Renesas Electronics America Inc - UPD78F0475GC-GAD-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	62
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	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f0475gc-gad-ax

EEPROM is a trademark of NEC Electronics Corporation.

SuperFlash is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

Caution: This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.

- **The information in this document is current as of June, 2008. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".

  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.

  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).

  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)

(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.

(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02. 11-1

(2/2)

Item		μPD78F0491	μPD78F0492	μPD78F0493	μPD78F0494	μPD78F0495
10-bit successive approximation type A/D converter		10-bit resolution × 8 channels (AV_{REF} = 2.3 to 5.5 V)				
16-bit ΔΣ type[Note 1] A/D converter		16-bit resolution × 3 channels (AV_{REF} = 2.7 to 5.5 V)				
Serial interface		• UART supporting LIN-bus[Note 2]: 1 channel • 3-wire serial I/O/UART[Note 3]: 1 channel • Automatic transmit/receive 3-wire CSI: 1 channel				
LCD controller/driver		• External resistance division and internal resistance division are switchable. • Segment signal outputs: 32 (28) [28 (24)][Note 4, 5] • Common signal outputs: 4 (8)[Note 4]				
Remote controller receiver		Provided				
Manchester code generator		Provided				
Vectored interrupt sources	Internal	22				
	External	7				
<R> Segment key source signal output		Segment key source signal outputs: 8 (SEG24(KS0)-SEG31(KS7))				
Key interrupt		Key interrupt (INTKR) occurs by detecting falling edge of key input pins (KR0 to KR7).				
Reset		• Reset using $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-clear • Internal reset by low-voltage detector				
On-chip debug function		Provided				
Power supply voltage		V_{DD} = 1.8 to 5.5 V				
Operating ambient temperature		T_A = −40 to +85°C				
Package		• 80-pin plastic LQFP (14 × 14) • 80-pin plastic LQFP (fine pitch) (12 × 12)				

Notes 1. The specifications of the 16-bit ΔΣ A/D converter may have been changed.
For details of the specifications, contact an NEC Electronics sales representative or authorized dealer.

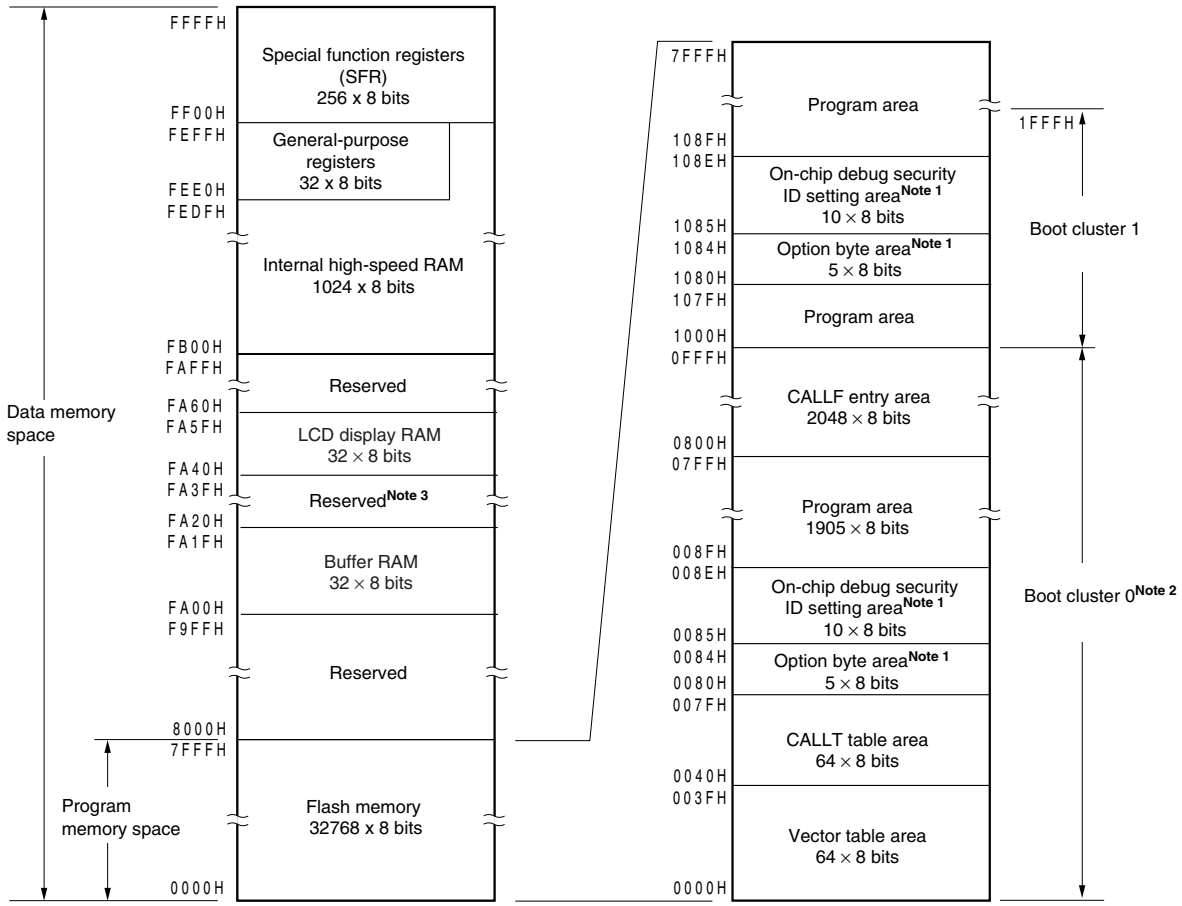
2. The LIN-bus supporting UART pins can be changed to the automatic transmit/receive 3-wire CSI pins (pin numbers 75 and 76).

