Nios Development Board
Cyclone II Edition Reference Manual

Part Number MNL-N051805-1.3

I.S. EN ISO 9001

Contents

Appendix A. Restoring the Factory Configuration

Appendix B. Connecting to the Board via Ethernet

About this Manual

This manual provides details about the Nios® development board, Cyclone™ II Edition.Nios Development Board Cyclone II Edition

The table shows this document's revision history.

Date & Revision	Description
May 2007, 1.3	• Corrected Figure 1-1 and Figure 2-21. • Revised "How to Contact Altera". • Updated headers and footers.
October 2006, 1.2	• Corrected statement: LEDs D0 - D7 turn on when driven to 0, not 1. • Updated headers and footers.
June 2006, 1.1	• Updated part numbers to RoHS compliant parts • Corrected D7 pin information in LED pin table • Removed pin labels from J19 figure • Added J19 pin table • Changed PROTO1 and PROTO2 figures to use board net names • Added PROTO1 and PROTO2 pin tables • Corrected FPGA pin label for CON3 pin 9 in PMC Connector pin table • Added new pin AE15 to PMC Connector pin table • Added U69 pin table • Corrected factory config button figure • Added pin and device information and corrected net name for U3 Starting Configuration step 3 • Improved clock circuitry figure • Added clock signal pin tables
May 2005, 1.0	First publication.

How to Contact Altera

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

Contact (1)	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com

Contact *(1)*	Contact Method	Address
Product literature	Website	www.altera.com/literature
Altera literature services	Email	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

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

Typographic Conventions

This document uses the typographic conventions shown below.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, filenames, filename extensions, and software utility names are shown in bold type. Examples: **f$_{MAX}$**, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Design.*
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, *n* + 1. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: *<file name>*, *<project name>***.pof** file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
"Subheading Title"	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: "Typographic Conventions."
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input`. Active-low signals are denoted by suffix `n`, e.g., `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword `SUBDESIGN`), as well as logic function names (e.g., `TRI`) are shown in Courier.
1., 2., 3., and a., b., c., etc.	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ● •	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.

Visual Cue	Meaning
	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.
w	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.

Features Overview

The Nios Development Board, Cyclone II Edition, provides a hardware platform for developing embedded systems based on Altera® Cyclone II devices. The Nios Development Board, Cyclone II Edition provides the following features:

- Nios Development Board Cyclone II EditionA Cyclone II EP2C35F672C5 or EP2C35F672C5N FPGA with 33,216 logic elements (LE) and 483,840 bits of on-chip memory
- 16 MBytes of flash memory
- 2 MBytes of synchronous SRAM
- 32 MBytes of double data rate (DDR) SDRAM
- On-board logic for configuring the FPGA from flash memory
- On-board Ethernet MAC/PHY device and RJ45 connector
- Two 5.0 V-tolerant expansion/prototype headers each with access to 41 FPGA user I/O pins
- CompactFlash connector for Type I CompactFlash cards
- 32-bit PMC Connector capable of 33 MHz and 66 MHz operation
- Mictor connector for hardware and software debug
- RS-232 DB9 serial port
- Four push-button switches connected to FPGA user I/O pins
- Eight LEDs connected to FPGA 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 comes pre-programmed with a Nios II processor reference design. Hardware designers can use the reference design as an example of how to build systems using the Nios II processor and to gain familiarity with the features included. 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. A complete set of schematics, a physical layout database, and GERBER files for the development board are installed with the Nios II development tools in the *<Nios II EDS install path>*/**documents** directory.

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.

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

Figure 1–1. Nios Development Board, Cyclone II Edition Block Diagram
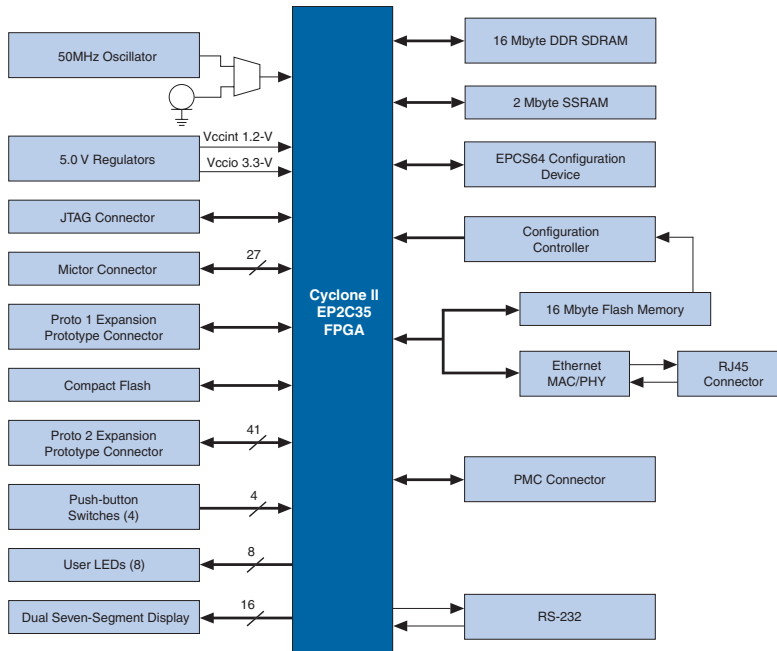
Factory-Programmed Reference Design

When power is applied to the board, on-board logic configures the FPGA using hardware configuration data stored in flash memory. After successful configuration, 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, refer to *Appendix B: Connecting to the Board via Ethernet*.

In the course of development, you might 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. Refer to *Appendix A: Restoring the Factory Configuration* for more information.

Component List

This section introduces all the important components on the Nios development board. See Figure 2–1 and Table 2–1 for component locations and brief descriptions of all board features.

Figure 2–1. Nios Development Board
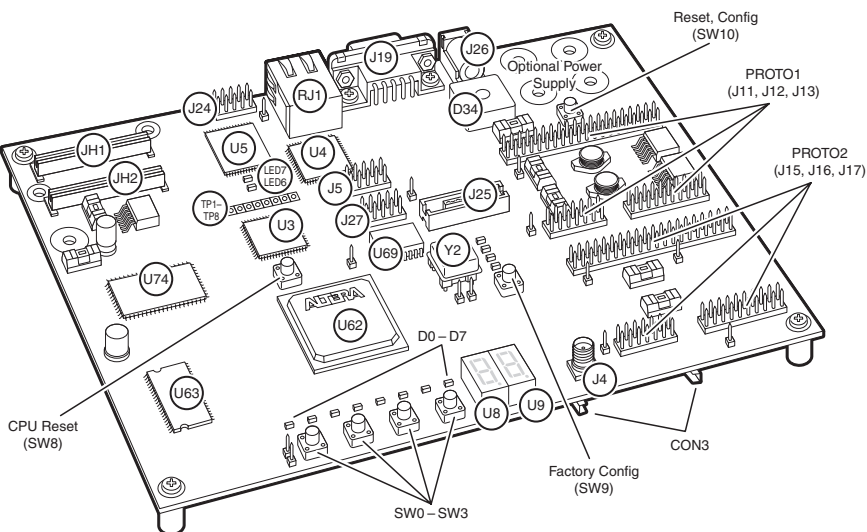
Table 2–1. Nios Development Board, Cyclone II Edition Components & Interfaces

Board Designation	Name	Description
U62	Cyclone II FPGA	EP2C35F672C5 or EP2C35F672C5N device.
User Interface		
SW0 – SW3	Push-button switches	Four momentary contact switches for user input to the FPGA.
D0 – D7	Individual LEDs	Eight individual LEDs driven by the FPGA.
U8, U9	Seven-segment LEDs	Two seven-segment LEDs that display numeric output from the FPGA.

Table 2–1. Nios Development Board, Cyclone II Edition Components & Interfaces (Continued)		
Board Designation	**Name**	**Description**
Memory		
U74	SSRAM memory	2 Mbytes of synchronous SRAM.
U5, LED7	Flash memory	16 Mbytes of nonvolatile memory for use by both the FPGA and the configuration controller. LED7 lights whenever the flash chip-enable is asserted.
U63	DDR SDRAM memory	32 Mbytes of DDR SDRAM.
Connections & Interfaces		
U4, RJ1	Ethernet MAC/PHY	10/100 Ethernet MAC/PHY chip connected to an RJ-45 Ethernet connector.
J19	Serial connector	RS-232 serial connector with 5 V-tolerant buffers. Supports all RS-232 signals.
PROTO1 (J11, J12, J13)	Expansion prototype connector	Expansion headers connecting to 41 I/O pins on the FPGA. Supplies 3.3V and 5.0V for use by a daughter card.
PROTO2 (J15, J16, J17)	Expansion prototype connector	Expansion headers connecting to 41 I/O pins on the FPGA. Supplies 3.3V and 5.0V for use by a daughter card.
CON3	CompactFlash connector	CompactFlash connector for memory expansion.
JH1, JH2	PMC connector	Expansion connector for a PCI mezzanine card.
J25	Mictor connector	Mictor connector providing access to 27 I/O pins on the FPGA. Allows debugging Nios II systems using a First Silicon Solutions (FS2) debug probe.
TP1 – TP8	Test Points	Test points providing access to eight FPGA I/O pins.
J24	JTAG connector	JTAG connection to the FPGA allowing hardware configuration using the Quartus® II software and software debug using the Nios II IDE.
J5	JTAG connector	JTAG connection to the MAX® configuration controller.
J27	EPCS configuration header	Connects to the EPCS serial configuration device for in-system programming.
Configuration & Reset		
U3	MAX Configuration controller	Altera MAX EPM7256AE device used to configure the FPGA from flash memory.
U69	Serial configuration device	Altera EPCS64 low-cost serial configuration device to configure the FPGA.
SW8	CPU Reset button	Push-button switch to reboot the Nios II processor configured in the FPGA.

