Cyclone III Development Kit

User Guide

101 Innovation Drive
San Jose, CA 95134
www.altera.com

P25-36208-03

Document Version: 1.2
Document Date: March 2009

UG-01027-1.2

Contents

Additional Information

1. About This Kit

Introduction

Welcome to the Altera® Cyclone® III 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 kit provides an integrated control environment that includes a USB command controller, a multi-port SRAM/DDR SDRAM/flash memory controller, Ethernet, an on-board meter, and example designs with demonstration circuitry specified in Verilog code to help you get started quickly with your own designs.

The development board includes an Altera Cyclone III FPGA configured with a hardware reference design stored in flash memory. Hardware designers can use the development board, along with example designs included in the kit, as a platform to prototype complex embedded systems.

The development kit includes these kit features and documentation:

- Cyclone III development board
- Altera Complete Design Suite DVD containing:
  - Quartus® II Web Edition Software
  - MegaCore® IP Library
  - Nios® II Embedded Software Design Tools
- Cyclone III Development Kit CD-ROM including design examples
- Accessory daughter cards
- Power supply, cables, and documentation

Kit Features

This section briefly describes the Cyclone III Development Kit features.

- **Cyclone III Development Board**—a prototyping platform that allows you to develop and prototype high-speed bus interfaces as well as evaluate Cyclone III transceiver performance.

  For specific information about board components and interfaces, refer to the *Cyclone III 3C120 Development Board Reference Manual*.

- **Quartus II Web Edition Software**—The Quartus II software (available on the DVD) integrates into nearly any design environment, with interfaces to industry-standard EDA tools. The kit includes:
  - The SOPC Builder system development tool
  - Free Quartus II Web Edition software license, Windows platform only

For more information, refer to the Altera website at *www.altera.com/products/software/products/quartus2web/sof-quarwebmain.html*.

- **MegaCore IP Library**—This library (available on the DVD) contains Altera IP MegaCore functions. You can evaluate MegaCore functions by using the OpenCore® Plus feature to do the following:
  - Simulate behavior of a MegaCore function within your system
  - Verify functionality of your design, and quickly and easily evaluate its size and speed
  - Generate time-limited device programming files for designs that include MegaCore functions
  - Program a device and verify your design in hardware
- **Nios II Embedded Software Design Tools**—This full-featured set of tools (available on the DVD) allows you to develop embedded software on the Nios II processor running on Altera FPGAs.
- **Cyclone III Development Kit Application and Drivers**—The application and drivers (available on the CD-ROM) allow you to execute memory read and write transactions to the board.
- **Design Examples**—The design examples (available on the CD-ROM) are useful for a variety of hardware applications and let you quickly begin board prototyping and device verification.

You only need to purchase a license for a MegaCore function when you are completely satisfied with its functionality and performance, and want to take your design to production.

The OpenCore Plus hardware evaluation feature is an evaluation tool for prototyping only. You must purchase a license to use a MegaCore function in production.

For more information about OpenCore Plus, refer to *AN 320: OpenCore Plus Evaluation of Megafunctions*.

Documentation

The Cyclone III Development Kit CD-ROM contains the following documents:

- *Readme.txt*—Contains special instructions and refers to the kit documentation.
- *Cyclone III Development Kit User Guide* (this document)—Describes how to use the kit.
- *Cyclone III 3C120 Development Board Reference Manual*—Provides specific information about the board components and interfaces, steps for using the board, and pin-outs and signal specifications.

2. Getting Started

Introduction

This user guide familiarizes you with the contents of the kit and guides you through the Cyclone III development board setup. Using this guide, you can do the following:

- Inspect the contents of the kit
- Install the Altera Development Suite Tools
- Set up licensing
- Install the Cyclone III Development Kit CD-ROM
- Set up, power up, and verify correct operation of the development board
- Configure the Cyclone III FPGA
- Find and use the tutorials
- Set up and run included application examples and demonstrations

For complete information about the development board, refer to the *Cyclone III 3C120 Development Board Reference Manual*.

Before You Begin

Before using the kit or installing the software, check the kit contents and inspect the board to verify that you received all of the items listed in this section. If any of the items are missing, contact Altera before you proceed.

Check the Kit Contents

The Cyclone III Development Kit (ordering code: DK-DEV-3C120N) contains the following items:

- Cyclone III development board with an EP3C120F780C7N Cyclone III device
- Altera Complete Design Suite DVD containing:
  - Quartus II Web Edition Software
  - MegaCore IP Library
  - Nios II Embedded Software Design Tools

- Cyclone III Development Kit CD-ROM, which includes:
  - Reference designs and demonstrations
  - Cyclone III development kit application and device drivers
  - Design examples
  - *Cyclone III 3C120 Development Board Reference Manual*
  - *Cyclone III Development Kit User Guide* (this document)
  - Device datasheets and tutorials
  - Schematic and board design files
- USB cable
- Accessory daughter cards:
  - Two loopback high-speed mezzanine connector (HSMC) cards
  - Debug HSMC card
  - One 16 character × 2 line Liquid Crystal Display (LCD)
  - One 2.8-in. thin-film transistor (TFT) LCD module
- 16-V DC Power supply and adapters for North America, Europe, the United Kingdom, and Japan.

To ensure that you have the most up-to-date information about this product, go to the Altera website at *www.altera.com/products/devkits/altera/kit-cyc3-Development.html*.

Inspect the Board

Place the board on an anti-static surface and inspect it to ensure that it has not been damaged during shipment.

CAUTION

Without proper anti-static handling, the Cyclone III development board can be damaged.

Verify that all components are on the board and appear intact.

In typical applications with the Cyclone III development board, a heatsink is not necessary. However, under extreme conditions the board may require additional cooling to stay within operating temperature guidelines. You may wish to perform power consumption and thermal modeling to determine whether your application requires additional cooling.

