



EXCALIBUR™

Nios Embedded Processor

Getting Started

**Version 1.1
User Guide
March 2001**



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

A-UG-NIOSKIT-01
P25-05899-01

Altera, ACEX, APEX, APEX 20K, FLEX, FLEX 10KE, MAX+PLUS II, MegaCore, MegaWizard, OpenCore, and Quartus are trademarks and/or service marks of Altera Corporation in the United States and other countries. Altera Corporation acknowledges the trademarks of other organizations for their respective products or services mentioned in this document, including the following: Verilog is a registered trademark of Cadence Design Systems, Incorporated. Java is a trademark of Sun Microsystems Inc. ModelSim is a trademark of Model Technologies. MATLAB is a registered trademark of the MathWorks. Microsoft is a registered trademark and Windows is a trademark of Microsoft Corporation. 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.



I.S. EN ISO 9001

Copyright © 2001 Altera Corporation. All rights reserved.



About this User Guide

March 2001, ver. 1.1

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

Table 1 shows the Nios Getting Started User Guide revision history.

Table 1. Revision History		
Revision	Date	Description
Version 1.0	Aug. 2000	Nios Embedded Processor Quick Start Guide - printed
Version 1.1	Mar. 2001	Nios Embedded Processor Getting Started User Guide - printed

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 additional information about Altera products, consult the sources shown in Table 2.

Information Type	Access	USA & Canada	All Other Locations
Altera literature services	Electronic mail	lit_req@altera.com	lit_req@altera.com
Non-technical customer service	Telephone hotline	(800) SOS-EPLD	(408) 544-7000 (7:30 a.m. to 5:30 p.m. Pacific Time)
	Fax	(408) 544-7606	(408) 544-7606
Technical support	Telephone hotline	(800) 800-EPLD (6:00 a.m. to 6:00 p.m. Pacific Time)	(408) 544-7000 (7:30 a.m. to 5:30 p.m. Pacific Time)
	Fax	(408) 544-6401	(408) 544-6401
	Electronic mail	telecom@altera.com	telecom@altera.com
	FTP site	ftp.altera.com	ftp.altera.com
General product information	Telephone	(408) 544-7104	(408) 544-7104
	World-wide web site	http://www.altera.com	http://www.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.

<i>Table 3. Conventions</i>	
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} , \maxplus2 directory, d: drive, chiptrip.gdf file.
<i>Bold italic type</i>	Book titles are shown in bold italic type with initial capital letters. Example: <i>1999 Device Data Book</i> .
<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 the Quartus software and the MAX+PLUS II software 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., reset_n. Anything that must be typed exactly as it appears is shown in Courier type. For example: c:\max2work\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

March 2001, ver. 1.1

User Guide

Introduction

The Excalibur Development Kit, featuring the Nios embedded processor, 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. Using the software and hardware included in the kit, you can create a complete project and specify your own Nios peripherals and bus configuration. In addition to the factory default design loaded on the board, there are several other 16-bit and 32-bit reference designs provided. You can use these reference designs as a starting point or create your own.

The purpose of this user guide is to familiarize you with all the items in the kit, guide you through the process of installing the Nios embedded processor software, and show you how to set up and test the Nios development board. Once set up, you are ready to begin using the Excalibur Development Kit featuring the Nios embedded processor, to tailor your own Nios system module.

Software Descriptions

Quartus II Software

The Quartus™ II 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.

Nios Embedded Processor (System Builder MegaWizard)

The Nios System Builder MegaWizard runs from the Quartus II's MegaWizard Plug-In Manager™. The MegaWizard Plug-In Manager is used to develop Nios embedded processor systems, using a 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.

The Nios Development Board

Before You Begin

GNUPro Software Development Tool

The GNUPro Toolkit is an industry-standard open-source software development toolkit optimized for the Nios embedded processor. The toolkit includes a C/C++ compiler, macro-assembler, linker, debugger and numerous binary utilities, and libraries.

The Nios development board is a fully-functional embedded systems hardware development platform. With it, you can immediately start developing embedded systems running on an Altera programmable device. A 32-bit Nios CPU running a boot monitor is pre-loaded in 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.

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 support@altera.com before you proceed.

