Microcomputer Technical Information

CP(K), O

μPD789046 Subseries Usage Restrictions	Document No.	SBG-DT-03-0187-E		1/2
	Date issued	July 18, 2003		
	Issued by	Microcomputer Group 2nd Solutions Division Solutions Operations Unit NEC Electronics Corporation		
Related documents μPD789046 Subseries User's manual: U13600EJ2V0UMJ1 μPD789046 Data sheet: U13380EJ1V1DS00 μPD78F9046 Data sheet: U13546EJ1V0DS00	Notification classification	√	Usage restriction	
			Upgrade	
		√	Document modification	
			Other notification	

1. Affected products

μPD789046

μPD78F9046

2. Details of modification and added restriction

(1) Error on starting 8-bit timer

The description in **7.5 Notes on Using 8-Bit Timer/Event Counter (1) Error on starting timer** in the user's manual (U13600EJ2V0UMJ1) has been corrected as follows.

(Incorrect)

An error of up to 1 clock is included in the time between the timer being started and a match signal is being generated.

(Correct)

An error of up to 1.5 clocks is included in the time between the timer being started and a match signal is being generated.

(2) Restriction when an external event is selected for the timer count clock

If an external event (TI pin input) is selected for the timer count clock when the 8-bit timer/event counter is used, the count may be incremented after the count operation flag (TCExx) is set, depending on the status of the valid edge and TI pin.

See attachment 8 for details.

3. Correction

Regard the restriction added in this document as a usage restriction.

4. List of restrictions

The restriction history and detailed information is described in attachment 1.

5. Document revision history

μPD789046 Subseries Usage Restrictions

Document Number	Date Issued	Description
SBG-DT-0011-E	October 5, 2001	Newly created.
SBG-DT-03-0187-E	July 18, 2003	2nd edition Addition of new restriction (No.7)

List of Usage Restrictions in μPD789046 Subseries

1. Product History

<Mask ROM version>

Description		μPD789046
	Rank	All Ranks
1	Restriction on 16-bit timer (1)	Δ
2	Restriction on 8-bit timer (1)	Δ
3	Restriction on 16-bit timer (2)	Δ
4	Restriction on 8-bit timer (2)	Δ
5	Restriction on 16-bit timer (3)	Δ
6	Restriction on UART	Δ
7	Restriction on 8-bit timer (3)	Δ

<Flash memory version>

Description		μPD78F9046
	Rank	All Ranks
1	Restriction on 16-bit timer (1)	Δ
2	Restriction on 8-bit timer (1)	Δ
3	Restriction on 16-bit timer (2)	Δ
4	Restriction on 8-bit timer (2)	Δ
5	Restriction on 16-bit timer (3)	Δ
6	Restriction on UART	Δ
7	Restriction on 8-bit timer (3)	Δ

Notes 1. The rank is indicated by the fifth character from the left in the lot number marked on the package.

2. The meaning of each symbol is as follows.

–: Restriction does not apply

√: Restriction already corrected

×: Restriction applies (correction is planned)

Δ: Restriction applies (correction is not planned)

2. Details of Usage Restrictions

No.1: Refer to attachment 2 for details.

No.2: Refer to attachment 3 for details.

No.3: Refer to attachment 4 for details.

No.4: Refer to attachment 5 for details.

No.5: Refer to attachment 6 for details.

No.6: Refer to attachment 7 for details.

No.7: A restriction added this time. Refer to attachment 8 for details.

No.1 Restriction on 16-bit timer (1)

[Description]

When the value of the compare register is overwritten in the interrupt routine for a match between the count value and the 16-bit compare register (CRxx) when the 16-bit timer is used as an interval timer, be sure to perform the following processing before overwriting the value.

<1> Mask the interrupt.

<2> Disable inversion control (TOCxx) of the timer output data.

An interrupt request may be generated if the value of the compare register is overwritten while interrupts are enabled.

No.2 Restriction on 8-bit timer (1)

<Restriction 1>

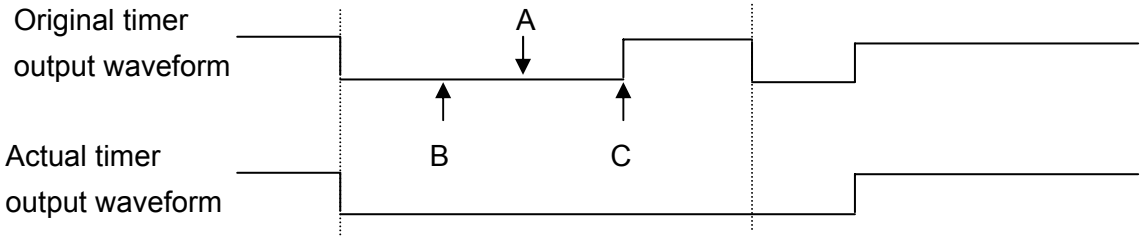
When the 8-bit timer is used in the interval timer mode, the value of the compare register (CR80) should be overwritten while the timer operation is disabled.

