Hitachi America, Ltd.

TN-0170

Application Engineering
TechNote

Cristian Tomescu

Implementing a software interrupt on the H8/320

The H8/300 Microcontroller family offers high programming versatility for a wide range of applications by controlling the code execution through a variety of hardware-initiated interrupt requests. These requests can either be caused by external sources (via NMI and IRQ input lines), or by the internal peripheral modules within the microcontroller (such as SCI, Timers, A/D, etc.). The main obvious advantage of using the hardware interrupt mechanism is to allow the peripheral device to indicate its readiness for a course of action by asserting an interrupt request line, thus enabling the CPU to perform other tasks while the peripheral is busy. Moreover, in certain instances, a user may want to periodically access a subroutine within his/her program without running the risk that the flag bits in the CCR may be altered. Using the JSR instruction will push the PC-value into the stack but not the contents of the CCR; hence, the need for a software-initiated interrupt arises.

The H8/300 Family instruction set does not have a TRAP command to request an interrupt, but the programmer can implement it using a few instructions in a subroutine that the user can call from anyplace in his program. Also, a suitable location in the interrupt vector table must be used to contain the start address of the interrupt service routine. The H8/320 series are the only 8-bit devices that could "rent" a vector location for this purpose, namely the location of the Input Strobe Interrupt (ISI) generated from the Parallel Handshaking Interface module (if the user is not employing this feature). By pulling the Input Strobe pin (IS-) low while the ISI enable bit (ISIE) in the Handshake Control/Status Register (HCSR) is set will cause the current contents of the PC and CCR to be pushed into the stack, the Input Strobe Flag (ISF) in HCSR to be set, and the CPU to start executing instructions beginning at the address contained in the ISI vector table entry at H'000E. The user can utilize an assembly instruction such as MOV to toggle the output port line (to which the IS- line is multiplexed) from high-to-low, thus initiating the interupt. Regardless of the operating mode of the chip, an ISI interrupt will **always** be requested upon the conditions stated above when the CPU operates in the normal programming mode. Since the IS- pin will be toggled by the user's software, this option cannot be used in the software standby mode.

The following example shows one possible implementation of a software-generated interrupt delay routine on the H8/320. Prior to activating the interrupt, the following initializations are performed: P7(0) is configured to output, all interrupts are un-masked in CCR, and bit ISIE of HCSR is set to 1. Subroutine SOFT_INT, when called, turns off P7(0), and since the output is fed back to the input IS- line, the CPU will start executing the DELAY interrupt service routine. The source file is listed below:

```
;       This code implements a software-triggered interrupt
;       using the Parallel Handshake Interface input-strobe signal

HCSR    .equ    H'FFFE
P7_DDR  .equ    H'FFBC
P7_DR   .equ    H'FFBE

        .org    H'000E
        .data.w DELAY           ;interrupt service routine start address
        .org    H'100
        mov.w   #H'FF80,R7      ;load stackpointer
        ldc     #0,CCR          ;un-mask all interrupts
        mov.b   #H'01,R0L
        mov.b   R0L,@P7_DDR     ;P7(0) is an output pin
        mov.b   R0L,@P7_DR      ;P7(0) is at a high level
        bset    #6,@HCSR        ;enable input strobe interrupt
        nop
        nop
```

```
        bra     CONTINUE
        nop
SOFT_INT:
        mov.b   #H'00,R0L
        mov.b   R0L,@P7_DR      ;pull IS pin low to request interrupt
        nop
        rts
CONTINUE:
        (user code)
        jsr     SOFT_INT        ;call software interrupt
        (user code)
        bra     NEXT
DELAY:  inc.b   R0L             ;start delay interrupt routine
        cmp.b   #100,R0L
        bne     SOFT_INT
        mov.b   #H'02,R0L
        mov.b   R0L,@P7_DR      ;restore high level at IS-
        rte
NEXT:   (user code)
        .end
```


HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300