EXCALIBUR™

Nios Embedded Processor

Getting Started

User Guide
January 2002
Version 2.0

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

A-UG-NIOSGSG-2.0

About this User Guide

This user guide provides comprehensive information about the Excalibur™ Development Kit, featuring the Nios™ embedded processor.

Table 1 shows the user guide revision history.

Go to the following sources for more information:

- See http://www.altera.com for more information about the features of this kit, including new features in this release.
- See page 8 for a list of available Nios documentation.
- See the Nios embedded processor *readme* file for installation and licensing information.

Table 1. User Guide Revision History

Date	Description
August 2000	Initial release
March 2001	1.1 release
January 2002	2.0 release

How to Find Information

- The Adobe Acrobat Find feature allows you to search the contents of a PDF file. Click the binoculars toolbar icon to open the Find dialog box.
- Bookmarks serve as an additional table of contents.
- Thumbnail icons, which provide miniature previews of each page, provide a link to the pages.
- Numerous links, shown in green text, allow you to jump to related information.

How to Contact Altera

For the most up-to-date information about Altera® products, go to the Altera world-wide web site at http://www.altera.com.

For technical support on this product, go to http://www.altera.com/mysupport. For additional information about Altera products, consult the sources shown in Table 2.

Table 2. How to Contact Altera

Information Type	USA & Canada	All Other Locations
Product literature	http://www.altera.com	http://www.altera.com
Altera literature services	lit_req@altera.com *(1)*	lit_req@altera.com *(1)*
Non-technical customer service	(800) 767-3753	(408) 544-7000 (7:30 a.m. to 5:30 p.m. Pacific Time)
Technical support	(800) 800-EPLD (3753) (7:30 a.m. to 5:30 p.m. Pacific Time)	(408) 544-7000 *(1)* (7:30 a.m. to 5:30 p.m. Pacific Time)
	http://www.altera.com/mysupport/	http://www.altera.com/mysupport/
FTP site	ftp.altera.com	ftp.altera.com

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

Typographic Conventions

The ***Nios Embedded Processor Getting Started User Guide*** uses the typographic conventions shown in Table 3.

Table 3. Conventions

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: $\mathbf{f_{MAX}}$, **\QuartusII** directory, **d:** drive, **chiptrip.gdf** file.
Bold italic type	Book titles are shown in bold italic type with initial capital letters. Example: ***1999 Device Data Book***.
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.

Table 3. Conventions

Visual Cue	Meaning
"Subheading Title"	References to sections within a document and titles of Quartus II Help topics are shown in quotation marks. Example: "Configuring a FLEX 10K or FLEX 8000 Device with the BitBlaster™ Download Cable."
`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:\quartusII\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.,...	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 angled arrow indicates you should press the Enter key.
👣	The feet direct you to more information on a particular topic.

Notes:

Nios Embedded Processor Getting Started

Introduction

The Nios Development Kit includes software, hardware, accessories, and documentation to create working embedded systems projects. The Nios development board is shipped with a factory default 32-bit reference design. In addition to the factory default design loaded on the board, several other 16-bit and 32-bit reference designs are provided. You can use these reference designs as a starting point for designing your own Nios project with the included Nios peripherals and bus configuration.

The purpose of this getting started user guide is to familiarize you with all the items in the kit. This user guide tells you how to install the Nios embedded processor software and how to set up and test the Nios development board. Once set up, you are ready to begin using the Nios Development Kit to create your own Nios system module. A description of the documents is available to assist you in choosing which documents to use for designing your Nios embedded system.

Before You Begin

Be sure to check the contents in each box of the kit to verify that you received all the items. If any of these items are missing, contact http://www.altera.com/mysupport before you proceed.

