Section III. Diagnostic Tests

Revision History .................................................................................................................................. Section III–i

Chapter 5. Perform the Production Diagnostic Tests

Required Hardware & Software ........................................................................................................ 5–1
Set Up the Board .................................................................................................................................. 5–1
Perform the Standard Tests ................................................................................................................ 5–10
  User I/O ......................................................................................................................................... 5–10
  Stratix GX-to-Stratix Bridge ............................................................................................................. 5–12
  Source Synchronous HM-Zd Interface (Stratix GX HM-Zd SPI 4.2) ........................................ 5–14
  Source Synchronous DPA SMA interface (Stratix GX SMA DPA) .............................................. 5–15
  Gigabit Transceivers with SMA Interface (Stratix GX SMA XCVR) ........................................... 5–17
  Gigabit Transceivers with HM-Zd Interface (Stratix GX HM-Zd XCVR) .................................... 5–18
  Gigabit Transceivers with SFP Interface (Stratix GX SFP XCVR) ................................................ 5–19
  Gigabit Transceivers with XPAK Interface (Stratix GX XPAK XCVR) ...................................... 5–20
  Gigabit Transceivers with HSSDC2 Interface (Stratix GX HSSDC2 XCVR) (GX40 Device Only) ........................................................................................................................................ 5–21
Nios-Based Tests .............................................................................................................................. 5–22
  DDR Interface (Stratix GX Nios DDR) .......................................................................................... 5–22
  PMC Card Interface (Stratix Nios PMC) ....................................................................................... 5–23
  Ethernet, On-Board Flash & EPC16 Flash Interface (Stratix Nios Ethernet) (Stratix Nios On-Board Flash) (Stratix Nios EPC16 Flash) ......................................................................................... 5–25
  10/100 Ethernet Network-Interface Card I/O Interface (Stratix Nios PROTO1 IO) .................. 5–29
  Compact Flash Interface (Stratix Nios Compact Flash) .............................................................. 5–32
  Finishing Test & Breakdown .......................................................................................................... 5–35

Chapter 6. Troubleshooting

Chapter 7. Diagnostic Test Details

Standard Tests ..................................................................................................................................... 7–1
  User I/O ......................................................................................................................................... 7–1
Section IV. Demonstrations

Revision History ............................................................................................................................................ Section IV–i

Chapter 8. Stratix GX SPI-4.2 Demonstration with 10-Port Gigabit Ethernet MAC

Chapter 9. 10-Gigabit Ethernet MAC Demonstration

Introduction .................................................................................................................................................. 9–1
Contact .................................................................................................................................................... 9–3
About this User Guide

This user guide provides comprehensive information about the Altera® High-Speed Development Kit, Stratix™ GX Edition.

How to Find Information

You can find more information in the following ways:

- The Adobe Acrobat Find feature, which searches the text of a PDF document. Click the binoculars toolbar icon to open the Find dialog box.
- Acrobat bookmarks, which serve as an additional table of contents in PDF documents.
- Thumbnail icons, which provide miniature previews of each page, provide a link to the pages.
- Numerous links, shown in green text, which 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 in Table 1–1.

Table 1–1. Contact Altera

<table>
<thead>
<tr>
<th>Information Type</th>
<th>USA &amp; Canada</th>
<th>All Other Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical support</td>
<td><a href="http://www.altera.com/mysupport/">www.altera.com/mysupport/</a></td>
<td>altera.com/mysupport/</td>
</tr>
<tr>
<td></td>
<td>(800) 800-EPLD (3753)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(7:00 a.m. to 5:00 p.m. Pacific Time)</td>
<td>(408) 544-7000 (1)</td>
</tr>
<tr>
<td>Product literature</td>
<td><a href="http://www.altera.com">www.altera.com</a></td>
<td><a href="http://www.altera.com">www.altera.com</a></td>
</tr>
<tr>
<td>Altera literature services</td>
<td><a href="mailto:lit_req@altera.com">lit_req@altera.com</a> (1)</td>
<td><a href="mailto:lit_req@altera.com">lit_req@altera.com</a> (1)</td>
</tr>
<tr>
<td>Non-technical customer service</td>
<td>(800) 767-3753</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(408) 544-7000 (1)</td>
<td></td>
</tr>
<tr>
<td>FTP site</td>
<td>ftp.altera.com</td>
<td>ftp.altera.com</td>
</tr>
</tbody>
</table>

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 in Table 1–2.

<table>
<thead>
<tr>
<th>Visual Cue</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bold Type with Initial Capital Letters</strong></td>
<td>Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: <em>Save As</em> dialog box.</td>
</tr>
<tr>
<td><strong>bold type</strong></td>
<td>External timing parameters, directory names, project names, disk drive names, filenames, filename extensions, and software utility names are shown in bold type. Examples: <em>fMAX</em>, <em>\qdesigns</em> directory, <em>d</em>: drive, <em>chiptrip.gdf</em> file.</td>
</tr>
<tr>
<td><strong>Italic Type with Initial Capital Letters</strong></td>
<td>Document titles are shown in italic type with initial capital letters. Example: <em>AN 75: High-Speed Board Design.</em></td>
</tr>
<tr>
<td><strong>Italic type</strong></td>
<td>Internal timing parameters and variables are shown in italic type. Examples: <em>tPIA</em>, <em>n</em> + 1.</td>
</tr>
<tr>
<td>Variable names are enclosed in angle brackets (&lt; &gt;) and shown in italic type. Example: <em>&lt;file name&gt;</em>, <em>&lt;project name&gt;</em>.pof file.</td>
<td></td>
</tr>
<tr>
<td>Initial Capital Letters</td>
<td>Keyboard keys and menu names are shown with initial capital letters. Examples: <em>Delete key</em>, the <em>Options</em> menu.</td>
</tr>
<tr>
<td>“Subheading Title”</td>
<td>References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: “Typographic Conventions.”</td>
</tr>
<tr>
<td><strong>Courier type</strong></td>
<td>Signal and port names are shown in lowercase Courier type. Examples: <em>data1</em>, <em>tdi</em>, <em>input</em>. Active-low signals are denoted by suffix <em>n</em>, e.g., <em>resetn</em>.</td>
</tr>
<tr>
<td>Anything that must be typed exactly as it appears is shown in Courier type. For example: <em>c:\qdesigns\tutorial\chiptrip.gdf</em>. Also, sections of an actual file, such as a <em>Report File</em>, references to parts of files (e.g., the <em>AHDL</em> keyword <em>SUBDESIGN</em>), as well as logic function names (e.g., <em>TRI</em>) are shown in Courier.</td>
<td></td>
</tr>
<tr>
<td>1., 2., 3., and a., b., c., etc.</td>
<td>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.</td>
</tr>
<tr>
<td>■ ●</td>
<td>Bullets are used in a list of items when the sequence of the items is not important.</td>
</tr>
<tr>
<td>✔</td>
<td>The checkmark indicates a procedure that consists of one step only.</td>
</tr>
<tr>
<td>☐</td>
<td>The hand points to information that requires special attention.</td>
</tr>
<tr>
<td>←</td>
<td>The angled arrow indicates you should press the Enter key.</td>
</tr>
<tr>
<td>■</td>
<td>The feet direct you to more information on a particular topic.</td>
</tr>
</tbody>
</table>
Section I. Introduction

This section provides information about the High-Speed Development Kit, Stratix GX Edition contents.

This section includes the following chapter:

- Chapter 1. About this Kit

Revision History

The table below shows the revision history for Chapter 1.

<table>
<thead>
<tr>
<th>Chapter(s)</th>
<th>Date / Version</th>
<th>Changes Made</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>July 2003</td>
<td>First publication.</td>
</tr>
</tbody>
</table>
1. About this Kit

General Description

The High-Speed Development Kit, Stratix GX Edition is designed for the development and rapid prototyping of applications that incorporate single-channel or multi-channel transceiver interfaces up to 3.125 gigabits per second (Gbps) and/or source-synchronous interfaces with dynamic phase alignment (DPA) up to 1 Gbps per channel. Figure 1–1 shows the development board included in the kit.

*Figure 1–1. Stratix GX Development Board*
The development kit includes the following items:

- **Stratix GX Development Board**—The development board has a variety of connectors from the Stratix GX device to external sources that you can use for a wide array of applications, including Gigabit Ethernet, 10 Gigabit Ethernet, OC-12/STM-4 and OC-48/STM-16 SONET/SDH, Fibre Channel (1 and 2 Gbps), System Packet Interface – Level 4 Phase 2 (SPI-4.2) (also known as POS-PHY Level 4), PCI Express, 8-bit RapidIO, and 1x/4x Serial RapidIO. You can use a Nios® embedded processor to control the application, or you can access an external processor via the PCI mezzanine card (PMC) interface.

- **Board Design Files**—The kit includes a complete package of schematic and layout design files along with a viewer tool, which you can use as reference to accelerate PCB design using the Stratix GX device.

- **Design Examples**—The kit includes design examples that illustrate the use of Stratix GX transceivers. You can use these design examples as a reference when implementing your proprietary backplane interface, accelerating the design, verification, and prototyping cycle.

- **Demonstrations**—The kit includes two device programming files that demonstrate a 10 Gigabit Ethernet MAC with a 10 Gigabit attachment unit interface (XAUI) and a SPI-4.2 interface connected to a 10-port multi-physical layer (PHY) device.

- **Loopback Connectors**—The kit includes 2 cards with reciprocal source-synchronous connectors wired for external loopback mode and 1 card for XCVR loopback. Additionally, the kit includes 4 loopback cards for the SFP connectors.

- **Cables**—The kit includes 4 SMA cables that you can use to connect to an external oscilloscope or use for loopback testing of quad-channel transceiver interfaces such as XAUI.

**Features**

- Supports a wide array of applications, including:
  - Gigabit and 10 Gigabit Ethernet
  - 1 and 2 Gbps Fibre Channel
  - OC-12/STM-4 and OC-48/STM-16 SONET/SDH
  - SPI-4.2
  - SDI
  - PCI Express
  - 8-bit RapidIO and 1x/4x Serial RapidIO
  - PCI
  - Proprietary backplane
Includes a broad array of external connectors, including:

- HM-Zd interconnect for SPI-4.2/DPA, compatible with Intel’s 10-Channel Gigabit Ethernet media access controller (MAC) Evaluation Board (#IXD1110)
- SMA interconnect on two LVDS/DPA channels
- SMA interconnect on four transceiver channels
- HM-Zd interconnect on four transceiver channels, compatible with Tyco’s XAUI HM-Zd Evaluation Backplane (#B024519-013)
- XPAK connector on four transceiver channels (optical module not included)
- Four small form factor pluggable (SFP) optical connectors (optics not included)
- Four high speed serial data connectors (HSSDC2)
- High-speed logic analyzer access, compatible with Agilent and Tektronix connectorless probes
- 10/100 Ethernet MAC/PHY for remote device access

Documentation

The High-Speed Development Kit, Stratix GX Edition contains the following documentation:

- *Stratix GX Development Board Data Sheet*—Describes the specifications for the board and explains how to use it. This document is printed.
- *High-Speed Development Kit, Stratix GX Edition User Guide* (this document)—Describes how to use the kit, including setting up the board, running the diagnostic tests, and describing the example designs. This document is printed.
- *Stratix GX Board Design Guidelines*—This document provides design guidelines for the board. This document is printed.
- *Stratix GX Device Family Data Sheet*—Provides the technical specifications for Stratix GX devices.
- *AN 249: Implementing 10 Gigabit Ethernet XAUI in Stratix GX Devices*—Describes the fundamentals of 10 Gigabit Ethernet and XAUI electrical specifications, and illustrates how to implement a XAUI interface in a Stratix GX device.
- *POS-PHY Level 4 MegaCore Function User Guide*—Provides the specifications for the Altera POS-PHY Level 4 MegaCore® function and explains how to use it.
- *10 Gigabit Ethernet MAC Product Brief*—Describes the features of the 10 Gigabit Ethernet MAC megafuction from AMPPSM partner MorethanIP.
- *SONET/SDH Compiler User Guide*—Provides the specifications of the Altera SONET/SDH Framer and Overhead Processor MegaCore function and explains how to use it.
- **PCI Compiler User Guide**—Provides the specifications of the Altera PCI MegaCore functions and explains how to use them.
- **DDR SDRAM Controller User Guide**—Provides the specifications of the Altera DDR SDRAM Controller MegaCore function and explains how to use it.
Section II. Getting Started

This section describes how to get started with the board, including describing the board components, explaining how to set up the board, and describing how to perform the preloaded diagnostic tests.

This section includes the following chapters:

- Chapter 2. Before You Begin
- Chapter 3. Board Setup
- Chapter 4. Run the Preloaded Diagnostic Tests

Revision History

The table below shows the revision history for these chapters.

<table>
<thead>
<tr>
<th>Chapter(s)</th>
<th>Date / Version</th>
<th>Changes Made</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 - 4</td>
<td>July 2003</td>
<td>First publication.</td>
</tr>
</tbody>
</table>
2. Before You Begin

Before using the kit or installing the software, be sure to check the contents of the kit and inspect the board to verify that you received all of the items. If any of these items are missing, contact Altera before you proceed. You should also verify that your PC meets the software and system requirements of the kit.

Development Kit Contents

The High-Speed Development Kit, Stratix GX Edition contains the following items:

- Stratix GX development board
- ByteBlaster™ II cable
- ATX power supply
- 4 SMA cables
- RS-232 cable
- 3 mini-cards with reciprocal HM-Zd connectors wired for loopback
- 4 mini-cards to loop back the SFP connectors
- Stratix GX Reference Designs & Software CD-ROM
- Quartus II Development Software CD-ROM (version 3.0)
- High-Speed Development Kit, Stratix GX Edition User Guide (this document)
- Stratix GX Development Board Data Sheet
- Stratix GX Board Design Guidelines
- Introductory letter

The Stratix GX Reference Designs & Software CD-ROM contains all of the supporting files and documentation, including:

- Stratix GX development board schematic
- Stratix GX development board layout file (Allegro format)
- Stratix GX development board layout guidelines
- Stratix GX development board test designs
- Stratix GX development board example designs and demonstrations
- 60-day evaluation versions of Altera MegaCore functions
- Related documentation in Adobe PDF
Inspect the Board

Place the board on an anti-static surface and inspect it to ensure that it has not been damaged during shipment. Verify that all components are on the board and appear intact.

The board can be damaged without proper anti-static handling. Therefore, you should take anti-static precautions before handling the board.

Refer to the Stratix GX Development Board Data Sheet for information on the board components and their locations.

Software Requirements

You should install the following software before you begin using the kit.

- Quartus® II software version 3.0 or higher.
- The software on the High-Speed Development Kit, Stratix GX Edition CD-ROM.

Refer to “Set Up Licensing” for information on obtaining licenses for the software.

Figure 2–1 shows the development kit directory structure on the Stratix GX Reference Designs & Software CD-ROM.
Altera provides archived Quartus II projects for the designs included in the kit. Before compiling an archived project, you must restore it. To restore and compile, perform the following steps:

1. Choose **Restore Archived Project** (Project menu).

2. In the **Archive name** box, type the path and file name of the Quartus II Archive File (.qar) you wish to restore, or select a QAR File with **Browse (...)**.

3. In the **Destination folder** box, type or select the path of the folder into which you wish to restore the contents of the QAR File, or select a folder with **Browse (...)**.

4. Click **Show log** to view the Quartus II Archive Log File (.qarlog) for the project you are restoring from the QAR File.

5. Click **OK**.

6. Compile the project.
Set Up Licensing

You must have a valid license to use the Quartus II development software and to compile and generate programming files for designs that include Altera MegaCore functions. The kit includes a full, 1-year license for the Quartus II development software and temporary 60-day evaluation licenses for the Altera POS-PHY Level 4, SONET/SDH Framer and Overhead Processor, PCI, and DDR SDRAM Controller MegaCore functions. To purchase full licenses for the MegaCore functions, visit the Altera web site at www.altera.com or contact your local Altera sales representative.

To design for Stratix GX devices using the Quartus II software, you need a special FEATURE line. Therefore, you have to request and install a license file before creating designs with the kit.

To obtain your Quartus II and temporary MegaCore licenses, perform the following steps:

1. Point your web browser to the Altera web site at www.altera.com/licensing.
2. Scroll to the Development Kit Licenses section of the Altera Licensing Center page.
3. Click the High-Speed Development Kit, Stratix GX Edition link.
4. Follow the on-line instructions to request your license. A license file is e-mailed to you.
5. To install your license, refer to “Specifying the License File” in the Quartus II Installation & Licensing Manual for PCs, which is included with the kit.

The 60-day period for MegaCore evaluation licenses starts from the request date and cannot be renewed. After this period, you will still be able to compile and simulate using these MegaCore functions, but you will not be able to generate programming files.

Next Steps

This user guide contains the following chapters to help you get started working with the board:

- “Board Setup” on page 3–1 explains how to setup and configure the Stratix GX development board.
- “Run the Preloaded Diagnostic Tests” on page 4–1 describes how to set up and run each preloaded design and the required equipment.
- “Perform the Production Diagnostic Tests” on page 5–1 explains how to run the production tests.
“Troubleshooting” on page 6–1 describes how to solve problems you may encounter with the test designs or with setting up the board.

“Diagnostic Test Details” on page 7–1 explains each diagnostic test, including board-level block diagrams, design-level block diagrams, and simulations or screenshots of each test.

“Stratix GX SPI-4.2 Demonstration with 10-Port Gigabit Ethernet MAC” on page 8–1 describes this provided demonstration.

“10-Gigabit Ethernet MAC Demonstration” on page 9–1 describes this provided demonstration.
The Stratix GX development board has a variety of interfaces and features. See Figure 3–1 and Table 3–1.