3. Select either of the functions of these alternate-function pins.

4. The values in parentheses are the number of signal outputs when 8com is used.

<R> **5.** The values in square brackets are the number of signal outputs when using the UART6 pins (RxD6, TxD6) on the bottom side.

Figure 3-6. Memory Map (μPD78F0493)



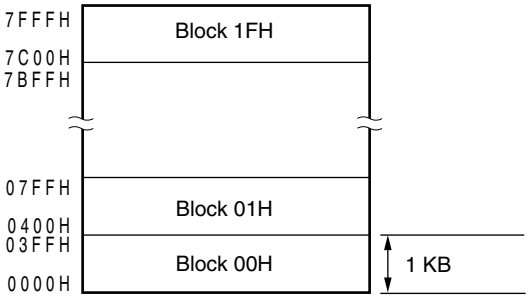
Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **28.8 Security Setting**).

<R> **3.** However, FA26H and FA27H can be used (See **13.3 Registers Used in 16-Bit ΔΣ Type A/D Converter**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.



(4) Example of setting procedure when stopping the high-speed system clock

The high-speed system clock can be stopped in the following two ways.

- Executing the STOP instruction to set the STOP mode
- Setting MSTOP to 1 and stopping the X1 oscillation (disabling clock input if the external clock is used)

(a) To execute a STOP instruction

<1> Setting to stop peripheral hardware

Stop peripheral hardware that cannot be used in the STOP mode (for peripheral hardware that cannot be used in STOP mode, see **CHAPTER 23 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after standby release

When the CPU is operating on the X1 clock, set the value of the OSTS register before the STOP instruction is executed.

<3> Executing the STOP instruction

When the STOP instruction is executed, the system is placed in the STOP mode and X1 oscillation is stopped (the input of the external clock is disabled).

(b) To stop X1 oscillation (disabling external clock input) by setting MSTOP to 1

<1> Confirming the CPU clock status (PCC and MCM registers)

Confirm with CLS and MCS that the CPU is operating on a clock other than the high-speed system clock.

When CLS = 0 and MCS = 1, the high-speed system clock is supplied to the CPU, so change the CPU clock to the subsystem clock or internal high-speed oscillation clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Stopping the high-speed system clock (MOC register)

When MSTOP is set to 1, X1 oscillation is stopped (the input of the external clock is disabled).

Caution Be sure to confirm that MCS = 0 or CLS = 1 when setting MSTOP to 1. In addition, stop peripheral hardware that is operating on the high-speed system clock.

5.6.2 Example of controlling internal high-speed oscillation clock

The following describes examples of clock setting procedures for the following cases.

(1) When restarting oscillation of the internal high-speed oscillation clock
(2) When using internal high-speed oscillation clock as CPU clock, and internal high-speed oscillation clock or high-speed system clock as peripheral hardware clock
(3) When stopping the internal high-speed oscillation clock

(6) Port mode register 3 (PM3)

This register sets port 3 input/output in 1-bit units.

When using the P34/TI52/TI010/TO00/RTC1HZ/INTP1 pin for timer output, set PM34 and the output latches of P34 to 0.

When using the P33/TI000/RTCDIV/RTCCL/BUZ/INTP2 and P34/TI52/TI010/TO00/RTC1HZ/INTP1 pins for timer input, set PM33 and PM34 to 1. At this time, the output latches of P33 and P34 may be 0 or 1.

PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM3 to FFH.

