Nios® II

Nios Development Board

Reference Manual, Cyclone Edition

ALTERA®

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

Printed on recycled paper

MNL-N2DEVLBDCYC-1.2

About this Manual

This manual provides component details about the Nios® development board, Cyclone Edition.

Table 1–1 shows the reference manual revision history.

Table 1–1. Reference Manual Revision History

Date	Description
December 2004	Corrected PROTO1 pinout information listed in Figures 1-5, 1-6, and 1-7. These figures were incorrect in the September 2004 version of this document (MNL-N2DEVLBDCYC-1.1). The document part number is found on page ii.
September 2004	Updates for Nios II 1.01 release.
May 2004	Updated Appendix B: Restore the Factory Configuration.
January 2004	Pin table corrections.
July 2003	Reflects new directory structure for SOPC Builder 3.0 and Nios Development Kit version 3.1.
May 2003	Minor revisions and edits.
March 2003	First publication of a reference manual. This manual is Cyclone-device specific.

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 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/	altera.com/mysupport/
	(800) 800-EPLD (3753) (7:00 a.m. to 5:00 p.m. Pacific Time)	(408) 544-7000 *(1)* (7:00 a.m. to 5:00 p.m. Pacific Time)
Product literature	www.altera.com	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)
FTP site	ftp.altera.com	ftp.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 below.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, check box 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.”

Visual Cue	Meaning
`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.
⚠ CAUTION	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.

Contents

About this Manual iii

How to Find Information iii

How to Contact Altera iii

Typographic Conventions iv

Board Components 1

Features 1–1

General Description 1–1

Block Diagram 1–2

Default Reference Design 1–2

Restoring the Default Reference Design to the Board 1–3

Nios Development Board Components 1–3

The Cyclone EP1C20 Device 1–4

Flash Memory Device 1–5

CompactFlash Connector 1–6

SDRAM Device 1–8

Dual SRAM Devices 1–10

Ethernet MAC / PHY 1–11

Expansion Prototype Connector (PROTO1) 1–11

Expansion Prototype Connector (PROTO2) 1–13

Mictor Connector 1–15

Serial Port Connectors 1–17

Dual 7-Segment Display 1–18

Push-Button Switches 1–19

Individual LEDs 1–19

Serial Flash Connector 1–19

Serial Configuration Device (EPCS4) 1–20

Configuration Controller Device (EPM7128AE) 1–20

Reset Distribution 1–21

Starting Configuration 1–21

Cyclone Configuration 1–21

Configuration Data 1–22

Safe & User Configurations 1–22

The Configuration-Status LEDs 1–25

Configuration & Reset Buttons 1–26

Power-Supply Circuitry 1–28

Clock Circuitry 1–28

JTAG Connectors 1–29

JTAG Connector to Cyclone Device (J24) 1–29

JTAG Connector to EPM7128AE Device (J5) 1–30

Appendix A. Shared Bus Table

Description A–1

Appendix B. Restoring the Factory Configuration

Introduction B–1
Reprogramming the Flash Memory B–1
Reprogramming the EPM7128AE Configuration Controller Device B–2

Appendix C. Connecting to the Board via Ethernet

Introduction C–1
Connecting the Ethernet Cable C–1
Connecting the LCD Screen C–2
Obtaining an IP Address C–2
LAN Connection C–2
Point–to–Point Connections C–4
Browsing Your Board C–5

Index

Board Components

Features

- A Cyclone™ EP1C20F400C7device
- 8 Mbytes of flash memory
- 1 Mbyte of static RAM
- 16 Mbytes of SDRAM
- On-board logic for configuring the Cyclone device from flash memory
- EPCS4 serial configuration device
- On-board Ethernet MAC/PHY device
- Two 5-V-tolerant expansion/prototype headers each with access to 41 Cyclone user I/O pins
- CompactFlash™ connector header for Type I CompactFlash (CF) cards
- Mictor connector for hardware and software debug
- Two RS-232 DB9 serial ports
- Four push-button switches connected to Cyclone user I/O pins
- Eight LEDs connected to Straix user I/O pins
- Dual 7-segment LED display
- JTAG connectors to Altera devices via Altera download cables
- 50 MHz Oscillator and zero-skew clock distribution circuitry
- Power-on reset circuitry

General Description

The Nios development board, Cyclone Edition, provides a hardware platform for developing embedded systems based on Altera Cyclone devices. The Nios development board features a Cyclone EP1C20F400C7 device with 20,060 logic elements (LEs) and 294, 912 bits of on-chip memory.

The Nios development board comes pre-programmed with a Nios II processor reference design. Hardware designers can use the reference design as an example of how to use the features of the Nios development board. Software designers can use the pre-programmed Nios II processor design on the board to begin prototyping software immediately.

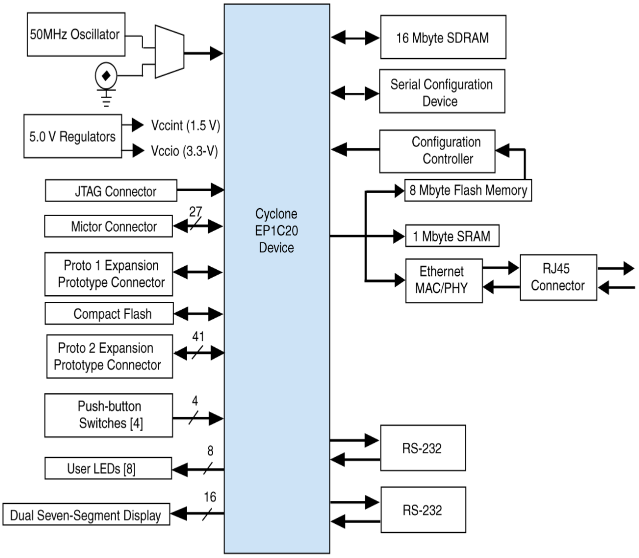
This document describes the hardware features of the Nios development board, including detailed pin-out information, to enable designers to create custom FPGA designs that interface with all components on the board.

See the *Nios II Development Kit, Getting Started User Guide* for instructions on setting up the Nios development board and installing Nios II development tools.

Block Diagram

Figure 1–1 shows a block diagram of the board.

Figure 1–1. Nios Development Board, Cyclone Edition Block Diagram



Default Reference Design

When power is applied to the board, the on-board logic configures the Cyclone FPGA using hardware configuration data stored in flash memory. When the device is configured, the Nios II processor design in the FPGA wakes up and begins executing boot code from flash memory.

The board is factory-programmed with a default reference design. This reference design is a web server that delivers web pages via the Ethernet port. For further information on the default reference design, see Appendix C, Connecting to the Board via Ethernet.

Restoring the Default Reference Design to the Board

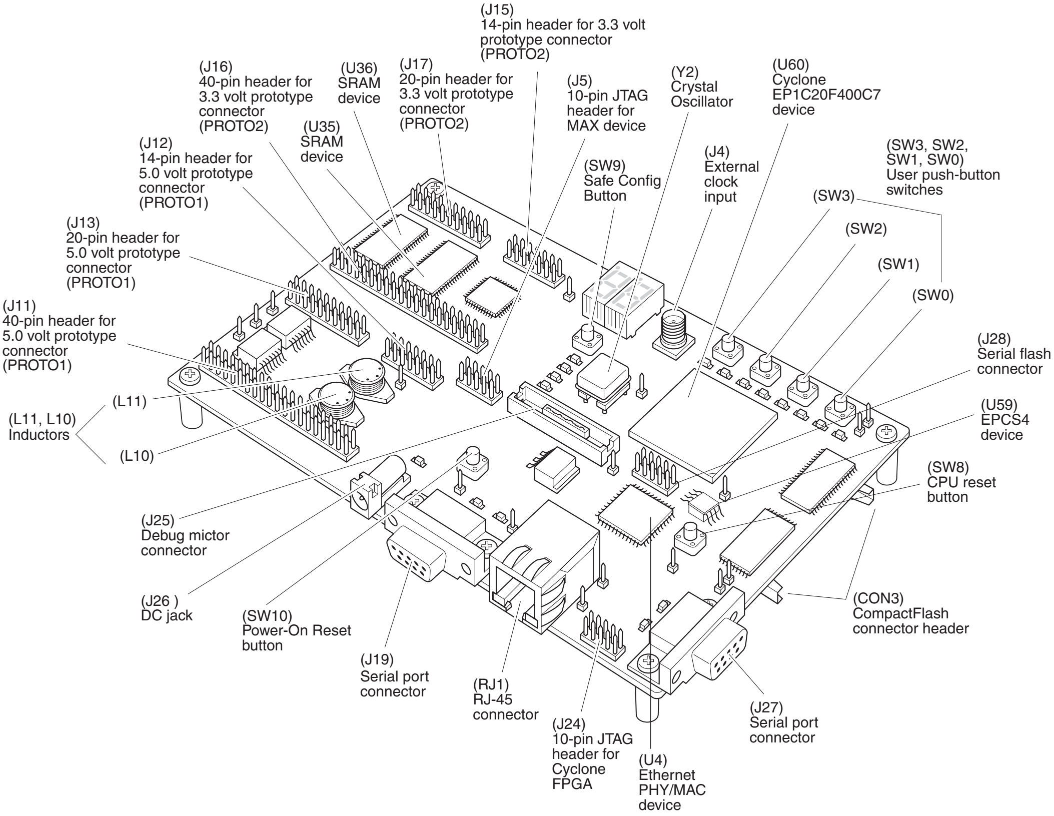
In the course of development, you may overwrite or erase the flash memory space containing the default reference design. Altera provides the flash image for the default reference design, so you can return the board to its default state. See "Restoring the Factory Configuration" on page B–1 for more information.