Table 2–1. Nios Development Board, Cyclone II Edition Components & Interfaces (Continued)		
Board Designation	**Name**	**Description**
SW9	Factory Config button	Push-button switch to reconfigure the FPGA with the factory-programmed reference design.
SW10	Reset, Config	Push-button switch to reset the board.
LED0 – LED3, LED6	Configuration status LEDs	LEDs that display the current configuration status of the FPGA.
Clock Circuitry		
Y2	Oscillator	50 MHz clock signal driven to FPGA.
J4	External clock input	Connector to FPGA clock pin.
Power Supply		
J26	DC power jack	16V DC unregulated power source.
D34	Bridge rectifier	Power rectifier allows for center-negative or center-positive power supplies.
J28, J29, J30, J33 (and more)	Optional Power Supply	External power supply can be connected for high-current applications.

The sections that follow describe each component in detail.

Cyclone II EP2C35 Device (U62)

U62 is a Cyclone II FPGA in a 672-pin FineLine BGA® package. Depending on the board revision, the part number is EP2C35F672C5 or EP2C35F672C5N. Table 2–2 lists the device features.

Table 2–2. Cyclone II EP2C35 Device Features	
LEs	33,216
M4K Memory Blocks	105
Total RAM Bits	483,840
Embedded 18x18 Multiplier Blocks	35
PLLs	4
User I/O Pins	475

☞ Preproduction builds of the Nios Development Board, Cyclone II Editon have an EP2C35F6728ES device.

The development board provides two separate methods for configuring the FPGA:

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 FPGA JTAG header (J24).

2. When power is applied to the board, a configuration controller device (U3) attempts to configure the FPGA with hardware configuration data stored in flash memory. For more information on the configuration controller, refer to "Configuration Controller Device (U3)" on page 2–33.

For Cyclone II-related documentation including pin out data for the EP2C35 device, see the Altera Cyclone II literature page at **www.altera.com/literature/lit-cyc2.jsp**.

Push-Button Switches (SW0 - SW3)

SW0 – SW3 are momentary-contact push-button switches to provide stimulus to designs in the FPGA. Refer to Figure 2–2. Each switch is connected to an FPGA general-purpose I/O pin with a pull-up resistor as shown in Table 2–3. Each I/O pin perceives a logic 0 when its corresponding switch is pressed.

Figure 2–2. Push-Button Switches (SW0 – SW3)
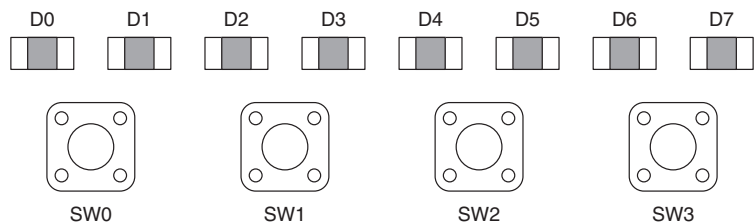
Table 2–3. Push Button Switches Pin Table		
Button	**FPGA Pin**	**Board Net Name**
SW0	Y11	user_pb0
SW1	AA10	user_pb1
SW2	AB10	user_pb2
SW3	AE6	user_pb3

Individual LEDs (D0 - D7)

This Nios development board provides eight individual LEDs connected to the FPGA. Refer to "Push-Button Switches (SW0 - SW3)" on page 2–4. D0 – D7 are connected to general purpose I/O pins on the FPGA as shown in Table 2–4. When a pin drives logic 0, the corresponding LED turns on.

Table 2–4. LED Pin Table

LED	FPGA Pin	Board Net Name
D0	AC10	pld_led0
D1	W11	pld_led1
D2	W12	pld_led2
D3	AE8	pld_led3
D4	AF8	pld_led4
D5	AE7	pld_led5
D6	AF7	pld_led6
D7	AA11	pld_led7

Seven-Segment LEDs (U8 & U9)

U8 and U9 connect to the FPGA, and each segment is individually controlled by a general-purpose I/O pin. Refer to Figure 2–3. When a pin drives logic 0, the corresponding U8 and U9 LED turns on. See Table 2–5 for pin-out details.

Figure 2–3. Dual Seven-Segment Display
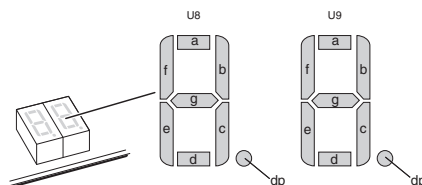
Table 2–5. Dual Seven-Segment Display			
FPGA Pin	**U8 & U9 Pin**	**Pin Function**	**Board Net Name**
U8			
AE13	10	a	hex_0A
AF13	9	b	hex_0B
AD12	8	c	hex_0C
AE12	5	d	hex_0D
AA12	4	e	hex_0E
Y12	2	f	hex_0F
V11	3	g	hex_0G
U12	7	dp	hex_0DP
U9			
V14	10	a	hex_1A
V13	9	b	hex_1B
AD11	8	c	hex_1C
AE11	5	d	hex_1D
AE10	4	e	hex_1E
AF10	2	f	hex_1F
AD10	3	g	hex_1G
AC11	7	dp	hex_1DP

SSRAM Chip (U74)

U74 is a 32-bit, 2 Mbyte Cypress SSRAM chip. Depending on the board revision, the part number is CY7C1380C-167AC or CY7C1380D-167AXC. The chip is rated for synchronous accesses up to 167 MHz. U74 connects to the FPGA so it can be used by a Nios II embedded processor as general-purpose memory. The factory-programmed Nios II reference design identifies the SSRAM devices in its address space as a contiguous 2 Mbyte, 32-bit-wide, zero-wait-state main memory.

Table 2–6 shows all connections between the FPGA and the SSRAM chip.

Table 2–6. SSRAM Pin Table			
FPGA Pin	**U74 Pin**	**Pin Function**	**Board Net Name**
AB3	37	A0	ssram_a0
AB4	36	A1	ssram_a1
G5	35	A2	ssram_a2
G6	34	A3	ssram_a3
B2	33	A4	ssram_a4
B3	32	A5	ssram_a5
C2	38	NC/A19	ssram_a6
C3	39	NC/A20	ssram_a7
L9	42	A6	ssram_a8
F7	43	A7	ssram_a9
L10	44	A8	ssram_a10
J5	45	A9	ssram_a11
L4	46	A10	ssram_a12
C6	47	A11	ssram_a13
A4	48	A12	ssram_a14
B4	49	A13	ssram_a15
A5	50	A14	ssram_a16
B5	81	A15	ssram_a17
B6	82	A16	ssram_a18
A6	99	A17	ssram_a19
C4	100	A18	ssram_a20
G9	85	ADSC_N	ssram_adsc_n
M3	93	BE_n0	ssram_be_n0
M2	94	BE_n1	ssram_be_n1
M4	95	BE_n2	ssram_be_n2
M5	96	BE_n3	ssram_be_n3
C7	98	CE1_n	ssram_ce1_n
L2	52	D0	ssram_d0
L3	53	D1	ssram_d1
L7	56	D2	ssram_d2
L6	57	D3	ssram_d3
N9	58	D4	ssram_d4

Table 2–6. SSRAM Pin Table (Continued)			
FPGA Pin	**U74 Pin**	**Pin Function**	**Board Net Name**
P9	59	D5	ssram_d5
K1	62	D6	ssram_d6
K2	63	D7	ssram_d7
K4	68	D8	ssram_d8
K3	69	D9	ssram_d9
J2	72	D10	ssram_d10
J1	73	D11	ssram_d11
H2	74	D12	ssram_d12
H1	75	D13	ssram_d13
J3	78	D14	ssram_d14
J4	79	D15	ssram_d15
H3	18	D24	ssram_d16
H4	19	D25	ssram_d17
G1	22	D26	ssram_d18
G2	23	D27	ssram_d19
F2	24	D28	ssram_d20
F1	25	D29	ssram_d21
K8	28	D30	ssram_d22
K7	29	D31	ssram_d23
G4	2	D16	ssram_d24
G3	3	D17	ssram_d25
K6	6	D18	ssram_d26
K5	7	D19	ssram_d27
E2	8	D20	ssram_d28
E1	9	D21	ssram_d29
J8	12	D22	ssram_d30
J7	13	D23	ssram_d31
D5	86	OE_n	ssram_oe_n
J9	87	WE_n	ssram_we_n
D7	84	ADSP_n	ssram_adsp_n
H10	83	ADV_n	ssram_adv_n
B7	97	CE2	ssram_ce2
A7	92	CE3_n	ssram_ce3_n

Table 2–6. SSRAM Pin Table (Continued)

FPGA Pin	U74 Pin	Pin Function	Board Net Name
K9	88	GW_n	ssram_gw_n
E5	89	CLK	sram_clk

The following pins on U74 have fixed connections, which restricts the usable modes of operation:

■ MODE is pulled low to enable Linear Burst
■ ZZ is pulled low to leave the chip enabled
■ GLOBALW_n is pulled high to disable the global write
■ CE2 and CE3_n are wired high and low respectively to be enabled and to make CE1_n the master chip enable

See **www.cypress.com** for detailed information about the SSRAM chip.

DDR SDRAM Chip (U63)

U63 is a Micron DDR SDRAM chip. Depending on the board revision, the part number is MT46V16M16TG or MT46V16M16P-6T. The DDR SDRAM pins are connected to the FPGA as shown in Table 2–7. Altera provides a DDR SDRAM controller that allows a Nios II processor to access the DDR SDRAM device as a large, linearly-addressable memory.