Figure 6-11. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

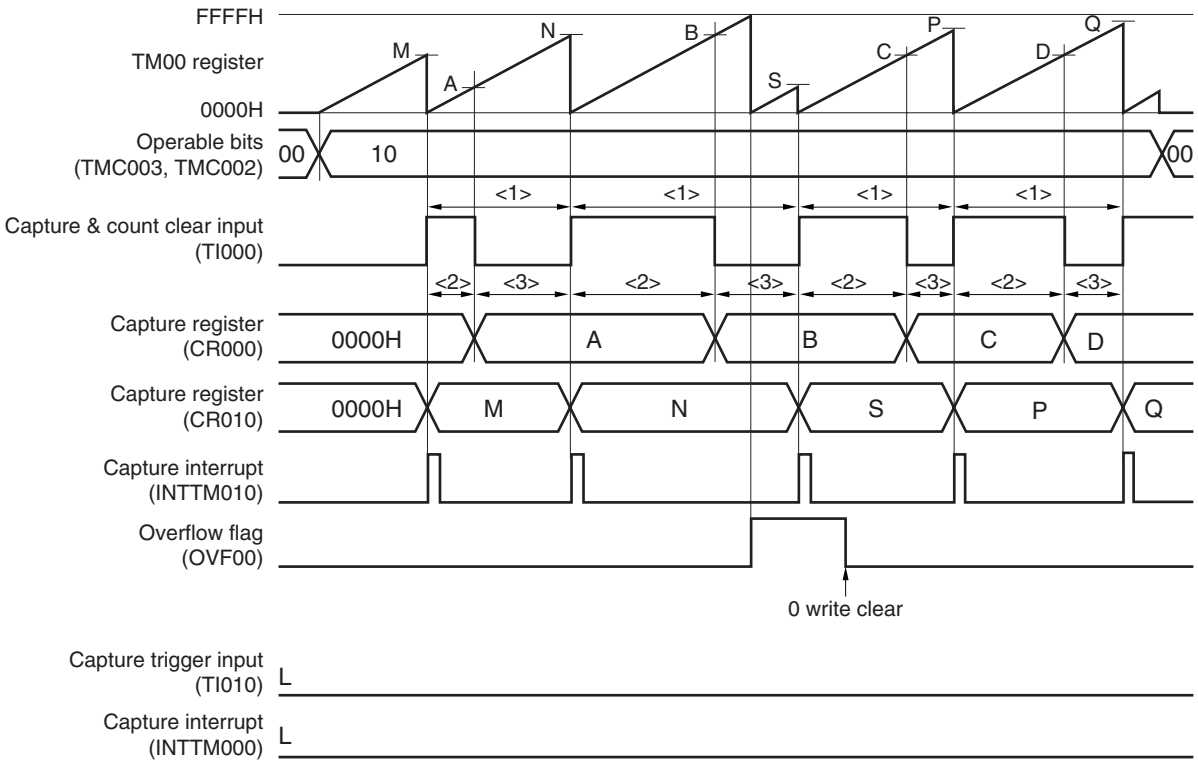
(3) Measuring the pulse width by using one input signal of the TI000 pin (clear & start mode entered by the TI000 pin valid edge input)

Set the clear & start mode entered by the TI000 pin valid edge (TMC003 and TMC002 = 10). The count value of TM00 is captured to CR000 in the phase reverse to the valid edge of the TI000 pin, and the count value of TM00 is captured to CR010 and TM00 is cleared (0000H) when the valid edge of the TI000 pin is detected. Therefore, a cycle is stored in CR010 if TM00 does not overflow.

If an overflow occurs, take the value that results from adding 10000H to the value stored in CR010 as a cycle. Clear bit 0 (OVF00) of 16-bit timer mode control register 00 (TMC00) to 0.

Figure 6-51. Timing Example of Pulse Width Measurement (3)

• TMC00 = 08H, PRM00 = 10H, CRC00 = 07H



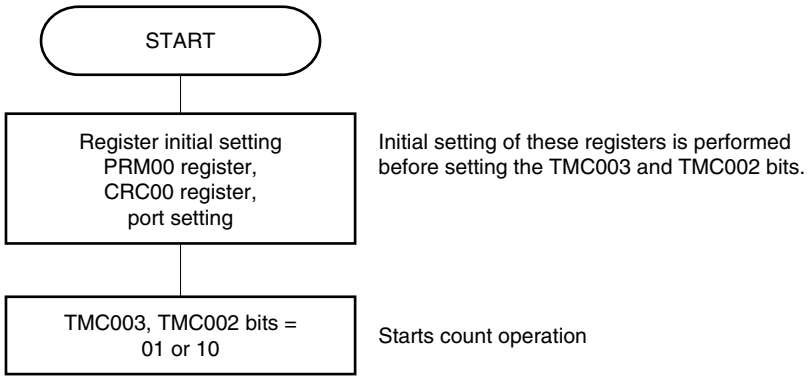
<1> Pulse cycle = (10000H × Number of times OVF00 bit is set to 1 + Captured value of CR010) × Count clock cycle

<2> High-level pulse width = (10000H × Number of times OVF00 bit is set to 1 + Captured value of CR000) × Count clock cycle