Nios Printed Documentation Box

- Nios 16-Bit Programmer's Reference Manual
- Nios 32-Bit Programmer's Reference Manual

Nios Embedded Processor Design Tools Box

- Nios Embedded Processor CD-ROM
- GNUPro Toolkit CD-ROM
- Quartus® II Limited Edition CD-ROM
- LeonardoSpectrum® CD-ROM
- Third-Party Tool Evaluation CD-ROMS

GNUPro Printed Documentation Box

- GNUPro printed documentation

Nios Development Board Box

- APEX™ 20K200E development board
- 9-V DC power supply
- 4 power cables (US, Japan, UK, Europe)
- Serial Y cable
- 6-foot 25-pin parallel port extension cable
- ByteBlasterMV™ cable
- LCD Module

Important Considerations

- You will need the serial number for installing and licensing the Quartus II software. The sticker with the serial number is on the Nios Embedded Processor Design Tools CD holder.
- You will need your network identification card (NIC) number for licensing Quartus II software. To get your NIC number, go to the DOS prompt and enter `ipconfig/all`. The 12-digit hexadecimal number on the line marked "physical address" is your NIC number. When entering the NIC number, exclude the dashes.
- Do not choose a directory name with spaces, such as *Program Files*. Spaces in a directory name may cause the Quartus II software to not work properly.
- You will need administrative privileges to install parts of the Quartus II software on WinNT or Win2000.

CD Software Description

Nios Embedded Processor

The SOPC Builder MegaWizard® is used to create Nios embedded processor systems with either the 16-bit or 32-bit CPU core, built-in peripherals, on- and off -chip ROM and RAM support and bus support for external hardware modules.

To Install the Nios Embedded Processor CD-ROM

The Nios embedded processor CD-ROM is found in the Nios Embedded Processor Design Tools Box. When you insert the Nios embedded processor CD-ROM in your CD-ROM drive, the install program begins. Follow the install directions.

Quartus II Limited Edition Software

The Quartus II Limited Edition software is a programmable logic development tool for programming Altera devices. The Quartus II tool supports system-level solutions with block-level editing, workgroup computing, and expanded support for megafunctions. In addition, an embedded logic analysis feature allows engineers to verify chip functionality and timing by observing internal and I/O signal values at system clock speeds.

Follow the instructions in the *readme* for licensing your Quartus II software.

LeonardoSpectrum

LeonardoSpectrum is bundled with the Nios embedded processor and installs automatically when you install the Nios embedded processor CD-ROM. This version of LeonardoSpectrum is required for Nios system development. This version of LeonardoSpectrum does not conflict with any previously installed versions of the Nios embedded processor.

Third-Party Evaluation CD-ROMS

The third-party development tools included in the Nios development kit support the Nios embedded processor. Refer to the *readme* for more information about these third-party evaluation CD-ROMS.

The Nios Development Board

Use the Nios development board to begin developing embedded systems running on an Altera programmable device. A 32-bit Nios CPU running a boot monitor is pre-loaded into Flash memory and boots on power-up. The monitor provides the interface between the CPU, memory and your developed code. Custom hardware configuration and software applications can be loaded into Flash memory to create a stand-alone embedded system.

Refer to the *Nios Embedded Processor Development Board* manual and board schematics for more information by selecting **Start > Programs > Altera > Excalibur Nios 2.0 > Altera Excalibur Nios Documentation** from the Windows Start Menu.

Setting Up the Development Board

To set up the Nios development board, follow these steps:

1. Take the Nios development board out of the box and remove its anti-static shipping bag. Take care not to expose the board to electrostatic discharge (ESD) during setup or use.

2. The Y serial cable has three labeled connectors; **board**, **main** and **debug**. Connect the **board** connector to the serial port connector (J3) on the Nios development board and the **main** connector to either the COM1 or COM2 serial port on your personal computer.

The **debug** connector is optionally available for dedicated debug communications.

3. Connect one end of the 25-pin parallel cable to the parallel port of your personal computer and the other end to the ByteBlasterMV download cable.

For more information about the ByteBlasterMV download cable see the *ByteBlasterMV Parallel Port Download Cable* Data Sheet on the website at http://www.altera.com/literature/lit-index.html.

