Order this document
by EB274/D

Motorola Semiconductor Engineering Bulletin

EB274

Generating Interrupts on the Time Processor Unit

By Sharon Darley
Austin, Texas

General Information

Several steps must be followed to generate interrupts on a time processor unit (TPU) channel.

1. The first step is to store the starting address of the interrupt routine in an address in the CPU vector table.

   To calculate the appropriate address in the table, first choose a base vector number and write it to bits 7– 4 in the TPU interrupt configuration register (TICR). A base vector number is a hexadecimal number such as $6, $7, or $8. The base vector number is concatenated with the channel number to yield an interrupt vector number. For example, choosing a base vector number of $8 would assign interrupt vector $80 to channel 0, interrupt vector $81 to channel 1, interrupt vector $82 to channel 2, and so on through interrupt vector $8f to channel 15. Since this example uses channel 4, the interrupt vector is $84.

   The vector address is where the starting address of the interrupt routine is stored. It is determined from the vector number. For the MC68332, the vector address is calculated as four times the vector number plus the value in the vector base register. If the vector base register already has been initialized to $400 by CPU32Bug, then in this example the vector address is (4 x $84) +

$400, which is equal to $610. Thus, the starting address of the interrupt routine must be stored in location $610.

For the HC16, the vector address is calculated as two times the vector number, so the vector address in this example is 2 x $84, which is equal to $108.

2. The next step is to choose an interrupt level for the TPU module and store it in bits 10–8 in the TICR.

   The interrupt level must be a number between 1 and 7. The interrupt level determines the interrupt's priority. Level 7 has the highest priority, and level 1 has the lowest.

   Once an interrupt level has been chosen, the interrupt priority level bits must be modified to allow recognition of that level of interrupt. For the MC68332, these are bits 10–8 in the CPU status register. For the HC16, these are bits 7–5 in the condition code register. These bits must be set to a number that is lower than the interrupt level number. Interrupts at the same level or lower than the number in the CPU status register/condition code register will be masked out and will not be recognized by the CPU.

   Another parameter that must be set is the interrupt arbitration (IARB) field in the TPU module configuration register (TMCR). The number in the IARB field determines the interrupting module's priority when the CPU receives more than one interrupt request on the same interrupt priority level. Each interrupting module must have a unique IARB number between $01 and $0F.

3. The final step to enable an interrupt on a TPU channel is to set the interrupt enable bit for that particular channel in the channel interrupt enable register (CIER).

   This simply involves writing the channel's bit number to a 1.

   To clear interrupts on a TPU channel, clear any pending interrupts by negating the appropriate interrupt status flag in the channel interrupt status register (CISR). To do this, the flag must be read in its asserted state and then written back with a 0.

CPU32 Code

This is a short example that generates interrupts on channel 4.

```
*** Interrupt Initialization on Channel 4 ***
       move.w          #$0680,(TICR).l       ;Interrupt level = 6, base
                                             ;vector = $80
       move.l          #INT,($0610).l        ;start interrupt routine at label INT
       andi.w          #$f5ff,SR             ;allow interrupts on level 6 and
                                             ;above (assume reset condition of SR)
       ori.w           #$0005,(TMCR).l       ;set IARB field to $5
       ori.w           #$0010,(CIER).l       ;enable interrupts for channel 4

*** Interrupt Routine ***

INT    andi.w           #$ffef,(CISR).l      ;read and clear interrupt

**** Code for interrupt routine ****

       rte                                   ;return from exception

CPU16 Code

*** Interrupt Initialization on Channel 4 ***
       clrb
       tbzk
       ldz     #$0000                        ;use IZ for indexed offset
       ldd     #INT
       std     $0108,z                       ;start interrupt routine at INT
       ldd     TPUMCR
       ord     #$0005                        ;set IARB field
       std     TPUMCR
       ldd     #$0680
       std     TICR                          ;interrupt level 6, base vector=$80
       andp    #$ffaf                        ;allow interrupts level 6 & 7, assuming
                                              ;reset values in CCR
       ldd     CIER
       ord     #$0010
       std     CIER                          ;enable interrupts on channel 4

*** Interrupt Routine ***

INT    ldd     CISR                          ;read interrupt
       andd    #$ffef
       std     CIS                           ;clear interrupt

       rti
```

Engineering Bulletin

How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution, P.O. Box 5405, Denver, Colorado 80217.
1-800-441-2447 or 1-303-675-2140. Customer Focus Center, 1-800-521-6274

JAPAN: Motorola Japan Ltd.: SPD, Strategic Planning Office, 141, 4-32-1 Nishi-Gotanda, Shinagawa-Ku, Tokyo, Japan, 03-5487-8488

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd., Silicon Harbour Centre, 2 Dai King Street, Tai Po Industrial Estate,
Tai Po, New Territories, Hong Kong, 852-26629298

Mfax™, Motorola Fax Back System: RMFAX0@email.sps.mot.com; http://sps.motorola.com/mfax/;
TOUCHTONE, 1-602-244-6609; US and Canada ONLY, 1-800-774-1848

HOME PAGE: http://motorola.com/sps/

MOTOROLA