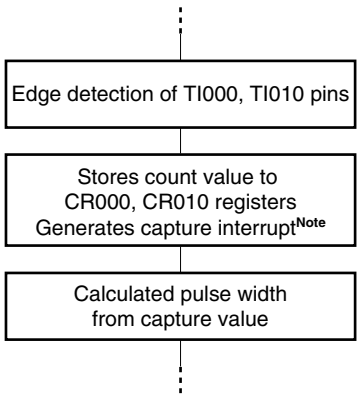
<3> Low-level pulse width = (Pulse cycle − High-level pulse width)

Figure 6-53. Example of Software Processing for Pulse Width Measurement (2/2)

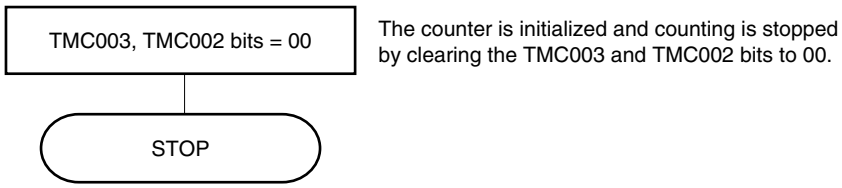
<1> Count operation start flow



<2> Capture trigger input flow



<3> Count operation stop flow



Note The capture interrupt signal (INTTM000) is not generated when the reverse-phase edge of the TI000 pin input is selected to the valid edge of CR000.

6.4.9 External 24-bit event counter operation

16-bit timer/event counter 00 can be operated to function as an external 24-bit event counter, by connecting 16-bit timer/event counter 00 and 8-bit timer/event counter 52 in cascade, and using the external event counter function of 8-bit timer/event counter 52.

It operates as an external 24-bit event counter, by counting the number of external clock pulses input to the TI52 pin via 8-bit timer counter 52 (TM52), and counting the signal which has been output upon a match between the TM52 count value and 8-bit timer compare register 52 (CR52 = FFH[Note]) via 16-bit timer counter 00 (TM00).

When using 16-bit timer/event counter 00 as an external 24-bit event counter, external event input enable can be controlled via 8-bit timer counter H2 output.

The valid edge of the input to the TI52 pin can be specified by timer clock selection register 52 (TCL52) of 8-bit timer counter 52 (TM52). Also, input enable for TM52 external event input can be controlled via 8-bit timer counter H2 output, by setting bit 2 (ISC2) of the input switch control register (ISC) to “1”.

Count operation using 8-bit timer 52 output as the count clock is started, by setting bits 2, 1, and 0 (PRM002, PRM001, and PRM000) of prescaler mode register 00 (PRM00) of 16-bit timer/event counter 00 to “1”, “1”, and “1” (TM52 output is selected as a count clock), and bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) to “1” and “1” (count clear & start mode entered upon a match between TM00 and CR000). TM00 is cleared to “0” and an interrupt request signal (INTTM000) is generated upon a match between the TM00 count value and 16-bit timer compare register 000 (CR000) value.

Subsequently, INTTM000 is generated upon every match between the TM00 and CR000 values.

Note When operating 16-bit timer/event counter 00 as an external 24-bit event counter, the 8-bit timer compare register 52 (CR52) value must be set to FFH. Also, the TM52 interrupt request signal (INTTM52) must be masked (TMMK52 = 1).

7.4 Operations of 8-Bit Timer/Event Counters 50, 51, and 52

7.4.1 Operation as interval timer

8-bit timer/event counter 5n operates as an interval timer that generates interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare register 5n (CR5n).

When the count value of 8-bit timer counter 5n (TM5n) matches the value set to CR5n, counting continues with the TM5n value cleared to 0 and an interrupt request signal (INTTM5n) is generated.

The count clock of TM5n can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

Setting

<1> Set the registers.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on a match of TM5n and CR5n.
  (TMC5n = 0000×××0B × = Don't care)

<2> After TCE5n = 1 is set, the count operation starts.

<3> If the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<4> INTTM5n is generated repeatedly at the same interval.
Set TCE5n to 0 to stop the count operation.

