TMS320 DSP
Number 80
DESIGNER'S NOTEBOOK
TEXAS INSTRUMENTS

Monitoring the TMS320C240 Peripheral Registers Using the Debugger Software

Contributed by Jeff Crankshaw

Design Problem How do I view the peripheral registers on a 'C240 or 'F240 device while I am running the emulator?

Solution The 'C24x DSP controllers are designed around the 'C2xLP core processor and utilize the same software development tools as the other devices in the 'C2xx generation. However, the 'C24x devices integrate many peripherals that are unique to the DSP family, such as the Event Manager and the dual 10-bit ADCs. Also, all of the peripheral registers are mapped into data space.

The 'C2xx C source debugger software provides the capability to customize the debugger display. By using the *wa* command, you may display any data memory location and give it a meaningful name. Typing the following on the command line of the debugger will bring up the Watch window with the contents of data memory location 701Ah and label that data as SYSSR.

*wa *0x0701A, SYSSR*

This is the location of the System Status register for 'C24x devices.

Details of the *wa* command and Watch window may be found in Chapter 8 and Chapter 13 of the *TMS320C2xx C Source Debugger User's Guide* (SPRU151).

A command file customized for the 'C240 that includes memory map definitions and *wa* commands for all registers is included below. All register *wa* commands are commented out. To view a specific peripheral register in the watch window, simply remove the semicolon preceding the *wa* command of choice. This file is also available on the TI ftp site and the TMS320 BBS.

```
echo C240init.CMD for 'C240
mr
;DATA MEMORY
ma 0x00000,1,0x005F,ram        ;MMRs
ma 0x00060,1,0x0020,ram        ;On-Chip RAM B2
ma 0x00200,1,0x0100,ram        ;On-Chip RAM B0 if CNF=0
ma 0x00300,1,0x0100,ram        ;On-Chip RAM B1
ma 0x07000,1,0x043F,ram        ;On-Chip Data mapped peripherals
```

Figure 1. Example init file

```
;PROGRAM MEMORY
ma 0x00000,0,0x04000 ,rom  ;Internal Prog mem (Flash or ROM)
                           ;Available if MPNMC=0.
ma 0x0fe00,0,0x100, ram    ;Available if CNF=1 i.e. B0

;System Module Registers
;~~~~~~~~~~~~~~~~~~~~~~
;wa *0x07018, SYSCR      ;System Module Control Register
;wa *0x0701A, SYSSR      ;System Module Status Register
;wa *0x0701E, SYSIVR     ;System Interrupt Vector Register

;Interrupt Registers
;~~~~~~~~~~~~~~~~~~~
;wa *0x07070, XINT1      ;Int1 (type A) Control reg
;wa *0x07072, NMI        ;Non maskable Int (type A) Control reg
;wa *0x07078, XINT2      ;Int2 (type C) Control reg
;wa *0x0707A, XINT3      ;Int3 (type C) Control reg

;Digital I/O
;~~~~~~~~~~~
;wa *0x07090, OCRA       ;Output Control Reg A
;wa *0x07092, OCRB       ;Output Control Reg B
;wa *0x07094, ISRA       ;Input Status Reg A
;wa *0x07096, ISRB       ;Input Status Reg B
;wa *0x07098, PADATDIR   ;I/O port A Data & Direction reg.
;wa *0x0709A, PBDATDIR   ;I/O port B Data & Direction reg.
;wa *0x0709C, PCDATDIR   ;I/O port C Data & Direction reg.
;wa *0x0709E, PDDATDIR   ;I/O port D Data & Direction reg.

;WatchDog(WD)/Real Time Int(RTI)/Phase Locked Loop (PLL)
; Registers
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;wa *0x07021, RTICNTR    ;RTI Counter reg
;wa *0x07023, WDCNTR     ;WD Counter reg
;wa *0x07025, WDKEY      ;WD Key reg
;wa *0x07027, RTICR      ;RTI Control reg
;wa *0x07029, WDCR       ;WD Control reg
;wa *0x0702B, CKCR0      ;PLL control reg 1
;wa *0x0702D, CKCR1      ;PLL control reg 2

;Analog-to-Digital Converter (ADC) registers
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;wa *0x07032, ADCTRL1    ;ADC Control Register 1
;wa *0x07034, ADCTRL2    ;ADC Control Register 2
;wa *0x07036, ADCFIFO1   ;ADC 2-Level-Deep Data Register FIFO
                         ;for ADC1.
;wa *0x07038, ADCFIFO2   ;ADC 2-Level-Deep Data Register FIFO
                         ;for ADC2.
```

