Nios® II

Nios Development Board

Reference Manual, Stratix Edition

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


Printed on recycled paper

MNL-N2DEVLBDSTX-1.1

About this Manual

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

Table ii–1 shows the reference manual revision history.

Table ii–1. Reference Manual Revision History

Date	Description
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	First publication of a manual for the Nios Development Kit, Stratix Edition development board.

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."
`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

Board Components

Features

- A Stratix™ EP1S10F780C6 device
- 8 Mbytes of flash memory
- 1 Mbyte of static RAM
- 16 Mbytes of SDRAM
- On board logic for configuring the Stratix device from flash memory
- On-board Ethernet MAC / PHY device
- Two 5-V-tolerant expansion / prototype headers each with access to 41 Stratix 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 Stratix user I / O pins
- Eight LEDs connected to Stratix 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, Stratix Edition, provides a hardware platform for developing embedded systems based on Altera Stratix devices. The Nios development board, features a Stratix EP1S10F780C6 device with 10,570 logic elements (LEs) and 920, 448 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.

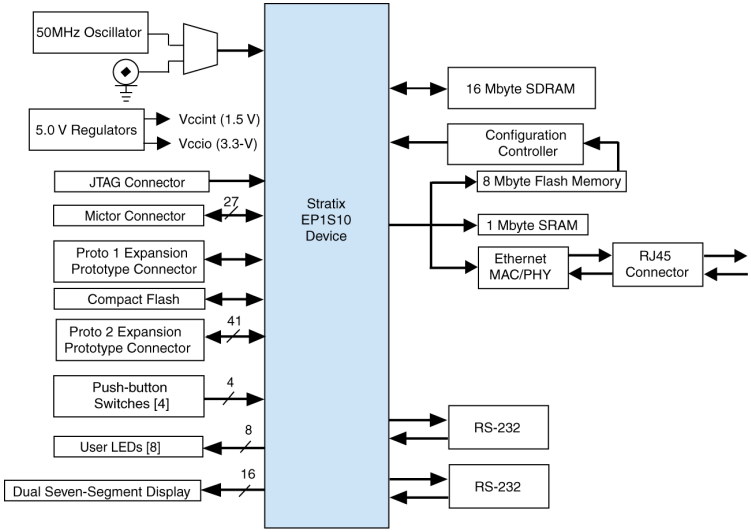
This document describes the hardware features of the Nios development board, including detailed pin-out information, to enable designers to create custom FPGA designs that interface with all components on the board.

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 Stratix board.

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



Default Reference Design

When power is applied to the board, on-board logic configures the Stratix 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 "Connecting to the Board via Ethernet" on page C–1.

Restoring the Default Reference Design to the Board

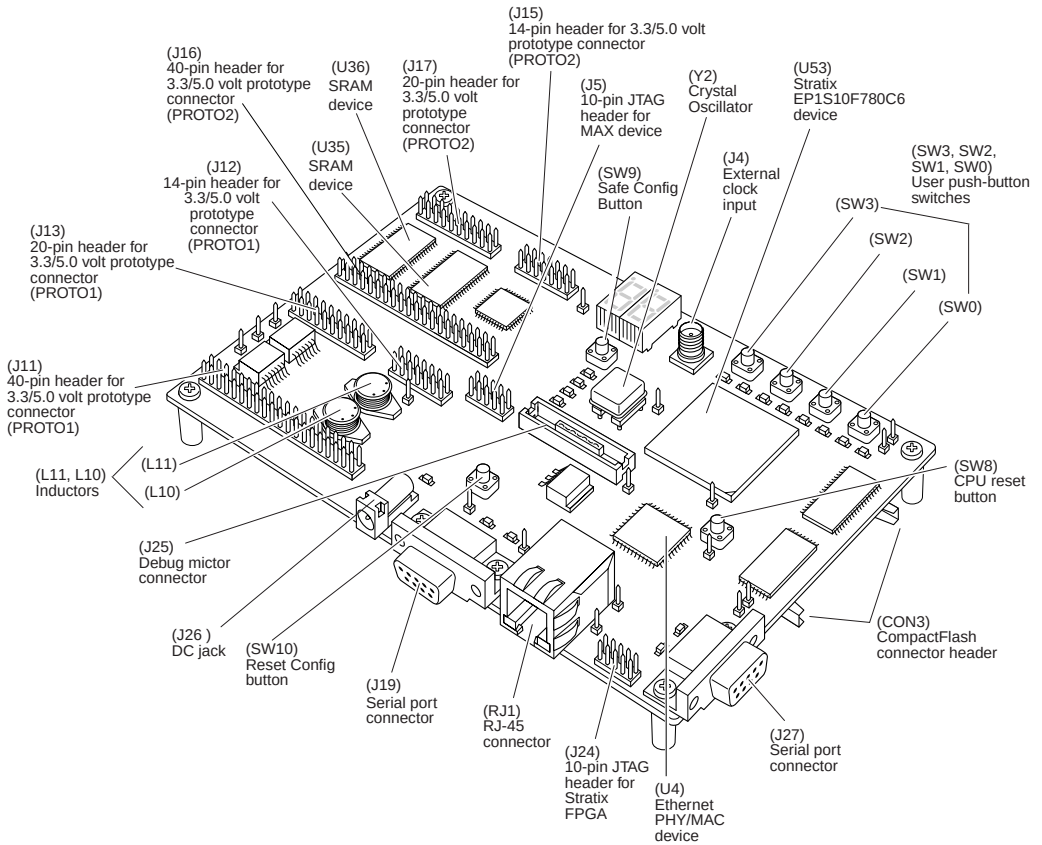
In the course of development, you may overwrite or erase the flash memory space containing the default reference design. Altera provides the flash image for the default reference design so you can return the board to its default state. See Appendix B, Restoring the Factory Configuration for more information.