Figure 3–1. Features of the Stratix GX Development Kit Board
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Power connector and circuitry</td>
</tr>
<tr>
<td>2</td>
<td>Configuration switches</td>
</tr>
<tr>
<td>3</td>
<td>Divide by 20 and divide by 2 clock circuitry</td>
</tr>
<tr>
<td>4</td>
<td>Stratix Mictor connector</td>
</tr>
<tr>
<td>5</td>
<td>Stratix 20-pin test header</td>
</tr>
<tr>
<td>6</td>
<td>10/100 Ethernet interface</td>
</tr>
<tr>
<td>7</td>
<td>Expansion prototype card (PROTO1) interface</td>
</tr>
<tr>
<td>8</td>
<td>On-board flash interface</td>
</tr>
<tr>
<td>9</td>
<td>RS-232 interface for Stratix and Stratix GX devices</td>
</tr>
<tr>
<td>10</td>
<td>General-purpose interface (user pushbutton switches, dipswitches, LEDs, and displays)</td>
</tr>
<tr>
<td>11</td>
<td>Configuration circuitry</td>
</tr>
<tr>
<td>12</td>
<td>Stratix device</td>
</tr>
<tr>
<td>13</td>
<td>Source synchronous HM-Zd SPI-4.2 interface</td>
</tr>
<tr>
<td>14</td>
<td>Clock source (crystal oscillators and clock input and output)</td>
</tr>
<tr>
<td>15</td>
<td>DPA SMA interface</td>
</tr>
<tr>
<td>16</td>
<td>Edge-launched XCVR SMA connectors</td>
</tr>
<tr>
<td>17</td>
<td>XPAK interface</td>
</tr>
<tr>
<td>18</td>
<td>Tektronix high-speed differential probe for Stratix GX/Stratix device bridge</td>
</tr>
<tr>
<td>19</td>
<td>MDIO interface for XPAK</td>
</tr>
<tr>
<td>20</td>
<td>Stratix GX 20-pin test header</td>
</tr>
<tr>
<td>21</td>
<td>Agilent high-speed differential probe for source synchronous (HM-Zd SPI-4.2) interface</td>
</tr>
<tr>
<td>22</td>
<td>SFP interface</td>
</tr>
<tr>
<td>23</td>
<td>Stratix GX device</td>
</tr>
<tr>
<td>24</td>
<td>Stratix GX Mictor connector</td>
</tr>
<tr>
<td>25</td>
<td>HSSDC2 interface</td>
</tr>
<tr>
<td>26</td>
<td>Vertical launched XCVR SMA connectors</td>
</tr>
<tr>
<td>27</td>
<td>DDR SDRAM interface</td>
</tr>
<tr>
<td>28</td>
<td>Recovered clock SMA connectors</td>
</tr>
<tr>
<td>29</td>
<td>XAUI HM-Zd interface</td>
</tr>
<tr>
<td>Not shown</td>
<td>PMC interface (on the back of the board)</td>
</tr>
<tr>
<td></td>
<td>Compact flash interface (on the back of the board)</td>
</tr>
</tbody>
</table>
To power up the Stratix GX development board, you need the hardware listed in Table 3–2. See “Development Kit Contents” on page 2–1 for a list of items provided in the kit.

### Table 3–2. Required Equipment

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Manufacturer</th>
<th>Part Number</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stratix GX Development Board</td>
<td>Altera</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>ATX Power Supply</td>
<td>Sparkle Power</td>
<td>FSP250-60GTA</td>
<td>1</td>
</tr>
<tr>
<td>Programming Cable</td>
<td>Altera</td>
<td>ByteBlaster II</td>
<td>1</td>
</tr>
<tr>
<td>SMA DC Block (1)</td>
<td>Any</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>SMA 20-dB 50-Ω Attenuator (1)</td>
<td>Any</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>High-Speed Digital Sampling Oscilloscope (1), (2)</td>
<td>Tektronix</td>
<td>CSA8000</td>
<td>1</td>
</tr>
<tr>
<td>3-Foot SMA Cable (1)</td>
<td>Any</td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

*Note:*
(1) This item is required to view the XCVR eye.
(2) You can also use other oscilloscopes.

**Power Up the Board & View the XCVR Eye**

Perform the following steps to power up the Stratix GX development board. You will set switches on the board so that the Stratix GX device displays the transceiver (XCVR) eye in an oscilloscope.

Before you attempt to power up the board, make sure that you have the equipment listed in Table 3–2.

1. Set switch SW3 to off (middle position).
2. Connect one end of the ATX power supply to J31 and the other end to a power outlet.
3. Set the Stratix GX device switches as shown in Table 3–3. These settings display the XCVR eye.

### Table 3–3. Stratix GX XCVR Eye Switches Settings

<table>
<thead>
<tr>
<th>SW1</th>
<th>SW2</th>
<th>SW4</th>
<th>SW5</th>
<th>SW8</th>
<th>J48</th>
<th>J90</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>NO</td>
<td>EPC16</td>
<td>STD</td>
<td>000</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

4. Set switch SW3 to the on position (all of the way up). All of the board LEDs illuminate.
The Stratix GX development board is powered up. To view the Stratix GX XCVR eye, perform the following steps:

1. Place a DC block on the signals $\text{TX}_N$ (J25) and $\text{TX}_P$ (J17).
2. Place the SMA attenuator on J97.
3. Connect one end of an SMA cable to the attenuator and connect the other end to the trigger input of the oscilloscope.
4. Connect one end of an SMA cable to J25 and connect the other end to an input channel of the oscilloscope.
5. Connect one end of an SMA cable to J17 and connect the other end to an input channel of the oscilloscope.
6. Set the switch SW9 to select the crystal oscillator.
7. On the oscilloscope, set the vertical setting on each channel to 52 mV/div.
8. Set the oscilloscope’s horizontal setting to 100 ps/div.
9. On the oscilloscope’s trigger section, press the SET TO 50% button.
10. Turn on and set the math channel to C2-C1.
11. Turn off channels 1 and 2. The oscilloscope should capture an eye pattern similar to the one shown in Figure 3–2.
12. Stratix GX devices can dynamically control the transceiver’s VOD and pre-emphasis settings. To view this control, perform the following steps.

a. Press the adjust pre-emphasis pushbutton (S4). You should see the pre-emphasis change on the oscilloscope. There are 6 pre-emphasis settings: pressing S4 5 times in succession cycles through the settings. Pressing S4 a sixth time sets the transceiver back to the original setting. The value of the setting is indicated by the left digit of the Stratix GX seven segment display (D9).

b. Press the adjust VOD pushbutton (S5). You should see the VOD change on the oscilloscope. There are 6 VOD settings: pressing S5 5 times in succession cycles through the settings. Pressing S5 a sixth time sets the transceiver back to the original setting. The value of the setting is indicated by the right digit of the Stratix GX seven segment display (D9).
4. Run the Preloaded Diagnostic Tests

Each interface on the Stratix GX development board has an associated diagnostic test that exercises the interface at the supported I/O rates. Although the tests are not exhaustive, they help you confirm that each interface runs according to its intended design. A subset of the diagnostic tests are loaded into the Stratix GX development board’s configuration devices, making it easy for you to run any of the tests when you power up the board. This chapter describes how to set up and perform these preloaded diagnostic tests, including:

- User I/O
- Stratix GX Double Data Rate (DDR) SDRAM Interface
- Stratix GX HM-Zd SPI-4.2 Loopback
- Stratix GX HM-Zd XCVR Loopback
- Stratix GX SFP XCVR Loopback
- Stratix GX XCVR Eye Diagram

The following sections describe how to perform each test, including the equipment you need to perform each test, how to set up the board, and the test procedure. Table 4–1 shows the switch settings you must make to load the designs on power up. For more details on these designs, refer to Chapter 7, Diagnostic Test Details.

<table>
<thead>
<tr>
<th>Diagnostic Test</th>
<th>SW1</th>
<th>SW2</th>
<th>SW4</th>
<th>SW5</th>
<th>SW7</th>
<th>SW8 (1)</th>
<th>J48</th>
<th>J90</th>
</tr>
</thead>
<tbody>
<tr>
<td>XCVR Eye Diagram</td>
<td>YES</td>
<td>NO</td>
<td>EPC16</td>
<td>STD</td>
<td>---</td>
<td>000</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DDR</td>
<td>YES</td>
<td>NO</td>
<td>EPC16</td>
<td>STD</td>
<td>---</td>
<td>011</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>HM-Zd SPI-4.2</td>
<td>YES</td>
<td>NO</td>
<td>EPC16</td>
<td>STD</td>
<td>---</td>
<td>001</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>HM-Zd XCVR Loopback</td>
<td>YES</td>
<td>NO</td>
<td>EPC16</td>
<td>STD</td>
<td>---</td>
<td>001</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SFP XCVR Loopback</td>
<td>YES</td>
<td>NO</td>
<td>EPC16</td>
<td>STD</td>
<td>---</td>
<td>001</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>User I/O</td>
<td>NO</td>
<td>NO</td>
<td>EPC16</td>
<td>STD</td>
<td>111</td>
<td>111</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

**Note:**
(1) Slider 3 is the MSB and slider 1 is the LSB. A slider in the up position is 0 and a slider in the down position is 1.

**User I/O**

The user I/O design tests all of the user LEDs, dipswitches, pushbuttons, and 7-segment displays on the board (the Stratix and Stratix GX devices each have a set). One pair of pushbuttons controls each device’s user
LEDs, while the other pair increments or decrements the counter shown on the 7-segment display. The dipswitches enter numbers on each device’s 7-segment display.

**Required Hardware**

In addition to the board, you need the Altera-provided ATX power supply.

**Test Setup**

Perform the following steps to set up the user I/O test.

1. Set the switches as shown in Table 4–1 for the user I/O test.

2. Connect the power supply to the board.

3. Confirm that the Stratix and Stratix GX devices have finished configuration (the GX_CONF_DONE (D7) and S_CONF_DONE (D6) LEDs illuminate as shown in Figure 4–1).

![Figure 4–1. GX_CONF_DONE, S_CONF_DONE & User I/O LEDs](image)
Run the User I/O Test

Perform the following steps to execute the test for the Stratix GX device user I/O.

1. Set all of the Stratix GX user dipswitches (S11) to 0.

2. Press GX_CLR (S14) to reset the design. When the pushbutton is pressed, the GX_CLR LED (D16) illuminates. When you release the pushbutton, the 7-segment display (D9) shows 00.

3. Press the Stratix GX channel pushbutton switches, Stratix_GX_PB_0 (S2) for up and Stratix_GX_PB_1 (S3) for down. The channel numbers on the Stratix GX 7-segment display (D9) increment and decrement as you press S2 and S3, respectively.

4. Press S14 once.

5. Press S2 three times. D9 displays 03.


7. Press the Stratix GX volume pushbutton switches, Stratix_GX_PB_2 (S4) for up and Stratix_GX_PB_3 (S5) for down. As you press S4 and S5, the number of Stratix_GX LEDs (D10-15) that are illuminated should increment and decrement, respectively.

8. Press S14 once.

9. Press S4 six times. All of the Stratix_GX LEDs illuminate.

10. Press S5 six times. All of the Stratix_GX LEDs turn off.

11. Using the Stratix GX user dipswitches (S11), enter the binary number 15.

   a. Set dipswitches 4-1 to 1.

   b. Set dipswitch 7 to the up position and then to the down position to enter the value. D9 displays the number 15.

12. Enter the binary number 127.

   a. Set dipswitches 7-1 to 1.

   b. Set dipswitch 8 to the up position and then to the down position to enter the value. D9 displays the number 99.
You are finished testing the Stratix GX device user I/O. Perform the following steps to execute the test for the Stratix device user I/O.

13. Set all of the Stratix user dipswitches (D6) to 0.

14. Press S_CLR (S12) to reset the design. The S_CLR LED (D24) illuminates and the 7-segment display (D8) shows 00.

15. Press the Stratix channel pushbutton switches, Stratix_PB_0 (S8) for up and Stratix_PB_1 (S9) for down. The channel numbers on the 7-segment display (D8) increment and decrement.

16. Press S12 once.

17. Press S8 three times. D8 displays 03.


19. Press the Stratix volume pushbutton switches, Stratix_PB_2 (S10) for up and Stratix_PB_3 (S7) for down. The number of illuminated Stratix LEDs increment and decrement.

20. Press S12 once.

21. Press S10 six times. All of the LEDs illuminate.

22. Press S7 six times. All of the LEDs turn off.

23. Enter the binary number 15.
   a. Set dipswitches 4-1 to 1.
   b. Set dipswitch 8 to the up position and then to the down position to enter the value. D8 displays the number 15.

24. Enter the binary number 127.
   a. Set dipswitches 7-1 to 1.
   b. Set dipswitch 8 to the up position and then to the down position to enter the value. D8 displays the number 99.

You have finished the user I/O test.
The Stratix GX DDR SDRAM interface connects to a 184-pin, 200-MHz Micron DDR DIMM module. To test this interface, you use a Nios embedded processor-based test that you run from the SOPC Builder SDK Shell. You observe the test output in the shell. An RS-232 cable connected to your PC’s COM 1 port allows communication between the board and the software running on the PC. The RS-232 connector serves the Stratix GX device when the serial cable is attached to the bottom connector and the Stratix device when the serial cable is attached to the top connector as shown in Figure 4–2.

**Figure 4–2. RS-232 Connector**

---

**Required Hardware & Software**

The Stratix GX DDR SDRAM test design uses the following equipment and software:

- DDR DIMM module
- RS-232 cable
- ATX power supply
- Nios embedded processor version 2.0 or higher
Test Setup

Perform the following steps to set up the test.

1. Remove power from the board.

2. Insert the DDR DIMM module in the location defined in Figure 4–3.

![Figure 4–3. Inserting DDR DIMM](image)

3. Attach one end of the RS-232 cable to your PC’s COM 1 port and the other end to the bottom connector of the RS-232 serial port on the board.

4. Set the board’s switches as shown in Table 4–1 for this test.

5. Supply power to the board.

6. Confirm that the Stratix GX device has finished configuration (the GX_CONF_DONE (D7) LED illuminates).

Run the Stratix GX DDR SDRAM Interface Test

Perform the following steps to run the test.

1. Choose Programs > Altera > SOPC Builder <version> > SOPC Builder SDK Shell (Windows Start menu).

   This shell may be named Nios SDK Shell in older versions of the Nios embedded processor.

2. Change to the Nios SOFs directory by typing the following command:

   ```
   cd /Stratix_GX_kit/Test_designs/Nios_test_designs/Nios_sofs
   ```

3. Type `nr -t` to open a terminal window that connects to COM 1.
4. Press the Enter key twice in the terminal window. You should be at a Nios prompt.

5. Confirm that you are connected to the Nios processor running on the board by pressing the Enter key a few times in the terminal window to display the processor’s memory contents.

6. Exit the terminal window by pressing the Ctrl + C keys.

7. Type `nr DDR.srec` to start the DDR test. Observe the results to see if any errors are reported. Figure 4–4 shows the text that should display in the shell.

---

### Figure 4–4. DDR Test Display in SOPC Builder SDK Shell

```plaintext
= Altera GX Development Board DDR SDRAM Demonstration =
  - Nios v2.2  (66MHz)  =
  - Altera DDR SDRAM Controller v1.1.0 (200MHz)  =
  - Avalon test interface to DDR SDRAM Controller =

Doing a sequence of 100 back-to-back writes (of 8) followed by 100 back-to-back reads (of 8)
  Repeating this test 20000 times
  Testing... [####################]

Walking ones on the addresses test
  Repeating this test 20000 times
  Testing... [####################]

Burst length test (burst reads and writes of 2,4 and 8)
  5 sequences of 4 write & read bursts
  Repeating this test 20000 times
  Testing... [####################]

Entire DIMM write and read test
  Write the entire DIMM with the address of each location,
  Then read each location to check that the address and data are the same.

  Writing... [####################]
  Finished writing.
  Reading... [####################]
  Read and checked 8388608 locations (128 Mb).

= END OF TEST =
= Type "g0" and press Enter to restart the test =
```
8. The test continues until you press the Ctrl + C keys. Press these keys to stop the status display in the SOPC Builder SDK Shell.

9. Press S2 to stop the test.

Stratix GX HM-Zd SPI-4.2 Loopback

The Stratix GX HM-Zd SPI-4.2 loopback test design tests the SPI-4.2 source synchronous signals at up to 1 Gbps using the provided HM-ZD SPI-4.2 loopback cards.

Required Hardware

This test requires the following hardware:

- HM-Zd SPI-4.2 loopback cards (labeled J108 and J109)
- ATX power supply

Test Setup

Perform the following steps to set up the test.

1. Remove power from the board.

2. Install the HM-Zd SPI-4.2 loopback cards as shown in Figure 4–5.

3. Set the board’s switches as shown in Table 4–1 for this test.
Run the Preloaded Diagnostic Tests

4. Supply power to the board.

5. Confirm that the Stratix GX device has finished configuration (the GX_CONF_DONE (D7) LED illuminates).

Run the Stratix GX HM-Zd SPI-4.2 Loopback Test

Perform the following steps to run the test.

1. Set the Stratix GX dipswitch 5 to 1 (up position).

2. Press the reset pushbutton switch (GX_CLR).

3. Press the start pushbutton switch (Stratix_GX_PB_0). Several LEDs illuminate.

   - LED0 is the high-speed data match LED. If it illuminates, the test is successful.
   - LED1 indicates that the GX device has received the data.
LED2 indicates a per channel match. This LED is used for the per channel feature.

LED3 indicates that the test has started.

LED4 is the match LED for the low-speed control signals.

LED5 is the error indicator.

4. Press the stop pushbutton switch (GX PB1) to stop the device from transmitting data.

5. Press the reset pushbutton switch (GX_CLR).

6. Press the Stratix_GX_PB_0.

7. Inject an error by pressing Stratix_GX_PB_2 (S4) once. The Stratix GX 7-segment display (D9) shows 01 and LED5 illuminates.

8. Inject two more errors by pressing S4 twice. D9 displays 03.

The Stratix GX HM-Zd XCVR loopback test design tests the HM-Zd XCVR signals at up to 3.125 Gbps using the provided HM-Zd XCVR loopback card.

**Required Hardware**

This test requires the following hardware:

- HM-Zd XCVR loopback card (labeled J1)
- ATX power supply

**Test Setup**

Perform the following steps to set up the test.

1. Remove power from the board.

2. Install the HM-Zd XCVR loopback card as shown in Figure 4–6.
3. Set the board’s switches as shown in Table 4–1 for this test.

4. Supply power to the board.

5. Confirm that the Stratix GX device has finished configuration (the GX_CONF_DONE (D7) LED illuminates).

Run the Stratix GX HM-Zd XCVR Loopback Test

Perform the following steps to run the test.

1. Set Stratix GX user dipswitches 5 and 6 to the up position to set the XCVR setting to HM-Zd.

2. Press the reset pushbutton switch (GX_DEV_CLR) to initialize the design. The GX_DEV_CLR LED illuminates.

3. Press the start pushbutton switch (GX PB0). Several LEDs illuminate.
   - LED0, LED1, LED2, and LED3 are the match lights on a per channel basis. They illuminate if the test is successful.
   - LED4 indicates that the test is running.
   - LED5 is the error signal.