Nios Development Board Components

This section contains a brief overview of important components on the Nios development board (see Figure 1–2). Links to the component manufacturers are included where available.

A complete set of schematics, a physical layout database, and GERBER files for the development board are installed in the **documents** directory for the Nios II development kit.

Figure 1–2. Nios Development Board Components


(J15)
14-pin header for 3.3 volt prototype connector (PROTO2)
(J16)
40-pin header for 3.3 volt prototype connector (PROTO2)
(U36)
SRAM device
(J17)
20-pin header for 3.3 volt prototype connector (PROTO2)
(J5)
10-pin JTAG header for MAX device
(Y2)
Crystal Oscillator
(U60)
Cyclone EP1C20F400C7 device
(J12)
14-pin header for 5.0 volt prototype connector (PROTO1)
(U35)
SRAM device
(SW9)
Safe Config Button
(J4)
External clock input
(SW3, SW2, SW1, SW0)
User push-button switches
(SW3)
(SW2)
(SW1)
(SW0)
(J13)
20-pin header for 5.0 volt prototype connector (PROTO1)
(J11)
40-pin header for 5.0 volt prototype connector (PROTO1)
(J28)
Serial flash connector
(L11)
(L11, L10)
Inductors
(L10)
(U59)
EPCS4 device
(SW8)
CPU reset button
(J25)
Debug mictor connector
(J26)
DC jack
(SW10)
Power-On Reset button
(CON3)
CompactFlash connector header
(J19)
Serial port connector
(RJ1)
RJ-45 connector
(J24)
10-pin JTAG header for Cyclone FPGA
(U4)
Ethernet PHY/MAC device
(J27)
Serial port connector

The Cyclone EP1C20 Device

U60 is a Cyclone EP1C20F400C7 device in a 400-pin FineLine BGA® package. Table 1–1 lists the Cyclone device features.

Table 1–1. Cyclone EP1C20 Device Features	
Logic Elements	20,060
M4K RAM blocks (128 X 36 bits)	64
Total RAM bits	294,912
PLLs	2
Maximum user I/O pins	301

The development board provides two separate methods for configuring the Cyclone device:

1. Using the Quartus II software running on a host computer, a designer configures the device directly via an Altera download cable connected to the Cyclone JTAG header (J24).

2. When power is applied to the board, a configuration controller device (U3) attempts to configure the Cyclone device with hardware configuration data stored in flash memory. For more information on the configuration controller, see "Configuration Controller Device (EPM7128AE)" on page 1–20.

See the Altera Cyclone literature page for Cyclone-related documentation at **www.altera.com/literature/lit-cyc.html** including a Cyclone EP1C20 pinout document.

Flash Memory Device

U5 is an 8 Mbyte AMD AM29LV065D flash memory device connected to the Cyclone device and can be used for two purposes:

3. A Nios II embedded processor implemented on the Cyclone device can use the flash memory as general-purpose readable memory and non-volatile storage.

4. The flash memory can hold Cyclone configuration data that is used by the configuration controller to load the Cyclone device at power-up. See "Configuration Controller Device (EPM7128AE)" on page 1–20 for related information.

Hardware configuration data that implements the Nios II reference design is pre-stored in this flash memory. The pre-loaded Nios II reference design, once loaded, can identify the 8 Mbyte flash memory in its address space, and can program new data (either new Cyclone configuration data, Nios II embedded processor software, or both) into flash memory. The Nios II embedded processor software includes subroutines for writing and erasing this specific type of AMD flash memory.

The flash memory device shares address and data connections with the SRAM chips and the Ethernet MAC / PHY chip. For shared bus information, see"Shared Bus Table" on page A–1.

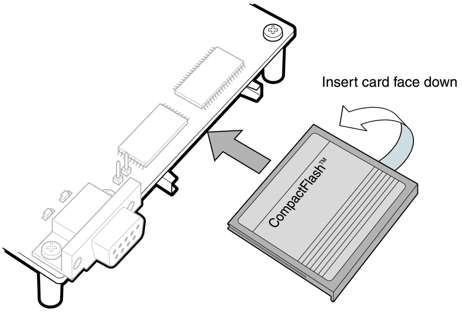
See **www.amd.com** for detailed information about the flash memory device.

CompactFlash Connector

The CompactFlash connector (CON3) enables hardware designs to access a CompactFlash card (see Figure 1–3). The following two access modes are supported:

- ATA (hot swappable mode)
- IDE (IDE hard disk mode)

Figure 1–3. CompactFlash Connector



Most pins of CON3 connect to I/O pins on the FPGA. The following pins have special connections:

- Pin 13 of CON3 (VCC) is driven by a power MOSFET that is controlled by an FPGA I/O pin. This allows the FPGA to control power to the CompactFlash card for the IDE connection mode.
- Pin 26 of CON3 (-CD1) is pulled up to 5V through a 10 Kohm resistor. This signal is used to detect the presence of a CompactFlash card; when the card is not present, the signal is pulled high through the pull-up resistor.
- Pin 41 of CON3 (RESET) is pulled up to 5V through a 10 Kohm resistor, and is controlled by the EPM7128AE configuration controller. The FPGA can cause the configuration controller to assert RESET, but the FPGA does not drive this signal directly.

☞ The CompactFlash connector shares several Cyclone I/O pins with expansion prototype connector PROTO1. See “Expansion Prototype Connector (PROTO1)” on page 1–11 for details on PROTO1.

Table 1–2 on page 1–7 provides CompactFlash pin out details.

Table 1–2. CompactFlash (CON3) Pin Table

Pin on CompactFlash (CON3)	CompactFlash Function	Connects to (1)
1	GND	GND
2	D03	F18
3	D04	E17
4	D05	D17
5	D06	D18
6	D07	C18
7	-CE	H20
8	A10	J15
9	-OE	D13
10	A09	J20
11	A08	H14
12	A07	J14
13	VCC	M13(2)
14	A06	J17
15	A05	J18
16	A04	K15
17	A03	W18
18	A02	H19
19	A01	H18
20	A00	H17
21	D00	F20
22	D01	F15
23	D02	E19
24	WP	H16
25	-CD2	GND(3)
26	-CD1	B13
27	D11	F17
28	D12	E18
29	D13	F16
30	D14	F19
31	D15	G16
32	-CE2	U19

Table 1–2. CompactFlash (CON3) Pin Table

Pin on CompactFlash (CON3)	CompactFlash Function	Connects to [1]
33	-VS1	GND[3]
34	-OIORD	G19
35	-IOWR	G20
36	-WE	V18
37	RDY/BSY	G17
38	VCC	M13[2]
39	-CSEL	GND[3]
40	-VS2	no connect[3]
41	RESET	[4]
42	-WAIT	G14
43	-INPACK	V19
44	-REG	U20
45	BVD2	J16
46	BVD1	J19
47	D081	C19
48	D091	D19
49	D101	D20
50	GND	GND[3]

Note to Table 1–2

[1] All pin numbers represent I/O pins on the FPGA, unless otherwise noted.

[2] This FPGA I/O pin controls a power MOSFET that supplies 5V VCC to CON3.

[3] This pin does not connect to the FPGA directly.

[4] RESET is driven by the EPM7128AE configuration controller device.

For more information on the CompactFlash connector (CON3), see **www.compactflash.org** and **www.molex.com**.

SDRAM Device

The SDRAM device (U57) is a Micron MT48LC4M32B2 chip with PC100 functionality and self refresh mode. The SDRAM is fully synchronous with all signals registered on the positive edge of the system clock.

The SDRAM device pins are connected to the Cyclone device (see Table 1–3 on page 1–9). An SDRAM controller peripheral is included with the Nios II development kit, allowing a Nios II processor to view the SDRAM device as a large, linearly-addressable memory.

