Understanding the NS32CG160 Timers

National Semiconductor
Application Note 701
Zeev Bikowsky
July 1990

INTRODUCTION

The NS32CG160 contains a set of three programmable timers. Each timer can operate in one of three optional modes. The NS32CG160 timers are very important for real time application. Integrating them on-chip reduces the chip count in the user board.

This application note describes how to program the NS32CG160 internal timers to work with the three basic modes of operation. It contains an example of one of many ways a user can program the three timers. This example and variations on it may be used when programming the timers for a specific application.

TIMER DESCRIPTION

The NS32CG160 provides three on-chip timer blocks. Since these blocks are identical, the descriptions that follow are equally applicable to any of them. Each timer block consists of a 16-bit counter (TC), a control register (TCNTL) and two support registers (TRCA and TRCB). Two external signals (TXA and TXB) for each block handle all the interactions with external logic.

Following are the important fields of the TCNTL:

IENB Interrupt Enable Bit B. When set to 1, enables the interrupts from IPFB.

IPFB Interrupt Pending Flag B.

IENA Interrupt Enable Bit A. When set to 1, enables the interrupts from IPFA, if Mode 3 is selected, it enables the interrupts from TCS as well.

IPFA Interrupt Pending Flag A.

TCS Timer Control and Status. In mode 1 or 2, this bit is used to start and stop the timer. The timer starts when TCS is 1. In mode 3, TCS is the underflow interrupt pending flag.

TMC Timer Mode Control. This three-bit field selects the timer mode of operation (see Table I).

PRC Prescaler Control. Used only in modes 1 and 3. PRC controls the frequency of the timer input clock (TCLK). When PRC = 0, TCLK = CTTL/8, when PRC = 1, TCLK = CTTL/4096.

Each timer can operate in one of three modes (Table I):

1. Processor Independent
2. External Event Counter
3. Input Capture

Following is a detailed description of the three timer processor modes.

PROCESSOR INDEPENDENT MODE (Mode 1)

This mode can be used to generate an output signal with minimal software intervention. The software only needs to define the ON and OFF times for the waveform to be generated. Once started, the timer will generate a periodic waveform without further intervention, except when the parameters need to be updated.

In this mode the time counts down at the Tclk rate. Upon the occurrence of every underflow the timer is alternately reloaded with the contents of one of the support registers TRCA or TRCB. The first timer underflow causes a reload from the TRCA register. Reloads from subsequent underflows alternate from the two support registers, starting with TRCB. Each underflow toggles the TXA output pin.

Timer underflows are alternately latched into the IPFA and IPFB flags. The software is responsible for resetting these flags. Two enable bits, IENA and IENB, allow timer underflow interrupts to be enabled or disabled. Setting the IENA bit will cause an interrupt when a timer underflow causes a reload from TRCA, while setting IENB will cause an interrupt when the reload is from TRCB.

The IENA and IENB bits give the user the flexibility to enable or disable interrupts on either or both edges of the timer output waveform.

EXTERNAL EVENT COUNTER MODE (Mode 2)

This mode is similar to the processor independent mode. The difference being that the timer is clocked by the rising edge of the signal applied on the TXB input pin.

TABLE I. Timer Modes

TMC Field	Timer Mode	Interrupt A Source	Interrupt B Source	Timer Counts On
000	IDLE			
001	MODE 1 (Processor Independent)	Autoreload TRCA	Autoreload TRCB	Tclk
010	MODE 2 (External Event Counter)	Autoreload TRCA	Autoreload TRCB	TXB Rising Edge
100	MODE 3 (Input Capture) Captures On: TXA Rising Edge TXB Rising Edge	TXA Rising Edge or Timer Underflow	TXB Rising Edge	Tclk

  RRD-B30M75/Printed in U. S. A.

INPUT CAPTURE MODE (Mode 3)

This mode allows the user to perform precise measurements of external frequencies and to time external events.

The timer constantly runs at the Tclk rate. The registers TRCA and TRCB act as capture registers and are controlled by external signals applied on the TXA and TXB pins.

The timer value gets copied into the corresponding register when a trigger event is signaled on either TXA or TXB. A trigger event is specified as a rising edge of the input signal.