4. Press the stop pushbutton switch (GX PB1) to stop the device from transmitting.

5. Press the reset pushbutton switch (GX_DEVCLR).
6. Press the start pushbutton switch (Stratix_GX_PB_0).

7. Inject an error by pressing Stratix_GX_PB_2 (S4) once. The Stratix GX 7-segment display (D9) shows 01 and Stratix GX LED5 illuminates.

8. Inject two more errors by pressing S4 twice. D9 displays 03.

---

**Stratix GX SFP XCVR Loopback**

The Stratix GX SFP XCVR loopback test design tests the SFP XCVR signals at up to 2.488 Gbps using the provided SFP XCVR loopback cards.

**Required Hardware & Software**

This test requires the following hardware and software:

- SFP XCVR loopback cards
- ATX power supply

**Test Setup**

To set up the test, perform the following steps.

1. Remove power from the board.

2. Insert the SFP loopback cards on the board as shown in Figure 4–6.

3. Attach the SFP loopback card (labeled J3 on the card) to the SFP transceiver connectors at J54, J64, J45, and J38 as shown in Figure 4–7.
Run the Preloaded Diagnostic Tests

4. Set the board’s switches as shown in Table 4–1 for this test.

5. Supply power to the board.

6. Confirm that the Stratix GX device has finished configuration (the GX_CONF_DONE (D7) LED illuminates).

Run the Stratix GX SFP XCVR Loopback Test

Perform the following steps to run the test.

1. Set Stratix GX user dipswitches 5 and 6 to the down position to set the XCVR setting to SFP.

2. Press the reset pushbutton switch (GX_DEV CLR) to initialize the design. The GX_DEV_CLR LED illuminates.

3. Press the start pushbutton switch (GX_PB0). Several LEDs illuminate.
   - LED0, LED1, LED2, and LED3 are the match lights on a per channel basis. They illuminate if the test is successful.
   - LED4 indicates that the test is running.
   - LED5 is the error signal.

4. Press the stop pushbutton switch (GX_PB1) to stop the device from transmitting.
5. Press the reset pushbutton switch (GX_DEVCLR).

6. Press the start pushbutton switch (Stratix_GX_PB_0).

7. Inject errors by pressing Stratix_GX_PB_2 (S4) once. The Stratix GX 7-segment display (D9) shows 01 or more and Stratix GX LED5 illuminates. Because this test runs all channels at once, the Stratix GX 7-segment display (D9) may show more than one error when you press Stratix_GX_PB_2 (S4).

8. Inject two more errors by pressing S4 twice. D9 displays 03 or more.

**Stratix GX XCVR Eye Diagram**

The Stratix GX XCVR eye test design produces an eye on an oscilloscope at 3.125 Gbps using the on-board crystal oscillator and SMA cables. This design also shows the dynamic control of the Stratix GX device for both the pre-emphasis and VOD settings using the pushbutton switches S4 and S5. For this design, the Stratix GX transceiver block has the parameter settings shown in Table 4–2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Rate</td>
<td>3.125 GHz</td>
</tr>
<tr>
<td>Clock In</td>
<td>156.25 MHz</td>
</tr>
<tr>
<td>PLL Clock</td>
<td>156.25 MHz</td>
</tr>
<tr>
<td>PLL DC Coupling</td>
<td>Disabled</td>
</tr>
<tr>
<td>PLL Bandwidth</td>
<td>High</td>
</tr>
<tr>
<td>PPM Threshold</td>
<td>1000</td>
</tr>
<tr>
<td>8b/10b Encoder/Decoder</td>
<td>Bypassed</td>
</tr>
<tr>
<td>Run Length Violation</td>
<td>Bypassed</td>
</tr>
<tr>
<td>Rx Bandwidth</td>
<td>Disabled</td>
</tr>
<tr>
<td>Word Alignment Mode</td>
<td>Manual</td>
</tr>
<tr>
<td>Alignment Pattern</td>
<td>0101111100</td>
</tr>
</tbody>
</table>

To perform this test, refer to “Power Up the Board & View the XCVR Eye” on page 3–3.
Each interface on the Stratix GX development board has an associated diagnostic test that exercises the interface at the supported I/O rates. Although the tests are not exhaustive, they help you confirm that each interface runs according to its intended design. The production tests execute each diagnostic test. Altera executes the production tests on every Stratix GX development board to confirm that all of the interfaces pass.

This section includes the following chapters:

- Chapter 5. Perform the Production Diagnostic Tests
- Chapter 6. Troubleshooting
- Chapter 7. Diagnostic Test Details

**Revision History**

The table below shows the revision history for these chapters.

<table>
<thead>
<tr>
<th>Chapter(s)</th>
<th>Date / Version</th>
<th>Changes Made</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 - 7</td>
<td>July 2003</td>
<td>First publication.</td>
</tr>
</tbody>
</table>
5. Perform the Production Diagnostic Tests

Required Hardware & Software

The following list describes the hardware and software you need to run the production diagnostic tests.

- Windows-based PC
- Quartus II software
- Parallel port cable
- Serial port cable
- PMC card (part number 853-10010 PMC100TX)
- 10/100-Mbit Ethernet network-interface daughter board (part number NIOS-EDKX)
- Six to ten short (1-foot) SMA cables
- High-Speed Development Kit, Stratix GX Edition
  - Stratix GX development board revision C
  - ATX power supply
  - ByteBlaster II cable
  - Programming files
  - Four long (3-foot) SMA cables
  - HM-Zd loopback cards
  - SFP loopback card
  - Micron DDR DIMM
  - Compact flash card

Set Up the Board

The following procedure describes how to set up the board, including attaching the daughter cards and cables.

1. Install the PMC card on the back side of the board as shown in Figure 5–1.
Figure 5–1. Attach the PMC Card

2. Attach the HM-Zd SPI-4.2 loopback card (labeled J108 on the card) to the connector at J108, which is located on the bottom edge of the board as shown in Figure 5–2.

3. If your version of the Stratix GX development board has an EP1SGX40GF1020C5 device, you may also need to attach a second HM-Zd loopback card (labeled J109 on the card) to the connector at J109 for the optional differential status signals.

Figure 5–2. Attach the HM-Zd Loopback Card
4. Install six short SMA cables to the DPA SMA connectors:
   - Connect DPA_CLKOUT (P & N) to DPA_CLKIN
   - Connect DPA_TX0 (P & N) to DPA_RX0
   - Connect DPA_TX1 (P & N) to DPA_RX1

See Figure 5–3.

Take care to connect the cables to the proper connectors and ensure that the cables are completely seated.

5. Install eight SMA cables to the transceiver SMA connectors as shown in Figures 5–4 and 5–5. Do not overtighten the connectors (using your fingers is good enough). Connect TX_P[3:0] to RX_P[3:0] and TX_N[3:0] to RX_N[3:0].

Take care to connect the cables to the proper connectors (there are 2 sets of connectors, 8 vertical launch and 8 edge launch). Channels 0 and 1 are the edge launch and Channels 2 and 3 are the vertical launch connectors.
6. Attach the HM-Zd XCVR loopback card (labeled J1 on the card) to the HM-Zd transceiver connector as shown in Figure 5–6.
Figure 5–6. Attach the XCVR HM-Zd Loopback Card

7. Attach the SFP loopback card (labeled J3 on the card) to the SFP transceiver connector at J54 as shown in Figure 5–7.

Figure 5–7. Attach the XCVR SFP Module

8. Attach the XPAK module to the XPAK transceiver connector at U27 as shown in Figure 5–8.
9. Attach the HSSDC2 loopback cables to the HSSDC2 transceiver connectors at J23 through J32 and J15 through J11 as shown in Figure 5–9.
10. Attach the DDR DIMM module to the DDR connector shown in Figure 5–10.

11. Attach the 10/100-Mbit Ethernet network-interface daughter card to the board as shown in Figure 5–11.
12. Attach the one end of the parallel cable to the PC’s parallel port if it is not already attached.

13. Connect one end of the ByteBlaster II cable to the parallel cable.

14. Connect the other end of the ByteBlaster II cable to the JTAG header (J87) on the Stratix GX development board, as shown in Figure 5–12. Align the red edge of the cable with pin 1.

**Figure 5–12. Attach the ByteBlaster II Cable**

15. Move the power selection switch (SW3) to the middle position (off).

16. Attach the main power cable from the ATX supply to the power connector (J31) on the board.

17. Set the Stratix bypass (SW1) and Stratix GX bypass (SW2) switches to NO.

18. Set the configuration switch (SW4) to PS_HDR as shown in Figure 5–13.
19. Set switches SW6 and SW9 to OSC, which is the up position.

20. Set the user dipswitches, S6 and S11, to zero, which is the down position.

Figure 5–14 shows the board after you have attached all of the daughter cards and cables, and set all of the switches.
21. Power up the board by moving switch SW3 to the on position. The board power ups with all of the user LEDs illuminated and the 7-segment displays turned on.

The board is now set up and you are ready to perform the tests.

**Perform the Standard Tests**

In this section you will perform all of the standard tests. For each test, you will load a SOF onto the board. These SOFs are located in the `Stratix_GX_kit\test_designs\standard_test_designs\standard_sofs` directory.

![Figure 5–14. Board Completely Set Up](image)

Table 5–1 on page 5–36 is a worksheet for the tests. You should have a printed copy of this worksheet when you perform the tests so that you can note the test results in the appropriate row/column in the worksheet.

**User I/O**

For this test, perform the following steps.

1. Run the Quartus II software. Keep the software open until you finish all of the tests.

2. Choose Programmer (Tools menu). Leave the Programmer open until you finish all of the tests.
3. Click Auto-Detect. The Stratix, Stratix GX, and 2 EPC16 devices display in the JTAG chain.

4. Change the programming file for the Stratix GX device:
   a. Right-click the filename next to the Stratix GX device.
   b. Browse to the standard SOFs directory.
   c. Select the file GX40_User_IO.sof.
   d. Click Open. Wait until the checksum field is updated.

5. Change the programming file for the Stratix device to STX40_User_IO.sof using the steps described above. Wait until the checksum field is updated.

6. Turn on the Program/Configure option for the Stratix and Stratix GX devices. See Figure 5–15.

7. Click Start to configure both devices. When configuration is complete for both devices, the GX_CONF_DONE (D7) and S_CONF_DONE (D6) LEDs illuminate as shown in Figure 4–1 on page 4–2.

8. Test the user I/O for the Stratix GX device by performing steps 1 through 12 in “Run the User I/O Test” on page 4–3.
9. If all of the tests pass, write “PASS” in the column under Stratix GX in the User I/O row in Table 5–1 on page 5–36.

10. Test the user I/O for the Stratix device by performing steps 13 through 24 in “Run the User I/O Test” on page 4–3.

11. If all of the tests pass, write “PASS” in the column under Stratix in the User I/O row in the worksheet (Table 5–1 on page 5–36).

**Stratix GX-to-Stratix Bridge**

For this test, perform the following steps in the Programmer.

1. Change the programming file for the Stratix GX device to GX40_Bridge.sof.

2. Change the programming file for the Stratix device to STX40_Bridge.sof. Wait until the checksum field is updated.

3. Turn on the Program/Configure option for the Stratix and Stratix GX devices.

   Figure 5–16 shows the Programmer after you have done these steps.

   **Figure 5–16. Quartus II Programmer for the Stratix GX-to-Stratix Bridge Test**

4. Click Start to configure both devices. When configuration is complete for both devices, the GX_CONF_DONE (D7) and S_CONF_DONE (D6) LEDs illuminate.
5. Although this design uses both the Stratix and Stratix GX devices, the Stratix GX device controls it. Set all of the Stratix GX user dipswitches to 0.

6. Press GX_CLR (S14) to reset the design. The GX_CLR (D16) LED illuminates and the Stratix GX LEDs 0 through 4 (D10, D11, D12, D13, and D14) should be off. Additionally, the Stratix LEDs 0 and 1 (D18 and D19) illuminate while the rest are off.

7. Press the start pushbutton switch (Stratix_GX_PB_0). Several LEDs illuminate.
   - Stratix GX LED0 is the match LED. If it illuminates, the test is successful.
   - Stratix GX LED1 indicates the test has started.
   - Stratix GX LED2 indicates that the GX device is transmitting to the Stratix device.
   - Stratix GX LED3 indicates that the GX device has received the data from the Stratix device.
   - Stratix GX LED5 illuminates if there is an error.

8. Press the stop pushbutton switch (Stratix_GX_PB_1) to stop the device from transmitting.

9. Press the reset pushbutton switch (GX_CLR).

10. Press the start pushbutton switch (Stratix_GX_PB_0).

11. Inject an error by pressing Stratix_GX_PB_2 (S4) once. The number 01 displays on the Stratix GX 7-segment display (D9) and Stratix GX LED5 illuminates.

12. Inject two more errors by pressing S4 twice. 03 displays on the Stratix GX 7-segment display (D9).

13. Press start and then reset three times to insure that the test is working.

14. If all of these tests pass, write the word “PASS” in the column under Stratix and Stratix GX in the row Stratix GX-to-Stratix Bridge in the worksheet.
Source Synchronous HM-Zd Interface (Stratix GX HM-Zd SPI 4.2)

For this test, perform the following steps in the Programmer.

1. Change the programming file for the Stratix GX device to GX40_SS_HMZd.sof. Wait until the checksum field is updated.

2. Turn on the Program/Configure option for the Stratix GX device.

Figure 5–17 shows the Programmer after you have done these steps.

![Figure 5–17. Quartus II Programmer for the Source-Synchronous HM-Zd Test](image)

3. Click Start. When configuration is complete, the GX_CONF_DONE (D7) LED illuminates.

4. Press the reset pushbutton switch (GX_CLR).

5. Press the start pushbutton switch (Stratix_GX_PB_0). Several LEDs illuminate.
   - LED0 is the high-speed data and control signal match LED. If it illuminates, the test is successful.
   - LED1 indicates that the GX device has received the data.
   - LED2 indicates a per channel match. This LED is used for the per channel feature.
   - LED3 indicates that the test has started.
   - LED4 is the match LED for the low-speed control signals.
   - LED5 is the error indicator.
6. Press the stop pushbutton switch (Stratix_GX_PB1) to stop the device from transmitting.

7. Press the reset pushbutton switch (GX_CLR).

8. Press the start pushbutton switch (Stratix_GX_PB_0).

9. Inject an error by pressing Stratix_GX_PB_2 (S4) once. The number 01 displays on the Stratix GX 7-segment display (D9) and Stratix GX LED5 illuminates.

10. Inject two more errors by pressing Stratix_GX_PB_2 (S4) twice. 03 displays on the Stratix GX 7-segment display (D9).

11. Press start and then reset three times to insure that the test is working.

12. If all of these tests pass, write the word “PASS” in the column under Stratix GX in the row Stratix GX HM-Zd (SPI 4.2) in the worksheet.

**Source Synchronous DPA SMA interface (Stratix GX SMA DPA)**

For this test, perform the following steps in the **Programmer**.

1. Click **Auto-Detect** to display all of the devices in the JTAG chain.

2. Change the programming file for the Stratix GX device to **GX40_SS_DPA.sof**. Wait until the checksum field is updated.

3. Turn on the **Program/Configure** option for the Stratix GX device.

   Figure 5–18 shows the **Programmer** after you have done these steps.
4. Click **Start**. When configuration is complete, the GX_CONF_DONE (D7) LED illuminates.

5. Press the reset pushbutton switch (GX_CLR) to initialize the design. The GX_DEV_CLR LED illuminates.

6. Press the start pushbutton switch (GX PB0). Several LEDs illuminate.
   - LED0 is the match LED; it illuminates if the test is successful. LED1 indicates that the test is running.
   - LED2 indicates that the device has received the data through the cables.
   - LED3 is the per channel match. This LED is used for the per channel feature.
   - LED4 is turned off.
   - LED5 is the error signal.

7. Press the stop pushbutton switch (Stratix_GX_PB1) to stop the device from transmitting.

8. Press the reset pushbutton switch (GX_CLR).

9. Press the start pushbutton switch (Stratix_GX_PB_0).

10. Inject an error by pressing Stratix_GX_PB_2 (S4) once. The number 01 displays on the Stratix GX 7-segment display (D9) and Stratix GX LED5 illuminates.
11. Inject two more errors by pressing Stratix_GX_PB_2 (S4) twice. 03 displays on the Stratix GX 7-segment display (D9).

12. Press start and then reset three times to insure that the test is working.

13. If all of these tests pass, write the word “PASS” in the column under Stratix GX in the row Stratix GX SMA DPA in the worksheet.

**Gigabit Transceivers with SMA Interface (Stratix GX SMA XCVR)**

For this test, perform the following steps in the Programmer.

1. Change the programming file for the Stratix GX device to GX40_XCVRs.sof. Wait until the checksum field is updated.

2. Turn on the Program/Configure option for the Stratix GX device.

   Figure 5–19 shows the Programmer after you have done these steps.

   ![Figure 5–19. Quartus II Programmer for the Gigabit Transceiver Test](image)

3. Click Start. When configuration is complete, the GX_CONF_DONE (D7) LED illuminates.

4. Change the XCVR setting to SMA by setting the Stratix GX user dipswitch 6 to the up position and dipswitch 5 to the down position.

5. Press the reset pushbutton switch (GX_DEV CLR) to initialize the design. The GX_DEV_CLR LED illuminates.
6. Press the start pushbutton switch (GX PB0). Several LEDs illuminate.

   - LED0, LED1, LED2, and LED3 are the match LEDs on a per channel basis; they illuminate if the test is successful.
   - LED4 indicates that the test is running.
   - LED5 is the error signal.

7. Press the stop pushbutton switch (GX PB1) to stop the device from transmitting.

8. Press the reset pushbutton switch (GX_CLR).

9. Press the start pushbutton switch (Stratix_GX_PB_0).

10. Inject an error by pressing Stratix_GX_PB_2 (S4) once. The number 01 displays on the Stratix GX 7-segment display (D9) and Stratix GX LED5 illuminates.

11. Inject two more errors by pressing Stratix_GX_PB_2 (S4) twice. 03 displays on the Stratix GX 7-segment display (D9).

12. Press start and then reset three times to insure that the test is working.

13. If all of these tests pass, write the word “PASS” in the column under Stratix GX in the row Stratix GX SMA XCVR in the worksheet.

**Gigabit Transceivers with HM-Zd Interface (Stratix GX HM-Zd XCVR)**

For this test, perform the following steps.

