

DESIGNER'S NOTEBOOK



Setting Up and Simulating Interrupts on the TMS320C5x

Contributed by Helga Stevenson

Design Problem How do I set up an interrupt and then simulate it on a TMS320C5x DSP?

Solution The following example shows the code required to set up serial port interrupts on a 'C5x DSP, although it can be applied to any other interrupt.

The code shown in Figure 1 shows the interrupt service routine vector set-up for data received on the synchronous serial port and the initial vector "RINT" to jump to the start of the main code on reset.

```
.sect      "vectors"
B          BEGIN      ; reset vector jumps to label 'BEGIN'
.space     16*8        ; fills the next 8 words with 0s
                ; to locate RINT at the correct location
                ; in the interrupt vector table
RINT:      B          RECEIVE,* ,AR1 ; Serial port receive interrupt RINT.
```

Figure 1. ISR set-up

The first section of the program code should set up the interrupt service routine. For additional information, please refer to the *TMS320C5x User's Guide* (SPRU056).

Figure 2 details the steps for setting up the various registers. First, this enables the serial port receive (bit 4) of the Interrupt mask (IMR) register. Second, it then sets up the serial port control (SPC) register by placing it in reset and then taking it out. In this example it also configures the serial port to expect frame syncs. Finally, it enables all unmasked interrupts.

```

.text
BEGIN:    LACC    #10h    ; Enable RINT
          SACL    IMR
          LACC    #0h    ; reset serial port
          SACL    SPC
          LACC    #0c8h   ; take serial port out of reset
          SACL    SPC
          EINT                ; enable all unmasked interrupts

```

Figure 2. Enabling interrupts

The final stage on the code development side is the interrupt service routine, shown in Figure 3, which takes the data from the serial port data receive register and processes it.

```

RECEIVE:
          LAC      DRR      ; read data from DRR
          :
          :
RETURN    EINT

```

Figure 3. Interrupt service routine

In order to simulate data appearing at the serial port receive pin, the file containing hex values must be set up with one value per line. For example:

```

0x1
0x0a
:
:

```

This file must then be connected to the simulator as shown in figure 4 in the 'siminit.cmd' file. This file defines the 'C5x memory map for the simulator.

```

:
:
:
ma 0x20,1,0x1,IPORT      ; define addr 0x20 to be an input port
mc 0x20,1,data.in,read    ; connects IPORT to file 'data.in'

```

Figure 4. Section of siminit.cmd

The final stage is to set up the serial port pseudo-registers. In this case, we will use the Receive Interrupt Period register (RIRP), which is a basic counter that counts down to zero from a user-determined value. When it reaches zero, it simulates a receive at the serial port receive pin, transferring a word from the 'data.in' file to the serial port receive shift register (RSR) and starting the ISR. The RIRP is set up as follows:

`?rirp=x`

where x is any decimal value. This can either be set up at the end of the 'siminit.cmd' file or manually in the simulator. The "RIRP" will automatically reset itself to x after triggering the interrupt.

More information on connecting files to the simulator and serial port pseudo-registers can be found in the *TMS320C5x C Source Debugger User's Guide* (SPRU055).