Nios Embedded Processor Box

- Nios Embedded Processor Programmer's Reference Manual
- Nios Embedded Processor CD-ROM (Nios HDK)

Quartus II Programmable Logic Development Tools Software Box

- Quartus II software for PC CD-ROM
- Internet Explorer CD-ROM
- Quartus II Installation & Licensing for PCs Manual
- Quartus II Tutorial Manual

Nios Development Board Box

- Nios Development Board
- Nios Development Board Reference Manual
- LCD Module Kit
- 9-V DC Power Supply
- 4 Power Cables (US, Japan, UK, Europe)
- 6-foot 9-Pin Serial Cable
- 9-Pin to 25-Pin serial cable adapter
- 6-foot 25-Pin Parallel port extension cable
- ByteBlasterMV Cable
- ByteBlasterMV Data Sheet

Important Considerations

GNUPro Software Development Tools Box

- GNUPro Toolkit Manual
- GNUPro Quick Start Guide
- GNUPro Software Development Tools CD-ROM (Nios SDK)
- You must install BOTH the Nios Embedded Processor CD-ROM and the GNUPro Software Development Tools CD-ROM before you begin developing Nios processor-based designs.
- You will need Internet Explorer 4.x or 5.x to register the Quartus II software properly. If needed, use the enclosed Internet Explorer CD-ROM found in the Quartus II software box.
- You will need the serial number for installing and licensing the Quartus II software. Keep your Quartus II CD-ROM cover or Quartus II software box close for entering serial number information.
- 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 not to work properly.
- You will need administrative privileges to install parts of the Quartus II software on WinNT.

Software Installation

Altera recommends that you use the factory default directories when installing the Excalibur Development Kit, featuring the Nios embedded processor software.

Installing the Quartus II Software

The Quartus II software CD-ROM is found in the Quartus II Programmable Logic Development Tools box. Insert the Quartus II CD-ROM into your CD-ROM drive and the install program begins. Follow the install directions. After installing the Quartus II software, follow the instructions in the enclosed letter for licensing your Quartus II software.



All additional information for installing and licensing Quartus II is found in the first section of the *Quartus Installation & Licensing for PCs* manual.

Installing Nios Hardware Development Kit (HDK)

The Nios HDK is for hardware developers creating a Nios processor-based system. The Nios embedded processor CD-ROM is found in the Nios Embedded Processor box. Insert the Nios HDK in your CD-ROM drive and the install program begins. Follow the install directions.



Unix users: Refer to the Readme file for installation instructions.

Installing Cygnus GNUPro and Software Development Kit (SDK)

The SDK is for software engineers who want to compile C/C++ programs for Nios using the Cygnus GNUPro toolkit. The GNUPro software development tools CD-ROM is found in the GNUPro Software Development Tools box. Insert the GNUPro CD-ROM in your CD-ROM drive and the install program begins. Follow the install directions.

Cygwin is required for the GNUPro tools to run. The Nios SDK will check for Cygwin during installation. If Cygwin is not found, the system will install Cygwin first before installing the rest of the GNUPro toolkit. Install Cygwin in the C:\Cygwin or D:\Cygwin directory. After the SDK is installed, we recommend you restart your computer.



In the Cygwin install you have a choice of whether to install the source-code for Cygwin or not. Unless you are going to rebuild Cygwin, you do not need to install the source-code at this time.



Unix users: Refer to the Readme file for installation instructions.

The following table describes which top-level directories the Quartus II software, Nios HDK, GNUPro and Nios SDK CD-ROMs are written to during installation.

CD-ROM	Directory	Description
Quartus II software	Qdesigns	Quartus II example designs
	Quartus II software	Program Files
Nios HDK	Quartus II software	Program Files
	Altera	Reference design and documentation
GNUPro & Nios SDK	Cygwin	Compiler and utilities
	Altera	Libraries, documentation and example code

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. Connect one end of the 9-pin serial cable to the serial connector (J3) and connect the other end to either the COM1 or COM2 serial port on your personal computer.



The default setting is COM1 for the nios-run utility. If you need to use a 25-pin serial port instead of a 9-pin version, use the enclosed 25-pin to 9-pin adapter.

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 ByteBlaster download cable see the *ByteBlaster MV Parallel Port Download Cable Data Sheet* or visit the website at www.altera.com.

1. Connect the ByteBlasterMV download cable to the JTAG port (JP3).
2. 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 insulation displacement connector (IDC) 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.

3. 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.

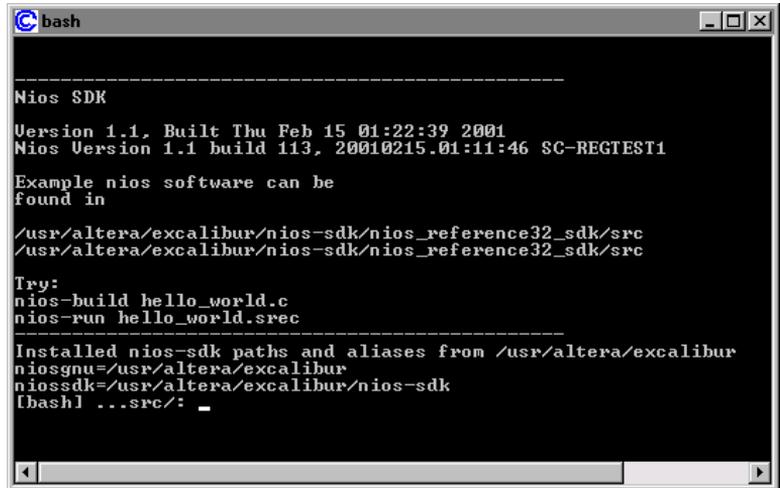


If the Nios development board LEDs are not lit, check all the connections and make sure the power is getting to the board. For further assistance, call (800) 800-EPLD.