Table 2–7. DDR SDRAM Pin Table

FPGA Pin	U63 Pin	Board Net Name
R2	2	sdram_dq0
R3	4	sdram_dq1
R4	5	sdram_dq2
P7	7	sdram_dq3
P6	8	sdram_dq4
T2	10	sdram_dq5
T3	11	sdram_dq6
R6	13	sdram_dq7
W2	54	sdram_dq8
W1	56	sdram_dq9
U6	57	sdram_dq10
U7	59	sdram_dq11
U5	60	sdram_dq12
Y1	62	sdram_dq13

Table 2–7. DDR SDRAM Pin Table (Continued)		
FPGA Pin	**U63 Pin**	**Board Net Name**
V5	63	sdram_dq14
V6	65	sdram_dq15
P3	16	sdram_dqs0
W4	51	sdram_dqs1
U2	20	sdram_dm0
AA1	47	sdram_dm1
T6	29	sdram_a0
V2	30	sdram_a1
R8	31	sdram_a2
W3	32	sdram_a3
R5	35	sdram_a4
U10	36	sdram_a5
P4	37	sdram_a6
V1	38	sdram_a7
T9	39	sdram_a8
T8	40	sdram_a9
AA2	28	sdram_a10
T10	41	sdram_a11
U3	42	sdram_a12
U9	26	sdram_ba0
Y4	27	sdram_ba1
U1	22	sdram_cas_n
R7	44	sdram_cke
Y3	24	sdram_cs_n
V4	23	sdram_ras_n
U4	21	sdram_we_n
AA6	46	sdram_clk_n
AA7	45	sdram_clk_p

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

Flash Memory (U5)

U5 is an 8-bit, 16 Mbyte AMD flash memory device connected to the FPGA. Depending on the board revision, the part number is AM29LV128M or S29GL128M10TFIR1. Refer to Table 2–8 for connections between the FPGA and the flash memory chip. U5 can be used for two purposes:

1. A Nios II embedded processor implemented on the FPGA can use the flash memory as general-purpose memory and non-volatile storage.

2. The flash memory can hold FPGA configuration data that is used by the configuration controller to load the FPGA at power-up. Refer to "Configuration Controller Device (U3)" on page 2–33 for related information.

A Nios II processor design in the FPGA can identify the 16 Mbyte flash memory in its address space, and can program new data (either new FPGA configuration data, Nios II software, or both) into flash memory. The Nios II development software includes subroutines for writing and erasing flash memory.

☞ The flash memory device shares address and data connections with the Ethernet MAC/PHY device.

Table 2–8. Flash Memory Pin Table		
FPGA Pin	**U5 Pin**	**Board Net Name**
F9	51	fe_a0
H8	31	fe_a1
D11	26	fe_a2
E8	25	fe_a3
B14	24	fe_a4
A14	23	fe_a5
F14	22	fe_a6
G14	21	fe_a7
F13	20	fe_a8
G13	10	fe_a9
C15	9	fe_a10
B15	8	fe_a11
B16	7	fe_a12
C16	6	fe_a13
D15	5	fe_a14

Table 2–8. Flash Memory Pin Table (Continued)		
FPGA Pin	**U5 Pin**	**Board Net Name**
E15	4	fe_a15
H15	3	fe_a16
H16	54	fe_a17
A17	19	fe_a18
B17	18	fe_a19
G15	11	fe_a20
F15	12	fe_a21
F16	15	fe_a22
G16	2	fe_a23
D8	35	fe_d0
C8	37	fe_d1
F10	39	fe_d2
G10	41	fe_d3
D9	44	fe_d4
C9	46	fe_d5
B8	48	fe_d6
A8	50	fe_d7
H17	32	flash_cs_n
F17	34	flash_oe_n
G17	13	flash_rw_n
B18	16	flash_wp_n
C17	53	flash_byte_n (1)
D17	17	flash_ry_by_n

Note to Table 2–8:
(1)   BYTE_n on U5 is pulled low to keep the flash memory in byte mode which restricts the usable modes of operation.

The on-board configuration controller makes assumptions about what-resides-where in flash memory. For details refer to "SW10 – Reset, Config" on page 2–35.

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

Ethernet MAC/PHY (U4) & RJ45 Connector (RJ1)

The LAN91C111 chip (U4) is a 10/100 Ethernet media access control and physical interface (MAC/PHY) chip. The control pins of U4 are connected to the FPGA so that Nios II systems can access Ethernet networks via the RJ-45 connector (RJ1) as shown in Figure 2–4. The Nios II development tools include hardware and software components that allow Nios II processor systems to communicate with the LAN91C111 Ethernet device.

Figure 2–4. Ethernet RJ-45 Connector
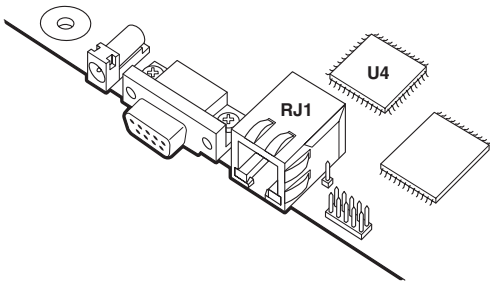
Refer to Table 2–9 for connections between the FPGA and the MAC/PHY device.

☞ The Ethernet MAC/PHY device shares both address and data connections with the flash memory.

Table 2–9. Ethernet MAC/PHY Pin Table

FPGA Pin	U4 Pin	Pin Function	Board Net Name (1)
E26	41	Address Enable	enet_aen
J17	43	Synchronous Ready	enet_srdy_n
F18	40	VL Bus Access	enet_vlbus_n
G18	45	Local Device	enet_ldev_n
D18	38	IO Char Ready	enet_iochrdy
E18	37	Address Strobe	enet_ads_n
A19	42	Local Bus Clock	enet_lclk
B19	46	Ready/Return	enet_rdyrtn_n
D20	35	Bus Cycle	enet_cycle_n
D14	36	Write/Read	enet_w_r_n
Y15	34	Bus Chip Select	enet_datacs_n
AA15	29	Interrupt	enet_intr0

Table 2–9. Ethernet MAC/PHY Pin Table (Continued)			
FPGA Pin	**U4 Pin**	**Pin Function**	**Board Net Name (1)**
C25	94	Byte Enable 0	enet_be_n0
C24	95	Byte Enable 1	enet_be_n1
D26	96	Byte Enable 2	enet_be_n2
D25	97	Byte Enable 3	enet_be_n3
E20	31	Read	enet_ior_n
D16	32	Write	enet_iow_n
H8	78	Address Line	fe_a1
D11	79	Address Line	fe_a2
E8	80	Address Line	fe_a3
B14	81	Address Line	fe_a4
A14	82	Address Line	fe_a5
F14	83	Address Line	fe_a6
G14	84	Address Line	fe_a7
F13	85	Address Line	fe_a8
G13	86	Address Line	fe_a9
C15	87	Address Line	fe_a10
B15	88	Address Line	fe_a11
B16	89	Address Line	fe_a12
C16	90	Address Line	fe_a13
D15	91	Address Line	fe_a14
E15	92	Address Line	fe_a15
D8	107	Data Line	fe_d0
C8	106	Data Line	fe_d1
F10	105	Data Line	fe_d2
G10	104	Data Line	fe_d3
D9	102	Data Line	fe_d4
C9	101	Data Line	fe_d5
B8	100	Data Line	fe_d6
A8	99	Data Line	fe_d7
H11	76	Data Line	fe_d8
H12	75	Data Line	fe_d9
F11	74	Data Line	fe_d10
E10	73	Data Line	fe_d11
B9	71	Data Line	fe_d12

Table 2–9. Ethernet MAC/PHY Pin Table (Continued)			
FPGA Pin	**U4 Pin**	**Pin Function**	**Board Net Name (1)**
A9	70	Data Line	fe_d13
C10	69	Data Line	fe_d14
D10	68	Data Line	fe_d15
B10	66	Data Line	fe_d16
A10	65	Data Line	fe_d17
E12	64	Data Line	fe_d18
D12	63	Data Line	fe_d19
J13	61	Data Line	fe_d20
J14	60	Data Line	fe_d21
F12	59	Data Line	fe_d22
G12	58	Data Line	fe_d23
J10	56	Data Line	fe_d24
J11	55	Data Line	fe_d25
C11	54	Data Line	fe_d26
B11	53	Data Line	fe_d27
C12	51	Data Line	fe_d28
B12	50	Data Line	fe_d29
D6	49	Data Line	fe_d30
G11	48	Data Line	fe_d31
Note to Table 2–9: (1) Nets fe_a0 and fe_a16 to fe_a23 do not connect to U4.			

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

Serial Connector (J19)

J19 is a standard DB-9 serial connector, and is typically used for communication between the FPGA and a host computer via an RS-232 serial cable. Level-shifting buffer (U52) is used between J19 and the FPGA, because the FPGA device cannot interface to RS-232 voltage levels directly.

J19 is able to transmit all RS-232 signals. Alternately, the FPGA design can use only the signals it needs, such as J19's RXD and TXD pins. LEDs are connected to the RXD and TXD signals and visually indicate when data is being transmitted or received. Figure 6 and Table 2–10 show the pin connections between the serial connectors and the FPGA.

Figure 2–5. Serial Connector J19
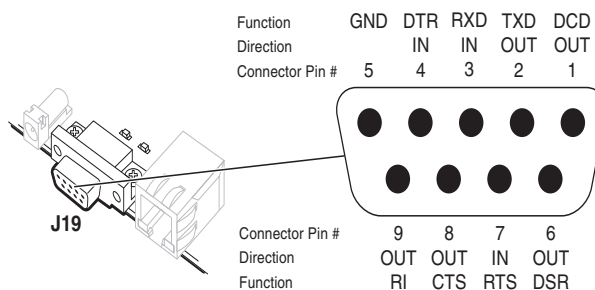
Table 2–10. Serial Connector Pin Table

FPGA Pin	J19 Pin	Board Net Name
AB15	3	serial_rxd
J22	2	serial_txd
H21	4	serial_dtr
K22	1	serial_dcd
H19	6	serial_dsr
L19	9	serial_ri
L23	8	serial_cts
AC15	7	serial_rts

Expansion Prototype Connectors (PROTO1 & PROTO2)

PROTO1 and PROTO2 are standard-footprint, mechanically-stable connections that can be used (for example) as an interface to a special-function daughter card. Headers J11, J12, and J13 collectively form PROTO1, and J15, J16 and J17 collectively form PROTO2.