For more information about power consumption and thermal modeling, refer to *AN 358: Thermal Management for FPGAs*.

Hardware Requirements

The Quartus II software has some minimum system requirements. Otherwise, the Cyclone III development kit provides all of the hardware needed to use the board.

For Quartus II requirements, refer to the Altera website at *www.altera.com/products/software/products/quartus2web/sof-quarwebmain.html*.

Software Requirements

This kit requires the following software:

- Windows XP operating system
- Quartus II Web Edition software (refer to the **readme.txt** file on the CD-ROM for specific version requirements)

Although it is already available on the DVD included in the kit, you can also download the Quartus II software from the Altera website at *www.altera.com/products/software/products/quartus2web/sof-quarwebmain.html*.

Refer to *Quartus II Installation & Licensing for Windows* for further information on the Quartus II system software requirements, especially heeding the following:

- A web browser, Microsoft Internet Explorer version 5.0 or later or Firefox version 2.0 or later. You need a web browser to register the Quartus II software and request license files. Refer to "Licensing Considerations" on page 3–3.
- Version 2.0 or later of the .NET framework.

If you receive an "Application Error" message when launching the demo application, install version 2.0 or later versions of the .NET framework. Some Windows versions do not have runtime DLL for the .NET application. You can download the .NET framework application from the following location: *www.microsoft.com/download*.

References

For other related information, refer to the following websites:

- For additional daughter cards available for purchase: www.altera.com/products/devkits/kit-daughter_boards.jsp
- For the Cyclone III handbook: www.altera.com/literature/lit-cyc3.jsp
- For the Cyclone III reference designs: www.altera.com/endmarkets/refdesigns/device/cyclone3/cyclone3-index.jsp
- For eStore if you want to purchase devices: www.altera.com/buy/devices/buy-devices.html
- For Cyclone III Orcad symbols: www.altera.com/support/software/download/pcb/pcb-pcb_index.html

- For Nios II 32-bit embedded processor solutions:
  www.altera.com/technology/embedded/emb-index.html

3. Software Installation

Introduction

The instructions in this chapter explain how to install the following:

- Cyclone III Development Kit CD-ROM
- Cyclone III Development Kit demo application and drivers
- The Quartus II Web Edition Software DVD, including MegaCore functions from the MegaCore IP Library

Before starting the installation, verify that you have complied with the conditions described in "Software Requirements" on page 2–3.

Installing the Development Kit CD-ROM

The Cyclone III Development Kit CD-ROM contains the following items:

- The sample design and board design files
- *Quick Start Guide*
- *My First FPGA Design Tutorial*
- *My First Nios II Software Tutorial*
- *Cyclone III Development Kit User Guide* (this document)
- *Cyclone III 3C120 Development Board Reference Manual*

To install the Cyclone III Development Kit CD-ROM, perform the following steps:

1. Insert the Cyclone III 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 on-screen instructions to complete the installation process.

The installation program copies the Cyclone III development kit files to the computer hard disk and creates a **Programs > Altera > Cyclone III Development Kit <*version#*>** Windows Start menu shortcut. Use this shortcut to launch the development kit graphical user interface (GUI).

When the installation is complete, the Cyclone III Development Kit installation program creates the directory structure shown in Figure 3–1, where <*path*> is the Cyclone III Development Kit installation directory.

Figure 3–1. Cyclone III Development Kit Installed Directory Structure

Table 3–1 lists the file directory names and a description of their contents.

Table 3–1. Installed Directory Contents

Directory Name	Description of Contents
board_design_files	Contains schematic, layout, assembly, and bill of material board design files. Use these files as a starting point for a new prototype board design.
demos	Contains demonstration applications that may change from release to release.
documents	Contains the development kit documentation.
examples	Contains the sample design files for the Cyclone III Development Kit.
factory recovery	Contains the original data programmed onto the board before shipment. Use this data to put the board into the original condition.

Installing the Quartus II DVD

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

Additionally, you may want to install the Nios II Embedded Design Suite found in the Altera Complete Design Suite at *www.altera.com/support/software/download/nios2/dnl-nios2.jsp*. The Nios II soft-core embedded processor runs on Altera FPGAs. Some of the reference designs included in this development kit use the Nios II processor.

Load the Altera Complete Design Suite DVD into the DVD player, and click **Install free package** on the startup screen. Follow the on-screen instructions and accept all default settings. After installing the software, request and install a license to enable it.

For information about obtaining a license file, refer to "Licensing Considerations" on page 3–3.

Installing the USB-Blaster Driver

The Cyclone III development board includes integrated USB-Blaster™ circuitry for FPGA programming. However, for the host computer and development board to communicate, you must install the USB-Blaster driver on the host computer.

To download the USB-Blaster driver, go to the Altera support site at www.altera.com/support/software/drivers/dri-index.html.
To install it, go to www.altera.com/support/software/drivers/usb-blaster/dri-usb-blaster-xp.html.

Licensing Considerations

Before using the Quartus II software, you must request a license file from the Altera website at www.altera.com/licensing and install it on your computer. When you request a license file, Altera emails you a **license.dat** file that enables the software.

To license the Quartus II software, you need your network interface card (NIC) ID, a 12-digit hexadecimal number that identifies your computer. Networked (or floating-node) licensing requires a NIC ID or server host ID. When obtaining a license file for network licensing, use the NIC ID from the computer that will issue the Quartus II licenses to distributed users over a network. You can find the NIC ID for your card by typing `ipconfig/all` at a command prompt. Your NIC ID is the number on the physical address line, without the dashes.

4. Development Board Setup

Introduction

The instructions in this chapter explain how to install the development board and configure the FPGA.