1. Change the XCVR setting to HM-Zd by moving the Stratix GX user dipswitch 6 to the up position and dipswitch 5 to the up position.

2. Press the reset pushbutton switch (GX_DEV CLR) to initialize the design. The GX_DEV_CLR LED illuminates.

3. Press the start pushbutton switch (GX PB0). Several LEDs illuminate.

   - LED0, LED1, LED2, and LED3 are the match LEDs on a per channel basis; they illuminate if the test is successful.
   - LED4 indicates that the test is running.
   - LED5 is the error signal.
Perform the Production Diagnostic Tests

4. Press the stop pushbutton switch (GX PB1) to stop the device from transmitting.

5. Press the reset pushbutton switch (GX_CLR).

6. Press the start pushbutton switch (Stratix_GX_PB_0).

7. Inject an error by pressing Stratix_GX_PB_2 (S4) once. The number 01 displays on the Stratix GX 7-segment display (D9) and Stratix GX LED5 illuminates.

8. Inject two more errors by pressing Stratix_GX_PB_2 (S4) twice. 03 displays on the Stratix GX 7-segment display (D9).

9. Press start and then reset three times to insure that the test is working.

10. If all of these tests pass, write the word “PASS” in the column under Stratix GX in the row Stratix GX HM-Zd XCVR in the spreadsheet.

Gigabit Transceivers with SFP Interface (Stratix GX SFP XCVR)

For this test, perform the following steps.

1. Change the XCVR setting to SFP by setting the Stratix GX user dipswitch 6 to the down position and dipswitch 5 to the down position.

2. Press the reset pushbutton switch (GX_DEV CLR) to initialize the design. The GX_DEV_CLR LED illuminates.

3. Press the start pushbutton switch (GX PB0). The LED for channel 0 (LED0) illuminates.

4. Press start and then reset several times to insure that the test is working. LED0, LED1, LED2, and LED3 are the match lights on a per channel basis.

5. Press the reset pushbutton switch.

6. Remove the SFP card.

7. Re-attach the card in the connector at J64.

8. Press the start pushbutton switch. LED1 illuminates if the test is successful.
9. Repeat the process (press reset, move the card, and press start) for connector J45. LED2 illuminates.

10. Repeat the process once more for connector J38. LED3 illuminates. LED4 indicates that the test is running. LED5 is the error signal.

11. Press the reset pushbutton switch (GX_CLR).

12. Press the start pushbutton switch (Stratix_GX_PB_0).

13. Inject an error by pressing Stratix_GX_PB_2 (S4) once. The number 01 displays on the Stratix GX 7-segment display (D9) and Stratix GX LED5 illuminates.

14. Press start and then reset three times to insure that the test is working.

15. Inject two more errors by pressing Stratix_GX_PB_2 (S4) twice. 03 displays on the Stratix GX 7-segment display (D9).

16. If all of these tests pass, write the word “PASS” in the column under Stratix GX in the row Stratix GX SFP XCVR in the worksheet.

Gigabit Transceivers with XPAK Interface (Stratix GX XPAK XCVR)

For this test, perform the following steps.

This test requires an XPAK module, which is not included with the kit.

1. Change the XCVR setting to XPAK by setting the Stratix GX user dipswitch 6 to the down position and dipswitch 5 to the up position.

2. Press the reset button (GX_DEV CLR) to initialize the design. The GX_DEV_CLR LED illuminates.

3. Press start (GX PB0). Several LEDs illuminate.

   - LED0 is the match LED
   - LED1 is the channel synchronization LED
   - LED2 is the channel alignment LED
   - LED3 is the data valid LED
   - LED4 indicates that the test is running
   - LED5 is the error signal

4. Press stop (GX PB1) to stop the device from transmitting.
5. Press reset (GX_CLR).

6. Press start (Stratix_GX_PB_0).

7. Inject an error by pressing Stratix_GX_PB_2 (S4) once. The number 01 displays on the Stratix GX 7-segment display (D9) and Stratix GX LED5 illuminates.

8. Inject two more errors by pressing Stratix_GX_PB_2 (S4) twice. 03 displays on the Stratix GX 7-segment display (D9).

9. Press start and then reset three times to insure that the test is working.

10. If all of these tests pass, write the word “PASS” in the column under Stratix GX in the row Stratix GX XPAK XCVR in the worksheet.

**Gigabit Transceivers with HSSDC2 Interface (Stratix GX HSSDC2 XCVR) (GX40 Device Only)**

For this test, perform the following steps.

1. Change the XCVR setting to HSSDC2 by setting the Stratix GX user dipswitch 6 to the down position, dipswitch 5 to the down position, and dipswitch 4 to the up position.

2. Press reset (GX_DEV_CLR). This will initialize the design. The GX_DEV_CLR LED should light up.

3. Press start (GX PB0). Several LEDs illuminate. LED0, LED1, LED2, and LED3 are the match lights on a per channel basis.

4. Press reset. LED4 illuminates, indicating that the test is running. LED5 is the error signal.

5. Press stop (GX PB1) to stop the device from transmitting.

6. Press reset (GX_CLR).

7. Press start (Stratix_GX_PB_0).

8. Inject an error by pressing Stratix_GX_PB_2 (S4) once. The number 01 displays on the Stratix GX 7-segment display (D9) and Stratix GX LED5 illuminates.

9. Inject two more errors by pressing Stratix_GX_PB_2 (S4) twice. 03 displays on the Stratix GX 7-segment display (D9).
10. Press start and then reset three times to insure that the test is working.

11. If this test passes, write the word “PASS” in the column under Stratix GX in the row Stratix GX HSSDC2 XCVR in the worksheet.

**Nios-Based Tests**

In this section you will perform all of the Nios-based tests. For each test, you will load a SOF onto the board. These SOFs are located in the `Stratix_GX_kit\test_designs\Nios_test_designs\Nios_sofs` directory.

Use the same printed copy of Table 5–1 on page 5–36 that you used for “Perform the Standard Tests” on page 5–10 as your checklist worksheet for the tests.

You run all of the Nios-based tests from the SOPC Builder SDK Shell. You observe the test output in the shell. An RS-232 cable connected to your PC’s COM 1 port allows communication between the board and the software running on the PC. The RS-232 connector serves the Stratix GX device when the serial cable is attached to the bottom connector and the Stratix device when the serial cable is attached to the top connector as shown in Figure 4–2 on page 4–5.

**DDR Interface (Stratix GX Nios DDR)**

For this test, perform the following steps.

1. Attach the serial cable to the Stratix GX RS-232 connector (bottom connector).

2. Change programming file for the Stratix GX device to the `GX40_DDR.sof` file. Wait until the checksum field is updated.

3. Turn on the **Program/Configure** option for the Stratix GX device.

4. Click **Start**. When configuration is complete, the GX_CONF_DONE (D7) LED illuminates.

5. Perform the steps in “Run the Stratix GX DDR SDRAM Interface Test” on page 4–6 twice.

6. Press reset (Stratix_GX_PB_0) to stop the test and reset the processor in the device.

7. Press the Ctrl + C keys in the SOPC Builder SDK Shell to exit the test program.
8. If all of these tests pass, write the word “PASS” in the column under Stratix GX in the row Stratix GX NIOS DDR in the worksheet.

**PMC Card Interface (Stratix Nios PMC)**

This test requires a user-provided PCI card that is plugged into the PMC connector located on the back of the board. The test results depend on the card you installed. For this test, perform the following steps.


2. Change programming file for the Stratix device to the `pt_1s40.sof` file. Wait until the checksum field is updated.

3. Turn on the Program/Configure option for the Stratix device.

4. Click Start. When configuration is complete, the S_CONF_DONE LED illuminates.

5. Choose Programs > Altera > SOPC Builder <version> > SOPC Builder SDK Shell (Windows Start menu).

6. Change to the Nios SOFs directory by typing the following command:

   ```
   cd /Stratix_GX_kit/Test_designs/Nios_test_designs/Nios_sofs
   ```

7. Type `nr -t` to open a terminal window that connects to COM 1.

8. Press the Enter key twice in the terminal window. You should be at a Nios prompt.

9. Confirm that you are connected to the Nios processor running on the board by pressing the Enter key a few times in the terminal window to display the processor’s memory contents.

10. Exit the terminal window by pressing the Ctrl + C keys.

11. Start the test by typing `nr pt.srec`. This test enumerates the PCI bus and gives information on the PCI card connected to the PCI bus. Observe the results to see if they match your expectations for the PCI card you installed. Figure 5–20 shows an example of what you should see in the terminal window.
Figure 5–20. Example Stratix/Nios PCI Card Test Results

[SOPC Builder]$ nr PCI_test_rev1.srec

nios-run: Ready to download PCI_test_rev1.srec over COM1: at 115200 bps

nios-run: Downloading...........................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................

nios-run: Terminal mode (Control-C exits)

-----------------------------------------

NIOS test PCI bridge API ver 1.1
Testing config write to BAR0
Initial value BAR0 0x01000000
Final value BAR0 0x01000000
Testing config write to CMD
Initial value CMD 0x04000157
Final value CMD 0x04000157

Passed Self Config RW
Idsel 11, AISR value = 0x00000000, card id = 0x00041172, ID select = 0x00000800
Idsel 12 - No card detected
Idsel 13 - No card detected
Idsel 14 - No card detected
Idsel 15 - No card detected
Idsel 16, AISR value = 0x00000000, card id = 0x20001022, ID select = 0x00010000
Idsel 17 - No card detected
Idsel 18 - No card detected
Idsel 19 - No card detected
Idsel 20 - No card detected
Idsel 21 - No card detected
Idsel 22 - No card detected
Idsel 23 - No card detected
Idsel 24 - No card detected
Idsel 25 - No card detected
Idsel 26 - No card detected
Idsel 27 - No card detected
Idsel 28 - No card detected
Idsel 29 - No card detected
Idsel 30 - No card detected
Idsel 31 - No card detected

---------- PROBING CARD ON IDSEL 11 ----------

Vendor: Altera Corporation
Class and sub class:
Unknown
Device Does Not Fit in a Defined Class
UDF

---------- PROBING CARD ON IDSEL 16 ----------
Vendor: Advanced Micro Devices
Perform the Production Diagnostic Tests

Network Controller
Ethernet

**** Tests Complete ****

NOTE!!! Test repeats after a long delay.
Just make sure it recognizes TWO cards in the long list above.

12. Press reset (Stratix_PB_0) to stop the test.

13. Press the Ctrl + C keys in the shell to exit the test program.

14. If this test passes, write the word “PASS” in the column under Stratix in the row Stratix NIOS PMC in the worksheet.

Ethernet, On-Board Flash & EPC16 Flash Interface (Stratix Nios Ethernet) (Stratix Nios On-Board Flash) (Stratix Nios EPC16 Flash)

For this test, perform the following steps.

1. Verify that the serial cable is still attached to the Stratix RS-232 connector (top connector of RS-232).

2. Change programming file for the Stratix device to the ft_et_1s40.sof file. Wait until the checksum field is updated.

3. Turn on the Program/Configure option for the Stratix device.

4. Click Start. When configuration is complete, the S_CONF_DONE LED illuminates.

5. Choose Programs > Altera > SOPC Builder <version> > SOPC Builder SDK Shell (Windows Start menu).

6. Change to the Nios SOFs directory by typing the following command:

   ```
   cd /Stratix_GX_kit/Test_designs/Nios_test_designs/Nios_sofs
   ```

7. Type `nr -t` to open a terminal window that connects to COM 1.

8. Press the Enter key twice in the terminal window. You should be at a Nios prompt.
9. Confirm that you are connected to the Nios processor running on the board by pressing the Enter key a few times in the terminal window to display the processor's memory contents.

10. Exit the terminal window by pressing the Ctrl + C keys.

11. Start the Ethernet test by typing `nr et.srec`. This command initializes the on-board MAC/PHY chip. Observe the shell for the proper response. See Figure 5–21.

Figure 5–21. Ethernet Test Results

```
[NOTE: w/ethernet cable plugged in (below)]

[SOPC Builder]$ nr hello_plugs.srec
nios-run: Ready to download hello_plugs.srec over COM1: at 115200 bps
nios-run: Downloading...........................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
...  
nios-run: Terminal mode (Control-C exits)
--------------------------------------------------------
Reinitializing...
[lan91c111] nr_lan91c111_reset: chip id = SMC91C11xFD
[lan91c111] r_lan91c111_detect_phy: found lan83C183 (lan91C111 internal)
[lan91c111] r_lan91c111_init_phy: 10bt
[lan91c111] r_lan91c111_init_phy: half duplex
[accepting offer] from server 137.57.185.13, for ip address 137.57.185.196, lease time 446301
[ ignoring offer] from server 137.57.185.13, for ip address 137.57.185.196, lease time 446301
[lan91c111] nr_lan91c111_reset: chip id = SMC91C11xFD
[lan91c111] r_lan91c111_detect_phy: found lan83C183 (lan91C111 internal)
[lan91c111] r_lan91c111_init_phy: 10bt
[lan91c111] r_lan91c111_init_phy: half duplex
[plugs] +-----------------------
[plugs] | initialized adapter lan91c111 at 0x00c204a0, 137.57.185.196
[lan91c111] nr_lan91c111_reset: chip id = SMC91C11xFD
[lan91c111] r_lan91c111_detect_phy: found lan83C183 (lan91C111 internal)
[lan91c111] r_lan91c111_init_phy: 10bt
[lan91c111] r_lan91c111_init_phy: half duplex
```
Perform the Production Diagnostic Tests

Nios-Based Tests

[accepting offer] from server 137.57.185.13, for ip address 137.57.185.196, lease time 446296
[ ignoring offer] from server 137.57.185.13, for ip address 137.57.185.196, lease time 446296
[lan91c111] nr_lan91c111_reset: chip id = SMC91C11xFD
[lan91c111] r_lan91c111_detect_phy: found lan83C183 (lan91c111 internal)
[lan91c111] r_lan91c111_init_phy: 10bt
[lan91c111] r_lan91c111_init_phy: half duplex
[plugs] +-----------------------
[plugs] | initialized adapter lan91c111 at 0x00c204a0, 137.57.185.196

Main Menu

a: Network Settings...
b: Network Actions...
c: Exit to monitor

q: Main Menu

----->

#################################################
NOTE: w/o ethernet cable plugged in (below)

[SOPC Builder]$ nr hello_plugs.srec
nios-run: Ready to download hello_plugs.srec over COM1: at 115200 bps
nios-run: Downloading...........................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
......
12. Press reset (Stratix_PB_0) to stop the test.

13. Press Ctrl + C in the shell to exit the test program.

14. If this test passes, write the word “PASS” in the column under Stratix in the row Stratix Nios Ethernet in the worksheet.

15. Start the on-board flash memory test and EPC16 flash memory test by typing `nr ft.srec`. This command starts the writing and reading from the on-board flash memory device. Observe the shell for the proper response. See Figure 5–22.

---

Figure 5–22. Flash & EPC16 Memory Tests

```
[SOPC Builder]$ nr ft.srec

nios-run: Ready to download ft.srec over COM1: at 115200 bps

nios-run: Downloading...........................................................
................................................................................
................................................................................
....................

nios-run: Terminal mode (Control-C exits)
```

Erasing Flash Sector for EPC16 One
Erasing Flash Sector for EPC16 Two
Erasing Flash Sector for AMD
Writing 512 shorts to all Flash interfaces
Perform the Production Diagnostic Tests

Nios-Based Tests

Checking Written Data:
Test Done: Goodbye...

#77153008
lan16_all_flash4_pcb_revb
+

16. Press reset (Stratix_PB_0) to stop the test.

17. Press Ctrl + C in the shell to exit the test program.

18. If this test passes, write the word “PASS” in the column under Stratix in the row Stratix Nios On-Board Flash and Nios EPC16 Flash in the worksheet.

10/100 Ethernet Network-Interface Card I/O Interface (Stratix Nios PROTO1 IO)

For this test, perform the following steps.

1. Change the programming file for the Stratix device to the st_1s40.sof file. Wait until the checksum field is updated.

2. Turn on the Program/Configure option for the Stratix device.

3. Click Start. When configuration is complete, the S_CONF_DONE LED illuminates.


5. Change to the Nios SOFs directory by typing the following command:

    cd /Stratix_GX_kit/Test_designs/Nios_test_designs/Nios_sofs

6. Type nr -t to open a terminal window that connects to COM 1.

7. Press the Enter key twice in the terminal window. You should be at a Nios prompt.

8. Confirm that you are connected to the Nios processor running on the board by pressing the Enter key a few times in the terminal window to display the processor’s memory contents.
9. Exit the terminal window by pressing the Ctrl + C keys.

10. Start the test by typing `nr st.srec`.
    This command initializes the on-board MAC/PHY chip on the 10/100-Mbit Ethernet network-interface card. Observe the shell for the proper response. See Figure 5–23.