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 FPGA. Each signal passes through analog switches to protect the FPGA from 5V logic levels. These analog switches are permanently enabled. The output logic-level on the expansion prototype connector pins is 3.3V.

● PROTO1 switches: U19, U20, U21, U22 and U25
● PROTO2 switches: U27, U28, U29, U30 and U31

■ A buffered, zero-skew copy of the on-board oscillator output from U2.
■ A buffered, zero-skew copy of the FPGA phase-locked loop (PLL) output.
■ A logic-negative power-on reset signal.
■ Five regulated 3.3V power-supply pins (2A total max load for both PROTO1 & PROTO2).
■ One regulated 5V power-supply pin (1A total max load for both PROTO1 & PROTO2).
■ Numerous ground connections.

The PROTO1 expansion prototype connector shares FPGA I/O pins with the CompactFlash connector (CON3). Designs can use either the PROTO1 connector or the CompactFlash connector.

☞ Do not connect cards to PROTO1 and CON3 at the same time. Damage to one or both cards might result.

👣 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.**

Table 2–11, Figure 2–6 and Figure 2–7 show connections from the PROTO1 expansion headers to the FPGA. Unless otherwise noted, labels indicate FPGA pin numbers...

Table 2–11. PROTO1 Pin Table			
FPGA Pin	**PROTO1 Pin**	**Connector**	**Board Net Name**
J11			
U3 pin 56	1	J11	proto1_RESET_n
E25	3	J11	proto1_io0
F24	4	J11	proto1_io1
F23	5	J11	proto1_io2
J21	6	J11	proto1_io3
J20	7	J11	proto1_io4
F25	8	J11	proto1_io5
F26	9	J11	proto1_io6
N18	10	J11	proto1_io7
P18	11	J11	proto1_io8
G23	12	J11	proto1_io9
G24	13	J11	proto1_io10

Table 2–11. PROTO1 Pin Table (Continued)			
FPGA Pin	**PROTO1 Pin**	**Connector**	**Board Net Name**
G25	14	J11	proto1_io11
G26	15	J11	proto1_io12
H23	16	J11	proto1_io13
H24	17	J11	proto1_io14
J23	18	J11	proto1_io15
J24	21	J11	proto1_io16
H25	23	J11	proto1_io17
H26	25	J11	proto1_io18
K18	27	J11	proto1_io19
K19	28	J11	proto1_io20
K23	29	J11	proto1_io21
K24	31	J11	proto1_io22
J25	32	J11	proto1_io23
J26	33	J11	proto1_io24
M21	35	J11	proto1_io25
T23	36	J11	proto1_io26
R17	37	J11	proto1_io27
K21	38	J11	proto1_cardsel_n
P17	39	J11	proto1_io28
J12			
Y22	3	J12	proto1_io40
T18	4	J12	proto1_io29
T17	5	J12	proto1_io30
U26	6	J12	proto1_io31
R19	7	J12	proto1_io32
T19	8	J12	proto1_io33
U20	9	J12	proto1_io34
U21	10	J12	proto1_io35
V26	11	J12	proto1_io36
V25	12	J12	proto1_io37
V24	13	J12	proto1_io38
V23	14	J12	proto1_io39
J13			
U2 pin 19	9	J13	proto1_osc

Table 2–11. PROTO1 Pin Table (Continued)			
FPGA Pin	**PROTO1 Pin**	**Connector**	**Board Net Name**
F21	11	J13	proto1_pllclk
N26	13	J13	proto1_clkout

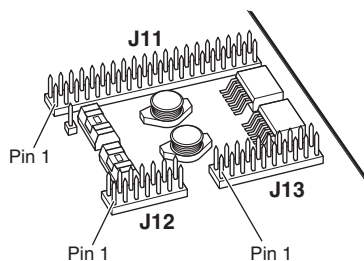*Figure 2–6. PROTO1 Expansion Prototype Connector - J11, J12 & J13*

Figure 2–7. PROTO1 Pin Information – J11, J12, & J13

J11

proto1_RESET_n	1	○ ●	2	GND
proto1_io0	3	○ ○	4	proto1_io1
proto1_io2	5	○ ○	6	proto1_io3
proto1_io4	7	○ ○	8	proto1_io5
proto1_io6	9	○ ○	10	proto1_io7
proto1_io8	11	○ ○	12	proto1_io9
proto1_io10	13	○ ○	14	proto1_io11
proto1_io12	15	○ ○	16	proto1_io13
proto1_io14	17	○ ○	18	proto1_io15
GND	19	● ○	20	NC
proto1_io16	21	○ ●	22	GND
proto1_io17	23	○ ●	24	GND
proto1_io18	25	○ ●	26	GND
proto1_io19	27	○ ○	28	proto1_io20
proto1_io21	29	○ ●	30	GND
proto1_io22	31	○ ○	32	proto1_io23
proto1_io24	33	○ ○	34	NC
proto1_io25	35	○ ○	36	proto1_io26
proto1_io27	37	○ ○	38	proto1_cardsel_n
proto1_io28	39	○ ●	40	GND

J12

GND	1	● ●	2	VCC5
proto1_io40	3	○ ○	4	proto1_io29
proto1_io30	5	○ ○	6	proto1_io31
proto1_io32	7	○ ○	8	proto1_io33
proto1_io34	9	○ ○	10	proto1_io35
proto1_io36	11	○ ○	12	proto1_io37
proto1_io38	13	○ ○	14	proto1_io39

J13

(1) Vunreg	1	● ●	2	GND
NC	3	○ ●	4	GND
VCC3_3	5	● ●	6	GND
VCC3_3	7	● ●	8	GND
(2) proto1_osc	9	○ ●	10	GND
(3) proto1_pllclk	11	○ ●	12	GND
(4) proto1_clkout	13	○ ●	14	GND
VCC3_3	15	● ●	16	GND
VCC3_3	17	● ●	18	GND
VCC3_3	19	● ●	20	GND

Notes to Figure 2–7:
(1) Unregulated voltage from DC power supply.
(2) Clk from board oscillator.
(3) Clk from FPGA.
(4) Clk output from PROTO1 card to FPGA.

Table 2–12, Figure 2–8 and Figure 2–9 show connections from the PROTO2 expansion headers to the FPGA. Unless otherwise noted, the labels indicate FPGA pin numbers.

FPGA Pin	PROTO2 Pin	Connector	Board Net Name
Table 2–12. PROTO2 Pin Table			
J16			
U3 pin 57	1	J16	proto2_RESET_n
AE24	3	J16	proto2_io0
T21	4	J16	proto2_io1
V22	5	J16	proto2_io2
AF23	6	J16	proto2_io3
AE23	7	J16	proto2_io4
AC22	8	J16	proto2_io5
AB21	9	J16	proto2_io6
AD23	10	J16	proto2_io7
AD22	11	J16	proto2_io8
AC21	12	J16	proto2_io9
AD21	13	J16	proto2_io10
AF22	14	J16	proto2_io11
AE22	15	J16	proto2_io12
V18	16	J16	proto2_io13
W19	17	J16	proto2_io14
U17	18	J16	proto2_io15
U18	21	J16	proto2_io16
AF21	23	J16	proto2_io17
AE21	25	J16	proto2_io18
AB20	27	J16	proto2_io19
AC20	28	J16	proto2_io20
AF20	29	J16	proto2_io21
AE20	31	J16	proto2_io22
AD19	32	J16	proto2_io23
AC19	33	J16	proto2_io24
AA17	35	J16	proto2_io25
AA18	36	J16	proto2_io26
W17	37	J16	proto2_io27
AA20	38	J16	proto2_cardsel_n

Table 2–12. PROTO2 Pin Table (Continued)

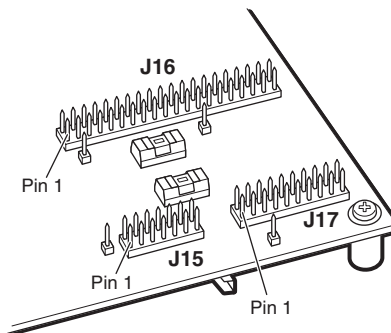| FPGA Pin	PROTO2 Pin	Connector	Board Net Name
V17	39	J16	proto2_io28
J15			
AE17	3	J15	proto2_io40
AB18	4	J15	proto2_io29
AC18	5	J15	proto2_io30
AF19	6	J15	proto2_io31
AE19	7	J15	proto2_io32
AF18	8	J15	proto2_io33
AE18	9	J15	proto2_io34
AA16	10	J15	proto2_io35
Y16	11	J15	proto2_io36
AC17	12	J15	proto2_io37
AD17	13	J15	proto2_io38
AF17	14	J15	proto2_io39
J17			
U2 pin 18	9	J17	proto2_osc
F20	11	J17	proto2_pllclk
AF14	13	J17	proto2_clkout

Figure 2–8. PROTO2 Expansion Prototype Connector - J15, J16 & J17

Figure 2–9. PROTO2 Pin Information – J15, J16 & J17

J16

proto2_RESET_n	1	○ ●	2	GND
proto2_io0	3	○ ○	4	proto2_io1
proto2_io2	5	○ ○	6	proto2_io3
proto2_io4	7	○ ○	8	proto2_io5
proto2_io6	9	○ ○	10	proto2_io7
proto2_io8	11	○ ○	12	proto2_io9
proto2_io10	13	○ ○	14	proto2_io11
proto2_io12	15	○ ○	16	proto2_io13
proto2_io14	17	○ ○	18	proto_2io15
GND	19	● ○	20	NC
proto2_io16	21	○ ●	22	GND
proto2_io17	23	○ ●	24	GND
proto2_io18	25	○ ●	26	GND
proto2_io19	27	○ ○	28	proto2_io20
proto2_io21	29	○ ●	30	GND
proto2_io22	31	○ ○	32	proto2_io23
proto2_io24	33	○ ○	34	NC
proto2_io25	35	○ ○	36	proto2_io26
proto2_io27	37	○ ○	38	proto2_cardsel_n
proto2_io28	39	○ ●	40	GND