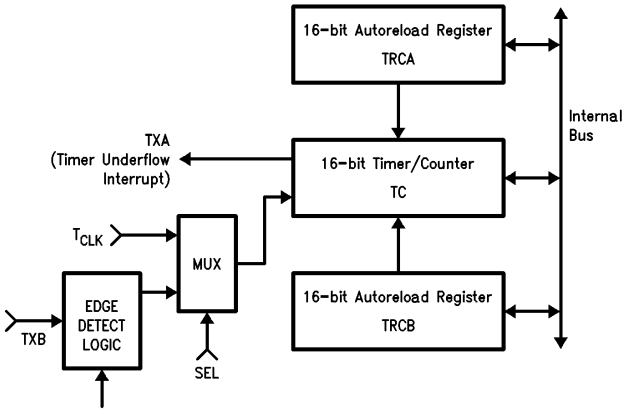
Trigger events can be programmed to generate interrupts. The occurrence of a trigger event on either TXA or TXB will be latched to IPFA or IPFB respectively. Interrupts are controlled by the setting of IENA and IENB.

Timer underflows can also generate interrupts. Since the underflow interrupt pending flag TCS has a different function in the other timer modes, the software should always reset it when the Input Capture Mode is selected. Timer underflow interrupts are also enabled by the IENA bit. Therefore, when IENA is set to 1 and an interrupt occurs, both IPFA and TCS should be checked to determine the origin of the interrupt.

PROGRAMMING THE TIMER MODES

These sections present three examples of how to program the timers to function in different modes. Example 1 shows how Timer 0 may be programmed to function in Mode 1 (Processor Independent). Example 2 shows how Timer 1 may be programmed to function in Mode 2 (External Event Counter). Example 3 shows how Timer 2 may be programmed to function in Mode 3 (Input Capture).

Note that these examples are not restricted to the specific instances described here. Thus the code for programming Timer 0 to function in Mode 1 may also be used to program Timer 1 and 2 to function in the same mode. The same holds true for the code for programming the other modes.

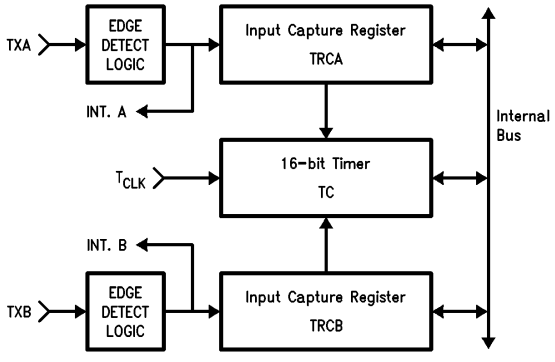


TL/EE/10870–1

Note 1: Tclk = CTTL/8 if TCNTL PRC - Bit = 0.
Tclk = CTTL/4096 if TCNTL PRC - Bit = 1.

Note 2: Specifying a counter value of 0 yields a count of 2^{**16}.

FIGURE 1. Timer Block Diagram for the Processor Independent and External Event Counter Modes



TL/EE/10870–2

FIGURE 2. Timer Block Diagram for the Input Capture Mode

Example 1: Programming Processor Independent Mode

This example programs Timer 0 to function in Mode 1 (Processor Independent).

The code below loads values to TC, TRCA and TRCB to generate a pulse width modulation where high time is 8 cycles and low time is 16 cycles (a precise description of high__0 and low__0 for this example follows the code listing).

The timer begins executing after the TCNTL register is loaded. When this happens, the TXA0 pin toggles from high to low.

Note that this example causes interrupts to be disabled because IENA and IENB are cleared in the TCNTL register.

```
.set high_0, h'00000001            # high time will be 8 cycles
.set low_0, h'00000002             # low time will be 16 cycles
.set tc_0, h'fffff810              # timer 0 counter
.set trca_0, h'fffff814            # timer 0 reload register a
.set trcb_0, h'fffff818            # timer 0 reload register b
.set tcnt_0, h'fffff81c            # timer 0 control register
                                   #
.text
start_0:                           # programming timer 1
    movw $high_0 ,@tc_0            # initialize timer counter tc
    movw $low_0 ,@trca_0           # initialize trca register
    movw $high_0 ,@trcb_0          # initialize trcb register
    movw $h'30 ,@tcnt_0            # set control register to mode 1
                                   # no interrupts, timer frequency
                                   # is CTTL/8
```

A Description of the high-low parameters:

high__0 Specifies the value which will be loaded to TRCB register.

