Renesas Electronics America Inc - UPD78F0078GK-9ET-A 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	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/upd78f0078gk-9et-a

EEPROM, FIP, and IEBus are trademarks of NEC Electronics Corporation.

Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

PC/AT is a trademark of International Business Machines Corporation.

HP9000 Series 700 and HP-UX are trademarks of Hewlett-Packard Company.

SPARCstation is a trademark of SPARC International, Inc.

Solaris and SunOS are trademarks of Sun Microsystems, Inc.

These commodities, technology or software, must be exported in accordance with the export administration regulations of the exporting country. Diversion contrary to the law of that country is prohibited.

The major functional differences between the subseries are shown below.

- **Subseries without the suffix Y**

Subseries Name \ Function		ROM Capacity	Timer 8-Bit	Timer 16-Bit	Timer Watch	Timer WDT	8-Bit A/D	10-Bit A/D	8-Bit D/A	Serial Interface	I/O	V_{DD} MIN. Value	External Expansion
Control	μPD78075B	32 KB to 40 KB	4 ch	1 ch	1 ch	1 ch	8 ch	–	2 ch	3 ch (UART: 1 ch)	88	1.8 V	Yes
	μPD78078	48 KB to 60 KB											
	μPD78070A	–									61	2.7 V	
	μPD780058	24 KB to 60 KB	2 ch							3 ch (time-division UART: 1 ch)	68	1.8 V	
	μPD78058F	48 KB to 60 KB								3 ch (UART: 1 ch)	69	2.7 V	
	μPD78054	16 KB to 60 KB										2.0 V	
	μPD780065	40 KB to 48 KB							–	4 ch (UART: 1 ch)	60	2.7 V	
	μPD780078	48 KB to 60 KB		2 ch			–	8 ch		3 ch (UART: 2 ch)	52	1.8 V	
	μPD780034A	8 KB to 32 KB		1 ch						3 ch (UART: 1 ch)	51		
	μPD780024A						8 ch	–					
	μPD780034AS						–	4 ch			39		–
	μPD780024AS						4 ch	–					
	μPD78014H						8 ch			2 ch	53		Yes
	μPD78018F	8 KB to 60 KB											
	μPD78083	8 KB to 16 KB		–	–					1 ch (UART: 1 ch)	33		–
Inverter control	μPD780988	16 KB to 60 KB	3 ch	**Note**	–	1 ch	–	8 ch	–	3 ch (UART: 2 ch)	47	4.0 V	Yes
VFD drive	μPD780208	32 KB to 60 KB	2 ch	1 ch	1 ch	1 ch	8 ch	–	–	2 ch	74	2.7 V	–
	μPD780232	16 KB to 24 KB	3 ch	–	–		4 ch				40	4.5 V	
	μPD78044H	32 KB to 48 KB	2 ch	1 ch	1 ch		8 ch			1 ch	68	2.7 V	
	μPD78044F	16 KB to 40 KB								2 ch			
LCD drive	μPD780354	24 KB to 32 KB	4 ch	1 ch	1 ch	1 ch	–	8 ch	–	3 ch (UART: 1 ch)	66	1.8 V	–
	μPD780344						8 ch	–					
	μPD780338	48 KB to 60 KB	3 ch	2 ch			–	10 ch	1 ch	2 ch (UART: 1 ch)	54		
	μPD780328										62		
	μPD780318										70		
	μPD780308	48 KB to 60 KB	2 ch	1 ch			8 ch	–	–	3 ch (time-division UART: 1 ch)	57	2.0 V	
	μPD78064B	32 KB								2 ch (UART: 1 ch)			
	μPD78064	16 KB to 32 KB											
Bus interface supported	μPD780948	60 KB	2 ch	2 ch	1 ch	1 ch	8 ch	–	–	3 ch (UART: 1 ch)	79	4.0 V	Yes
	μPD78098B	40 KB to 60 KB		1 ch					2 ch		69	2.7 V	–
	μPD780816	32 KB to 60 KB		2 ch			12 ch		–	2 ch (UART: 1 ch)	46	4.0 V	
Meter control	μPD780958	48 KB to 60 KB	4 ch	2 ch	–	1 ch	–	–	–	2 ch (UART: 1 ch)	69	2.2 V	–
Dashboard control	μPD780852	32 KB to 40 KB	3 ch	1 ch	1 ch	1 ch	5 ch	–	–	3 ch (UART: 1 ch)	56	4.0 V	–
	μPD780828B	32 KB to 60 KB									59		

Note 16-bit timer: 2 channels
10-bit timer: 1 channel

4.2 Description of Pin Functions

4.2.1 P00 to P03 (Port 0)

P00 to P03 function as a 4-bit I/O port. Besides serving as an I/O port, they also function as external interrupt inputs and an A/D converter external trigger input.

The following operating modes can be specified in 1-bit units.

(1) Port mode

These pins function as a 4-bit I/O port.

P00 to P03 can be specified as input or output in 1-bit units using port mode register 0 (PM0). On-chip pull-up resistors can be connected by setting pull-up resistor option register 0 (PU0).

(2) Control mode

In this mode, these pins function as external interrupt request inputs and an A/D converter external trigger input.

(a) INTP0 to INTP3

INTP0 to INTP3 are external interrupt request input pins for which the valid edge can be specified (rising edge, falling edge, or both rising and falling edges).

(b) ADTRG

A/D converter external trigger input pin.

Caution When P03 is used as an A/D converter external trigger input, specify the valid edge by using bits 1 and 2 (EGA00, EGA01) of the A/D converter mode register (ADM0) and set the interrupt mask flag (PMK3) to 1.

4.2.2 P10 to P17 (Port 1)

P10 to P17 function as an 8-bit input-only port. Besides serving as an input port, they also function as A/D converter analog inputs.

The following operating modes can be specified in 1-bit units.

(1) Port mode

These pins function as an 8-bit input-only port.