4. Connect the ByteBlasterMV download cable to the JTAG connector (JP3).

5. Connect the 14-pin LCD module ribbon cable to JP12. Pin 1 on JP12 must connect to pin 1 on the LCD module. Pin 1 is marked with a small triangle arrow molded into the plastic of the connector. You will use the LCD module later for testing the system.

Pin 3 on the LCD module has been deliberately removed. This pin controls the contrast of the LCD module.

6. Connect the 9-V DC power-supply to J1. Select the appropriate power cord for your location. Connect one end to the 9-V DC power supply and the other end to an outlet.

You have finished connecting the board. LED 7, LED 3 and the dual 7-segment LED display lights and one row of the LCD module displays black squares.

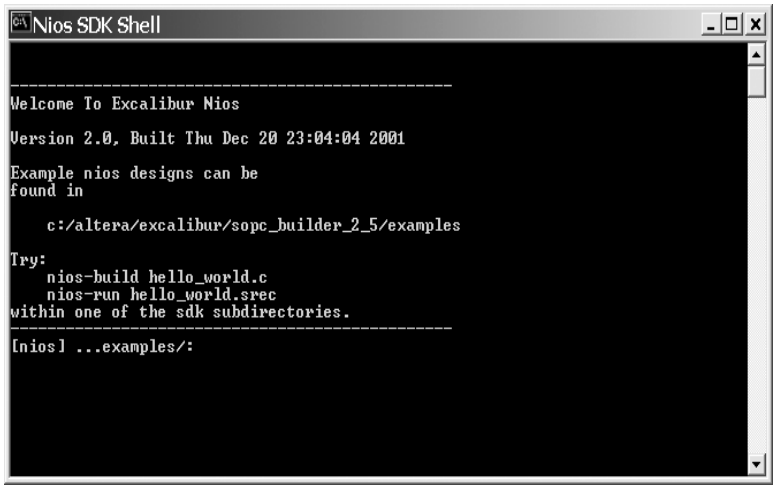
If the Nios development board LEDs are not lit, check all the connections to make sure power is supplied to the board properly. For further assistance, call (800) 800-EPLD (3753).

Setting Up Communication with the Board

The development board is now working with the factory default settings. Nios SDK shell is the terminal communication program you will use. To establish interaction and communication with the development board, do the following:

1. From the Windows Start Menu, select **Altera> Excalibur Nios 2.0> Nios SDK Shell.** The shell window appears and displays a shell prompt, as shown in Figure 1.

Figure 1. Nios SDK Shell Prompt



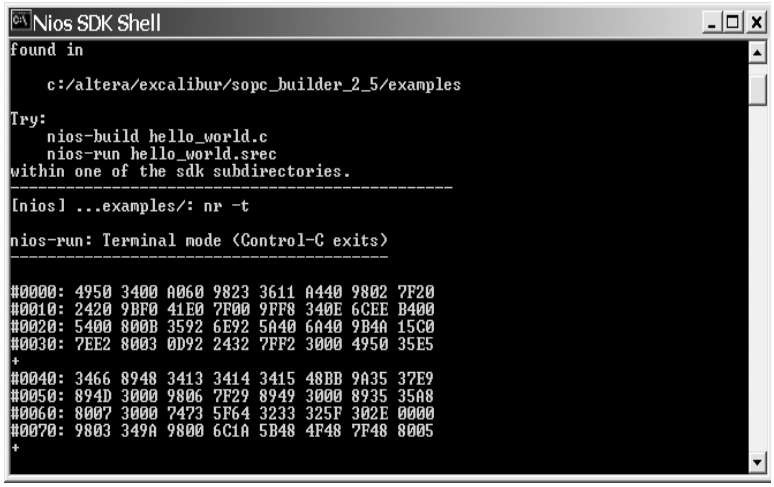
The terminal communications program for Nios 2.0 is Nios SDK Shell. Selecting **Program> Cygwin> bash** from the Windows Start Menu accesses the previous program used for terminal communication. Do not use this method for Nios 2.0.