Setting Up Communication with the Board

The development board is now working with the factory default settings. You will use bash as the terminal communication program. To establish interaction and communication with the development board, do the following:

1. From the Windows Start Menu, select **Program > Cygwin > bash**. The bash window appears and displays the bash prompt, as shown in the following illustration:



```

bash
-----
Nios SDK
Version 1.1, Built Thu Feb 15 01:22:39 2001
Nios Version 1.1 build 113, 20010215.01:11:46 SC-REGTEST1

Example nios software can be
found in

/usr/altera/excalibur/nios-sdk/nios_reference32_sdk/src
/usr/altera/excalibur/nios-sdk/nios_reference32_sdk/src

Try:
nios-build hello_world.c
nios-run hello_world.srec
-----
Installed nios-sdk paths and aliases from /usr/altera/excalibur
niosgnu=/usr/altera/excalibur
niossdk=/usr/altera/excalibur/nios-sdk
[bash] ...src/; _
  
```



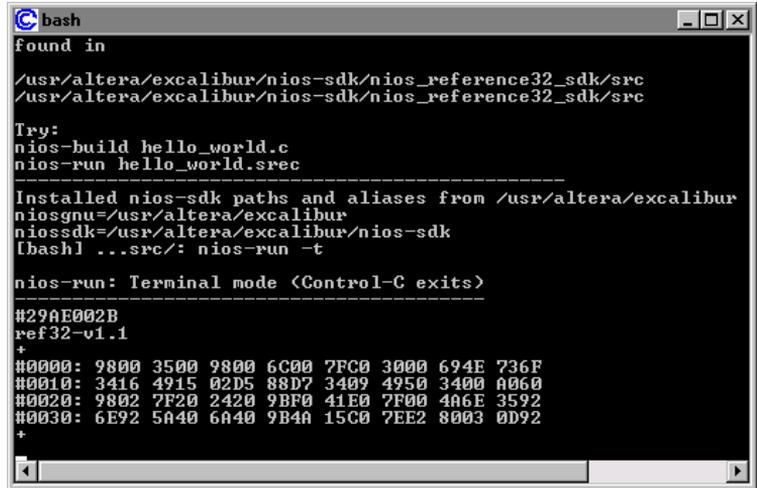
If you are not in the correct directory, do the following: At the [bash] : prompt, type `mount -f d:\\cygwin /`

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



If you are using COM2, type `nios-run -p com2 -t ↵`

- 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 the following illustration:



```

bash
found in
/usr/altera/excalibur/nios-sdk/nios_reference32_sdk/src
/usr/altera/excalibur/nios-sdk/nios_reference32_sdk/src

Try:
nios-build hello_world.c
nios-run hello_world.srec
-----
Installed nios-sdk paths and aliases from /usr/altera/excalibur
niosgnu=/usr/altera/excalibur
niossdk=/usr/altera/excalibur/nios-sdk
[bash] ...src/: nios-run -t
-----
nios-run: Terminal mode <Control-C exits>
-----
#29AE002B
ref32-v1.1
+
#0000: 9800 3500 9800 6C00 7FC0 3000 694E 736F
#0010: 3416 4915 02D5 88D7 3409 4950 3400 A060
#0020: 9802 7F20 2420 9BF0 41E0 7F00 4A6E 3592
#0030: 6E92 5A40 6A40 9B4A 15C0 7EE2 8003 0D92
+

```

- You have established communications and are now ready to download application software to the development board.
- Press Ctrl+C to exit the terminal communication program and return to the [bash]: prompt.

