

# ***Getting Started With TMS320C28x Digital Signal Controllers***

Christine Peng

C2000/AEC

## **ABSTRACT**

This guide is organized by development flow and functional areas to make your design effort as seamless as possible. Tips on getting started with TMS320C28x™ DSP software and hardware development are provided to aid in your initial design and debug efforts. Each section includes pointers to valuable information including technical documentation, software, and tools for use in each phase of design.

## **Contents**

	Trademarks .....	1
1	Development Quick Start Overview .....	2
2	Getting Started .....	3
3	Running Your First C28x Application on an eZdsp .....	5
4	Technical Publication Descriptions .....	9

## **List of Figures**

1	Spectrum Digital C2000 Development Tools Website .....	5
2	Boot to SARAM Switch 1 Configuration on eZdspF2808 .....	6
3	Code Composer Studio File View Window .....	7
4	280x CPU Timer Example Watch Window .....	7
5	Adding Variables to the Watch Window .....	8
6	Real-time Mode Option in the Debug Menu .....	8

## **Trademarks**

TMS320C28x, Code Composer Studio, C28x are trademarks of Texas Instruments.

eZdsp is a trademark of Spectrum Digital.

All other trademarks are the property of their respective owners.

## 1 Development Quick Start Overview

This section gives a brief overview of the steps to take when first developing for a C28x device. For more detail on each of these steps, see the corresponding section.

### Step 1. Acquire the appropriate development tools —

The quickest way to begin working with a C28x device is to acquire an eZdsp™ kit for initial development, which, in one package, includes:

- On-board JTAG emulation via USB or parallel port
- Appropriate emulation driver
- Code Composer Studio™ IDE for eZdsp

Once you have become familiar with the device and begin developing on your own hardware, purchase Code Composer Studio™ IDE separately for software development and a JTAG emulation tool to get started on your project.

### Step 2. Download starter software —

To simplify programming for C28xx devices, it is recommended that users download and use the C/C++ Header Files and Example(s) (see [Section 2.4](#) for links to the files) to begin developing software for the C28x devices and their various peripherals.

After downloading the appropriate header file package for your device, refer to the following resources for step-by-step instructions on how to run the peripheral examples and use the header file structure for your own software

- The Quick Start Readme in the /doc directory to run your first application.
- *Programming TMS320x28xx and 28xxx Peripherals in C/C++ Application Report* (literature number [SPRAA85](#))

### Step 3. Download flash programming software —

Many C28x devices include on-chip flash memory and tools that allow you to program the flash with your software IP.

- Flash Tools: [C28x Flash Tools](#)
- *TMS320F281x Flash Programming Solutions* (literature number [SPRB169](#))
- *Running an Application from Internal Flash Memory on the TMS320F28xx DSP* (literature number [SPRA958](#))

### Step 4. Move on to more advanced topics —

For more application software and other advanced topics, visit the TI website at <http://www.ti.com> or <http://www.ti.com/c2000getstarted>.

## 2 Getting Started

This section expands on the topics mentioned in [Section 1](#) and lists online resources to help you in the early stages of application development.

### 2.1 Registration on my.TI

my.TI is a customizable area within the Texas Instruments website. By registering for a [my.TI account](#), you can receive the following benefits:

- Quick reference to information you select as part of your profile
- Email alerts that inform you of updates to products, technical documentation, and errata
- The my.TI newsletter providing information on the latest innovations and product releases

To register on my.TI for updates related to the this device:

Step 1. Go to the device product folder.

Step 2. Select the link called "ADD To my.TI" in the upper right hand corner, and follow the on-screen instructions.

Step 3. Select Customize my.TI to specify what notification you would like to receive.

### 2.2 Where to Start

For a comprehensive list of online resources to help you begin developing for C28x devices, visit the C2000 Getting Started web page:

[Getting Started with TMS320C2000 Digital Signal Controllers](#)

The key area for obtaining documentation for a device is the product folder. A list of links to all the product folders can be accessed from the Getting Started Web Page. Click on the TMS320C28x link under Device Information. When getting started, it is of great importance to have the latest data sheet and silicon errata. Listed below are links to this key information:

#### 280x

- [List of 280x devices](#) with links to each device folder
- TMS320F2809, TMS320F2808, TMS320F2806, TMS320F2802, TMS320F2801, TMS320C2802, TMS320C2801, and TMS320F2801x DSPs Data Manual (literature number [SPRS230](#))
- TMS320F280x, TMS320C280x, and TMS320F2801x DSP Silicon Errata (literature number [SPRZ171](#))