Nios Development Board Components

This section contains a brief overview of several 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 Nios II development kit **documents** directory.

Figure 1–2. Nios Development Board



The Stratix EP1S10 Device

U53 is a Stratix EP1S10F780C6 device in a 780-pin FineLine BGA package. Table 1–1 lists the Stratix device features.

Table 1–1. Stratix EP1S10 Device Features

LEs	10,570
M512 RAM blocks (32 X 18 bits)	94
M4K RAM blocks (128 X 36 bits)	60
M-RAM blocks (4K X 144 bits)	1
Total RAM bits	920,448
DSP blocks	6
Embedded multipliers	48
PLLS	6
Maximum user I/O pins	426

The development board provides two separate methods for configuring the Stratix 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 Stratix header (J24).

2. When power is applied to the board, a configuration controller device (U3) attempts to configure the Stratix device with hardware configuration data stored in flash memory. For more information on the configuration controller, see "Configuration Controller Device (EPM7128AE)" on page 1–21.

See the Altera Stratix literature page for Stratix-related documentation including Stratix EP1S10 pin out data at **www.altera.com/literature/ lit-stx.html**.

Flash Memory Device

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

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

2. The flash memory can hold a Stratix device configuration file that is used by the configuration controller to load the Stratix device at power-up. "Configuration Controller Device (EPM7128AE)" on page 1–21 for related information.

Hardware configuration data that implements the Nios II reference design is pre-stored in this flash memory. The factory programmed Nios II reference design, once loaded, can identify the 8 Mbyte flash memory in its address space, and can program new data (either new Stratix 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 Appendix A, Shared Bus Table.

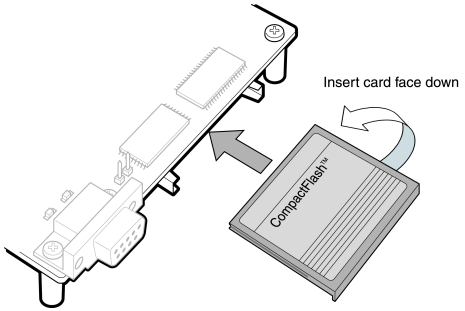
See **www.amd.com** for detailed information about the flash memory device.

CompactFlash Connector

The CompactFlash connector header (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 Stratix I/O pins with expansion prototype connector PROTO1. See 2"Expansion Prototype Connector (PROTO1)" on page 1–11 for details on PROTO1.

Table 1–2 below provides CompactFlash pin out details.

Table 1–2. CompactFlash (CON3) Pin Table

Pins on CompactFlash (CON3)	CompactFlash Function	Connects To *(1)*
1	GND	GND
2	D03	M4
3	D04	N6
4	D05	N1
5	D06	N9
6	D07	P3
7	-CE	J2
8	A10	M7
9	-OE	K7
10	A09	K3
11	A08	H3
12	A07	L7
13	VCC	H4 *(2)*
14	A06	L8
15	A05	H2
16	A04	H1
17	A03	L6
18	A02	L10
19	A01	J3
20	A00	L9
21	D00	N3
22	D01	L2
23	D02	N8
24	WP	K4
25	-CD2	GND *(3)*
26	-CD1	R3
27	D11	M3
28	D12	N7
29	D13	L1
30	D14	N4
31	D15	L3
32	-CE2	K8
33	-VS1	GND *(3)*

Table 1–2. CompactFlash (CON3) Pin Table		
Pins on CompactFlash (CON3)	**CompactFlash Function**	**Connects To *(1)***
34	-OIORD	M9
35	-IOWR	M10
36	-WE	L5
37	RDY/BSY	M5
38	VCC	H4 *(2)*
39	-CSEL	GND *(3)*
40	-VS2	no connect *(3)*
41	RESET	*(4)*
42	-WAIT	K1
43	-INPACK	J4
44	-REG	G2
45	BVD2	J1
46	BVD1	M8
47	D081	N10
48	D091	M2
49	D101	N5
50	GND	GND *(3)*