If the value of the compare register (CR80) is overwritten while the timer operation is enabled, a match signal may be generated. (An interrupt request is generated when interrupts are enabled.)

<Restriction 2>

The 8-bit timer does not have a compare register reload function. When the 8-bit timer is used in PWM mode, therefore, the following phenomenon may occur if the value of the compare register (CR80) is overwritten while the timer is operating.

<Phenomenon>



[Description]

If the value of the compare register is set to “B” while the count value is “A” (A < C, A > B) where B is the overwritten values of the compare register (CR80) and C is the original value of the compare register (CR80), no pulse is output during this period.

No.3 Restriction on 16-bit timer (2)

[Description]
When the 16-bit timer is used under the following conditions, the timer output function and timer interrupt function cannot be used.

(Condition)

Timer clock:	Subclock (f_{XT})
CPU clock:	Subclock (f_{XT})
Main clock:	Stopped
BZOE90:	“0” (no buzzer output)

<Measures if the main clock is stopped to lower the current consumption and the subclock is selected for the 16-bit timer to release the HALT mode>

Timer clock:	Subclock
CPU clock:	Subclock
Main clock:	Stopped
BZOE90:	“1” (buzzer output enabled)

When the port mode register and output latch of the buzzer output alternate function pin are 0, a waveform is output from the alternate function pin. Perform either of the following processing before implementing the above measures.

- Set the buzzer output alternate function pin to input mode.
- If the buzzer output alternate function pin cannot be set to the input mode, set the value of the port latch to 1.

(The buzzer output alternate function pin outputs a high level.)

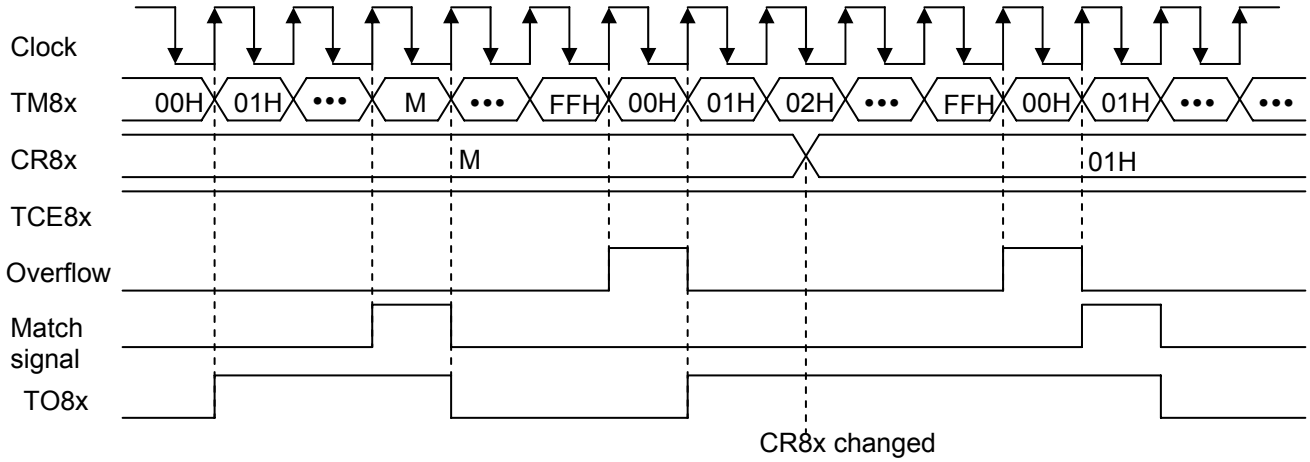
No.4 Restriction on 8-bit timer (2)

[Description]

The 8-bit timer (TM8) has a PWM function, however the following restrictions apply when this function is used.

• When modifying the compare register (CR8x)

While the timer is operating (during PWM output), if a value is written to the compare register (CR8x), a high level may be output for the next cycle (count pulse × 256). This phenomenon occurs if a value smaller than the value of TM8x is written to CR8x.



No.5 Restriction on 16-bit timer (3)

[Description]

When the 16-bit timer (timer 90) is used as an interval timer, the interval may become twice the expected value depending on the timing at which the compare register is written.

