7544 Group
Timer A Operation (Event Counter Mode)

1. Abstract

The following article introduces and shows an application example of event counter mode of timer A.

2. Introduction

The explanation of this issue is applied to the following condition:
Applicable MCU: 7544 Group

3. Contents

Outline: The frequency of the pulse which is input to the $P0_0/CNTR_1$ pin ("H" active) is measured by the number of events in a certain period.

Specifications: The count source of timer A is input from the $P0_0/CNTR_1$ pin, and the timer A starts counting the count source. Clock ($f(X_{IN}) = 8$ MHz) is divided by timer X to detect 1 ms. The frequency of the pulse input to the $P0_0/CNTR_1$ pin is calculated by the number of events counted within 1 ms.

Operation clock: $f(X_{IN}) = 8$ MHz, high-speed mode

3.1 Example of Measurement Method of Frequency

Figure 1 shows an example of measurement method of frequency.

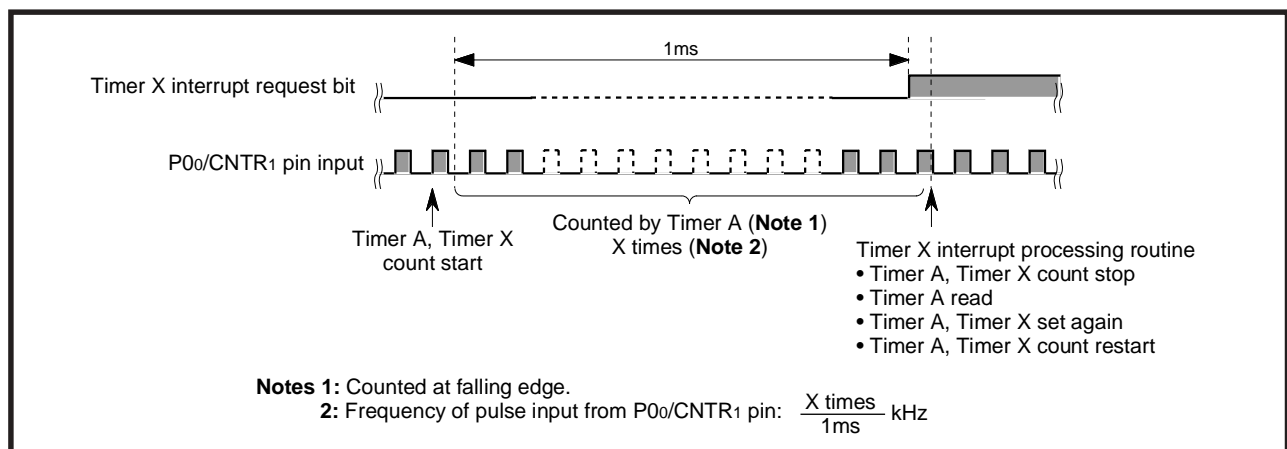**Figure 1 Example of measurement method of frequency**

