HC08 Timer with an External Clock Source

by: Alban Rampon
   8/16-bit Division
   East Kilbride

Introduction

All HC08 microcontrollers include at least one timer module (TIM). This module is very useful for generating or capturing time-dependent signals.

The data sheet describes what the timer module is capable of and how to use it.

This application note provides further information on how the timer behaves with an external clock source. A typical application that could use the timer in such a way would be a frequency counter.

As an aid to understanding, an example with sample code is provided in the appendix at the end of this document.

Functional Description

Features of the TIM can include

- Two to six input capture/output compare channels
  - Rising-edge, falling-edge or any-edge input capture trigger

- – Set, clear or toggle output compare action
- Buffered and unbuffered pulse width modulation (PWM) signal generation
- Programmable TIM clock input
  - – Seven frequency internal bus clock prescaler selection
  - – External TIM clock input (4 MHz maximum frequency)
- Free-running or modulo up-count operation
- Toggle any channel pin on overflow
- TIM counter Stop and Reset bits

Block Diagram

Figure 1 shows the structure of the TIM, taken from an MC68HC908GZ60 (two channels represented out of the six available). The central component of the TIM is the 16-bit counter, which can operate as a free-running counter or a modulo up-counter. The TIM counter provides the timing reference for the input capture and output compare functions. The TIM counter modulo registers, TMODH:TMODL, control the modulo value of the timer counter. Software can read the TIM counter value at any time without affecting the counting sequence.

In the case of the HC908GZ60, the six timer channels are programmable independently as input capture or output compare channels. In other devices, the number of channels may vary from two to six; however, the rest of the module remains identical.
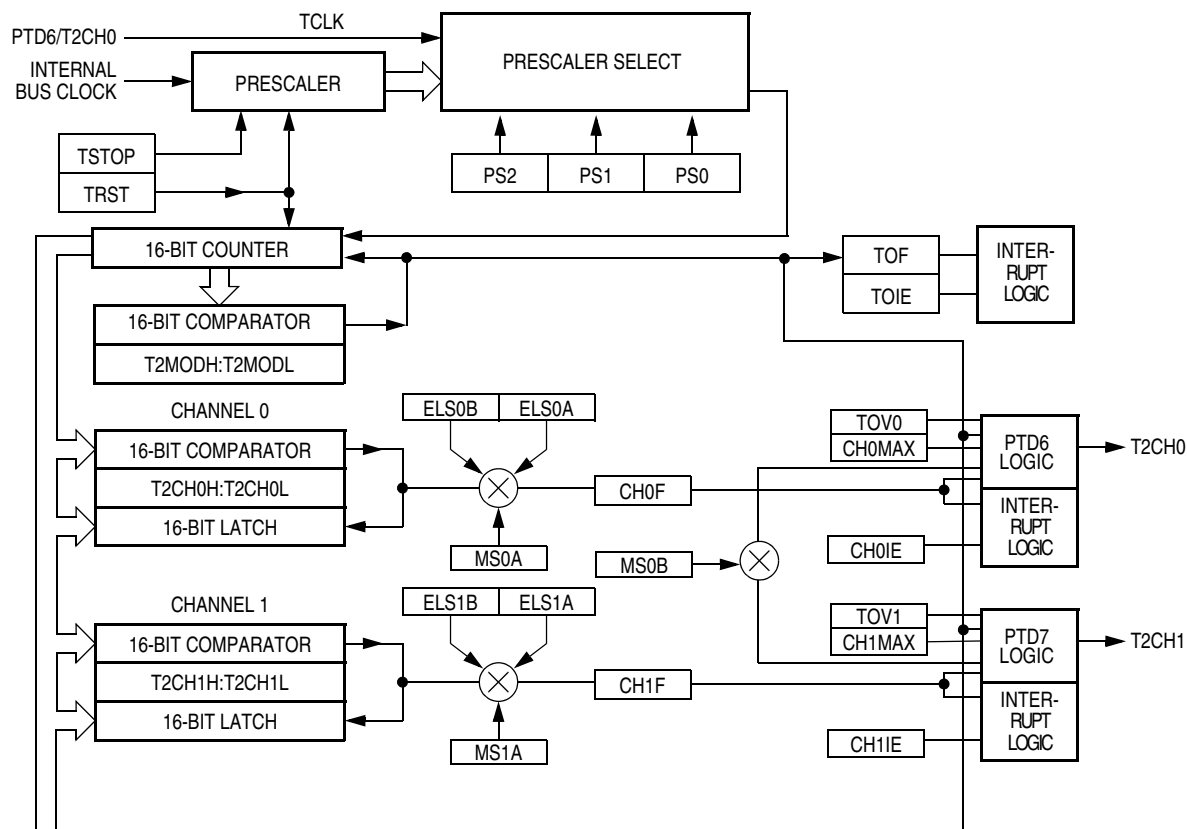
Figure 1. HC08 Generic Timer Module Block Diagram