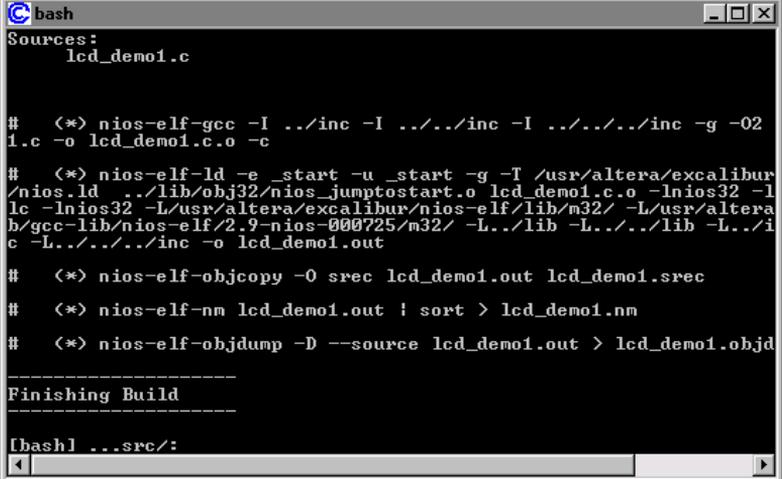
Testing the System

You will now compile and download the lcd demo to the Nios embedded processor system. When you open the bash shell, the system places you in the following directory:

```
\usr\altera\excalibur\nios-sdk\nios-reference32-
sdk\src
```

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

1. Type `nios-build lcd_demo1.c` ↵ at the `[bash]:` prompt. The `nios-build` utility will invoke the compiler and linker that produces several intermediate files and an executable (.srec) file. See the following illustration:



```
bash
Sources:
  lcd_demo1.c

# (*) nios-elf-gcc -I ../inc -I ../../inc -I ../../../inc -g -O2
i.c -o lcd_demo1.c.o -c

# (*) nios-elf-ld -e _start -u _start -g -T /usr/altera/excalibur
/nios.ld ../lib/obj32/nios_jumptostart.o lcd_demo1.c.o -lnios32 -l
lc -lnios32 -L/usr/altera/excalibur/nios-elf/lib/m32/ -L/usr/altera
b/gcc-lib/nios-elf/2.9-nios-000725/m32/ -L../lib -L../lib -L../i
c -L../../inc -o lcd_demo1.out

# (*) nios-elf-objcopy -O srec lcd_demo1.out lcd_demo1.srec

# (*) nios-elf-nm lcd_demo1.out | sort > lcd_demo1.nm

# (*) nios-elf-objdump -D --source lcd_demo1.out > lcd_demo1.objd

-----
Finishing Build
-----

[bash] ...src/:
```

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 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. Use the *Nios Embedded Processor System Builder Tutorial* found in the `nios_tutorial` folder (`\Altera\Excalibur\nios_tutorial`) to learn how to build a Nios system with the Nios System Builder MegaWizard Plug-In Manager.

Documentation

Below is a brief description of the documents available to you with the Excalibur Development Kit, featuring the Nios embedded processor. A copy of the Nios Programmer's Reference Manual is found in the kit. All the documents listed below can be found in either PDF or text format in the *nios_documentation folder* (\Altera\Excalibur\nios_documentation).

Nios Embedded Processor System Builder Tutorial

This tutorial found in the *nios_tutorial* folder, shows you how to use the MegaWizard Plug-In Manager to build a Nios processor-based system and custom software development kit.

Nios Development Board Reference Manual

This reference manual describes the features and functionality of the development board and gives detailed descriptions of the all the components you will be using.

Nios Software Development Reference Manual

This reference manual provides a single source of information to address the needs of the embedded systems programmer creating code for the Nios soft core processor. This document focuses on development using C programming language with several sections dedicated to the use of assembly code for application specific requirements.

Nios Programmer's Reference Manual

This reference manual contains details on the Nios processor architecture including registers, addressing modes, program-flow control, exceptions and pipeline implementation. This manual also includes complete documentation on the Nios assembly language instruction set.

Data Sheets for the following Nios Peripherals:

- Nios Embedded Processor Timer Peripheral
- Nios Embedded Processor UART Peripheral
- Nios Embedded Processor P I/O Module Peripheral
- Nios Embedded Processor SPI Peripheral

Each of these data sheets gives detailed information to assist you in using these peripherals for your own project design.

Nios FAQs

FAQs covering both hardware and software design issues.

Avalon Bus Specifications

Avalon is a simple bus for building on-chip microprocessor systems using the Nios embedded processor. This document specifies the characteristics of this bus, which connects the Nios CPU to peripherals, internal and external memory devices.

Readme File

This file explains the installation and operating requirements for Windows, the Solaris Workstations and the HP Workstation for the Excalibur Development Kit, featuring the Nios embedded processor.

For More Information

For the most recent documentation, go to the Excalibur section of the Altera web site at <http://www.altera.com>. There are several PDF documents and schematics that give more detailed information about the Excalibur Nios development board. If you need technical support, contact the Altera Applications department at (800) 800-EPLD.