(2) Control mode

These pins function as A/D converter analog input pins (ANI0 to ANI7). When using these pins as analog input pins, see **(4) ANI0/P10 to ANI7/P17** in **13.6 Cautions for A/D Converter**.

4.2.3 P20 to P25 (Port 2)

P20 to P25 function as a 6-bit I/O port. Besides serving as an I/O port, they function as data I/O and clock I/O for serial interfaces CSI1 and UART0.

The following operating modes can be specified in 1-bit units.

(1) Port mode

These pins function as a 6-bit I/O port. They can be specified as input or output in 1-bit units using port mode register 2 (PM2). On-chip pull-up resistors can be connected by setting pull-up resistor option register 2 (PU2).

Table 4-1. Pin I/O Circuit Types (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P50/A8 to P57/A15	5-H	I/O	Input: Independently connect to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor. Output: Leave open.
P64/$\overline{RD}$			
P65/$\overline{WR}$			
P66/$\overline{WAIT}$			
P67/ASTB			
P70/TI000/TO00	8-C		
P71/TI010			
P72/TI50/TO50			
P73/TI51/TO51			
P74/TI011/PCL			
P75/TI001/TO01/BUZ			
P80/$\overline{SS1}$			Input: Connect to V_{SS0} or V_{SS1} via a resistor. Output: Leave open.
<R> $\overline{RESET}$	2	Input	Connect to V_{DD}.
XT1	16		Connect directly to V_{DD0} or V_{DD1}.
XT2		—	Leave open.
AV_{REF}	—		Connect directly to V_{SS0} or V_{SS1}.
AV_{SS}			
IC (for mask ROM version)			
V_{PP} (for flash memory version)			Independently connect a 10 kΩ pull-down resistor or connect directly to V_{SS0} or V_{SS1}.

Table 5-3. Special Function Register List (3/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FF94H	Asynchronous serial interface status register 2	ASIS2		R	—	√	—	00H
FF95H	Asynchronous serial interface transmit status register 2	ASIF2			—	√	—	
FFA0H	Asynchronous serial interface mode register 0	ASIM0		R/W	√	√	—	
FFA1H	Asynchronous serial interface status register 0	ASIS0		R	—	√	—	
FFA2H	Baud rate generator control register 0	BRGC0		R/W	—	√	—	
FFA8H	IIC control register 0Note 1	IICC0			√	√	—	
FFA9H	IIC status register 0Note 1	IICS0		R	√	√	—	
FFAAH	IIC transfer clock select register 0Note 1	IICCL0		R/W	√	√	—	
FFABH	Slave address register 0Note 1	SVA0			—	√	—	
FFB0H	Serial operation mode register 1	CSIM1			√	√	—	
FFB1H	Serial clock select register 1	CSIC1			√	√	—	10H
FFB8H	Serial operation mode register 3	CSIM3			√	√	—	00H
FFE0H	Interrupt request flag register 0L	IF0	IF0L		√	√	√	
FFE1H	Interrupt request flag register 0H		IF0H		√	√		
FFE2H	Interrupt request flag register 1L	IF1L			√	√	—	
FFE4H	Interrupt mask flag register 0L	MK0	MK0L		√	√	√	FFH
FFE5H	Interrupt mask flag register 0H		MK0H		√	√		
FFE6H	Interrupt mask flag register 1L	MK1L			√	√	—	
FFE8H	Priority level specification flag register 0L	PR0	PR0L		√	√	√	
FFE9H	Priority level specification flag register 0H		PR0H		√	√		
FFEAH	Priority level specification flag register 1L	PR1L			√	√	—	
FFF0H	Memory size switching register	IMS			—	√	—	CFHNote 2
FFF4H	Internal expansion RAM size switching register	IXS			—	√	—	0CHNote 3
FFF8H	Memory expansion wait setting register	MM			√	√	—	10H
FFF9H	Watchdog timer mode register	WDTM			√	√	—	00H
FFFAH	Oscillation stabilization time select register	OSTS			—	√	—	04H
FFFBH	Processor clock control register	PCC			√	√	—	

