The reference designs demonstrate wire-speed operation of the Altera® 10-Gbps Ethernet (10GbE) reference design component described in AN516: 10-Gbps Ethernet Reference Design; one using Arria® II GX devices and the other using Stratix® IV GX devices. The reference designs are SOPC Builder systems that include one instance of a custom 10GbE component configured as a media access control (MAC) with a XAUI transceiver.

The designs provide a flexible test and demonstration platform on which you can control, test, and monitor 10GbE operations in different loopback options. You can use the reference designs in a stand-alone configuration or in a multi-equipment configuration with an external 10-Gbps Ethernet legacy system or standard test equipment.

The 10GbE hardware demonstration reference design has the following features:

- Stand-alone and easy-to-use reference design example with flexibility to support two-system interoperability in various configurations.
- Requires minimal hardware.
- Supports 10GbE operations in XAUI mode.
- Supports programmable settings for number of packets, packet length, and payload-data type.
- Demonstrates transmission and reception of Ethernet packets at the maximum theoretical data rates without errors, through external loopback, internal loopback, or both types of loopback path.
- Supports different loopback mechanisms including external loopback through a CX4 loopback device, external loopback through X2 modules with a loopback optical cable assembly, external loopback through an Altera high-speed mezzanine connector (HSMC) loopback board, internal physical medium attachment (PMA) loopback, and internal MAC loopback.
- Includes support for gathering throughput statistics.
- Provides custom command-line interface (CLI) commands to control the design, monitor transmitted and received packets, and generate and display statistics information from within the Nios II IDE.

**General Description**

Each reference design uses an Altera development kit—Arria II GX or Stratix IV GX FPGA—in a hardware platform that is capable of supporting different loopback mechanisms.

The reference designs also provide software, including drivers, programming information, and a custom command-line interface (CLI) to control the hardware.
Ethernet Packet Loopback Datapath

The 10GbE hardware demonstration reference designs support several different types of loopback. This flexibility supports applications for various devices and is very useful for functional and performance demonstrations. Figure 1 shows the different loopback options that are available.

Figure 1. Loopback Options in the 10GbE Hardware Demonstration Reference Design

The loopback type is configurable at run time at the beginning of each test. Only one loopback option should be configured to be active at any given time. There are two user loopback modes:

- The following options for transmit-to-receive (Tx-to-Rx) local loopback without the need for any external test equipment:
  
  A. External CX4 loopback using a HSMC to CX4 adapter card and a CX4 loopback testing device. Using this local external loopback option, you can validate the operation of the Altera 10GbE MAC, XAUI, physical coding sublayer (PCS), and PMA (serial transceivers) through an external IEEE 802.3 standard CX4 (XAUI) cable assembly.
  
  B. External fiber optic cable loopback (SC-to-SC simplex cable) at X2 module. Using this local external loopback option allows you to easily validate the operation of the Altera 10GbE MAC, XAUI, PCS, and PMA (serial transceivers) through an external IEEE 802.3 standard XAUI and X2 optical plug-in module.
  
  C. External loopback using a HSMC loopback board.
  
  D. Internal PMA loopback. Using local internal loopback at the serial transceiver, you can validate the operation of the Altera 10GbE MAC, XAUI, PCS, and PMA (serial transceivers) through the XAUI PMA.
  
  E. Internal MAC loopback. This local internal MAC loopback at the internal XGMII interface, you can validate the operation of an Altera 10GbE MAC.

- The following option for receive-to-transmit (Rx-to-Tx) line loopback:
  
  F. Internal line loopback outside the MAC. This option is used to check the interoperability of the Altera 10GbE component and another external 10GbE component. Data generation and checking is done by the external 10GbE component.
HSMC to CX4 adapter cards are available from MoreThanIP. Refer to their website at: http://www.morethanip.com/boards_cx4.htm. HSMC 10 Gbps X2 boards are not a generally available product but are available for demonstration only. For more information, contact your local Altera sales office or sales representative.

Interoperability Usage Examples

Figure 2 shows a one-port configuration example in which another external MAC uses Rx-to-Tx line loopback or external test equipment to check the interoperability of the Altera 10-Gbps Ethernet component and external 10 Gbps Ethernet logic using a HSMC to CX4 adapter card.

Packet generation and checking are performed by the Altera FPGA and the 10GbE hardware demonstration reference design.

Figure 2. One-Port Configuration Example, External MAC Rx-to-Tx Line Loopback

Figure 3 shows a one-port configuration example in which Altera MAC Rx-to-Tx line loop-back checks the interoperability of the Altera 10GbE component and an external 10 Gbps Ethernet logic using X2 optical modules. Packet generation and checking are performed by user logic.

Figure 3. One-Port Configuration Example, Altera MAC Rx-to-Tx Line Loopback
Figure 4 shows a two-port configuration example that performs a two-port loopback data integrity and throughput test using a HSMC to CX4 adapter card. This test also verifies the correct operation of the PCS clock rate difference compensation and MAC flow control in both systems.

**Figure 4.** Two-Port Loopback Test Configuration Example

![Figure 4. Two-Port Loopback Test Configuration Example](image)

Figure 5 shows the connections for two Stratix IV GX development boards connected using a standard CX4 copper cable assembly.

**Figure 5.** Connections for Two-Port Loopback Test

![Figure 5. Connections for Two-Port Loopback Test](image)
Figure 6 shows a two-port configuration example that illustrates 10GbE reference design interoperability with a standard 10GbE switch using X2 optical modules.

**Figure 6.** Two-Port Configuration Example, 10GbE Switch Interoperability
Figure 7 shows the connections for an example system with two Stratix IV GX development boards connected to a Cisco Catalyst 3560E Ethernet switch with two 10GbE ports. In this example, one board is connected to the Cisco switch using an HSMC-to-CX4 adapter and a standard CX4 cable assembly; the second board is connected to the Cisco switch using an X2 module and an SC-SC duplex fiber-optic cable assembly.

Figure 7. Connections for Two-Port 10GbE Switch Interoperability
Reference Design Overview

Figure 8 shows a high level block diagram of the reference design.

**Figure 8. 10GbE Hardware Demonstration Reference Design Block Diagram**

![Block Diagram of the Reference Design](image-url)
In local loopback configurations (Tx-to-Rx loopback), the 10GbE hardware demonstration reference design generates a stream of packets from the Ethernet Packet Generator and the Ethernet Packet Monitor block verifies incoming packet checksums and monitors packet receipt.

You can use the 10GbE hardware demonstration reference design stand-alone in local loopback options A, B, C, D, and E as described in “Ethernet Packet Loopback Datapath” on page 2.

In local loopback configuration, after system initialization, the Ethernet Packet Generator begins generating packets to the transmitting 10GbE reference design function through the Transmit FIFO interface. The 10GbE reference design MAC prefixes a 1-byte Start field, a 6-byte Preamble, and a 1-byte SFD. It then appends a 4-byte FCS to each packet and transmits the packet through the loopback datapath to the receiving function of the 10GbE component.

For information about the output frame format, refer to Figure 11 on page 12.