Note to Table 1–1

(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 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 Stratix device (see Table 1–3). 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 Stratix Pin
A0	25	AE4
A1	26	W12
A2	27	AC11
A3	60	W10
A4	61	AA11
A5	62	AC10
A6	63	AB11
A7	64	AC8
A8	65	AB10
A9	66	V11
A10	24	Y11
A11	21	AB7
BA0	22	AG19
BA1	23	AF19
DQ0	2	AH4
DQ1	4	AE5
DQ2	5	AG3
DQ3	7	AG5
DQ4	8	AG4
DQ5	10	AF4
DQ6	11	AH5
DQ7	13	AF5
DQ8	74	AE6
DQ9	76	AG6
DQ10	77	AH6
DQ11	79	AD6
DQ12	80	AF7
DQ13	82	AH7
DQ14	83	AG7
DQ15	85	AF6

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

Pin Name	Pin Number	Connects to Stratix Pin
DQ16	31	AG8
DQ17	33	AF8
DQ18	34	AD8
DQ19	36	AH9
DQ20	37	AH8
DQ21	39	AE9
DQ22	40	AF9
DQ23	42	AG9
DQ24	45	AD10
DQ25	47	AF10
DQ26	48	AH10
DQ27	50	AE10
DQ28	51	AF11
DQ29	53	AE11
DQ30	54	AH11
DQ31	56	AG11
DQM0	16	AE14
DQM1	71	Y13
DQM2	28	AE7
DQM3	59	AG10
RAS_N	19	AH3
CAS_N	18	AD18
CKE	67	AE18
CS_N	20	AG18
WE_N	17	AH19
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 Stratix 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 factory programmed Nios II reference design identifies these SRAM devices in its address space as a contiguous 1Mbyte, 32-bit-wide, zero-wait-state main memory.

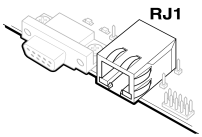
The SRAM devices share address and data connections with the flash memory and the Ethernet MAC / PHY device. For shared bus information, see Appendix A, Shared Bus Table.

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 Stratix 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 Appendix A, Shared Bus Table

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

Expansion Prototype Connector (PROTO1)

The PROTO1 expansion prototype connectors share Stratix 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 Stratix device. Each signal passes through analog switches (U19, U20, U21, U22 and U25) to protect the Stratix 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 Stratix's phase-locked loop (PLL)-output from U53.
- 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 with 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 Stratix device. Unless otherwise noted, labels indicate Stratix device pin numbers.

Figure 1–5. Expansion Prototype Connector - J11

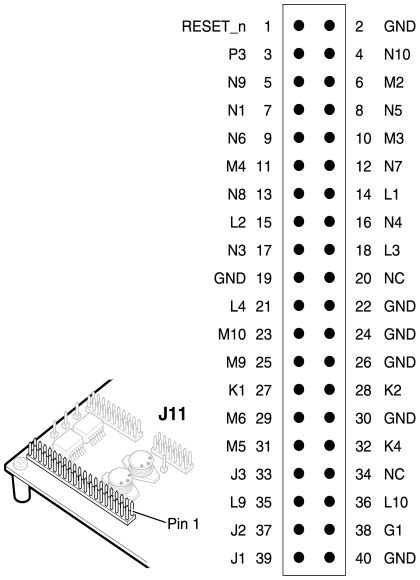


Figure 1–6. Expansion Prototype Connector - J12

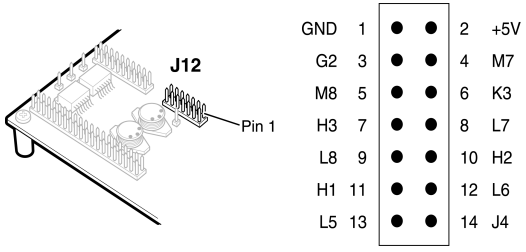
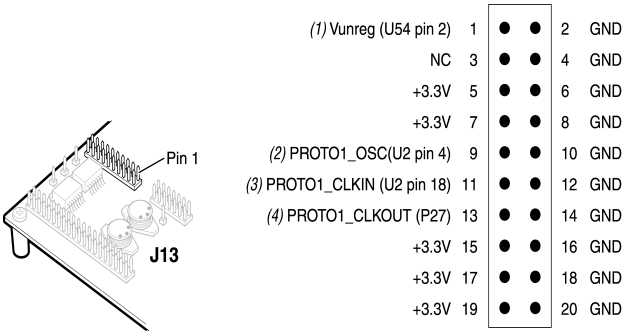


Figure 1–7. Expansion Prototype Connector - J13



Note 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 JP15, JP16, and JP17 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 Stratix device. Each signal passes through analog switches (U27, U28, U29, U30 and U31) to protect the Stratix 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 Stratix's phase-locked loop (PLL)-output (from U53).
- 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 with 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 Stratix device. Unless otherwise noted, the labels indicate Stratix device pin numbers.