Requirements

Before starting the installation, verify that you have complied with the conditions described in "Hardware Requirements" on page 2–3 and have completed the following requirements:

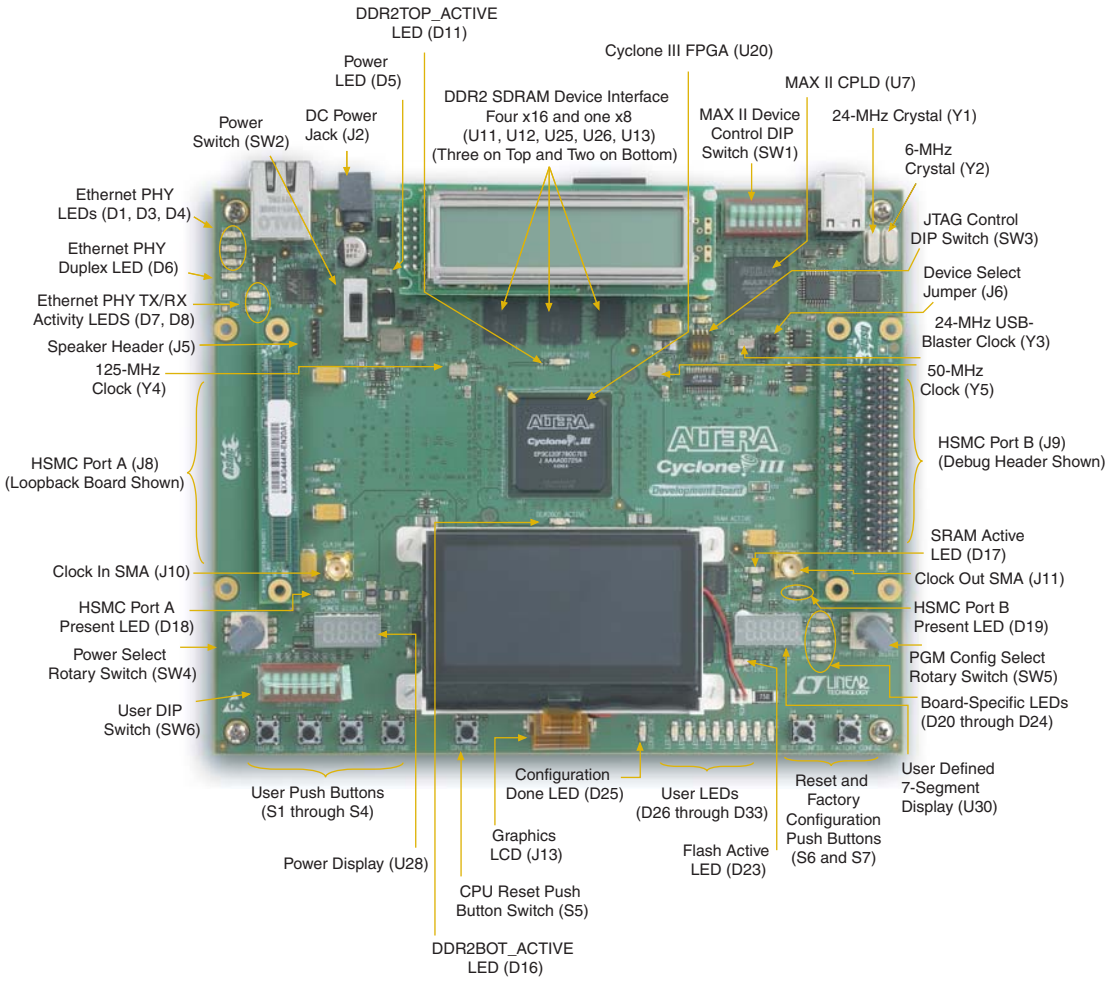
- Quartus II software installed on the host computer
- USB-Blaster driver software installed on the host computer.

The Cyclone III development board includes integrated USB-Blaster circuitry for FPGA programming. Host computer and development board cannot communicate without the USB-Blaster driver software installed. For installation information, refer to "Installing the USB-Blaster Driver" on page 3–3.

Powering Up the Board

Figure 4–1 shows the Cyclone III development board and its components.

Figure 4–1. Cyclone III Development Board Layout and Components



Before powering up, prepare the board by performing the following steps:

1. If cards are plugged into the high-speed mezzanine connector (HSMC) ports, remove them (Figure 4–1) shows a daughter card plugged into both port A and port B).
2. Ensure that the POWER switch SW2 is in the OFF (or DOWN) position.
3. Configure the 8-position SW1 DIP switch to the default settings in Table 4–1.

Table 4–1. Switch SW1 Settings (Part 1 of 2)

Switch	Name	Function		Default Position
		Position 0	Position 1	
1	mW/mA	mW	mA	0
2	V/W	V	W	1
3	RSV0	MAX_RESERVE0		X
4	RSV1	MAX_RESERVE1		X

Table 4–1. Switch SW1 Settings (Part 2 of 2)

Switch	Name	Function		Default Position
		Position 0	Position 1	
5	MAX0	PFL Disable	PFL Enable	1
6	MAX1	MAX_DIP1		X
7	MAX2	MAX_DIP2		X
8	MAX3	MAX_DIP3		X

Note to Table 4–1:

(1) X = don't care

4. Ensure that the 4-position SW3 mini-DIP switches and the two jumpers are set to the default positions shown in Table 4–2.

Table 4–2. Initial Switch and Jumper Settings

DEV_SEL-J6	JTAG_SEL-J7	SW3.1	SW3.2	SW3.3	SW3.4
ON	ON	ON	OFF	OFF	OFF

5. Verify that the PGM CONFIG SELECT rotary switch SW5 is set to 0.

At power up, the development board uses a preloaded configuration to demonstrate that the board is operating correctly.

Power up the development board by performing the following steps:

1. Connect the 16-V DC adapter to the development board and to a power source.

Use only the supplied 16-V power supply. Power regulation circuitry on the board could be damaged by supplies greater than 16 V.