After the receiving 10GbE component receives a packet, the component verifies the packet with its FCS. The remainder of the packet is streamed out through the Receive FIFO interface to the Ethernet Packet Monitor, which verifies the checksum at the end of and inside the payload.

After the system receives the total number of expected packets, it reports the packet statistics and the throughput rate based on the statistics registers. This information confirms the real throughput rate that the 10GbE reference design can achieve.

**Functional Description**

Figure 9 on page 9 shows the SOPC Builder components that comprise the 10GbE hardware demonstration reference design system.
The 156.25 MHz clock required by the 10GbE reference design component is created when you generate the IP component.

For more information about the 156.25 MHz clock, refer to AN516: 10-Gbps Ethernet Reference Design.

The following sections describe the roles of the main components in the system, and describe the loopback datapath, the memory map, the system-specific registers, and the top-level interface signals of the SOPC Builder system.

**Nios II Embedded Processor**

The Nios II embedded processor is the control plane for setting up and configuring the system components. The embedded processor triggers the start of Ethernet packet generation and monitors the status of packet reception. Figure 9 shows the interconnections to the embedded processor.
**On-Chip Memory**

This memory stores the executable program. After the program is compiled, it is stored in this memory and executed by the Nios II embedded processor. The on-chip memory is a 256-KByte RAM.

**Phase-Locked Loop (PLL)**

The phase-locked loop (PLL) component takes its input clock signal from a 100-MHz crystal on the development board and generates the PLL output 83.33-MHz clock (sys_clk). This clock is the system-wide clock source for the SOPC Builder system, and is the only PLL required by the 10GbE hardware demonstration reference design.

**JTAG UART**

The JTAG UART component sends serial character streams between the Nios II embedded processor and the rest of the SOPC Builder system. This component provides the mechanism for the FPGA to communicate with the nios2-terminal. The component provides a simple Avalon Memory-Mapped (Avalon-MM) interface to the JTAG interface. The Nios II embedded processor communicates with the 10GbE component by reading and writing control and data registers.

For information about Avalon Memory-Mapped interfaces, refer to the *Avalon Interface Specifications*.

**Interval Timer**

The interval timer component is a 32-bit timer that the Nios II processor system uses to calculate the performance and throughput of the 10GbE operations.

**Clock Crossing Bridge**

The Avalon-MM clock-crossing bridge allows you to connect Avalon-MM masters and slaves that operate in different clock domains.

**Altera 10-Gbps Ethernet Component**

The 10GbE component is an instance of the Altera 10GbE reference design. This component provides an integrated Ethernet MAC, PCS, and PMA solution for Ethernet applications. The component transmits Ethernet packets from an Avalon-ST interface to four lanes of 3.125-Gbaud XAUI serial transceiver interface, and receives packets sent to it on the XAUI interface.

For additional information about the 10GbE reference design, refer to *AN516: 10-Gbps Ethernet Reference Design*. For information about Avalon Streaming interfaces, refer to the *Avalon Interface Specifications*. 
**Ethernet Packet Generator**

The Ethernet Packet Generator module is a custom SOPC Builder component. The generator has an Avalon-MM slave interface for control signals and Avalon-ST source interface for Ethernet packet information. This generator drives the 10GbE component Transmit FIFO interface, by transmitting a stream of Ethernet packets to the 10GbE Transmit FIFO.

Figure 10 shows a high level block diagram of the Ethernet Packet Generator module. The module contains the following components: Avalon-MM register, Ethernet packet generation block, Altera CRC Generator MegaCore function, and Shift Register (RAM-based) megafunction.

**Figure 10.** Ethernet Packet Generator Block Diagram

The Avalon-MM slave interface provides a register interface for the system to configure all of the parameters and settings necessary for Ethernet packet generation. After the settings are configured, the Ethernet Packet Generator starts generating and transmitting a stream of Ethernet packets to the 10GbE reference design module.

The system configures the following programmable settings through the register interface:

- Total number of packets to be transmitted
- Incremental or random data type
- Fixed or random packet length
- Source and destination MAC address
- Random seed for the pseudo-random binary sequence (PRBS) generation block

In addition, the Avalon-MM register provides the status of the transmit operation and reports the number of packets that were successfully transmitted. Refer to Table 2 on page 15 for details of the Ethernet Packet Generator registers.

After the system configures all the register settings, it sets the start bit of the Operation register to 1. Setting this start bit resets the Ethernet packet generation block state machine and all the Ethernet Packet Generator status registers. The Ethernet packet generation block generates an Ethernet packet header and a data payload for each packet.
To abort transmission, you can set the stop bit of the Operation register by pressing Enter. When the stop bit is asserted, the state machine completes generating the current packet and then stops.

The Ethernet packet generation block sends each packet to the CRC Generator MegaCore function and to the RAM-based shift register megafunction, ALTSHIFT_TAPS. The CRC Generator MegaCore function calculates the checksum for the packet, and the RAM-based shift register stores the packet until the checksum is available. After the Ethernet Packet Generator merges the valid CRC checksum with the packet stream, it sends the complete packet to the Avalon-ST interface.

Embedding the CRC checksum in the generated packet allows the receive interface to verify the packet easily. If a packet is dropped, the system can continue verifying the incoming packets based on their individual embedded CRC checksum values. This method enables the system to keep an accurate count of the total number of packets received. The packets that the Ethernet Packet Generator generates are standard Ethernet packet contents—Ethernet packets without the 1-byte Start field, 6-byte Preamble, 1-byte Start Frame Delimiter (SFD), and 4-byte MAC-calculated Frame Check Sequence (FCS) that the 10GbE component adds later.

Figure 11 on page 12 describes the format of the frames generated by the Ethernet Packet Generator.

You program the destination and source MAC addresses through the register interface using the benchmark application (See “Running the Benchmark Application Software” on page 33). You can set the destination MAC address to be a unicast or a broadcast address. In the 10GbE hardware demonstration reference design, a unicast destination MAC address must be the 10GbE component’s receiving MAC address. The source MAC address must be programmed to be the 10GbE component’s transmitting MAC address. This practice ensures that the packets are transmitted to the 10GbE component correctly and that the component can verify the contents of the packets it receives.

The packet length can be set to a programmable value or set to be random packet size. A fixed packet length can range from 24 to 9600 bytes, and a random packet length can range from 24 to 1518 bytes. A packet with length set to less than 64 bytes is automatically padded with padding bytes (0x00) to make it at least 64 bytes long.
The data payload can be incremental or pseudo-random. If you select the incremental data setting, the payload's initial data value is 0x00, and each subsequent data value is incremented by 1. However, if the pseudo-random data type is selected, the random value generated by the PRBS block becomes the data content of the payload. To program the random seed used by the PRBS block, set the `rand_seed0`, `rand_seed1`, and `rand_seed2` registers.

The 2-byte sequence number and 4-byte CRC checksum are part of the packet payload in the 10GbE hardware demonstration reference design. The minimum payload length for packets is therefore six bytes. The sequence number is stored in the first two bytes of every packet payload and is used to keep track of the sequence of packets received for debugging purposes. The CRC checksum calculated by the CRC Generator MegaCore function is stored in the final four bytes of the packet payload. The checksum increases confidence in the data integrity of the packets received by the Ethernet Packet Monitor.