3.2 Example of Control Procedure

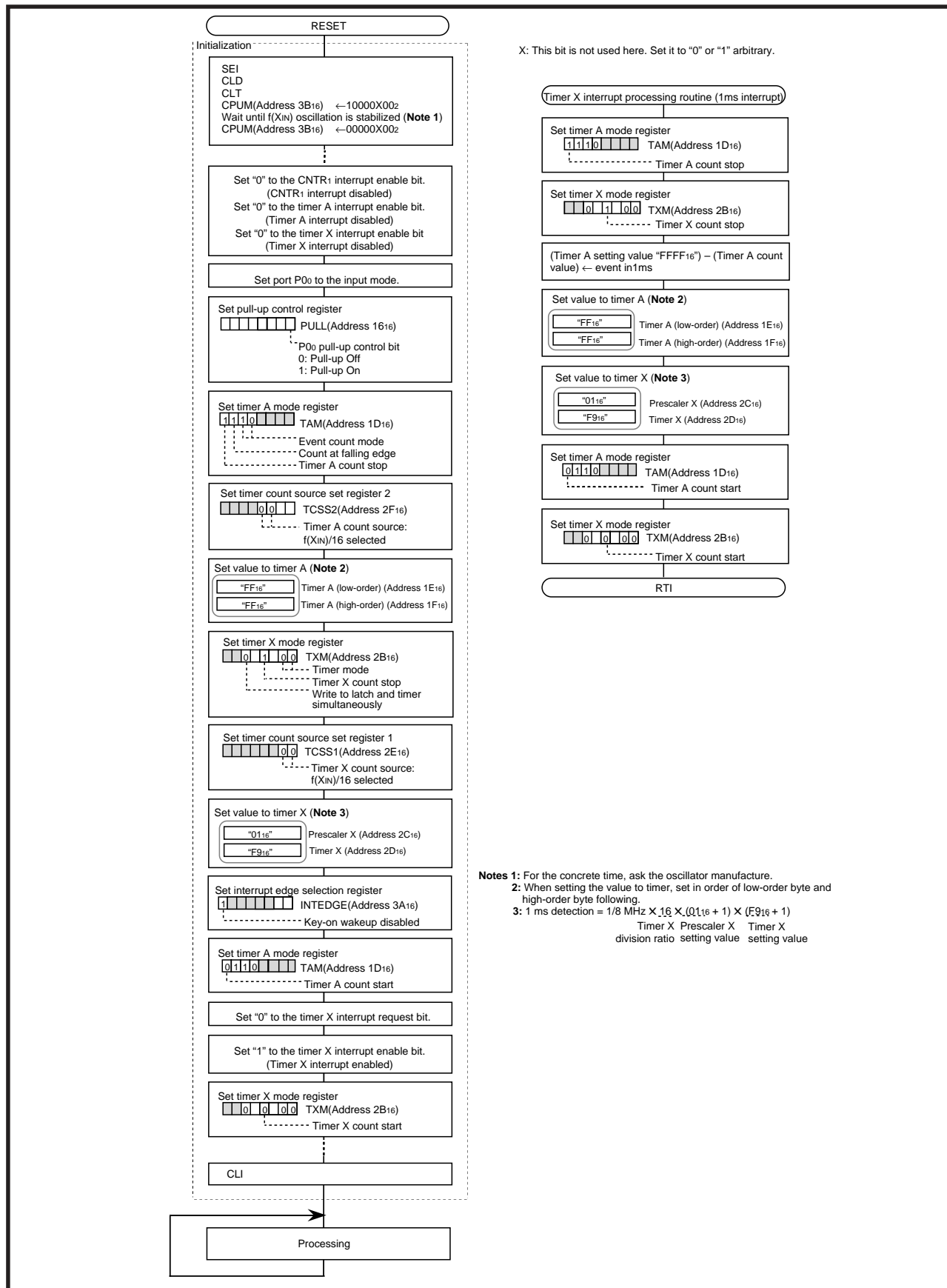Figure 2 shows an example of control procedure.

RESET

Initialization

SEI
CLD
CLT
CPUM(Address $3B_{16}$) ← $10000X00_2$
Wait until $f(X_{IN})$ oscillation is stabilized (**Note 1**)
CPUM(Address $3B_{16}$) ← $00000X00_2$

Set "0" to the $CNTR_1$ interrupt enable bit.
($CNTR_1$ interrupt disabled)
Set "0" to the timer A interrupt enable bit.
(Timer A interrupt disabled)
Set "0" to the timer X interrupt enable bit
(Timer X interrupt disabled)

Set port $P0_0$ to the input mode.

Set pull-up control register
PULL(Address 16_{16})
P0₀ pull-up control bit
0: Pull-up Off
1: Pull-up On

Set timer A mode register
1 1 1 0   TAM(Address $1D_{16}$)
Event count mode
Count at falling edge
Timer A count stop

Set timer count source set register 2
0 0   TCSS2(Address $2F_{16}$)
Timer A count source:
$f(X_{IN})/16$ selected

Set value to timer A (**Note 2**)
"FF₁₆"   Timer A (low-order) (Address $1E_{16}$)
"FF₁₆"   Timer A (high-order) (Address $1F_{16}$)

Set timer X mode register
0 1 0 0   TXM(Address $2B_{16}$)
Timer mode
Timer X count stop
Write to latch and timer
simultaneously

Set timer count source set register 1
0 0   TCSS1(Address $2E_{16}$)
Timer X count source:
$f(X_{IN})/16$ selected

Set value to timer X (**Note 3**)
"01₁₆"   Prescaler X (Address $2C_{16}$)
"F9₁₆"   Timer X (Address $2D_{16}$)

Set interrupt edge selection register
1   INTEDGE(Address $3A_{16}$)
Key-on wakeup disabled

Set timer A mode register
0 1 1 0   TAM(Address $1D_{16}$)
Timer A count start

Set "0" to the timer X interrupt request bit.

Set "1" to the timer X interrupt enable bit.
(Timer X interrupt enabled)

Set timer X mode register
0 0 0 0   TXM(Address $2B_{16}$)
Timer X count start

CLI

Processing

X: This bit is not used here. Set it to "0" or "1" arbitrary.

