Number 83E

TMS320 DSP DESIGNER'S NOTEBOOK

Monitoring the TMS320C240 Peripheral Registers using the Debugger Code Composer (GO DSP) version 3.03.

Contributed by Pascal Dorster

Design Problem

How do I view the peripheral registers on a 'C240 device while I am running the emulator Code Composer?

Solution

The same software development tools 'C2xx Code Composer should be used with all elements of the 'C2xx family, 'C240 is one of this element. An Extension Language in GO DSP allows users to customize the emulation windows by adding functions to the menu item. This Designer Note Page give a set of functions to add actions in the menu bar to access 'C240 peripherals memory mapped into data space.

'C240 peripherals registers displayed by this customization are gathered in eight sub-group which are:

- Event Manager
- Digital I/O
- Watchdog
- PLL
- ADC
- SPI
- SCI
- Interrupt
- System

Details on the GO DSP Extension Language (GEL) may be found in Chapter 12 of the Code Composer User's Guide.

A GEL file customized for the 'C240 that includes memory map for all registers is included below. This file is also available on the TI Web site and the TMS320 BBS.

To setup the 'C240 environment with the following GEL file, select 'Load/Unload GEL command from the menu bar and load the C240.gel file.

If you want to setup the 'C240 environment with the following GEL file each time you open up a Control Window you have to add the C240.gel file in the command line of Code Composer or include the following lines in the init2xx.gel file.

Texas Instruments provides customer support in varied technical areas. Since TI does not possess full access to data concerning all of the uses and applications of customers' products, TI assumes no responsibility for customer product design or the use of application of customers' products or for any infringements of patents or rights of others which may result from TI assistance.