Figure 1–8. Expansion Prototype Connector - J16

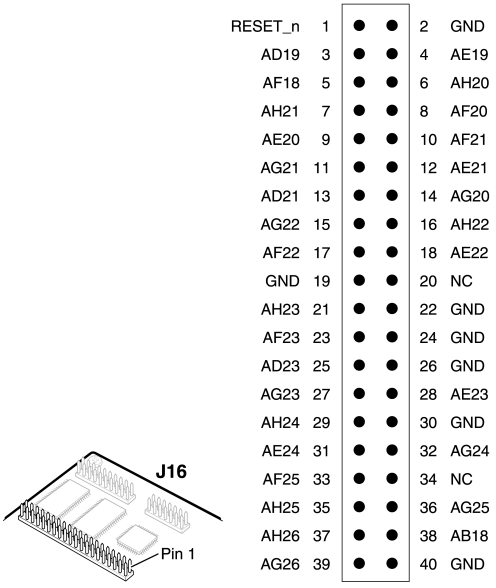


Figure 1–9. Expansion Prototype Connector - J15

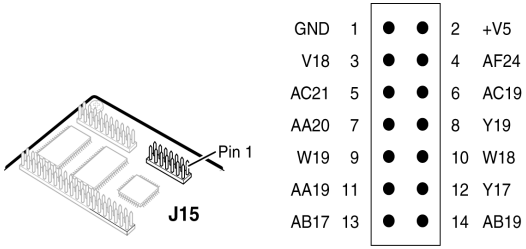
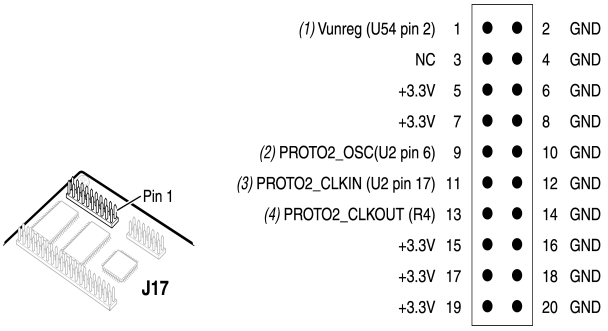


Figure 1–10. Expansion Prototype Connector - J17



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

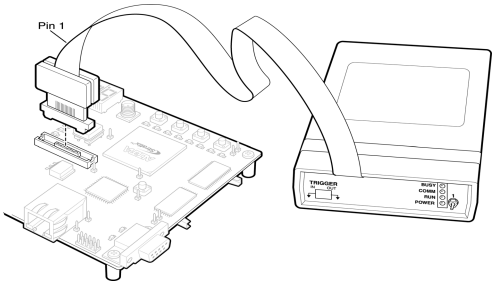
The Mictor connector (J25) can be used to transmit up to 27 high-speed I/O signals with very low noise via a shielded Mictor cable. J25 is used as a debug port. Twenty five of the Mictor connector signals are used as data, and two signals are used as clock input and clock output.

Most pins on J25 connect to I/O pins on the Stratix device (U53). 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 on page 1–17 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 to both the JTAG pins on the Stratix device (U53), and the Stratix device's JTAG connector (J24). The JTAG signals have special usage requirements. You cannot use J25 and J24 at the same time.

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

Figure 1–12. Mictor Connector Signaling

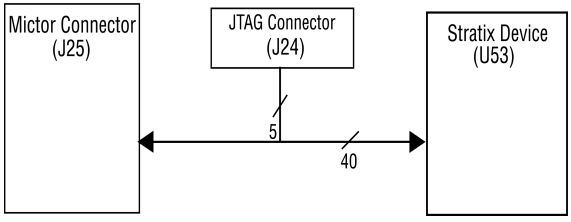
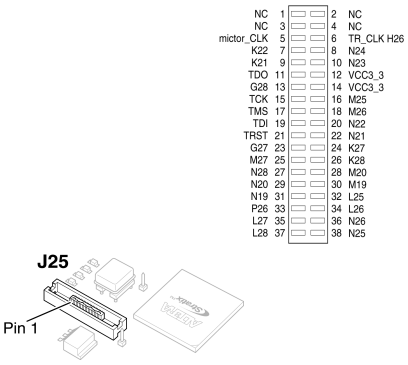


Figure 1–13. Debug Mictor Connector - J25



Serial Port Connectors

J19 and 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 Stratix device, because the Stratix 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 Stratix 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 Stratix device.