For more information about the CRC Generator MegaCore function, refer to the CRC Compiler User Guide. For more information about the RAM-based shift register megafuction, ALTSHIFT_TAPS, refer to the Shift Register (RAM-Based) (ALTSHIFT_TAPS) Megafunction User Guide.

**Ethernet Packet Monitor**

The Ethernet Packet Monitor module is a custom SOPC Builder component. The monitor, which is driven by the 10-Gbps Ethernet component Receive FIFO interface, has an Avalon-MM slave interface for control signals and an Avalon-ST sink interface for receiving Ethernet packets. This monitor receives the stream of Ethernet packets and verifies the contents of the payload.

Figure 12 shows a block diagram of the Ethernet Packet Monitor. This module contains the Altera CRC Checker MegaCore function and an Avalon-MM register space.

**Figure 12.** Ethernet Packet Monitor Block Diagram
The Ethernet packets that arrive at the receiving 10GbE component in the 10GbE hardware demonstration reference design are streamed directly to the CRC Checker MegaCore function in the Ethernet Packet Monitor module. The CRC Checker MegaCore function calculates the correct CRC checksum for the received packet and verifies the calculated value against the checksum value embedded in the final four bytes of the packet payload. It then outputs a status signal, crcbad, that identifies whether the packet received is good or corrupted, and updates the statistics registers accordingly.

The Avalon-MM register file provides a register interface for the system to configure all of the settings necessary to start and monitor Ethernet packet reception. Through this register interface, the system can configure the total number of packets to be received. The Ethernet Packet Monitor module provides the status of the receive operation and a set of statistics counters that track the number of good packets received, the number of bad packets received, the number of bytes received, and the number of clock cycles. This information is used to calculate the performance and throughput rate of the 10GbE hardware demonstration reference design.

Refer to Table 5 on page 17 for descriptions of the fields in the Ethernet Packet Monitor register.

After the system configures the register settings, it sets the start bit of the Ethernet Packet Monitor Receive Control and Status register. Setting this start bit resets the packet reception statistics counters and enables the Ethernet Packet Monitor to receive incoming packets. When the stop bit of the Receive Control and Status register is asserted, the Ethernet Packet Monitor stops receiving packets and stops updating the statistics counters. Refer to Table 6 on page 18 for descriptions of the fields in the Ethernet Packet Monitor Receive Control and Status register.

For more information about the CRC Checker MegaCore function, refer to the CRC Compiler User Guide.

**Ethernet Multiplexer**

The Ethernet Multiplexer is a custom SOPC Builder module. The main function of this component is to select the Ethernet transmit-side traffic source. The source can be either from the Ethernet Packet Generator or packet loopback from the receive side. Because the SOPC system is point-to-point, each Avalon-ST source needs to have one Avalon-ST sink. There are two Avalon-ST sinks and two Avalon-ST sources in the Ethernet Multiplexer module to achieve the Ethernet transmit-side traffic source selection mechanism.

**Control Features**

This section describes the details of the 10GbE hardware demonstration reference design control features. It includes the following topics:

- Memory Map
- Register Map
- Interface Signal Descriptions
Memory Map

Table 1 describes the memory map of the SOPC Builder system for the 10GbE hardware demonstration reference design.

<table>
<thead>
<tr>
<th>Address</th>
<th>SOPC Builder Component Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00040000 – 0x0007FFFF</td>
<td>onchip_mem</td>
<td>On-Chip Memory - RAM (256KB)</td>
</tr>
<tr>
<td>0x00080000 – 0x00080FFF</td>
<td>cpu</td>
<td>CPU JTAG Debug Module</td>
</tr>
<tr>
<td>0x00081000 – 0x000813FF</td>
<td>eth_10ginst</td>
<td>10-Gbps Ethernet</td>
</tr>
<tr>
<td>0x00081800 – 0x0008183F</td>
<td>gen</td>
<td>Ethernet Packet Generator</td>
</tr>
<tr>
<td>0x00081840 – 0x0008187F</td>
<td>mon</td>
<td>Ethernet Packet Monitor</td>
</tr>
<tr>
<td>0x00081880 – 0x0008189F</td>
<td>pll</td>
<td>Phased-locked loop</td>
</tr>
<tr>
<td>0x000818A0 – 0x000818BF</td>
<td>timer</td>
<td>Interval timer 0</td>
</tr>
<tr>
<td>0x000818C0 – 0x000818DF</td>
<td>timer_1</td>
<td>Interval timer 1</td>
</tr>
<tr>
<td>0x000818E0 – 0x000818E7</td>
<td>jtag_uart</td>
<td>JTAG UART</td>
</tr>
<tr>
<td>0x00100000 – 0x001FFFFF</td>
<td>clock_crossing_bridge</td>
<td>Avalon-MM clock crossing bridge</td>
</tr>
</tbody>
</table>

Register Map

This section provides detailed information about the Ethernet Packet Generator and Ethernet Packet Monitor registers.

Table 2 describes the Ethernet Packet Generator registers. Table 3 on page 16 describes the fields of the Ethernet Packet Generator Configuration Setting register. Table 4 on page 17 describes the fields of the Ethernet Packet Generator Operation register.

<table>
<thead>
<tr>
<th>Address Offset</th>
<th>Register Name</th>
<th>Number of Bits</th>
<th>Description</th>
<th>Access</th>
<th>Hardware Reset</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>number_packet</td>
<td>32</td>
<td>Number of packets register. The total number of packets generated by the Ethernet Packet Generator and sent to the 10GbE component.</td>
<td>RW</td>
<td>0x0</td>
</tr>
<tr>
<td>0x04</td>
<td>config_setting</td>
<td>16</td>
<td>Configuration setting register. Refer to Table 3 on page 16 for register field descriptions.</td>
<td>RW</td>
<td>0x0</td>
</tr>
<tr>
<td>0x08</td>
<td>operation</td>
<td>3</td>
<td>Operation register. Refer to Table 4 on page 17 for register field descriptions.</td>
<td>RW/RO</td>
<td>0x0</td>
</tr>
<tr>
<td>0x10</td>
<td>source_addr0</td>
<td>32</td>
<td>Lower source MAC address register. Bits [31:0] of the source MAC address field in the Ethernet packet. This source MAC address should be programmed to the MAC address of the transmitting 10GbE component.</td>
<td>RW</td>
<td>0x0</td>
</tr>
<tr>
<td>0x14</td>
<td>source_addr1</td>
<td>16</td>
<td>Upper source MAC address register. Bits [15:0] of this register hold bits [47:32] of the source MAC address field in the Ethernet packet. This source MAC address should be programmed to the MAC address of the transmitting 10GbE component. Bits [31:16] of this register are reserved.</td>
<td>RW</td>
<td>0x0</td>
</tr>
</tbody>
</table>
Table 2. Ethernet Packet Generator Register Map (Part 2 of 2)