Table 1–3. SDRAM (U57) Pin Table (Part 1 of 2)

Pin Name	Pin Number	Connects to Cyclone Pin
A0	25	M2
A1	26	M1
A2	27	M6
A3	60	M4
A4	61	J8
A5	62	J7
A6	63	J6
A7	64	J5
A8	65	J4
A9	66	J3
A10	24	H6
A11	21	H5
BA0	22	H7
BA1	23	H1
DQ0	2	M5
DQ1	4	M3
DQ2	5	M7
DQ3	7	N6
DQ4	8	N1
DQ5	10	N2
DQ6	11	N4
DQ7	13	N3
DQ8	74	N5
DQ9	76	N7
DQ10	77	P7
DQ11	79	P2
DQ12	80	P1
DQ13	82	P6
DQ14	83	P5
DQ15	85	P3

Table 1–3. SDRAM (U57) Pin Table (Part 2 of 2)

Pin Name	Pin Number	Connects to Cyclone Pin
DQ16	31	P4
DQ17	33	R1
DQ18	34	R2
DQ19	36	R6
DQ20	37	R5
DQ21	39	R3
DQ22	40	R4
DQ23	42	T4
DQ24	45	T2
DQ25	47	T3
DQ26	48	U1
DQ27	50	U4
DQ28	51	U2
DQ29	53	U3
DQ30	54	V3
DQ31	56	V2
DQM0	16	J2
DQM1	71	J1
DQM2	28	H4
DQM3	59	H3
RAS_N	19	H2
CAS_N	18	G3
CKE	67	G7
CS_N	20	G6
WE_N	17	G4
CLK	68	L13

See **www.micron.com** for detailed SDRAM information.

Dual SRAM Devices

U35 and U36 are two 512 Kbyte x 16-bit asynchronous SRAM devices. They are connected to the Cyclone device so they can be used by a Nios II embedded processor as general-purpose memory. The two 16-bit devices can be used in parallel to implement a 32-bit wide memory subsystem.

The Nios II factory-programmed reference design identifies these SRAM devices in its address space as a contiguous 1Mbyte, 32-bit-wide, zero-wait-state main memory.

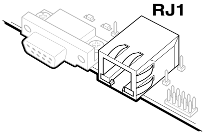
The SRAM devices share address and data connections with the flash memory and the Ethernet MAC / PHY device. For shared bus information, see "Shared Bus Table" on page A–1.

See **www.idt.com** for detailed information about the SRAM devices.

Ethernet MAC/PHY

The LAN91C111 (U4) is a mixed-signal analog / digital device that implements protocols at 10 Mbps and 100 Mbps. The control pins of U4 are connected to the Cyclone device so that Nios II systems can access Ethernet via the RJ-45 connector (RJ1). See Figure 1–4. The Nios II development kit includes hardware and software components that allow Nios II processor systems to communicate with the LAN91C111 Ethernet device.

Figure 1–4. Ethernet RJ-45 Connector



The Ethernet MAC / PHY device shares address and data connections with the flash memory and the SRAM chips. For shared bus information, see "Shared Bus Table" on page A–1.

See **www.smsc.com** for detailed information about the LAN91C111 device.

Expansion Prototype Connector (PROTO1)

The PROTO1 expansion prototype connectors share Cyclone I / O pins with the CompactFlash connector. Designs may use either the PROTO1 connectors or the CompactFlash.

Headers J11, J12, and J13 collectively form the standard-footprint, mechanically-stable connection that can be used (for example) as an interface to a special-function daughter card.

See the Altera web site for a list of available expansion daughter cards that can be used with the Nios development board at **www.altera.com/devkits**.

The expansion prototype connector interface includes:

- 41 I/O pins for prototyping. All 41 I/O pins connect to user I/O pins on the Cyclone device. Each signal passes through analog switches (U19, U20, U21, U22 and U25) to protect the Cyclone device from 5-V logic levels. These analog switches are permanently enabled.
- A buffered, zero-skew copy of the on-board OSC output from U2.
- A buffered, zero-skew copy of the Cyclone's phase-locked loop (PLL)-output from U60.
- A logic-negative power-on reset signal
- Five regulated 3.3-V power-supply pins (2A total max load for both PROTO1 & PROTO2)
- One regulated 5-V power-supply pin (1A total max load for both PROTO1 & PROTO2)
- Numerous ground connections

The output logic-level on the expansion prototype connector pins is 3.3V. The power supply included wit the Nios II development kit cannot supply the maximum load current specified above.

Figure 1–5, Figure 1–6, and Figure 1–7 show connections from the PROTO1 expansion headers to the Cyclone device. Unless otherwise noted, labels indicate Cyclone device pin numbers.

Figure 1–5. Expansion Prototype Connector - J11

RESET_n	1	●	●	2	GND
C18	3	●	●	4	C19
D18	5	●	●	6	D19
D17	7	●	●	8	D20
E17	9	●	●	10	F17
F18	11	●	●	12	E18
E19	13	●	●	14	F16
F15	15	●	●	16	F19
F20	17	●	●	18	G16
GND	19	●	●	20	NC
G15	21	●	●	22	GND
G20	23	●	●	24	GND
G19	25	●	●	26	GND
G14	27	●	●	28	H15
G18	29	●	●	30	GND
G17	31	●	●	32	H16
H18	33	●	●	34	NC
H17	35	●	●	36	H19
H20	37	●	●	38	U18
J16	39	●	●	40	GND

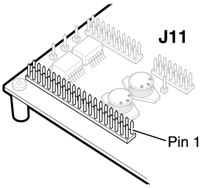
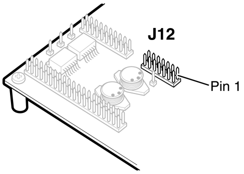
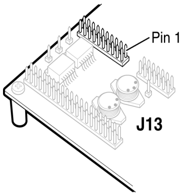


Figure 1–6. Expansion Prototype Connector - J12



GND	1	●	●	2	+5V
U20	3	●	●	4	J15
J19	5	●	●	6	J20
H14	7	●	●	8	J14
J17	9	●	●	10	J18
K15	11	●	●	12	W18
V18	13	●	●	14	V19

Figure 1–7. Expansion Prototype Connector - J13



(1) Vunreg (U54 pin 2)	1	●	●	2	GND
NC	3	●	●	4	GND
+3.3V	5	●	●	6	GND
+3.3V	7	●	●	8	GND
(2) PROTO1_OSC(U2 pin 4)	9	●	●	10	GND
(3) PROTO1_CLKIN (U2 pin 18)	11	●	●	12	GND
(4) PROTO1_CLKOUT (K6)	13	●	●	14	GND
+3.3V	15	●	●	16	GND
+3.3V	17	●	●	18	GND
+3.3V	19	●	●	20	GND

Notes to Figure 1–7:

(1) Unregulated voltage from AC to DC power transformer
(2) Clk from board oscillator
(3) Clk from FPGA via buffer
(4) Clk output from protocard to FPGA

Expansion Prototype Connector (PROTO2)

Headers J15, J16, and J17 collectively form the standard-footprint, mechanically-stable connection that can be used (for example) as an interface to a special-function daughter card.

The expansion prototype connector interface includes:

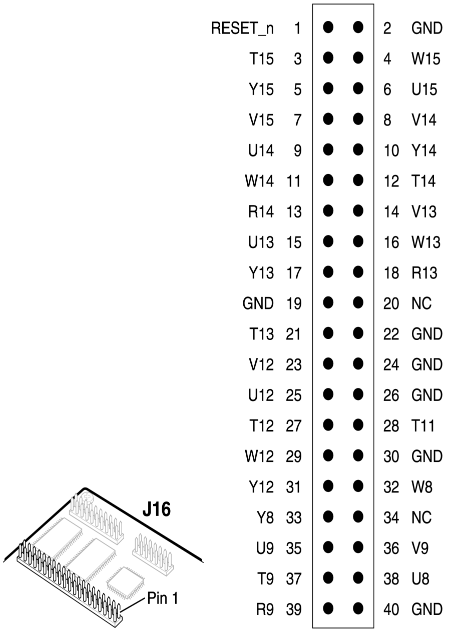
- 41 I/O pins for prototyping. All 41 I/O pins connect to user I/O pins on the Cyclone device. Each signal passes through analog switches (U27, U28, U29, U30 and U31) to protect the Cyclone device from 5-V logic levels. These analog switches are permanently enabled.
- A buffered, zero-skew copy of the on-board OSC output (from U2).
- A buffered, zero-skew copy of the Cyclone's phase-locked loop (PLL)-output (from U60)
- A logic-negative power-on-reset signal

- Five regulated 3.3-V power-supply pins (2A total max load for both PROTO1 & PROTO2)
- One regulated 5-V power-supply pin (1A total max load for both PROTO1 & PROTO2)
- Numerous ground connections

The output logic-level on the expansion prototype-connector pins is 3.3 V. The power supply included wit the Nios II development kit cannot supply the maximum load current specified above.

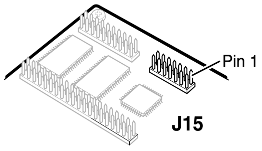
Figure 1–8, Figure 1–9, and Figure 1–10 show connections from the PROTO2 expansion headers to the Cyclone device. Unless otherwise noted, labels indicate Cyclone device pin numbers.

