM2)
- Port 2

(1) Serial operation mode register 20 (CSIM20)

CSIM20 is used to make the settings related to 3-wire serial I/O mode.

CSIM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM20 to 00H.

Figure 11-3. Format of Serial Operation Mode Register 20

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM20	CSIE20	0	0	0	0	DIR20	CSCK20	0	FF72H	00H	R/W

CSIE20	3-wire serial I/O mode operation control
0	Operation disabled
1	Operation enabled

DIR20	First-bit specification
0	MSB
1	LSB

CSCK20	3-wire serial I/O mode clock selection
0	External clock input to the $\overline{\text{SCK20}}$ pin
1	Output of the dedicated baud rate generator

Cautions 1. Bits 0 and 3 to 6 must be set to 0.

2. CSIM20 must be cleared to 00H if UART mode is selected.

★ **3. When the external input clock is selected in 3-wire serial I/O mode, set input mode by setting bit 0 of port mode register 2 (PM2) to 1.**

4. Switch operating modes after halting the serial transmit/receive operation.

(b) Generation of UART baud rate transmit/receive clock from external clock input to ASCK20 pin

The transmit/receive clock is generated by dividing the clock input from the ASCK20 pin. The baud rate of a clock generated from the clock input to the ASCK20 pin is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{16} \text{ [bps]}$$

f_{ASCK}: Frequency of clock input to the ASCK20 pin

Table 11-4. Relationship Between ASCK20 Pin Input Frequency and Baud Rate (When BRGC20 Is Set to 80H)

Baud Rate (bps)	ASCK20 Pin Input Frequency (kHz)
75	1.2
150	2.4
300	4.8
600	9.6
1,200	19.2
2,400	38.4
4,800	76.8
9,600	153.6
19,200	307.2
31,250	500.0
38,400	614.4

(c) Generation of serial clock from system clock in 3-wire serial I/O

The serial clock is generated by dividing the system clock. The frequency of the serial clock can be obtained by the following expression. If the serial clock is externally input to the $\overline{SCK20}$ pin, it is unnecessary to set BRGC20.

$$\text{Serial clock frequency} = \frac{f_X}{2^{n+1}} \text{ [Hz]}$$

f_X: Main system clock oscillation frequency

n: Values in Figure 11-6 determined by the settings of TPS200 to TPS203 ($1 \leq n \leq 8$)

Table 11-5. Example of Relationship Between System Clock and Baud Rate

Baud Rate (bps)	n	BRGC20 Set Value	Error (%)	
			f_X = 5.0 MHz	f_X = 4.9152 MHz
1,200	8	70H	1.73	0
2,400	7	60H		
4,800	6	50H		
9,600	5	40H		
19,200	4	30H		
38,400	3	20H		
76,800	2	10H		

★ **Caution Do not select n = 1 during operation at f_X > 2.5 MHz because the resulting baud rate exceeds the rated range.**

(ii) Generation of baud rate transmit/receive clock from external clock input to ASCK20 pin

The transmit/receive clock is generated by dividing the clock input from the ASCK20 pin. The baud rate of a clock generated from the clock input to the ASCK20 pin is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{16} \text{ [bps]}$$

f_{ASCK}: Frequency of clock input to ASCK20 pin

Table 11-6. Relationship Between ASCK20 Pin Input Frequency and Baud Rate (When BRGC20 Is Set to 80H)

Baud Rate (bps)	ASCK20 Pin Input Frequency (kHz)
75	1.2
150	2.4
300	4.8
600	9.6
1,200	19.2
2,400	38.4
4,800	76.8
9,600	153.6
19,200	307.2
31,250	500.0
38,400	614.4

(e) Receive errors

The following three errors may occur during a receive operation: a parity error, framing error, and overrun error. After data reception, an error flag is set in asynchronous serial interface status register 20 (ASIS20). Receive error causes are shown in Table 11-7.

It is possible to determine what kind of error occurred during reception by reading the contents of ASIS20 in the reception error interrupt servicing (see **Table 11-7** and **Figure 11-10**).