In addition, when the timer output is being used, a waveform shorter than expected or twice the frequency expected is output if the above conditions are met.

[Workaround]

Implement one of the following workarounds when rewriting the compare register (CR90) of the 16-bit timer (timer 90)

(1) Write writing using 8-bit access

Write to the 16-bit CR90 register by writing the highest byte first, then the lowest byte. At this time, be sure to clear the interrupt request flag (TMIF90) before enabling timer interrupts and timer output inversion.

After writing to CR90, do not set TOC90 until at least half a cycle of the count clock has passed since the start of an interrupt. (The signal INTTM90 is high for a period of half a cycle of the count clock after an interrupt is generated. If TOC90 is set within this period, the output is inverted.)

<Program example>

(When count clock = 64/fx, CPU clock = fx)

```
TM90_VCT:           SET1    TMMK90      ; Disable timer interrupts (6 clock pulses)
                    CLR1    TOC90       ; Disable timer output inversion (6 clock pulses)
                    MOV     A,#xxH      ; Set value to write highest byte (6 clock pulses)
  32 clock          MOV     !0FF17H,A   ; Write higher byte of CR90 (8 clock pulses)
  pulses in total   MOV     A,#xxH      ; Set value to write lowest byte (6 clock pulses)
                    MOV     !0FF16H,A   ; Write lowest byte of CR90 (8 clock pulses)
                    CLR1    TMIF90      ; Clear interrupt request flag (6 clock pulses)
                    CLR1    TMMK90      ; Enable interrupts (6 clock pulses)
                    SET1    TOC90       ; Enable output inversion
```

(2) When writing using 16-bit access

After the instruction (MOVW CR90, AX) that writes to CR90, be sure to insert a wait of at least 1 clock cycle of the count clock before enabling timer interrupts and timer output inversion.

<Program example>

(When count clock = 64/fx, CPU clock = fx)

```
SET1        TMMK90      ; Disable timer interrupts
CLR1        TOC90       ; Disable timer output inversion
MOVW        AX,#xxH     ; Set value to write CR90
MOVW        CR90,AX     ; Write CR90
NOP                     ;
NOP                     ;
 :                      ; 32 NOPs (64/fx wait)
NOP                     ;
NOP                     ;
CLR1        TMIF90      ; Clear interrupt request flag
CLR1        TMMK90      ; Enable interrupts
SET1        TOC90       ; Enable output inversion
```

Also, be sure to disable timer interrupts and output inversion before writing CR90.

No.7 Restriction on UART

[Description]

The LSB should be read first in UART reception. Under certain conditions, however, the MSB of the receive data is read first.

[Workaround]

(1) Read the receive data when RXE = 1.

(2) If it is necessary to use the receive data before changing RXE from 1 to 0, read the receive data first, after the receive interrupt is generated, and then change RXE.

(3) When changing RXE from 1 to 0 before reading the receive data, insert a wait equivalent to 1 or more cycles of the source clock selected by BRGC before changing RXE.

```
MOV    BRGC,#00H    ; A wait equivalent to 1 cycle of the source clock selected by BRGC00 is
                    ; necessary. (e.g. 0.4 μs @ fx = 5 MHz)
```

```
Insert a wait according to the selected source clock (see the examples below.)
```

```
CLR1   RXE          ;
MOV    A,RXB        ; Read the receive data.
```

(Example)

♦ When 1 source clock cycle ≥ 3.2 μs is selected

```
        MOV   A,#01H ; 6 clocks (1.2 μs @ fx = 5 MHz)
LABEL:  DEC   A      ; 4 clocks (0.8 μs @ fx = 5 MHz)
        BNZ   BNZ    LABEL ; 6 clocks (1.2 μs @ fx = 5 MHz)
```

♦ When 1 source clock cycle ≤ 3.2 μs is selected

```
        NOP          ; 2 clocks (0.4 μs @ fx = 5 MHz)
```

(4) When changing RXE from 1 to 0 without inserting a wait equivalent to 1 or more cycles of the source clock selected by BRGC, read the receive data when RXE = 0 and DIR = 1.

```
SET1   DIR          ; Set the DIR flag to LSB-first.
CLR1   RXE          ;
MOV    A,RXB        ; Read the receive data.
```

No.7 Restriction on 8-bit timer (3)