2. Slide the POWER switch to ON. The nearby blue POWER light-emitting diode (LED) lights up.
3. Confirm that user LEDs 0-7 flash in a scrolling, side-to-side pattern. For customized configurations, the pattern depends on the application.

For information about custom configurations, refer to "Programming the Flash Device" on page A–4.

Configuring the FPGA

Before configuring the FPGA, ensure that the Quartus II software and the USB-Blaster driver software are installed on the host computer and the development board is powered on.

For USB-Blaster driver installation information, refer to "Installing the USB-Blaster Driver" on page 3–3.

To configure the Cyclone III FPGA, perform the following steps:

1. Verify that the 4-position SW3 mini-DIP switches and the two jumpers are set to the default positions shown in Table 4–2.
2. Connect the USB cable to the development board USB port.
3. Cycle the POWER switch OFF then ON.
4. Start the Quartus II software.
5. On the Tools menu, click **Programmer**. The Quartus II Programmer appears.
6. Click **Add File** and select *<path>***...\examples\cycloneIII_3c120_dev_my_first_fpga\cycloneIII_3c120_dev_my_first_fpga.sof**.
7. Turn on **Program/Configure** to select the added file.
8. Click **Start** to download the selected file to the FPGA. The FPGA is configured when the progress bar reaches 100%.
9. Confirm that user LEDs 0-3 flash in a slow binary counting pattern.

5. Power Measurement

Introduction

One of the main features of the Cyclone III FPGA device is its low power consumption. You can measure the power used by the 3C120 FPGA device on the Cyclone III development board for various conditions with a power design example provided with the kit.

With the power design example you can control the amount of logic utilized in the FPGA, the clock frequency, and the number of I/Os used, and measure the effect on power used by the Cyclone III device.

Power Design Example

The power design example uses a replicated module, **stamp.v**, that contains combinational logic, randomly filled ROMs, multiplier blocks, and shift registers that change with every clock cycle. The frequency and resource states indicated in Table 5–2 and Table 5–3 on page 5–2 , respectively, represent the percent of the full design used. As compiled, the full example design uses the following FPGA resources:

- Total logic elements: 111,673 / 119,088 (94%)
- Total combinational functions: 12,293 / 119,088 (10%)
- Dedicated logic registers: 104,192 / 119,088 (87%)
- Total registers: 104,192
- Total pins: 165 / 532 (31%)
- Total virtual pins: 0
- Total memory bits: 2,490,368 / 3,981,312 (63%)
- Embedded multiplier 9-bit elements: 576 / 576 (100%)
- Total phase-locked loops (PLLs): 1/4 (25%)

Table 5–1 describes the functionality of the four user push buttons that control the power design example. The on-board 50-MHz oscillator provides the input clock (`i_clk`, `PIN_AH15`).

Table 5–1. Four Input Button Functionality

User Push Button	FPGA Pin	Type	Description
User_PB0	AD7	Reset	Resets the demo to the beginning, `i_nrst`
User_PB1	AC12	Toggle	Advances the design example to the next higher frequency, `i_nfreq_next`
User_PB2	AH3	Toggle	Advances the design example to the next higher resource utilization, `i_nperc_next`
User_PB3	AA12	Toggle	Enables the outputs to toggle, `i_noutput_ena`

The LEDs in Table 5–2 indicate the power-state values of the design example as User_PB1 advances frequency.

Table 5–2. Power State Indicators for Frequency

LED1 (AE20)	LED0 (AD15)	Frequency (MHz)
0	0	0
0	1	33
1	0	67
1	1	100

The LEDs in Table 5–3 indicate the power-state values of the design example (and number of output pins, when enabled by User_PB3) as User_PB2 advances resource utilization.

Table 5–3. Power State Indicators for Resources

LED6 (AG19)	LED5 (AC17)	LED4 (AE15)	LED3 (AD19)	LED2 (AF18)	Resources (%)	Number of Outputs
0	0	0	0	0	5	8
0	0	0	0	1	10	16
0	0	0	1	0	15	24
0	0	0	1	1	20	32
0	0	1	0	0	25	40
0	0	1	0	1	30	48
0	0	1	1	0	35	56
0	0	1	1	1	40	64
0	1	0	0	0	45	72
0	1	0	0	1	50	80
0	1	0	1	0	55	88
0	1	0	1	1	60	96
0	1	1	0	0	65	104
0	1	1	0	1	70	112
0	1	1	1	0	75	120
0	1	1	1	1	80	128
1	0	0	0	0	85	136
1	0	0	0	1	90	144
1	0	0	1	0	95	152

The resource state (Table 5–3) controls the number of I/O pins used. Each resource increment adds 8 additional I/O pins (Table 5–3). Similarly, the overall design frequency (Table 5–2) sets the toggle frequency of these I/O pins.

Measuring Power

You can measure power by using the analog-to-digital (A/D) circuitry on the development board or by using a digital multi-meter (DMM) across on-board sense resistors. However, note that, depending on the DMM accuracy, the on-board A/D measurements tend to produce considerably more accurate results.

A/D Measurements

The POWER SELECT rotary switch SW4 sets the development board to measure and display FPGA core power or I/O output power (Table 5–4).

Table 5–4. Switch SW4 Power Selection

Switch Position	FPGA Power	I/O Banks
0	Core: V_{CC_INT}, 1.2 V	—
5	I/O: 2.5 V	1 and 2
6	I/O: 2.5 V	5 and 6

Measuring V_{CC_INT} Power

To measure FPGA core power at V_{CC_INT} = 1.2 V for various power states, perform the following steps:

1. Ensure that the 8-position SW1 DIP switch is configured to the default settings shown in Table 4–1 on page 4–2.
2. Download the **cycloneIII_3c120_dev_power_demo.sof** file as described in "Configuring the FPGA" on page 4–3. You can find the power design example in *<path>*\...**\examples\cycloneIII_3c120_dev_power_demo**.
3. Set the POWER SELECT rotary switch SW4 to 0 to measure the internal V_{CC_INT} power in watts.
4. Observe the power on the 4-digit hexadecimal power display.
   The on-board power measurements are performed at a high rate. For this reason, it appears that the display is dithering; however, what is actually taking place is the very fast momentary changes on the power rail.
5. Using the user input push buttons (Table 5–1 on page 5–1), advance through the power states in Table 5–2 and Table 5–3 on page 5–2. Notice how power increases as frequency and resources increase.