2. Type `nios-run -t ↵` to begin communicating with the development board.

Step 2 assumes the serial cable is connected to COM1. If you are using COM2, type `nios-run -p com2 -t ↵`

3. Press Enter several times. The GERMS monitor (code that executes and controls the boot process) running on the Nios CPU, displays the first few lines of memory starting at address 0000, as shown in Figure 2. You have established communications and are now ready to download application software to the development board.

Figure 2. GERMS Monitor



See the ***Software Development Reference Manual*** for a table of GERMS monitor commands and usage examples.

4. Press Ctrl+C to exit the terminal communication program and return to the Nios SDK Shell prompt.

Testing the System

You will now compile and download the lcd demo to the Nios embedded processor system. Change directories to the standard 32-bit example design by typing the following command:

```
cd vhdl/standard_32/cpu_sdk/src
```

This command places you in the default software development directory for the vhdl version of the example design. This design is pre-programmed into the development board Flash memory.

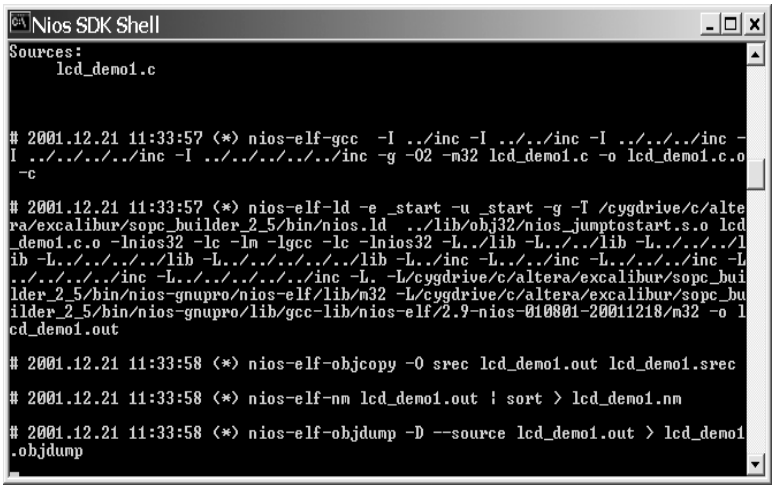
If you prefer working with verilog designs, type the following command:

```
cd verilog/standard_32/cpu_sdk/src
```

To compile and download the lcd demo, do the following:

1. Type `nios-build lcd_demo1.c` ↵ at the Nios SDK Shell prompt. The nios-build utility will invoke the compiler and linker that produces several intermediate files and an executable (.srec) file as shown in Figure 3.

Figure 3. Nios-Build Message



2. Type `nios-run lcd_demo1.srec` ↵ to download the srec file over the serial port and begin execution. You will see a message appear on the LCD screen. Press any of the push-button switches (SW4–SW7) and the message is replaced with a timer-count. Your Nios system is now working.

You are now ready to start designing your own Nios system. Select **Start > Programs > Altera > Excalibur Nios 2.0 > Altera Excalibur Nios Documentation** from the Windows Start Menu to access the *Nios Tutorial.* Use the *Nios Tutorial* to learn how to build a Nios system with the SOPC Builder MegaWizard.

Nios Documentation

Below is a list and description of the documents available with the Nios Development Kit. Use Table 1 as a reference for deciding which document are appropriate. All the documents listed can be found in either PDF or text format. These documents are available by selecting **Start > Programs > Altera>Excalibur Nios 2.0 > Altera Excalibur Nios Documentation** from the Windows Start Menu.