Timer Counter Prescaler

The timer clock source can be one of the seven prescaler outputs or the timer clock pin (PTD6/T2CH0 on HC908GZ60). The prescaler generates seven clock rates from the internal bus clock.

The prescaler select bits, PS[2:0], in the timer status and control register select the timer clock source.

Table 1. Timer Prescaler Values[1]

PS[2:0]	TIM2 Clock Source
000	Internal Bus Clock ÷ 1
001	Internal Bus Clock ÷ 2
010	Internal Bus Clock ÷ 4
011	Internal Bus Clock ÷ 8
100	Internal Bus Clock ÷ 16
101	Internal Bus Clock ÷ 32
110	Internal Bus Clock ÷ 64
111	T2CH0

NOTES:
   1. Not all timers allow all possible values; refer to the device data sheet.

Using an External clock

On some timer modules, it is possible to count using an external signal rather than the bus clock or the bus clock divided by 2, 4, … up to 64. In this case, the prescaler PS[2:0] must be set to %111, i.e., TxSC_PS0 = 1, TxSC_PS1 = 1, and TxSC_PS2 = 1. (Refer to the data sheet for a full description of the timer status and control register.)

On the HC908GZ60, PTD6/T2CH0 is an external clock input that can be selected as the clock source for the TIM counter, instead of selecting the divided internal bus clock.

The minimum TCLK pulse width, $TCLKLow_{MIN}$ or $TCLKHigh_{MIN}$, is:

$$\frac{1}{BusFrequency} + t_{SU} \hspace{4cm} \text{Eq. 1}$$

The maximum TCLK frequency is the lower of the following values:

$$4 \text{ MHz or } \frac{BusFrequency}{2} \hspace{4cm} \text{Eq. 2}$$

NOTE
PTD6/T2CH0 is available as a general-purpose I/O pin or ADC channel when not used as the TIM clock input. When the PTD6/T2CH0 pin is the TIM clock input, it is an input regardless of the state of the DDRD4 bit in data direction register D.

This mode can be used to build a frequency counter. The count is taken periodically with a programmable interrupt timer (PIT) or time base module (TBM), and the frequency is calculated (i.e., the number of counts per second), depending on the counting period. Both PIT and TBM are periodic interrupt generator modules.

The timer block diagram shows the timer reset signal, TxSC_TRST, linked to the prescaler. The reset is synchronous with the timer counter clock.

The external square wave counted cannot have its edge occurrences compared with or linked to those of the bus clock.

With a bus clock running at 32 MHz, the timer reset command will be present for $1/(32 \times 10^6) = 31.25$ ns. If the timer clock does not receive an edge within this time, the timer will NOT see the reset request properly and, therefore, the counter value might not be zero when the following count starts. Consequently, if the counter is not reset properly, the next count will be wrong.

To avoid this potential risk, the reset should be asynchronous with the external signal.