Notes 1. μPD780078Y Subseries only

2. Although the default value of this register is CFH, set the value corresponding to each product as indicated below.

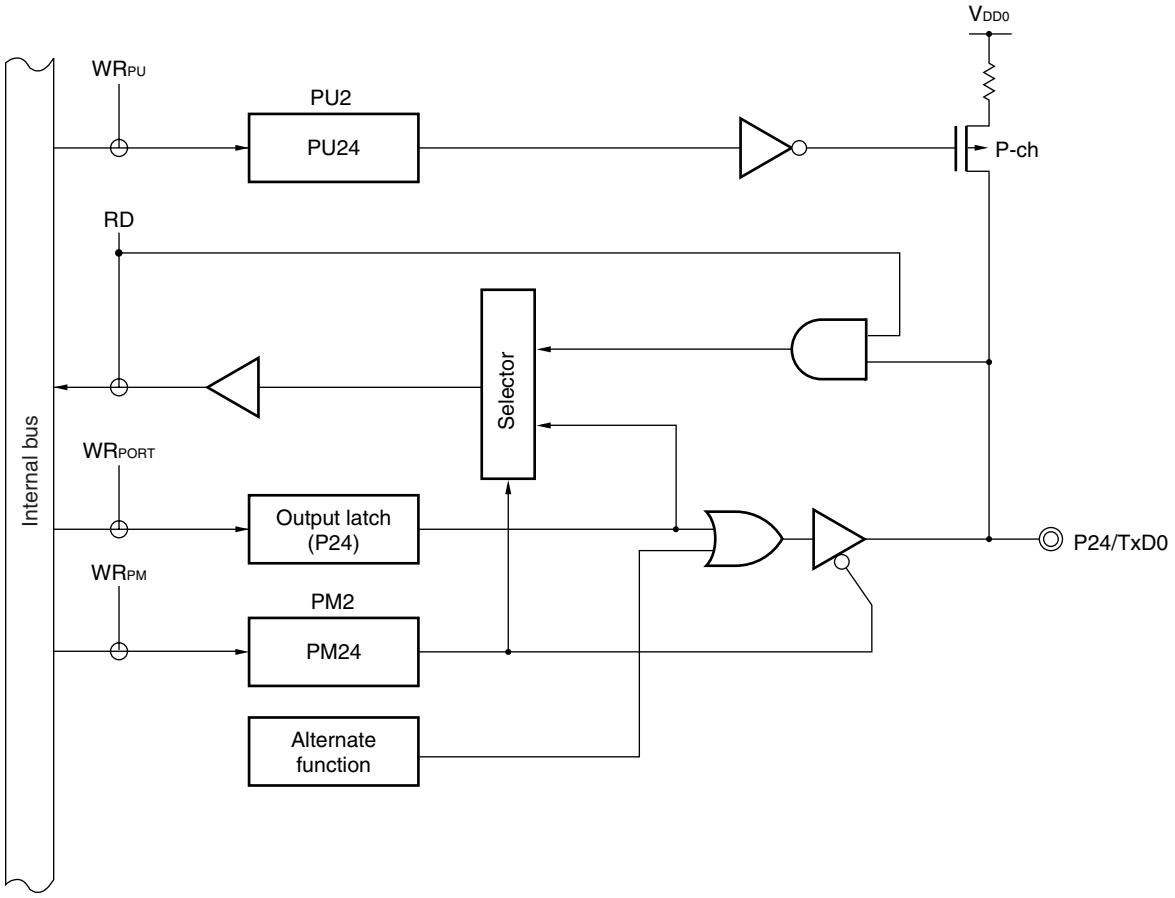
μPD780076, 780076Y: CCH

μPD780078, 780078Y: CFH

μPD78F0078, 78F0078Y: Value for mask ROM version

3. Although the default value of this register is 0CH, initialize this register to 0AH.

Figure 6-7. Block Diagram of P24



PU2: Pull-up resistor option register 2
PM2: Port mode register 2
RD: Read signal
WR××: Write signal

6.2.6 Port 4

Port 4 is an 8-bit I/O port with an output latch. Port 4 can be set to the input or output mode in 1-bit units using port mode register 4 (PM4). An on-chip pull-up resistor can be connected to P40 to P47 in 1-bit units using pull-up resistor option register 4 (PU4).

The interrupt request flag (KRIF) can be set to 1 by detecting falling edges.

This port can also be used as an address/data bus in external memory expansion mode.

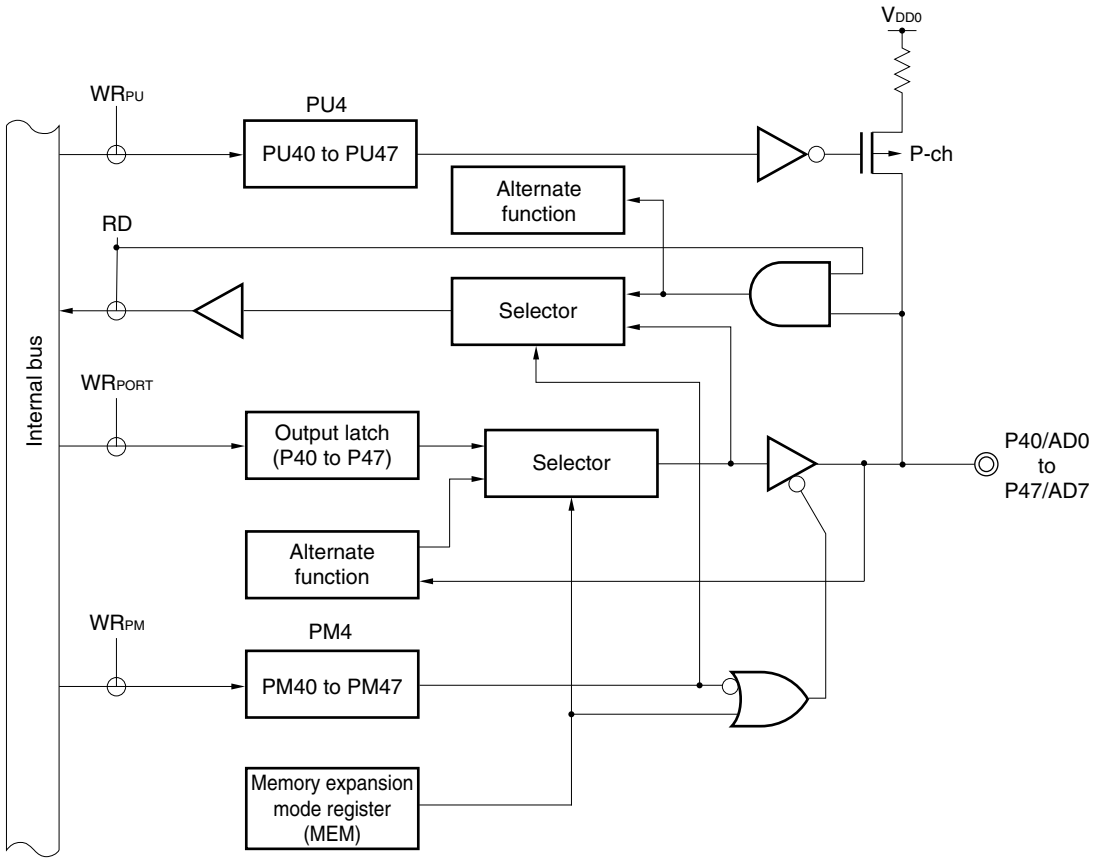
$\overline{\text{RESET}}$ input sets port 4 to input mode.

Figures 6-14 and 6-15 show a block diagram of port 4 and a block diagram of the falling edge detector, respectively.

Cautions 1. An on-chip pull-up resistor is not disconnected even if the external memory expansion mode is set when PU4n = 1 (n = 0 to 7).

2. When using the falling edge detection interrupt (INTKR), be sure to set the memory expansion mode register (MEM) to 01H.