Figure 1–8. Expansion Prototype Connector - J16



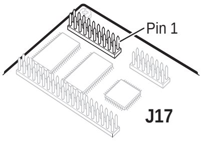
RESET_n	1	●	●	2	GND
T15	3	●	●	4	W15
Y15	5	●	●	6	U15
V15	7	●	●	8	V14
U14	9	●	●	10	Y14
W14	11	●	●	12	T14
R14	13	●	●	14	V13
U13	15	●	●	16	W13
Y13	17	●	●	18	R13
GND	19	●	●	20	NC
T13	21	●	●	22	GND
V12	23	●	●	24	GND
U12	25	●	●	26	GND
T12	27	●	●	28	T11
W12	29	●	●	30	GND
Y12	31	●	●	32	W8
Y8	33	●	●	34	NC
U9	35	●	●	36	V9
T9	37	●	●	38	U8
R9	39	●	●	40	GND

Figure 1–9. Expansion Prototype Connector - J15



GND	1	2	+5V
R11	3	4	Y9
W9	5	6	T10
U10	7	8	V10
W10	9	10	Y10
V11	11	12	U11
W11	13	14	Y11

Figure 1–10. Expansion Prototype Connector - J17



(1) Vunreg (U54 pin 2)	1	2	GND
NC	3	4	GND
+3.3V	5	6	GND
+3.3V	7	8	GND
(2) PROTO2_OSC(U2 pin 6)	9	10	GND
(3) PROTO2_CLKIN (U2 pin 17)	11	12	GND
(4) PROTO2_CLKOUT (K14)	13	14	GND
+3.3V	15	16	GND
+3.3V	17	18	GND
+3.3V	19	20	GND

Notes to Figure 1–10:

(1) Unregulated voltage from AC to DC power transformer
(2) Clk from board oscillator
(3) Clk from FPGA via buffer
(4) Clk output from protocard to FPGA

Mictor Connector

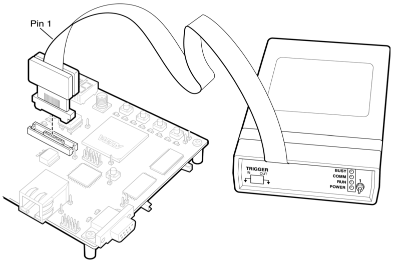
The Mictor connector (J25) can be used to transmit up to 27 high-speed I/O signals with very low noise via a shielded Mictor cable. J25 is used as a debug port. Twenty five of the Mictor connector signals are used as data, and two signals are used as clock input and clock output.

Most pins on J25 connect to I/O pins on the Cyclone device (U60). For systems that do not use the Mictor connector for debugging the Nios II processor, any on-chip signals can be routed to I/O pins and probed at J25 via a Mictor cable. External scopes and logic analyzers can connect to J25 and analyze a large number of signals simultaneously.

For details on Nios II debugging products that use the Mictor connector, see **www.altera.com**.

Figure 1–11 shows an example of an in-target system analyzer ISA-NIOS/T (sold separately) by First Silicon Solutions (FS2) Inc. connected to the Mictor connector. For details see **www.fs2.com.**

Figure 1–11. An ISA-Nios/T Connecting to the Mictor Connector (J25)



Five of the signals connect directly to both the JTAG pins on the Cyclone device (U60), and the Cyclone device's JTAG connector (J24). The JTAG signals have special usage requirements. You cannot use J24 and J25 at the same time.

Figure 1–12 below shows connections from the Mictor connector to the Cyclone device. Figure 1–13 shows the pin out for J25. Unless otherwise noted, labels indicate Cyclone device pin numbers.

Figure 1–12. Mictor Connector Signaling

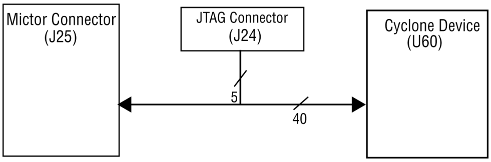
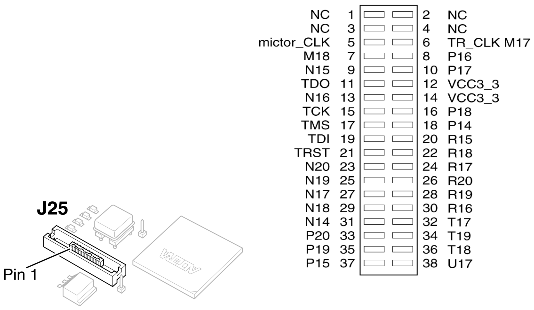


Figure 1–13. Debug Mictor Connector - J25



NC	1	2	NC
NC	3	4	NC
mictor_CLK	5	6	TR_CLK M17
M18	7	8	P16
N15	9	10	P17
TDO	11	12	VCC3_3
N16	13	14	VCC3_3
TCK	15	16	P18
TMS	17	18	P14
TDI	19	20	R15
TRST	21	22	R18
N20	23	24	R17
N19	25	26	R20
N17	27	28	R19
N18	29	30	R16
N14	31	32	T17
P20	33	34	T19
P19	35	36	T18
P15	37	38	U17

Serial Port Connectors

J19 & J27 are standard DB-9 serial connectors. These connectors are typically used for communication with a host computer using a standard, 9-pin serial cable connected to (for example) a COM port. Level-shifting buffers (U52 and U58) are used between J19 & J27 and the Cyclone device, because the Cyclone device cannot interface to RS-232 voltage levels directly.

The Nios development board provides two serial connectors, one labeled Console and the other labeled Debug. Many processor systems make use of multiple UART communication channels during prototype and debug stages.

The Console serial port is able to transmit all RS-232 signals. Alternately, the Cyclone design may use only the signals it needs, such as RXD and TXD. LEDs are connected to the RXD and TXD signals, giving a visual indication when data is being transmitted or received. Figure 1–14 and Figure 1–15 show the pin connections between the Console and Debug serial connectors and the Cyclone device.

Figure 1–14. Console Serial Port Connector - J19

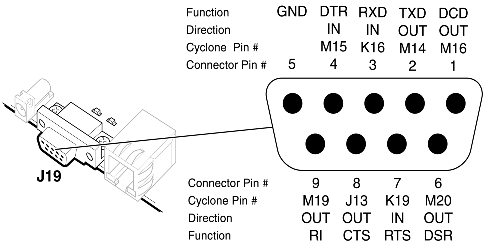
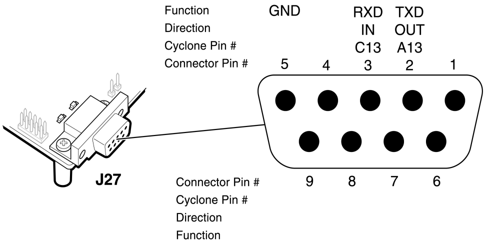

Function GND DTR RXD TXD DCD
Direction IN IN OUT OUT
Cyclone Pin # M15 K16 M14 M16
Connector Pin # 5 4 3 2 1
J19
Connector Pin # 9 8 7 6
Cyclone Pin # M19 J13 K19 M20
Direction OUT OUT IN OUT
Function RI CTS RTS DSR

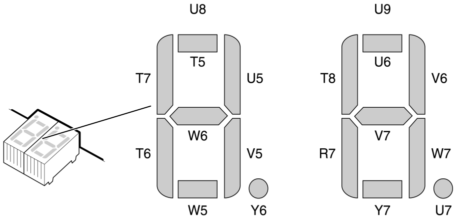
Figure 1–15. Debug Serial Port Connector – J27



Dual 7-Segment Display

U8 and U9 are connected to the Cyclone device so that each segment is individually controlled by a general-purpose I/O pin. When the Cyclone pin drives logic 0, the corresponding LED turns on. See Figure 1–16 for Cyclone device pin-out details.

Figure 1–16. Dual-7-Segment Display



The factory-programmed Nios II reference design includes parallel input/output (PIO) registers and logic for driving this display.

Push-Button Switches

SW0 – SW3 are momentary-contact push-button switches that provide stimulus to designs in the Cyclone device. See Figure 1–17. Each switch is connected to an Cyclone general-purpose I/O pin with a pull-up resistor as shown in Table 1–4. The Cyclone device pin will see a logic 0 when each switch is pressed.

Table 1–4. Push Button Switches Pin Out Table

Button	SW0	SW1	SW2	SW3
Cyclone Pin	W3	Y4	V4	W4

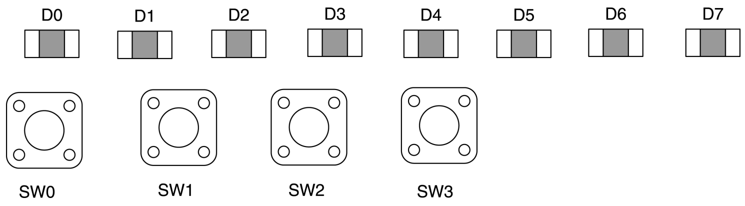
Individual LEDs

This Nios development board provides eight individual LEDs connected to the Cyclone device. See Figure 1–17. D0 – D7 are connected to general purpose I/O pins on the Cyclone device as shown in Table 1–5. When the Cyclone pin drives logic 1, the corresponding LED turns on.

Table 1–5. LED Pin Out Table

LED	D0	D1	D2	D3	D4	D5	D6	D7
Cyclone Pin	E14	E13	C14	D14	E12	F12	B3	B14

Figure 1–17. SW0 – SW3 Push Button Switches and Individual LEDs



Serial Flash Connector

The serial flash connector (J28) connects to the EPCS4 serial configuration device and allows designers to program the EPCS4 using the Quartus II software. Hardware designers do not need the serial flash connector for using the reference designs that came with the Nios II development kit. The reference designs allow the designer to access data from the EPCS4 device without using this connector. See "Cyclone Configuration" on page 1–21.