Table 1. Nios Documentation

Document	Hardware Developer	Software Developer	IP Developer
Nios Embedded Processor Getting Started Guide	x	x	x
Nios Embedded Processor Development Board	x	x	x
Nios 2.0 Tutorial	x	x	x
Nios 16-Bit Programmer's Reference Manual		x	
Nios 32-Bit Programmer's Reference Manual		x	
Nios Embedded Processor Software Reference Manual		x	
Nios Embedded Processor Peripherals Reference Manual	x	x	
Avalon Bus Specification Data Sheet	x		x
Custom Instruction Application Note	x	x	
Nios Embedded Processor SOPC Builder Reference Manual	x		x
Nios Ethernet Development Kit User Guide	x	x	
Simultaneous Multi-Mastering Application Note	x		x
Simulating Nios Embedded Processor Designs Application Note	x		x
GNUPro Printed documentation	x	x	

Nios Development Board Reference Manual

This reference manual (*nios_development_board.pdf*) describes the features and functionality of the Nios development board and provides the technical details of its components. For Nios board schematics, select **Start > Programs > Altera > Excalibur Nios 2.0 > Altera Excalibur Nios Documentation** from the Windows Start Menu.

Nios Software Development Reference Manual

This reference manual (*nios_software_development_reference.pdf*) provides information for programmers developing software for the Nios embedded processor. The primary focus of this document is for developers writing code in C and includes sections addressing code written in assembly.

Nios 16-Bit Programmer's Reference Manual

This reference manual (*nios_programmers_reference_16.pdf*) contains details on the Nios processor architecture including registers, addressing modes, program-flow control, exceptions and pipeline implementation. It includes complete documentation on the Nios assembly language instruction set for a 16-bit CPU.

Nios 32-Bit Programmer's Reference Manual

This reference manual (*nios_programmers_reference_32.pdf*)contains details on the Nios processor architecture including registers, addressing modes, program-flow control, exceptions and pipeline implementation. It includes complete documentation on the Nios assembly language instruction set for a 32-bit CPU.

Nios Peripherals Reference Manual

This reference manual (*nios_peripherals_reference_manual.pdf*) describes the peripherals included in the Nios development kit. It includes details about peripheral registers, peripheral template file (PTF) assignments and software routines.

SOPC Builder Reference Manual

This reference manual (*sopc_reference_manual.pdf*) provides an explanation of the process for constructing a Nios system, an overview of the SOPC Builder MegaWizard, an explanation of the PTF structure and a PTF dictionary.

Nios Ethernet Development Kit User Guide

This user guide (*nios_edk_user_guide.pdf*) describes installing the kit's development tools and hardware and how to access the software application examples. It includes a functional description of the daughter card and PC-board design files and the supported protocol C-language library with a plugs library reference for the software routines. The Nios Ethernet Development Kit can be purchased as an optional add-on kit.

Avalon Bus Specifications Data Sheet

This data sheet (*avalon_bus_spec.pdf*) is for developers creating custom peripherals for the Avalon™ bus. It defines terms and concepts of SOPC designs based on the Avalon bus architecture used for connecting on-chip processors and peripherals into a system on a programmable chip (SOPC). Avalon bus signal functions and timing are defined.

Nios Tutorial

This tutorial (*nios_tutorial_2_0.pdf*) shows the developer how to use the SOPC Builder MegaWizard to build a Nios processor-based system and custom software development kit (SDK).

Simultaneous Multi-Mastering Avalon Bus Application Note

This application note (*nios_an184_avalon_multimaster.pdf*) describes the simultaneous multi-master Avalon bus with an explanation about how it differs from existing bus arbitration schemes. It includes an in-depth explanation of bus arbitration priorities and most commonly used configurations for your Nios embedded system design.

Custom Instruction Application Note

This application note (*nios_an188_custom_instructions.pdf*) describes the custom instruction feature and how to implement custom instructions in Nios based-designs. It includes a design example that shows the benefits of using custom instructions.

Simulating Nios Embedded Processor Designs

This application note (*nios_an189_simulating_designs.pdf*) describes the simulation flow of a simulation project and walks the user through the process of simulating a design using ModelSim® created by the SOPC Builder.

Nios FAQ

The FAQ covers both hardware and software design issues.

Readme File

This file explains the installation and operating requirements for Windows, Solaris and HPUX. It also contains instructions for software licensing.

For More Information

For updates and newly created Nios documentation, go to the Excalibur section of the Altera web site at http://www.altera.com. If you need technical support, contact the Altera Applications department at http://www.altera.com/mysupport.