(1) Error upon timer start

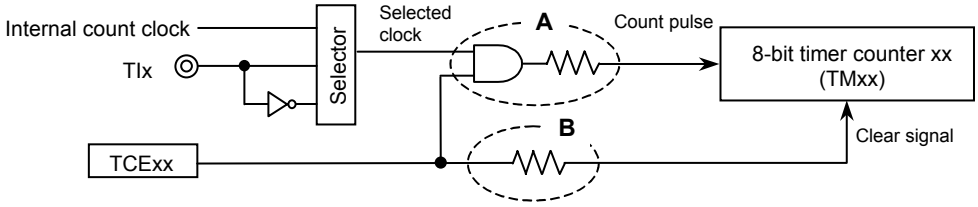
When the 8-bit timer/event counter is used, an error of up to 1.5 clocks is generated after the timer has been started until a match signal is generated. This is because a rising edge is detected when the timer is started while the selected clock is high, which results in the count value getting incremented.

[Details of restriction]

The basic timer configuration applicable to this restriction is shown in the figure below.

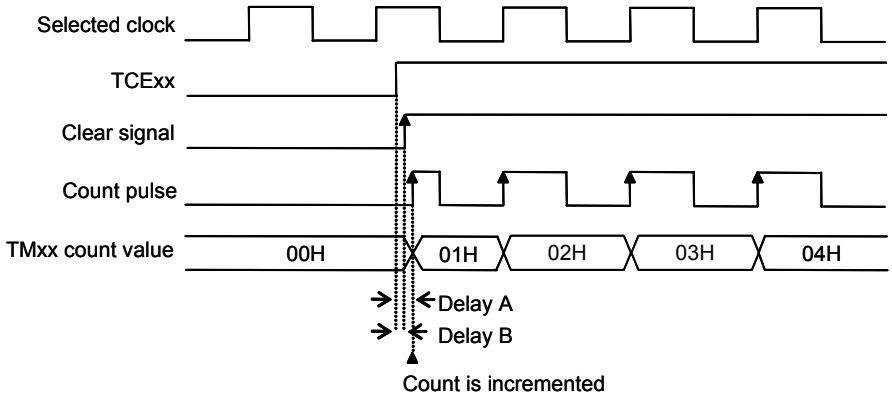
- The selected clock and the count operation flag (TCExx) are ANDed, and the result is supplied to the counter
- The TMxx counter value is cleared when the count operation flag (TCExx) is set

Basic Timer Configuration



If a count pulse is delayed for the clear signal (“delay A > delay B”) as the result of setting the count operation flag (TCExx) while the selected clock is high, the TMxx count value is incremented. An error of up to 1.5 clocks is generated.

Case when an error of up to 1.5 clocks is generated



Remark See Basic Timer Configuration above for details of the selected clock, TCE, clear signal, and count pulse.

(2) Restriction when an external event is selected for the timer count clock

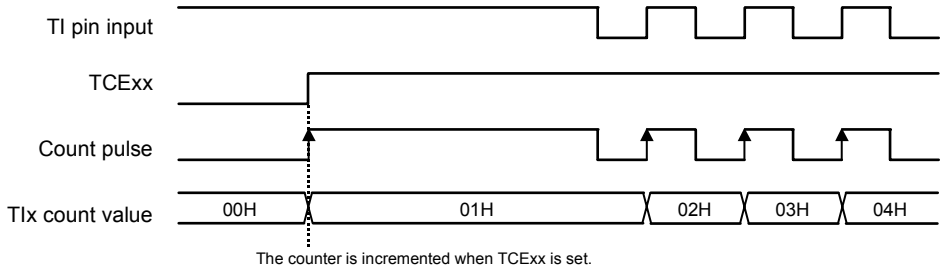
If an external event (TIx pin input) is selected for the timer count clock when the 8-bit timer/event counter is used, the count may be incremented after the count operation flag (TCExx) is set, depending on the status of the valid edge and TI pin.

[Details of restriction]

In the case of “delay A > delay B” as described in (1), the TMxx count value is incremented if any of the following conditions is satisfied.

- The count operation flag (TCExx) is set while the TI pin is high and the rising edge is selected.
- The count operation flag (TCExx) is set while the TI pin is low and the falling edge is selected.

Count Operation When Timer Is Started
While TI Pin Is High and Rising Edge Is Selected



[Workaround]

Make allowances for 1-count error in the counter, or implement one of workarounds (1) and (2) below.

Workaround (1)

- Always start the timer while the TI pin is low if the rising edge is selected.
- Always start the timer while the TI pin is high if the falling edge is selected.

Workaround (2)

- Save the count value upon timer start in the control register. When reading the count value, subtract this value from the count value saved in the control register and use the result as a true count value.