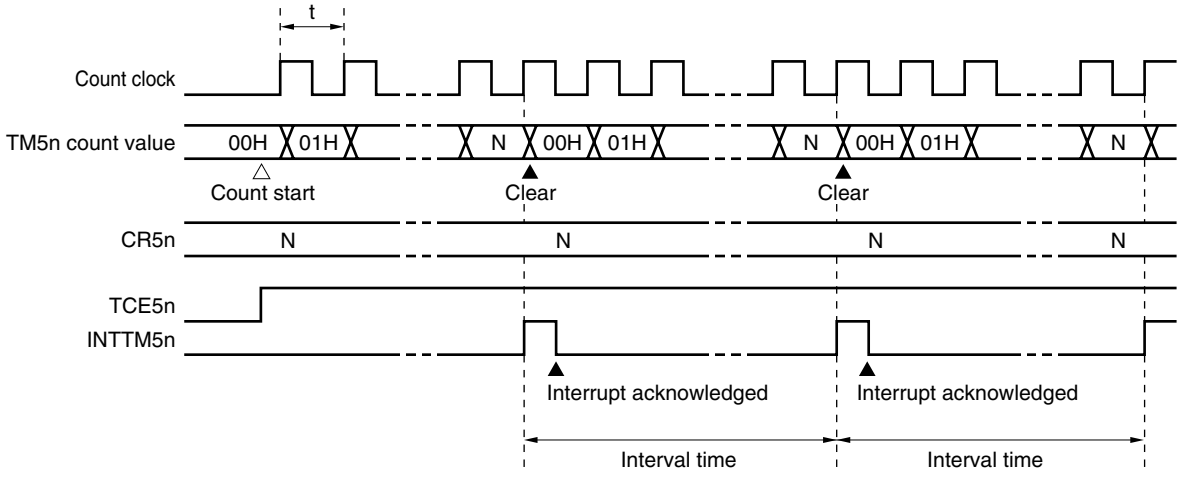
Caution Do not write other values to CR5n during operation.

Remarks 1. For how to enable the INTTM5n signal interrupt, see **CHAPTER 21 INTERRUPT FUNCTIONS**.

2. n = 0 to 2

Figure 7-15. Interval Timer Operation Timing (1/2)

(a) Basic operation



Remark Interval time = (N + 1) × t
N = 01H to FFH
n = 0 to 2

<10> By performing the procedures above, an arbitrary carrier clock is obtained. To stop the count operation, clear TMHE1 to 0.

If the setting value of the CMP01 register is N, the setting value of the CMP11 register is M, and the count clock frequency is f_{CNT}, the carrier clock output cycle and duty are as follows.

- Carrier clock output cycle = (N + M + 2)/f_{CNT}
- Duty = High-level width/carrier clock output width = (M + 1)/(N + M + 2)

Cautions 1. Be sure to set the CMP11 register when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to the CMP11 register).

2. Set so that the count clock frequency of TMH1 becomes more than 6 times the count clock frequency of TM51.

3. Set the values of the CMP01 and CMP11 registers in a range of 01H to FFH.

4. The set value of the CMP11 register can be changed while the timer counter is operating. However, it takes the duration of three operating clocks (signal selected by the CKS12 to CKS10 bits of the TMHMD1 register) since the value of the CMP11 register has been changed until the value is transferred to the register.

5. Be sure to set the RMC1 bit before the count operation is started.

Remarks 1. For the setting of the output pin, see **8.3 (3) Port mode register 3 (PM3)**.

2. For how to enable the INTTMH1 signal interrupt, see **CHAPTER 21 INTERRUPT FUNCTIONS**.

(16) Alarm week register (ALARMWW)

This register is used to set date of alarm.

ALARMWW can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-17. Format of Alarm Week Register (ALARMWW)

Address: FF88H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

Here is an example of setting the alarm.

Time of Alarm	Day							12-Hour Display				24-Hour Display			
	Sunday WW0	Monday WW1	Tuesday WW2	Wednesday WW3	Thursday WW4	Friday WW5	Saturday WW6	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

(10) Port mode register 11 (PM11)

This register sets port 11 input/output in 1-bit units.

When using the P112/SEG18/TxD6 pin for serial interface data output, clear PM112 to 0 and set the output latch of P112 to 1.

When using the P113/SEG19/RxD6 pin for serial interface data input, set PM113 to 1. The output latch of P113 at this time may be 0 or 1.

PM11 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 15-14. Format of Port Mode Register 11 (PM11)

Address: FF2BH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM11	1	1	1	1	PM113	PM112	PM111	PM110

PM11n	P11n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

15.4.2 Asynchronous serial interface (UART) mode

In this mode, data of 1 byte is transmitted/received following a start bit, and a full-duplex operation can be performed.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

(1) Registers used

- Asynchronous serial interface operation mode register 6 (ASIM6)
- Asynchronous serial interface reception error status register 6 (ASIS6)
- Asynchronous serial interface transmission status register 6 (ASIF6)
- Clock selection register 6 (CKSR6)
- Baud rate generator control register 6 (BRGC6)
- Asynchronous serial interface control register 6 (ASICL6)
- Input switch control register (ISC)
- Port mode register 1 (PM1)
- Port register 1 (P1)
- Port mode register 11 (PM11)
- Port register 11 (P11)

The basic procedure of setting an operation in the UART mode is as follows.