By switching the clock source prescaler to the internal bus clock before resetting the counter, the reset command is no longer dependent on the external signal. The clock source should be switched back to the external clock source before the counter restarts.
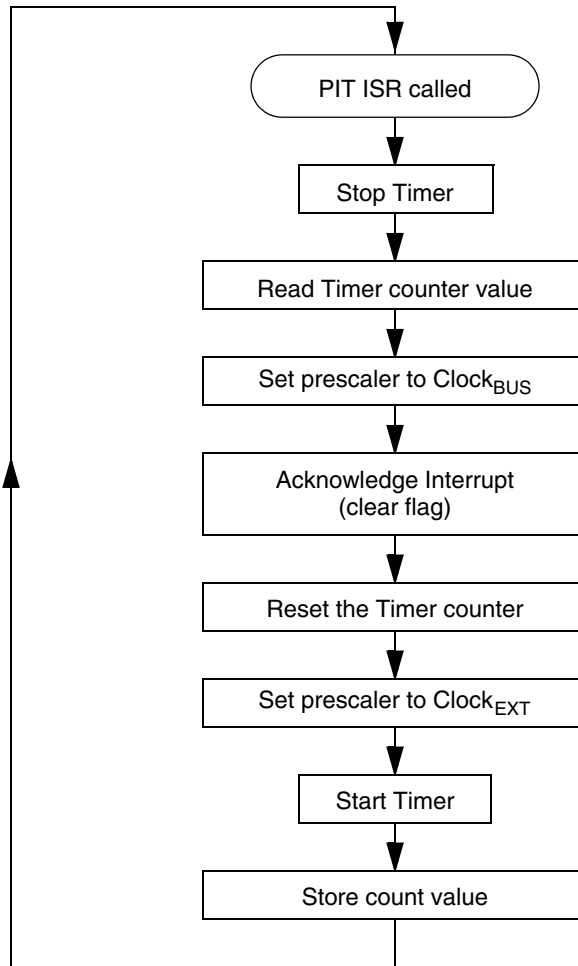
The appendix at the end of this document contains an example (flowchart and software) that illustrates a possible way to implement this recommendation.

In the example, a frequency is counted. The PIT is programmed to generate an interrupt every 0.5 seconds. The signal frequency is calculated by multiplying the result by two and storing it in an array. This gives the count per second, i.e., the frequency.

However, depending on the application, the user may find that the asynchronous reset presents a suitable solution.

APPENDIX

Program Flowchart

```
      ┌──────────────────────────────────────┐
      │                          ┌────────────────────────┐
      │                          │      PIT ISR called      │
      │                          └────────────────────────┘
      │                                      │
      │                          ┌────────────────────────┐
      │                          │       Stop Timer         │
      │                          └────────────────────────┘
      │                                      │
      │                     ┌─────────────────────────────┐
      │                     │    Read Timer counter value    │
      │                     └─────────────────────────────┘
      │                                      │
      │                     ┌─────────────────────────────┐
      │                     │  Set prescaler to Clock_BUS    │
      │                     └─────────────────────────────┘
      │                                      │
      │                     ┌─────────────────────────────┐
      │                     │    Acknowledge Interrupt       │
      │                     │         (clear flag)           │
      │                     └─────────────────────────────┘
      │                                      │
      │                     ┌─────────────────────────────┐
      │                     │    Reset the Timer counter     │
      │                     └─────────────────────────────┘
      │                                      │
      │                     ┌─────────────────────────────┐
      │                     │  Set prescaler to Clock_EXT    │
      │                     └─────────────────────────────┘
      │                                      │
      │                          ┌────────────────────────┐
      │                          │       Start Timer        │
      │                          └────────────────────────┘
      │                                      │
      │                     ┌─────────────────────────────┐
      │                     │      Store count value         │
      │                     └─────────────────────────────┘
      │                                      │
      └──────────────────────────────────────┘
```

Flowchart nodes in order:

- PIT ISR called
- Stop Timer
- Read Timer counter value
- Set prescaler to Clock$_{BUS}$
- Acknowledge Interrupt (clear flag)
- Reset the Timer counter
- Set prescaler to Clock$_{EXT}$
- Start Timer
- Store count value