See the "Serial Configuration Devices (EPCS1 & EPCS4) Data Sheet" in the *Cyclone Device Handbook* for more information about this device and the "EPCS Device Controller Core with Avalon Interface" chapter in the *Nios II Processor Handbook* for information about the EPCS device controller component in SOPC Builder.

Serial Configuration Device (EPCS4)

U59 is a serial configuration device connected to the Cyclone FPGA. Serial configuration devices are flash memory devices with a serial interface that can store configuration data for a Cyclone FPGA, and load the data into the FPGA upon power-up or re-configuration. You can use the EPCS4 serial configuration device to store FPGA configuration data, or Nios II program data, or both.

The SOPC Builder EPCS Serial Flash Controller component enables Nios II processor systems to access the EPCS4 device. You can use the EPCS serial flash controller to write new data into the EPCS4 device, and to read Nios II program code or data from the device.

U59 is blank by default. You can use the Quartus II software to program FPGA configuration data (a **.pof** file) into the EPCS4 device connected to connector J28. J28 connects to the pins of U59, enabling you to program the device via an Altera download cable, such as the USB Blaster cable.

☞ The orientation of J28 is the reverse of J24. When the Altera download cable is connected to J28 correctly, the cable will rest over the Cyclone device.

For details on the Cyclone configuration process with the EPCS4 device, see "Configuration Controller Device (EPM7128AE)" on page 1–20.

See the "Serial Configuration Devices (EPCS1 & EPCS4) Data Sheet" in the *Cyclone Device Handbook* for more information about the EPCS4 device. See the "EPCS Device Controller Core with Avalon Interface" chapter in the *Nios II Processor Handbook* for information about the EPCS serial flash controller component in SOPC Builder.

Configuration Controller Device (EPM7128AE)

The configuration controller (U3), is an Altera MAX® 7000 EPM7128AE device. It comes pre-programmed with logic for managing board reset conditions and configuring the Cyclone device from data stored in flash memory and the EPCS4 serial configuration device.

Reset Distribution

The EPM7128AE takes a power-on reset pulse from the Linear Technologies 1326 power-sense / reset-generator chip and distributes it (through internal logic) to other reset pins on the board, including the:

- LAN91C111 (Ethernet MAC / PHY) reset
- Flash memory reset
- Reset signals delivered to the expansion prototype connectors (PROTO1 & PROTO2)

Starting Configuration

There are four methods to start a configuration sequence. The four methods are the following:

1. Board power-on.

2. Pressing the *Power-On Reset* button (SW10).

3. Asserting (driving 0 volts on) the EPM7128AE's `reconfigreq_n` input pin (from a Cyclone design).

4. Pressing the *Force Safe* button (SW9).

Cyclone Configuration

At power-up or reset, the configuration controller attempts to configure the Cyclone device with data from one of three sources, in the following order:

1. The EPCS4 serial configuration device

2. The User configuration from flash memory

3. The Safe configuration from flash memory

First, the configuration controller puts the Cyclone FPGA in active serial (AS) configuration mode. The Cyclone FPGA will then attempt to read configuration data from the EPCS4. If the Cyclone FPGA is successfully configured, the configuration controller stops.

If configuration from the EPCS4 was not successful, the configuration controller puts the Cyclone FPGA into passive serial mode and attempts to load the user configuration from flash memory. If this also fails, the configuration controller attempts to load the safe configuration in flash memory.

When SW9 (*Force Safe*) is pressed, the configuration controller immediately begins configuring the Cyclone FPGA from the safe configuration in flash memory—data in the EPCS4 and in the user hardware image are ignored.

Most users will never need to re-program the configuration controller design in the EMP7128AE device. Reprogramming the configuration controller may result in an inoperable development board. A programming file (**config_controller.pof**) with the original configuration controller logic is included with the Nios II development kit. If you have changed the EPM7128AE device logic, you can restore the factory configuration using this programming file located in the **EPM7128_PS_config** folder of the **examples** directory for this board.

See the MAX 7000 family literature at **www.altera.com/literature/lit-m7k.html** for detailed information about the Altera EPM7128AE device.

Configuration Data

FPGA configuration data files are generated by the Quartus II software. You can write new configuration data to the board's flash memory using the Nios II IDE.

For details on programming configuration data to flash memory, see the *Nios II Flash Programmer' User Guide* or refer to the Nios II IDE online help.

Safe & User Configurations

The configuration controller can manage three separate Cyclone device configurations: one configuration stored in the EPSC4 configuration device, and two configurations stored in flash memory. These two configurations stored in flash memory are conventionally referred to as the safe configuration and the user configuration.

The configuration controller expects user configuration and safe configuration files to be stored at fixed locations (offsets) in flash memory. Table 1–7 on page 1–25 shows how the configuration controller expects flash memory contents to be arranged.

A Nios II reference design is factory programmed into the safe-configuration region of the flash memory. Altera recommends that users avoid overwriting the safe configuration data.

When SW9 (Force Safe) is pressed, the configuration controller will ignore the user configuration and always configure the Cyclone device from the safe configuration. This switch allows you to “escape” from the situation where a valid-but-nonfunctional user configuration is present in flash memory or the serial configuration device.

Using Conventional Flash Memory

The Nios development board includes an 8 MByte flash-memory device (U5). See Table 1–6. It is divided into 128 individually-erasable 64K sectors. The factory-programmed design, and (more importantly) the on-board configuration controller, makes certain assumptions about what–resides–where in flash memory.

In the factory-programmed state, the upper four (4) MBbytes of flash memory are used to store either FPGA configuration data or web-page data. Your application software may safely use the lower half (4 MBytes) of flash memory without interfering with FPGA configuration or web-server operation.

Table 1–6. Flash Memory Allocation

Address (hex)	Flash Allocation
000000	**4MB**
100000	
200000	
300000	
400000	**Web Pages (2MB)**
500000	
600000	**User Configuration Data (1 MB)**
700000	**Safe Configuration Data (1 MB)**

☞ The factory-programmed reference design implements a web server. Network settings and web pages are pre-programmed in the flash memory, as shown in Table 1–7 on page 1–25.

User Hardware Image

At power on, or when the *Power-On Reset* button (SW10) is pressed, the configuration controller begins reading user configuration data out of flash at address 0x600000. This data, and suitable control signals, are used in an attempt to configure the FGPA. FPGA configuration data written into this region of flash memory is conventionally called the user hardware image. Nios II development tools documentation on how to create your own user hardware image data and several facilities for burning your user hardware image into flash memory.

Safe Hardware Image

If there is no valid user hardware image, or if SW9 (Force Safe) is pressed, the configuration controller begins reading data out of flash at address 0x700000. Any FPGA configuration data stored at this location is conventionally called the safe hardware image. Your development board was factory-programmed with a safe hardware image, plus additional data located in the range 0x700000-0x7FFFFF, as shown in Table 1–7 on page 1–25.

☞ The Nios II development kit includes the source files for the factory-programmed reference design.

The configuration controller will stop reading data when the FPGA successfully configures. The **safe example** design is setup to begin executing code from address 0x7B0000. This region of flash memory is factory-programmed with the web-server application software.

☞ ***Do Not Erase*** your safe hardware image (safe hardware configuration data). If you do so inadvertently, see “Restoring the Factory Configuration” on page B–1 for instructions on how to restore your board to its factory configuration.

Table 1–7. Safe Hardware Configuration Data Memory Allocation

Address (hex)	Safe Hardware Image
700000	**FPGA Configuration Data**
710000	
720000	
730000	
740000	
750000	
760000	
770000	
780000	
790000	
7A0000	
7B0000	**Web Server Software**
7C0000	
7D0000	
7E0000	
7F0000	**Network Settings**

The Configuration-Status LEDs

The EPM7128AE device is connected to four status LEDs that show the configuration status of the board at a glance (see Figure 1–18). The user can tell which configuration, if any, was loaded into the board at power-on by looking at the LEDs (see Table 1–8). If a new configuration was downloaded into the Cyclone device via JTAG, then all of the LEDs will turn off.

Figure 1–18. LED1 – LED4

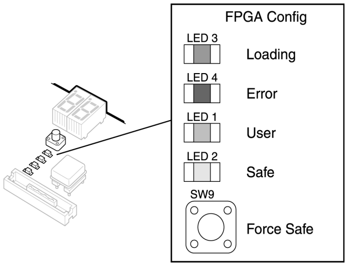


Table 1–8. Configuration Status LED Indicators

LED	LED Name	Color	Description
LED3	Loading	Green	This LED blinks while the configuration-controller is actively transferring data from flash memory into the Cyclone FPGA.
LED4	Error	Red	If the red Error LED is on, then configuration was NOT transferred from flash memory into the Cyclone device. This can happen if, for example, the flash memory contains neither a valid user or safe configuration.
LED1	User	Green	If the Cyclone device was successfully configured with data from the EPCS4, LED1 will blink slowly. If the Cyclone device was successfully loaded with the user configuration from flash memory, LED1 will remain on continuously.
LED2	Force Safe	Amber	This LED turns on when the safe configuration is being transferred from flash memory and stays illuminated if the safe configuration was successfully loaded into the Cyclone device.