#### 2804x

- [TMS320F28044 DSP product folder](#)
- TMS320F28044 Digital Signal Processor Data Manual (literature number [SPRS357](#))
- TMS320F28044 DSP Silicon Errata (literature number [SPRZ255](#))

#### 281x

- [List of 281x devices](#) with links to each product folder
- TMS320F2810, TMS320F2811, TMS320F2812, TMS320C2810, TMS320C2811, TMS320C2812 DSPs Data Manual (literature number [SPRS174](#))
- TMS320F2810\F2811\F2812, TMS320C2810\C2811\C2812 DSP Silicon Errata (literature number [SPRZ193](#))
- TMS320R2811, TMS320R2812 Digital Signal Processors Data Manual (literature number [SPRS257](#))
- TMS320R2811, TMS320R2812 Digital Signal Processors Silicon Errata (literature number [SPRZ226](#))

### 2.3 Development Tools

Acquire an eZdsp kit for initial development, or when developing on your own hardware, purchase Code Composer Studio for software development, and a JTAG emulation tool to get started on your project.

*eZdsp Development Board Kits:* The eZdsp kits provide all of the hardware and software tools required during initial development (including on-board emulation via USB or parallel port and Code Composer Studio IDE).

- [TMDSEZS2808](#) - F2809, F2808, F2806, F2802, F2801, F28016, F28015
- [TMDSEZS28044](#) - F28044
- [TMDSEZD2812](#) - F2812, F2811, and F2810

To get started with the eZdsp kit, visit <http://c2000.spectrumdigital.com> and select the appropriate eZdsp for schematics and quick start guides.

*Debugger and Emulation Tools:* If developing on your own hardware, you will need a robust software development studio and an emulator that supports real-time JTAG emulation.

- Code Composer Studio IDE for C2000
- A standalone JTAG emulator
  - From TI: Visit the [TI e-Store](#)
  - From one of TI's third parties: For emulators from companies such as Signum Systems, Blackhawk, or Spectrum Digital, go to the [Third Party Products and Services website](#) and type the company name in the "Search by Name" textbox under "Company Search"

*Development Tool Drivers:* For Code Composer Studio IDE to interface with the eZdsp and/or other emulation tools, software drivers may be required. See your vendor's website for drivers specific to your emulator. Code Composer Studio includes emulation drivers only for TI JTAG emulators.

## 2.4 Starter Software

It is recommended users download and use these header files and peripheral examples to begin easily developing software for the C28x devices. The header files implement a hardware abstraction layer method to allow easy C/C++ code access to memory-mapped peripheral registers.

TMS320C28x Header Files and Example projects

- [C281x C/C++ Header Files and Peripheral Examples \(SPRC097\)](#)
- [C280x C/C++ Header Files and Peripheral Examples \(SPRC191\)](#)
- [C2804x C/C++ Header Files and Peripheral Examples \(SPRC324\)](#)

After downloading the appropriate header file package, see:

- The Quickstart Readme in the /doc directory to run your first application
- [Programming TMS320x28xx and 28xxx Peripherals in C/C++ \(SPRAA85\)](#)

These resources include step-by-step instructions on header file directory structure, coding, and usage in both the examples and in your own software.

## 2.5 Flash Programming Software

Many C2000 devices include on-chip flash memory. TI supplies tools that allow you to program the flash with your software IP.

These tools include:

- CCStudio plugin - a JTAG-based Flash plugin for Code Composer Studio
- SD Flash utility - a standalone JTAG-based GUI interface maintained by Spectrum Digital
- Serial Port Programmer - a serial port (RS232-SCI) based flash programming example that works as a special patch to SD Flash

For more information on C28x Flash tools and to download Flash algorithms, visit:

<http://www.ti.com/c2000getstarted> and click on C28x flash tools under Programming Software

- Application Notes:
  - [TMS320F281x Flash Programming Solutions \(SPRB169\)](#)
  - [Running an Application from Internal Flash Memory on the TMS320F28xx DSP \(SPRA958\)](#)

## 2.6 Training and Support

Texas Instruments offers a variety of training options tailored for your specific needs and requirements. Options include on-line training, webcasts, seminars, single and multi-day workshops, and conferences. For more information about training, visit Texas Instruments [Training Home](#). For assistance with technical questions regarding Texas Instruments Semiconductor products and services, you can access the [Semiconductor Technical Support KnowledgeBase](#).

## 3 Running Your First C28x Application on an eZdsp

This section gives step-by-step instructions on how to run your first C28x application on an eZdsp. It is assumed that you have already acquired an eZdsp kit specific to a C28x device. Although these instructions include references to the F2808, they apply to any device running an application on an eZdsp. Minor adjustments may be required for your specific device.

### 3.1 Setting up the eZdsp and Code Composer Studio

Follow these steps:

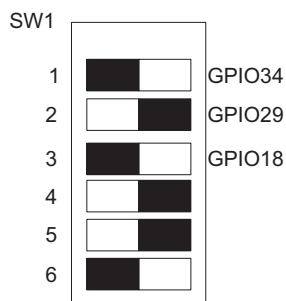
- Step 1. On the C2000 Getting Started website (<http://www.ti.com/c2000getstarted>), download the appropriate header file package for your device under Starter Software (i.e. C280x, C2801x C/C++ Header Files and Peripheral Examples (literature number [SPRC191](#)) for F280x)
- Step 2. Extract the header files to a directory of your choosing. It is recommended to extract them to the default: C:\tidcs\c28\DSP28yx\vxxyz\ directory. For more information on the header file directory structure and files, see the device Quick Start Readme in the /doc folder.
- Step 3. Visit: <http://c2000.spectrumdigital.com/> and click on the appropriate eZdsp device link under the Site Index. (i.e. eZdspF2808 link for F280x devices)



Figure 1. Spectrum Digital C2000 Development Tools Website

- Step 4. See the Quick Start guide corresponding to your device (eZdspF2808 Quick Start Guide for the F280x devices) for step-by-step instructions from Spectrum Digital on how to set up Code Composer Studio, configure the appropriate drivers, and power up the eZdsp board. The Quick Start Guide can also be found on the eZdsp CD-ROM included in the eZdsp kit.
- Step 5. Download the eZdsp board schematic (eZdspF2808 Schematic for F280x devices) for future reference. The board schematic can also be found on the eZdsp CD-ROM included in the eZdsp kit.
- Step 6. Code Composer Studio should now be running on the powered eZdsp board (This step assumes that Code Composer Studio is properly configured according to the instructions in [Step 4](#) above). If not, then click on the F2808 EzDSP Code Composer icon to start Code Composer again.
- Step 7. Configure the Boot Mode pins on the eZdsp board (SW1 positions 1-3 on eZdspF2808) for "Boot

to SARAM" mode (factory default for eZdspF2808). [Figure 2](#) demonstrates the Boot Mode pin configuration for "Boot to SARAM" mode on the eZdspF2808.



**Figure 2. Boot to SARAM Switch 1 Configuration on eZdspF2808**

- Step 8. In Code Composer Studio, go to the Debug menu-> Connect in order to connect the debugger to the device. A balloon with the statement, "The Target is now connected", should appear in the lower left corner of the window.
- Step 9. Continue with the steps in either [Section 3.2](#) or [Section 3.3](#).

### 3.2 Setting up the Application in Code Composer Using a GEL file

The steps below assume you are using the GEL file provided within each of the example folders (i.e., Example\_C280xCpuTimer.gel) to load code, build projects, and set up the Watch Window. A GEL file is a file written in the GEL scripting language specific to Code Composer Studio, which allows you to set up Code Composer interfaces and simplify operations by selecting options in a pull-down menu. When developing your own project, it may not be desirable to use the example GEL file.

If you choose not use the example GEL file, skip the steps in this section and move to [Section 3.3](#) and continue with those steps.

- Step 1. Go to File menu -> Load GEL.
- Step 2. Navigate to the folder where the header files were extracted in Step 2 of [Section 3.1](#).
- Step 3. Open the DSP280x\_examples folder (or the examples folder corresponding to your device). By default, this should be C:\tidcs\c28\DSP280x\vxxyz\DSP280x\_examples\
- Step 4. Click on the cpu\_timer folder.
- Step 5. Inside the folder, open the Example\_280xCpuTimer.gel file.
- Step 6. Go to the GEL menu -> DSP280x CpuTimerExample -> Load and Build Project.

This will compile and build the CPU Timer example code and load it into the DSP SARAM. The left-hand side of the Code Composer window should now look something like [Figure 3](#). To see the source code and header files, expand the plus sign next to Example\_280xCpuTimer.pjt.

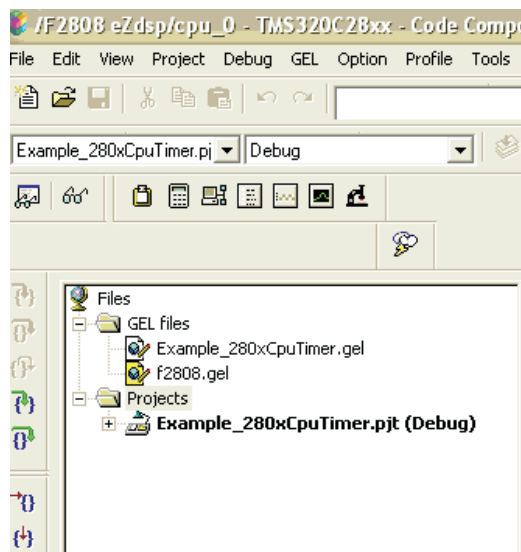


Figure 3. Code Composer Studio File View Window

Step 7. To monitor variables of interest in the project, Go to the GEL menu-> DSP280x CpuTimerExample-> Load Setup\_WatchWindow. The Watch Window will appear in the lower right-hand corner of the Code Composer Studio window with some CPU Timer variable names as shown in Figure 4. This can also be accomplished by highlighting any variable name in the source code, right-clicking on it, and selecting "Add to Watch Window".

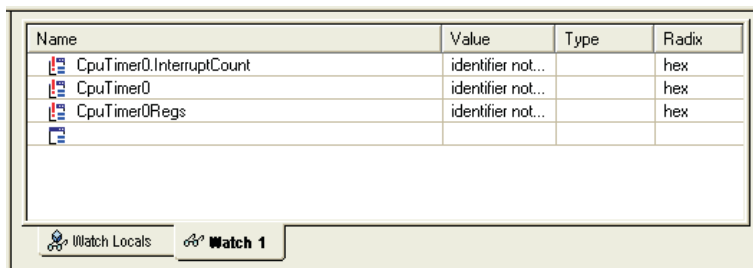


Figure 4. 280x CPU Timer Example Watch Window

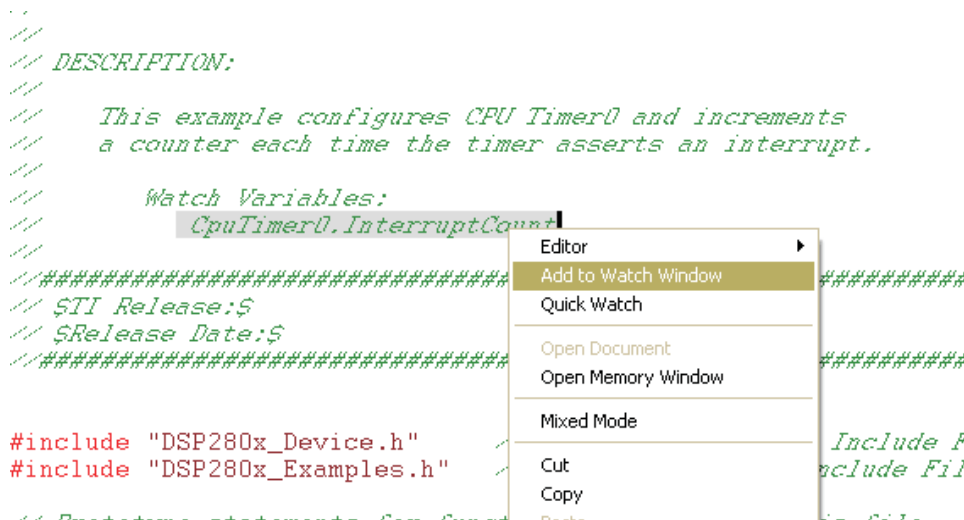
Step 8. Continue with the steps in Section 3.4

### 3.3 Setting up the Application in Code Composer Without Using a GEL file

The steps below replace the steps in Section 3.2, and set up the application without using a GEL file.

1. Go to Project menu -> Open... and navigate to the folder where the header files were extracted in Step 2 of Section 3.1.
2. Open the DSP280x\_examples folder (or the examples folder corresponding to your device). By default, this should be C:\tidcs\c28\DSP280x\vxxyz\DSP280x\_examples\.
3. Click on the cpu\_timer folder, and inside the folder, open the Example\_280xCpuTimer.pjt file.
4. Go to Project menu-> Build.
5. Then when the build completes, go to File menu->Load program
6. Browse to the cpu\_timer folder, click on the Debug folder, and select the Example\_280xCpuTimer.out file. Files with the extension .out are generated by the linker and contain the data that is loaded into the C28x device SARAM.
7. Now set up the Watch window by highlighting the variable or register to be observed (CpuTimer0.InterruptCount), right-clicking, and selecting "Add to Watch Window" as shown in Figure 5. Another option is to type the register name or variable directly into the Watch Window.





**Figure 5. Adding Variables to the Watch Window**

8. Continue with the steps in [Section 3.4](#) to run the application.

### 3.4 Running the Application

- Step 1. To observe the Watch variables changing in real-time, click on Debug menu-> Real-time mode (See [Figure 6](#)). A dialog box will pop up asking if you really want to allow real-time mode switching. Click OK. Then right-click in the watch window, and select Continuous Refresh.



**Figure 6. Real-time Mode Option in the Debug Menu**

- Step 2. Reset the CPU by going to Debug menu -> Reset CPU
- Step 3. Run the program by clicking on the blue running man icon on the Debug Toolbar, or by going to Debug->Run. The CPU Timer example sets up a CPU Timer to interrupt once every 1 second. Observe in the Watch Window as CpuTimerx.InterruptCount (where x = 0, 1, or 2) increments once per second.



- Step 4. Expand CpuTimerx and/or CpuTimerxRegs to see the CPU Timer registers and related variables change as the program runs.
- Step 5. When you want to halt the program, click on the blue man icon with the red X on the Debug Toolbar or by going to Debug->Halt.
- Step 6. If you want to change the frequency of the interrupts, expand the source folder in the File View and click on Example\_280xCpuTimer.c. Scroll down to the ConfigCpuTimer() function. replace 1000000 with your desired period in microseconds. Then go back to [Section 3.2](#) or [Section 3.3](#) to rebuild and run your application.

Congratulations! You have finished running your first C28x application on an eZdsp. To further understand the general flow of C28x application software setup, it is recommended to read through the example and/or header file comments. The device initialization flow is the same in all the examples.

## 4 Technical Publication Descriptions

This section describes the content contained in technical publications that support this device. All of the technical publications described below can be found in the device product folder. Check your device product folder frequently for the most recent technical documentation.

### 4.1 Using Texas Instruments Incorporated Literature Numbers

All Texas Instruments Incorporated documentation is assigned a literature number. This number can be used to search for the document on the Web. Technical documentation revisions are indicated by the alpha character at the end of the literature number on the title page, and in the file name.

Use the literature number (without the trailing alpha character) to search the Texas Instruments Incorporated website for the document. For example, if a data manual has a literature number of SPRS205B, the "B" indicates the revision of the document. If the document has no trailing alpha character, it is the original version of the document. When searching for this document on the Texas Instruments Incorporated web site, you simply enter "SPRS205" as the search keyword.

#### Data Sheets and Data Manuals

The Data Sheet or Data Manual is the functional specification for the device. Topics covered in this document include but are not limited to the following:

- High-level functional overview
- Pinouts and packaging information
- Signal descriptions
- Device-specific information about peripherals and registers
- Electrical specifications

#### Silicon Errata

The Silicon Errata documents exceptions to the functional specification as defined in the Data Sheet or Data Manual.

#### Reference Guides

Reference Guides provide additional information describing the architecture and operation of hardware components of the DSP platform, generation, or device, beyond the scope of the Data Sheet or Data Manual.

#### Application Reports

Application Reports are written to describe implementation details specific to a device, peripheral, use of technology, or explanation of usage.

## 4.2 280x DSP Data Manuals

### Data Manuals —

**SPRS230** — [TMS320F2809, F2808, F2806, F2802, F2801, F2801x UCD9501, C2802, C2801 DSPs Data Manual](#) contains the pinout, signal descriptions, as well as electrical and timing specifications for the F280x devices.

**SPRS357** — [TMS320F28044 Digital Signal Processor Data Manual](#) contains the pinout, signal descriptions, as well as electrical and timing specifications for the F28044 device.

## 4.3 C280x DSP User's Guides

### CPU User's Guides —

**SPRU430** — [TMS320C28x DSP CPU and Instruction Set Reference Guide](#) describes the central processing unit (CPU) and the assembly language instructions of the TMS320C28x fixed-point digital signal processors (DSPs). It also describes emulation features available on these DSPs.

**SPRU712** — [TMS320x28xx, 28xxx System Control and Interrupts Reference Guide](#) describes the various interrupts and system control features of the 280x digital signal processors (DSPs).

### Peripheral Guides —

**SPRU566** — [TMS320x28xx, 28xxx Peripheral Reference Guide](#) describes the peripheral reference guides of the 28x digital signal processors (DSPs).

**SPRU716** — [TMS320x280x, 2801x, 2804x Analog-to-Digital Converter \(ADC\) Reference Guide](#) describes how to configure and use the on-chip ADC module, which is a 12-bit pipelined ADC.

**SPRU791** — [TMS320x28xx, 28xxx Enhanced Pulse Width Modulator \(ePWM\) Module Reference Guide](#) describes the main areas of the enhanced pulse width modulator that include digital motor control, switch mode power supply control, UPS (uninterruptible power supplies), and other forms of power conversion

**SPRU924** — [TMS320x28xx, 28xxx High-Resolution Pulse Width Modulator \(HRPWM\)](#) describes the operation of the high-resolution extension to the pulse width modulator (HRPWM)

**SPRU807** — [TMS320x28xx, 28xxx Enhanced Capture \(eCAP\) Module Reference Guide](#) describes the enhanced capture module. It includes the module description and registers.

**SPRU790** — [TMS320x28xx, 28xxx Enhanced Quadrature Encoder Pulse \(eQEP\) Reference Guide](#) describes the eQEP module, which is used for interfacing with a linear or rotary incremental encoder to get position, direction, and speed information from a rotating machine in high performance motion and position control systems. It includes the module description and registers

**SPRU074** — [TMS320x28xx, 28xxx Enhanced Controller Area Network \(eCAN\) Reference Guide](#) describes the eCAN that uses established protocol to communicate serially with other controllers in electrically noisy environments.

**SPRU051** — [TMS320x28xx, 28xxx Serial Communication Interface \(SCI\) Reference Guide](#) describes the SCI, which is a two-wire asynchronous serial port, commonly known as a UART. The SCI modules support digital communications between the CPU and other asynchronous peripherals that use the standard non-return-to-zero (NRZ) format.

**SPRU059** — [TMS320x28xx, 28xxx Serial Peripheral Interface \(SPI\) Reference Guide](#) describes the SPI - a high-speed synchronous serial input/output (I/O) port - that allows a serial bit stream of programmed length (one to sixteen bits) to be shifted into and out of the device at a programmed bit-transfer rate.

**SPRU721** — [TMS320x28xx, 28xxx Inter-Integrated Circuit \(I2C\) Reference Guide](#) describes the features and operation of the inter-integrated circuit (I<sup>2</sup>C) module that is available on the TMS320x280x digital signal processor (DSP).

**SPRU722** — [TMS320x280x, 2801x, 2804x Boot ROM Reference Guide](#) describes the purpose and features of the bootloader (factory-programmed boot-loading software). It also describes other contents of the device on-chip boot ROM and identifies where all of the information is located within that memory.

#### Tools Guides —

**SPRU513** — [TMS320C28x Assembly Language Tools User's Guide](#) describes the assembly language tools (assembler and other tools used to develop assembly language code), assembler directives, macros, common object file format, and symbolic debugging directives for the TMS320C28x device.

**SPRU514** — [TMS320C28x Optimizing C Compiler User's Guide](#) describes the TMS320C28x™ C/C++ compiler. This compiler accepts ANSI standard C/C++ source code and produces TMS320 DSP assembly language source code for the TMS320C28x device.

**SPRU608** — [The TMS320C28x Instruction Set Simulator Technical Overview](#) describes the simulator, available within the Code Composer Studio for TMS320C2000 IDE, that simulates the instruction set of the C28x™ core.

**SPRU625** — [TMS320C28x DSP/BIOS Application Programming Interface \(API\) Reference Guide](#) describes development using DSP/BIOS.

## 4.4 281x DSP Data Manuals

#### Data Manuals —

**SPRS174** — [TMS320F2810, TMS320F2811, TMS320F2812, TMS320C2810, TMS320C2811, TMS320C2812 Digital Signal Processors Data Manual](#) contains the electrical and timing specifications for these devices, as well as signal descriptions and pinouts for all of the available packages.

**SPRS257** — [TMS320R2811, TMS320R2812 Digital Signal Processors Data Manual](#) contains the electrical and timing specifications for these devices, as well as signal descriptions and pinouts for all of the available packages.

## 4.5 281x User's Guides

#### CPU User's Guides —

**SPRU430** — [TMS320C28x DSP CPU and Instruction Set Reference Guide](#) describes the central processing unit (CPU) and the assembly language instructions of the TMS320C28x fixed-point digital signal processors (DSPs). It also describes emulation features available on these DSPs.

**SPRU078** — [TMS320x281x System Control and Interrupts Reference Guide](#) describes the various interrupts and system control features of the 281x digital signal processors (DSPs).

#### Peripheral Guides —

**SPRU566** — [TMS320x28xx, 28xxx Peripheral Reference Guide](#) describes the peripheral reference guides of the 28x digital signal processors (DSPs).

**SPRU060** — [TMS320x281x Analog-to-Digital Converter \(ADC\) Reference Guide](#) describes the ADC module, which is a 12-bit pipelined ADC. The analog circuits of this converter, referred to as the core in this document, include the front-end analog multiplexers (MUXs), sample-and-hold (S/H) circuits, the conversion core, voltage regulators, and other analog supporting circuits. Digital circuits, referred to as the wrapper in this document, include programmable conversion sequencer, result registers, interface to analog circuits, interface to device peripheral bus, and interface to other on-chip modules.

**SPRU065** — [TMS320x281x Event Manager \(EV\) Reference Guide](#) describes the EV modules that provide a broad range of functions and features that are particularly useful in motion control and motor control applications. The EV modules include general-purpose (GP) timers, full-compare/pulse width modulator (PWM) units, capture units, and quadrature-encoder pulse (QEP) circuits.

- SPRU074** — [TMS320x28xx, 28xxx Enhanced Controller Area Network \(eCAN\) Reference Guide](#) describes the eCAN that uses established protocol to communicate serially with other controllers in electrically noisy environments.
- SPRU051** — [TMS320x28xx, 28xxx Serial Communication Interface \(SCI\) Reference Guide](#) describes the SCI, which is a two-wire asynchronous serial port, commonly known as a UART. The SCI modules support digital communications between the CPU and other asynchronous peripherals that use the standard non-return-to-zero (NRZ) format.
- SPRU059** — [TMS320x28xx, 28xxx Serial Peripheral Interface \(SPI\) Reference Guide](#) describes the SPI - a high-speed synchronous serial input/output (I/O) port - that allows a serial bit stream of programmed length (one to sixteen bits) to be shifted into and out of the device at a programmed bit-transfer rate.
- SPRU061** — [TMS320x281x Multi-channel Buffered Serial Ports \(McBSPs\) Reference Guide](#) describes the McBSP available on the C28x devices. The McBSPs allow direct interface between a DSP and other devices in a system.
- SPRU067** — [TMS320x281x External Interface \(XINTF\) Reference Guide](#) describes the external interface (XINTF) of the 281x digital signal processors (DSPs).
- SPRU095** — [TMS320x281x Boot ROM Reference Guide](#) describes the purpose and features of the bootloader (factory-programmed boot-loading software). It also describes other contents of the device on-chip boot ROM and identifies where all of the information is located within that memory.
- Tools Guides —**
- SPRU513** — [TMS320C28x Assembly Language Tools User's Guide](#) describes the assembly language tools (assembler and other tools used to develop assembly language code), assembler directives, macros, common object file format, and symbolic debugging directives for the TMS320C28x device.
- SPRU514** — [TMS320C28x Optimizing C Compiler User's Guide](#) describes the TMS320C28x™ C/C++ compiler. This compiler accepts ANSI standard C/C++ source code and produces TMS320 DSP assembly language source code for the TMS320C28x device.
- SPRU608** — [The TMS320C28x Instruction Set Simulator Technical Overview](#) describes the simulator, available within the Code Composer Studio for TMS320C2000 IDE, that simulates the instruction set of the C28x™ core.
- SPRU625** — [TMS320C28x DSP/BIOS Application Programming Interface \(API\) Reference Guide](#) describes development using DSP/BIOS.

## 4.6 Application Reports

Application reports provide help for specific applications.

### Application Reports —

- SPRAAM0** — [Getting Started With TMS320C28x™ Digital Signal Controllers](#) is organized by development flow and functional areas to make your design effort as seamless as possible. Tips on getting started with C28x™ DSP software and hardware development are provided to aid in your initial design and debug efforts. Each section includes pointers to valuable information including technical documentation, software, and tools for use in each phase of design.
- SPRAAD5** — [Power Line Communication for Lighting Apps using BPSK w/ a Single DSP Controller](#) presents a complete implementation of a power line modem following CEA-709 protocol using a single DSP.
- SPRAA85** — [Programming TMS320x28xx and 28xxx Peripherals in C/C++](#) explores a hardware abstraction layer implementation to make C/C++ coding easier on 28x DSPs. This method is compared to traditional #define macros and topics of code efficiency and special case registers are also addressed.

- SPRA958** — [Running an Application from Internal Flash Memory on the TMS320F28xx DSP](#) covers the requirements needed to properly configure application software for execution from on-chip flash memory. Requirements for both DSP/BIOS™ and non-DSP/BIOS projects are presented. Example code projects are included.
- SPRAA91** — [TMS320F280x DSC USB Connectivity Using TUSB3410 USB-to-UART Bridge Chip](#) presents hardware connections as well as software preparation and operation of the development system using a simple communication echo program.
- SPRAA58** — [TMS320x281x to TMS320x280x Migration Overview](#) describes differences between the Texas Instruments TMS320x281x and TMS320x280x DSPs to assist in application migration from the 281x to the 280x. While the main focus of this document is migration from 281x to 280x, users considering migrating in the reverse direction (280x to 281x) will also find this document useful.
- SPRAAD8** — [TMS320280x and TMS320F2801x ADC Calibration](#) describes a method for improving the absolute accuracy of the 12-bit ADC found on the TMS320280x and TMS3202801x devices. Inherent gain and offset errors affect the absolute accuracy of the ADC. The methods described in this report can improve the absolute accuracy of the ADC to levels better than 0.5%. This application report has an option to download an example program that executes from RAM on the F2808 EzDSP.
- SPRAAI1** — [Using Enhanced Pulse Width Modulator \(ePWM\) Module for 0-100% Duty Cycle Control](#) provides a guide for the use of the ePWM module to provide 0% to 100% duty cycle control and is applicable to the TMS320x280x family of processors.
- SPRAA88** — [Using PWM Output as a Digital-to-Analog Converter on a TMS320F280x](#) presents a method for utilizing the on-chip pulse width modulated (PWM) signal generators on the TMS320F280x family of digital signal controllers as a digital-to-analog converter (DAC).
- SPRAAH1** — [Using the Enhanced Quadrature Encoder Pulse \(eQEP\) Module](#) provides a guide for the use of the eQEP module as a dedicated capture unit and is applicable to the TMS320x280x, 28xxx family of processors.
- SPRA820** — [Online Stack Overflow Detection on the TMS320C28x DSP](#) presents the methodology for online stack overflow detection on the TMS320C28x™ DSP. C-source code is provided that contains functions for implementing the overflow detection on both DSP/BIOS™ and non-DSP/BIOS applications.
- SPRA806** — [An Easy Way of Creating a C-callable Assembly Function for the TMS320C28x DSP](#) provides instructions and suggestions to configure the C compiler to assist with understanding of parameter-passing conventions and environments expected by the C compiler.

## 4.7 BSDL Models

The Boundary Scan Description Language (BSDL) file for a given device describes how the boundary scan components are implemented on the device according to the JTAG standard IEEE 1149.1. This file can be used to assist in the testing of the connectivity of the device to other board level components.

### Software —

- SPRM194** — F2801 100-Pin GGM/ZGM BSDL Model
- SPRM195** — F2801 100-Pin PZ BSDL Model
- SPRM196** — F2806 100-Pin PZ BSDL Model
- SPRM197** — F2808 100-Pin PZ BSDL Model
- SPRM198** — F2808 100-Pin GGM/ZGM BSDL Model
- SPRM200** — F2806 100-Pin GGM/ZGM BSDL Model
- SPRM244** — F2809 GGM/ZGM BSDL Model

- SPRM245** — F2809 PZ BSDL Model
- SPRM246** — F28044 GGM/ZGM BSDL Model
- SPRM247** — F28044 PZ BSDL Model
- SPRM177** — F2810 PBK BSDL Model (Silicon Revision C and higher)
- SPRM178** — F2811 PBK BSDL Model (Silicon Revision C and higher)
- SPRM179** — F2812 PGF BSDL Model (Silicon Revision C and higher)
- SPRM180** — F2812 GHH BSDL Model (Silicon Revision C and higher)
- SPRM187** — C2810 PBK BSDL Model (Silicon Revision C and higher)
- SPRM188** — C2811 PBK BSDL Model (Silicon Revision C and higher)
- SPRM189** — C2812 GHH BSDL Model (Silicon Revision C and higher)
- SPRM190** — C2812 PGF BSDL Model (Silicon Revision C and higher)



## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

### Products

Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
Low Power Wireless	<a href="http://www.ti.com/lpw">www.ti.com/lpw</a>

### Applications

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Automotive	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Broadband	<a href="http://www.ti.com/broadband">www.ti.com/broadband</a>
Digital Control	<a href="http://www.ti.com/digitalcontrol">www.ti.com/digitalcontrol</a>
Military	<a href="http://www.ti.com/military">www.ti.com/military</a>
Optical Networking	<a href="http://www.ti.com/opticalnetwork">www.ti.com/opticalnetwork</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Telephony	<a href="http://www.ti.com/telephony">www.ti.com/telephony</a>
Video & Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>
Wireless	<a href="http://www.ti.com/wireless">www.ti.com/wireless</a>

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2007, Texas Instruments Incorporated