(loops back to PIT ISR called)

HC08 Timer with an External Clock Source, Rev. 0

Sample Code

```
/****************************************************************************
*                               COPYRIGHT (c) FREESCALE SEMICONDUCTOR 2004
*                                               ALL RIGHTS RESERVED
* FILE NAME: AN2701_PIT_ISR.c
*
* PURPOSE: HC908AZ60A Sample Code for the Periodic Interrupt Timer ISR
*
****************************************************************************
****************************************************************************
**   THIS CODE IS ONLY INTENDED AS AN EXAMPLE FOR THE METROWERKS COMPILER AND **
**THE MMDS0508/EM08QA24 EVB AND HAS ONLY BEEN GIVEN A MIMIMUM LEVEL OF TEST. **
**   IT IS PROVIDED 'AS SEEN' WITH NO GUARANTES AND NO PROMISE OF SUPPORT.   **
****************************************************************************
****************************************************************************
*
* DOCUMENTATION SOURCE: EKB Apps
*
* TARGET DEVICE: HC908AZ60A (any HC08 with PIT)
*
* COMPILER: Metrowerks for HC08                     VERSION: v5.2.1149
*
* DESCRIPTION: This ISR can also be updated to work with a Timebase module (TBM)
*              present in other devices (i.e. GZ...)
*
* AUTHOR: A.RAMPON      LOCATION: EKB      LAST EDIT DATE: 15/08/04
*
* UPDATE HISTORY
* REV       AUTHOR       DATE        DESCRIPTION OF CHANGE
* ---       ------       ---------   --------------------
* 1.0       A.Rampon     11/08/04    Initial Release
****************************************************************************/
#ifndef __AN2701_PIT_ISR_C
#define __AN2701_PIT_ISR_C    /* if this H file not included, include */

#include "hc08az60a.h"

/**********************************************************************
Function Name : _PitISR
Engineer      : A.Rampon
Date          : 01/04/04
Parameters    : NONE
Returns       : NONE
Notes         : Interrupt service routine for PIT.
**********************************************************************/
#pragma TRAP_PROC
Void _PitISR(void) {

                /* Temporarily variable used to clear flag */
    unsigned char dummyRead;

                /* Stop TIMB */
    TSTOP_TB = 1;
                /* Take the counter value */
    RealCount = TBCNT;
```

HC08 Timer with an External Clock Source, Rev. 0

```
                          /* Set prescaler to internal bus clock */
    PS0_TB = 0;
    PS1_TB = 0;
    PS2_TB = 0;


                          /* load PSC: needed for clearing POF */
    dummyRead = PSC;
                          /* reset POF flag=PIT overflow flag bit*/
                          /* =   PIT_Interrupt-Acknowledge   */
    PSC&= ~0x80;
                          /* Clear prescaler and counter */
    TRST_TB = 1;


                          /* Set prescaler to external clock */
    PS0_TB = 1;
    PS1_TB = 1;
    PS2_TB = 1;


                          /* Start TIMB */
    TSTOP_TB = 0;

    /* Insert in Buffer */
    if (ResultBufferPtr<50)
    {
        ResultBuffer[ResultBufferPtr] = RealCount *2;
        ResultBufferPtr++;
    }
}
```

How to Reach Us:

USA/Europe/Locations not listed:
Freescale Semiconductor Literature Distribution
P.O. Box 5405, Denver, Colorado 80217
1-800-521-6274 or 480-768-2130

Japan:
Freescale Semiconductor Japan Ltd.
SPS, Technical Information Center
3-20-1, Minami-Azabu
Minato-ku
Tokyo 106-8573, Japan
81-3-3440-3569

Asia/Pacific:
Freescale Semiconductor H.K. Ltd.
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T. Hong Kong
852-26668334

Learn More:
For more information about Freescale
Semiconductor products, please visit
http://www.freescale.com

AN2784
Rev. 0, 10/2004

freescale™
semiconductor