For information about measuring power sources, refer to the *Cyclone III 3C120 Development Board Reference Manual*.

Measuring I/O Power

This example uses FPGA I/O banks 1, 2, 5, and 6. Using the SW4 settings (Table 5–4), measure the power for I/O banks 1 and 2, then for I/O banks 5 and 6, by performing the following steps:

1. Ensure that the 8-position SW1 DIP switch is configured to the default settings shown in Table 4–1 on page 4–2.

2. Download the **cycloneIII_dev_powerdemo.sof** file as described in "Configuring the FPGA" on page 4–3. The power design example is in *<path>*\...**examples\cycloneIII_3c120_dev_powerdemo**.
3. Set the POWER SELECT rotary switch SW4 to 5.
4. Observe the 4-digit hexadecimal display for the I/O output power in watts on banks 1 and 2.
5. Using the user input push buttons (Table 5–1 on page 5–1), advance through the power states in Table 5–2 and Table 5–3 on page 5–2. Notice how power increases as frequency and resources increase.
6. Press the **User_PB3** button to enable the output pins on the HSMC connectors J8 and J9. LED7 (AF19), signal `o_noutput_ena_state`, lights to indicate that the outputs are enabled and toggling. Enabling the outputs further increases power for each resource utilization percentage used by the Cyclone III FPGA.
7. Press the **User_PB3** button again to disable the outputs. LED7 turns off.
8. Set the POWER SELECT rotary switch SW4 to 6.
9. Repeat steps 4 through 7 to observe the I/O output power in watts on banks 5 and 6.

The sum of power results from the two sets of I/O banks provides the total FPGA I/O power.

For specific information about on-board measurements and the POWER SELECT rotary switch, refer to the *Cyclone III 3C120 Development Board Reference Manual*.

Measuring Voltage with a DMM

To obtain power values by using a DMM, measure voltage across the sense resistors, R49, R48, and R51 on the board, then use the voltage measurements to calculate power.

For best results, use a DMM with six-digit for greater accuracy.

FPGA I/O power is distributed by banks (Table 5–5), for which the sense resistors are components R48 and R51. For the I/O power calculation, use the sum of voltage measurements across these resistors while outputs are enabled. For the FPGA core power calculation, measure the sense resistor voltage across R49.

Table 5–5. Sense Resistors

FPGA Power		Voltage	Sense Resistor	Resistor Value
I/O	Banks 1 and 2	2.5 V	R48	0.009 Ω
	Banks 5 and 6	2.5 V	R51	0.009 Ω
Core: V_{CC_INT}		1.2 V	R49	0.009 Ω

Calculating Power

To obtain the power P in watts, measure the voltage across the sense resistors, V_{SENSE}, and calculate the nominal power as follows:

If

V_{SENSE} = Voltage measured across the sense resistor
I_{SENSE} = Current through the sense resistor
V_{SUPPLY} = FPGA supply voltage
R_{SENSE} = Sense resistor value in Table 5–5.

then

Equation 5–1.

$$P = VI = V_{SUPPLY} \times I_{SENSE} = \frac{V_{SUPPLY} \times V_{SENSE}}{R_{SENSE}}$$

where

V_{SUPPLY} is 1.2 V for the FPGA core and 2.5 V for FPGA I/O.

Voltage measurements on the DMM should increase as frequency and resource utilization increases according to Table 5–2 and Table 5–3 on page 5–2.

Changing the Design Example

The development kit includes source code for the Cyclone III power design example so you can use it as a starting point for your own measurements.

The design example uses 19 **stamp.v** modules, each with 8 outputs, for a total of 152 output pins. It assigns the pins selected as outputs to the HSMC connectors J8 and J9.

To change the number of outputs, modify the design example and assign the pins appropriately.

Power should track linearly with frequency and percentage resources. If you observe superlinear power measurements, some temperature issue may be the cause.

6. Design Tutorials

Introduction

The example designs and tutorials included in the Cyclone III Development Kit help familiarize new users with development board features. *My First FPGA Design Tutorial* and *My First Nios II Software Tutorial* provide step-by-step guidance for the first-time user.

My First FPGA Tutorial

My First FPGA Design Tutorial describes how to create a simple Altera FPGA design. The tutorial takes less than an hour to complete and provides an overview of the design flow using the Quartus II software to build a simple logic counter which drives LEDs to flash on the development board.

After installing the Cyclone III Development Kit CD-ROM, the *My First FPGA Design Tutorial* design and documentation can be found at <*path*>\... **\documents\tutorials\hardware_tutorials\my_first_fpga_tutorial.pdf**.

My First Nios II Software Tutorial

My First Nios II Software Tutorial introduces the basic system development flow for the Nios II processor. This tutorial provides a good starting point if you are new to the Nios II processor or to the general concept of using an embedded processor in an FPGA. In this tutorial, you use a standard, existing Nios II hardware system and create a software program to run on it.

After installing the Cyclone III Development Kit CD-ROM, the *My First Nios II Software Tutorial* design and documentation can be found at <*path*>\...**\documents\tutorials\software_tutorials\my_first_nios2_software_tutorial.pdf**.