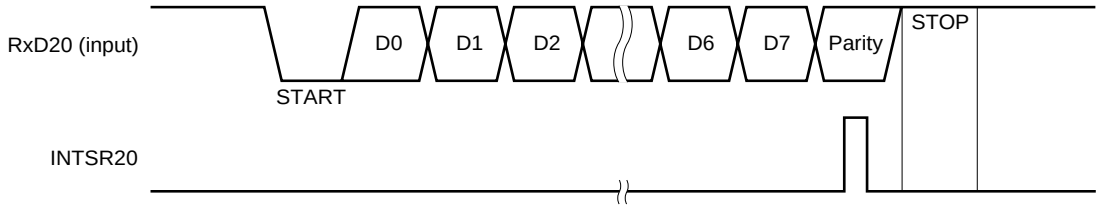
The contents of ASIS20 are reset (0) by reading receive buffer register 20 (RXB20) or receiving the next data (if there is an error in the next data, the corresponding error flag is set).

Table 11-7. Receive Error Causes

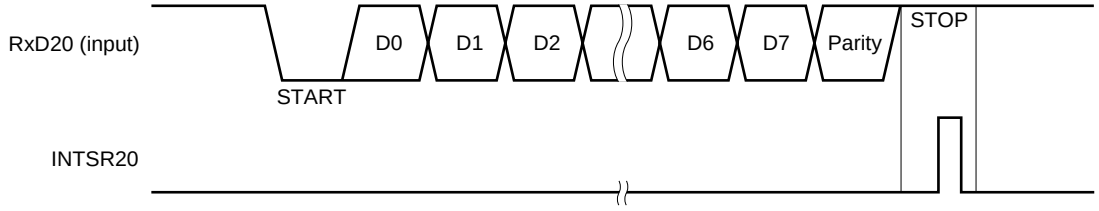
Receive Errors	Receive Errors	Value of ASIS20
Parity error	Parity at transmission and reception do not match	04H
Framing error	Stop bit not detected	02H
Overrun error	Reception of next data is completed before data is read from receive buffer register	01H

Figure 11-10. Receive Error Timing

(a) Parity error occurrence



(b) Framing error or overrun error occurrence

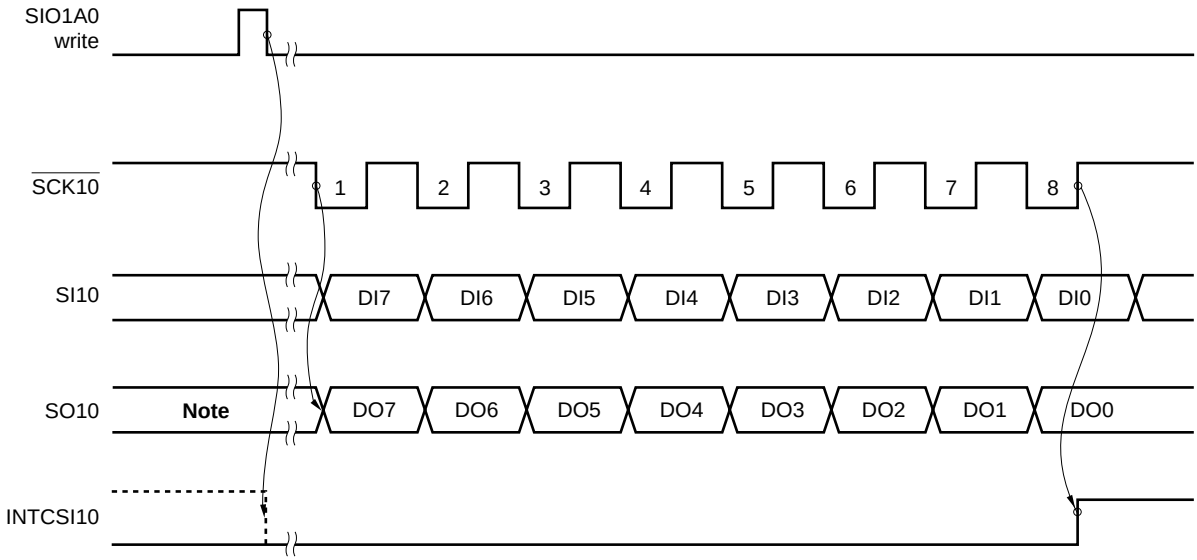


Cautions 1. The contents of the ASIS20 register are reset (0) by reading receive buffer register 20 (RXB20) or receiving the next data. To ascertain the error contents, read ASIS20 before reading RXB20.