This value specifies the high time that TXA0 will stay in.

low__0 Specifies the value which will be loaded to TRCA register.

This value specifies the low time that TXA0 will stay in.

Example 2: Programming External Event Counter Mode

This example programs Timer 1 to function in Mode 2 (External Event Counter).

The code below loads values for the TC, TRCA and TRCB to generate a pulse width modulation. This mode is similar to the processor independent mode. The only difference is that the timer is clocked by the rising edge of the signal applied on the TXB1 input pin. A precise description of high__1 and low__1 for this example follows the code listing.

The timer begins executing after the TCNTL register is loaded. When this happens, the TXA1 pin toggles from high to low, the high time becomes one rising edge of TXB1 and the low time becomes two rising edges of TXB1.

Note that this example causes interrupts to be disabled because IENA and IENB are cleared in the TCNTL register.

```
.set high_1, h'00000001            # high time 1 rising edge
.set low_1, h'00000002             # low time 2 rising edges
.set tc_1, h'fffff820              # timer 1 counter
.set trca_1, h'fffff824            # timer 1 reload register a
.set trcb_1, h'fffff828            # timer 1 reload register b
.set tcnt_1, h'fffff82c            # timer 1 control register

start_1:                           # programming timer 1

    movw $high_1 ,@tc_1            # initialize timer counter tc
    movw $low_1 ,@trca_1           # initialize trca register
    movw $high_1 ,@trcb_1          # initialize trcb register
    movw $h'50 ,@tcnt_1            # Set control register to mode 2
                                   # no interrupts, timer frequency
                                   # is external
```

A Description of the high-low parameters:

high__1 Specifies the value which will be loaded to TRCB register.

This value is the number of rising edges of TXB1.

It specifies the high time that TXA1 will stay in.

low__1 Specifies the value which will be loaded to TRCA register.

This value is the number of rising edges of TXB1.

It specifies the low time that TXA1 will stay in.

Example 3: Input Capture Mode

This example programs Timer 2 to function in Mode 3 (Input Capture).

The code below initializes values for the TC, TRCA and TRCB. On each rising edge of TXA2, TRCA will be loaded with the content of the TC. On each rising edge of TXB2, TRCB will be loaded with the content of the TC.

The timer begins executing after the TCNTL register is loaded.

Also note that this example causes interrupts to be disabled because IENA and IENB are cleared in the TCNTL register.

```
.set tc_2, h'fffff830           # timer 2 counter
.set trca_2, h'fffff834         # timer 2 reload register a
.set trcb_2, h'fffff838         # timer 2 reload register b
.set tcnt_2, h'fffff83c         # timer 2 control register

start_2:                        # programming timer 2

    movw $h'100 ,@t_2           # initialize timer counter tc
    movw $h'0 ,@trca_2          # initialize trca register
    movw $h'0 ,@trcb_2          # initialize trcb register
    movw $h'80 ,@tcnt_2         # set control register to mode 3
                                # no interrupts, timer frequency
                                # is CTTL/8
```


National Semiconductor Corporation
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: 1(800) 272-9959
TWX: (910) 339-9240

National Semiconductor GmbH
Livry-Gargan-Str. 10
D-82256 Fürstenfeldbruck
Germany
Tel: (81-41) 35-0
Telex: 527649
Fax: (81-41) 35-1

National Semiconductor Japan Ltd.
Sumitomo Chemical
Engineering Center
Bldg. 7F
1-7-1, Nakase, Mihama-Ku
Chiba-City,
Ciba Prefecture 261
Tel: (043) 299-2300
Fax: (043) 299-2500

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductores Do Brazil Ltda.
Rue Deputado Lacorda Franco
120-3A
Sao Paulo-SP
Brazil 05418-000
Tel: (55-11) 212-5066
Telex: 391-1131931 NSBR BR
Fax: (55-11) 212-1181

National Semiconductor (Australia) Pty, Ltd.
Building 16
Business Park Drive
Monash Business Park
Nottinghill, Melbourne
Victoria 3168 Australia
Tel: (3) 558-9999
Fax: (3) 558-9998