<table>
<thead>
<tr>
<th>Address Offset</th>
<th>Register Name</th>
<th>Number of Bits</th>
<th>Description</th>
<th>Access</th>
<th>Hardware Reset</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x18</td>
<td>destination_addr0</td>
<td>32</td>
<td>Lower destination MAC address register. Bits [31:0] of the destination MAC address field in the Ethernet packet. This destination MAC address should be programmed to the receiving 10GbE component's MAC address for unicast packets.</td>
<td>RW</td>
<td>0x0</td>
</tr>
<tr>
<td>0x1C</td>
<td>destination_addr1</td>
<td>16</td>
<td>Upper destination MAC address register. Bits [15:0] of this register hold bits [47:32] of the destination MAC address field in the Ethernet packet. This destination MAC address should be programmed to the receiving 10GbE component's MAC address for unicast packets. Bits [31:16] of this register are reserved.</td>
<td>RW</td>
<td>0x0</td>
</tr>
<tr>
<td>0x24</td>
<td>packet_tx_count</td>
<td>32</td>
<td>Packet transmit count register. The current count of the data packets successfully transmitted by the Ethernet Packet Generator. This register resets when operation[0] is asserted.</td>
<td>RO</td>
<td>0x0</td>
</tr>
<tr>
<td>0x30</td>
<td>rand_seed0</td>
<td>32</td>
<td>Lower random seed register. The value in this register is preloaded in bits [31:0] of the PRBS generator when config_setting[15] is set to 1.</td>
<td>RW</td>
<td>0x0</td>
</tr>
<tr>
<td>0x34</td>
<td>rand_seed1</td>
<td>32</td>
<td>Middle random seed register. The value in this register is preloaded in bits [63:32] of the PRBS generator when config_setting[15] is set to 1.</td>
<td>RW</td>
<td>0x0</td>
</tr>
<tr>
<td>0x38</td>
<td>rand_seed2</td>
<td>32</td>
<td>Upper random seed register. The value in this register is preloaded in bits [91:64] of the PRBS generator when config_setting[15] is set to 1.</td>
<td>RW</td>
<td>0x0</td>
</tr>
</tbody>
</table>

Table 3. Configuration Setting Register Field Descriptions

<table>
<thead>
<tr>
<th>Bits</th>
<th>Field Name</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>length_sel</td>
<td>Packet-length type select. The value in this bit determines the packet-length type according to the following list:</td>
<td>RW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0: Fixed packet length</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1: Random packet length</td>
<td></td>
</tr>
<tr>
<td>[14:1]</td>
<td>pkt_length</td>
<td>Fixed packet length. The programmable packet length is specified in bytes, and ranges from 24 to 9600. This field is valid only when the Packet-length type select bit has value 0.</td>
<td>RW</td>
</tr>
<tr>
<td>15</td>
<td>pattern_sel</td>
<td>Data type select. The value in this bit determines the data pattern type according to the following list:</td>
<td>RW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0: Incremental data pattern</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1: Random data pattern</td>
<td></td>
</tr>
<tr>
<td>[31:16]</td>
<td>Reserved</td>
<td>Reserved. Read returns 0.</td>
<td>RO</td>
</tr>
</tbody>
</table>
Table 4. Operation Register Field Descriptions

<table>
<thead>
<tr>
<th>Bits</th>
<th>Field Name</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>start</td>
<td>Start operation. Set to 1 to start the packet generation from the Ethernet Packet Generator to the 10GbE component. This bit is automatically cleared after packet generation begins.</td>
<td>RW</td>
</tr>
<tr>
<td>1</td>
<td>stop</td>
<td>Stop operation. Set to 1 to stop the Ethernet Packet Generator from generating and sending further packets to the 10GbE component. The generator completes the current packet after this bit is asserted. The stop bit is cleared when the Operation register start bit is set to 1.</td>
<td>RW</td>
</tr>
<tr>
<td>2</td>
<td>tx_done</td>
<td>Transmit-done status. This bit is set to 1 when the Ethernet Packet Generator completes transmission of all the packets programmed in the Number of Packets register. This bit is cleared when the Operation register start bit is set to 1.</td>
<td>RO</td>
</tr>
<tr>
<td>[31:3]</td>
<td>Reserved</td>
<td>Reserved. Read returns 0.</td>
<td>RO</td>
</tr>
</tbody>
</table>

Table 5 describes the Ethernet Packet Monitor registers. Table 6 on page 18 describes the fields of the Ethernet Packet Monitor Receive Control and Status register.

Table 5. Ethernet Packet Monitor Register Map

<table>
<thead>
<tr>
<th>Address Offset</th>
<th>Name</th>
<th>Number of Bits</th>
<th>Description</th>
<th>Access</th>
<th>Hardware Reset</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>number_packet</td>
<td>32</td>
<td>Number of packets register. The total number of packets to be received by the Ethernet Packet Monitor.</td>
<td>RW</td>
<td>0x0</td>
</tr>
<tr>
<td>0x04</td>
<td>packet_rx_ok</td>
<td>32</td>
<td>Packet received OK register. The current count of packets received without error.</td>
<td>RO</td>
<td>0x0</td>
</tr>
<tr>
<td>0x08</td>
<td>packet_rx_error</td>
<td>32</td>
<td>Packet received with error register. The current count of packets received with an error.</td>
<td>RO</td>
<td>0x0</td>
</tr>
<tr>
<td>0x0C</td>
<td>byte_rx_count0</td>
<td>32</td>
<td>Lower byte received count register. Bits [31:0] of the current count of data bytes received.</td>
<td>RO</td>
<td>0x0</td>
</tr>
<tr>
<td>0x10</td>
<td>byte_rx_count1</td>
<td>32</td>
<td>Upper byte received count register. Bits [64:32] of the current count of data bytes received.</td>
<td>RO</td>
<td>0x0</td>
</tr>
<tr>
<td>0x14</td>
<td>cycle_rx_count0</td>
<td>32</td>
<td>Lower cycle receive count register. Bits [31:0] of the current count of cycles from start of first packet byte received to end of last packet byte received.</td>
<td>RO</td>
<td>0x0</td>
</tr>
<tr>
<td>0x18</td>
<td>cycle_rx_count1</td>
<td>32</td>
<td>Upper cycle receive count register. Bits [64:32] of the current count of cycles from start of first packet byte received to end of last packet byte received.</td>
<td>RO</td>
<td>0x0</td>
</tr>
<tr>
<td>0x1C</td>
<td>receive_ctrl_status</td>
<td>32</td>
<td>Receive control and status register. Refer to Table 6 for register field descriptions.</td>
<td>RW/RO</td>
<td>0x0</td>
</tr>
</tbody>
</table>
### Interface Signal Descriptions

This section describes the top-level interface signals in the 10GbE hardware demonstration reference design.

Table 7 lists the clock and reset signals. Table 8 lists the top-level 10GbE component (eth_ginst) signals.