J15

GND	1	● ●	2	VCC5
proto2_io40	3	○ ○	4	proto2_io29
proto2_io30	5	○ ○	6	proto2_io31
proto2_io32	7	○ ○	8	proto2_io33
proto2_io34	9	○ ○	10	proto2_io35
proto2_io36	11	○ ○	12	proto2_io37
proto2_io38	13	○ ○	14	proto2_io39

J17

(1) Vunreg	1	● ●	2	GND
NC	3	○ ●	4	GND
VCC3_3	5	● ●	6	GND
VCC3_3	7	● ●	8	GND
(2) proto2_osc	9	○ ●	10	GND
(3) proto2_pllclk	11	○ ●	12	GND
(4) proto2_clkout	13	○ ●	14	GND
VCC3_3	15	● ●	16	GND
VCC3_3	17	● ●	18	GND
VCC3_3	19	● ●	20	GND

Notes to Figure 2–9:
(1) Unregulated voltage from DC power supply.
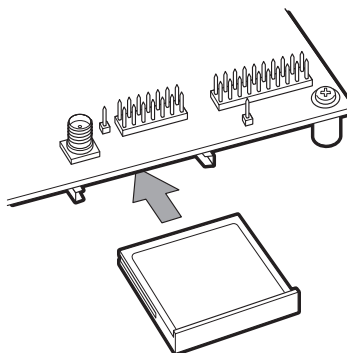(2) Clk from board oscillator.
(3) Clk from FPGA.
(4) Clk output from PROTO2 card to FPGA.

CompactFlash Connector (CON3)

The CompactFlash connector header (CON3) enables hardware designs to access a CompactFlash card. Refer to Figure 2–10. The following two access modes are supported:

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

Figure 2–10. CompactFlash Connector

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

■ Pin 13 and 38 of CON3 (VCC) are 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 EPM7256AE 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 FPGA I/O pins with expansion prototype connector PROTO1. Refer to "Expansion Prototype Connectors (PROTO1 & PROTO2)" on page 2–16 for details on PROTO1.

☞ Do not connect cards to PROTO1 and CON3 at the same time. Damage to one or both cards might result.

Table 2–13 lists connections between CON3 and the FPGA.

Table 2–13. CompactFlash Pin Table			
FPGA Pin	**CON3 Pin**	**Pin Function**	**Board Net Name (1)**
E25	6	D7	proto1_io0
F24	47	D8	proto1_io1

Table 2–13. CompactFlash Pin Table (Continued)			
FPGA Pin	**CON3 Pin**	**Pin Function**	**Board Net Name** (1)
F23	5	D6	proto1_io2
J21	48	D9	proto1_io3
J20	4	D5	proto1_io4
F25	49	D10	proto1_io5
F26	3	D4	proto1_io6
N18	27	D11	proto1_io7
P18	2	D3	proto1_io8
G23	28	D12	proto1_io9
G24	23	D2	proto1_io10
G25	29	D13	proto1_io11
G26	22	D1	proto1_io12
H23	30	D14	proto1_io13
H24	21	D0	proto1_io14
J23	31	D15	proto1_io15
H25	35	IOWR_n	proto1_io17
H26	34	IORD_n	proto1_io18
K18	42	IORDY_n	proto1_io19
K24	37	INTRQ	proto1_io22
J25	24	IOCS16_n	proto1_io23
J26	19	A1	proto1_io24
M21	20	A0	proto1_io25
T23	18	A2	proto1_io26
R17	7	CS0_n	proto1_io27
P17	45	DASP	proto1_io28
T18	8	A10	proto1_io29
T17	46	PDIAG	proto1_io30
U26	10	A9	proto1_io31
R19	11	A8	proto1_io32
T19	12	A7	proto1_io33
U20	14	A6	proto1_io34
U21	15	A5	proto1_io35
V26	16	A4	proto1_io36
V25	17	A3	proto1_io37
V24	36	WE_n	proto1_io38

Table 2–13. CompactFlash Pin Table (Continued)

FPGA Pin	CON3 Pin	Pin Function	Board Net Name (1)
V23	43	INPACK_n	proto1_io39
Y22	44	REG_n	proto1_io40
W16	32	CS1_n	cf_cs_n
AE16	9	ATA_SEL_n	cf_atasel_n
AD16	5	Power supply enable	cf_power (2)
W15	26	CD1_n	cf_present_n
56 (U3)	41	RESET#	proto1_RESET_n (3)

Notes to Table 2–13:
(1)   Nets proto_io16, proto_io20, and proto_io21 do not connect to CON3.
(2)   The FPGA I/O pin controls a power MOSFET that supplies 5V VCC to this net.
(3)   proto1_RESET_n is driven by the EPM7256AE configuration controller device (U3).

For more information on the CompactFlash connector (CON3), see **www.compactflash.org** and **www.molex.com**.
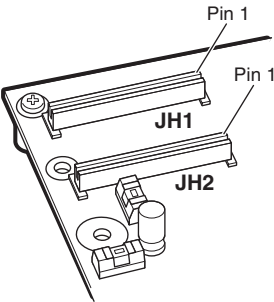
PMC Connector (JH1 & JH2)

The PCI mezzanine card (PMC) connector, formed by JH1 and JH2, allows Nios II systems in the FPGA to interface to daughter cards using the standard 32-bit PMC form factor. Refer to Figure 2–11. The PMC connector is capable of 33MHz and 66 MHz, and is configured as the PMC host.

w   Before connecting a daughter card to the PMC connector, the FPGA must first be configured with a design that includes a PMC interface. Damage to either the FPGA or daughter card can result if the FPGA is not configured correctly.

The factory-programmed Nios II reference design does not include a PMC interface.

Figure 2–11. PMC Connector

The PMC connector supplies +3.3V, +5V and +/- 12V, as required by the PCI specification. However, DC power regulators for these supplies cannot provide enough power to fully satisfy the PCI power specification. The current that the board can supply through JH1 and JH2 is dependent on the design configured in the FPGA. As a general guideline, if the PMC card power requirements exceed the specifications shown in Table 2–14, you must connect an external power source.

Table 2–14. PMC Card Power Specifications

DC Supply	Maximum Power	Apply External Power Source
+3.3V	9.5 Watts	J29
+5V	15 Watts	J28
+12V	45 Watts	J31
-12V	1.2 Watts	TP13

w   When connecting an external power supply, the fuse for the corresponding voltage should be removed to prevent the two power supplies from interfering with each other. Refer to "Power-Supply Circuitry" on page 2–44 for more information.

Table 2–15 lists the connections between the PMC connector and the FPGA.

Table 2–15. PMC Connector Pin Table

FPGA Pin	JH1 & JH2 Pin	Connector	Board Net Name
L20	61	JH1	pmc_ad0
L21	60	JH1	pmc_ad1

Table 2–15. PMC Connector Pin Table (Continued)			
FPGA Pin	**JH1 & JH2 Pin**	**Connector**	**Board Net Name**
L24	59	JH1	pmc_ad2
L25	58	JH1	pmc_ad3
M19	55	JH1	pmc_ad4
M22	54	JH1	pmc_ad5
M23	53	JH1	pmc_ad6
R24	51	JH2	pmc_ad7
U22	49	JH2	pmc_ad8
U25	49	JH1	pmc_ad9
W21	48	JH2	pmc_ad10
W23	48	JH1	pmc_ad11
W24	47	JH1	pmc_ad12
W25	46	JH2	pmc_ad13
Y21	45	JH2	pmc_ad14
Y23	46	JH1	pmc_ad15
Y24	31	JH2	pmc_ad16
Y25	32	JH1	pmc_ad17
Y26	29	JH2	pmc_ad18
AA23	29	JH1	pmc_ad19
AA24	28	JH2	pmc_ad20
AA25	28	JH1	pmc_ad21
AA26	27	JH1	pmc_ad22
AB23	26	JH2	pmc_ad23
AB24	23	JH2	pmc_ad24
AB25	23	JH1	pmc_ad25
AB26	22	JH2	pmc_ad26
AC23	22	JH1	pmc_ad27
AC25	21	JH1	pmc_ad28
AC26	20	JH2	pmc_ad29
AD24	19	JH2	pmc_ad30
AD25	20	JH1	pmc_ad31
R20	52	JH1	pmc_be_n0
T22	43	JH2	pmc_be_n1
T24	32	JH2	pmc_be_n2
T25	26	JH1	pmc_be_n3

Table 2–15. PMC Connector Pin Table (Continued)			
FPGA Pin	**JH1 & JH2 Pin**	**Connector**	**Board Net Name**
T20	43	JH1	pmc _par
W26	13	JH1	pmc_clk
U24	39	JH2	pmc_perr_n
U23	42	JH2	pmc_serr_n
R25	37	JH1	pmc_devsel_n
P24	38	JH2	pmc_stop_n
P23	36	JH1	pmc_irdy_n
M20	4	JH1	pmc_inta_n
Y14	5	JH1	pmc_intb_n
AA13	6	JH1	pmc_intc_n
Y13	9	JH1	pmc_intd_n
G22	13	JH2	pmc_reset_n
N24	33	JH1	pmc_frame_n
N23	35	JH2	pmc_trdy_n
M25	25	JH2	pmc_idsel
M24	16	JH1	pmc_gnt_n
N20	17	JH1	pmc_req_n
K26	40	JH1	pmc_lock_n
K25	47	JH2	pmc_m66en
AE15	64	JH1	pmc_req64_n

Mictor Connector (J25)

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 can be used as a debug port for the Nios II processor or as a general-purpose I/O connector to the FPGA. 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 FPGA (U62). 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. 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 2–12 shows an example of an in-target system analyzer ISA-Nios/T by First Silicon Solutions (FS2) Inc. connected to the Mictor connector. For details, see **www.fs2.com**.