Example C240.gel File

```
/*************************************************/
/* File : C240.gel                               */
/* Descrition: GEL File for Code Composer and    */
/*             ’C240                             */
/* Date : 12/96                                  */
/* Author: Pascal Dorster (PADZ)                 */
/*                                               */
/* More : This file allows you to view the       */
/*       peripherals registers on a ’C240 device */
/*       while running the emulator.             */
/*                                               */
/* Instruction: Replace the init2xx.gel file by  */
/*              this file or customize your      */
/*              init2xx.gel with these lines of  */
/*              code.                            */
/*************************************************/

menuitem "C240"


hotmenu Event_Manager()
{
/*Event Manager (EV) ~~~~~~~~~~~~~~~~~~~~~~~~~~*/
GEL_WatchAdd("*(int*)0x07400,x", "GPTCON");  /*General Timer Controls*/
GEL_WatchAdd("*(int*)0x07401,x", "T1CNT ");  /*T1 Counter Register*/
GEL_WatchAdd("*(int*)0x07402,x", "T1CMP ");  /*T1 Compare Register*/
GEL_WatchAdd("*(int*)0x07403,x", "T1PER ");  /*T1 Period Register*/
GEL_WatchAdd("*(int*)0x07404,x", "T1CON ");  /*T1 Control Register*/
GEL_WatchAdd("*(int*)0x07405,x", "T2CNT ");  /*T2 Counter Register*/
GEL_WatchAdd("*(int*)0x07406,x", "T2CMP ");  /*T2 Compare Register*/
GEL_WatchAdd("*(int*)0x07407,x", "T2PER ");  /*T2 Period Register*/
GEL_WatchAdd("*(int*)0x07408,x", "T2CON ");  /*T2 Control Register*/
GEL_WatchAdd("*(int*)0x07409,x", "T3CNT ");  /*T3 Counter Register*/
GEL_WatchAdd("*(int*)0x0740a,x", "T3CMP ");  /*T3 Compare Register*/
GEL_WatchAdd("*(int*)0x0740b,x", "T3PER ");  /*T3 Period Register*/
GEL_WatchAdd("*(int*)0x0740c,x", "T3CON ");  /*T3 Control Register*/
GEL_WatchAdd("*(int*)0x07411,x", "COMCON");  /*Compare Unit Control Register*/
GEL_WatchAdd("*(int*)0x07413,x", "ACTR  ");  /*Full Compare Output Action Ctl. Reg.*/
GEL_WatchAdd("*(int*)0x07414,x", "SACTR ");  /*Simple Compare Output Action Ctl. Reg.*/
GEL_WatchAdd("*(int*)0x07415,x", "DBTCON");  /*Dead Band Timer Control*/
GEL_WatchAdd("*(int*)0x07417,x", "CMPR1 ");  /*Full Compare Channel 1*/
GEL_WatchAdd("*(int*)0x07418,x", "CMPR2 ");  /*Full Compare Channel 2*/
GEL_WatchAdd("*(int*)0x07419,x", "CMPR3 ");  /*Full Compare Channel 3*/
GEL_WatchAdd("*(int*)0x0741a,x", "SCMPR1");  /*Simple Compare Channel 1*/
GEL_WatchAdd("*(int*)0x0741b,x", "SCMPR2");  /*Simple Compare Channel 2*/
GEL_WatchAdd("*(int*)0x0741c,x", "SCMPR3");  /*Simple Compare Channel 3*/
GEL_WatchAdd("*(int*)0x07420,x", "CAPCON");  /*Capture Unit Control*/
GEL_WatchAdd("*(int*)0x07422,x", "CAPFIFO"); /*FIFO1-4 Status Register*/
GEL_WatchAdd("*(int*)0x07423,x", "FIFO1 ");  /*Capture Channel 1 FIFO Top*/
GEL_WatchAdd("*(int*)0x07424,x", "FIFO2 ");  /*Capture Channel 2 FIFO Top*/
GEL_WatchAdd("*(int*)0x07425,x", "FIFO3 ");  /*Capture Channel 3 FIFO Top*/
GEL_WatchAdd("*(int*)0x07426,x", "FIFO4 ");  /*Capture Channel 4 FIFO Top*/
GEL_WatchAdd("*(int*)0x0742c,x", "IMRA  ");  /*Group A Interrupt Mask Register*/
GEL_WatchAdd("*(int*)0x0742d,x", "IMRB  ");  /*Group B Interrupt Mask Register*/
GEL_WatchAdd("*(int*)0x0742e,x", "IMRC  ");  /*Group C Interrupt Mask Register*/
GEL_WatchAdd("*(int*)0x0742f,x", "IFRA  ");  /*Group A Interrupt Flag Register*/
GEL_WatchAdd("*(int*)0x07430,x", "IFRB  ");  /*Group B Interrupt Flag Register*/
GEL_WatchAdd("*(int*)0x07431,x", "IFRC  ");  /*Group C Interrupt Flag Register*/
GEL_WatchAdd("*(int*)0x07432,x", "IVRA  ");  /*Group A Int. Vector Offset Reg*/
GEL_WatchAdd("*(int*)0x07433,x", "IVRB  ");  /*Group B Int. Vector Offset Reg*/
GEL_WatchAdd("*(int*)0x07431,x", "IVRC  ");  /*Group C Int. Vector Offset Reg*/
}


hotmenu Digital_IO()
```

```
{
/*Digital IO (IO) ~~~~~~~~~~~~~~~~~~~~~~~~~~*/
GEL_WatchAdd("*(int*)0x07090,x", "OPCRA");    /*Output control Register A*/
GEL_WatchAdd("*(int*)0x07092,x", "OPCRB");    /*Output control Register B*/
GEL_WatchAdd("*(int*)0x07094,x", "IPSRA");    /*Input Status Register A*/