#### Table 6. Receive Control and Status Register Field Descriptions

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Bit Name</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>start</td>
<td>Start operation. Write 1 to start packet reception by the Ethernet Packet Monitor. This bit is automatically cleared after packet reception begins.</td>
<td>RW</td>
</tr>
<tr>
<td>1</td>
<td>stop</td>
<td>Stop operation. Write 1 to stop the Ethernet Packet Monitor from receiving further packets. The Ethernet Packet Monitor stops updating the status registers immediately after this bit is asserted. This bit is cleared when the Receive start bit is set to 1.</td>
<td>RW</td>
</tr>
<tr>
<td>2</td>
<td>rx_done</td>
<td>Receive-done status. This bit is set to 1 when the Ethernet Packet Monitor has received all of the packets programmed in the Number of Packets register. This bit is cleared when the Receive start bit is set to 1.</td>
<td>RO</td>
</tr>
<tr>
<td>3</td>
<td>crcbad</td>
<td>CRC error packet. This bit sets to 1 when the current packet received is detected as a CRC error packet. This status bit is updated continuously for every packet received.</td>
<td>RO</td>
</tr>
<tr>
<td>4</td>
<td>rx_err</td>
<td>Receive error status. This bit is the mapping of rx_err signals from the 10GbE reference design function (eth_10ginst) receive FIFO interface during every final frame octet. Refer to AN516: 10-Gbps Ethernet Reference Design for the full signal definitions.</td>
<td>RO</td>
</tr>
</tbody>
</table>

Table 7. Clock and Reset Signals

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk</td>
<td>In</td>
<td>Reference design clock. This clock is sourced from the 100-MHz oscillator (X1) on the development board.</td>
</tr>
<tr>
<td>ref_clk</td>
<td>In</td>
<td>10GbE component transceiver reference clock. This clock is sourced from the 156.25-MHz oscillator (X3) on the development board.</td>
</tr>
<tr>
<td>reset_n</td>
<td>In</td>
<td>Single reset for all the logic in the reference design. This signal is connected to the RESET (S4) push-button switch on the development board.</td>
</tr>
</tbody>
</table>

#### Table 8. 10GbE Component Signals

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a_xaui_rx[3:0]</td>
<td>In</td>
<td>Port A - XAUI Differential Receive Interface. This bus is connected to the HSMC board.</td>
</tr>
<tr>
<td>a_xaui_tx[3:0]</td>
<td>Out</td>
<td>Port A - XAUI Differential Transmit Interface. This bus is connected to the HSMC board.</td>
</tr>
</tbody>
</table>

### Resource Utilization

Table 9 shows resource utilization figures for the 10GbE hardware demonstration reference design.
For more detailed resource utilization information, refer to the Quartus II Fitter report (eth_10g_hw_ref_design_top.fit.rpt) file located in the <installation path>/10GbE_Hardware_Demo/hardware/ directory after running compilation.

Table 9. Resource Utilization

<table>
<thead>
<tr>
<th>Device Family</th>
<th>Combinational ALUTS</th>
<th>Logic Registers</th>
<th>Memory</th>
<th>Phase-Locked Loops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arria II GX</td>
<td>10508</td>
<td>10350</td>
<td>315</td>
<td>0</td>
</tr>
<tr>
<td>Stratix IV GX</td>
<td>10,676</td>
<td>12,042</td>
<td>68</td>
<td>16</td>
</tr>
</tbody>
</table>

Notes to Table 9:
(1) ALUTs are adaptive look-up tables.

Using the Reference Design

The following sections describe how to set up and use the 10GbE hardware demonstration reference design:

- Hardware Requirements and Setup
- Reference Design Installation
- Preparing to Run the Application
- Running the Benchmark Application Software

Hardware Requirements and Setup

The reference design application requires the following basic hardware for all loopback options:

- A computer running the Windows XP operating system
- Arria II GX FPGA Development Kit and/or Stratix IV GX FPGA Development Kit
- Altera USB-Blaster™ or ByteBlaster™ download cable

To connect the basic hardware, perform the following steps:

1. Connect the Altera programming cable to the USB connector port (J7 on the Stratix IV GX FPGA development board and J6 on the Arria II GX FPGA development board).

2. Connect the power supply cord to the power supply input (J4).

Table 10 on page 20 summarizes the hardware setup for each loopback option. Subsequent sections provide more detailed information. Only one loopback option should be configured to be active at any given time.
Figure 13 on page 21 shows the basic connections and the positions on the user DIP switch.

Table 10. Hardware Setup Summary

<table>
<thead>
<tr>
<th>Loopback Option Name</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic hardware</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>HSMC-to-CX4 adapter with CX4 loopback testing device</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSMC 10-Gbps X2 with X2 module and loopback optical cable</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSMC loopback board</td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Position 6 (or 3) on user DIP switch</td>
<td>Off</td>
<td>On</td>
<td>On</td>
<td>On</td>
<td>On</td>
<td>(Note 1)</td>
</tr>
<tr>
<td>Position 7 (or 2) on user DIP switch</td>
<td>On</td>
<td>On</td>
<td>On</td>
<td>On</td>
<td>Off</td>
<td></td>
</tr>
<tr>
<td>Position 8 (or 1) on user DIP switch</td>
<td>On</td>
<td>On</td>
<td>Off</td>
<td>On</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>Internal MAC loopback mode on the user application Test menu</td>
<td>Disable</td>
<td>Disable</td>
<td>Disable</td>
<td>Disable</td>
<td>Enable</td>
<td>Disable</td>
</tr>
</tbody>
</table>

Note to Table 10:
(1) This option requires a connection to another 10GbE board. Set to On if a X2 connection is used, or Off if a CX4 connection is used.
On the development board user dual in-line package (DIP) switch, On means logical zero and the opposite position of On (Off in the summary table) means logical one.

The DIP switch may be mounted in reverse on some boards.

Refer to Figure 13 for the positions relevant to your board. The switch positions have the following functions:

- Position 6 (or 3) = CX4 tx_invpolarity enable
- Position 7 (or 2) = internal line loopback
- Position 8 (or 1) = internal PMA loopback
For the Arria II GX FPGA development kit, the following switch positions apply:

- Position 1 = internal PMA loopback
- Position 2 = internal line loopback
- Position 3 = CX4 tx_invpolarity enable


## Configuring the Clock on the Arria II GX FPGA Development Board

The 10GbE reference design requires a 156.25-MHz reference clock. If you are using the Arria II GX FPGA development board, you must configure the clock before configuring any of the loopback options. You configure the reference clock to the required frequency using the programmable oscillator.

To configure the reference clock on the Arria II GX FPGA development board, perform the following steps:

1. Set the following DIP switches and jumper on the board:
   - DIP switch bank (SW2) to match Table 11.
   - DIP switch bank (SW4) to match Table 12.
   - Jumper block (J9) to match

### Table 11. Dip Switch Bank (SW2)

<table>
<thead>
<tr>
<th>Switch</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>On</td>
</tr>
<tr>
<td>2</td>
<td>On</td>
</tr>
<tr>
<td>3</td>
<td>On</td>
</tr>
<tr>
<td>4</td>
<td>On</td>
</tr>
</tbody>
</table>

### Table 12. Dip Switch Bank (SW4)

<table>
<thead>
<tr>
<th>Switch</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Off</td>
</tr>
<tr>
<td>2</td>
<td>Off</td>
</tr>
<tr>
<td>3</td>
<td>Off</td>
</tr>
<tr>
<td>4</td>
<td>Off</td>
</tr>
<tr>
<td>5</td>
<td>Off</td>
</tr>
<tr>
<td>6</td>
<td>Off</td>
</tr>
<tr>
<td>7</td>
<td>On</td>
</tr>
<tr>
<td>8</td>
<td>On</td>
</tr>
</tbody>
</table>
Ensure that the power and USB-Blaster cables are connected to the Arria II GX development board. Upon power-on, the LCD shows "Not Connected".