Figure 2–12. An ISA-Nios/T Connecting to the Mictor Connector (J25)
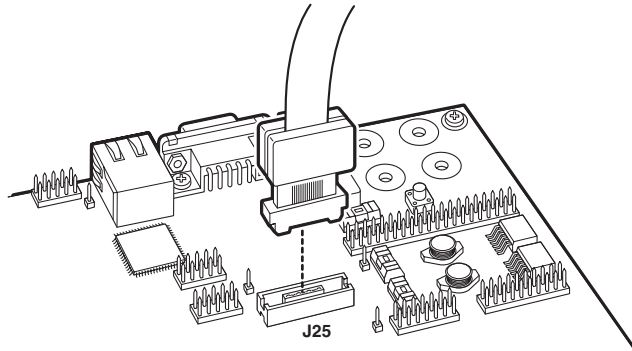
Five of the signals connect to both the JTAG pins on the FPGA (U62), and the FPGA's JTAG connector (J24). The JTAG signals have special usage requirements. J25 and J24 cannot be used at the same time.

Figure 2–13 below shows connections from the Mictor connector to the FPGA.

Figure 2–13. Mictor Connector Signaling
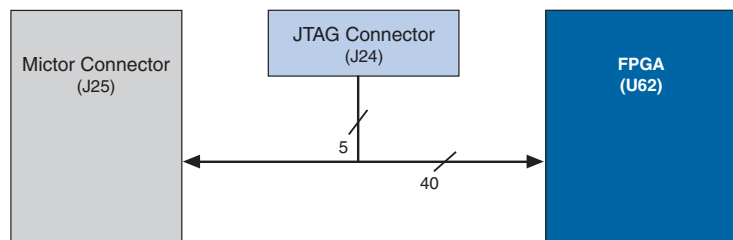
Table 2–16 shows the pin out information for J25. Unless otherwise noted, labels indicate FPGA pin numbers.

Table 2–16. Mictor Connector Pin Table		
FPGA Pin	**J25 Pin**	**Board Net Name**
V21	5	mictor_clk
AC8	38	mictor0
AD8	36	mictor1
W10	34	mictor2
Y10	32	mictor3
V10	30	mictor4
V9	28	mictor5
AD6	26	mictor6
AD7	24	mictor7
AE5	22	mictor8
AF5	20	mictor9
AD4	18	mictor10
AD5	16	mictor11
AC5	10	mictor12
AC6	8	mictor13
AF4	37	mictor14
AE4	35	mictor15
B21	33	mictor16
B22	31	mictor17
A22	29	mictor18
A23	27	mictor19
B23	25	mictor20
D21	23	mictor21
C21	13	mictor22
C22	9	mictor23
C23	7	mictor24
B25	6	mictor_trclk

Test Points (TP1 – TP8)

TP1 – TP8 are test points connected to I/O pins on the FPGA. FPGA designs can route signals to these I/O pins to be probed. TP1 –TP8 also connect to the configuration controller (U3).

Table 2–17 lists the connections between the FPGA, U3, and the test points.

Table 2–17. Test Point Pin Table

Test Point	FPGA Pin	CPLD Pin	Board Net Name
TP1	D19	75	pld_user0
TP2	C19	76	pld_user1
TP3	A20	77	pld_user2
TP4	B20	78	pld_user3
TP5	K16	79	pld_user4
TP6	J16	80	pld_user5
TP7	K17	81	pld_user6
TP8	J18	83	pld_user7

EPCS64 Serial Configuration Device (U69)

U69 is a serial configuration device connected to the FPGA. Serial configuration devices are flash memory devices with a serial interface which can store configuration data, and load the data into the FPGA upon power up or reconfiguration. U69 can store FPGA configuration data, or Nios II program data, or both.

Table 2–18 lists the connections between U69 and the FPGA.

Table 2–18. EPCS64 Pin Table

FPGA Pin	U69 Pin	Board Net Name
N6	16	pld_dclk
N3	8	pld_data0
D3	7	pld_cs_n
E3	15	pld_asdo

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

U69 is blank by default. The Quartus II software can program FPGA configuration data (a **.pof** file) into U69 through an Altera download cable connected to J27. Alternately, software running on a Nios II processor design can write configuration data to U69.

☞ The orientation of J27 is the reverse of J24.

See the *Serial Configuration Devices* chapter in Altera's *Configuration Device Handbook* for more information about the EPCS64 device. See the *EPCS Device Controller Core with Avalon Interface* chapter in the *Quartus II Handbook, Volume 5: Altera Embedded Peripherals* for information about the EPCS serial flash controller component in SOPC Builder.

Configuration Controller Device (U3)

The configuration controller (U3) is an Altera MAX 7000 EPM7256AE device. It comes preprogrammed with logic for managing board reset conditions and configuring the FPGA from data stored in flash memory and the EPCS64 serial configuration device (U69).

FPGA configuration data files are generated by the Quartus II software. The Nios II integrated development environment (IDE) can write new configuration data to the board's flash memory.

For complete details on the configuration controller connections, see the board schematic. For detailed information about the Altera EPM7256AE device, see the MAX 7000 family literature at **www.altera.com/literature/lit-m7k.html.** For details on programming configuration data to flash memory, see the *Nios II Flash Programmer User Guide*, or refer to the Nios II IDE help system.

Configuration-Status LEDs

The configuration controller is connected to four status LEDs that show the configuration status of the board at a glance as shown in Figure 2–14. The LEDs indicate which configuration, if any, was loaded into the FPGA at power-on as shown in Table 2–19.
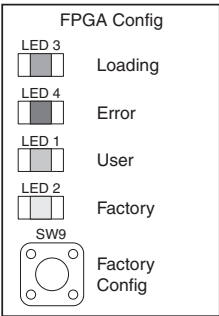
Figure 2–14. LED1 – LED 4

Table 2–19. 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 FPGA.
LED4	Error	Red	If this LED is on, then configuration was not transferred from flash memory into the FPGA. This can happen if, for example, the flash memory contains either a valid user or factory configuration.
LED1	User	Green	This LED turns on when the user configuration is being transferred from flash memory and stays illuminated when the user configuration data is successfully loaded into the FPGA. If the FPGA was successfully configured by the EPCS64, LED1 will blink slowly.
LED2	Factory	Amber	This LED turns on when the factory configuration is being transferred from flash memory and stays illuminated if the factory configuration was successfully loaded into the FPGA.
LED6	LED6	Red	This LED is an indicator of the CONFIG_DONE_signal from the FPGA. This LED illuminates when FPGA configuration completes successfully and CONFIG_DONE goes high.
LED7	LED7	Red	This LED is an indicator of the flash_CE_n line. It illuminates when the flash is being accessed and the CE_n line is being asserted.

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 FPGA I/O pin C5 (DEV_CLRn). The result of pressing SW8 depends on how the FPGA is configured. Refer to Figure 2–15.

The factory-programmed Nios II reference design treats SW8 as a CPU-reset button. The Nios II reference design resets and starts executing code from its reset address when SW8 is pressed.

*Figure 2–15. CPU Reset Button*
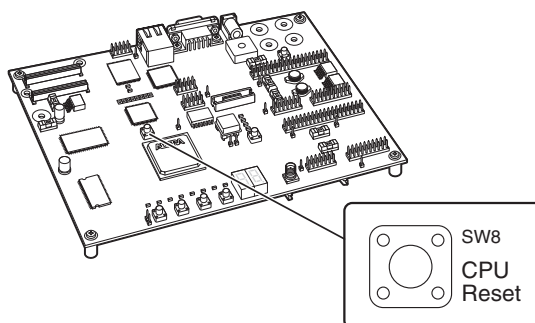
SW9 – Factory Config

Pressing Factory Config (SW9) commands the configuration controller to reconfigure the FPGA with the factory configuration. Refer to Figure 2–16.

Figure 2–16. Factory Config Button
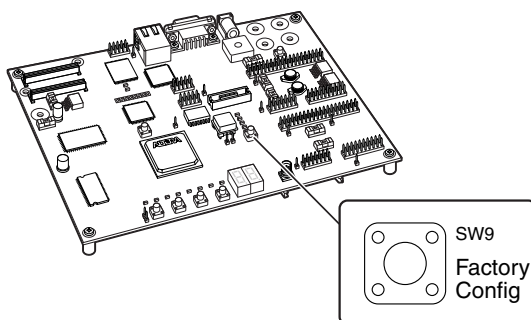
SW10 – Reset, Config

Reset, Config (SW10) is the power-on reset button. Refer to Figure 2–17. When SW10 is pressed, a logic 0 is driven to the power on reset controller (U18). Refer to "Power-Supply Circuitry" on page 2–44 for more details. Whenever SW10 is pressed, the configuration controller attempts to reconfigure the FPGA.

Figure 2–17. Reset, Config Button
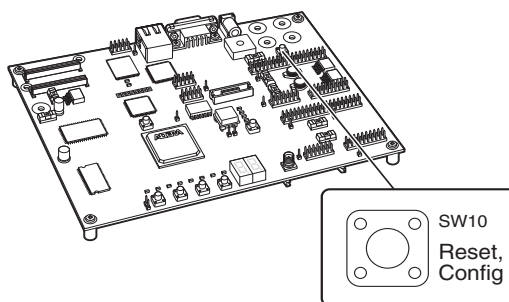
Reset Distribution

The EPM7256AE device takes a power-on reset pulse from the Linear Technologies 1326 power-sense/reset-generator chip (U18) and distributes it (through internal logic) to other reset pins on the board, that include the following:

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

Starting Configuration

The following four methods start a configuration sequence:

1. Board power-on

2. Pressing the Reset, Config button (SW10).

3. Asserting (driving 0 volts on) the `pld_reconfigreq_n` input pin of the EPM7256AE device (U3 pin 94) from the FPGA (U62 pin AA14).