<1> Set the CKSR6 register (see **Figure 15-8**).
<2> Set the BRGC6 register (see **Figure 15-9**).
<3> Set bits 0 to 4 (ISRM6, SL6, CL6, PS60, PS61) of the ASIM6 register (see **Figure 15-5**).
<4> Set bits 0 and 1 (TXDLV6, DIR6) of the ASICL6 register (see **Figure 15-10**).
<5> Set bit 7 (POWER6) of the ASIM6 register to 1.
<6> Set bit 6 (TXE6) of the ASIM6 register to 1. → Transmission is enabled.
Set bit 5 (RXE6) of the ASIM6 register to 1. → Reception is enabled.
<7> Write data to transmit buffer register 6 (TXB6). → Data transmission is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

Notes 1. When LCD display is not to be performed or not required, power consumption can be reduced by using the following settings.

<1> Set both SCOC and VAON to 0.

<2> When the internal resistance division method is used, assume MDSET1, MDSET0 = (0, 0). (The current flowing to the internal resistors can be reduced.)

2. This bit is used to control boosting of the internal gate signal of the LCD controller/driver.

If set to "Internal gate voltage boosting", the LCD drive performance can be enhanced.

Set VAON based on the following conditions.

<When set to the static display mode>

- When 2.0 V ≤ V_{LCD} ≤ V_{DD} ≤ 5.5 V: VAON = 0
- When 1.8 V ≤ V_{LCD} ≤ V_{DD} ≤ 3.6 V: VAON = 1

<When set to the 1/3 bias method>

- When 2.5 V ≤ V_{LCD} ≤ V_{DD} ≤ 5.5 V: VAON = 0
- When 1.8 V ≤ V_{LCD} ≤ V_{DD} ≤ 3.6 V: VAON = 1

<When set to the 1/2 bias method or 1/4 bias method >

- When 2.7 V ≤ V_{LCD} ≤ V_{DD} ≤ 5.5 V: VAON = 0
- When 1.8 V ≤ V_{LCD} ≤ V_{DD} ≤ 3.6 V: VAON = 1

3. When using the segment key scan function (KSON = 1), “number of time slices + 1” is added for segment key scan signal output.

4. When the P40/KR0/V_{LC3} pin is set to the 1/4 bias method, it is used as V_{LC3}. When the pin is set to another bias method, it is used for the port function (P40) or the key interrupt function (KR0).

Cautions 1. Bits 3 and 5 must be set to 0.

2. When displaying in a mode with a large number of COMs, such as 8 COM, V_{LC0} may not be able to obtain sufficient contrast under the low-voltage conditions, depending on the panel characteristics. Use the LCD controller/driver after having performed thorough LCD display evaluation and confirmed that there are no problems regarding the display quality.

Cautions 3. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).
If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

Table 24-2. Hardware Statuses After Reset Acknowledgment (2/4)

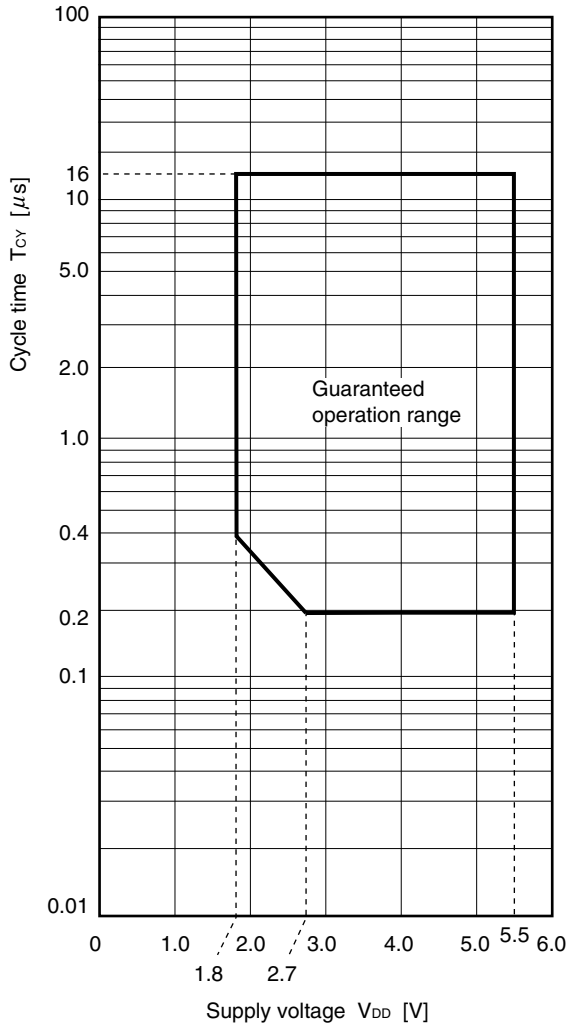
Hardware		Status After Reset Acknowledgment[Note 1]
8-bit timers H0, H1, H2	Compare registers 00, 10, 01, 11, 02, 12 (CMP00, CMP10, CMP01, CMP11, CMP02, CMP12)	00H
	Mode registers (TMHMD0, TMHMD1, TMHMD2)	00H
	Carrier control register 1 (TMCYC1)[Note 2]	00H
Real-time counter	Clock selection register (RTCCL)	00H
	Sub-count register (RSUBC)	0000H
	Second count register (SEC)	00H
	Minute count register (MIN)	00H
	Hour count register (HOUR)	12H
	Week count register (WEEK)	00H
	Day count register (DAY)	01H
	Month count register (MONTH)	01H
	Year count register (YEAR)	00H
	Watch error correction register (SUBCUD)	00H
	Alarm minute register (ALARMWM)	00H
	Alarm hour register (ALARMWH)	12H
	Alarm week register (ALARMWW)	00H
	Control register 0 (RTCC0)	00H
	Control register 1 (RTCC1)	00H
	Control register 2 (RTCC2)	00H
Clock output/buzzer output controller	Clock output selection register (CKS)	00H
Watchdog timer	Enable register (WDTE)	1AH/9AH[Note 3]
10-bit successive approximation type A/D converter	10-bit A/D conversion result register (ADCR)	0000H
	8-bit A/D conversion result register (ADCRH)	00H
	A/D converter mode register (ADM)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register 0 (ADPC0)	08H
16-bit ΔΣ type A/D converter	ΔΣ A/D converter control register 0 (ADDCTL0)	00H
	ΔΣ A/D converter control register 1 (ADDCTL1)	00H
	16-bit ΔΣ A/D conversion status register (ADDSTR)	00H
	16-bit ΔΣ A/D conversion result register (ADDCR)	0000H
	8-bit ΔΣ A/D conversion result register (ADDCRH)	00H
Serial interface UART0	Receive buffer register 0 (RXB0)	FFH
	Transmit shift register 0 (TXS0)	FFH
	Asynchronous serial interface operation mode register 0 (ASIM0)	01H
	Asynchronous serial interface reception error status register 0 (ASIS0)	00H
	Baud rate generator control register 0 (BRGC0)	1FH