GEL_WatchAdd("*(int*)0x07096,x", "IPSRB");    /*Input Status Register B*/
GEL_WatchAdd("*(int*)0x07098,x", "IOPA_DDR"); /*I/O Port A data and direction reg*/
GEL_WatchAdd("*(int*)0x0709A,x", "IOPB_DDR"); /*I/O Port B data and direction reg*/
GEL_WatchAdd("*(int*)0x0709C,x", "IOPC_DDR"); /*I/O Port C data and direction reg*/
GEL_WatchAdd("*(int*)0x0709E,x", "IOPD_DDR"); /*I/O Port D data and direction reg*/
}


hotmenu Watchdog()
{
/*Watchdog (WD) ~~~~~~~~~~~~~~~~~~~~~~~~~~*/
GEL_WatchAdd("*(int*)0x07021,x", "RTICNTR");  /*RTI Counter Register*/
GEL_WatchAdd("*(int*)0x07023,x", "WDTCNTR");  /*WD Counter Register*/
GEL_WatchAdd("*(int*)0x07025,x", "WDKEY");    /*WD Key Register*/
GEL_WatchAdd("*(int*)0x07027,x", "RTICR");    /*RTI Control Register*/
GEL_WatchAdd("*(int*)0x07029,x", "WDCR");     /*WD Control Register*/
}


hotmenu PLL()
{
/*PLL (PLL) ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/
GEL_WatchAdd("*(int*)0x0702B,x", "CKCR1");    /*PLL Control Register 1*/
GEL_WatchAdd("*(int*)0x0702D,x", "CKCR2");    /*PLL Control Register 2*/
}


hotmenu ADC()
{
/*Analog-to-Digital Convertor (ADC) ~~~~~*/
GEL_WatchAdd("*(int*)0x07030,x", "AC2CNTL1"); /*ADC Control reg 1*/
GEL_WatchAdd("*(int*)0x07032,x", "AC2CNTL2"); /*ADC Control reg 2*/
GEL_WatchAdd("*(int*)0x07034,x", "AC2DR1");   /*ADC Data reg 1*/
GEL_WatchAdd("*(int*)0x07036,x", "AC2DR2");   /*ADC Data reg 2*/
}


hotmenu SPI()
{
/*Serial Peripheral Interface (SPI) ~~~~~*/
GEL_WatchAdd("*(int*)0x07040,x", "SPICCR");   /*SPI Config Control Register*/
GEL_WatchAdd("*(int*)0x07041,x", "SPICTL");   /*SPI Operation Control Register*/
GEL_WatchAdd("*(int*)0x07042,x", "SPISTS");   /*SPI Status Register*/
GEL_WatchAdd("*(int*)0x07044,x", "SPIBRR");   /*SPI Baud Rate Control Register*/
GEL_WatchAdd("*(int*)0x07046,x", "SPIEMU");   /*SPI Emulation Buffer Register*/
GEL_WatchAdd("*(int*)0x07047,x", "SPIBUF");   /*SPI Serial Input Buffer Register*/
GEL_WatchAdd("*(int*)0x07049,x", "SPIDAT");   /*SPI Serial Data Register*/
GEL_WatchAdd("*(int*)0x0704D,x", "SPIPC1");   /*SPI Port Control Register 1*/
GEL_WatchAdd("*(int*)0x0704E,x", "SPIPC2");   /*SPI Port Control Register 2*/
GEL_WatchAdd("*(int*)0x0704F,x", "SPIPRI");   /*SPI Priority Register*/
}







hotmenu SCI()
```

```
{
/*Serial Communication Interface (SPI) ~~*/
GEL_WatchAdd("*(int*)0x07050,x", "SCICCR");   /*SCI Communication Control Register*/
GEL_WatchAdd("*(int*)0x07051,x", "SCICTL1");  /*SCI Control Register 1*/
GEL_WatchAdd("*(int*)0x07052,x", "SCIHBAUD"); /*SCI Baud rate reg, High byte*/
GEL_WatchAdd("*(int*)0x07053,x", "SCILBAUD"); /*SCI Baud Rate reg, Low byte*/
GEL_WatchAdd("*(int*)0x07054,x", "SCICTL2");  /*SCI Control Register 2*/
GEL_WatchAdd("*(int*)0x07055,x", "SCIRXST");  /*SCI Receive Status Register*/
GEL_WatchAdd("*(int*)0x07056,x", "SCIRXEMU"); /*SCI EMU Data Buffer*/
GEL_WatchAdd("*(int*)0x07057,x", "SCIRXBUF"); /*SCI Receiver Data Buffer*/
GEL_WatchAdd("*(int*)0x07059,x", "SCITXBUF"); /*SCI Transmit Data Buffer*/
GEL_WatchAdd("*(int*)0x0705D,x", "SCIPC1");   /*SCI Port Control Register 1*/
GEL_WatchAdd("*(int*)0x0705E,x", "SCIPC2");   /*SCI Port Control Register 2*/
GEL_WatchAdd("*(int*)0x0705F,x", "SCIPRI");   /*SCI Priority Control Register*/
}


hotmenu Interrupt()
{
/*Interrupt register (INT) ~~~~~~~~~~~~~~*/
GEL_WatchAdd("*(int*)0x07070,x", "XINT1_CR"); /*Int1 (type A) Control Register*/
GEL_WatchAdd("*(int*)0x07072,x", "NMI_CR");   /*Non Maskable Int (Type A) Cont. Reg*/
GEL_WatchAdd("*(int*)0x07078,x", "XINT2_CR"); /*Int2 (type C) Control Register*/
GEL_WatchAdd("*(int*)0x0707A,x", "XINT3_CR"); /*Int3 (type C) Control Register*/
GEL_WatchAdd("*(int*)0x0742C,x","PDPINT_CR"); /*Power Drive Protection Int Cntl reg*/
}


hotmenu System()
{
/*System Module Register (SYS) ~~~~~~~~~~*/
GEL_WatchAdd("*(int*)0x07018,x", "SYSCR");    /*System Module Control Register*/
GEL_WatchAdd("*(int*)0x0701A,x", "SYSSR");    /*System Module Status Register*/
GEL_WatchAdd("*(int*)0x0701E,x", "SYSIVR");   /*System Interrupt Vector Register*/
}
```