4. Pressing the Factory Config button (SW9).

Factory & User Configurations

The configuration controller can manage two separate FPGA configurations stored in flash memory U5. These two configurations are referred to as the factory configuration and the user configuration. A

Nios II reference design is factory-programmed into the factory configuration region of the flash memory. In addition, the FPGA can be configured by the EPCS64 serial configuration device.

Configuration Process

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

1. The EPCS64 serial configuration device

2. The user configuration from flash memory

3. The factory configuration from flash memory

First, the configuration controller puts the FPGA in active serial (AS) configuration mode. The FPGA then attempt to read configuration data from the EPCS64. If the FPGA finishes configuration successfully, the configuration controller stops.

If configuration from the EPCS64 does not succeed, the configuration controller puts the FPGA into passive serial (PS) mode and attempts to load the user configuration from flash memory. If this also fails (because the user configuration is either invalid or not present), the configuration controller attempts to load the factory configuration from flash memory.

When SW9 (Factory Config) is pressed, the configuration controller ignores the user configuration and EPCS64, and configures the FPGA with the factory configuration. SW9 provides an escape from a situation in which a valid-but-nonfunctional design is present in user flash memory or the EPCS64.

Flash Memory Partitions

The configuration controller expects user and factory configuration data to be stored at fixed locations (offsets) in flash memory. In addition, the factory-programmed reference design expects Nios II software and data to exist at certain locations in flash memory. Table 2–20 shows the expected flash memory partitioning.

Table 2–20. Flash Memory Partitions		
Offset	**Usage**	**Factory-Programmed Content**
0x00000000 – 0x00BFFFFF	User Application Space (11 MB)	
0x00C00000 – 0x00CFFFFF	User Configuration (1 MB)	
0x00D00000 –0x00DFFFFFF	Free Space (1 MB)	
0x00E00000 – 0x00EFFFFF	Factory Configuration (1 MB)	Nios II Processor Reference Design
0x00F00000 – 0x00FAFFFF	Web Pages	Web Pages
0x00FB0000 – 0x00FEFFFF	Web Server Software	Web Server Software
0x00FF0000 – 0x00FFFFFF	Persistent Data	Network Settings for Web Server

☞ This partitioning scheme is merely a convention used by the configuration controller and the factory-programmed reference design. Custom FPGA designs can use the flash memory space in any way necessary.

⚠ CAUTION   Altera recommends that you do not overwrite the factory-programmed flash memory contents. Without a valid factory configuration, the configuration controller may not be able to successfully configure the FPGA. If you alter the factory configuration, you can restore the board to its factory-programmed state. Refer to *Appendix B: Restoring the Factory Configuration.*

User Application Space

The lower 11 MB of flash memory is the user application space. This is free space for user designs to store code and data for Nios II programs. The Nios II IDE allows you to compile Nios II programs and program them into the user application space.

User Configuration

The user configuration partition is 1 MB, starting at offset 0x00C00000. This section contains the FPGA configuration data for the user configuration. Nios II development tools include documentation on how to create your own user configuration image and program it into flash memory.

Factory Configuration

The factory configuration partition is 1 MB, starting at offset 0x00E00000. This section contains the FPGA configuration data for the factory configuration. The Nios II processor system in the factory configuration is designed to start executing code from offset 0x00000000 in the flash memory. The Nios II development tools include the source files for the factory programmed hardware and software reference designs.

Persistent Data

The persistent data partition is 64 KB, starting at offset 0x00FF0000. This partition is for maintaining nonvolatile settings and data, such as the MAC address and IP address for the factory-programmed web server reference design. Persistent data is technically no different than other application data, but it is often convenient to think of certain data as independent from the user hardware or software.

JTAG Connectors (J24 & J5)

The Nios development board has two 10-pin JTAG headers (J24 and J5) compatible with Altera download cables, such as the USB-Blaster™. On the Nios development board, each JTAG header connects to one Altera device and forms a single-device JTAG chain. J24 connects to the FPGA (U62), and J5 connects to the EPM7256AE device (U3).

JTAG Connector to FPGA (J24)

J24 connects to the JTAG pins (TCK, TDI, TDO, TMS, TRST) of the FPGA (U62) as shown in Figure 2–18. Altera Quartus II software can directly configure the FPGA with a new hardware image via an Altera download cable as shown in Figure 2–19. In addition, the Nios II IDE can access the Nios II processor JTAG debug module via a download cable connected to the J24 JTAG connector.

Figure 2–18. JTAG Connector (J24) to Cyclone II Device
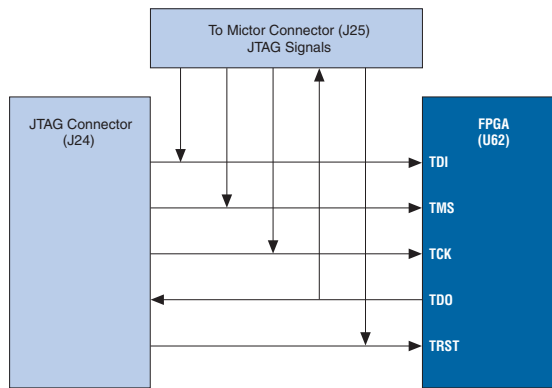
Figure 2–19. USB Blaster Connected to J24 JTAG Connector
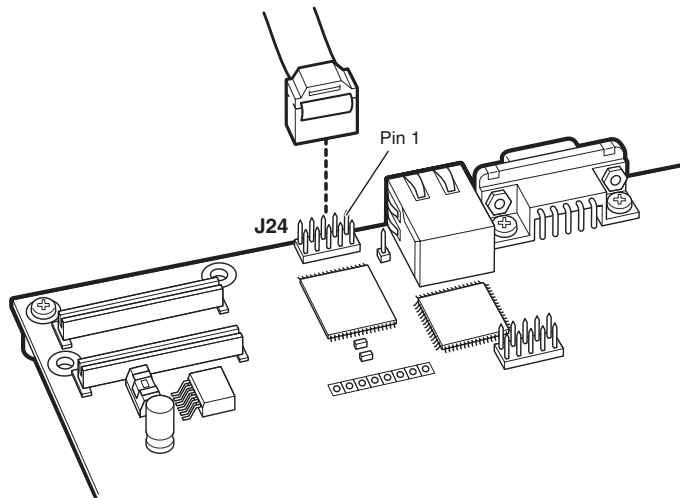
The FPGA's JTAG pins can also be accessed via the Mictor connector (J25). The pins of J24 are connected directly to pins on J25, and care must be taken so that signal contention does not occur between the two connectors.

JTAG Connector to EPM7256AE Device (J5)

J5 connects to the JTAG pins (TCK, TDI, TDO, TMS, TRST) of the EPM7256AE device (U3). Altera Quartus II software can perform in-system programming (ISP) to reprogram the EPM7256AE device (U3) with a new hardware image via an Altera download cable as shown in Figure 2–20.

☞   The orientation of J5 is rotated 180 degrees compared to J24.

Most users never need to reprogram the configuration controller design in the EMP7256AE device. Reprogramming the configuration controller can result in an inoperable development board.

🖝   To restore the board to its factory-programmed condition, see *Appendix B: Restoring the Factory Configuration.*

Figure 2–20. JTAG Connector (J5) to MAX Device
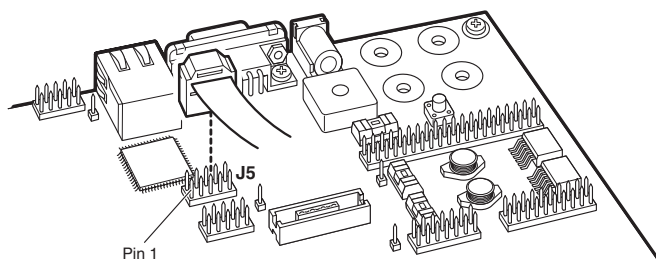
Clock Circuitry

The Nios development board includes a 50 MHz free-running oscillator (Y2) and a zero-skew, point-to-point clock distribution network that drives the FPGA (U62), the EPM7256AE configuration controller device (U3), and pins on the PROTO1 & PROTO2 connectors. The zero-skew buffer (U2) drives the clock distribution network using the free-running 50 MHz clock. Refer to Figure 2–21.

Figure 2–21. Clock Circuitry
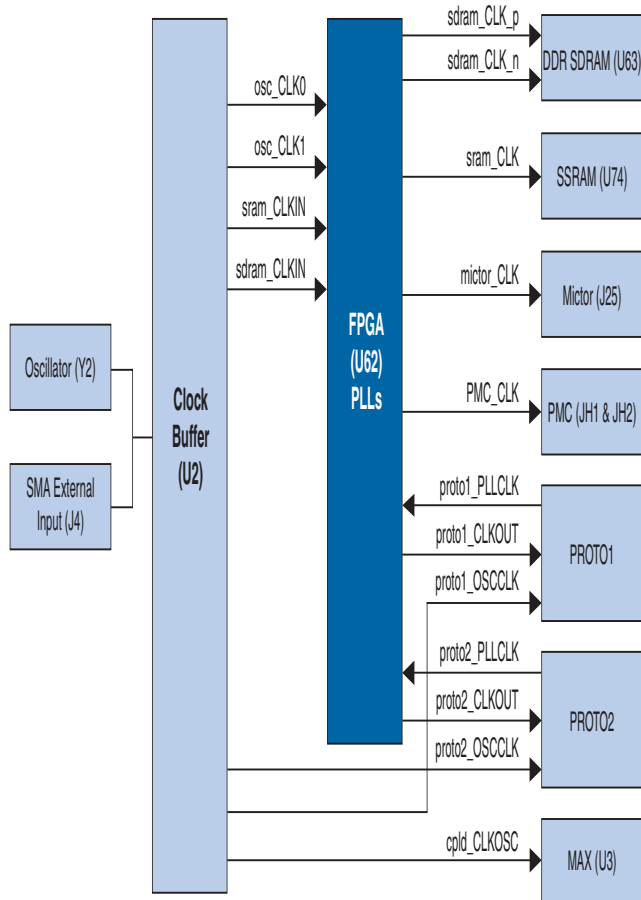
Note to Figure 2–21:

(1)   To use an external clock signal, remove the crystal oscillator from its socket. Make sure to note the correct orientation of the oscillator before removing it.