Figure 6-14. Block Diagram of P40 to P47



PU4: Pull-up resistor option register 4

PM4: Port mode register 4

RD: Read signal

WR××: Write signal

Figure 8-13. Format of Prescaler Mode Register 01 (PRM01)

Address: FF65H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM01	ES111	ES101	ES011	ES001	0	0	PRM011	PRM001

ES111	ES101	TI011 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES011	ES001	TI001 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM011	PRM001	Count clock selection		
			f_X = 8.38 MHz	f_X = 12 MHz[Note 1]
0	0	$f_X/2$	4.19 MHz	6 MHz
0	1	$f_X/2^3$	1.04 MHz	1.5 MHz
1	0	$f_X/2^9$	16.36 kHz	23.43 kHz
1	1	TI001 pin valid edge[Notes 2, 3]		

Notes 1. Expanded-specification products of μPD780078 Subseries only.

2. The external clock requires a pulse two cycles longer than internal clock ($f_X/2^3$).

3. When the valid edge of the TI001 pin is selected, the main system clock is used as the sampling clock for noise elimination. The valid edge of the TI001 pin can be used only when the main system clock is operating.

Cautions 1. Always set data to PRM01 after stopping the timer operation.

2. If the valid edge of the TI001 pin is to be set as the count clock, do not set the clear & start mode and the capture trigger at the valid edge of the TI001 pin.

3. When P75 is used as the valid edge of the TI001 pin, it cannot be used as the timer output (TO01 pin), and when used as the TO01 pin, it cannot be used as the valid edge of the TI001 pin.

4. If the TI001 or TI011 pin is high level immediately after system reset, the rising edge is immediately detected after the rising edge or both the rising and falling edge are set as the valid edge(s) of the TI001 pin or TI011 pin to enable the operation of 16-bit timer counter 01 (TM01). Be careful when pulling up the TI001 pin or the TI011 pin. However, the rising edge is not detected if the TI001 pin or the TI011 pin is high level at restart after the operation has been stopped.

<R>

Remarks 1. f_X: Main system clock oscillation frequency

2. TI001 or TI011 pin: 16-bit timer/event counter 01 input pin

(5) Port mode register 7 (PM7)

This register sets port 7 input/output in 1-bit units.

When using the P70/TO00/TI000 and P75/TO01/TI001/BUZ pins for timer output, set PM70 and PM75, and the output latches of P70 and P75 to 0.

When using the P70/TO00/TI000 and P75/TO01/TI001/BUZ pins for timer input, set PM70 and PM75 to 1. At this time, the output latches of P70 and P75 can be either 0 or 1.

PM7 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM7 to FFH.

Figure 8-14. Format of Port Mode Register 7 (PM7)

Address: FF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	P7n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

8.5.2 Pulse width measurement by free-running counter and one capture register

```
/************************************************************************/
/*                                                                      */
/*           Timer 00 operation sample                                  */
/*       Pulse width measurement example by free-running and CR010      */
/*        Measurement results to be up to 16 bits and not checked for errors  */
/*        data[0]: End flag                                             */
/*        data[1]: Measurement results (pulse width)                    */
/*        data[2]: Previous read value                                  */
/*                                                                      */
/************************************************************************/
#pragma sfr
#pragma EI
#pragma DI
#pragma interrupt INTTM010 intervalint rb2
        unsigned int data[3];           /* Data area */

void main(void)
{
        unsigned int length;
        PCC = 0x0;                      /* Set high-speed operation mode */
        data[0] = 0;
        data[1] = 0;
        data[2] = 0;
                                        /* Set port */
        PM7.0 = 1;                      /* Set P70 as input */
                                        /* Set interrupt */
        TMMK010 = 0;                    /* Cancel INTTM010 interrupt mask */
                                        /* Set timer 00 */
        PRM00 = 0b00110010;             /* Both rising and falling edges for TI000 */
                                        /* Count clock is fx/2^6              */
        CRC00 = 0b00000100;             /* Set CR010 to capture register */
        TMC00 = 0b00000100;             /* Start in free-run mode */
        EI();
        while(1){                       /* Dummy loop */
                while(data[0] == 0);    /* Wait for measurement completion */
                DI();                   /* Disable interrupt for exclusive operation */
                length = data[1];       /* Read measurement results */
                data[0] = 0;            /* Clear end flag */
                EI();                   /* Exclusive operation completed */
        }
}

/* Timer 00 interrupt function */
void intervalint()
{
        unsigned int work;
/*******************************************************/
/*                                                     */
/* Define variables required for interrupt here        */
/*                                                     */
/*******************************************************/
        work = CR010;                   /* Read capture value */
        data[1] = work - data[2];       /* Calculate and update interval */
        data[2] = work;                 /* Update read value */
        data[0] = 0xffff;               /* Set measurement completion flag*/

/*********************************************************/
/*                                                       */
/* Describe processing required for interrupt below      */
/*                                                       */
/*********************************************************/
}
```

8.5.4 Pulse width measurement by restart

```
/************************************************************************/
/*                                                                      */
/*           Timer 00 operation sample                                  */
/*       Pulse width measurement example by restart                     */
/*       Measurement results up to 16 bits, not to be checked for errors  */
/*        data[0]: End flag                                             */
/*        data[1]: Measurement results (pulse width)                    */
/*        data[2]: Previous read value                                  */
/*                                                                      */
/************************************************************************/
#pragma sfr
#pragma EI
#pragma DI
#pragma interrupt INTTM010 intervalint rb2
        unsigned int data[3];           /* Data area */

void main(void)
{
        unsigned int length;
        PCC = 0x0;                      /* Set high-speed operation mode */
        data[0] = 0;
        data[1] = 0;
        data[2] = 0;
                                        /* Set port */
        PM7.0 = 1;                      /* Set P70 as input */
                                        /* Set interrupt */
        TMMK010 = 0;                    /* Cancel INTTM010 interrupt mask */
                                        /* Set timer 00 */
        PRM00 = 0b00110010;             /* Both rising and falling edges  */
                                        /* Count clock is fx/2^6        */
        CRC00 = 0b00000100;             /* Set CR010 to capture register   */
        TMC00 = 0b00001000;             /* Clear & start at TI000 valid edge */
        EI();
        while(1){                       /* Dummy loop */
                if(data[0] != 0)        /* Wait for TI000 measurement completion */
                {
                        TMMK010 = 1;            /* Disable INTTM010 for exclusive
                                                   operation */
                        length = data[1]+data[2];      /* Cycle calculation based on
                                                          measurement results */
                        data[0] = 0;            /* Clear end flag */
                        TMMK010 = 0;            /* Exclusive operation completed */
                }
        }
}

/* Timer00 interrupt function */
void intervalint()
{
/********************************************************/
/*                                                      */
/* Define variables required for interrupt here         */
/*                                                      */
/********************************************************/
        data[2] = data[1];              /* Update old data */
        data[1] = CR010;                /* Update read value */
        data[0] = 0xffff;               /* Set measurement completion flag*/

/********************************************************/
/*                                                      */
/* Describe processing required for interrupt below     */
/*                                                      */
/********************************************************/
}
```

13.4 A/D Converter Operation

13.4.1 Basic operations of A/D converter

<1> Select one channel for A/D conversion using analog input channel specification register 0 (ADS0).
<2> Set bit 0 (ADCE0) of A/D converter mode register 0 (ADM0) to 1 and wait for 14 μs or longer.
<3> Set bit 7 (ADCS0) of the ADM0 register to 1 to start the A/D conversion operation.
(<4> to <10> are operations performed by hardware)
<4> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.
<5> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the input analog voltage is held until the A/D conversion operation is finished.
<6> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF} by the tap selector.
<7> The voltage difference between the series resistor string voltage tap and analog input is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB of SAR remains set. If the analog input is smaller than (1/2) AV_{REF}, the MSB is reset.
<8> Next, bit 8 of SAR is automatically set, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.
- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

The voltage tap and analog input voltage are compared and bit 8 of SAR is manipulated as follows.
- Analog input voltage ≥ Voltage tap: Bit 8 = 1
- Analog input voltage < Voltage tap: Bit 8 = 0

<9> Comparison is continued in this way up to bit 0 of SAR.
<10> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to and latched in A/D conversion result register 0 (ADCR0).
At the same time, the A/D conversion end interrupt request (INTAD0) can also be generated.
<11> Repeat steps <4> to <10>, until ADCS0 is cleared to 0.
To stop the A/D converter, clear ADCS0 to 0.
To restart A/D conversion from the status of ADCE0 = 1, start from <3>. To restart A/D conversion from the status of ADCE0 = 0, however, start from <2>.

Cautions 1. If bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) is set to 1 without setting bit 0 (ADCE0) to 1, the first A/D conversion value immediately after A/D conversion has been started may not satisfy the rated value. Take measures such as polling the A/D conversion end interrupt request (INTAD0) and removing the first conversion results.
The same may apply if ADCS0 is set to 1 without the lapse of a wait time of 14 μs (MIN.) after ADCE0 has been set to 1. Make sure that the specified wait time elapses.
2. The A/D converter stops operation in standby mode.

Figure 16-2. Format of Serial Operation Mode Register 3 (CSIM3)

Address: FFB8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM3	CSIE3	0	0	0	0	MODE3	SCL31	SCL30

CSIE3	Enable/disable specification for SIO3		
	Shift register operation	Serial counter	Port
0	Operation disabled	Clear	Port function**Note 1**
1	Operation enabled	Count operation enabled	Serial function + port function**Note 2**

MODE3	Transfer operation modes and flags		
	Operation mode	Transfer start trigger	SO3/P35/RxD2 pin function
0	Transmit/transmit and receive mode	Write to SIO3	SO3
1	Receive-only mode	Read from SIO3	P35**Note 3**

SCL31	SCL30	Clock selection		
			f_X = 8.38 MHz	f_X = 12 MHz**Note 4**
0	0	External clock input to $\overline{SCK3}$	–	–
0	1	$f_X/2^3$	1.04 MHz	1.50 MHz
1	0	$f_X/2^4$	523 kHz	750 kHz
1	1	$f_X/2^5$	261 kHz	375 kHz

Notes 1. When CSIE3 = 0 (SIO3 operation stopped status), the SI3, SO3, and $\overline{SCK3}$ pins can be used as UART2 or for port functions.

2. When CSIE3 = 1 (SIO3 operation enabled status), the SI3 pin can be used as a port pin if only the transmit function is used, and the SO3 pin can be used as a port pin if only the receive-only mode is used.

3. When MODE3 = 1 (receive-only mode), the SO3 pin can be used for port functions.

4. Expanded-specification products of μPD780078 Subseries only.

Caution Do not rewrite the value of CSIM3 during transfer. However, CSIE3 can be rewritten using a 1-bit memory manipulation instruction.

Remark f_X: Main system clock oscillation frequency

(3) SO0 latch

The SO0 latch is used to retain the SDA0 pin's output level.

(4) Wake-up controller

This circuit generates an interrupt request when the address received by this register matches the address value set to slave address register 0 (SVA0) or when an extension code is received.

(5) Prescaler

This selects the sampling clock to be used.

(6) Serial clock counter

This counter counts the serial clocks that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

(7) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIIC0).

An I^2C interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by WTIM0 bit[Note])
- Interrupt request generated when a stop condition is detected (set by SPIE0 bit[Note])

Note WTIM0 bit: Bit 3 of IIC control register 0 (IICC0)
SPIE0 bit: Bit 4 of IIC control register 0 (IICC0)

(8) Serial clock controller

In master mode, this circuit generates the clock output via the SCL0 pin from a sampling clock.

(9) Serial clock wait controller

This circuit controls the wait timing.

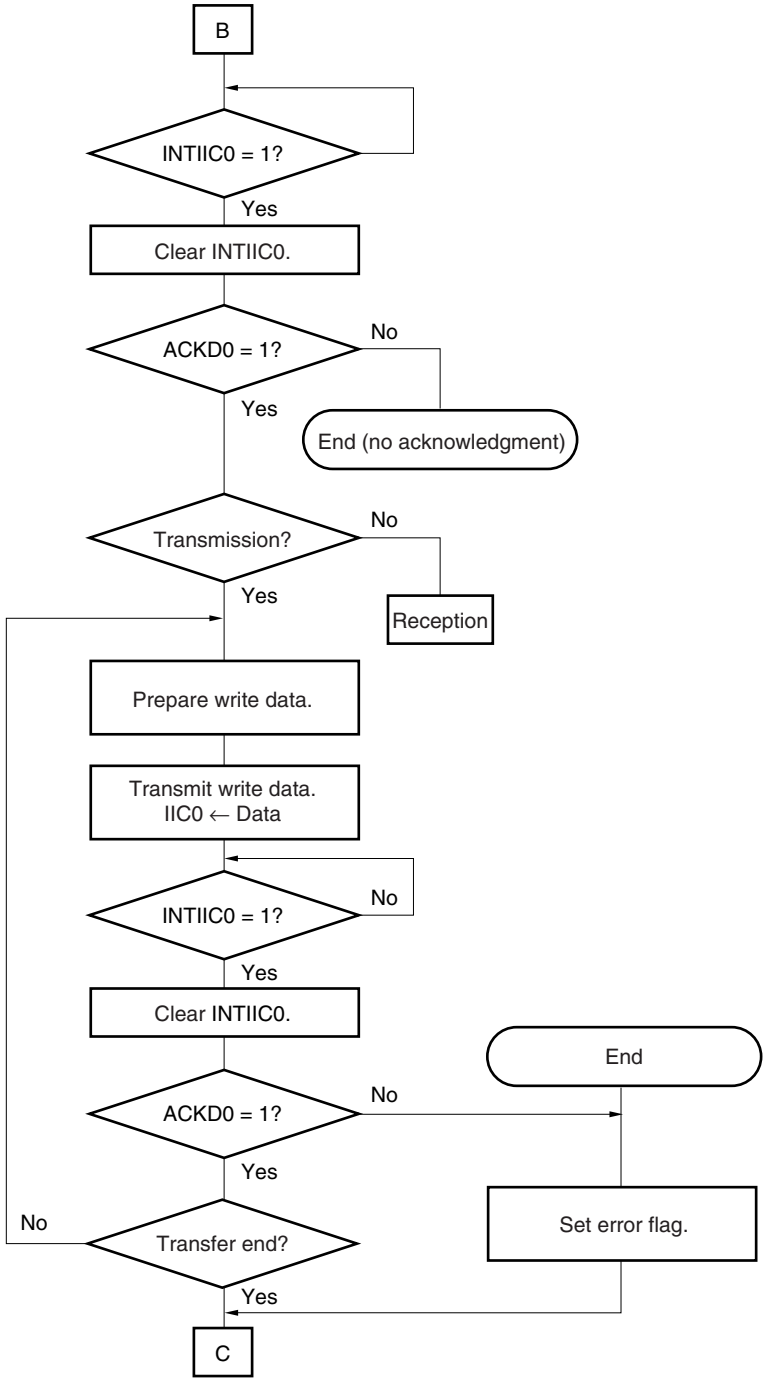
(10) $\overline{\text{ACK}}$ output circuit, stop condition detector, start condition detector, and $\overline{\text{ACK}}$ detector

These circuits are used to output and detect various control signals.

(11) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

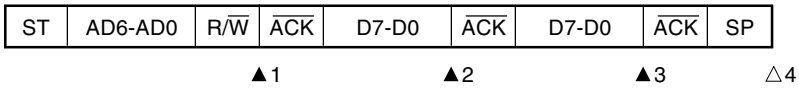
Figure 18-21. Master Operation Flowchart (3/5)



When writing data to EEPROM, continue writing data.
When reading data from EEPROM, start reception processing.
Prepare data to be written to EEPROM, and transmit it to EEPROM.
Each time data has been transmitted, the slave returns $\overline{ACK}$. If any error occurs before transmission of the necessary data is completed, $\overline{ACK}$ may not be returned. In this case, end transfer.
In the case of an error, set the error flag as shown on the left, and release the bus.

(b) When arbitration loss occurs during transmission of extension code

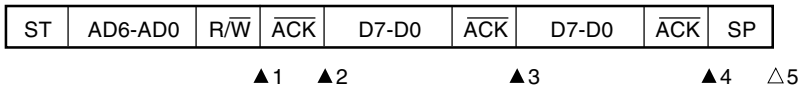
(i) When WTIM0 = 0



▲1 : IICS0 = 0110×010B (**Example** When ALD0 is read during interrupt servicing)
▲2 : IICS0 = 0010×000B
▲3 : IICS0 = 0010×000B
△4 : IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