A. Programming the Flash Device

Overview

There is a Common Flash Interface (CFI) type flash memory device on the Cyclone III development board. When you first receive the kit, this CFI flash device arrives programmed with a default factory configuration that was loaded from a Programmer Object File (**.pof**).

When you power up the board, the CFI flash device configures the FPGA with the default factory configuration using Passive Serial (PS) programming. If the configuration loads correctly, the user LEDs on the board flash sequentially from side to side.

As you develop your own project using the Altera tools, you may wish to program the flash device so that, upon power up, it loads the FPGA with your own design. Or you may wish to restore the default factory configuration to your board.

This appendix describes how to program the flash device. You can load an existing design from a **.pof**, but if your design exists only as an SRAM Object File (**.sof**), then first you must convert the **.sof** to a **.pof**. Programming the flash device also requires the use of the Altera parallel flash loader (PFL). Using this appendix, you can do the following:

- Create a flash file by converting a **.sof** to **.pof**
- Install the PFL
- Use the Quartus II Programmer to write a **.pof** to the flash device
- Restore the default factory configuration

Creating a Flash File

To create a flash-programmable configuration **.pof**, perform the following steps:

1. On the File menu in the Quartus II software, click **Convert Programming Files**.
2. In the **Convert Programming Files** dialog box, select the parameter values as shown in the following table:

Parameter	Value
Programming file type	Programmer Object File (**.pof**)
Configuration device	CFI_512MB
Mode	1-bit Passive Serial
File name	<*output_file.pof*> This is the default file name. Change this to the file name you wish to use for your application. Save the file in <*path*>\...**\examples\cycloneIII_3c120_dev_pfl**.
Memory Map File	selected (default)

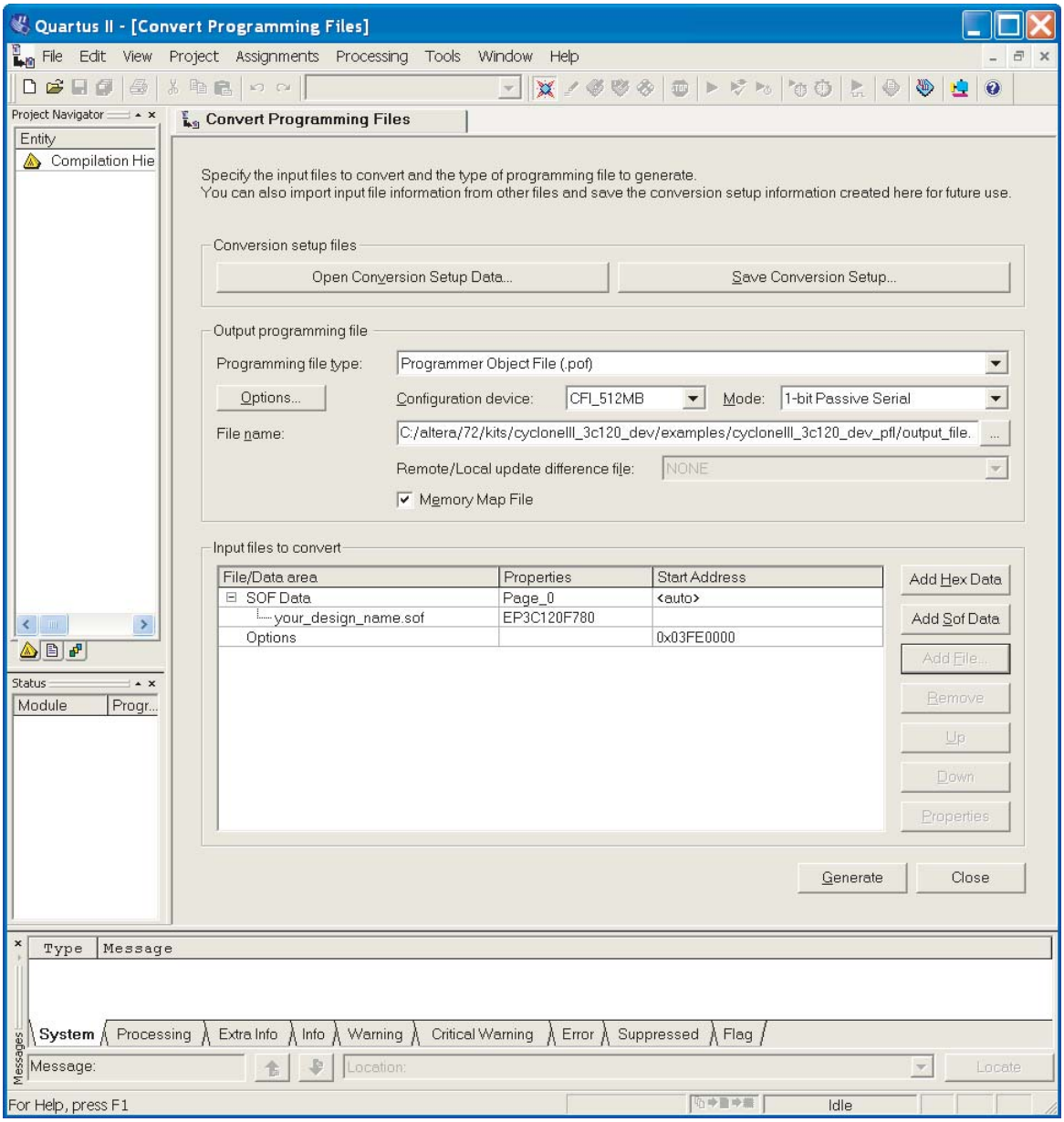
3. Click **Options**. In the **Options** dialog box, enter `0x3FE0000` and click **OK**. This sets the option bit base address for the development kit to the required default, `0x3FE0000`. The option bit sector stores the start address for each page of memory and also stores the Page Valid bits. The Page Valid bits indicate whether each page is successfully programmed.