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

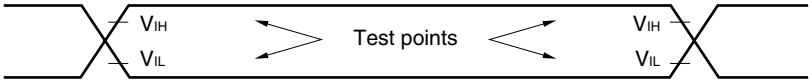
2. 8-bit timer H1 only.

3. The reset value of WDTE is determined by the option byte setting.

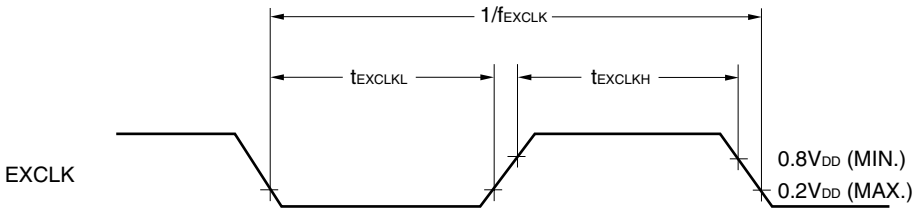
T_{CY} vs. V_{DD} (Main System Clock Operation)



AC Timing Test Points (Excluding External Main System Clock)



External Main System Clock Timing



34.2 Peripheral Hardware That Generates Wait

Table 34-1 lists the registers that issue a wait request when accessed by the CPU, and the number of CPU wait clocks and Table 34-2 lists the RAM accesses that issue a wait request and the number of CPU wait clocks.

Table 34-1. Registers That Generate Wait and Number of CPU Wait Clocks

Peripheral Hardware	Register	Access	Number of Wait Clocks
Serial interface UART0	ASIS0	Read	1 clock (fixed)
Serial interface UART6	ASIS6	Read	1 clock (fixed)
10-bit successive approximation type A/D converter	ADM	Write	1 to 5 clocks (when $f_{AD} = f_{PRS}/2$ is selected) 1 to 7 clocks (when $f_{AD} = f_{PRS}/3$ is selected) 1 to 9 clocks (when $f_{AD} = f_{PRS}/4$ is selected) 2 to 13 clocks (when $f_{AD} = f_{PRS}/6$ is selected) 2 to 17 clocks (when $f_{AD} = f_{PRS}/8$ is selected) 2 to 25 clocks (when $f_{AD} = f_{PRS}/12$ is selected)
	ADS	Write	
	ADPC	Write	
	ADCR	Read	
	The above number of clocks is when the same source clock is selected for f_{CPU} and f_{PRS}. The number of wait clocks can be calculated by the following expression and under the following conditions. <Calculating number of wait clocks> • Number of wait clocks = $\frac{2 f_{CPU}}{f_{AD}} + 1$ * Fraction is truncated if the number of wait clocks ≤ 0.5 and rounded up if the number of wait clocks > 0.5. f_{AD}: A/D conversion clock frequency ($f_{PRS}/2$ to $f_{PRS}/12$) f_{CPU}: CPU clock frequency f_{PRS}: Peripheral hardware clock frequency f_{XP}: Main system clock frequency <Conditions for maximum/minimum number of wait clocks> • Maximum number of times: Maximum speed of CPU (f_{XP}), lowest speed of A/D conversion clock ($f_{PRS}/12$) • Minimum number of times: Minimum speed of CPU ($f_{SUB}/2$), highest speed of A/D conversion clock ($f_{PRS}/2$)		

Caution When the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped, do not access the registers listed above using an access method in which a wait request is issued.