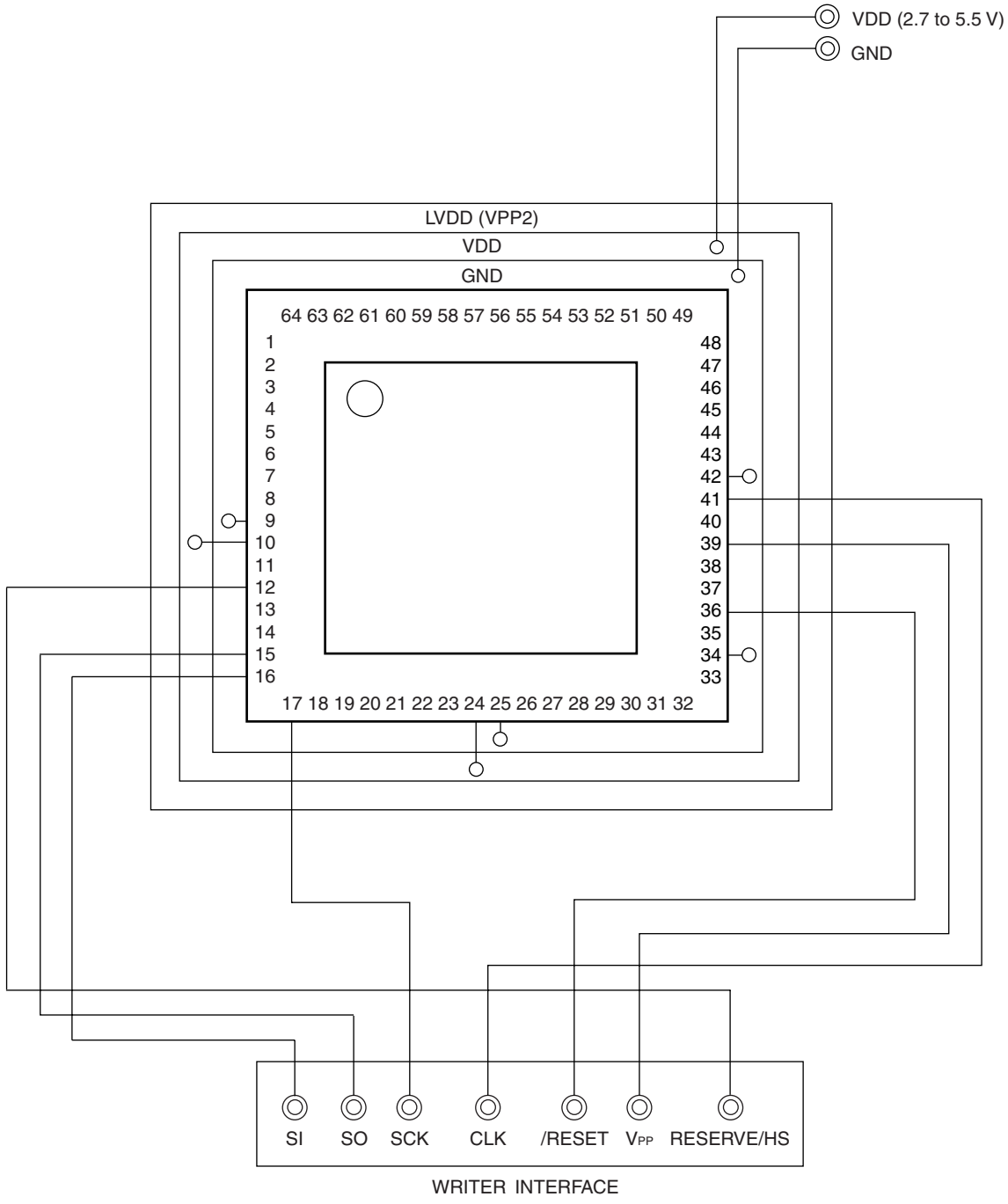
(ii) When WTIM0 = 1



▲1 : IICS0 = 0110×010B (**Example** When ALD0 is read during interrupt servicing)
▲2 : IICS0 = 0010×110B
▲3 : IICS0 = 0010×100B
▲4 : IICS0 = 0010××00B
△5 : IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Figure 23-11. Wiring Example for Adapter for Flash Writing with 3-Wire Serial I/O (SIO3) with Handshake



(2) μPD78F0078, 78F0078Y

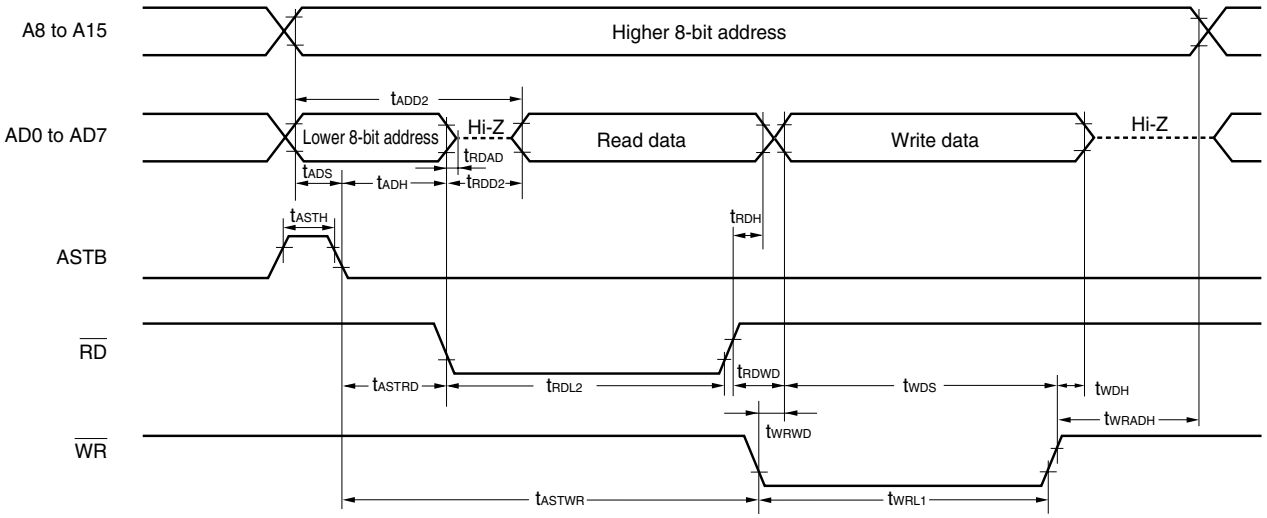
Main system clock: Ceramic resonator (T_A = –40 to +85°C)