   For more information about option bits and Page Mode Implementation of memory, refer to *AN 386: Using the Parallel Flash Loader with the Quartus II Software*.

4. Choose the **.sof** file you want to convert by selecting the row labeled **SOF Data** in the **Input files to convert** area and click **Add File**.
5. Browse to and select the file you wish to convert and click **OK**. Figure A–1 shows the **Convert Programming Files** dialog box updated with the factory image **your_design_name.sof**.

   If you choose to overwrite an existing **.pof** file, you receive a warning message.

Figure A–1. Convert Programming Files Settings



6. Click **Generate**. Generation takes a short time and it is confirmed by a "Generated… pof successfully" message.

You now have a successfully generated **.pof** that can be programmed to the flash device to automatically configure the FPGA on your Cyclone III development board.

Parallel Flash Loader Instantiation

The development kit includes a PFL megafunction design, **cycloneIII_3c120_dev_pfl,** in the directory <*path*>\...**examples**. The Quartus II software uses the PFL to write programming files to the flash device, which then loads the FPGA on power up.

To write to a flash device, you must first program the PFL into the FPGA by using the Quartus II software as described in "Programming the Flash Device", steps 1 through 8.

For more information about the PFL megafunction, refer to *AN 386: Using the Parallel Flash Loader with the Quartus II Software*.

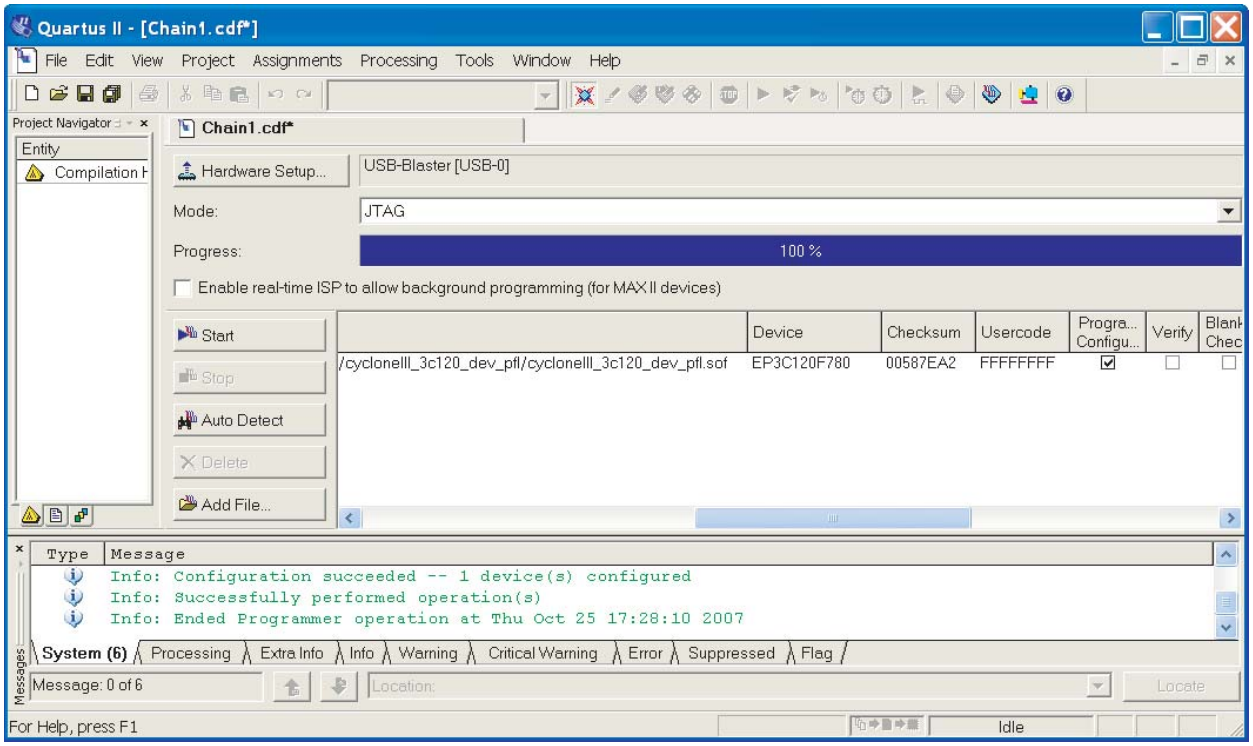
Programming the Flash Device

To program the flash device on the development board, you must first create a **.pof** flash file as described in "Creating a Flash File" on page A–1. The following procedure describes programming the PFL into the FPGA first, then uses the PFL to write the **.pof** flash file into the flash device.

To download a configuration bit stream into the flash device, perform the following steps:

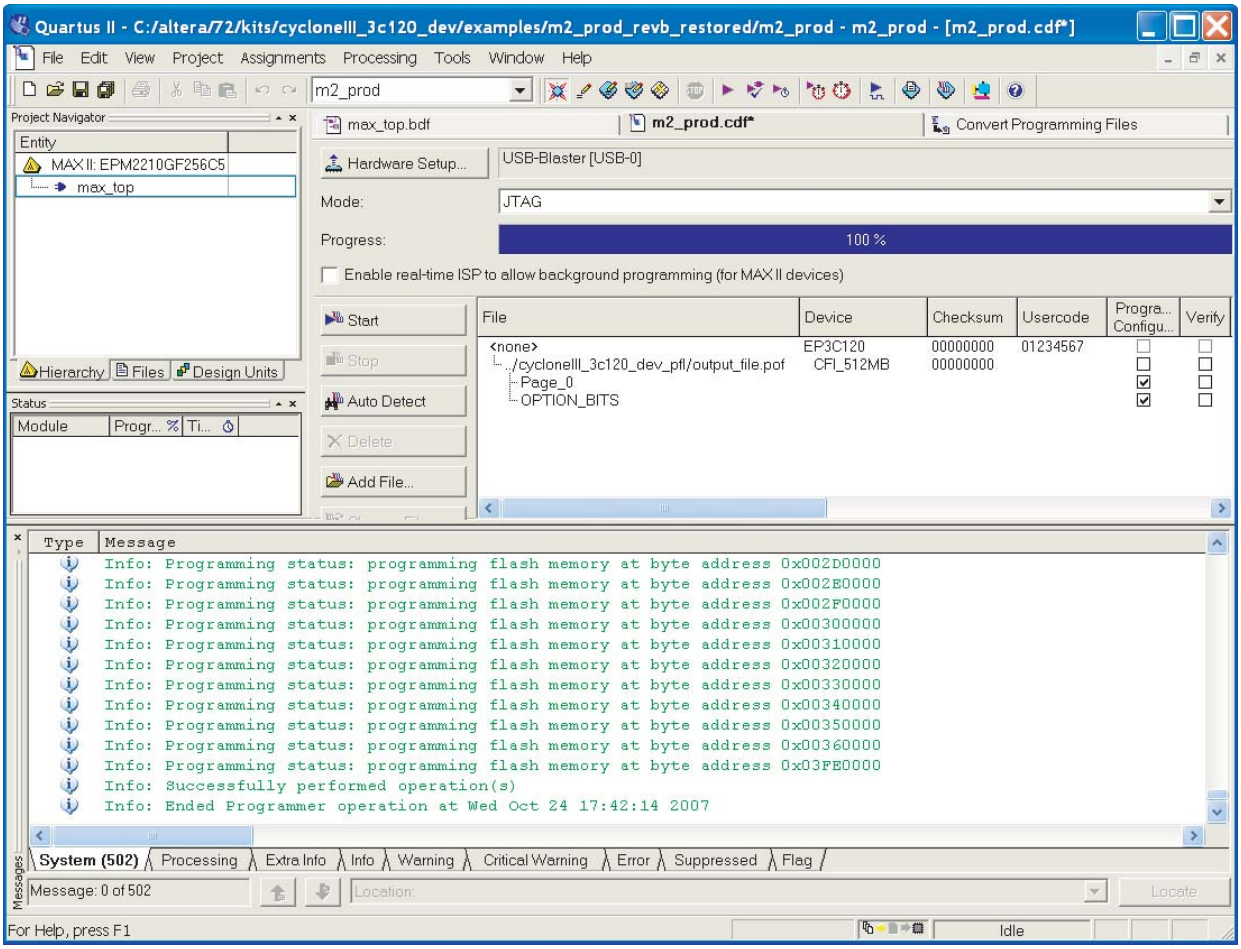
1. Ensure that the POWER switch SW2 is in the OFF (or DOWN) position.
2. Verify the switch SW3 and jumper settings shown in Table 4–2 on page 4–3.
3. Connect the USB cable to the USB port on the board.
4. Cycle the POWER switch OFF then ON.
5. On the Tools menu in the Quartus II software, click **Programmer**.
6. Click **Add File** and select <*path*>\...**\examples\cycloneIII_3C120_dev_pfl\cycloneIII_3C120_dev_pfl.sof**.
7. Turn on the **Program/Configure** option for the added file.
8. Click **Start** to download the selected configuration file to the FPGA (Figure A–2). The FPGA is configured when the progress bar reaches 100%, after which it is ready to access and program the flash device.

Figure A–2. PFL Programming



9. Click **Auto Detect.** The EP3C120 device and a child CFI_512MB device appear in the list of devices to be programmed.
10. Double-click the <*File*><*none*> field of the CFI_512MB row. The **Select New Programming File** dialog box appears. Select the desired **.pof**, in this example the <*output_file*>**.pof** flash file you created earlier, and click **Open**.
11. Turn on **Page_0** and **OPTION_BITS** options in the **Program/Configure** column that correspond to the CFI_512MB device (Figure A–3). This results in writes only to the flash page zero and the option bit register.

Figure A–3. Program/Configure Options



12. Click **Start**. The message window details the flash writing progress to successful completion. Flash writing to one page, as in this case, can take five to six minutes.

You have now successfully programmed the flash device with a configuration for your board. To configure the board from the flash device, power cycle the board as described in "Powering Up the Board" on page 4–1.

Powering on the board causes the flash device to load a new configuration into the FPGA device. The Configuration Done LED lights up and the hardware functions associated with the design take effect.

Restoring the Factory Design to the Flash Device

To restore the development board to factory conditions, repeat the steps for writing a new POF to the flash device as described in "Programming the Flash Device" on page A–4, except select the **cycloneIII_3c120_dev_factory_recovery.pof** file.

Revision History

The following table displays the revision history for this user guide.

Date	Version	Changes Made
March 2009	1.2	Updated Table 4–2.
August 2008	1.1	■ Updated "Introduction" section. ■ Updated directory structure and information about the directory contents in Figure 3–1 and Table 3–1, respectively. ■ Removed Figure 3-2 and Figure 4-2. ■ Corrected SW3.4 information in Table 4–2. ■ Updated "Power Design Example" section with new compilation numbers. ■ Updated "Measuring Power" section. ■ Converted document to new frame template and made textual and style changes.
October 2007	1.0	First publication

How to Contact Altera

For the most up-to-date information about Altera products, refer to the following table.

Contact Note 1	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note to Table:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

This document uses the typographic conventions shown in the following table.

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: $\mathbf{f_{MAX}}$, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Design.*
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, $n + 1$. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: *<file name>*, *<project name>***.pof** 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."
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input.` Active-low signals are denoted by suffix `n`, e.g., `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword `SUBDESIGN`), as well as logic function names (e.g., `TRI`) 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 calls attention to a condition that could damage the product or design and should be read prior to starting or continuing with the procedure or process.
	The warning calls attention to a condition that could cause injury to the user and 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.