Remark The clock is the CPU clock (f_{CPU}).

Table 34-2. RAM Accesses That Generate Wait and Number of CPU Wait Clocks

Area	Access	Number of Wait Clocks
Buffer RAM of CSIA0	Write	See the following calculation formula[Note]
<Calculating maximum number of wait clocks> • Maximum Number of wait clocks = $\frac{5 f_{CPU}}{f_W} + 1$ * Fraction is truncated if the number of wait clocks multiplied by ($1/f_{CPU}$) is equal or lower than t_{CPUL} and rounded up if higher than t_{CPUL}. f_W: Frequency of base clock selected by CKS00 bit of CSIS0 register (CKS00 = 0: f_{PRS}, CKS00 = 1: $f_{PRS}/2$) f_{CPU}: CPU clock frequency t_{CPUL}: CPU clock low-level width f_{PRS}: Peripheral hardware clock frequency		

Note No waits are generated when five CSIA0 operating clocks or more are inserted between writing to the RAM from the CSIA0 and writing to the buffer RAM from the CPU.

B.2 Revision History up to Previous Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/3)

Edition	Description	Applied to:
2nd edition	Addition of **Note** to **1.3 Ordering Information**	**CHAPTER 1 OUTLINE**
	Addition of **Caution 3** to **(1)** in **1.4 Pin Configuration (Top View)**	
	Change of **Table 2-2. Pin I/O Circuit Types**	**CHAPTER 2 PIN FUNCTIONS**
	Modification of **Figure 3-9. Memory Map (μPD78F0475, 78F0485)**	**CHAPTER 3 CPU ARCHITECTURE**
	Modification of **Figure 3-10. Memory Map (μPD78F0495)**	
	Modification of **Figure 4-5. Block Diagram of P13**	**CHAPTER 4 PORT FUNCTIONS**
	Modification of **Figure 4-6. Block Diagram of P16**	
	Modification of **Figure 4-29. Format of Pull-up Resistor Option Register**	
	Change of **Table 4-5. Settings of PFALL, PF2, PF1, ISC, Port Mode Register, and Output Latch When Using Alternate Function**	
	Change of **5.3 (9) Internal high-speed oscillation trimming register (HIOTRM)**	**CHAPTER 5 CLOCK GENERATOR**
	Addition of explanation to **Table 5-4. Clocks Supplied to CPU and Peripheral Hardware, and Register Setting**	
	Addition of explanation to **Table 5-6. Changing CPU Clock**	
	• TO00 pin output → TO00 output • Addition of TO00 output in block diagram	**CHAPTER 6 16-BIT TIMER/EVENT COUNTER 00**
	Addition of explanation to **Figure 6-8 Format of 16-bit Timer Output Control Register 00 (TOC00)**	
	Addition of **Notes 1** and **2** to and change of **Note 3** in **Figure 6-9 Format of Prescaler Mode Register 00 (PRM00)**	
	Change of **Figure 6-54. Configuration Diagram of External 24-bit Event Counter**	
	Change of **Figure 6-55. Operation Timing of External 24-bit Event Counter**	
	• TO50 pin output → TO50 output, TO51 pin output → TO51 output • Addition of TO50, TO51 output in block diagram	**CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 50, 51, AND 52**
	Addition of **Notes 1** and **2** to **Figure 7-6 Format of Timer Clock Selection Register 50 (TCL50)**	
	Addition of **Notes 1** and **2** to **Figure 7-7 Format of Timer Clock Selection Register 51 (TCL51)**	
	Addition of **Notes 1** and **2** to **Figure 7-8 Format of Timer Clock Selection Register 52 (TCL52)**	
	Chenge of **Caution 3** in and addition of **Caution 4** to **Figure 7-9 Format of 8-bit Timer Mode Control Register 50 (TMC50)** and **Figure 7-10 Format of 8-bit Timer Mode Control Register 51 (TMC51)**	
	Full-scale revision	**CHAPTER 8 8-BIT TIMERS H0, H1, AND H2**
	Addition of **Note 1** to **Figure 11-2 Format of Clock Output Selection Register (CKS)**	**CHAPTER 11 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER**