

Cyclone II FPGA Starter Development Kit

User Guide



101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
<http://www.altera.com>

Document Version
Document Date

1.0.0
October 2006

P25-36048-00

Copyright © 2006 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.



Printed on recycled paper

UG-CDK01012-10





About This User Guide	v
------------------------------------	----------

Chapter 1. Getting Started

Introduction	1-1
Before You Begin	1-1
Further Information	1-2
Hardware Installation	1-3
Software Installation	1-3
Installing the Cyclone II FPGA Starter Development Kit CD-ROM	1-4
Installing the Quartus II Software	1-5

Chapter 2. Development Board Setup

Development Board Overview	2-1
Requirements	2-1
Powering Up the Development Board	2-2
Confirming Board Operation	2-2

Chapter 3. Control Panel Setup

Requirements	3-1
Hardware Setup	3-1
FPGA Configuration	3-1
Control Panel Start	3-2

Chapter 4. Using the Control Panel

Control Panel Overview	4-1
Controlling the 7-Segment Displays	4-2
Lighting the LEDs	4-3
PS/2 Keyboard	4-3
SDRAM/SRAM Controller and Programmer	4-4
Read/Write Data	4-4
Sequential Write	4-5
Sequential Read	4-6
Flash Memory Programmer	4-6
Read/Write Data	4-7
Sequential Write	4-8
Sequential Read	4-8
Configuring User Ports	4-9
Flash Music Player	4-9
VGA Display	4-11
Displaying the Default Image	4-11

Displaying Another Image from a Downloaded Bitmap File	4-12
Displaying Any Image Files	4-15

Chapter 5. Using the Development Board

Configuring the Cyclone II FPGA	5-1
JTAG Programming	5-1
AS Programming	5-1
Configuration Procedure	5-2
Configuring the FPGA in JTAG Mode	5-2
Configuring the EPCS4 Device in AS Mode	5-3

Chapter 6. Advanced Examples

Factory Configuration	6-1
File Locations	6-1
Demonstration Setup	6-1
Music Synthesizer Demonstration	6-2
File Locations	6-4
Demonstration Setup	6-4
SD Card Music Player	6-5
File Locations	6-6
Demonstration Setup	6-6



About This User Guide

This user guide describes how to start using the Altera® Cyclone® II FPGA Starter Development Kit, including unpacking the kit, installing required software, connecting the development board to a PC, and running sample software.

For a full description of the development board and its use, refer to the *Cyclone II FPGA Starter Development Kit Reference Manual*.



The document revision history in [Table 1–1](#) shows this document's current version. To ensure that you have the most up-to-date information on this product, refer to the **readme** file on the provided CD_ROM for late-breaking information that is not available in this document.

Table 1–1. Document Revision History

Date	Description
October 2006	Initial publication of the Cyclone II FPGA Starter Development Kit, version 1.0.0

How to Find Information

The following methods enable you to quickly find information in this Portable Document Format (PDF) type document:

- Search the contents by using the Adobe® Acrobat® or Reader® **Edit/Find** command or click on the binoculars/**Search** toolbar icon.
- The **Bookmarks** window serves as an additional table of contents. Click on a topic to jump to that section in the document.
- Thumbnail icons in the **Pages** window provide miniature previews of each page and provide a link to the pages.
- Within the text, hypertext links, highlighted in green, enable you to jump to related information.

How to Contact Altera

To get help regarding this product, use the following contact information:

- Altera Corporation
101 Innovation Drive
San Jose, California, 95134 USA
www.altera.com

For the most up-to-date information about Altera products, go to the Altera world-wide web site at www.altera.com. For technical support on this product, go to www.altera.com/mysupport. For additional information about Altera products, consult the sources shown below.

Information Type	USA & Canada	All Other Locations
Technical support	www.altera.com/mysupport/	www.altera.com/mysupport/
	(800) 800-EPLD (3753) (7:00 a.m. to 5:00 p.m. Pacific Time)	+1 408-544-8767 7:00 a.m. to 5:00 p.m. (GMT -8:00) Pacific Time
Product literature	www.altera.com	www.altera.com
Altera literature services	literature@altera.com	literature@altera.com
Non-technical customer service	(800) 767-3753	+ 1 408-544-7000 7:00 a.m. to 5:00 p.m. (GMT -8:00) Pacific Time
FTP site	ftp.altera.com	ftp.altera.com

Typographic Conventions

This document uses the typographic conventions shown below.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: Save As dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, filenames, filename extensions, and software utility names are shown in bold type. Examples: f_{MAX} , lqdesigns directory, d: drive, chiptrip.gdf file.
<i>Italic Type with Initial Capital Letters</i>	Document titles are shown in italic type with initial capital letters. Example: <i>AN 75: High-Speed Board Design</i> .
<i>Italic type</i>	Internal timing parameters and variables are shown in italic type. Examples: <i>t_{PIA}</i> , <i>n + 1</i> . Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: <file name>, <project name>.pdf file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
"Subheading Title"	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: "Typographic Conventions."

Visual Cue	Meaning
Courier type	Signal and port names are shown in lowercase Courier type. Examples: <code>data1</code> , <code>tdi</code> , <code>input</code> . Active-low signals are denoted by suffix <code>n</code> , e.g., <code>resetn</code> . Anything that must be typed exactly as it appears is shown in Courier type. For example: <code>c:\qdesigns\tutorial\chiptrip.gdf</code> . Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword <code>SUBDESIGN</code>), as well as logic function names (e.g., <code>TRI</code>) are shown in Courier.
1., 2., 3., and a., b., c., etc.	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ● ●	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.
	The hand points to information that requires special attention.
	The caution indicates required information that needs special consideration and understanding and should be read prior to starting or continuing with the procedure or process.
	The warning indicates information that should be read prior to starting or continuing the procedure or processes
↵	The angled arrow indicates you should press the Enter key.
	The feet direct you to more information on a particular topic.

Introduction

Welcome to the Altera Cyclone II FPGA Starter Development Kit, which includes a full-featured field-programmable gate array (FPGA) development board, hardware and software development tools, documentation, and accessories needed to begin FPGA development.

The development board includes an Altera Cyclone II 2C20 FPGA and comes preconfigured with a hardware reference design stored in flash memory. Hardware designers can use the development board as a platform to prototype complex embedded systems.

The Development Kit provides the user with an integrated control environment that includes a software controller in C++, a USB command controller, a multi-port SRAM/SDRAM/flash memory controller, and demonstration circuitry specified in Verilog code. These features enable users to implement and test designs without the need to implement complex application programming interfaces (APIs), host control software, or SRAM/SDRAM/flash memory controllers.

This user guide addresses the following topics:

- How to set up, power up, and verify correct operation of the development board
- How to install the Altera Development Suite Tools and the Cyclone II FPGA Starter Development Kit CD-ROMs
- How to set up, and use the Control Panel, a graphical user interface (GUI), to manipulate components on the board, implement applications, and display images on a VGA monitor
- How to configure the Cyclone II FPGA
- How to set up and run application examples



For complete details on the development board, refer to the *Cyclone II FPGA Starter Development Kit Reference Manual*.

Before You Begin

Before proceeding, check the contents of the Development Kit:

- Cyclone II FPGA Starter Development board

- Cyclone II FPGA Starter Development Kit CD-ROM containing the development board documentation and supporting materials, including the User Guide and Reference Manual, Control Panel utility, reference designs and demonstrations, device datasheets, tutorials, and a set of laboratory exercises
- The Altera Development Suite Tools CD-ROMs containing Altera Quartus® II 6.0 Web Edition design software and the Nios® II 6.0 embedded processor
- Bag of six rubber (silicon) covers for the development board stands and extender pins that facilitate easier probing of the board I/O expansion headers with testing equipment
- 7.5 V DC wall-mount power supply
- Clear plastic cover for the board

Other items you will want to have available to work through this user guide are:

- VGA monitor
- Audio source such as a CD player or MP3 player
- Headphones
- PS/2 keyboard

Further Information

For other related information, refer to the following websites:

- For additional daughter cards available for purchase:
http://www.altera.com/products/devkits/kit-daughter_boards.jsp
- For on-line demonstrations & training:
<http://www.altera.com/education/demonstrations/dem-index.html> and <https://mysupport.altera.com/etraining/>
- For Cyclone II handbook:
<http://www.altera.com/literature/lit-cyc2.jsp>
- For Cyclone II reference designs:
<http://www.altera.com/end-markets/refdesigns/device/cyclone2/cyclone2-index.jsp>
- For eStore if wish to purchase devices:
<http://www.altera.com/buy/devices/buy-devices.html>

- For Cyclone II Orcad symbols:

http://www.altera.com/support/software/download/pcb/pcb-pcb_index.html

- For Nios II 32-bit embedded processor solutions:

<http://www.altera.com/technology/embedded/emb-index.html>

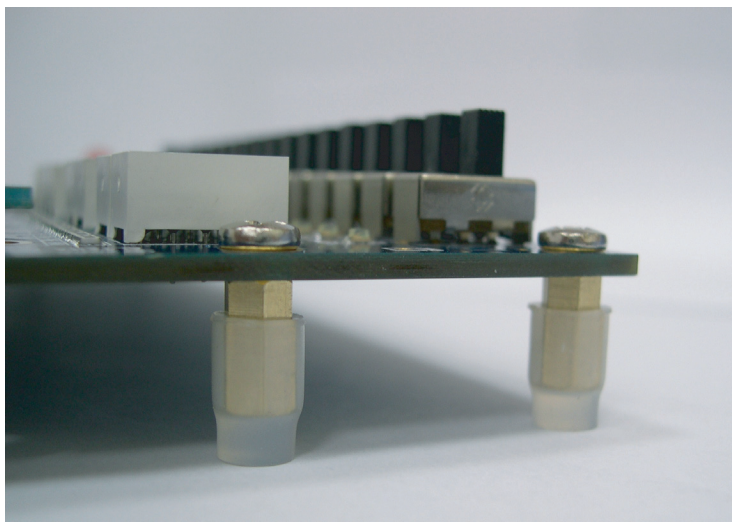
Hardware Installation



Before beginning any work, prepare the board as follows:

- Assemble a rubber (silicon) cover (Figure 1–1) for each of the six copper stands on the board
- Mount the clear plastic cover over the top of the board for extra protection by using additional stands and screws

Figure 1–1. Board Stand Covers



Software Installation

The instructions in this section describe how to install the following:

- *Cyclone II FPGA Starter Development* CD-ROM
- The Quartus II Software, Development Kit Edition

Installing the Cyclone II FPGA Starter Development Kit CD-ROM

The *Cyclone II FPGA Starter Development Kit* CD-ROM contains the following items:

- Sample design files and board design files for the kit
- *Cyclone II FPGA Starter Development Kit User Guide* (this document)
- *Cyclone II FPGA Starter Development Board Reference Manual*

To install the *Cyclone II FPGA Starter Development Kit* CD-ROM, perform the following steps:

1. Insert the *Cyclone II FPGA Starter Development Kit* CD-ROM into the CD-ROM drive.



The CD-ROM should start an auto-install process. If it does not, browse to the CD-ROM drive and double-click on the **setup.exe** file.

2. Follow the online instructions to complete the installation process. The installation program copies the Cyclone II FPGA Starter Development Kit files to the hard-disk and creates a

Programs > Altera > Cyclone II FPGA Starter Development Kit v1.0.0

icon, accessible from the Windows Start menu. Use this icon to launch the Windows-style development kit GUI.

The Cyclone II FPGA Starter Development Kit installation program creates a directory structure for the installed files ([Figure 1-2](#)), where *<path>* is the selected Cyclone II Starter Development Kit installation directory.

Figure 1-2. Cyclone II Starter Kit Installed Directory Structure

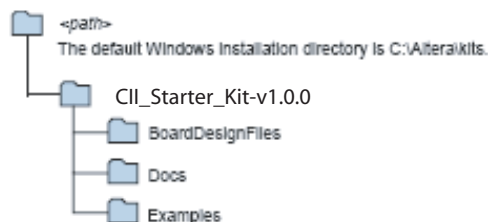


Table 1–1 lists the file directory names and a description of their contents.

Table 1–1. Installed Directory Contents	
Directory Name	Description of Contents
BoardDesignFiles	Contains the board design files. Use these files as a starting point for a new prototype board design.
Docs	Contains the development kit documentation
Examples	Contains the sample design files for the Cyclone II FPGA Starter Development Kit

Installing the Quartus II Software

Install the *Quartus II Web Edition Software Suite* found in the *Altera Design Software Suite* by performing the following steps:

1. Insert the *Quartus II Web Edition Software Suite* CD-ROM into the CD-ROM drive.



The CD-ROM should start an auto-install process. If it does not, browse to the CD-ROM drive and double-click on the **setup.exe** file.

2. Follow the online instructions to complete the installation process.



If you have difficulty installing the Quartus II software, refer to *Installing the Quartus II Software* in the *Quartus II Installation & Licensing Manual for PCs* found at www.altera.com.

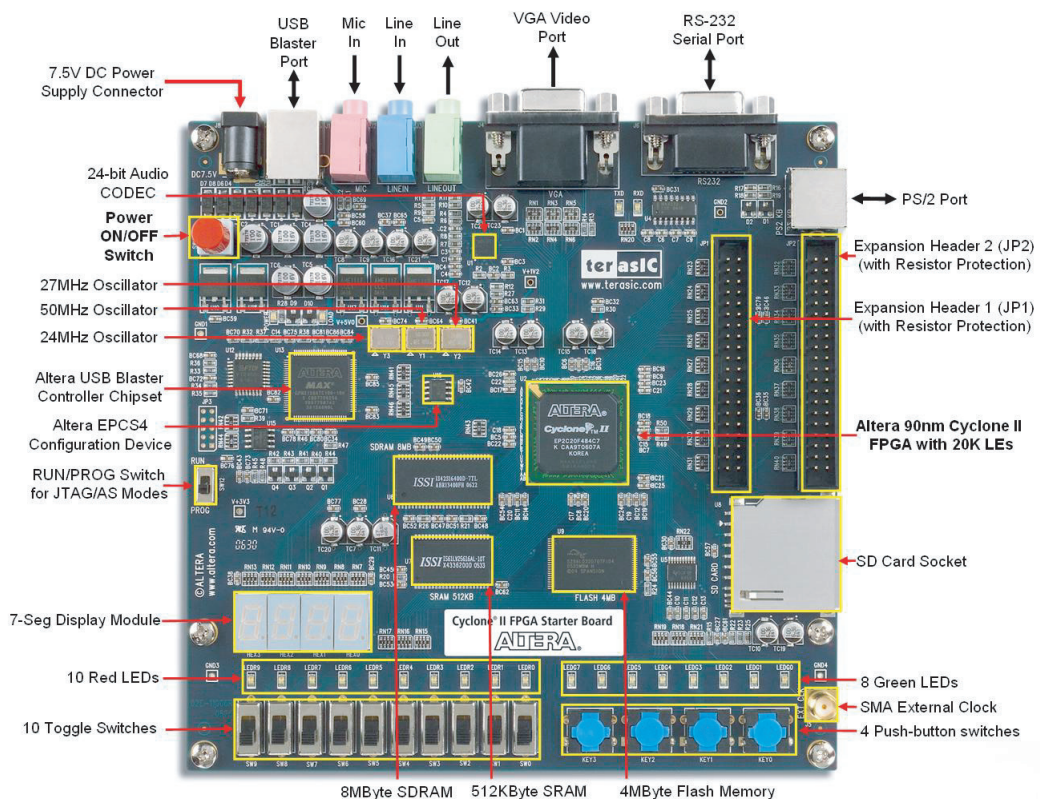
The Quartus II software is the primary FPGA development tool used to create the reference designs used in this development kit.

Additionally, you may want to install the *Nios II Embedded Design Suite* package also found in the *Altera Design Software Suite*. The Nios II soft-core embedded processor runs on Altera FPGAs. Many of the reference designs included in this development kit use the Nios processor.

Development Board Overview

Altera delivers the development board (Figure 2-1) with a preloaded configuration to demonstrate features of the board. At power up, the preloaded configuration also enables users to confirm quickly that the board is operating correctly.

Figure 2-1. Cyclone II Development Board Layout and Components



Requirements

Preparation for using the development board requires the following prerequisite actions:

- Install the Altera Quartus® software on the host computer, if not already installed
- The Cyclone II FPGA Starter Development Board includes an integrated USB-Blaster™ chip set for programming the FPGA. Install the Altera USB-Blaster driver software on the host computer, if not already installed.



Communication between the computer host and the development board requires Altera USB-Blaster driver software already installed on the host computer.

Powering Up the Development Board

To power up the development board, perform the following steps:

1. Connect the USB-Blaster cable from the host computer to the USB-Blaster port on the development board.
2. Connect the 7.5 V DC adapter to the development board and to a power source.
3. Connect a VGA monitor to the VGA video port on the development board.
4. Connect a headset to the **Line Out** audio port on the development board.
5. Slide the **RUN/PROG** switch on the edge of the development board to the RUN position.
6. Turn the power on by depressing the **ON/OFF** switch on the development board.

Confirming Board Operation

As the development board powers on, verify correct operation by performing the following steps:

1. Confirm that all user LEDs are flashing.
2. Confirm that all 7-segment displays cycle through numbers 0 through F.
3. Confirm that the VGA monitor displays the default image (Figure 2–2) with the **SW0** switch set to the DOWN (toward the edge of the board) position.

Figure 2–2. Default VGA Output Pattern, SW0 = DOWN



4. Confirm that the VGA monitor displays the default image (Figure 2–3) with the SW0 switch set to the UP (away from the edge of the board) position.

Figure 2–3. Default FGA Output Pattern, SW0 = UP



5. Set the toggle switch SW9 to the DOWN position and confirm that the computer produces a 1-kHz tone.

6. Set the toggle switch **SW9** to the UP position and connect the output of an audio player (for example: MP3, PC, iPod) to the **Line In** connector on the development board. Confirm that the headset produces the expected music or recorded sounds playing on the audio player.



If you also connect a microphone to the **Mic In** connector, the development board mixes the voice input with the output from the audio player.

The development kit includes a Control Panel facility that enables a user to access various components on the development board from a host computer through a USB connection. For an overview of the Control Panel and its use, refer to [“Using the Control Panel”](#) chapter.

Setting up the Control Panel involves the following actions:

- Set up the hardware
- Configure the FPGA
- Start the Control Panel

Requirements

Preparation for setting up and using the Control Panel requires the following prerequisite actions:

- Install the Altera USB-Blaster driver software on the host computer, if not already installed
- Install the Altera Quartus® software on the host computer, if not already installed

Hardware Setup

Ensure that the development board has powered up and is operating normally, as described in [“Development Board Setup”](#) on page 2–1. If the board is not powered up, but has already been set up and its operation verified, perform the following steps:

1. Check that the USB-Blaster cable connects the host computer to the development board.
2. Check that the 7.5 V DC adapter connects the development board to a power source.
3. Set the **RUN/PROG** switch to the RUN position.
4. Turn the power on by depressing the **ON/OFF** switch on the development board.

FPGA Configuration

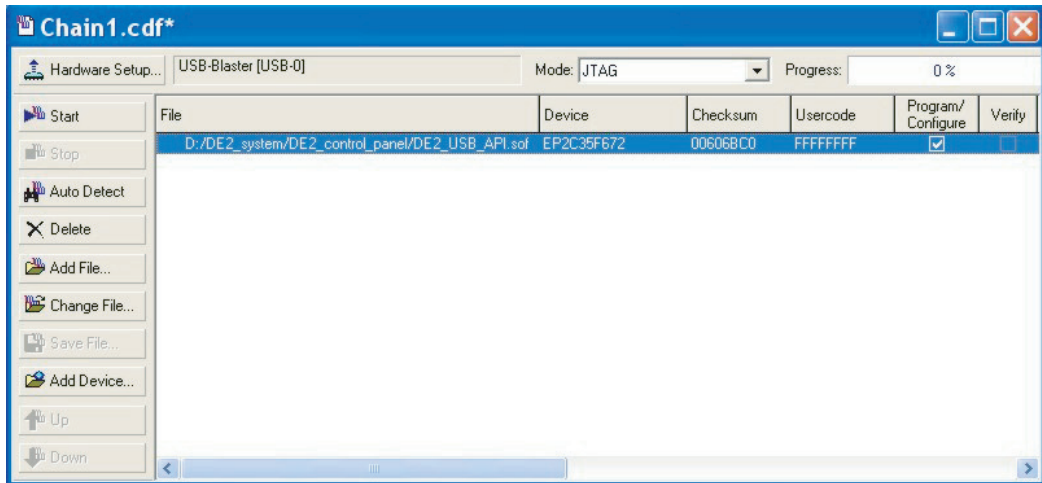
Before using the Control Panel, configure a corresponding circuit in the Cyclone II FPGA by downloading the **CII_Starter_USB_API.sof** configuration file from the

<kit path>\Examples\CII_Starter_demonstrations\CII_Starter_USB_API_v1\HW directory. Refer to Section 4.1 (Configuring the Cyclone II FPGA) for a detailed description of the downloading procedure.

Perform the following steps to configure the FPGA:

1. Start the Quartus II software.
2. Select **Tools > Programmer** to reach the window in [Figure 3-1](#).

Figure 3-1. Quartus II Programmer Window

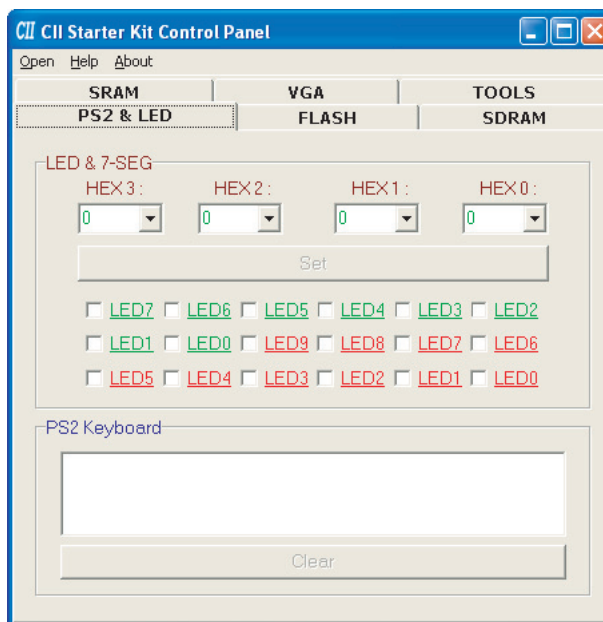


3. Click on **Add File** and select the **CII_Starter_USB_API.sof** file in the pop-up window.
4. Next, click on the **Program/Configure** box to select the added file.
5. Click **Start** to download the selected configuration file into the FPGA.

Control Panel Start

To start the Control Panel, perform the following steps:

1. Run the **CII_Starter_control_panel.exe** program found in the <kit path>\Examples\CII_Starter_demonstrations\CII_Starter_USB_API_v1\SW directory to launch the Control Panel user interface ([Figure 3-2](#)).

Figure 3–2. Control Panel Window

2. Select **Open** to list all USB ports connected to development boards. The Control Panel can control up to 4 development boards using the USB links.
3. Select **Open USB Port 0**. This step places the Control Panel in control of the development board.



The Control Panel occupies the USB port until that port closes. Quartus II can not download a configuration file into the FPGA while the Control Panel occupies the USB port. Closing the Control Panel GUI closes the port.

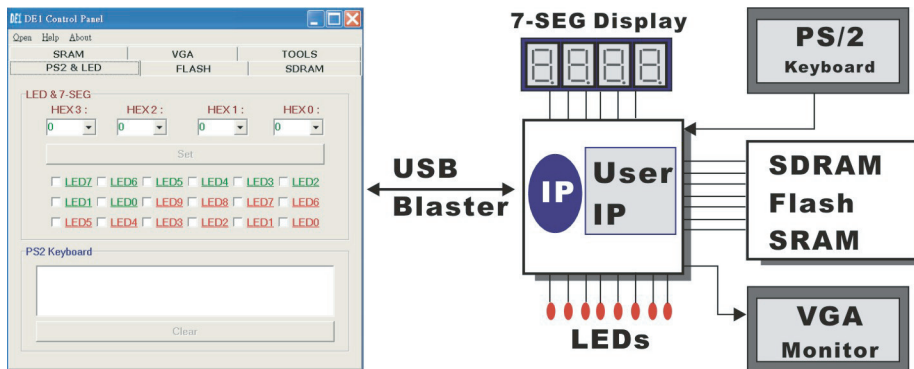
4. Experiment by setting the value of some 7-segment display and observing the result on the development board.

Control Panel Overview

The Control Panel consists of two parts: the GUI on the host and circuitry specified in Verilog code downloaded to the FPGA on the development board. After the kit CD-ROM has been installed, the control panel hardware and software can be found in the <kit path>\Examples\CII_Starter_demonstrations\CII_Starter_USB_API_v1 directory. The available Verilog code enables a knowledgeable user to change the functionality of the Control Panel.

Using the Control Panel GUI on the host computer, a user can issue commands to control circuitry on the development board through the USB-Blaster cable connection as illustrated in Figure 4-1. The design downloaded to the Cyclone II FPGA device on the development board implements a command controller that processes the commands. Then, to perform the appropriate actions, the command controller communicates with the controller of the affected input/output (I/O) device on the development board.

Figure 4-1. Control Panel Access to the Development Board



The user can perform the following actions with the Control Panel:

- Change the values that appear on the 7-segment displays
- Light up LEDs
- Communicate with the PS/2 keyboard
- Read/write from/to the SDRAM, SRAM, and flash memory

- Configure user ports
- Load music to memory and play music via the audio digital analog converter (DAC) output
- Load an image pattern for VGA output

The following sections describe how to perform these actions with the Control Panel already open on the host computer. If not already open, launch the Control Panel as described in [“Control Panel Start” on page 3–2](#).

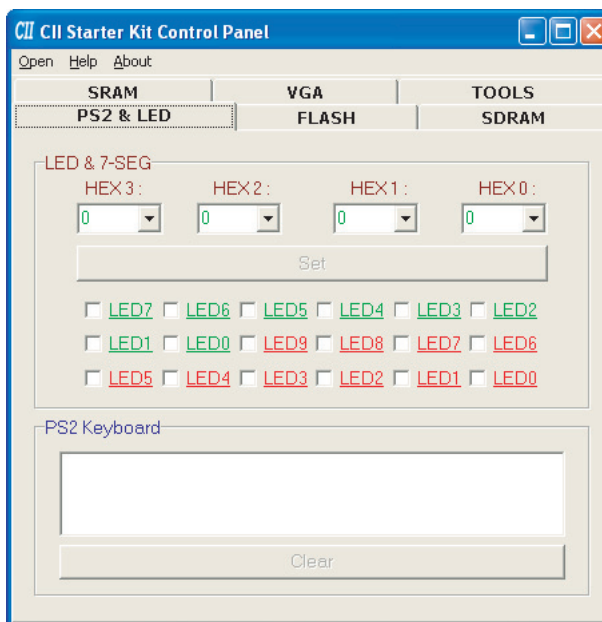
Controlling the 7-Segment Displays

Typical design activities do not require the ability to set arbitrary values into simple display devices. However, used for troubleshooting, this ability enables the user to verify that these devices operate correctly.

To set the value of a 7-segment module to display, perform the following steps:

1. Select the **PS2 & LED** tab on the Control Panel ([Figure 4–2](#)).

Figure 4–2. Control Panel Window for 7-Segment Controls



2. In the **LED & 7-SEG** area, enter a value for any of the 7-segment modules labeled HEX0 through HEX3.

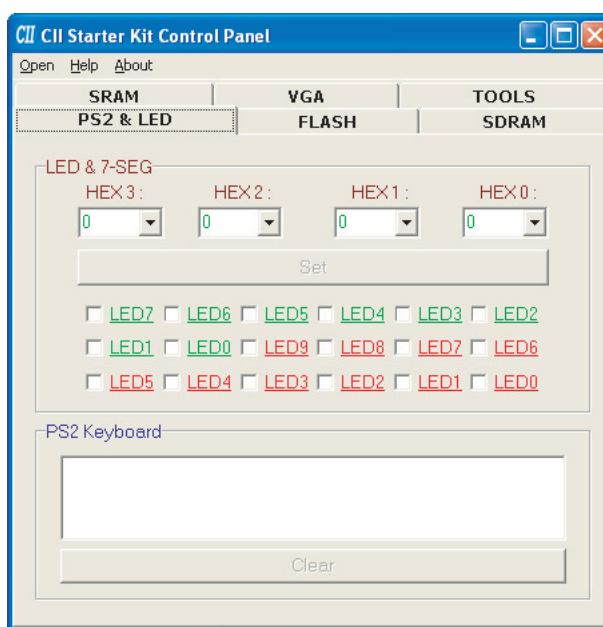
- Click on the **Set** button.

Lighting the LEDs

To light an LED, perform the following steps:

- Select the **PS2 & LED** tab on the Control Panel (Figure 4–3).
- In the **LED & 7-SEG** area, select the individual LEDs to turn on.
- Click on the **Set** button.

Figure 4–3. Control Panel Window for LED Controls



PS/2 Keyboard

The PS/2 keyboard in the Control Panel window (Figure 4–3) shows a working connection between the FPGA and the PS/2 port as well as the software processing the PS/2 interface commands and data.

To test the functionality of the PS/2 keyboard interface, perform the following steps:

- Plug a PS/2 keyboard into the PS/2 port on the development board.

2. Click in the blank area under "PS/2 Keyboard" in the CII Starter Kit Control Panel.
3. Start typing on the PS/2 keyboard.

The keystrokes should echo from the keyboard to the output on the control panel.

SDRAM/SRAM Controller and Programmer

The user can perform the following types of memory read/write operations with the Control Panel:

- Read/write data from/to the SDRAM or SRAM on the development board
- Write sequential data or the entire contents of a file, to the SDRAM or SRAM
- Read sequential data or the entire contents of the SDRAM or SRAM to a file

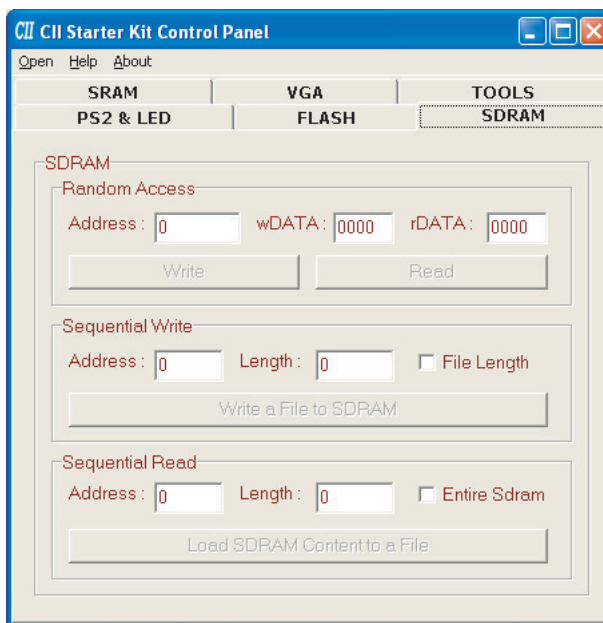
This following sections describe how to access the SDRAM; the same approach also applies to accessing the SRAM.

Read/Write Data

To write/read data from/to the SDRAM, perform the following steps:

1. Select the **SDRAM** tab on the Control Panel (Figure 4-4).

Figure 4–4. Control Panel SDRAM Tab Window



2. To write a 16-bit word into the SDRAM, use the **Random Access** boxes to enter the address of the desired location, specify the data to write, and click on the **Write** button.
3. To read the contents of a location, enter the address and click on the **Read** button.

Sequential Write

To write the contents of a file into the SDRAM, use the Sequential Write function of the Control Panel to perform the following steps:

1. Select the **SDRAM** tab on the Control Panel and use the **Sequential Write** boxes.
2. Specify the starting address in the **Address** box.
3. Specify the number of bytes to write in the **Length** box. To load the entire file, only checkmark the **File Length** box without specifying the number of bytes.

4. Click on the **Write a File to SDRAM** button to initiate the writing of the data.
5. Specify the source file in the pop-up Windows dialog box.



The Control Panel also supports loading files with a *.hex* extension. Files with a *.hex* extension are ASCII text files that specify memory values using ASCII characters to represent hexadecimal values. For example, a file containing the line

```
0123456789ABCDEF
```

defines four 16-bit values: 0123, 4567, 89AB, CDEF. These values load into the memory consecutively.

Sequential Read

To read the contents of the SDRAM and write them to a file, use the Sequential Read function of the Control Panel to perform the following steps:

1. Select the **SDRAM** tab on the Control Panel and use the **Sequential Read** boxes.
2. Specify the starting address in the **Address** box.
3. Specify the number of bytes to copy into the file in the **Length** box. To copy the entire contents of the SDRAM, 8 MBytes of data, into a file, only checkmark the **Entire SDRAM** box without specifying the number of bytes.
4. Click on the **Load SDRAM Content to a File** button.
5. Specify the destination file in the pop-up Windows dialog box.

Flash Memory Programmer

Using the Control Panel, the user can perform the following read/write operations from/to the flash memory on the development board:

- Erase the entire flash memory
- Write one byte to the memory
- Read one byte from the memory
- Write a binary file to the memory
- Load the contents of the flash memory into a file



When performing these operations, consider the following characteristics and limitations of the flash memory:

- 4 Mbits x 8 bits organization
- Erasure of the entire flash memory is required before writing into it
- Flash memory tolerates only a limited number of erasures



Do not exit from the Control Panel while erasing the entire contents of the flash memory; this takes about 40 seconds.

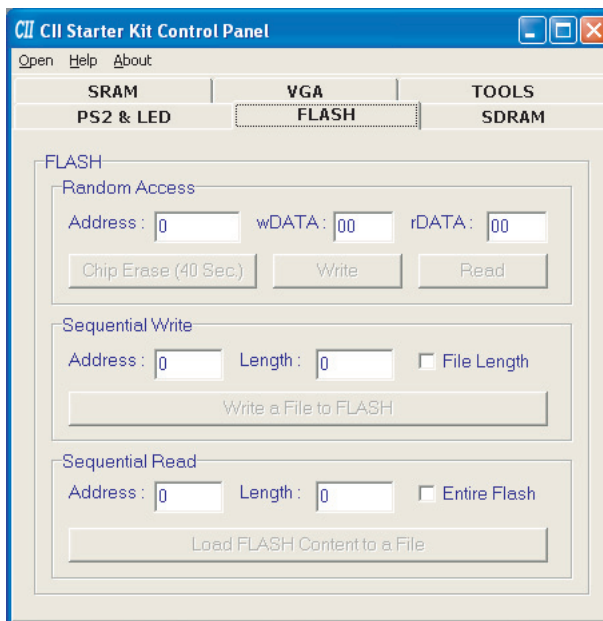
The feature of reading/writing a byte or an entire file from/to the flash memory enables the user to develop multimedia applications (for example, Flash Audio Player, Flash Picture Viewer) without worrying about how to build a Flash Memory Programmer.

Read/Write Data

To perform a read/write operation with a byte of data from/to the flash memory, take the following steps:

1. Select the **FLASH** tab in the Control Panel (Figure 4–5) and use the **Random Access** boxes.

Figure 4–5. Control Panel Flash Tab Window



2. Click on the **Chip Erase (40 Sec)** button. Observe the button and window frame title prompt and wait for the operation to finish; this takes about 40 seconds.
3. Enter the desired address into the **Address** box and the data byte into the **wDATA** box.
4. Click on the **Write** button.
5. To read a byte of data from a random location, enter the address of the location and click on the **Read** button. The **rDATA** box displays the content of the specified address.

Sequential Write

To write the contents of a file into flash memory, perform the following steps:

1. Select the **FLASH** tab on the Control Panel.
2. In the **Random Access** area, click on the **Chip Erase (40 Sec)** button. Observe the button and window frame title prompt and wait for the operation to finish; this takes about 40 seconds.
3. In the **Sequential Write** area, specify the starting address in the **Address** box.
4. Specify the number of bytes to write in the **Length** box. To load the entire file, only checkmark the **File Length** box without specifying the number of bytes.
5. Click on the **Write a File to FLASH** button to initiate the writing of the data.
6. Specify the source file in the pop-up Windows dialog box.

Sequential Read

To read the contents of the flash memory and write them to a file, use the Sequential Read function of the Control Panel to perform the following steps:

1. Select the **FLASH** tab on the Control Panel and use the **Sequential Read** boxes.
2. Specify the starting address in the **Address** box.

3. Specify the number of bytes of data to read from the flash memory in the **Length** box. To copy the entire contents of the flash memory into a file, only checkmark the **Entire Flash** box without specifying the number of bytes.
4. Click on the **Load FLASH Content to a File** button.
5. Specify the destination file in the pop-up Windows dialog box.

Configuring User Ports

The SDRAM, SRAM, and flash memory controllers each have four ports: a host port to connect to the command controller and three user-selectable asynchronous ports. These three user ports can connect the memory to other devices. For example, by connecting the flash controller to the VGA DAC controller, the user can send an image stored in flash memory to the VGA output.

Users can connect circuits of their own design to one of the user ports of the SRAM/SDRAM/flash controllers. After downloading binary data into the SRAM/SDRAM/flash memories, they can configure the memory controllers to enable their circuits to read/write the SDRAM/flash memories through the connected user ports.

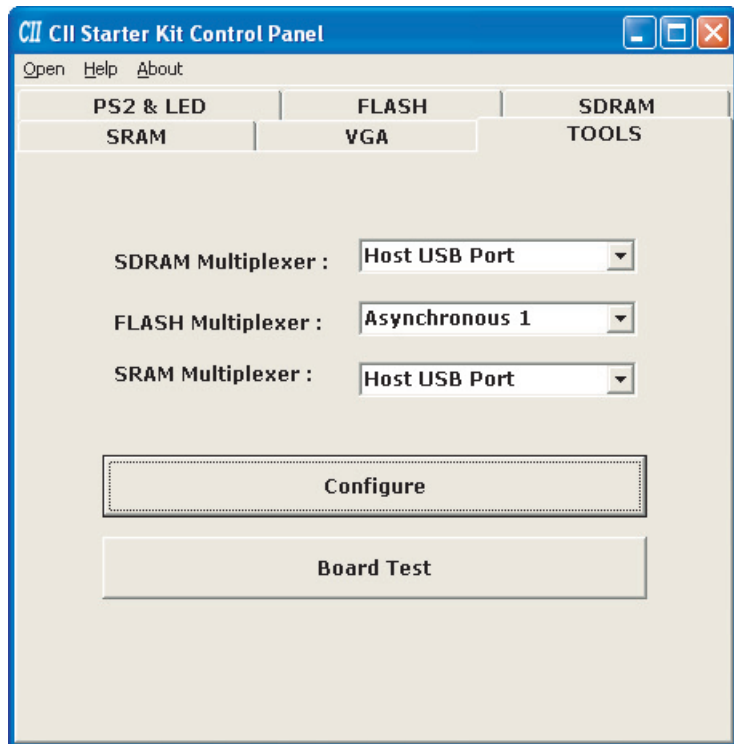
The sample applications of a flash music player and a VGA display illustrate the user port configuration.

Flash Music Player

In this sample application, music data loads into the flash memory. Options under the **Tools** tab of the Control Panel configure user ports. Through user port **Asynchronous 1** of the Flash Controller, the flash memory sends the music data to the Audio DAC Controller and out to the audio output jack.

To implement this application perform the following steps:

1. Erase the flash memory and write a music file into it (refer to [“Sequential Write” on page 4–8](#)). Use the file **music.wav** in the directory **CII_Starter_demonstrations\music** on the **CII Starter System** CD-ROM.
2. In the Control Panel, select the **TOOLS** tab ([Figure 4–6](#)).

Figure 4–6. Control Panel TOOLS Tab Window

3. Select the **Asynchronous 1** port for the **FLASH Multiplexer** and click on the **Configure** button to enable the connection from the flash memory to the **Asynchronous 1** port of the Flash Controller.
4. On the development board, set switches **SW1** to OFF (DOWN position) and **SW0** to ON (UP position).
5. Plug a headset or a speaker into the audio output jack and listen to music produced through the Audio DAC circuit.

This procedure connects the **Asynchronous 1** Port to the Audio DAC so that the Audio DAC Controller communicate with the flash memory directly. In this example, the **AUDIO_DAC** Verilog module defines a circuit that reads the contents of the flash memory and sends it to the external audio port.

VGA Display

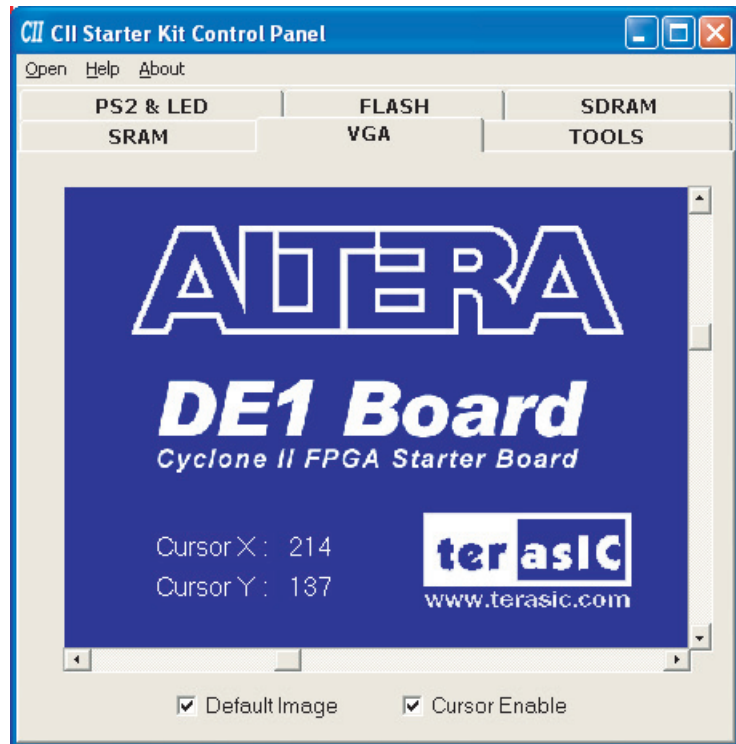
For this sample application, three examples illustrate how the development board produces images on a connected VGA monitor:

- Example 1 provides a default image previously loaded into an M4K memory block in the FPGA in the MIF/Hex (Intel) format during the default bit stream configuration stage (described in “[Configuring the Cyclone II FPGA](#)” section of the “[Using the Development Board](#)” chapter).
- Example 2 describes how to download another image from a bitmap file into the SRAM. The file can also load into an M4K memory block in the FPGA.
- Example 3 describes how to use other images to generate binary data patterns that the development board can display on the VGA monitor.

Displaying the Default Image

Perform the following steps to display a default image:

1. Select the **VGA** tab in the Control Panel to display the window in [Figure 4–7](#).

Figure 4–7. Control Panel VGA Tab Window

2. Ensure that the **Default Image** and **Cursor Enable** boxes are checked.
3. Connect a VGA monitor to the development board and observe the default image from the Control Panel displayed on the screen.
4. Confirm that the cursor appears on the screen; manipulate it by moving the X/Y-axes scroll bars on the Control Panel.

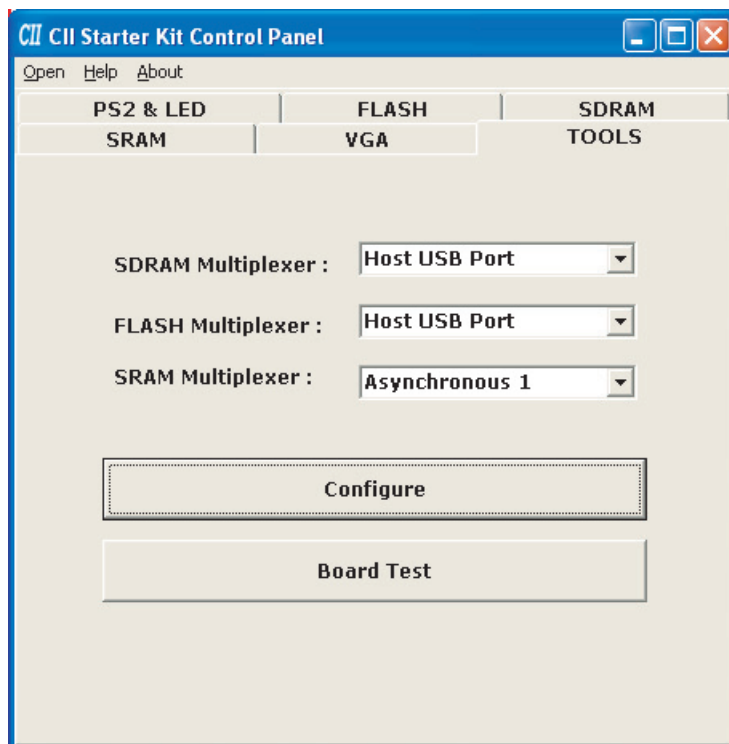
Displaying Another Image from a Downloaded Bitmap File

For another image to load, locate the **picture.dat** bitmap file in the **CII_Starter_demonstrations\pictures** directory of the **CII Starter System** CD-ROM. To display this image, perform the following steps:

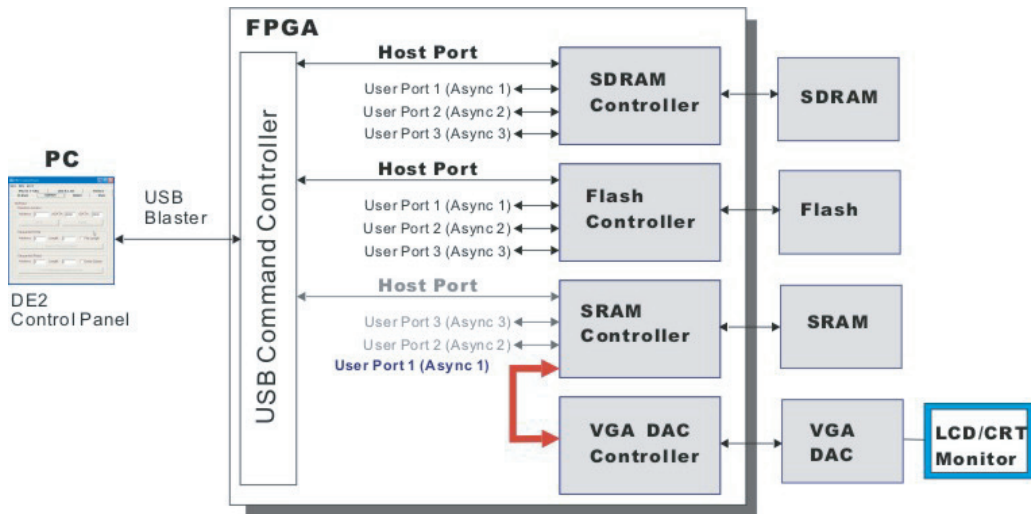
1. Select the **SRAM** tab of the Control Panel and load the file **picture.dat** into the SRAM.

2. Select the **TOOLS** tab and choose **Asynchronous 1** for the **SRAM Multiplexer** port (Figure 4-8).

Figure 4-8. Configuring the Port to Access the Image Data in the SRAM



3. Click on the **Configure** button to activate the multi-port setup in the FPGA (Figure 4-9).

Figure 4–9. Multiport Controller Configured to Display an Image from the SRAM

4. Select the **VGA** tab and deselect the checkbox **Default Image**.
5. Observe that the VGA monitor displays the **picture.dat** image from the SRAM (Figure 4–10). Turn off the cursor by deselecting the **Cursor Enable** checkbox.

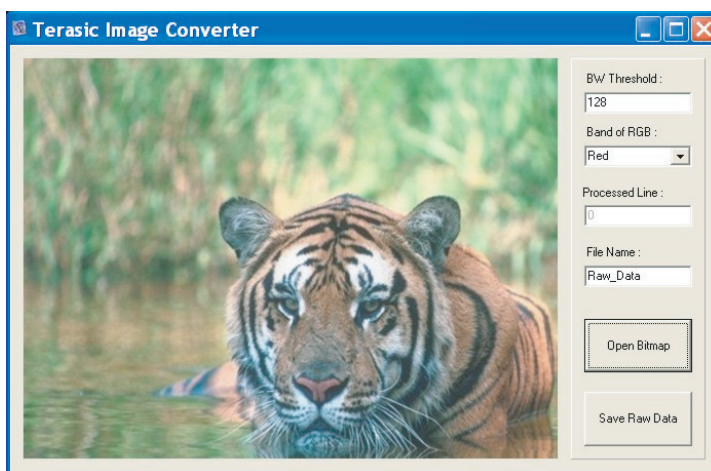
Figure 4–10. A Displayed Image

Displaying Any Image Files

Before loading any image file into SRAM memory or into an M4K memory block in the FPGA, first generate a bitmap file. Perform the following steps:

1. Load the desired image into an image processing tool, such as Corel® PhotoPaint®.
2. Resample the original image to have a 640 x 480 resolution. Save the modified image in the Windows bitmap format.
3. Run `CII_Starter_control_panel\ImgConv.exe`, an image conversion tool developed for the development board, to open the converter window (Figure 4-11).

Figure 4-11. The Image Converter Window



4. Click on the **Open Bitmap** button and select the 640 x 480 Grayscale photo for conversion.
5. When the processing of the file completes, click on the **Save Raw Data** button, which generates a file named **Raw_Data_Gray.dat** and stores in the same directory as the original image file. If desired, change the file name prefix from **Raw_Data** to another name by changing the **File Name** field in the displayed window.

6. Download the **Raw_Data_Gray.dat** file into the SRAM as described in “Displaying Another Image from a Downloaded Bitmap File” on page 4–12.

The **ImgConv** tool also generates a **Raw_Data_BW.dat** file (and its corresponding TXT format) for the black and white version of the image. The **BW Threshold** in Table 4–1 defines the threshold for judging black or white level. **Raw_Data_BW.txt** fills in the MIF/Intel Hex format for M4K SRAM.

<i>Table 4–1. BW Threshold</i>			
Image Source	R/G/B Band Filter	B&W Threshold Filter	Output Result (640x480)
Color Picture	R/G/B	N/A	Raw_Data_Gray
Color Picture	R/G/B (optional)	BW Threshold	Raw_Data_BW + Raw_Data_BW.txt
Grayscale Picture	N/A	N/A	Raw_Data_Gray
Grayscale Picture	N/A	BW Threshold	Raw_Data_BW + Raw_Data_BW.txt

This chapter provides instructions for using the development board and describes each of its I/O devices.

Configuring the Cyclone II FPGA

The Cyclone II FPGA Starter Development Board has integrated the programming circuitry normally found in a USB-Blaster programming cable, as well as a serial EEPROM chip (EPCS4) that stores configuration data for the Cyclone II FPGA. This configuration data loads automatically from the EEPROM chip into the FPGA each time power is applied to the board.

Using the Quartus II software, it is possible to reprogram the FPGA at any time, and it is also possible to change the non-volatile data stored in the serial EEPROM chip. The following sections describe the two ways to program the FPGA, JTAG programming and Active Serial (AS) programming.

JTAG Programming

In this method of programming, named after the IEEE standards *Joint Test Action Group*, the configuration bit stream downloads directly into the Cyclone II FPGA through the USB-Blaster circuitry. The FPGA retains this configuration as long as power is applied to the board; the FPGA loses the configuration when the power is turned off.



*For detailed information about the USB-Blaster circuitry, refer to the Cyclone II FPGA Starter Board schematic found in the **BoardDesignFiles / Schematic** directory in the kit installation directory.*

AS Programming

In the Active Serial programming method, the configuration bit stream downloads into the Altera EPCS4 serial EEPROM chip. The EEPROM provides non-volatile storage of the bit stream, retaining the information even when power to the Cyclone II FPGA Starter board is turned off. When the board powers up, the configuration data in the EPCS4 device automatically loads into the Cyclone II FPGA.

Configuration Procedure

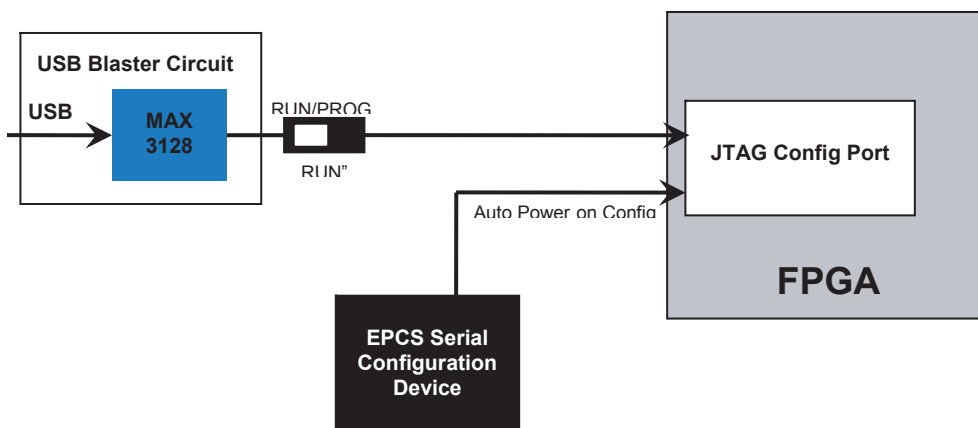
For both the JTAG and AS programming methods, the Cyclone II FPGA Starter board connects to a host computer via a USB cable. Because of this connection type, the host computer identifies the board as an Altera **USB-Blaster** device. The following sections describe the JTAG and AS programming steps.

Configuring the FPGA in JTAG Mode

Figure 5–1 illustrates the JTAG configuration setup. To download a configuration bit stream into the Cyclone II FPGA, perform the following steps:

1. Ensure that power is applied to the Cyclone II FPGA Starter board
2. Connect the supplied USB cable to the USB-Blaster port on the board
3. Configure the JTAG programming circuit on the board by setting the **RUN/PROG** switch (on the left side of the board) to the RUN position
4. To program the FPGA, use the Quartus II Programmer module to select a configuration bit-stream file with the **.sof** filename extension

Figure 5–1. The JTAG Configuration Scheme



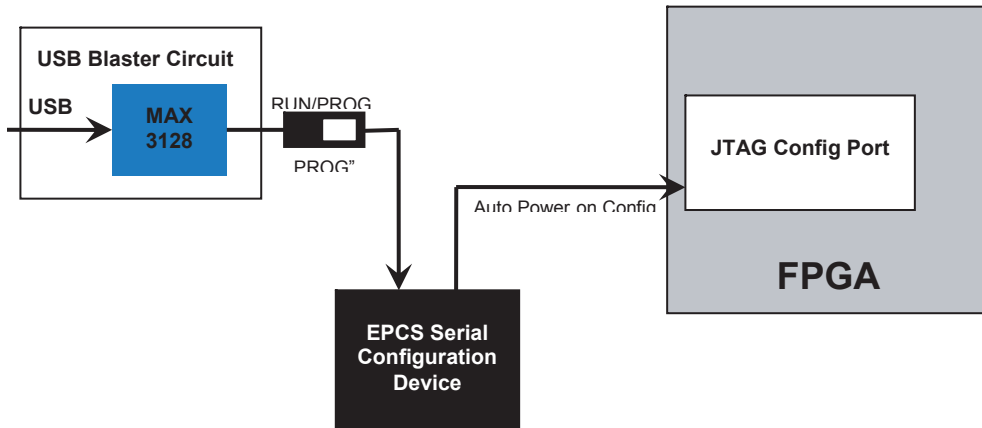
Configuring the EPCS4 Device in AS Mode

Figure 5–2 illustrates the AS configuration setup. To download a configuration bit stream into the EPCS4 serial EEPROM device, perform the following steps:

1. Ensure that power is applied to the Cyclone II FPGA Starter board
2. Connect the supplied USB cable to the USB-Blaster port on the board
3. Configure the JTAG programming circuit by setting the **RUN/PROG** switch (on the left side of the board) to the PROG position
4. To program the EPCS4 device, use the Quartus II Programmer module to select a configuration bit-stream file with the *.pof* filename extension
5. After the programming operation completes, set the **RUN/PROG** switch back to the RUN position
6. Reset the board by turning the power switch off and then on again. This action causes the new configuration data in the EPCS4 device to load into the FPGA chip.

Refer to the *Serial Configuration Devices* chapter in the *Altera Configuration Device Handbook* for more information about the EPCS4 device.

Figure 5–2. The AS Configuration Scheme



This chapter provides a number of examples of advanced circuits implemented on the development board. These circuits provide demonstrations of the major features on the board, such as its audio and video capabilities. For each demonstration, the kit includes the Cyclone II FPGA (or EPCS4 serial EEPROM) configuration file, as well as the full source code in Verilog HDL code.

All of the associated files reside in the **<kit path>\Examples** directory after installation. The section describing each demonstration provides the location for the example files.

Factory Configuration

The development board is shipped from the factory with a default configuration that demonstrates some of the basic features of the board.

File Locations

- Project directory: **CII_Starter_Default**
- Bit stream used: **CII_Starter_Default.sof** or **CII_Starter_Default.pof**

Demonstration Setup

To set up the demonstration, perform the following steps:

1. Power up the development board with the USB cable connected to the USB-Blaster port. If necessary (that is, if the default factory configuration of the development board is not currently stored in EPCS4 device), download the bit stream to the board by using either JTAG or AS programming
2. Observe that the 7-segment displays display a sequence of characters and that the red and green LEDs flash.
3. Optionally, connect a VGA display to the VGA D-SUB connector. When connected, the VGA display should show a pattern of colors.
4. Optionally, connect a powered speaker to the stereo audio-out jack.

5. Place toggle switch **SW9** in the UP position to hear a 1 kHz humming sound from the audio-out port. Alternatively, if switch **SW9** is DOWN, connect the **Mic In** port to a microphone to hear voice sounds or use the line-in port to play audio from an appropriate sound source.

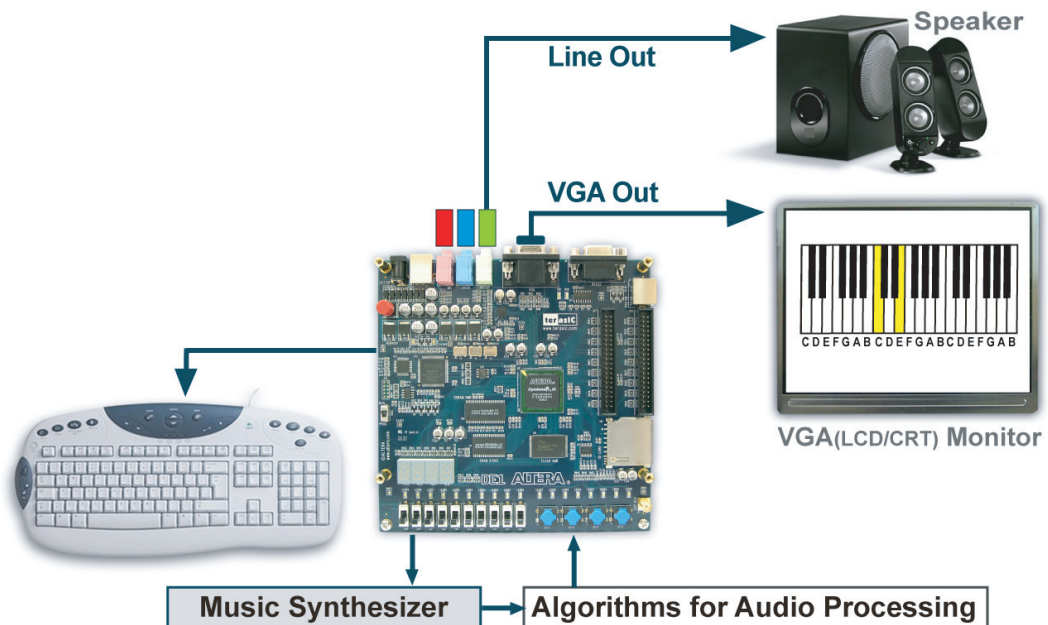
The **CII_Starter_Default** folder contains the Verilog source code for this demonstration. It also includes the necessary files for the corresponding Quartus II project. The top-level **CII_Starter_Default.v** Verilog file can serve as a template for other projects, because it defines ports that correspond to all of the user-accessible pins on the Cyclone II FPGA.

Music Synthesizer Demonstration

This demonstration shows how to implement a multi-tone electronic keyboard using the development board with a PS/2 keyboard and a speaker. Figure 6–1 shows the setup of the demonstration.

Use the PS/2 keyboard as the piano keyboard for input. The Cyclone II FPGA on the development board serves as the Music Synthesizer System on Chip (SOC) to generate music and tones. The VGA monitor connected to the development board shows which key is pressed during the playing of the music.

Figure 6–1. Music Synthesizer Demonstration Setup



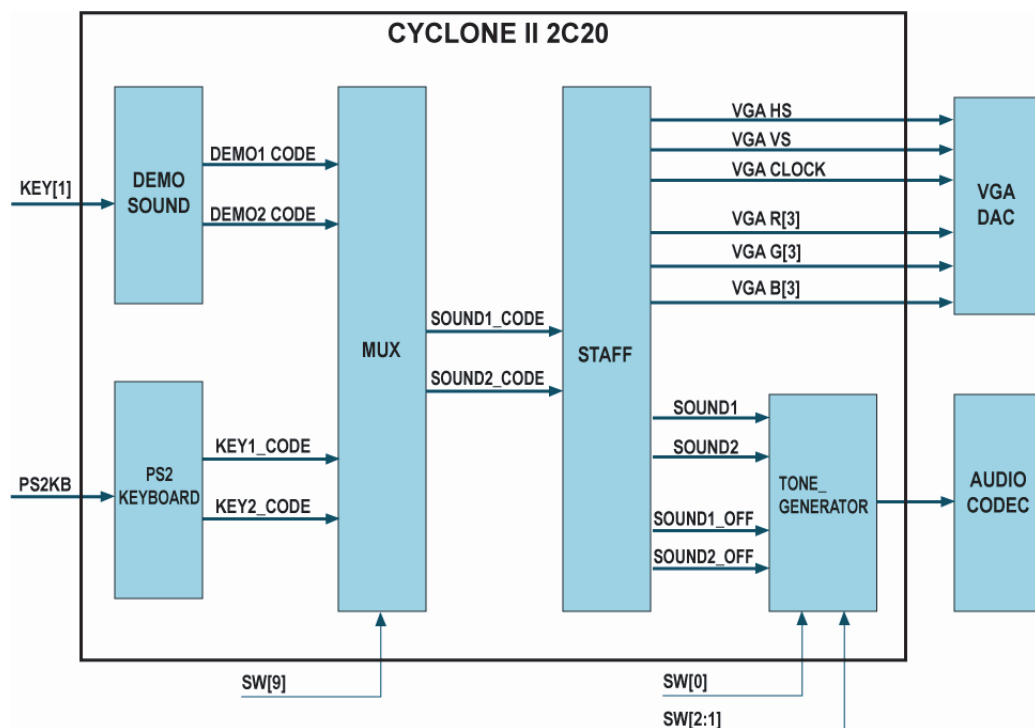
The block diagram (Figure 6–2) shows that the Music Synthesizer comprises four major blocks:

- **DEMO_SOUND** stores a sample sound for the user to play.
- **PS2_KEYBOARD** processes user input from the PS/2 keyboard.
- **STAFF** displays the corresponding keyboard diagram on the VGA monitor when a key is pressed
- **TONE_GENERATOR** represents the core of the music synthesizer SOC.

Use **SW9** to switch the music source between **PS2_KEYBOARD** and the **DEMO_SOUND** block. To repeat the demo sound, press **KEY1**.

The **TONE_GENERATOR** can produce two tones, (1) String and (2) Brass, selected by **SW0**. The audio CODEC used on the development board has two channels, turned ON/OFF by **SW1** and **SW2**.

Figure 6–2. Block Diagram of the Music Synthesizer Design



File Locations

- Project directory: **CII_Starter_Synthesizer**
- Bit stream used: **CII_Starter_Synthesizer.sof** or **CII_Starter_Synthesizer.pof**

Demonstration Setup

To set up the demonstration, perform the following steps:

1. Connect a PS/2 keyboard to the development board.
2. Connect the VGA output of the development board to a VGA monitor (both LCD and CRT type of monitors should work)
3. Connect a speaker to the **Line Out** port of the development board.
4. Load the bit stream into FPGA.
5. Ensure that all the switches, SW[9:0], are set to 0 (Down Position).
6. Press **KEY1** on the development board to start the music demo.
7. Press **KEY0** on the development board to reset the circuit.

Table 6–1 illustrates the usage of the switches and push buttons (KEYs).

Table 6–1. Switches and Push Buttons	
Signal Name	Description
KEY[0]	Reset
KEY[1]	Repeat the Demo Music
SW[0]	0: BRASS T 1: STRING
SW[9]	0: DEMO play 1: PS2 KEYBOARD play
SW[1]	0: Channel-1 ON 1: Channel-1 OFF
SW[2]	0: Channel-2 ON 1: Channel-2 OFF

Table 6–2 illustrates the usage of the PS/2 Keyboard.

Table 6–2. PS/2 Keyboard	
Signal Name	Description
Q	-#4
A	-5
W	-#5
S	-6
E	-#6
D	-7
F	1
T	#1
G	2
Y	#2
H	3
J	4
I	#4
K	5
O	#5
L	6
P	#6
:	7
"	+1

SD Card Music Player

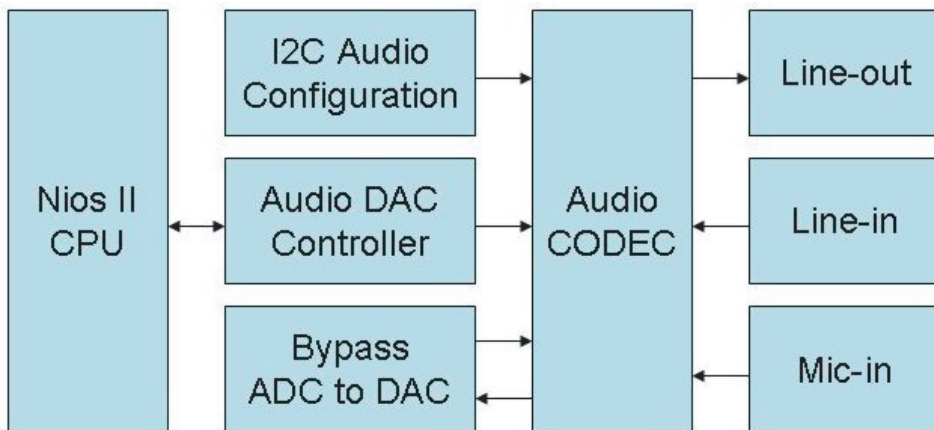
Many commercial media/audio players use a large external storage device, such as a Secure Digital (SD) card or Compact Flash (CF) card, to store music or video files. Such players may also include high-quality digital-to-analog converter (DAC) devices to produce good audio quality. The development board provides the hardware and software for SD card access and for the professional audio performance, which enable users to design advanced multimedia products on the development board.

This demonstration shows how to implement an SD Card Music Player on the development board. The sample music-player design stores the music files in an SD card and enables the board to play the music files via its CD-quality audio DAC circuits. The design uses the Nios II processor to read the music data stored in the SD Card and uses the Wolfson WM8731 audio CODEC to play the music.

Because the audio CODEC is configured in the slave mode, external circuitry must provide the ADC/DAC serial bit clock (BCK) and left/right channel clock (LRCK) to the audio CODEC. The block diagram (Figure 6–3) shows that the sample design provides an audio DAC controller for clock generation and data flow control. The audio DAC controller is integrated into the Avalon bus architecture, so that the Nios II processor can control the application.

During operation, the Nios II processor checks if the FIFO memory of the audio DAC controller becomes full. If the FIFO is not full, the processor reads a 512-byte sector and sends the data to the FIFO of the audio DAC controller via the Avalon bus. The audio DAC controller uses a 48 kHz sample rate to send the data and clock signals to the audio CODEC. The design also mixes the data from the **Mic In** port with data from the **Line In** port for the Karaoke-style effects.

Figure 6–3. SD Music Player Block Diagram



File Locations

- Project directory: **CII_Starter_SD_Card_Audio**
- Bit stream used: **CII_Starter_SD_Card_Audio.sof** (or **.pof**)
- Nios II Workspace: **CII_Starter_SD_Card_Audio**

Demonstration Setup

To set up the demonstration, perform the following steps:

1. Format the SD card into FAT16 format.
2. Playing a music file with this demonstration requires that the file use the WAV format. Copy one or more such WAV files onto the FAT16-formatted SD card. Due to a limitation in the software, to remove any WAV file requires reformatting the whole SD card.
3. Load the bit stream into the FPGA.
4. Run the Nios II IDE under the workspace **CII_Starter_SD_Card_Audio**.
5. Connect a headset or speaker to the development board and listen for music played from the SD Card.

Figure 6–4 illustrates the setup for this demonstration.

Figure 6–4. The Setup for the SD Music Player Demonstration