2. Be sure to read receive buffer register 20 (RXB20) even if a receive error occurs. If RXB20 is not read, an overrun error will occur when the next data is received, and the receive error state will continue indefinitely.

Figure 12-5. 3-Wire Serial I/O Mode Timing (2/2)

(ii) Slave operation timing



Note The value of the last bit previously output is output.

(2) Automatic transmit/receive data setting

(a) Transmit data setting

<1> Write transmit data from the least significant address FFA0H of buffer RAM (up to FFAFH). The transmit data should be in the order from higher address to lower address.
<2> Set the value obtained by subtracting 1 from the number of transmit data bytes to automatic data transmit/receive address pointer 0 (ADTP0).

(b) Automatic transmit/receive mode setting

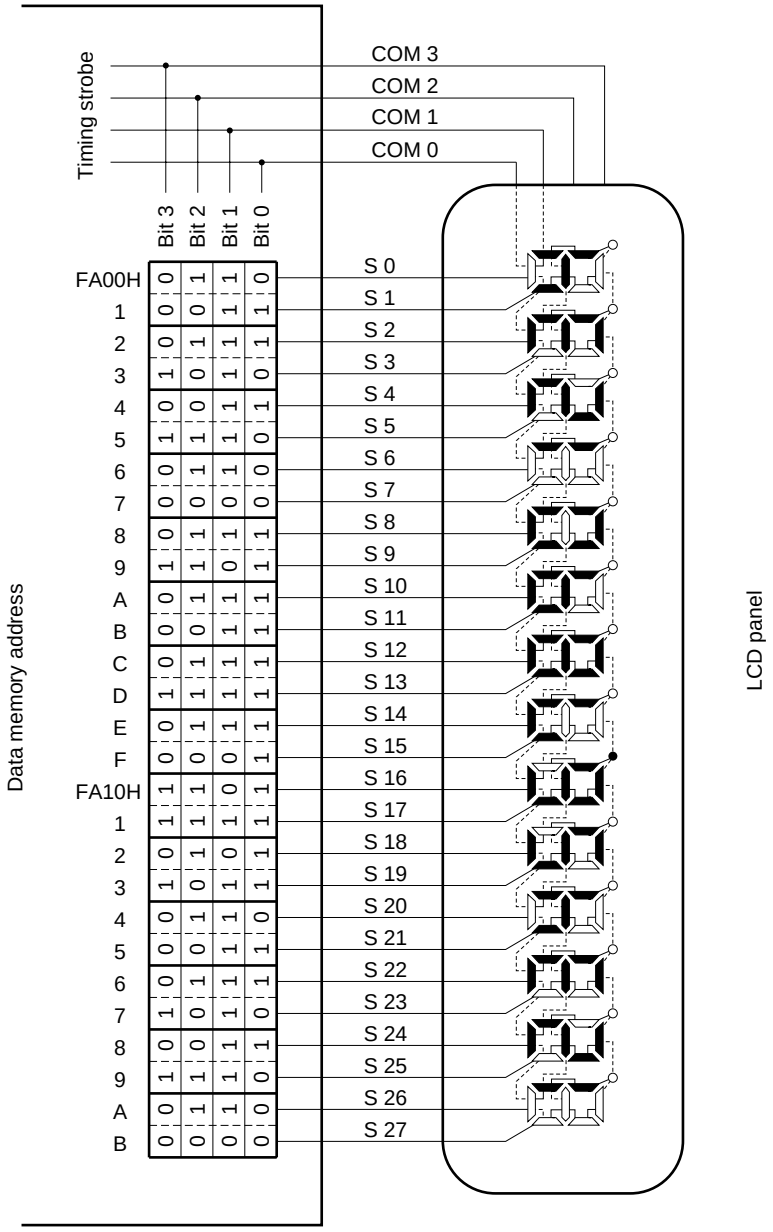
<1> Set bit 7 (CSIE10) and bit 5 (ATE0) of serial operation mode register 1A0 (CSIM1A0) to 1.
<2> Set bit 7 (RE0) of automatic data transmit/receive control register 0 (ADTC0) to 1.
<3> Set the data transmit/receive interval in automatic data transmit/receive interval specification register 0 (ADTI0).
<4> Write any value to serial I/O shift register 1A0 (SIO1A0) (transfer start trigger).

Caution Writing any value to SIO1A0 orders the start of automatic transmission/reception operation; the written value has no meaning.

The following operations are automatically carried out when (a) and (b) are carried out.

- After the buffer RAM data specified by ADTP0 is transferred to SIO1A0, transmission is carried out (start of automatic transmission/reception).
- The received data is written to the buffer RAM address specified by ADTP0.
- ADTP0 is decremented and the next data transmission/reception is carried out. Data transmission/reception continues until the ADTP0 decremental output becomes 00H and address FFA0H data is output (end of automatic transmission/reception).
- When automatic transmission/reception is terminated, bit 3 (TRF0) of ADTC0 is cleared to 0.

Figure 13-13. Example of Connecting Four-Time-Slice LCD Panel



(8) Remote controller receive DH0S compare register (RMDH0S)

This register is used to detect the high level of remote controller data 0 (short side).

RMDH0S is set with an 8-bit memory manipulation instruction.

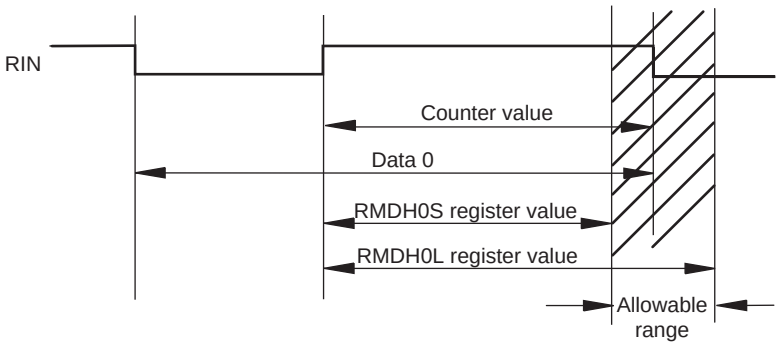
$\overline{\text{RESET}}$ input sets RMDH0S to 00H.

(9) Remote controller receive DH0L compare register (RMDH0L)

This register is used to detect the high level of remote controller data 0 (long side).

RMDH0L is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets RMDH0L to 00H.



If RMDH0S ≤ counter value < RMDH0L is satisfied, it is assumed that the high level of data 0 has been successfully received, and therefore RMSR receives the data.

(10) Remote controller receive DH1S compare register (RMDH1S)

This register is used to detect the high level of remote controller data 1 (short side).

RMDH1S is set with an 8-bit memory manipulation instruction.

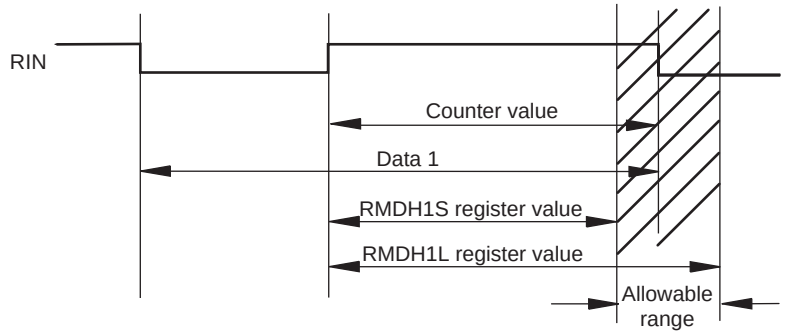
$\overline{\text{RESET}}$ input sets RMDH1S to 00H.

(11) Remote controller receive DH1L compare register (RMDH1L)

This register is used to detect the high level of remote controller data 1 (long side).