Timer X interrupt processing routine (1ms interrupt)

Set timer A mode register
1 1 1 0   TAM(Address $1D_{16}$)
Timer A count stop

Set timer X mode register
0 1 0 0   TXM(Address $2B_{16}$)
Timer X count stop

(Timer A setting value "FFFF₁₆") − (Timer A count value) ← event in1ms

Set value to timer A (**Note 2**)
"FF₁₆"   Timer A (low-order) (Address $1E_{16}$)
"FF₁₆"   Timer A (high-order) (Address $1F_{16}$)

Set value to timer X (**Note 3**)
"01₁₆"   Prescaler X (Address $2C_{16}$)
"F9₁₆"   Timer X (Address $2D_{16}$)

Set timer A mode register
0 1 1 0   TAM(Address $1D_{16}$)
Timer A count start

Set timer X mode register
0 0 0 0   TXM(Address $2B_{16}$)
Timer X count start

RTI

Notes 1: For the concrete time, ask the oscillator manufacture.
2: When setting the value to timer, set in order of low-order byte and high-order byte following.
3: 1 ms detection = $1/8$ MHz $\times$ 16 $\times$ (01_{16} + 1) $\times$ ($F9_{16}$ + 1)

Timer X   Prescaler X   Timer X
division ratio   setting value   setting value

Figure 2 Example of control procedure

4. Sample Programming Code

```
[Reset Start ••• Main Routine Process]

RESET:
        SEI                     ; Interrupt disable
        CLD
        CLT
;
        LDX  #$FF               ; Set stack bottom
        TXS
;
        LDM  #%10000000,CPUM    ; Set CPU mode register
;
; Wait f(XIN) oscillation stabilizing time
;
        LDM  #%00000000,CPUM    ; Set CPU mode register
;
        LDA  #0
        LDX  #>RAM_top
RAM_clear:   STA  $00,X
        INX
        BNE  RAM_clear
;
        CLB  6,ICON1            ; CNTR1 interrupt control disable
        CLB  2,ICON2            ; TimerA interrupt control disable
        CLB  7,ICON1            ; TimerX interrupt control disable
;
        LDM  #%00000000,P0D     ; Set Port P0 direction register
;
        CLB  0,PULL             ; Port P00 Pull_up off
;
        LDM  #%11100000,TAM     ; Set Timer A mode register
;
        LDM  #%00000000,TCSS2   ; Set Timer count source set register 2
;
        LDM  #$FF,TAL           ; Set Timer A (low-order)
        LDM  #$FF,TAH           ; Set Timer A (high-order)
;
        LDM  #%00001000,TXM     ; set timer X mode register
        LDM  #%00000000,TCSS1   ; Set Timer count source set register 1
;
        LDM  #$01,PREX          ; Set Prescaler X
        LDM  #$F9,TX            ; Set Timer X
;
        LDM  #%10000000,INTEDGE ; Set Interrupt edge selection register
;
        LDM  #%01100000,TAM     ; Set Timer A mode register
;
        CLB  7,IREQ1            ; TimerX interrupt request clear
;
        SEB  7,ICON1            ; TimerX interrupt enable
;
        LDM  #%00000000,TXM     ; Set Timer X mode register
;
        CLI
;
__MAIN:

        BRA  __MAIN
;
```

Figure 3 Sample Programming Code (1)

[Timer X Interrupt Process**]**

```
__int_TimerX:
        CLD
        CLT
        PHA
;
        SEB  7,TAM              ; stop timer A count
        SEB  3,TXM              ; stop timer X count
;
        SEC
        LDA  #$FF
        SBC  TAL
        STA  RESULT+0
        LDA  #$FF
        SBC  TAH
        STA  RESULT+1

        LDM  #$FF,TAL
        LDM  #$FF,TAH
;
        LDM  #$01,PREX          ; Set Prescaler X
        LDM  #$F9,TX            ; Set Timer X
;
        CLB  7,TAM              ; start timer A count
        CLB  3,TXM              ; start timer X count
;
        PLA
        RTI
;
```

Figure 4 Sample Programming Code (2)

5. Reference

Data Sheet
7544 Group Data sheet
7544 Group Data sheet (QzROM Version)

Before using this manual, please visit our website to verify that this is the most updated document available.

Renesas Technology Corporation Semiconductor Home Page
http://www.renesas.com/en/740

E-mail Support
E-mail: support_apl@renesas.com

| | | Description | |
| REVISION HISTORY | | 7544 Group Timer A Operation (Event Counter Mode) | |

| Rev. | Date | Description | |
		Page	Summary
1.00	Apr 01, 2003	-	First Edition issued
2.00	Nov 12, 2004	4-5	Sample Programming Code added.