Run the Arria II GX Board Test System on Windows by clicking **Start > All Programs > Altera > Arria II GX FPGA Development Kit v9.0.2 > Board Test System**. In the dialog box that appears, click **OK** to continue.

For more information about installing and using the Arria II GX FPGA Development Kit, refer to the *Arria II GX FPGA Development Kit User Guide*.

4. On the Configure menu, click **Configure with SRAM/Flash/GPIO Design**.

5. In the dialog box that appears, click **Configure** to download the design’s SRAM Object File (.sof) to the FPGA.

6. When configuration finishes, click **Close** to complete the configuration process. The LCD shows "Arria II GX FPGA Development Kit".

7. Set **OCR1** to **156.25 MHz** (see **Figure 14**).

   The clock frequency on the programmable oscillator U26 is now set to 156.25 MHz.

---

**Table 13. Jumper Block (J9)**

<table>
<thead>
<tr>
<th>Switch</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Not installed</td>
</tr>
<tr>
<td>2</td>
<td>Installed</td>
</tr>
<tr>
<td>3</td>
<td>Installed</td>
</tr>
<tr>
<td>4</td>
<td>Installed</td>
</tr>
</tbody>
</table>
8. Close the Arria II GX board Test System to release the control of the device for the Nios II command shell. Otherwise, the Nios II command shell cannot access the device (see Figure 15 on page 24).

**Figure 14.** Board Test System—Setting Programmable Oscillator

**Figure 15.** BTS Error
Loopback Option A, External CX4 Loopback

The following additional hardware is required for option A:

- One HSMC-to-CX4 adapter board
- One CX4 loopback testing device

Perform the following steps to setup the external CX4 loopback:

1. Connect the HSMC-to-CX4 adapter card to the respective HSMC port A connectors:
   - J1 on the Stratix IV GX FPGA development board
   - J6 on the Arria II GX FPGA development board
2. Connect the CX4 loopback testing device to the HSMC-to-CX4 card.
3. Verify that the following switches are set to the On position:
   - Switches 7 (or 2) and 8 (or 1) on the DIP switch (SW3) on the Stratix IV GX FPGA development board
   - Switches 1 and 2 on the DIP switch (SW2) on the Arria II GX FPGA development board
4. Set the following switches to the Off position:
   - Switch 6 (or 3) on the DIP switch (SW3) on the Stratix IV GX FPGA development board
   - Switch 3 on the DIP switch (SW2) on the Arria II GX development board

This selection enables transmitter polarity inversion in SERDES which is a requirement for the MoreThanIP HSMC-to-CX4 adapter card. For full information about this errata, refer to hsmc_cx4_board_ref10_errata.pdf in the 10GbE_Hardware_Demo/doc directory.

5. Verify that Enable/Disable MAC Loopback Mode is disabled in the benchmark application Test menu. See “Programming the FPGA with the Benchmark Application” on page 32.

6. Connect the Altera programming cable to the USB connector port (J7 on the Stratix IV GX FPGA development board and J6 on the Arria II GX FPGA development board).

7. Connect the power supply cord to the power supply input (J4).

Loopback Option B, External Fiber Optic Cable Loopback

The following additional hardware is required for option B:

- One HSMC-to-X2 adapter board
- One X2 optical module
- One optical loopback cable assembly

Perform the following steps to setup the external fiber optic cable loopback:
1. Connect the HSMC-to-X2 adapter card to the respective HSMC port A connectors:
   - J1 on the Stratix IV GX FPGA development board
   - J6 on the Arria II GX FPGA development board
2. Connect the X2 module to the HSMC-to-X2 adapter card.
3. Connect the optical loopback cable to the X2 module (Figure 16 on page 26).
4. Verify that the following switches are set to the On position:
   - Switches 6 (or 3), 7 (or 2), and 8 (or 1) on the DIP switch (SW3) on the Stratix IV
     GX FPGA development board
   - Switches 1, 2, and 3 on the DIP switch (SW2) on the Arria II GX FPGA
     development board
5. Verify that Enable/Disable MAC Loopback Mode is disabled in the benchmark
   application Test menu. See “Programming the FPGA with the Benchmark
   Application” on page 32.
6. Connect the Altera programming cable to the USB connector port (J7 on the Stratix
   IV GX FPGA development board and J6 on the Arria II GX FPGA development board).
7. Connect the power supply cord to the power supply input (J4).

Figure 16. External Fiber Optic Cable Loopback Connection
Loopback Option C, External HSMC Loopback

The following additional hardware is required for option C:

- One HSMC loopback board

Perform the following steps to setup the external HSMC loopback:

1. Connect the HSMC loopback board to the respective HSMC port A connectors:
   - J1 on the Stratix IV GX FPGA development board
   - J6 on the Arria II GX FPGA development board

2. Verify that the following switches are set to the On position:
   - Switches 6 (or 3), 7 (or 2), and 8 (or 1) on the DIP switch (SW3) on the Stratix IV GX FPGA development board
   - Switches 1, 2, and 3 on the DIP switch (SW2) on the Arria II GX FPGA development board

3. Verify that Enable/Disable MAC Loopback Mode is disabled in the benchmark application Test menu. See “Programming the FPGA with the Benchmark Application” on page 32.

4. Connect the Altera programming cable to the USB connector port (J7 on the Stratix IV GX FPGA development board and J6 on the Arria II GX FPGA development board).

5. Connect the power supply cord to the power supply input (J4).

Loopback Option D, Internal PMA Loopback

No additional hardware is required for option D.

Perform the following steps to setup the internal PMA loopback:

1. Set the following switches to the Off position:
   - Switch 8 (or 1) on the DIP switch (SW3) on the Stratix IV GX FPGA development board
   - Switch 1 on the DIP switch (SW2) on the Arria II GX FPGA development board

2. Verify that the following switches are set to the On position:
   - Switches 6 (or 3) and 7 (or 2) on the DIP switch (SW3) on the Stratix IV GX FPGA development board
   - Switches 2 and 3 on the DIP switch (SW2) on the Arria II GX FPGA development board.

3. Verify that Enable/Disable MAC Loopback Mode is disabled in the benchmark application Test menu. See “Programming the FPGA with the Benchmark Application” on page 32.

4. Connect the Altera programming cable to the USB connector port (J7 on the Stratix IV GX FPGA development board and J6 on the Arria II GX FPGA development board).

5. Connect the power supply cord to the power supply input (J4).
Loopback Option E, Internal MAC Loopback

No additional hardware is required for option E.