Figure 1–14. Console Serial Port Connector - J19

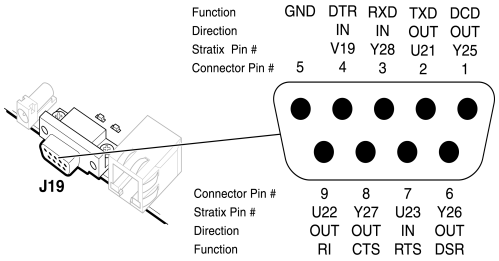
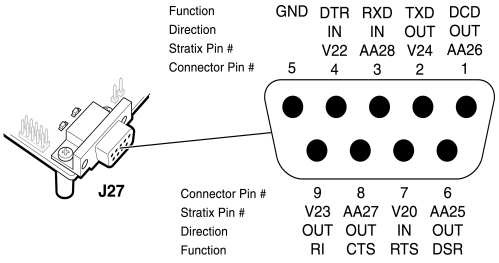


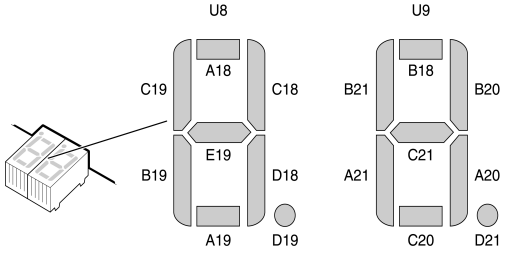
Figure 1–15. Debug Serial Port Connector - J27



Dual 7-Segment Display

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

Figure 1–16. Dual-Digit 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 and are used to provide stimulus to designs in the Stratix device. See Figure 1–17. Each switch is connected to a Stratix general-purpose I/O pin with a pull-up resistor as shown in Table 1–4. The Stratix 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
Stratix Pin	W5	W6	AB2	AB1

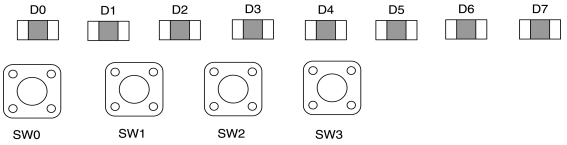
Individual LEDs

This Nios development board provides eight individual LEDs connected to the Stratix device. See Figure 1–17. D0 – D7 are connected to general purpose I/O pins on the Stratix device as shown in Table 1–5. When the Stratix 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
Stratix Pin	H27	H28	L23	L24	J25	J26	L20	L19

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



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 Stratix device from data stored in flash memory.

Reset Distribution

The EPM7128AE device 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 Reset, Config button (SW10).
3. Asserting (driving 0 volts on) the EPM7128AE's reconfigreq_n input pin (from a Stratix design).
4. Pressing the Safe Config button (SW9).

Stratix Configuration

At power-up or reset, the configuration controller reads data out of the flash memory, and presents the necessary control signals to configure the Stratix device. The Stratix device is configured using fast passive parallel mode.

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 integrated development environment (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 two separate Stratix device configurations stored in flash memory. These two configurations are conventionally referred to as the safe configuration and the user configuration. Upon reset or when the Reset, Config button (SW10) is pressed, the configuration controller will attempt to load the Stratix device with user configuration data. If this process fails (either because the user-configuration is invalid or not present), the configuration controller will then load the Stratix device with safe configuration data.

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

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

When SW9 (Safe Config) is pressed, the configuration controller will ignore the user configuration and always configure the Stratix 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.

Using Conventional Flash Memory

The Nios development board includes an 8 MByte flash memory device (U5) as shown in Table 1–6 on page 1–23. 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)**

Factory-programmed–***do not erase*** Available for user data.

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 1–24.

User Hardware Image

At power on, or when the Reset, Config button (SW10) is pressed, the configuration controller reads 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 include 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 (Safe Config) 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 1–24.

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 Appendix B, Restoring the Factory Configuration 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**

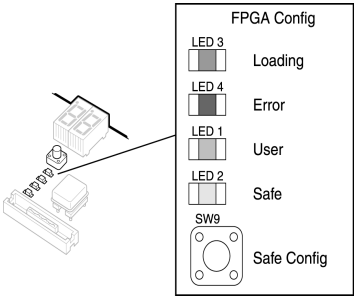
The Configuration-Status LEDs

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 on page 1–25). If a new configuration was downloaded into the Stratix device via JTAG, then all of the LEDs will turn off.

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 Stratix FPGA.
LED4	Error	Red	If the red Error LED is on, then configuration was NOT transferred from flash memory into the Stratix device. This can happen if, for example, the flash memory contains neither a valid **User** or **Safe** 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 Stratix device. If the Stratix device was successfully loaded with the user-configuration from flash memory, LED1 will remain on continuously.
LED2	Safe Config	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 Stratix device.