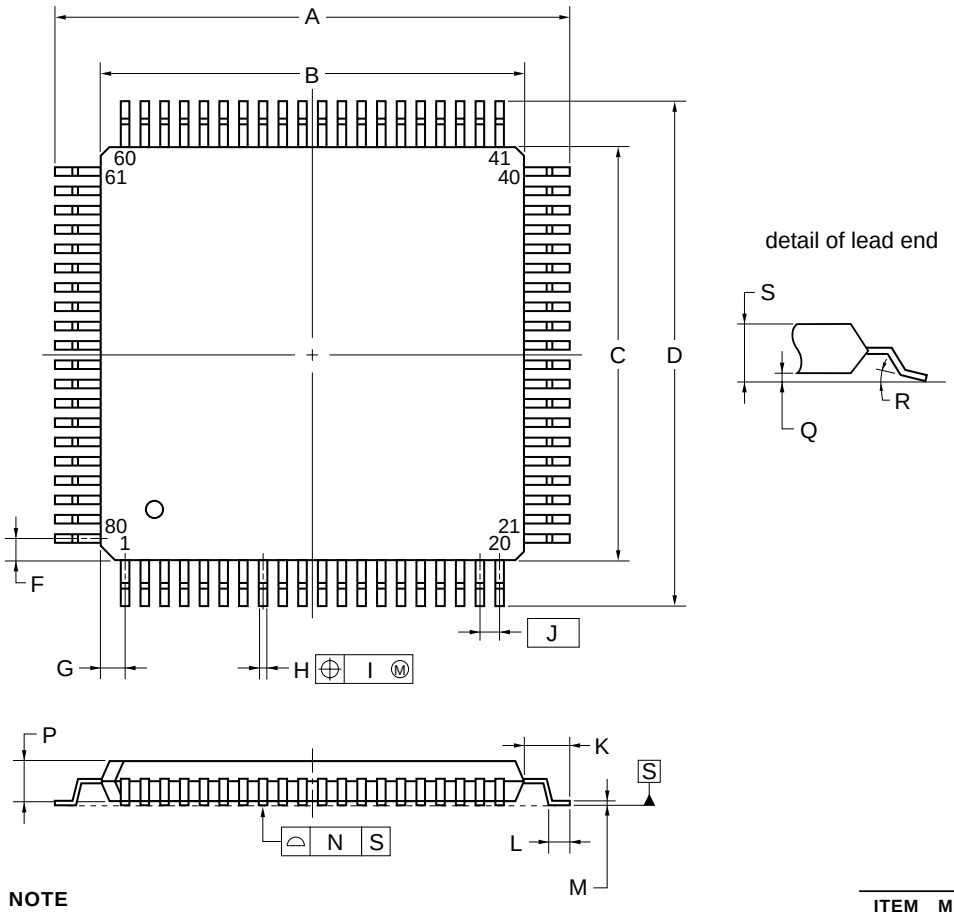
RMDH1L is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets RMDH1L to 00H.



If RMDH1S ≤ counter value < RMDH1L is satisfied, it is assumed that the high level of data 1 has been successfully received, and therefore RMSR receives the data.

80-PIN PLASTIC QFP (14x14)



NOTE

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	17.20±0.20
B	14.00±0.20
C	14.00±0.20
D	17.20±0.20
F	0.825
G	0.825
H	0.32±0.06
I	0.13
J	0.65 (T.P.)
K	1.60±0.20
L	0.80±0.20
M	$0.17^{+0.03}_{-0.07}$
N	0.10
P	1.40±0.10
Q	0.125±0.075
R	$3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$
S	1.70 MAX.

P80GC-65-8BT-1

(4/4)

Edition	Major Revision from Previous Edition	Applied to:
4th	Modification of descriptions in **Figure 12-4. Format of Automatic Data Transmit/Receive Interval Specification Register 0**	**CHAPTER 12 SERIAL INTERFACE 1A0**
	Addition of formal specifications of μPD789489 and 78F9489 to μPD789489, 78F9489	**CHAPTER 22 ELECTRICAL SPECIFICATIONS (μPD789488, 78F9488, 789489, 78F9489)**
	Addition of recommended conditions for μPD789489 and 78F9489	**CHAPTER 25 RECOMMENDED SOLDERING CONDITIONS**
4th (Modified version)	Addition of the lead-free products	Throughout
	Modification of descriptions of the voltage boost wait time	**CHAPTER 13 LCD CONTROLLER/DRIVER**
	Modification of **Figure 19-9. Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O with Handshake**	**CHAPTER 19 FLASH MEMORY VERSION**