Manufacturer	Part Number	Frequency (MHz)	Recommended Circuit Constant			Oscillation Voltage Range	
			C1 (pF)	C2 (pF)	R1 (kΩ)	MIN. (V)	MAX. (V)
Murata Mfg. Co., Ltd.	CSBFB1M00J58	1.00	100	100	3.3	1.8	5.5
	CSBLA1M00J58	1.00	100	100	3.3	1.8	5.5
	CSTCC2M00G56	2.00	On-chip	On-chip	0	1.8	5.5
	CSTLS2M00G56	2.00	On-chip	On-chip	0	1.8	5.5
	CSTCC3M58G53	3.58	On-chip	On-chip	0	1.8	5.5
	CSTLS3M58G53	3.58	On-chip	On-chip	0	1.8	5.5
	CSTCR4M00G53	4.00	On-chip	On-chip	0	1.8	5.5
	CSTLS4M00G53	4.00	On-chip	On-chip	0	1.8	5.5
	CSTCR4M19G53	4.19	On-chip	On-chip	0	1.8	5.5
	CSTLS4M19G53	4.19	On-chip	On-chip	0	1.8	5.5
	CSTCR4M91G53	4.91	On-chip	On-chip	0	1.8	5.5
	CSTLS4M91G53	4.91	On-chip	On-chip	0	1.8	5.5
	CSTCR5M00G53	5.00	On-chip	On-chip	0	2.7	5.5
	CSTLS5M00G53	5.00	On-chip	On-chip	0	2.7	5.5
	CSTCE8M00G52	8.00	On-chip	On-chip	0	3.0	5.5
	CSTLS8M00G53	8.00	On-chip	On-chip	0	3.0	5.5
	CSTCE8M38G52	8.38	On-chip	On-chip	0	3.0	5.5
	CSTLS8M38G53	8.38	On-chip	On-chip	0	3.0	5.5

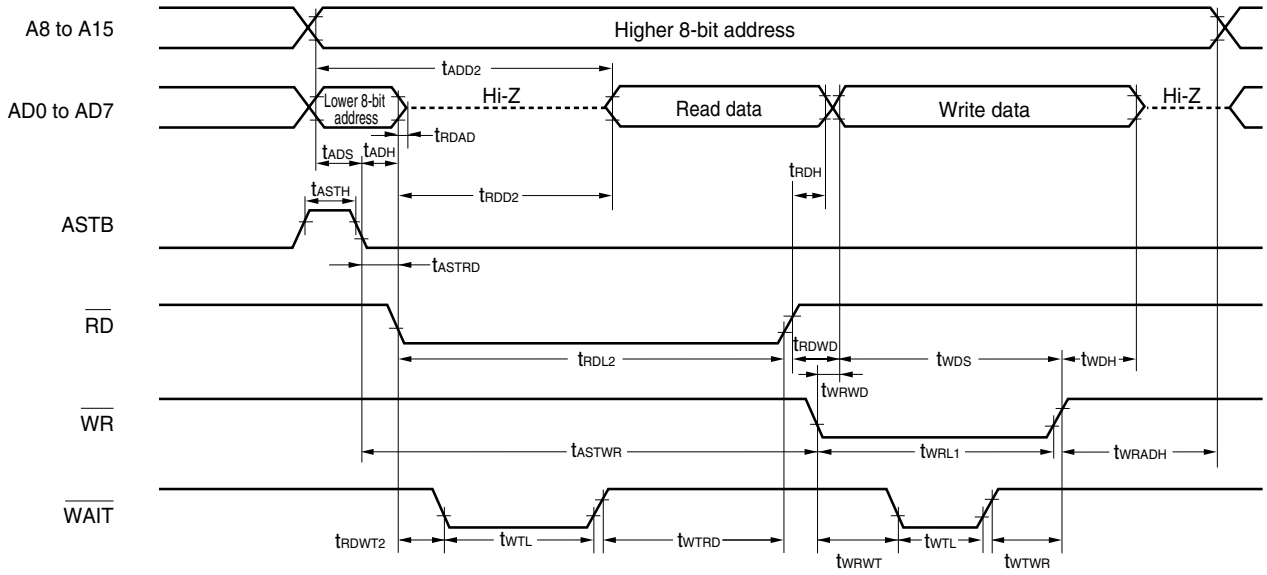
Caution The oscillator constant is a reference value based on evaluation in specific environments by the resonator manufacturer. If the oscillator characteristics need to be optimized in the actual application, request the resonator manufacturer for evaluation on the implementation circuit. Note that the oscillation voltage and oscillation frequency merely indicate the characteristics of the oscillator. Use the internal operation conditions of the μPD780078, 780078Y Subseries within the specifications of the DC and AC characteristics.

Remark For the resonator selection and oscillator constant, users are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

External data access (no wait):



External data access (wait insertion):



APPENDIX E REVISION HISTORY

E.1 Major Revisions in This Edition

Page	Description
Throughout	Addition of lead-free products
p. 140	Modification of **Table 7-3 Maximum Time Required for CPU Clock Switchover**
pp. 149, 150	Addition of explanation when register is used as capture register to **Figure 8-6 Format of 16-Bit Timer Mode Control Register 00 (TMC00)** and **Figure 8-7 Format of 16-Bit Timer Mode Control Register 01 (TMC01)**
pp. 155, 156	Modification of **Caution 4** in **Figure 8-12 Format of Prescaler Mode Register 00 (PRM00)** and **Figure 8-13 Format of Prescaler Mode Register 01 (PRM01)**
p. 186	Modification of description in **<1>** of **(10) Edge detection** in **8.6 Cautions for 16-Bit Timer/Event Counters 00, 01**
p. 324	Modification of PM22 and P22 in **Table 17-2 Relationship Between Register Settings and Pins (3-Wire Serial I/O Mode)**
p. 333	Modification of descriptions related to $\overline{ACK}$ and wait in **CHAPTER 18 SERIAL INTERFACE IIC0 (μPD780078Y SUBSERIES ONLY)**
p. 431	Modification of **(2) (c)** in **21.2.1 HALT mode**
p. 434	Modification of **(2) (b)** in **21.2.2 STOP mode**