Configuration & Reset Buttons

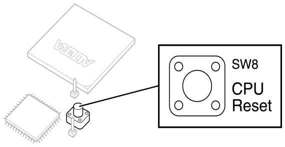
The Nios development board uses dedicated switches SW8, SW9 and SW10 for the following fixed functions:

SW8 – CPU Reset

When SW8 is pressed, a logic-0 is driven onto the Cyclone devices' DEV_CLRn pin (and user I/O C_4). The result of pressing SW8 depends on how the Cyclone device is currently configured.

The pre-loaded Nios II reference design treats SW8 as a CPU-reset pin (see Figure 1–19). The Nios II reference design will reset and start executing code from its reset address when SW8 is pressed.

Figure 1–19. Safe Config Button



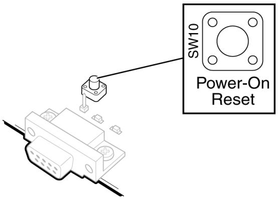
SW9 – Force Safe

Pressing Force Safe (SW9) commands the configuration controller to re-configure the Cyclone device with the factory-programmed safe configuration.

SW10 – Power-On Reset

Power-On Reset (SW10) is the power-on reset button (see Figure 1–20). When SW10 is pressed, a logic 0 is driven to the power-on reset controller (U18). See "Power-Supply Circuitry" on page 1–28 for more details. After SW10 is pressed, the configuration controller will load the Cyclone device from flash memory.

Figure 1–20. Power-On Reset Configuration Button



Power-Supply Circuitry

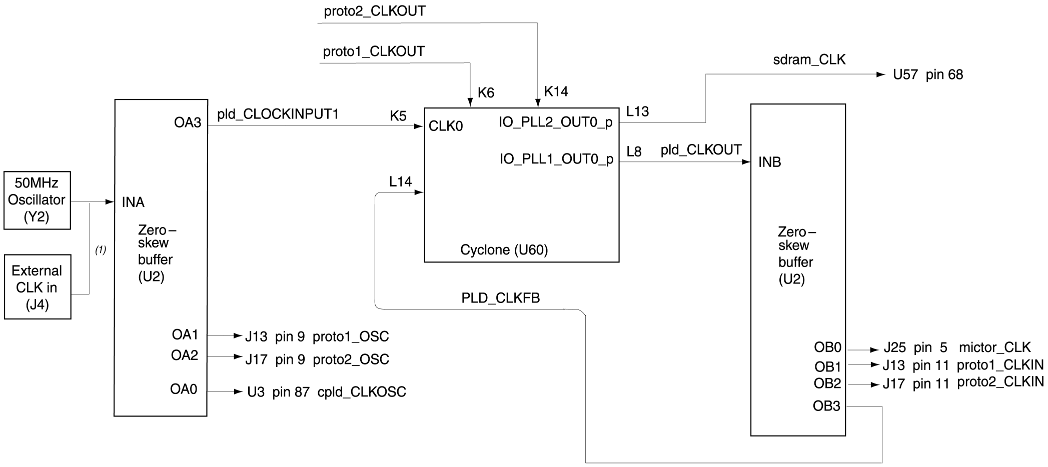
The Nios development board runs from a 9-V, unregulated, center-negative input power supply. On-board circuitry generates 5-V, 3.3-V, and 1.5-V regulated power levels.

- The 5-V supply is present on pin 2 of J12 and J15 for use by any device plugged into the PROTO1 or PROTO2 expansion connectors.
- The 3.3-V supply is used as the power source for all Cyclone device I/O pins. The 3.3-V supply is also available to PROTO1 and PROTO2 daughter cards.
- The 1.5-V supply is used only as the power supply for the Cyclone device core (VCCINT) and it is not available on any connector or header.

Clock Circuitry

The Nios development board includes a 50 MHz free-running oscillator and a zero-skew, point-to-point clock distribution network. The clock network drives the Cyclone device and pins on the expansion prototype connectors, the configuration controller device, and the Mictor connector. The zero-skew buffer distributes both the free-running 50 MHz clock and the clock-output from one of the Cyclone's device internal PLLs (CLKLK_OUT1). See Figure 1–21.

Figure 1–21. Clock Circuitry



Note to Figure 1–21:

(1) An external clock can be enabled by stuffing location R15 with a 49.9 ohm 0603 resistor and stuffing location R13 with a 330 ohm 0603 resistor.

A socketed 50 MHz free-running oscillator (Y2) supplies the fundamental operating frequency, and a clock buffer (U2) drives zero-skew clock signals to various points on the board.

The Cyclone device can synthesize a new clock signal internally using on-chip PLLs, and distribute the clock to various locations on the board by outputting the clock signal to the IO_PLL1_OUT0_p pin. The clock buffer drives this signal to the following locations:

- The PROTO1_CLKIN and PROTO2_CLKIN pins on the expansion prototype connectors, allowing a user-defined clock to drive each of the expansion prototype headers.
- The clock input for the SDRAM memory (U57), allowing SDRAM to run at a different rate than the clock oscillator.
- The CLK0 clock input on the Cyclone device.

The Cyclone device can also supply a clock from the IO_PLL2_OUT0_p pin to the SDRAM (U57).

☞ The 50 MHz oscillator (Y2) is socketed and can be changed by the user. However, the EPM7128AE device configuration control circuit and other Altera reference designs are not guaranteed to work at different frequencies. It is the user’s responsibility to accommodate a new clock oscillator when designing a system.

JTAG Connectors

The Nios development board, has three 10-pin JTAG headers (J24, J5 and J28) compatible with Altera download cables such as the USB Blaster™. Each JTAG header connects to one Altera device and forms a single-device JTAG chain. J24 connects to the Cyclone device (U60), J5 connects into the EPM7128AE device (U3), and J28 connects to the EPCS4 serial-configuration device (U59).

JTAG Connector to Cyclone Device (J24)

J24 connects to the JTAG pins (TCK, TDI, TDO, TMS, TRST) of the Cyclone device (U60) as shown in Figure 1–22. Altera Quartus II software can directly configure the Cyclone device with a new hardware image via an Altera download cable as shown in Figure 1–23. In addition, the Nios II IDE can access the Nios II processor’s JTAG debug module via a download cable connected to the J24 JTAG connector.

Figure 1–22. JTAG Connector (J24) to Cyclone Device

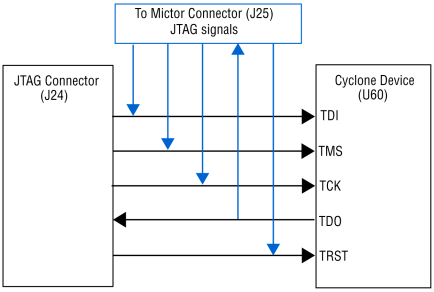
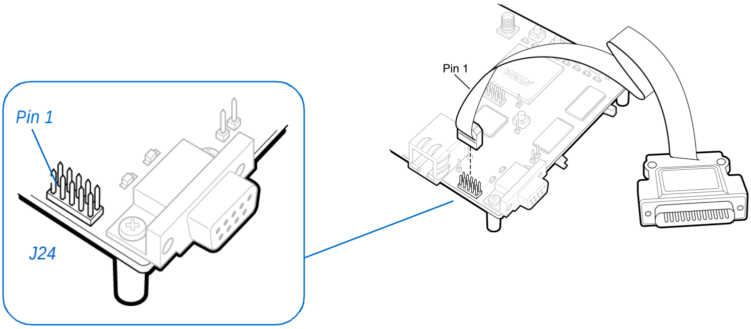


Figure 1–23. JTAG Connection to Download Cable



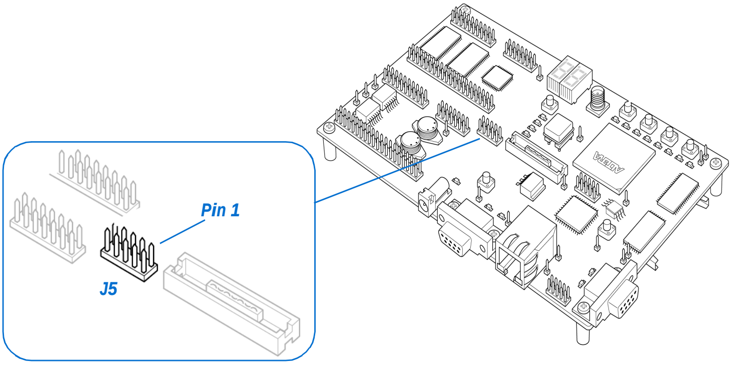
The Cyclone device's JTAG pins can also be accessed via the Mictor connector (J25). The pins of J24 are connected directly to pins on J25, and care must be taken so that signal contention does not occur between the two connectors.

JTAG Connector to EPM7128AE Device (J5)

J5 connects to the JTAG pins (TCK, TDI, TDO, TMS, TRST) of the EMP7128AE device (U3) as shown in Figure 1–24. Altera Quartus II software can perform in-system programming (ISP) to reprogram the EMP7128AE device (U3) with a new hardware configuration via an Altera download cable.