Figure 1–18. LED1 – LED4



Configuration & Reset Buttons

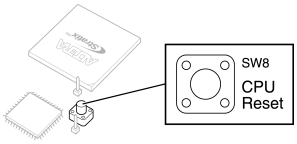
The Nios development board uses dedicated switches SW8, SW9 and SW10 for the following fixed functions:

SW8 – CPU Reset

When SW8 is pressed, a logic-0 is driven onto the Stratix devices' DEV_CLRn pin (and user I/O AC9). The result of pressing SW8 depends on how the Stratix 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



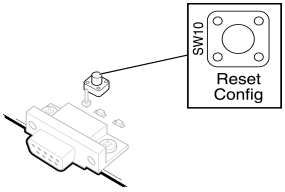
SW9 – Safe Config

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

SW10 – Reset, Config

Reset, Config (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 1–26 for more details. After SW10 is pressed, the configuration controller will load the Stratix device from flash memory.

Figure 1–20. Reset, Config Button



Power-Supply Circuitry

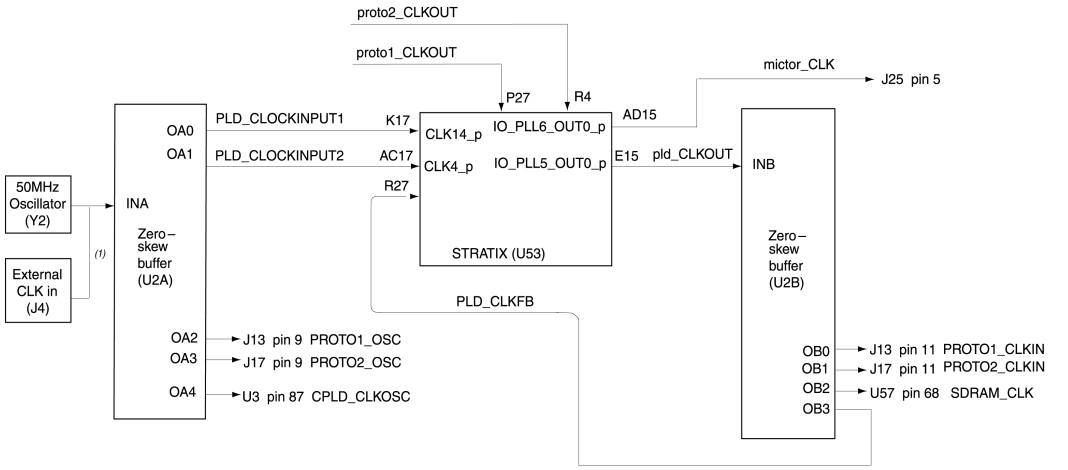
The Nios development board runs from a 9-V, unregulated, center-negative input power supply. On-board circuitry generates 5-V, 3.3-V, and 1.5-V regulated power levels.

- The 5-V supply is presented on pin 2 of J12 and J15 for use by any device plugged into the PROTO1 & PROTO2 expansion connectors.
- The 3.3-V supply is used as the power source for all Stratix device I/O pins. The 3.3-V supply is also available for PROTO1 & PROTO2 daughter cards.
- The 1.5-V supply is used only as the power supply for the Stratix 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 that drives both the Stratix device and pins on the expansion prototype connectors, the EPM7128AE 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 Stratix'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 Stratix 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_PLL5_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 CLK2_p clock input on the Stratix device. This clock feedback to the Stratix device is not used by Altera-supplied reference designs, but is available to the user if necessary.

The Stratix device can also supply a clock from the IO_PLL6_OUT0_p pin to the Mictor connector (J25).

The 50 MHz oscillator (Y2) is socketed and can be changed by the user. However, the EMP7128AE 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 two 10-pin JTAG headers (J5 and J24) 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 Stratix device (U53), and J5 connects to the EPM7128AE device (U3).

JTAG Connector to Stratix Device (J24)

J24 connects to the JTAG pins (TCK, TDI, TDO, TMS, TRST) of the Stratix device (U53) as shown in Figure 1–22. Altera Quartus II software can directly configure the Stratix 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 JTAG debug module via a download cable connected to the J24 JTAG connector.