The FPGA receives clock input from buffer U2, and from the PROTO1 and PROTO2 connectors, as shown in Table 2–21.

Table 2–21. FPGA Clock Input Pin Table

FPGA Pin	FPGA Pin Name	PLL	Signal Source	Board Net Name
B25	IO	N/A	J25 pin 6	mictor_TRCLK
N26	CLK5	PLL2	J13 pin 13	proto1_CLKOUT
AF4	CLK13	PLL4	J17 pin 13	proto2_CLKOUT
P25	CLK6	PLL2	U2 pin 2	osc_CLK0
AC13	CLK15	PLL4	U2 pin 3	osc_CLK1
N2	CLK0	PLL1	U2 pin 4	sram_CLKIN
B13	CLK8	PLL3	U2 pin 6	sdram_CLKIN

The FPGA can synthesize new clock signals internally using on-chip PLLs, and drive the clocks to various components on the board, as shown in Table 2–22.

Table 2–22. FPGA Clock Output Pin Table

FPGA Pin	FPGA Pin Name	PLL (1)	Signal Destination	Board Net Name
AA7	PLL1_OUTp	PLL1	U63 pin 45	sdram_CLK_p
AA6	PLL1_OUTn	PLL1	U63 pin 46	sdram_CLK_n
E5	PLL3_OUTp	PLL3	U74 pin 89	sram_CLK
W26	IO	N/A	JH1 pin 13	pmc_CLK
F21	PLL2_OUTp	PLL2	J13 pin 11	proto1_PLLCLK
F20	PLL2_OUTn	PLL2	J17 pin 11	proto2_PLLCLK
V21	PLL4_OUTp	PLL4	J25 pin 5	mictor_CLK

Note to Table 2–22:
(1) PLLs are only dedicated when using the Enhanced PLL. If you use the Fast PLL, the PLL inputs and outputs are interchangeable. For more information on using PLLs in the Cyclone II refer to the data sheet.

The 50 MHz oscillator (Y2) is socketed and can be changed or removed by the user. To drive the clock circuitry using the external clock connector (J4), remove Y2.

☞   The factory-programmed configuration controller and Altera-provided reference designs work only with the 50 MHz clock.

Power-Supply Circuitry

The Nios development board runs on a 16V, unregulated, input power supply connected to J26. On-board circuitry generates +/-12V, +5V, +3.3V, +2.5V, and +1.2V regulated power levels. For applications requiring high current, separate voltage levels can be supplied from a workbench power supply.

■ The input power-supply on J26 can be either center-negative or center-positive. A bridge rectifier (D34) presents the appropriate polarity to the voltage regulators.
■ The 5V supply is presented on pin 2 of J12 and J15 for use by any device plugged into the PROTO1 & PROTO2 expansion connectors.
■ The 3.3V supply is used as the power source for all FPGA I/O pins. The 3.3V supply is also available for PROTO1 & PROTO2 daughter cards.
■ The 2.5V supply is used only as the power supply for the DDR SDRAM chip and is not available on any connector or header.
■ The 1.2V supply is used only as the power supply for the Cyclone II device core (VCCINT) and it is not available on any connector or header.
■ The +/-12V supply is provided for the PMC connectors JH1 and JH2. Refer to "PMC Connector (JH1 & JH2)" on page 2–26 for more details. When workbench power supplies are connected to the board, a corresponding fuse must be removed to decouple the on-board voltage regulator. Each on-board regulator drives power through a 7A fuse. Refer to Table 2–23.

Table 2–23 lists the details of what voltage levels can be supplied to what points on the board.

Table 2–23. Power Supply and Fuse Details			
Voltage	**Pad**	**Fuse**	**Note**
1.2V	J30	F3	Core power for FPGA.
1.2V	TP12	F7	FPGA PLL power supply.
1.25V	TP10	F5	DDR SDRAM I/O VTT.
1.25V	TP9	F4	DDR SDRAM I/O VREF.
2.5V	TP11	F6	DDR SDRAM VDD power supply. FPGA VCCIO for pins that interface to DDR SDRAM.
3.3V	J29	F2	3.3V power for multiple components on the board.
5V	J28	F1	5.0V power for multiple components on the board.
+12V	J31	F8	Power for the PMC connectors.
-12V	TP13	F9	Power for the PMC connectors.

Appendix A. Restoring the Factory Configuration

Introduction

To restore the factory configuration, you must reprogram the flash memory on the board, and you must Nios Development Board Cyclone II Edition the EPM7256AE configuration controller device.

Nios II Embedded Design Suite provides the files required for this operation in the directory *<Nios II EDS install path>*/**examples /factory_recovery**.

Reprogramming the Flash Memory

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

1. Open a Nios II command shell.

   On a Windows PC, click **Windows Start, point to Programs, Altera, Nios II EDS** *<installed version>*, and then click **Nios II Command Shell**.

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

   ```
   cd factory_recovery/niosII_cycloneII_2c35
   ```

3. Run the flash-restoration script:

   ```
   ./restore_my_flash
   ```

4. Follow the script's instructions.

Reprogramming the EPM7256AE Configuration Controller Device

If the configuration controller design was modified, you must also reprogram the EMP7256AE device (U3). To reprogram the EMP7256AE configuration controller, perform the following steps:

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

   ☞ The orientation of J5 is opposite that of J24. When properly connected to J5, the programming cable lies naturally over the clock oscillator and the dual seven-segment display.

2. Launch the Quartus II software, and click **Programmer** on the Tools menu.

3. Click **Add File** and select the following programming file: *<Nios II EDS install path>***/examples/ factory_recovery/niosII_cycloneII_2c35/config_controller.pof**.

4. In the **Programmer**, turn on the **Program/Configure** checkbox, and click **Start** to reprogram the EPM7256AE device.

5. Press the **Factory Config** button to perform a power on reset and reconfigure the FPGA from flash memory. You should see the Factory LED turned on and activity on LEDs D0 through D7.

Your board is now reconfigured to the default factory condition.

Introduction

The Nios development board is factory-programmed with a reference design that implements a web server, among other functions as shown in Figure B–1. This chapter describes 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.

*Figure B–1. Web Server Reference Design*
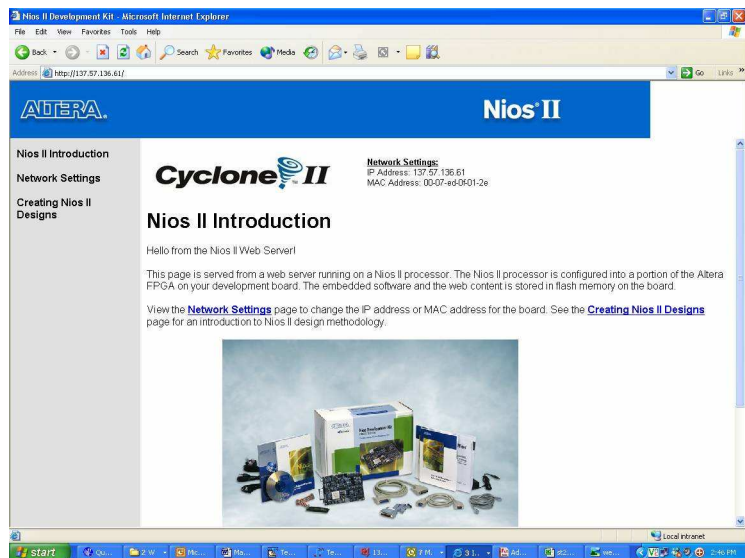
Connecting the Ethernet Cable

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 and insert the male end of the crossover adapter into RJ1 on the Nios development board as shown in Figure B–2.

Figure B–2. Point-to-Point Connection
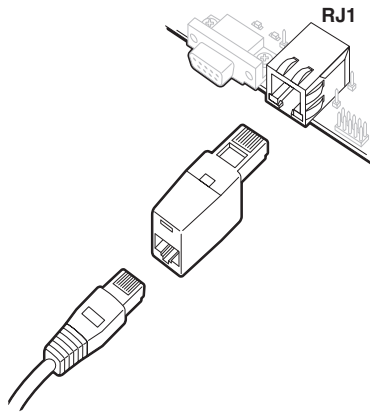
　　b. Connect the other end of the RJ45 connector directly to the network (Ethernet) port on your host computer.

Connecting the LCD Screen

The Nios II development kit includes 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 expansion prototype connector J12. Refer to the *Nios II Development Kit, Getting Started User Guide* for details.

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 attempts to acquire an IP address via the DHCP protocol. The board continues 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 ultimately fails, and the web server defaults to a static IP address.

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

Static IP Address
If the DHCP process fails, the board uses 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. Open a Nios II command shell. On Windows PCs, On a Windows PC, click **Windows Start, point to Programs, Altera, Nios II EDS** *<installed version>*, and then click **Nios II Command Shell**. A shell window appears with a command prompt.

3. Press the SW9 button labeled Factory Config on the board.

4. At the Nios II command 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 only need to assign an IP address to your board once. However, you can 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 command 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 displays 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 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 uses 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. Refer to "Browsing to Your Board" on page B–5 to continue.

If you don't have the ability to change the IP address of your host computer, you can 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 can 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 B–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 B–3.

Browsing to 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 to 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.