Figure 1. Example init file (continued)

```
;Serial Peripheral Interface (SPI) Registers
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;wa *0x07040, SPICCR         ;SPI Config Control Reg
;wa *0x07041, SPICTL         ;SPI Operation Control Reg
;wa *0x07042, SPISTS         ;SPI Status Reg
;wa *0x07044, SPIBRR         ;SPI Baud rate control reg
;wa *0x07046, SPIEMU         ;SPI Emulation buffer reg
;wa *0x07047, SPIBUF         ;SPI Serial Input buffer reg
;wa *0x07049, SPIDAT         ;SPI Serial Data reg
;wa *0x0704D, SPIPC1         ;SPI Port control reg1
;wa *0x0704E, SPIPC2         ;SPI Port control reg2
;wa *0x0704F, SPIPRI         ;SPI Priority control reg

;Serial Communications Interface (SCI) Registers
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;wa *0x07050, SCICCR         ;SCI Comms Control Reg
;wa *0x07051, SCICTL1        ;SCI Control Reg 1
;wa *0x07052, SCIHBAUD       ;SCI Baud rate control
;wa *0x07053, SCILBAUD       ;SCI Baud rate control
;wa *0x07054, SCICTL2        ;SCI Control Reg 2
;wa *0x07055, SCIRXST        ;SCI Receive status reg
;wa *0x07056, SCIRXEMU       ;SCI EMU data buffer
;wa *0x07057, SCIRXBUF       ;SCI Receive data buffer
;wa *0x07059, SCITXBUF       ;SCI Transmit data buffer
;wa *0x0705E, SCIPC2         ;SCI Port control reg2
;wa *0x0705F, SCIPRI         ;SCI Priority control reg

;Event Manager (EV)
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;wa *0x07400, GPTCON    ;General Timer Controls
;wa *0x07401, T1CNT     ;T1 Counter Register
;wa *0x07402, T1CMPR    ;T1 Compare Register
;wa *0x07403, T1PR      ;T1 Period Register
;wa *0x07404, T1CON     ;T1 Control Register
;wa *0x07405, T2CNT     ;T2 Counter Register
;wa *0x07406, T2CMPR    ;T2 Compare Register
;wa *0x07407, T2PR      ;T2 Period Register
;wa *0x07408, T2CON     ;T2 Control Register
;wa *0x07409, T3CNT     ;T3 Counter Register
;wa *0x0740a, T3CMPR    ;T3 Compare Register
;wa *0x0740b, T3PR      ;T3 Period Register
;wa *0x0740c, T3CON     ;T3 Control Register
;wa *0x07411, COMCON    ;Compare Unit Control Register
;wa *0x07413, ACTR      ;Full Compare Output Action Ctl. Reg.
;wa *0x07414, SACTR     ;Simple Compare Output Action Ctl. Reg.
;wa *0x07415, DBTCON    ;Dead Band Timer Control
;wa *0x07417, CMPR1     ;Full Compare Channel 1 Threshold
;wa *0x07418, CMPR2     ;Full Compare Channel 2 Threshold
;wa *0x07419, CMPR3     ;Full Compare Channel 3 Threshold
;wa *0x0741a, SCMPR1    ;Simple Comp Channel 1 Threshold
;wa *0x0741b, SCMPR2    ;Simple Comp Channel 2 Threshold
;wa *0x0741c, SCMPR3    ;Simple Comp Channel 3 Threshold
;wa *0x07420, CAPCON    ;Capture Unit Control
;wa *0x07422, CAPFIFO   ;FIFO1-4 Status Register
;wa *0x07423, CAP1FIFO  ;Capture Channel 1 FIFO Top
;wa *0x07424, CAP2FIFO  ;Capture Channel 2 FIFO Top
;wa *0x07425, CAP3FIFO  ;Capture Channel 3 FIFO Top
;wa *0x07426, CAP4FIFO  ;Capture Channel 4 FIFO Top
```

Figure 1. Example init file (continued)

```
;wa *0x0742c, EVIMRA    ;Group A Interrupt Mask Register
;wa *0x0742d, EVIMRB    ;Group B Interrupt Mask Register
;wa *0x0742e, EVIMRC    ;Group C Interrupt Mask Register
;wa *0x0742f, EVIFRA    ;Group A Interrupt Flag Register
;wa *0x07430, EVIFRB    ;Group B Interrupt Flag Register
;wa *0x07431, EVIFRC    ;Group C Interrupt Flag Register
;wa *0x07432, EVIVRA    ;Group A Int. Vector Offset Register
;wa *0x07433, EVIVRB    ;Group B Int. Vector Offset Register
;wa *0x07434, EVIVRC    ;Group C Int. Vector Offset Register

echo   C240init.CMD HAS BEEN LOADED
```

Figure 1. Example init file (continued)