---

**Figure 5–23. 10/100 Ethernet Network-Interface Card Test Results**

```
NOTE: w/ethernet cable plugged in (below)

[SOPC Builder]$ nr hello_plugs.srec
nios-run: Ready to download hello_plugs.srec over COM1: at 115200 bps
nios-run: Downloading...........................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
[accepting offer] from server 137.57.185.13, for ip address 137.57.185.196, lease time 446301
[ ignoring offer] from server 137.57.185.13, for ip address 137.57.185.196, lease time 446301
[lan91c111] nr_lan91c111_reset: chip id = SMC91C11xFD
[lan91c111] r_lan91c111_detect_phy: found lan83C183 (lan91C111 internal)
[lan91c111] r_lan91c111_init_phy: 10bt
[lan91c111] r_lan91c111_init_phy: half duplex
[plugs] +-----------------------
[plugs] | initialized adapter lan91c111 at 0x00c204a0, 137.57.185.196
[plugs] | initialized adapter lan91c111 at 0x00c204a0, 137.57.185.196
[lan91c111] nr_lan91c111_reset: chip id = SMC91C11xFD
[lan91c111] r_lan91c111_detect_phy: found lan83C183 (lan91C111 internal)
[lan91c111] r_lan91c111_init_phy: 10bt
[lan91c111] r_lan91c111_init_phy: half duplex
```

---
Perform the Production Diagnostic Tests

Nios-Based Tests

[accepting offer] from server 137.57.185.13, for ip address 137.57.185.196, lease time 446296
[ ignoring offer] from server 137.57.185.13, for ip address 137.57.185.196, lease time 446296
[lan91c111] nr_lan91c111_reset: chip id = SMC91C11xFD
[lan91c111] r_lan91c111_detect_phy: found lan83C183 (lan91c111 internal)
[lan91c111] r_lan91c111_init_phy: 10bt
[lan91c111] r_lan91c111_init_phy: half duplex
[plugs] +-----------------------
[plugs] | initialized adapter lan91c111 at 0x00c204a0, 137.57.185.196

Main Menu

a: Network Settings...
b: Network Actions...
c: Exit to monitor

q: Main Menu

----->

###################################

NOTE: w/o ethernet cable plugged in (below)

[SOPC Builder]$ nr hello_plugs.srec
nios-run: Ready to download hello_plugs.srec over COM1: at 115200 bps
nios-run: Downloading...........................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
.............................................................................
nios-run: Terminal mode (Control-C exits)
    --------------------------
Reinitializing...
    [lan91c111] nr_lan91c111_reset: chip id = SMC91C11xFD
    [lan91c111] r_lan91c111_detect_phy: found lan83C183 (lan91C111 internal)
    [lan91c111] r_lan91c111_init_phy: phy negotiation timed out
    [lan91c111] r_lan91c111_init_phy: 10bt
    [lan91c111] r_lan91c111_init_phy: half duplex
    [dhcp] 1 timing out
    [dhcp] 2 timing out
    [dhcp] 3 timing out
    [dhcp] 4 timing out
    [dhcp] 5 timing out
    [dhcp] 6 timing out
    [dhcp] 7 timing out
    [dhcp] 8 timing out
    [dhcp] 9 timing out
    [dhcp] 10 timing out
    [dhcp] 11 timing out
    [dhcp] 12 timing out

11. Press reset (Stratix_PB_0) to stop the test.

12. Press Cntrl + C in the shell to exit the test program.

13. If this test passes, write the word “PASS” in the column under Stratix in the row Stratix Nios PROTO1 I/O in the worksheet.

**Compact Flash Interface (Stratix Nios Compact Flash)**

For this test, perform the following steps.

1. Verify that the serial cable is still attached to the Stratix RS-232 connector (top connector).

2. Make sure that the compact flash card is inserted. See Figure 5–24.
3. Change the programming file for the Stratix device to the `cf_1s40.sof` file. Wait until the checksum field is updated.

4. Turn on the **Program/Configure** option for the Stratix device.

5. Click **Start**. When configuration is complete, the S_CONF_DONE LED illuminates.

6. Choose **Programs > Altera > SOPC Builder <version> > SOPC Builder SDK Shell** (Windows Start menu).

   ![Note Icon](image)

   This shell may be named Nios SDK Shell in older versions of the Nios embedded processor.

7. Change to the Nios SOFs directory by typing the following command:

   ```
   cd /Stratix_GX_kit/Test_designs/Nios_test_designs/Nios_sofs
   ```

8. Type `nr -t` to open a terminal window that connects to COM 1.

9. Press the Enter key twice in the terminal window. You should be at a Nios prompt.
10. Confirm that you are connected to the Nios processor running on the board by pressing the Enter key a few times in the terminal window to display the processor's memory contents.

11. Exit the terminal window by pressing the Ctrl + C keys.

12. Type `nr idemon.srec` to start the Compact Flash test. Observe the results to see if any errors are reported. Figure 5–25 shows the text that should display in the shell.

---

**Figure 5–25. Compact Flash Test Results**

```
[SOPC Builder]$ nr idemon.srec
nios-run: Ready to download idemon.srec over COM1: at 115200 bps
niosrun:Downloading....................................................................
........................................................................................
........................................................................................
........................................................................................
........................................................................................
........................................................................................
........................................................................................
........................................................................................
........................................................................................
........................................................................................
........................................................................................
........................................................................................
........................................................................................
........................................................................................
........................................................................................
........................................................................................
........................................................................................
........................................................................................
........................................................................................
........................................................................................
nios-run: Terminal mode (Control-C exits)
-----------------------------------------
IDE Monitor v 1.1alpha
Starting Comprehensive CF Test!
Getting Compact Flash Params:
Param Returns:
    Cylinders: 496
    Heads: 4
    Sectors: 32
Starting Test:
Completed Testing of 20 Cylinders
Completed Testing of 40 Cylinders
Completed Testing of 60 Cylinders
Completed Testing of 80 Cylinders
Completed Testing of 100 Cylinders
Completed Testing of 120 Cylinders
Completed Testing of 140 Cylinders
Completed Testing of 160 Cylinders
Completed Testing of 180 Cylinders
Completed Testing of 200 Cylinders
Completed Testing of 220 Cylinders
```
Completed Testing of 240 Cylinders
Completed Testing of 260 Cylinders
Completed Testing of 280 Cylinders
Completed Testing of 300 Cylinders
Completed Testing of 320 Cylinders
Completed Testing of 340 Cylinders
Completed Testing of 360 Cylinders
Completed Testing of 380 Cylinders
Completed Testing of 400 Cylinders
Completed Testing of 420 Cylinders
Completed Testing of 440 Cylinders
Completed Testing of 460 Cylinders
Completed Testing of 480 Cylinders
Completed Testing of 495 Cylinders

Comparison Complete!!!
+

nios-run: exiting.

13. Press reset (Stratix_PB_0) to stop the test.

14. Press the Ctrl + C keys in the shell to exit the test program.

15. If this test passes, write the word “PASS” in the column under Stratix in the row Stratix Nios Compact Flash in the worksheet.

**Finishing Test & Breakdown**

You have completed all of the production tests. Perform the following steps to break down the test setup.

1. Close the Programmer.

2. Close the Quartus II software.

3. Turn off power to the board.

4. Detach all of the SMA cables, the PMC card, the 10/100-Mbit Ethernet network-interface card, the DDR DIMM module, the ByteBlaster II cable, all loopback cards, the compact flash card, the serial cable, and the power supply.
<table>
<thead>
<tr>
<th>Test</th>
<th>Tested on Stratix (PASS, FAIL, N/A)</th>
<th>Tested on Stratix GX (PASS, FAIL, N/A)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>User I/O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stratix GX-to-Stratix Bridge</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stratix GX HM-Zd (SPI 4.2)</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Stratix GX SMA DPA</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Stratix GX SMA XCVR</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Stratix GX HM-Zd XCVR</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Stratix GX SFP XCVR</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Stratix GX XPAK XCVR</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Stratix GX HSSDC2 XCVR (GX40 device only)</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Stratix GX Nios DDR</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Stratix Nios PMC</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Stratix Nios Ethernet</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Stratix Nios On-Board Flash</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Nios EPC16 Flash</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Stratix Nios PROTO1 I/O</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Stratix Nios Compact Flash</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>
6. Troubleshooting

Table 6–1 shows problems and solutions for power-up, configuration, and errors with test designs.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Possible Solutions</th>
</tr>
</thead>
</table>
| LEDs fail to illuminate when power is applied to the board   | Check that the main ATX supply connector is properly connected to J31.  
                                                               | Check that the ATX supply switch is turned on.  
                                                               | Check that the ATX supply is plugged into a power outlet. |
| Devices fail to configure                                    | Check that you chose the correct programming file.  
                                                               | Check that the ByteBlaster II cable is attached correctly to J87 (check that the pin one indicator on the cable is oriented correctly).  
                                                               | Check that the configuration switches SW1, SW2, and SW4 are set appropriately. |
| Design does not work properly                                | If you are using on-board oscillators, check that switches SW6 and SW9 are set to OSC (the up position).  
                                                               | If you are using external oscillators, check that switches SW6 and SW9 are set to EXT (the down position)  
                                                               | Check that the RS-232 cable is plugged into the correct connector on the board and on the PC.  
                                                               | Check that all of the cables are attached correctly.  
                                                               | Check that you chose the correct programming file.  
                                                               | Check that you are using the correct pushbutton switches and dipswitches.  
                                                               | Check that the dipswitches, if used, are in the correct positions.  
                                                               | Ensure that all of the daughter cards are fully seated (e.g., DDR, PMC, 10/100-Mbit Ethernet network-interface daughter board)  
                                                               | Check that the correct loopback cards are installed in the correct location and fully seated. |
| RS-232 connection is not working                             | Check that the RS-232 connector is plugged into the correct connector on the board and on the PC. |
7. Diagnostic Test Details

Standard Tests

This section describes the functionality of all of the standard tests. Refer to “Perform the Standard Tests” on page 5–10 for information on how to run the tests.

User I/O

This section describes the user I/O test. Refer to “User I/O” on page 5–10 for information on how to perform the test.

User I/O Test Overview

The Stratix and Stratix GX devices each have a test for the user I/O and LEDs. This I/O test design imitates a television remote control. The pushbutton switches operate as channel up, channel down, volume up, and volume down. The LEDs indicates the volume level, and the 7-segment display indicates the channel. The channel up and channel down buttons increment and decrement the number displayed on the 7-segment displays. The volume up and volume down buttons illuminate more or fewer LEDs. Dipswitches 7 through 1 allow you to display a specific channel number, represented in binary, on the 7-segment displays. Dipswitch 8 is an “enter” button that loads the binary value from the dipswitches onto the 7-segment displays as a decimal value.

When the 7-segment displays show 99 and you press the channel up button, the channel rolls over to 00. When displays show 00 and you press the channel down button, the channel rolls over to 99. If all six LEDs are illuminated and you press the volume up button, the volume remains at its highest level. If all six LEDs are off and you press the volume down button, the volume remains off.

The board does not actually make sounds. The LEDs simply represent what the volume would be.

The device clear buttons (GX_DEV_CLR for the Stratix GX device and S_DEV_CLR for the Stratix device) reset the design. When pressed, the LEDs turn off and the 7-segment displays show 00. The board contains an external debouncing chip; therefore, the pushbutton switches do not have to be debounced inside the Altera devices.
An enhanced PLL clocks these test designs. The PLL uses the 33-MHz signal from the on-board crystal oscillator as the reference clock. The PLL generates three internal clocks that are used in three different parts of the circuit:

- One clock is multiplied by one to generate a 33-MHz clock, which clocks all of the modules responsible for testing the pushbutton switches, LEDs, 7-segment displays, and dipswitches.
- One clock is multiplied by 6 to generate a 200-MHz clock, which clocks the data generators creating the data that is sent to the 20-pin headers and the Mictor connector.
- One clock is multiplied by 12 to generate a 400-MHz clock, which is divided by 2 to generate 200-MHz clocks as reference clocks for the Mictor connector.

*User I/O Test Functional Description*

Figure 7–1 shows the user I/O logic diagram for the Stratix GX and Stratix devices.
The design is divided into several modules that monitor the inputs, generate the control signals, and drive the LEDs and logic analyzer outputs:

- Edge detection
- Control logic
- LED decode logic
- Two shift registers
The edge detection module monitors the pushbutton switches and dipswitch 8. When it detects a rising edge from the switch, it generates a negative pulse signal for the control logic block.

The control block has two counters that maintain the channel count and the volume level. The channel counter is a 7-bit register that increments or decrements when it receives a negative pulse from the channel up and down edge detection circuits. The counter resets to 0 if you attempt to increment when the current value is 99. The counter does a parallel load when you enter the channel number using the dipswitches. The value is checked for being out of range and the value loaded is adjusted as needed. Because the counter uses a binary value, it must be processed to generate the two digits in the display. The counter value is divided by 10. The quotient drives the 10s digit and the remainder drives the 1s digit. The 4-bit values for the digits are decoded to drive the individual LED segments in the display.

The volume counter is similar. It is a 3-bit counter that increments and decrements based on pulses from the edge detection circuits connected to the volume up/down pushbutton switches. Because there are only six LEDs, the counter stops at six and stays there if you attempt to increment further. The count value is decoded by a 3-to-6 decoder to drive the LEDs.

**Mictor & 20-Pin Header Logic Analyzer Interfaces**

The Mictor and 20-pin headers are general-purpose I/O pins for use in debugging designs at speed. They are intended to drive logic analyzer input pods. The data rate is set to 200 MHz. The Mictor header is set up as two 16-bit channels with independent clocks. For this test, both channels are combined into one 32-bit word and both clocks are set to the same data rate. The 20-pin header design does not include a dedicated clock pin; the outputs are data only.

The designs for these interfaces are shift registers that drive the logic analyzer header pins. The values are shifted left on each clock cycle with the most significant bit (MSB) wrapping around to the least significant bit (LSB).

The Mictor and 20-pin headers output a walking 1s pattern. The starting pattern for the Mictor header is 32’h00010001 and the pattern looks like:

32’h00010001, 32’h00020002, 32’h00040004, 32’h00080008, 32’h00100010 etc.

Figure 7–2 shows an example of this pattern.
The starting pattern for the 20-pin header is 20’h00001. The 20-pin header walking 1s pattern (with the bus expanded) looks like:

20’h00001, 20’h00002, 20’h00004, 20’h00008, 20’h00010 etc.

Figure 7–3 shows an example of this pattern.
Stratix GX-to-Stratix Bridge

This section describes the Stratix GX-to-Stratix Bridge test. Refer to “Stratix GX-to-Stratix Bridge” on page 5–12 for information on how to perform the test.

Stratix GX-to-Stratix Bridge Test Overview

In this test, data is transmitted from the Stratix GX device to the Stratix device and back. The Stratix GX pushbutton switches control data transmission (start and stop) and reset the circuit. The Stratix GX LEDs indicate the start of transmission, the reception of the synchronizing signal, the confirmation that correct data was received, and the reset condition.

The design uses channel 0 to indicate transmission status, start, and stop. The other channels use this information to perform byte alignment and begin data validation. The accuracy required for proper functioning is +/- .5 of the unit interval (1.2 ns at the maximum data rate).
The data is generated and processed in the following sequence:

- The data is generated in 8-bit words per channel using a pseudo random byte sequence (PRBS) generator with a repetition count of 31.
- The data is then sent to a 16-channel version of the Altera LVDS megafunction, which uses a high-speed PLL and a serializer/deserializer (SERDES) block to convert the data into a serial data stream. The LVDS megafunction also generates the transmit clock.
- The data is sent to the Stratix device.
- The receiver LVDS megafunction converts the serial data back to parallel.
- The parallel data is fed back to a transmit block and sent back to the Stratix GX device as serial data.
- At the Stratix GX device, the data is converted to parallel.
- Based on the status channel, the received data is compared to the output of another PRBS generator. If both data streams match, the match LED illuminates. Each channel has its own PRBS generator and comparator. The match LED only illuminates when all channels match.

This design also includes an error detection and counting block. An error occurs when data does not match while valid data is transmitted. If an error occurs, the error LED illuminates and stays on. Every time an error is detected, the error counter increments and the value displays on the 7-segment display. Pressing the reset turns off the error LED and resets the counter.

The bridge test has two Quartus II projects:

- **Stratix GX device design**—This design has a PLL, an LVDS transmitter and receiver, and a Verilog HDL block with the logic required to generate a pseudo-random byte sequence (PRBS) and verify that it was received correctly.

- **Stratix device design**—This design has an LVDS receiver and transmitter as a loopback implementation.

Both devices must be configured with the appropriate design for the test to work.

The Stratix GX design has a Block Design File (.bdf) as the top-level design, which allows you to modify the system clock rate as desired to emulate a particular system configuration. By varying the system clock PLL and LVDS parameters, you can adjust the per channel data rate from
300 to 840 Mbps. The design has a Verilog HDL wrapper to name and place all of the pins and to provide proper termination for the LVDS signals.

The Stratix design is a BDF with the LVDS receiver and transmitter blocks implemented using the Altera MegaWizard® Plug-In Manager. When you vary the Stratix GX data rates, you must adjust the Stratix data rates accordingly.

The main system clock (parallel data rate) is derived from the 33-MHz crystal oscillator using a 63/20 ratio, resulting in 105-MHz clock rate.

*Stratix GX-to-Stratix Bridge Functional Description*

Figure 7–4 shows the logic diagram for the Stratix GX-to-Stratix bridge design. Figures 7–5 and 7–6 show the Quartus II top-level BDFs.

Open the BDFs in the Quartus II software to view greater detail.
Figure 7–4. Stratix GX-to-Stratix Bridge Logic Diagram
The system clock is generated by an enhanced PLL using the on-board 33.33-MHz crystal oscillator as the reference clock. The PLL generates a 105-MHz clock to clock all of the data generation logic and serve as the reference for the LVDS transmitter on the Stratix GX device.
Each transmit channel has its own PRBS generator that comprises eight 5-bit linear feedback shift registers. The output is taken from the MSB of each shift register. The initial seed value is 8’h47. When the enable (start) signal is high, the generator outputs a 31-word sequence that repeats until stopped. On reset, the seed value is initialized into all of the registers. This generator creates the data stream that exercises the system.

The edge generator uses channel 0 to transmit a signal to the receive channel that data is being transmitted. When stopped, the output is 8’h00. When running, the output is 8’hFF. This signal is synchronized with the start of the PRBS generator.

The data from the PRBS generators and the edge generator is combined and sent to an LVDS transmit megafunction created using the Altera MegaWizard Plug-In Manager. The megafunction is configured with 16 channels running at 840 Mbps and a clock rate of 105 MHz. The signals are sent to the Stratix device as 16 serial data channels at 840 Mbps with a clock signal of 105 MHz.

The Stratix device receives the signals and converts them back into parallel data using an LVDS receive megafunction (128 bits at 105 MHz). The parallel data is sent to an LVDS transmitter and sent back to the Stratix GX device as serial data. An LVDS receive megafunction on the Stratix GX device converts the serial data back to parallel and generates the clock used to regulate the receive channel logic.

The edge detector monitors channel 0 for 1s. Depending on when the 1s appear in the 8-bit data word, it generates a shift value to realign the data to the proper byte boundary. It also generates a data valid signal to start the expected value PRBS generator.

The data shift block uses the shift value from the edge detector to shift the incoming data stream to the proper byte alignment. The parallel-to-serial-to-parallel conversion process used in the LVDS transmission can lose the byte alignment, and this block restores it.

A second receive channel PRBS generator generates the expected values to compare with the incoming data stream. This generator is identical to the one used in the transmit channels. The start signal for this generator comes from the edge detection module.

The compare module takes the output from the data shift block and compares it with the output from the receive channel PRBS. The 8-bit words are compared each clock cycle. The comparator output is high if the words match. The match output from all 15 receive channels is ANDed together and then stored in a single-bit match register. The output of this register drives the match LED.
The error detection block monitors the output of the edge detection module and the match register. If match goes low while data is valid, the error register is set. The error counter also increments every clock cycle that an error is detected. The count value is converted to decimal values for display. The counter rolls over at 99 to 0 and then increments. Pressing the reset signal clears the error register and resets the counter to 0.