Perform the following steps to setup the internal MAC loopback:

1. Verify that the following switches are set to the On position:
   - Switches 6 (or 3), 7 (or 2), and 8 (or 1) on the DIP switch (SW3) on the Stratix IV GX FPGA development board
   - Switches 1, 2, and 3 on the DIP switch (SW2) on the Arria II GX FPGA development board

2. Verify that Enable/Disable MAC Loopback Mode is enabled in the benchmark application Test menu. See “Programming the FPGA with the Benchmark Application” on page 32.

3. Connect the Altera programming cable to the USB connector port (J7 on the Stratix IV GX FPGA development board and J6 on the Arria II GX FPGA development board).

4. Connect the power supply cord to the power supply input (J4).

Loopback Option F, Internal Rx-to-Tx Line Loopback Outside MAC

The line loopback option checks the interoperability of the Altera 10GbE reference design and other external 10 Gbps Ethernet logic. Apart from the connection to other customer logic, no additional hardware is needed in addition to the basic hardware.

Perform the following steps to setup the internal Rx-to-Tx Line loopback outside MAC:

1. Verify that the following switches are set to the On position:
   - Switch 8 (or 1) on the DIP switch (SW3) on the Stratix IV GX FPGA development board
   - Switch 1 on the DIP switch (SW2) on the Arria II GX FPGA development board

2. If a CX4 connection is used, set the following switches to Off position:
   - Switch 6 (or 3) on the DIP switch (SW3) on the Stratix IV GX FPGA development board
   - Switch 3 on the DIP switch (SW2) on the Arria II GX FPGA development board

3. If an X2 connection is used, set the following switches to On position:
   - Switch 6 (or 3) on the DIP switch (SW3) on the Stratix IV GX FPGA development board
   - Switch 3 on the DIP switch (SW2) on the Arria II GX FPGA development board

4. Verify that the following switches are set to the Off position:
   - Switch 7 (or 2) on the DIP switch (SW3) on the Stratix IV GX FPGA development board
   - Switch 2 on the DIP switch (SW2) on the Arria II GX FPGA development board

5. Verify that Enable/Disable MAC Loopback Mode is disabled in the benchmark application Test menu. See “Programming the FPGA with the Benchmark Application” on page 32.
6. Connect the Altera programming cable to the USB connector port (J7 on the Stratix IV GX FPGA development board and J6 on the Arria II GX FPGA development board).

7. Connect the power supply cord to the power supply input (J4).

**Software Requirements**

The reference design application also requires the Altera Quartus® II 9.0 software, including the following features:

- USB-Blaster or ByteBlaster driver
- SOPC Builder
- Nios II Embedded Design Suite (EDS)
- Altera 10GbE reference design

**Reference Design Installation**

This section describes how to install the 10GbE hardware demonstration reference design.

**Downloading the Package**

All of the files necessary for the 10GbE hardware demonstration reference design are included in the `10GbE_Hardware_Demo_SIVGX_1instance.zip` file. This file is available for you to download from the 10-Gbps Ethernet Hardware Demonstration Reference Design web page.

**Extracting the Files**

Unzip the `10GbE_Hardware_Demo_SIVGX_1instance.zip` file in the directory you have designated for this project (for example, `10GbE_Hardware_Demo`). Figure 17 shows the directory structure.

**Figure 17.** Design Directory Structure After Installation
Preparing to Run the Application

Before you can run the application, you must install the required software listed in “Software Requirements” on page 29, and connect the hardware as described in “Hardware Requirements and Setup” on page 19.

Managing Multiple Development Boards

If you choose to run the reference designs simultaneously, connect the boards to the computer and identify each board using its cable ID. To obtain the cable ID, perform the following steps:

1. Ensure that the power and USB-Blaster cables are connected to each board, and the USB-Blaster cables are connected to your PC.
2. Turn on the board power.
3. When prompted, install the driver of the second USB-Blaster download cable.

   For more information on installing the USB-Blaster driver, refer to the USB-Blaster Download Cable User Guide.

4. Open a Nios II command shell.
   
   To start the Nios II command shell, click Start in Windows. On the All Programs submenu, point to Altera and then Nios II EDS <version>, and click Nios II <version> Command Shell.

5. Type jtagconfig. The system displays the cable IDs—USB-Blaster [USB-0] and USB-Blaster [USB-1]—for Altera FPGAs on the boards.

   Figure 18. Jtagconfig Output

6. Use the cable ID (USB-Blaster [USB-0] or USB-Blaster [USB-1]) to program the device and download user applications. For command-line help and tips, type jtagconfig --extrahelp in the Nios II command shell.
Configuring the Board With the Hardware Design

To program the Stratix IV GX or Arria II GX FPGA with the hardware image for the design, perform the following steps:

1. Open a Nios II command shell.

   To start the Nios II command shell, click **Start** in Windows. On the All Programs submenu, point to **Altera** and then **Nios II EDS** <version>, and click **Nios II Command Shell**.

2. Change to the **demo** subdirectory below the directory containing the unzipped design files, at `<installation path>/10GbE_Hardware_Demo/demo`. The `10gDemo.elf` and `eth_10g_hw_ref_design_top.sof` files are in this directory.

   If the `.sof` or `.elf` file is missing or corrupted, you must regenerate the design. For details of the regeneration process, refer to “Regenerating and Recompiling the Reference Design” on page 39.

3. Type the following command:

   ```
nios2-configure-sof "<cable ID>" --device 1 eth_10g_hw_ref_design_top.sof
   ```

   For example, to configure the FPGA on the board which is connected to the first USB-Blaster cable, type the following command:

   ```
nios2-configure-sof "USB-Blaster [USB-0]" --device 1 eth_10g_hw_ref_design_top.sof
   ```

   Omit the cable ID if you are only using one board.

   **Figure 19 on page 31** shows the Nios II command shell after the hardware image is successfully programmed.
Programming the FPGA with the Benchmark Application

The pre-compiled benchmark application is provided with the 10GbE hardware demonstration reference design files.

If your software application file, the *.elf* file, is missing or corrupted, you must regenerate the design. For information about how to regenerate and recompile the hardware and software application, refer to “Regenerating and Recompiling the Reference Design” on page 39. Altera recommends that you regenerate and recompile both the hardware and software application if you require regeneration of the software application files. For information about how to simulate the software application, refer to “Testbench Simulation” on page 41.

To download the pre-compiled benchmark application to system memory and start the Nios II processor, type the following command in the Nios II command shell:

```bash
nios2-download "<cable ID>" -g 10gDemo.elf && nios2-terminal "<cable ID>
```

Omit the cable ID if you are only using one board.

This command downloads the Nios II processor's image file to system memory, and restarts the processor to begin code execution. The command also opens a terminal application, nios2-terminal, which lets you interact with the software application.
If the benchmark application starts successfully, the Test menu appears, as shown in Figure 20.

**Figure 20. Benchmark Application Test Menu**

![Benchmark Application Test Menu](image)

**Running the Benchmark Application Software**

The Nios II embedded processor runs the custom software application that configures and runs benchmark tests for this system. The application provides a custom menu driven user interface, which you can use to parameterize and run tests on the hardware. As shown in “Menu Driven Interface Example” on page 37, you select options in the test menu by typing the menu item number, followed by the setting value, and pressing Enter. For user-specified options that require a numerical value, you may specify the value in hexadecimal format (for example, 0x1000) or in decimal format (for example, 4000).