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

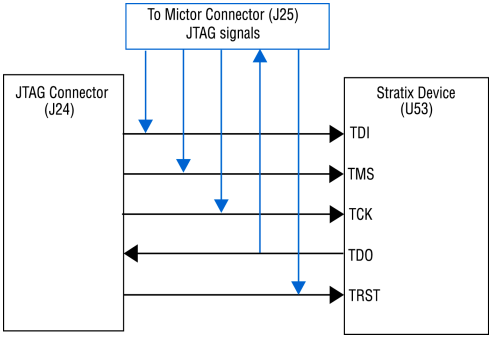
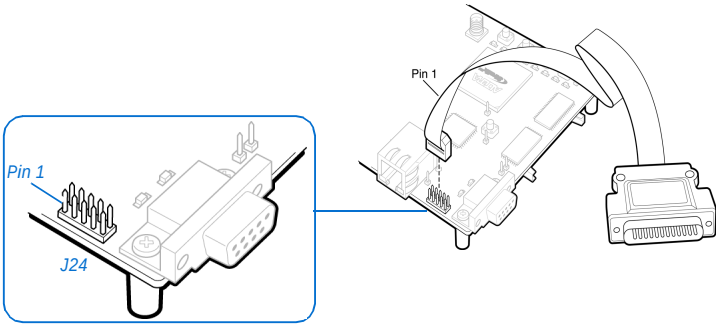


Figure 1–23. JTAG Connection to Download Cable



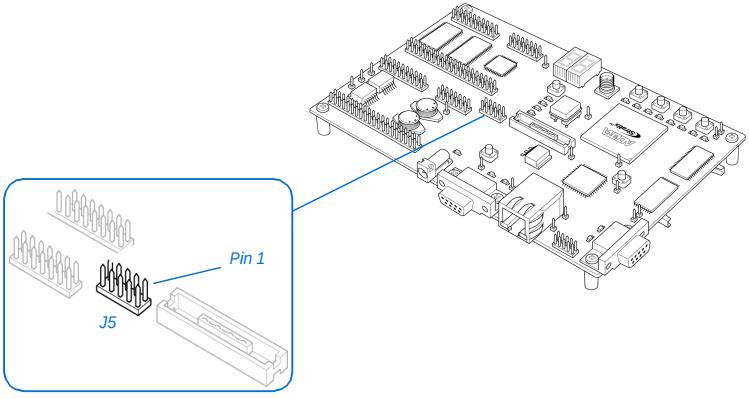
The Stratix device's JTAG pins can also be accessed via the Mictor connector (J25). The pins of J24 are connected directly to pins on J25, and care must be taken so that signal contention does not occur between the two connectors.

JTAG Connector to EPM7128AE Device (J5)

J5 connects to the JTAG pins (TCK, TDI, TDO, TMS, TRST) of the EPM7128AE device (U3). Altera Quartus II software can perform in-system programming (ISP) to reprogram the EPM7128AE device (U3) with a new hardware image via an Altera download cable.

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

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



Appendix A. Shared Bus Table

Description

On the Nios development board, Stratix 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 (U53)		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	A4	A0	27						
FSE_A1		IO	A3	A1	22					A1	78
FSE_A2		IO	B3	A2	21	A0	1	A0	1	A2	79
FSE_A3		IO	B5	A3	20	A1	2	A1	2	A3	80
FSE_A4		IO	B4	A4	19	A2	3	A2	3	A4	81
FSE_A5		IO	C4	A5	18	A3	4	A3	4	A5	82
FSE_A6		IO	A5	A6	17	A4	5	A4	5	A6	83
FSE_A7		IO	C5	A7	16	A5	18	A5	18	A7	84
FSE_A8		IO	D5	A8	10	A6	19	A6	19	A8	85
FSE_A9		IO	E6	A9	9	A7	20	A7	20	A9	86
FSE_A10		IO	A6	A10	42	A8	21	A8	21	A10	87
FSE_A11		IO	B7	A11	8	A9	22	A9	22	A11	88
FSE_A12		IO	D6	A12	7	A10	23	A10	23	A12	89
FSE_A13		IO	A7	A13	6	A11	24	A11	24	A13	90
FSE_A14		IO	D7	A14	5	A12	25	A12	25	A14	91
FSE_A15		IO	C6	A15	4	A13	26	A13	26	A15	92
FSE_A16		IO	C7	A16	3	A14	27	A14	27		
FSE_A17		IO	B6	A17	46	A15	42	A15	42		
FSE_A18		IO	D8	A18	15	A16	43	A16	43		
FSE_A19		IO	C8	A19	43	A17	44	A17	44		
FSE_A20		IO	E8	A20	44						
FSE_A21		IO	D9	A21	35						
FSE_A22		IO	B9	A22	2						