☞ Note that the orientation of J5 is rotated 180° compared to J24.

Figure 1–24. JTAG Connector (J5) to EPM7128AE Device



Appendix A. Shared Bus Table

Description

On the Nios development board, Cyclone Edition, the flash memory, SRAM, and Ethernet MAC / PHY devices share address and control lines. These shared lines are referred to as the Shared Bus. Using SOPC Builder, designers can interface a Nios II processor system to any device connected to the off-chip Shared Bus. Table A–9 on page A–1 lists all connections between the devices connected to the Shared Bus.

Table A–9. Shared Bus Table (Part 1 of 3)

NET Name	NET Description	PLD (U60)		Flash (U5)		SRAM (U35)		SRAM (U36)		Ethernet (U4)	
		Pin Name	Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name	Pin #
FSE_A0	Shared Address	IO	B4	A0	27						
FSE_A1		IO	A4	A1	22					A1	78
FSE_A2		IO	D5	A2	21	A0	1	A0	1	A2	79
FSE_A3		IO	D6	A3	20	A1	2	A1	2	A3	80
FSE_A4		IO	C5	A4	19	A2	3	A2	3	A4	81
FSE_A5		IO	B5	A5	18	A3	4	A3	4	A5	82
FSE_A6		IO	C2	A6	17	A4	5	A4	5	A6	83
FSE_A7		IO	D2	A7	16	A5	18	A5	18	A7	84
FSE_A8		IO	D4	A8	10	A6	19	A6	19	A8	85
FSE_A9		IO	D1	A9	9	A7	20	A7	20	A9	86
FSE_A10		IO	E4	A10	42	A8	21	A8	21	A10	87
FSE_A11		IO	E5	A11	8	A9	22	A9	22	A11	88
FSE_A12		IO	F3	A12	7	A10	23	A10	23	A12	89
FSE_A13		IO	E3	A13	6	A11	24	A11	24	A13	90
FSE_A14		IO	E2	A14	5	A12	25	A12	25	A14	91
FSE_A15		IO	F4	A15	4	A13	26	A13	26	A15	92
FSE_A16		IO	F5	A16	3	A14	27	A14	27		
FSE_A17		IO	F2	A17	46	A15	42	A15	42		
FSE_A18		IO	F1	A18	15	A16	43	A16	43		
FSE_A19		IO	F6	A19	43	A17	44	A17	44		
FSE_A20		IO	G5	A20	44						
FSE_A21		IO	G1	A21	35						
FSE_A22		IO	G2	A22	2						

Table A–9. Shared Bus Table (Part 2 of 3)

NET Name	NET Description	PLD (U60)		Flash (U5)		SRAM (U35)		SRAM (U36)		Ethernet (U4)	
		Pin Name	Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name	Pin #
FSE_D0	Shared Data	IO	C6	D0	31	D0	7			D0	107
FSE_D1		IO	E6	D1	32	D1	8			D1	106
FSE_D2		IO	B6	D2	33	D2	9			D2	105
FSE_D3		IO	A6	D3	34	D3	10			D3	104
FSE_D4		IO	F7	D4	38	D4	13			D4	102
FSE_D5		IO	E7	D5	39	D5	14			D5	101
FSE_D6		IO	B7	D6	40	D6	15			D6	100
FSE_D7		IO	A7	D7	41	D7	16			D7	99
FSE_D8		IO	D7			D8	29			D8	76
FSE_D9		IO	C7			D9	30			D9	75
FSE_D10		IO	F8			D10	31			D10	74
FSE_D11		IO	E8			D11	32			D11	73
FSE_D12		IO	B8			D12	35			D12	71
FSE_D13		IO	A8			D13	36			D13	70
FSE_D14		IO	D8			D14	37			D14	69
FSE_D15		IO	C8			D15	38			D15	68
FSE_D16		IO	B9					D0	7	D16	66
FSE_D17		IO	A9					D1	8	D17	65
FSE_D18		IO	D9					D2	9	D18	64
FSE_D19		IO	C9					D3	10	D19	63
FSE_D20		IO	E9					D4	13	D20	61
FSE_D21		IO	E10					D5	14	D21	60
FSE_D22		IO	B10					D6	15	D22	59
FSE_D23		IO	A10					D7	16	D23	58
FSE_D24		IO	F10					D8	29	D24	56
FSE_D25		IO	C10					D9	30	D25	55
FSE_D26		IO	D10					D10	31	D26	54
FSE_D27		IO	C11					D11	32	D27	53
FSE_D28		IO	D11					D12	35	D28	51
FSE_D29		IO	B11					D13	36	D29	50
FSE_D30		IO	A11					D14	37	D30	49
FSE_D31		IO	E11					D15	38	D31	48

Table A–9. Shared Bus Table (Part 3 of 3)

NET Name	NET Description	PLD (U60)		Flash (U5)		SRAM (U35)		SRAM (U36)		Ethernet (U4)	
		Pin Name	Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name	Pin #
FLASH_CS_n	Chip Select	IO	A12	CE_n	28						
FLASH_OE-N	Read Enable	IO	B12	OE_n	30						
FLASH_RW-N	Write Enable	IO	D12	WE_n	11						
FLASH_RY-BY_N	Ready/Busy	IO	C12	RY/BY_n	14						
SRAM_BE_N0	Byte Enable 0	IO	V17			BE0#	39				
SRAM_BE_N1	Byte Enable 1	IO	V16			BE1#	40				
SRAM_BE_N2	Byte Enable 2	IO	W16					BE2#	39		
SRAM_BE_N3	Byte Enable 3	IO	T16					BE3#	40		
SRAM_CS_N	Chip Select	IO	W17			CS_n	6	CS_n	6		
SRAM_OE_N	Read Enable	IO	Y17			OE_n	41	OE_n	41		
SRAM_WE_N	Write Enable	IO	U16			WE_n	17	WE_n	17		
ENET_ADS_N	Address Strobe	IO	A14							ADS#	37
ENET_AEN	Address Enable	IO	B15							AEN	41
ENET_BE_N0	Byte Enable 0	IO	C16							BE0#	94
ENET_BE_N1	Byte Enable 1	IO	B16							BE1#	95
ENET_BE_N2	Byte Enable 2	IO	D16							BE2#	96
ENET_BE_N3	Byte Enable 3	IO	E16							BE3#	97
ENET_CYCLE_N	Bus Cycle	IO	B17							CYCLE#	35
ENET_DATACS_N	Data Chip Select	IO	C15							DATACS#	34
ENET_INTRQ0	Interrupt	IO	D15							INTRO	29
ENET_IOCHRDY	IO Char Ready	IO	F14							ARDY	38
ENET_IOR_N	Read	IO	A15							RD#	31
ENET_IOW_N	Write	IO	E15							WR#	32
ENET_LCLK	Local Bus Clock	IO	C17							LCLK	42
ENET_LDEV_N	Local Device	IO	D3							LDEV#	45
ENET_RDYRTN_N	Ready Return	IO	B18							RDYRTN#	46
ENET_W_R_N	Write/Read	IO	A17							W/R#	36

Appendix B. Restoring the Factory Configuration

Introduction

Your Nios development board always can be restored to its factory-programmed configuration. To restore the factory configuration, you must reprogram the flash memory on the board and you must reprogram the EPM7128AE configuration controller device.

If you have a Nios development board, Cyclone Edition, already programmed with the first-generation Nios II processor, and you want to start using the Nios II processor, first you must update your development board using these instructions.

The files required for this operation are included in the Nios II development kit's <*Nios II kit path*>/**examples/factory_recovery** directory.

Reprogramming the Flash Memory

To reprogram the flash memory on the development board, perform the following steps:

1. Open a Nios II SDK Shell by choosing **Windows Start > Programs > Altera > Nios II Development Kit** <*installed version*> **> Nios II SDK Shell**.

2. From the **example** directory, change to the **factory_recovery** directory for your development kit.

   ```
   cd factory_recovery/niosII_cyclone_1c20
   ```

3. Run the flash-restoration script:

   ```
   ./restore_my_flash
   ```

Follow the script's instructions.

Reprogramming the EPM7128AE Configuration Controller Device

The EPM7128AE configuration controller device also must be reprogrammed.

1. Move the programming cable from J24 to J5, labeled *For U3*.

   ☞ The orientation of J5 is opposite that of J24. Be sure to rotate the connector on the end of the programming cable 180 degrees before plugging it into J5. When properly connected to J5, the programming cable lies naturally across the **FPGA Config LEDs** and the **dual seven-segment display**.

2. Launch the Quartus II software, and open the **Programmer** (Tools menu).

3. Click **Add File** and select the following programming file:

   *<Nios II kit path>***/examples/factory_recovery/**
   niosII_cyclone_1c20/config_controller.pof

4. In the **Programmer**, check the **Program/Configure** box, and click **Start** to reprogram the EPM7128AE device.

5. Push the *Force Safe* button to perform a power-on reset and reconfigure the Cyclone device from Flash memory. You should see the **Safe** LED turned on and activity on LEDs D0 through D7.

Your board is now re-configured to the default factory condition.