**Stratix GX HM-Zd (SPI-4.2)**

This section describes the Stratix GX HM-Zd (SPI-4.2) test. Refer to “Source Synchronous HM-Zd Interface (Stratix GX HM-Zd SPI 4.2)” on page 5–14 for information on how to perform the test.

**Stratix GX HM-Zd (SPI-4.2) Test Overview**

This design has a PLL, an LVDS transmitter and receiver, and a Verilog HDL block with the logic required to generate a PRBS and verify that it was received correctly. The design uses two HM-Zd loopback cards to complete the circuit. The pinout is compatible with the SPI-4.2 standard, including the high-speed control signal and the low-speed status signals.

You can use the top-level BDF to modify the system clock rate as desired to emulate a particular system configuration. By varying the system clock PLL and LVDS megafuction parameters, you can adjust the per channel data rate from 300 to 1,000 Mbps. This design uses the DPA feature of the Stratix GX family to boost the data rate to 1,000 Mbps.

The design has a Verilog HDL wrapper to name and place all of the pins and to provide proper termination for the LVDS signals. The Stratix GX pushbutton switches control the data transmission (start and stop), insert errors, and reset the circuit. The LEDs indicate the start of transmission, the start of reception of valid data, confirmation that correct data was received, error status, and the reset condition.

The main system clock (parallel data rate) is derived from the 33-MHz crystal oscillator using a 15/4 ratio resulting in 125-MHz clock rate. The data is generated in 8-bit words per channel using a PRBS generator with a repetition count of 31, resulting in a serial data rate of 1 Gbps. The data is then sent to a 17-channel version of the Altera LVDS megafuction, which uses a high-speed PLL and a SERDES block to convert the data into serial data streams. The LVDS megafuction also generates the transmit clock, which is 125 MHz. You can vary the clock speed by changing the parameters of the LVDS megafuction.
The loopback card feeds the serial data back to the receive inputs on the Stratix GX device. The data is converted back into parallel form by the receive LVDS megafunction. Because the design uses the Stratix GX DPA feature, the receive data byte alignment may not be correct. Therefore, the data is sent through a byte alignment and synchronization detection block. This block looks for the synchronization pattern (the first word of the PRBS sequence) in the data stream. When it finds the pattern, the data is shifted as needed and the data valid is asserted. This assertion triggers the start of an expected value PRBS generator. The two data streams are sent to a comparator to generate a match signal on a per channel basis. If all channels match, the match LED illuminates.

**Stratix GX HM-Zd (SPI-4.2) Functional Description**

Figure 7–7 shows the Stratix GX HM-Zd (SPI-4.2) logic diagram for the Stratix GX device. Figure 7–8 shows the Quartus II top-level BDF.

Open the BDF in the Quartus II software to view greater detail.
The system clock is generated by an enhanced PLL using the on-board 33.33 MHz crystal as the reference. The PLL generates a 125-MHz clock to run all of the data generation logic and serve as the reference for the LVDS transmitter on the Stratix GX device.

The transmit PRBS generator is constructed with eight 5-bit linear feedback shift registers. The output is taken from the MSB of each shift register. The initial seed value is 8'h47. When the enable (start) signal is high the generator outputs a 31-word sequence that repeats until stopped. On reset the seed value is initialized into all registers. This generator is used to generate the data stream used to exercise the system. Each transmit channel has its own PRBS generator.

DPA technology requires a training pattern to be sent prior to the start of data transmission. In this design, an 8-bit counter and a fixed pattern generator create the training pattern. When you press the start button, the counter starts and the fixed pattern transmits. When the counter reaches 256, a start signal is sent to the PRBS generator and the output multiplexer switches over to the PRBS data.

The data from the PRBS generators is sent to the LVDS transmit block created using the Altera MegaWizard Plug-In Manager. The megafunction is configured as 17 channels running at 1,000 Mbps with a clock rate of 125 MHz. The signals are sent to the HM-Zd connector and looped back to the Stratix GX device. The high-speed control channel is treated as any other data channel.
The slow speed status channels are controlled in a separate logic block that is clocked by the system clock at 125 MHz. The data transmitted is the output from a 4-bit counter. The data is captured in registers and compared to the expected value. A separate control match LED illuminates when the data matches.

An LVDS receive megafunction on the Stratix GX device converts the serial data back to parallel and generates the clock used to regulate the receive channel logic. The received data is sent through a pattern detector/data aligner block. When the pattern detector detects the synchronization pattern twice, it sets the data valid signal and starts passing the data to the comparator.

A second PRBS generator uses the data valid signal to start generating the expected data values. This second data set is also sent to the comparator. The comparator module takes the output from the data aligner block and compares it with the output from the receive channel PRBS. The 8-bit words are compared each clock cycle. The comparator output is high if the words match. The output from all 17 receive channels is ANDed together and then stored in a single-bit match register that drives the match LED.

The error detection and counting blocks monitor the match and data valid signals. If the match signal goes low while data is valid, the error flag is set and the error counter is incremented. Pressing the reset button clears the error flag and resets the counter. The error insertion pushbutton switch inverts one bit in one data channel for one clock cycle, which is enough to trigger the error detection circuit.

**Stratix GX SMA DPA**

This section describes the Stratix GX SMA DPA test. Refer to “Source Synchronous DPA SMA interface (Stratix GX SMA DPA)” on page 5–15 for information on how to perform the test.

**Stratix GX SMA DPA Test Overview**

The DPA SMA design consists of a PLL, a 2-channel LVDS transmitter and receiver, and a Verilog HDL block with the logic required to generate a PRBS and verify that it was received correctly. The design requires 6 SMA cables to complete the signal loopback.

You can use the top-level BDF to modify the system clock rate as desired to emulate a particular system configuration. By varying the system clock PLL and LVDS megafunction parameters, you can adjust the per channel data rate from 300 to 1,000 Mbps. This design uses the DPA feature of the Stratix GX family to boost the data rate to 1,000 Mbps.
The design has a Verilog HDL wrapper to name and place all of the pins and to provide proper termination for the LVDS signals. The Stratix GX pushbutton switches control the data transmission (start and stop), insert errors, and reset the circuit. The LEDs indicate the start of transmission, the start of reception of valid data, confirmation that correct data was received, error status, and the reset condition.

The main system clock (parallel data rate) is derived from the 33-MHz crystal oscillator using a 15/4 ratio resulting in 125-MHz clock rate. The data is generated in 8-bit words per channel using a PRBS generator with a repetition count of 31, resulting in a serial data rate of 1 Gbps. The data is then sent to a 2-channel version of the Altera LVDS megafunction, which uses a high-speed PLL and a SERDES block to convert the data into serial data streams. The LVDS megafunction also generates the transmit clock, which is 125 MHz. You can vary the clock speed by changing the parameters of the LVDS megafunction.

The loopback cables feed the serial data back to the receive inputs on the Stratix GX device. The data is converted back into parallel by the receive LVDS megafunction. Because the design used the Stratix GX DPA feature, the received data byte alignment can be incorrect. Therefore, the data is sent through a byte alignment and synchronization detection block. The block looks for the synchronization pattern (the first word of the PRBS sequence) in the data stream. When it finds this pattern, it shifts the data as needed and asserts the data valid signal. This assertion triggers the start of an expected value PRBS generator. The two data streams are sent to a comparator to generate a match signal on a per channel basis. If both channels match, the match LED illuminates.

**Stratix GX SMA DPA Functional Description**

Figure 7–9 shows the Stratix GX DPA logic diagram for the Stratix GX device. Figure 7–10 shows the Quartus II top-level BDF.

Open the BDF in the Quartus II software to view greater detail.
Figure 7–9. Stratix GX SMA DPA Logic Diagram

Transmit Channel (x2)

Start/Stop

8-Bit Counter

PRBS Generator

Pattern Generator

ALTLVDS TX

Loopback with 6 SMA cables

Receive Channel (x2)

Data Valid

Match Register

Comparator

Data Aligner

PRBS Generator

ALTLVDS RX W/ DPA

Error LED

Error Register

7-Segment Display

Match LED

Data Valid

Start/Stop

8-Bit Counter

Pattern Generator

ALTLVDS TX

Loopback with 6 SMA cables

Match LED

Match Register

Comparator

Data Aligner

PRBS Generator

ALTLVDS RX W/ DPA

Error LED

Error Register

7-Segment Display
The system clock is generated by an enhanced PLL using the on-board 33.33-MHz crystal oscillator as the reference clock. The PLL generates a 125-MHz clock that clocks the data generation logic and serves as the reference for the LVDS transmitter on the Stratix GX device.

The transmit PRBS generator is constructed with eight 5-bit linear feedback shift registers. The output is taken from the MSB of each shift register. The initial seed value is 8'h47. When the enable (start) signal is high, the generator outputs a 31-word sequence that repeats until stopped. On reset, the seed value is initialized into all of the registers. This generator generates the data stream that exercises the system. Each transmit channel has its own PRBS generator.

The DPA technology requires a training pattern to be sent prior to the start of data transmission. An 8-bit counter and a fixed pattern generator create the training pattern. When you press the start button, the counter starts and the fixed pattern is transmitted. When the counter reaches 256, a start signal is sent to the PRBS generator and the output multiplexer switches over to the PRBS data.
The PRBS generator data goes to the LVDS transmit block, which was created using the Altera MegaWizard Plug-In Manager. The megafuntion is configured as 2 channels running at 1,000 Mbps with a clock rate of 125 MHz. The signals then go to the SMA connectors and loop back to the Stratix GX device using 6 SMA cables. The total signal path lengths between channels 0 and 1 is offset by 4.5 inches.

An LVDS receive megafunction on the Stratix GX device converts the serial data back to parallel and generates the clock used to regulate the receive channel logic. The received data is sent through a pattern detector/data aligner block. When the pattern detector detects the synchronization pattern twice, it sets the data valid signal and starts passing the data to the comparator.

A second PRBS generator uses the data valid signal to start generating the expected data values. This second data set is also sent to the comparator. The compare module takes the output from the data aligner block and compares it with the output from the receive channel PRBS. The 8-bit words are compared each clock cycle. The comparator output is high if the words match. The output from both receive channels is ANDed together and stored in a single-bit match register. The output of this register drives the match LED.

The error detection and counting blocks monitor the match and data valid signals. If the match signal goes low while data is valid, the error flag is set and the error counter increments. The reset button clears the error flag and resets the counter. The error insertion pushbutton inverts one bit in one data channel for one clock cycle, which is enough to trigger the error detection circuit.

**Stratix GX SMA XCVR**

This section describes the Stratix GX SMA XCVR test. Refer to “Gigabit Transceivers with SMA Interface (Stratix GX SMA XCVR)” on page 5–17 for information on how to perform the test.

**Stratix GX SMA XCVR Test Overview**

The transceiver SMA design includes all of the Gigabit XCVR interface designs for the Stratix GX device. The SMA interface portion of the design comprises a 4-channel Altera Gigabit transceiver block (GXB) and a Verilog HDL block with the logic required to generate a PRBS and verify that it was received correctly. This design requires 8 SMA cables to complete the signal loopback.

For more information on the operation and parameterization of the GXB megafunction, refer to Quartus II Help.
The top-level BDF lets you easily modify the system parameters as desired to evaluate a particular system configuration. By varying the GXB megafunction parameters, you can adjust the per channel data rate up to 3,125 Mbps. The design has a Verilog HDL wrapper to name and place all of the pins and to provide proper termination for the signals. The Stratix GX pushbutton switches control the start and stop of the data transmission, insert errors, and reset the circuit. The LEDs indicate the start of transmission, confirm that data was received correctly, error status, and the reset condition.

The main system clock (parallel data rate) is derived from the 156.25-MHz crystal oscillator using the GXB megafunction clock output. The data is generated in 20-bit words per channel using a PRBS generator with a repetition count of 31, resulting in a serial data rate of 3.125 Gbps. The data is then sent to a 4-channel version of the GXB megafunction, which converts the data into gigabit serial data streams. The megafunction instance uses the basic protocol option.

The SMA cables feed the serial data back to the receive inputs on the Stratix GX device. The GXB megafunction converts the data back into parallel. Because the implementation uses the GXB megafunction’s double word feature, the byte alignment of the received data may be inaccurate. To adjust it, the data is sent through a byte swap block based on synchronization data from the GXB megafunction. The data is then passed to a pattern detection block to find the start of data in the PRBS (the first word of the PRBS sequence) data stream. When this pattern is found, the data valid signal is asserted. This assertion triggers an expected value PRBS generator to start. The two data streams are sent to a comparator to generate a match signal on a per channel basis. If the data streams match, the match LED illuminates on a per channel basis. If the match signal goes low while the data valid signal is high, the error flag is set and the error counter is incremented. Pressing the reset button resets the system state, error flag, and error count.

.Stratix GX SMA XCVR Functional Description

Figure 7–11 shows the Stratix GX SMA XCVR logic diagram. Figure 7–12 shows the Quartus II top-level BDF for all of the XCVR tests.

Open the BDF in the Quartus II software to view greater detail.
Figure 7–11. Stratix GX SMA XCVR Logic Diagram

Figure 7–12. Quartus II Stratix GX Top-Level BDF of XCVR Tests
The GXB megafunction transmit PLL generates the system clock using the 156.25-MHz crystal as the reference. The PLL generates a 156.25-MHz clock to clock all of the data generation logic.

The transmit PRBS generator comprises 20 5-bit linear feedback shift registers. The output is taken from the MSB of each shift register. The initial seed value is 20'h695A7. When the enable (start) signal is high, the generator outputs a 31-word sequence that repeats until stopped. On reset, the seed value is initialized into all registers. This generator generates the data stream that exercises the system. Each transmit channel has its own PRBS generator.

The GXB megafunction double word option requires synchronizing the control logic with the GXB receive section. A 10-bit alignment pattern (10'h1A7) allows for the internal synchronization of the megafunction and a status signal that is sent to the word swap block in the control logic. Data transmission is only allowed to start when a channel has been synchronized. This synchronization happens on a per channel basis.

The data from the PRBS generators is sent to the GXB transmit block created using the Altera MegaWizard Plug-In Manager. The megafunction is configured as 4 channels running at 3,125 Mbps with an input clock rate of 156.25 MHz. The signals are then sent to the vertical and edge launch SMA connectors and looped back to the Stratix GX device using 8 SMA cables.

The receive portion of the GXB megafunction converts the serial data back to parallel 20-bit words. The received data is sent through a byte/word swap block that is controlled by the GXB megafunction, which is required for double-word operation because the alignment pattern is only 10 bits. The data is then passed to the pattern detector to determine the start of the data packet. When the pattern detector has detected the synchronization pattern twice, it sets the data valid signal and starts passing the data to the comparator.

A second PRBS generator uses the data valid signal to start generating the expected data values. This second data set is also sent to the comparator. The comparator module takes the output from the pattern detector block and compares it to the output from the receive channel PRBS. The 20-bit words are compared each clock cycle. The comparator output is high if the words match. The output from each receive channel comparator is stored in a single-bit match register. The output of this register drives the match LED on a per channel basis.

The error detection and counting blocks monitor the match and data valid signals. If the match signal goes low while data is valid, the error flag is set and the error counter is incremented. The reset button clears the
error flag and resets the counter. The error insertion pushbutton switch inverts one bit in one data channel for one clock cycle, which is enough to trigger the error detection circuit.

**Stratix GX HM-Zd XCVR**

This section describes the Stratix GX HM-Zd XCVR test. Refer to “Gigabit Transceivers with HM-Zd Interface (Stratix GX HM-Zd XCVR)” on page 5–18 for information on how to perform the test.

**Stratix GX HM-Zd XCVR Test Overview**

The transceiver HM-Zd design has a PLL, a 4 channel Altera GXB transmitter/receiver block and a Verilog HDL block with the logic required to generate a PRBS and verify that it was received correctly. The design requires the transceiver HM-Zd loopback card to complete the signal loopback.

For more information on the operation and parameterization of the GXB megafuncion, refer to Quartus II Help or the *Stratix GX FPGA Family Data Sheet*.

The top-level BDF lets you easily modify the system clock rate as desired to evaluate a particular system configuration. By varying the GXB megafunction parameters, you can adjust the per channel data rate up to 3,125 Mbps. The design has a Verilog HDL wrapper to name and place all of the pins and to provide proper termination for the signals. The Stratix GX pushbutton switches control the start and stop of the data transmission and reset the circuit. The LEDs indicate the start of transmission, the start of reception of the valid data, the confirmation that data was received correctly, and the reset condition.

The main system clock (parallel data rate) is derived from the 156.25-MHz crystal oscillator using the GXB megafunction clock output. The data is generated in 20-bit words per channel using a PRBS generator with a repetition count of 31, which results in a serial data rate of 3.125 Gbps. The data is then sent to a 4-channel version of the Altera GXB megafunction, which converts the data into gigabit serial data streams. The megafunction uses the basic protocol option.

The loopback card feeds the serial data back to the receive inputs on the Stratix GX device. The data is converted back into parallel form by the GXB megafunction. Because the design uses the double word feature of the GXB megafunction, the byte alignment of the received data may be incorrect. To adjust it, the data is sent through a byte swap block based on synchronization data from the GXB megafunction. The data is then passed to a pattern detection block to find the start of data in the PRBS.
(the first word of the PRBS sequence) data stream. When this pattern is found, the data valid signal is asserted, which triggers an expected value PRBS generator to start. The two data streams are sent to a comparator to generate a match signal on a per channel basis. If the data streams match, the match LED illuminates on a per channel basis. If the match signal goes low while the data valid signal is high, the error flag is set and the error counter is incremented. Pressing the reset button resets the system state, error flag, and error count.

Stratix GX HM-Zd XCVR Functional Description

Figure 7–13 shows the Stratix GX HM-Zd XCVR logic diagram.

The GXB megafunction transmit PLL generates the system clock using the 156.25-MHz crystal as the reference. The PLL generates a 156.25-MHz clock to clock all of the data generation logic. The transmit PRBS generator comprises 20 5-bit linear feedback shift registers. The output is taken from the MSB of each shift register. The initial seed value is 20’h695A7. When the enable (start) signal is high, the generator outputs a 31-word sequence that repeats until stopped. On
reset, the seed value is initialized into all registers. This generator generates the data stream that exercises the system. Each transmit channel has its own PRBS generator.