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

NET Name	NET Description	PLD (U53)		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	H12	D0	31	D0	7			D0	107
FSE_D1		IO	F12	D1	32	D1	8			D1	106
FSE_D2		IO	J12	D2	33	D2	9			D2	105
FSE_D3		IO	M12	D3	34	D3	10			D3	104
FSE_D4		IO	H17	D4	38	D4	13			D4	102
FSE_D5		IO	K18	D5	39	D5	14			D5	101
FSE_D6		IO	H18	D6	40	D6	15			D6	100
FSE_D7		IO	G18	D7	41	D7	16			D7	99
FSE_D8		IO	B8			D8	29			D8	76
FSE_D9		IO	A8			D9	30			D9	75
FSE_D10		IO	A9			D10	31			D10	74
FSE_D11		IO	C9			D11	32			D11	73
FSE_D12		IO	E10			D12	35			D12	71
FSE_D13		IO	A10			D13	36			D13	70
FSE_D14		IO	C10			D14	37			D14	69
FSE_D15		IO	B10			D15	38			D15	68
FSE_D16		IO	A11					D0	7	D16	66
FSE_D17		IO	C11					D1	8	D17	65
FSE_D18		IO	D11					D2	9	D18	64
FSE_D19		IO	B11					D3	10	D19	63
FSE_D20		IO	D10					D4	13	D20	61
FSE_D21		IO	G10					D5	14	D21	60
FSE_D22		IO	F10					D6	15	D22	59
FSE_D23		IO	H11					D7	16	D23	58
FSE_D24		IO	G11					D8	29	D24	56
FSE_D25		IO	F8					D9	30	D25	55
FSE_D26		IO	J9					D10	31	D26	54
FSE_D27		IO	J13					D11	32	D27	53
FSE_D28		IO	L13					D12	35	D28	51
FSE_D29		IO	M11					D13	36	D29	50
FSE_D30		IO	L11					D14	37	D30	49
FSE_D31		IO	G7					D15	38	D31	48

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

NET Name	NET Description	PLD (U53)		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	K19	CE_n	28						
FLASH_OE-N	Read Enable	IO	F19	OE_n	30						
FLASH_RW-N	Write Enable	IO	G19	WE_n	11						
FLASH_RY-BY_N	Ready/Busy	IO	L18	RY/BY _n	14						
SRAM_BE_N0	Byte Enable 0	IO	M18			BE0#	39				
SRAM_BE_N1	Byte Enable 1	IO	F17			BE1#	40				
SRAM_BE_N2	Byte Enable 2	IO	J18					BE2#	39		
SRAM_BE_N3	Byte Enable 3	IO	L17					BE3#	40		
SRAM_CS_N	Chip Select	IO	B24			CS_n	6	CS_n	6		
SRAM_OE_N	Read Enable	IO	B26			OE_n	41	OE_n	41		
SRAM_WE_N	Write Enable	IO	C24			WE_n	17	WE_n	17		
ENET_ADS_N	Address Strobe	IO	V25							ADS#	37
ENET_AEN	Address Enable	IO	V28							AEN	41
ENET_BE_N0	Byte Enable 0	IO	T22							BE0#	94
ENET_BE_N1	Byte Enable 1	IO	U26							BE1#	95
ENET_BE_N2	Byte Enable 2	IO	U25							BE2#	96
ENET_BE_N3	Byte Enable 3	IO	T19							BE3#	97
ENET_CYCLE_N	Bus Cycle	IO	U27							CYCLE #	35
ENET_DATACS_N	Data Chip Select	IO	T20							DATA CS#	34
ENET_INTRQ0	Interrupt	IO	V27							INTRO	29
ENET_IOCHRDY	IO Char Ready	IO	V26							ARDY	38
ENET_IOR_N	Read	IO	T23							RD#	31
ENET_IOW_N	Write	IO	T24							WR#	32
ENET_LCLK	Local Bus Clock	IO	R26							LCLK	42
ENET_LDEV_N	Local Device	IO	T26							LDEV#	45
ENET_RDYRTN_N	Ready Return	IO	T28							RDYR TN#	46
ENET_W_R_N	Write/Read	IO	T21							W/R#	36

Appendix B. Restoring the Factory Configuration

Introduction

Your Nios development board can always 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, Stratix Edition, already programmed with the first-generation Nios 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/<development board>
   ```

   where <*development board*> is either the **niosII_stratix_1s10** or **niosII_stratix_1s10_ES** directory**.**

   ES development boards have Stratix EP1S10 devices whose part numbers end with "ES".

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** window (Tools menu).

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

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

   where *<development board>* is either the **niosII_stratix_1s10** board or **niosII_stratix_1s10_ES** board.

   ES development boards have Stratix EP1S10 devices whose part numbers end with “ES”.

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

5. Press the *Safe Config* button to perform a power-on reset and reconfigure the Stratix 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

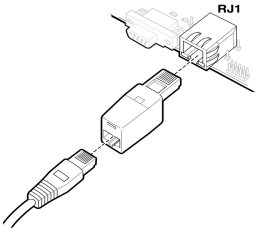
The Nios development board is factory-programmed with a default reference design that implements a web server, among other functions. The sections below describe how to connect a host computer to the board's Ethernet port, assign an IP address to the board, and browse to the web server from the host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

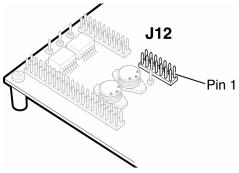
Figure C–1. Point-to-Point Connection



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 *Safe Config*.

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

R

S