Appendix C. Connecting to the Board via Ethernet

Introduction

The Nios development board is factory-programmed with a default reference design that implements a web server, among other functions. The sections below describe how to connect a host computer to the board's Ethernet port, assign an IP address to the board, and browse to the web server from the host computer.

Connecting the Ethernet Cable

The Nios II development kit includes an Ethernet (RJ45) cable and a male/female RJ45 crossover adapter. Before you connect these components, you must decide how you want to use the network features of your board. Select one of the two following connection methods:

1. *LAN Connection* — To use your Nios development board on a LAN (for example, connecting to an Ethernet hub) do the following:

    a. Connect one end of the RJ45 cable to the Ethernet connector on the development board (RJ1).

    b. Connect the other end to your LAN connection (hub, router, wall plug, etc.).

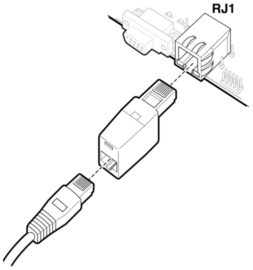
2. *Point-to-Point Connection* — To use your Nios development board connected directly to a host computer point-to-point (not on a LAN), do the following:

    a. Connect one end of your RJ45 cable to the female socket in the crossover adapter.

    b. Insert the male end of the crossover adapter into RJ1 on the Nios development board.

    c. Connect the other end of the RJ45 connector directly to the network (Ethernet) port on your host computer (see Figure C–1 on page C–2).

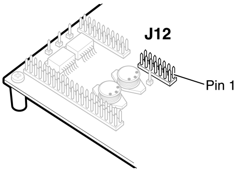
Figure C–1. Point-to-Point Connection



Connecting the LCD Screen

Your Nios II development kit was delivered with a two-line x 16-character LCD text screen. The web-server software displays useful status and progress messages on this display. If you wish to use the network features of the board, connect the LCD screen to the Expansion Prototype Connector (J12), as shown in Figure C–2. See the *Nios II Development Kit, Getting Started User Guide* for details.

Figure C–2. Expansion Prototype Header - J12



Obtaining an IP Address

In order to function on a network (either LAN or point-to-point), your board must have an IP address. This section describes the methods to assign an IP address to your board.

LAN Connection

If you have connected your board to a LAN, the board will either obtain a dynamic IP address using DHCP, or a static IP address stored in flash memory. If you do not know whether or not your LAN supports DHCP, it is easiest to try DHCP first.

DHCP

Upon reset, the web server will attempt to acquire an IP address via the DHCP protocol. The board will continue to attempt DHCP self-configuration for two minutes. You can determine if DHCP has succeeded, or if it is still in progress, by reading status messages on the LCD screen. If your LAN does not support DHCP, then DHCP configuration will ultimately fail and the web server will default to a static IP address.

If DHCP succeeds, the board will display a success message and the IP address on the LCD screen. The web server is now ready to display web pages. See "Browsing Your Board" on page C–5 to continue.

Static IP Address

If the DHCP process fails, the board will use a static IP address stored in flash memory. You need to obtain a safe IP address in your LAN's subnet from your system administrator. Once you know a safe IP address, you can assign it to your board using the steps below.

These steps send IP configuration data to the board via an Altera JTAG download cable, such as the USB Blaster cable.

1. Install the Nios II development tools, connect the JTAG download cable, and apply power to the board, as described in the *Nios II Development Kit, Getting Started User Guide*.

2. Choose **Start** > **Programs** > **Altera** > **Nios II Development Kit** > **Nios II SDK Shell** to open the Nios II SDK Shell. A shell window appears with a command prompt.

3. Press the SW9 button on the board labeled *Force Safe*.

4. At the Nios II SDK Shell command prompt, type:

   ```
   nios2-terminal<Enter>
   ```

   This command opens a terminal connection via the JTAG download cable to a monitor program running on the board. The monitor program displays status messages and text instructions that tell you how to set the IP address for your board.

5. Press the `!` key to abort the DHCP process and display a prompt. If you don't abort the DHCP process, it will fail after two minutes, and eventually a prompt will appear.

☞ The monitor's prompt is the + character. You can enter `h<Enter>` at the prompt for a complete list of supported commands.

6. At the prompt, type `xip:`<*safe IP address*>`<Enter>`

   The `xip` command saves the IP address in flash memory. In general, you will only need to assign an IP address to your board once. However, you may change it at any time by issuing another `xip` command. You can also use the commands `xsubnet` and `xgateway` to assign subnet and gateway addresses, but setting these addresses is not usually necessary.

7. Type `xdhcp:off<Enter>` to disable the board from attempting to obtain the IP address using DHCP in the future. (You can re-enable DHCP later, using the `xdhcp:on` command.)

8. Type CTRL+ C to terminate the JTAG terminal session and disconnect from the monitor program, then close the Nios II SDK Shell.

9. Press the SW8 button labeled *CPU Reset* to reboot the Nios II processor and start the web server using the new IP address. The LCD screen will display the static IP address assigned to the board, along with other status messages.

The web server is now ready to display pages using the IP address you assigned. See "Browsing Your Board" on page C–5 to continue.

Point–to–Point Connections

All boards are factory programmed with a default IP address of 10.0.0.51 stored in flash memory. The 10.0.0.x subnet is conventionally reserved for development, test, and prototyping. If DHCP fails or is aborted, the board will use this static IP address. The LCD screen displays status messages to indicate when the web server starts running using the default IP address.

Your host computer and the development board are the only two devices connected to this simple point-to-point network. For most host operating systems, it is necessary to assign your host computer an IP address on the same subnet as the board. For example, the address 10.0.0.1 will work fine. Any address in the 10.0.0.x subnet will work, and there is no possibility of conflicting with another device on the network. After modifying the host computer's IP address, your computer is ready to connect to the web server. See "Browsing Your Board" on page C–5 to continue.

If you don't have the ability to change the IP address of your host computer, you could change the IP address of the board to match the subnet of the host computer. For example, if your computer's IP address is 1.2.3.4, then you could assign the address 1.2.3.5 to your board. To change the board IP address, follow the steps in "Static IP Address" on page C–3.

Every time you reset the board, the web server will attempt to obtain an IP address via DHCP, which takes two minutes to time out. You can abort the DHCP process, or disable DHCP entirely by using the steps in "Static IP Address" on page C–3.

Browsing Your Board

Once your board has a valid IP address (obtained from either DHCP self-configuration or from flash memory), you can access the board via a web browser (e.g., Microsoft Internet Explorer). To browse this site, open a web browser and type the IP address of the board (four numbers separated by decimal-points) as a URL directly into the browser's **Address** input field. You can determine your board's IP address by reading the messages displayed on the LCD screen (the IP address is continuously displayed).

Index

A

Appendix A
    Shared bus table A–1
Appendix C
    Connecting to the Board via Ethernet C–1

B

Block diagram 1–2
Board Ethernet connection
    Browse the board C–5
    Connecting the Ethernet cable C–1
    Connecting the LCD display C–2
    Obtaining an IP Address C–2

C

Clock circuitry 1–28
CompactFlash connector 1–6
Configuration and reset buttons 1–26
    SW10 - Reset config 1–27
    SW8 - CPU reset 1–26
    SW9 - Safe config 1–27
Configuration controller device 1–20
    Configuration data 1–22
    Configuration-status LEDs 1–25
    Cyclone configuration 1–21
    Reset distribution 1–21
    Safe and user configurations 1–22
    Starting configuration 1–21
Configuration-status LEDs
    Indicators 1–26
Conventional flash memory usage 1–23
Cyclone EP1C20 device 1–4

D

Development board
    Features 1–1
    General description 1–1
Dual 7-segment display 1–18
    U8 & U9 pin information 1–18
Dual SRAM devices 1–10

E

Ethernet PHY/MAC 1–11
Expansion connector header (PROTO1) 1–11
    J11 pin information 1–12
    J12 pin information 1–13
    J13 pin information 1–13
Expansion connector header (PROTO2) 1–13
    J15 pin information 1–15
    J16 pin information 1–14
    J17 pin out information 1–15

F

Flash memory allocation 1–23
Flash memory device 1–5

I

Individual LEDs (D0 - D7) 1–19
    Pin information 1–19

J

JTAG connections
    JTAG to Cyclone device (J24) 1–29
    JTAG to MAX device (J5) 1–30
JTAG connectors 1–29

M

Mictor connector 1–15
    Debug port to OCI debug module 1–16
    J25 pin information 1–17

P

Power-supply circuitry 1–28
Push-button switches 1–19
  pin information 1–19

R

Reference design
  default 1–2
  restoring 1–3
Restore factory configuration
  Reprogramming the EPM7128AE configuration controller device B–2
  Reprogramming the flash memory B–1

S

Schematics 1–3
SDRAM device 1–8
Serial configuration device (EPCS4)
  EPCS4 device 1–20
Serial flash connector 1–19
Serial port connectors 1–17
  Console
    J19 pin information 1–18
  Debug
    J27 pin information 1–18
Shared bus table A–1
SRAM devices 1–10
SW10 button 1–27
SW8 button 1–26
SW9 button 1–27