The double word option requires synchronization of the control logic with the GXB receive section. The design uses a 10-bit alignment pattern (10’h1A7) that permits the internal synchronization of the megafonction and creates a status signal that is sent to the word swap block in the control logic. Data transmission can only start when a channel has been synchronized. Synchronization is done on a per channel basis.

The data from the PRBS generators is sent to the GXB transmit block created using the Altera MegaWizard Plug-In Manager. The megafonction is configured as 4 channels running at 3,125 Mbps with an input clock rate of 156.25 MHz. The signals are then sent to the HM-Zd XCVR connector (J1) and looped back to the Stratix GX device using the XCVR HM-Zd loopback card.

The design was also tested using the Tyco backplane on the 1- and 16-inch connections.

When using the long trace connection on the backplane, the receive equalization must be set for 40 inches for reliable data throughput. Dipswitches 2 and 1 set the equalization (00 for 0 inches, 01 for 20 inches, and 11 for 40 inches). The short trace and loopback card work without equalization.

The receive portion of the GXB megafonction on the Stratix GX device converts the serial data back to parallel 20-bit words. The received data is sent through a byte/word swap block that is controlled by the GXB megafonction, which is required for double word operation because the alignment pattern is only 10 bits. The data is then passed to the pattern detector to determine the start of the data packet. When the pattern detector has detected the synchronization pattern twice, it sets the data valid signal and starts passing the data to the comparator.

A second PRBS generator uses the data valid signal to start generating the expected data values. This second data set is also sent to the comparator. The comparator module takes the output from the pattern detector block and compares it with the output from the receive channel PRBS. The 20-bit words are compared each clock cycle. The comparator output is high if the words match. The output from each receive channel comparator is stored in a single-bit match register. The output of this register drives the match LED on a per channel basis.
The error detection and counting blocks monitor the match and data valid signals. If the match signal goes low while data is valid, the error flag is set and the error counter is incremented. The reset pushbutton clears the error flag and resets the counter. The error insertion pushbutton inverts one bit in one data channel for one clock cycle, which is enough to trigger the error detection circuit.

**Stratix GX SFP XCVR**

This section describes the Stratix GX SFP XCVR test. Refer to “Gigabit Transceivers with SFP Interface (Stratix GX SFP XCVR)” on page 5–19 for information on how to perform the test.

**Stratix GX SFP XCVR Test Overview**

This design consists of a PLL, a 4 channel Altera GXB transmitter/receiver block and a Verilog HDL block with the logic required to generate a PRBS and verify that it was received correctly. The design requires an SFP module qualified to 2.488 Gbps with an optical fiber loopback or a SFP loopback card to complete the signal loopback.

For more information on the operation and parameterization of the GXB megafunction, refer to Quartus II Help.

The top-level BDF lets you easily modify the system clock rate as desired to evaluate a particular system configuration. The GXB megafunction parameters are set to run at 2.488 Gbps per channel data rate compatible with OC-48. The design has a Verilog HDL wrapper to name and place all of the pins and to provide proper termination for the signals. The Stratix GX pushbutton switches control the start and stop of the data transmission and reset the circuit. The LEDs indicate the start of transmission, the start of reception of the valid data, confirm that data was received correctly, and indicate the reset condition.

The main system clock (parallel data rate) is derived from the 155.52-MHz crystal oscillator using the GXB megafunction clock output. The data is generated in 16-bit words per channel using a PRBS generator with a repetition count of 31, which results in a serial data rate of 2.488 Gbps. The data is then sent to a 4 channel version of the Altera GXB megafunction, which converts the data into gigabit serial data streams. The megafunction uses the basic protocol option.

The loopback card feeds the serial data back to the receive inputs on the Stratix GX device. The data is converted back into parallel by the GXB megafunction. Because the design uses the GXB megafunction’s double word feature, the byte alignment of the received data can be incorrect. To adjust it, the data is sent through a byte swap block based on
synchronization data from the GXB megafunction. The data is then passed to a pattern detection block to find the start of data in the PRBS (the first word of the PRBS sequence) data stream. When this pattern is found, the data valid signal is asserted, which triggers an expected value PRBS generator to start. The two data streams are sent to a comparator to generate a match signal on a per channel basis. If the data streams match, the match LED illuminates on a per channel basis. If the match signal goes low while the data valid signal is high, the error flag is set and the error counter is incremented. Pressing the reset button resets the system state, error flag, and error count.

**Stratix GX SFP XCVR Functional Description**

Figure 7–14 shows the Stratix GX SFP XCVR logic diagram.

The GXB megafunction transmit PLL generates the system clock using the 155.52-MHz crystal oscillator as the reference. The PLL generates a 155.52-MHz clock to clock all of the data generation logic.
The transmit PRBS generator is constructed with 16 5-bit linear feedback shift registers. The output is taken from the MSB of each shift register. The initial seed value is 16’h1A47. When the enable (start) signal is high, the generator outputs a 31-word sequence that repeats until stopped. On reset, the seed value is initialized into all registers. This generator generates the data stream that exercises the system. Each transmit channel has its own PRBS generator.

The double word option requires synchronization of the control logic with the GXB receive section. The design uses a 16-bit alignment pattern, 16’h1A47, for synchronization. The design also uses a custom protocol with manual word alignment. This protocol requires the use of a byte swap block to align the upper and lower bytes of the 16-bit word. Status signals from the GXB block control the byte swap circuit on a per channel basis.

The data from the PRBS generators is sent to the GXB transmit block created using the Altera MegaWizard Plug-In Manager. The megafunction is configured as 4 channels running at 2.488 Gbps with an input clock rate of 155.52 MHz. The signals are then sent to the SFP module connectors (J54, J64, J45, and J38). You can use either an OC-48 SFP optical module with loopback fiber or the SFP loopback card to provide the loopback function required to return the data to the Stratix GX device.

The SFP module has 3 status signals: TX_Fault, TX_Disable, and RX_LOS. The TX_Fault and RX_LOS signals are not monitored to allow the use of the SFP loopback card. TX_Disable is controlled by the stop pushbutton. The MOD-DEF[2:0] signals are not used or monitored in this design. The Altera SFP loopback card has DC blocking capacitors in the high-speed data path to preserve the DC levels on the transmit and receive buffers.

The receive portion of the GXB megafunction converts the serial data back to parallel 16-bit words. The received data is sent through a byte/word swap block that is controlled by the GXB megafunction. The data is then passed to the pattern detector to determine the start of the data packet. When the pattern detector has detected the synchronization pattern twice, it sets the data valid signal and starts passing the data to the comparator.

A second PRBS generator uses the data valid signal to start generating the expected data values. This second data set is also sent to the comparator. The comparator module takes the output from the pattern detector block and compares it with the output from the receive channel PRBS. The 16-bit words are compared each clock cycle. The comparator output is
high if the words match. The output from each receive channel comparator is stored in a single bit match register. The output of this register drives the match LED on a per channel basis.

The error detection and counting blocks monitor the match and data valid signals. If the match signal goes low while data is valid, the error flag is set and the error counter is incremented. The reset pushbutton clears the error flag and resets the counter. The error insertion pushbutton inverts one bit in each data channel for one clock cycle, which is enough to trigger the error detection circuit even if only one channel is active.

**Stratix GX XPAK XCVR**

This section describes the Stratix GX XPAK XCVR test. Refer to “Gigabit Transceivers with XPAK Interface (Stratix GX XPAK XCVR)” on page 5–20 for information on how to perform the test.

This test requires an XPAK module, which is not included with the kit.

**Stratix GX XPAK XCVR Test Overview**

The XPAK interface design consists of a 4-channel Altera GXB transmitter/receiver block configured in XAUI mode and a Verilog HDL block with the logic required to generate a simple XGMII packet stream and verify that it was received correctly. The design requires an XPAK module with optical loopback cable to complete the signal loopback.

The top-level BDF lets you easily modify the system clock rate to evaluate a particular system configuration. The design has a Verilog HDL wrapper to name and place all the pins and provide proper termination for the signals. The Stratix GX pushbutton switches control the start and stop of the data transmission and reset the circuit. The LEDs indicate the start of transmission, the start of reception of the valid data, confirm that data was received correctly, error status, and indicate the reset condition.

The main system clock (parallel data rate) is derived from the 156.25-MHz crystal oscillator using the GXB megafuction clock output. The data is generated as a 64-bit word using a simple packet generator. The generator creates idle packets and XGMII style packets with a preamble, data payload, and termination octets. The generator also inserts an inter packet gap (IPG) between packets, resulting in a serial data rate of 10 Gbps. The data is then sent to a 4-channel version of the Altera GXB megafuction configured for the XAUI protocol, which converts the data into four 3.125-Gbps serial data streams. These streams are sent to the XPAK module.
The module converts the data to a single 10-Gbps optical data stream. The loopback cable feeds the serial data back to the receive input of the XPAK device. The XPAK module converts the data back to XAUI format. The data is then passed to a pattern detection block to find the start of a data packet in the data stream. When this pattern is found, the data valid signal is asserted, which triggers an expected value packet generator to start. The two data streams are sent to a comparator to generate a match signal. If the data streams match, the match LED illuminates. Only the data packets are checked for errors. If the match signal goes low while the data valid signal is high, the error flag is set and the error counter is incremented. Pressing the reset button resets the system state, error flag, and error count.

**Stratix GX XPAK XCVR Functional Description**

Figure 7–15 shows the XPAK XCVR logic diagram.

The system clock is generated by the GXB megafuction’s transmit PLL using the 156.25-MHz crystal oscillator as the reference. The PLL generates a 156.25-MHz clock to clock all of the data generation logic.
The transmit packet generator is a Verilog HDL module. When enabled, the module generates simple XGMII style packets, otherwise, it outputs idle characters. An XGMII packet consists of a header, data payload and a termination character sequence. The data is created as a series of octets of bytes (64 bits) and an 8-bit wide control word. Each bit in the control word corresponds to one of the bytes in the octet. The data payload is the output of an 8-bit counter. The packet generator also inserts an interpacket gap between packets. The packet size and interpacket gap size is hard-coded in the Verilog HDL code.

The design has a control block that monitors the pushbuttons and status to control the test operation. Because this test is using the XAUI and XGMII protocols, the GXB megafunction channels must be synchronized and aligned to each other. The control block monitors this synchronization and alignment, and illuminates LED1 when the channels are synchronized and LED2 when they are aligned. The control block also prevents the start of the test until the channels are aligned. When you press PB0 to start the test, the control logic checks the status of the channel and triggers the packet generator to start. It also illuminates LED4.

The data from the packet generator is sent to the GXB transmit block created using the Altera MegaWizard Plug-In Manager. The megafunction is configured with the XAUI protocol using four channels running at 3,125 Mbps with an input clock rate 156.25 MHz. The GXB megafunction takes the data stream along with the control character information and converts the data from XGMII protocol to XAUI protocol, including 8B/10B encoding of the data stream. The signals are then sent to the XPAK module and looped back to the Stratix GX device using the optical loopback fiber.

The receive portion of the GXB megafunction on the Stratix GX device converts the serial data back to parallel 64-bit words. The data is then passed to the pattern detector to determine the start of the data packet. When the pattern detector detects the synchronization pattern twice, it sets the data valid signal and begins passing the data to the comparator. The data valid signal also drives LED3.

A second packet generator uses the data valid signal to start generating the expected data values. This second data set is sent to the comparator. The comparator module takes the output from the pattern detector block and compares it with the output from the receive channel packet generator. The 64-bit words are compared each clock cycle during the data payload section of the data stream. The comparator output is high if the words match. The output from the comparator is stored in a single-bit match register. The output of this register drives the match LED.
The error detection and counting blocks monitor the match and data valid signals. If the match signal goes low while data is valid, the error flag is set and the error counter is incremented. Pressing the reset button clears the error flag and resets the counter. The error insertion pushbutton inverts bit 0 in a data packet word for one clock cycle, which is enough to trigger the error detection circuit.

**Stratix GX HSSDC2 XCVR**

This section describes the Stratix GX HSSDC2 XCVR test. Refer to “Gigabit Transceivers with HSSDC2 Interface (Stratix GX HSSDC2 XCVR) (GX40 Device Only)” on page 5–21 for information on how to perform the test.

**Stratix GX HSSDC2 XCVR Test Overview**

The transceiver HSSDC2 design includes all of the Gigabit interface designs for the Stratix GX device. The HSSDC2 interface consists of a 4-channel Altera GXB transmitter/receiver block and a Verilog HDL block with the logic required to generate a PRBS and verify that it was received correctly. This design needs 2 HSSDC2 cables to complete the signal loopback.

For more information on the operation and parameterization of the GXB megafunction, refer to Quartus II Help.

The top-level BDF lets you easily modify the system parameters as desired to evaluate a particular system configuration. By varying the GXB megafunction parameters, you can adjust the per channel data rate up to 3,125 Mbps. The design has a Verilog HDL wrapper to name and place all of the pins and to provide proper termination for the signals. The Stratix GX pushbutton switches are used to control the start and stop of the data transmission, to insert errors, and to reset the circuit. The LEDs indicate the start of transmission, confirm that data was received correctly, indicate the error status, and the reset condition.

The main system clock (parallel data rate) is derived from the 156.25-MHz crystal oscillator using the GXB megafunction clock output. The data is generated in 20-bit words per channel using a PRBS generator with a repetition count of 31, which results in a serial data rate of 3.125 Gbps. The data is then sent to a 4 channel version of the Altera GXB megafunction, which converts the data into gigabit serial data streams. The megafunction uses the basic protocol option.

The HSSDC2 cables feed the serial data back to the receive inputs on the Stratix GX device. The data is converted back into parallel by the GXB megafunction. Because the design uses the GXB megafunction’s double
word feature, the byte alignment of the received data may be incorrect. To adjust it, the data is sent through a byte swap block based on synchronization data from the GXB megafuction. The data is then passed to a pattern detection block to find the start of data in the PRBS (the first word of the PRBS sequence) data stream. When this pattern is found, the data valid signal is asserted, which triggers an expected value PRBS generator to start. The two data streams are sent to a comparator to generate a match signal on a per channel basis. If the data streams match, the match LED illuminates on a per channel basis. If the match signal goes low while the data valid signal is high, the error flag is set and the error counter is incremented. Pressing the reset pushbutton resets the system state, error flag, and error count.

**Stratix GX HSSDC2 XCVR Functional Description**

Figure 7–16 shows the Stratix GX HSSDC2 XCVR logic diagram.

The GXB megafuction transmit PLL generates the system clock using the 156.25-MHz crystal as the reference. The PLL generates a 156.25-MHz clock to clock all of the data generation logic.
The transmit PRBS generator comprises 20 5-bit linear feedback shift registers. The output is taken from the MSB of each shift register. The initial seed value is 20’h695A7. When the enable (start) signal is high, the generator outputs a 31-word sequence that repeats until stopped. On reset, the seed value is initialized into all of the registers. This generator generates the data stream that exercises the system. Each transmit channel has its own PRBS generator.

The double word option requires synchronization of the control logic with the GXB receive section. This design uses a 10-bit alignment pattern (10’h1A7), which allows for the internal synchronization of the megafunction and a status signal that is sent to the word swap block in the control logic. Data transmission only starts when a channel has been synchronized on a per channel basis.

The data from the PRBS generators is sent to the GXB transmit block created using the Altera MegaWizard Plug-In Manager. The megafunction is configured as 4 channels running at 3,125 Mbps with an input clock rate 156.25 MHz. The signals are then sent to the HSSDC2 connectors and looped back to the Stratix GX device using 2 HSSDC2 cables. The transmit of one channel becomes the receive of the next.

The receive portion of the GXB megafunction on the Stratix GX device converts the serial data back to parallel 20-bit words. The received data is sent through a byte/word swap block that is controlled by the GXB megafunction. This function is required for double-word operation because the alignment pattern is only 10 bits. The data is then passed to the pattern detector to determine the start of the data packet. When the pattern detector has detected the synchronization pattern twice, it sets the data valid signal and starts passing the data to the comparator.

A second PRBS generator uses the data valid signal to start generating the expected data values. This second data set is also sent to the comparator. The comparator module takes the output from the pattern detector block and compares it with the output from the receive channel PRBS. The 20-bit words are compared each clock cycle. The comparator output is high if the words match. The output from each receive channel comparator is stored in a single bit match register. The output of this register drives the match LED on a per channel basis.

The error detection and counting blocks monitor the match and data valid signals. If the match signal goes low while data is valid, the error flag is set and the error counter is incremented. The reset pushbutton clears the error flag and reset the counter. The error insertion pushbutton inverts one bit in one data channel for one clock cycle, which is enough to trigger the error detection circuit.
Nios Designs

The Nios embedded processor is an Altera-based 32-bit RISC embedded configurable CPU for use with Altera FPGAs. It has an extensive suite of software development tools, Altera and third-party IP, and hardware development boards and kits. You can configure the Nios processor into either the Stratix or Stratix GX device as required.

You run Nios programs via the SOPC Builder SDK Shell interface. The shell is a UNIX-like command-line interface. Using the shell you can compile, run, and get statistics on the Nios programs and designs.

The information in this section assumes that you are familiar with the use and programming of Nios-based systems. Refer to the Altera web site (www.altera.com/nios) for more information.

The Nios-based designs were created using SOPC Builder, which is available in the Quartus II software. The process involves the following steps:

1. Use SOPC Builder to specify the Nios processor parameters.
2. Add the Avalon™ on-chip bus in SOPC Builder. The Avalon bus handles all communication between the CPU and the peripherals.
3. Attach customized peripherals to the Avalon on-chip bus. A UART allows communication to the outside world through the RS232 port. All of the peripherals are personalized via wizard tools.
4. Generate the system as a Verilog HDL or VHDL netlist. SOPC Builder generation also produces a software development kit (SDK) used for software development.
5. Compile, synthesize, and place-and-route using Quartus II software.

The SDK includes 3 directories:

- inc—The inc directory has the include files necessary to use the peripherals, some function prototypes, and Assembly language megafunctions.
- lib—The lib directory contains a makefile and archive, source, and object files for libraries usable by your Nios system.
- src—The src directory contains all user-developed source code. It also has some examples for use in testing the system.
For more information on the software development process, refer to the Nios literature page on the Altera web site.

**Stratix GX Nios DDR Test**

This design comprises a Nios processor combined with 2 dual-port RAMs, a test control state machine, and the Altera DDR Memory Controller MegaCore function. The design is captured in VHDL and has a VHDL wrapper to specify the I/O pin names and placements. Because this is a processor-based test, the actual test consists of running several object files on the Nios processor and observing the results in a terminal window on a PC.

The memory tests consist of a series of write, read, and compare cycles. These cycles write a value to a memory location, read the values back, and compare the expected value with the value read from the memory. If all of the values match, the test passes.

Refer to “DDR Interface (Stratix GX Nios DDR)” on page 5–22 for information on how to perform the test.

Figure 7–17 shows the Stratix GX Nios DDR logic diagram.

![Stratix GX Nios DDR Logic Diagram](image-url)
This design has 2 dual-port RAM modules controlled by the Nios processor. The command FIFO buffer handles all of the command sequences for the test state machine. The data RAM holds the data values used to write and read the memory. The DDR state machine reads the command sequences and the data values and generates the proper signals to control the DDR IP core. The DDR IP takes advantage of the built-in features of the Stratix GX device to implement the DDR interface at 200 MHz. These features include the on-chip termination and the ability to run the I/O banks at 2.5 volts.

For more information on the DDR SDRAM Controller MegaCore function, see the Altera web site.

The DDR test uses a C program that controls the commands and data patterns used during the memory test. The memory test comprises four tests that exercise reads and writes in various burst sizes.

- The first test, long burst test, writes 100 back-to-back bursts of eight words followed by reads of 100 back-to-back bursts of eight words. This test can use two different data patterns. One is an increment fs pattern and the other is an all 1s followed by all 0s pattern known as the worst-case pattern. For this test, the worst-case pattern was used.

- The second test, address bus walking 1s test, writes to the addresses 0x00000001, 0x00000010, 0x00000100, and so forth. After these addresses are written, they are read back and compared. This test uses an increment aa55 pattern.

- The third test, variable burst length test, writes small bursts (burst sizes of 2, 4, and 8) of varying data patterns then reads and compares. This test uses many data patterns, such as all es, increment fs, worst case, and increment aa55.

- The last test, full DIMM write then read test, writes the whole DIMM and then reads the whole DIMM and compares. For this test, the worst-case pattern was used.

All of these tests are run 20,000 times. If a mismatch is found in any of the tests, the processor reports the error to the shell. The test repeats until you stop it.
Stratix Nios PMC Test

This design comprises a Nios processor, a UART, and an Altera PCI core. The design is captured in VHDL and has a VHDL wrapper to specify the I/O pin names and placements. Because this a processor-based test, the actual test consists of running several object files on the Nios processor and observing the results in a terminal window on a PC.

The PMC interface test polls and initializes the attached PMC board. The board is queried for its ID. If the board is a registered PMC board, the shell displays the company name and board type.

Refer to “PMC Card Interface (Stratix Nios PMC)” on page 5–23 for information on how to perform the test.

Figure 7–18 shows the Stratix Nios PMC logic diagram.

Figure 7–18. Stratix Nios PMC Logic Diagram

This design has a Nios CPU, a UART for external communication, and an Altera PCI controller IP core. The PCI controller core acts as a bridge from the internal Nios-based Avalon bus to a 33-MHz, 32-bit PCI bus. The core can be configured as a master or target on the PCI bus. The core has a 32-bit PCI master/target interface, Nios target interface, DMA engine, FIFO buffers, interrupt controller, and a host/arbiter.

For more information on Altera PCI cores, refer to Processors & Peripherals in the IP section on the Altera web site.
The test is a C program that controls a series of PCI bus configuration transactions. It polls all of the available addresses and monitors the response. If it finds another board on the bus, it reports the manufacturer and board type.

**Stratix Nios Ethernet, On-Board Flash & EPC16 Flash Test**

These designs involve a Nios processor combined with several Avalon bus arbitration modules to handle all the I/O required to test the on-board flash memory, the 2 EPC16 configuration devices, and the built-in Ethernet interface. The design is captured in VHDL and has a VHDL wrapper to control the I/O pin names and placements. Because this a processor-based test, the actual test consists of running several object files on the Nios processor and observing the results in a terminal window on a PC.

The memory tests perform a series of write, read, and compare cycles. These cycles write a value to a memory location, read the values back, and compare the expected value with the value read from the memory. If all of the values match, the test passes.

The Nios processor tests the on-board LAN91C111 Ethernet MAC/PHY chip. The test initializes the 10/100 MAC/PHY chip and runs a set of Ethernet protocol tests using the **hello_plugs** program, which is a standard test program for this chip provided with the Nios embedded processor. The test results vary depending on whether the board is plugged into a functional network.

Refer to “Ethernet, On-Board Flash & EPC16 Flash Interface (Stratix Nios Ethernet) (Stratix Nios On-Board Flash) (Stratix Nios EPC16 Flash)” on page 5–25 for information on how to perform the test.

Figure 7–19 shows the Stratix Nios flash memory and Ethernet logic diagram for the Stratix device.
The Nios processor controls three Avalon bus slave modules. The first module handles all of the I/O signals and timing for the Stratix GX EPC16 configuration device. It controls the data, address, and controls signals and reports status to the Nios processor. The second module controls both the Stratix EPC16 device and the 4-Mbyte on-board flash chip. It is configured with extra address signals for the on-board flash chip. The third module interfaces with the LAN91C111 chip.

The test has two C programs: one to test the memories and one for the LAN chip. The memory tests write the address of a location to that location and read it back to confirm that the write operation was successful. It repeats this operation for each memory device. If a mismatch is found, it reports the error to the shell.

The Ethernet chip test runs the hello_plugs program. The test gives two different results depending on whether the board is attached to a network. If a board is not connected, the test initializes the chip and looks for the network, but because the board is not connected, the test generates timeouts. If a board is connected, the test uses DHCP to get a dynamic IP address for the board.
Stratix Nios 10/100 Ethernet Network-Interface Card I/O Test

This design is similar to the previous flash/Ethernet test because the card used for this test has the same MAC/PHY chip that is installed on the Stratix GX development board. The primary difference between the tests are the I/O pin assignments. The design is captured in VHDL and has a VHDL wrapper to control the I/O pin names and placements. Because this is a processor-based test, the actual test consists of running several object files on the Nios processor and observing the results in a terminal window on a PC.

The 10/100 Ethernet network-interface card has the LAN91C111 Ethernet MAC/PHY chip, which the Nios processor tests. The test initializes the 10/100 MAC/PHY chip and runs a set of Ethernet protocol tests from the hello_plugs program. The test results vary depending on whether the board is plugged into a functional network.

Refer to “10/100 Ethernet Network-Interface Card I/O Interface (Stratix Nios PROTO1 IO)” on page 5–29 for information on how to perform the test.

Figure 7–20 shows the Stratix Nios 10/100 Ethernet network-interface card I/O logic diagram for the Stratix device.
In this design, the Nios processor controls three Avalon bus slave modules. Although the flash interfaces are part of this design, they are not used during this test.

The first module handles all of the I/O signals and timing for the Stratix GX EPC16 configuration device. It controls the data, address, and controls signals, and reports status to the Nios processor. The second Avalon bus module controls both the Stratix EPC16 device and the 4-Mbyte on-board flash chip. It is configured with extra address signals for the on-board flash chip. The third module interfaces with the 10/100 Ethernet network-interface card I/O connector interface.

The 10/100 Ethernet network-interface card I/O test involves running the `hello_plugs` program. The test gives two different results depending on whether the board is attached to a network. If a board is not connected, the test initializes the chip and looks for the network, but because the board is not connected, the test generates timeouts. If a board is connected, the test uses DHCP to get a dynamic IP address for the board.
**Stratix Nios Compact Flash Test**

The design is captured in a top-level BDF that controls the I/O pin names and placements. Because this is a processor-based test, the actual test consists of running several object files on the Nios processor and observing the results in a terminal window on a PC.

The test uses the provided 32-MByte Kingston Technology compact flash card. The test initializes the compact flash card and writes a sector of random data to each usable sector and reads it back with a comparison.

Refer to “Compact Flash Interface (Stratix Nios Compact Flash)” on page 5–32 for information on how to perform the test.

Figure 7–21 shows the Stratix Nios compact flash interface logic diagram for the Stratix device.

---

**Figure 7–21. Stratix Nios Compact Flash Interface Logic Diagram**

In this design, the Nios processor controls an Avalon bus slave module that interfaces to the compact flash card using the shared data bus of the 10/100 Ethernet network-interface card I/O connector interface. Because these two interfaces share the same bus, you cannot use both interfaces at the same time.
This section provides demonstrations for use with the High-Speed Development Kit, Stratix GX Edition.

This section includes the following chapters:

- Chapter 8. Stratix GX SPI-4.2 Demonstration with 10-Port Gigabit Ethernet MAC
- Chapter 9. 10-Gigabit Ethernet MAC Demonstration

**Revision History**

The table below shows the revision history for these chapters.

<table>
<thead>
<tr>
<th>Chapter(s)</th>
<th>Date / Version</th>
<th>Changes Made</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 - 9</td>
<td>July 2003</td>
<td>First publication.</td>
</tr>
</tbody>
</table>
8. Stratix GX SPI-4.2 Demonstration with 10-Port Gigabit Ethernet MAC

The CD-ROM included with this kit has a device programming file that demonstrates Stratix GX SPI-4.2 interoperability with the Intel® IxF1110 and PMC-Sierra S/UNI®-10xGE 10-port Gigabit Ethernet MAC devices. The demonstration configures the Stratix GX device with Altera’s SPI-4.2-compliant POS-PHY Level 4 MegaCore function with the receiver connected to the transmitter over the Atlantic™ interface for loopback testing.

Table 8–1 describes the POS-PHY Level 4 parameters used in this demonstration.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Receiver</th>
<th>Transmitter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Family</td>
<td>Stratix GX</td>
<td>Stratix GX</td>
</tr>
<tr>
<td>Dynamic Phase Alignment</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>LVDS Data Rate</td>
<td>800 Mbps</td>
<td>800 Mbps</td>
</tr>
<tr>
<td>Data Path Width</td>
<td>128 bits</td>
<td>128 bits</td>
</tr>
<tr>
<td>Data Flow Direction</td>
<td>RX</td>
<td>TX</td>
</tr>
<tr>
<td>Number of Ports</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Buffer Mode</td>
<td>Shared FIFO with Embedded Addressing</td>
<td>Shared FIFO with Embedded Addressing</td>
</tr>
<tr>
<td>FIFO Buffer Size</td>
<td>8,192 bytes</td>
<td>8,192 bytes</td>
</tr>
<tr>
<td>Almost Empty</td>
<td>512 bytes</td>
<td>-</td>
</tr>
<tr>
<td>Almost Full</td>
<td>1,024 bytes</td>
<td></td>
</tr>
<tr>
<td>MaxBurst1</td>
<td>-</td>
<td>512 bytes</td>
</tr>
<tr>
<td>MaxBurst2</td>
<td>-</td>
<td>1,024 bytes</td>
</tr>
<tr>
<td>Atlantic Data Width</td>
<td>128 bits</td>
<td>128 bits</td>
</tr>
<tr>
<td>FIFO Threshold Low</td>
<td>32 bytes</td>
<td>-</td>
</tr>
<tr>
<td>FIFO Threshold High</td>
<td>-</td>
<td>320 bytes</td>
</tr>
<tr>
<td>Transmit Bandwidth Optimization</td>
<td>-</td>
<td>No</td>
</tr>
<tr>
<td>Burst Mode</td>
<td>-</td>
<td>No</td>
</tr>
<tr>
<td>Burst Size</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Multiple Continues</td>
<td>Yes</td>
<td>-</td>
</tr>
</tbody>
</table>
The demonstration assumes that the Stratix GX development board is connected through the HM-Zd connector on the source-synchronous side to a development board from Intel or PMC-Sierra that includes the multi-channel MAC device. For testing with Intel, you can use the IXD1110 development board with direct interconnect through the reciprocal HM-Zd connectors. For PMC-Sierra, you can use an interposer card with the PM2381-KIT development board to bridge the HM-Zd connector to the Molex 74057-1001 connector used on the PMC-Sierra board.

Contact your local Altera representative for information on using a PMC-Sierra interposer card with the development board.

To generate and monitor data traffic, you need an industry-standard Gigabit Ethernet stimulus, capture, and analysis system. Optionally, you can use an Agilent logic analysis system to probe the SPI-4.2 data in both the receive and transmit directions.

Figure 8–1 illustrates the demonstration set-up using the Intel IXD1110 development board as an example.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Receiver</th>
<th>Transmitter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Training Sequence Interval</td>
<td>-</td>
<td>32,000</td>
</tr>
<tr>
<td>Training Pattern Repetitions</td>
<td>-</td>
<td>32</td>
</tr>
<tr>
<td>Status Channel Clock Edge</td>
<td>Negative</td>
<td>Negative</td>
</tr>
<tr>
<td>Calendar Multiplier</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
The following instructions describe how to set up the Stratix GX development board and Intel BenNevis IXF 1110 for use with the demonstration.

1. Turn off power to the Stratix GX development board.
   a. Ensure that SW3 is in the off position (which is down, when viewing the board in an upright position).
   b. Ensure that SW2 GX_BYPASS is in the no position.
   c. Ensure that SW1 STRATIX_BYPASS is in the yes position.

2. Connect power to the Stratix GX development board.
   a. Connect the power supply cable to connector J31.
   b. Connect the ByteBlaster II cable to connector J87.

3. Connect power to Intel BenNevis board. Boards may vary, so refer to the Intel installation guide for instructions.
4. Connect Ethernet and serial links to the BenNevis board. Boards may vary, so refer to the Intel installation guide for instructions.

5. Connect fibre optic cables to all optic devices on the BenNevis board. Connect all optic ports to optic modules and cables. Optically loop back unused ports.

6. Connect the BenNevis board to the Stratix GX development board. The Ben Nevis connector at J28 connects to the Stratix GX development board connector at J108. This connection is keyed and may require the boards to be arranged to stand on their edges in an L-shaped pattern.

7. Turn on power to Stratix GX development board by switching SW3 to the on position. If both boards are connected to the same switching power supply (included), both boards should power up simultaneously when you turn on SW3.

8) Auto-detect and program the Stratix GX device.
   a. Run the Quartus II software.
   b. Choose Programmer (Tools menu).
   c. In the Programmer, click AutoDetect.
   d. Double-click the filename next to the Stratix GX device in the chain of devices.
   e. Select the SRAM Object File (.sof) from the <path>\Stratix_GX_kit\Demonstrations\SPI4 directory.
   f. Turn on the Program/Configure option next to the Stratix GX device.
   g. Press the Start button to configure the device.

8. Soft-reset the BenNevis board.
   a. Press the soft-reset button on the BenNevis board (S1).
   b. Wait 30 seconds for the BenNevis board to initialize. When the LEDs flash in unison, the initialization is complete.

Do not press S15 (labeled reset on the board) or the device will lock up.

10. Start Ethernet traffic incrementally. Traffic from a Gigabit Ethernet tester should have ports enabled one at a time for the Multi-PHY core.
9. 10-Gigabit Ethernet MAC Demonstration

Introduction

This demonstration application implements a 10-Gigabit Ethernet MAC connected to the Stratix GX embedded SERDES device, providing an XAUI connection to an optional XPAK optical transceiver module. The demonstration allows you to test the XAUI SERDES implementation with Ethernet LAN bandwidth.

The demonstration provides the Ethernet MAC with 2-Kbytes of FIFO buffers per direction, implementing a store-and-forward mechanism to transfer received frames back to the transmit port. Statistics are collected on the traffic received. A small, integrated Nios processor accesses these statistics and presents them on a standard RS-232 terminal. Table 9–1 shows the configuration for the Ethernet MAC core used in the demonstration.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit FIFO buffer</td>
<td>256 deep x 64 wide (2 Kbytes)</td>
<td>Can hold standard Ethernet frame.</td>
</tr>
<tr>
<td>Transmit FIFO buffer section thresholds</td>
<td>0</td>
<td>The TX FIFO buffer is configured in store-and-forward mode. Transmission begins after a complete frame is available in the transmit FIFO buffer.</td>
</tr>
<tr>
<td>Receive FIFO buffer</td>
<td>256 deep x 64 wide (2Kbyte)</td>
<td>Can hold complete standard Ethernet frame.</td>
</tr>
<tr>
<td>Receive FIFO buffer section available</td>
<td>32 (256 byte)</td>
<td>The RX FIFO buffer is configured to forward frames to the transmit FIFO buffer without waiting for end-of-frame if the frame exceeds 256 bytes. If less than 256 byte frames are received, the frame is forwarded with reception of end-of-frame.</td>
</tr>
<tr>
<td>RX FIFO threshold for automatic pause frame generation</td>
<td>128 (1024 byte)</td>
<td>As the RX FIFO buffer is programmed in forwarding mode, it usually will not reach this limit and the MAC operates normally. To force the receive FIFO buffer to fill up, the transmitter can be put on hold by generating a pause frame from the Testset.</td>
</tr>
<tr>
<td>Pause quanta</td>
<td>32767</td>
<td>If the MAC transmits pause frames, the pause quanta is set to this value.</td>
</tr>
<tr>
<td>MAC address</td>
<td>Disabled</td>
<td>MAC address recognition and insertion is disabled. All frames received are forwarded unmodified.</td>
</tr>
<tr>
<td>CRC remove on receive</td>
<td>Enabled</td>
<td>The received frame CRC is verified and removed from the frame.</td>
</tr>
</tbody>
</table>
Figure 9–1 shows the application overview. A separate statistics module implements 32-bit counters for received traffic statistics (e.g., packets received, errors received). You can access these statistics through an RS-232 connection that is served by an embedded Nios processor and a small control application running on the processor. A normal RS-232 terminal program is sufficient to access the port. The RS-232 connection is optional, that is, if it is not connected, the loopback operation is fully functional as no specific configuration application needs to run on the Nios processor.

The on-board Stratix GX LEDs indicate fault and traffic conditions.

Figure 9–1. Application Overview

The demonstration assumes that the optional XPAK module is connected to the board and the 156.25 MHz clock oscillator is enabled.
Contact

For more information, contact MorethanIP using one of the following methods.

E-Mail: info@morethanip.com
Internet: www.morethanip.com

Europe
An der Steinernen Bruecke 1
D-85757 Karlsfeld
Germany
Tel : +49 (0) 8131 333939 0
Fax : +49 (0) 8131 333939 1

North America
2130 Gold Street Ste. 250
Alviso, CA 95002
USA
Tel : +1 408 273 4567
Fax : +1 408 273 4667