The benchmark application has two menus, one for configuring and running a test, and one for viewing reports about tests already run.
**Test Menu**

The Test menu allows you to configure and run a benchmark test in the system. Initially, it displays the default values for the test parameters. You can use it to control the following test parameters:

- **Packet Length**—Controls the length of the packets sent over the link. Specify random size, or a specific fixed size. You can specify a fixed packet-length of 64 bytes to 9600 bytes. If you specify random size, the generated packets can range in length from 24 bytes to 1518 bytes.

- **Number of Packets**—Controls the number of packets transmitted in the test. Specify the value as a 32-bit unsigned integer. The value must be between 1 and \((2^{32} - 1)\), inclusive.

- **Packet Data**—Controls the data pattern type in the payload portion of the packet. Specify increment or random. If you specify the incremental data pattern, the data payload, a 32 bit unsigned integer, increments for each subsequent packet. After this integer reaches \((2^{32} - 1)\), it rolls over to 0 on the next increment operation. If you specify the random data type, a new pseudo-random, 32-bit unsigned integer is used for each data payload.

- **Start Test**—Starts a test. You can choose to start both the Packet Monitor and Packet Generator or only one of them. For the Packet Monitor, the test runs until the number of packets received is equal to the number specified in the **Number of Packets** setting or until time out. For the Packet Generator, the test runs until the number of packets transmitted is equal to the number specified in the **Number of Packets** setting, or until you interrupt the test by pressing Enter. After the test stops, the Report menu displays the results. During the operation of a test, throughput approximations appear periodically, printed in packets per second.

- **Generator Test Delay**—When a test starts, this is the delay time for the Packet Generator to actually start transmitting packets. The generator test delay time applies when the generator starts regardless of the monitor setting in the **Start Test** option. This is useful when you perform a two board test such as the one-port example shown in Figure 3 on page 3.

- **Enable/Disable MAC Loopback Mode**—This option controls loopback option D (Internal MAC loopback). Disable MAC loopback mode is the default option.

**Report Menu**

The Report menu displays the status of the last test run on the system, and allows you to scroll through previous test reports, or to switch to another menu. You can store multiple reports in first-in, first-out order. The default number of reports that can be stored is 10, but you can change this number by modifying the value of `GSIZE` in `test_menus/config.h`.

After the maximum number of test reports have been generated, the oldest report data is discarded to make room for new reports. Each report is displayed in the format illustrated in Figure 21 on page 35.
A report provides the following information about the test run:

- **Number**—An unique integer for each test. Starts at 1 and increments with every test started.
- **Status**—Test status is either Done, Interrupted, or Error.
- **Run Time**—Total run time of the test.
- **10G Sender MAC**—Source address in the Ethernet packet. This is the MAC address of the 10GbE reference design instance. This address also defines the source address field in the 10GbE packets generated by the Packet Generator. You can configure the value by selecting **10Gb MAC Menu** under Other Menus.
- **10G Receiver MAC**—Destination address in the Ethernet packet. This address defines the destination address field in the 10GbE packets generated by the Packet Generator. You can configure the value using the 10Gb MAC menu.
- **Link Speed**—The link speed (always 10000) in Mbits/second and the duplex mode (always full).
- **Packets to Send**—Number of packets the test was configured to send, the packet length, the data type selected for the packets, and the seed value used if the data type is random.
- **Packets Sent**—Total number of packets sent by the Ethernet Packet Generator during the test, and the final throughput rate in packets per second. The actual number of packets may be less than the number specified for the test, if an error occurred in the system.

- **Packets Received**—Total number of packets received by the Packet Monitor during the test, number of valid packets, and number of error packets.

  The total number of packets should be equal to the sum of the number of valid packets and the number of error packets.

- **Bytes Received**—Total number of bytes received by the receiver, and the final throughput rate in bits per second. The byte count includes the complete packet lengths (header plus data).

- **Report Control**—Controls the selection of the report. The current test number, first available test number, and final available test number are all displayed. Specify a report to view using the *next* and *previous* controls or by typing a report number.

  The total amount of time to complete the test is calculated from the 64-bit Cycle Receive Count register in the Ethernet Packet Monitor block. This cycle-count register is reset to 0 at the start of the test and begins incrementing at the system clock frequency rate after the first start-of-packet is received. The register stops counting immediately after all packets are received by the monitor block, that is, after the end-of-packet from the final packet is received. You can also stop the register from incrementing by interrupting the test. The total time for the test is computed by dividing the 64-bit value in the Cycle Receive Count register by the reference clock (ref_clk) frequency (156.25 MHz), using floating point operations.

  Although this method of calculating time is very accurate, its accuracy depends on the monitor receiving all the packets during the test. If the monitor does not receive all the packets during the test, or if a major error occurs in the Ethernet Packet Monitor block, inaccurate time values are reported.

### 10Gb MAC Menu

You use the 10Gb MAC menu to configure the 10G sender MAC address and 10G receiver MAC address.

The 10G sender MAC address is the MAC address of the Altera 10GbE reference design instance. This address also defines the source address field in the 10GbE packets generated by the Packet Generator.

The 10G receiver MAC address defines the destination address field in the 10GbE packets generated by the Packet Generator.
Menu Driven Interface Example

The following simple local loopback (Tx-to-Rx) test example shows you how to use the Test menu.

In this example, the system is configured to send 10,000,000 packets with a data length of 1200 bytes each. To configure and run this test, perform the following steps:

1. To configure the packet size to 1,200 bytes, type the following at the Test menu command prompt: `2e 1200`  
2. To configure the number of packets to 10,000,000, type the following at the Test menu command prompt: `3b`  
3. To run the test (and start both the Packet Monitor and the Packet Generator), type the following at the Test menu command prompt: `7a`  

Figure 22 shows the Nios II command shell after you perform steps 1 and 2 and type `6a`, but before you press Enter. The new values for the packet size and number of packets are displayed.

Figure 22. Benchmark Application Test Menu With Selections

The test starts displaying a series of status messages as shown in Figure 23 on page 38.
The messages describe the progress of the test. The length of the arrow represents the percentage of total packets already sent; when the test completes, the arrowhead reaches the right margin. Each arrow is printed with a progressive throughput calculation, providing a rough approximation of the packets per second achieved in the test so far.

After the test completes, the Report Menu is displayed. Figure 21 on page 35 shows the Report Menu for this example.

**Performing Two-Board Testing**

You can run tests on two boards simultaneously and configure the boards to send packets to the other. To set up the boards for testing, perform the following steps:

1. Connect the development boards to a switch using one of the configurations described in the “Hardware Requirements and Setup” section.

2. Obtain the USB-Blaster cable IDs. Refer to the “Managing Multiple Development Boards” on page 30 for the step-by-step procedure.

3. On the 10 Gb MAC Menu page, set the MAC addresses. For each board, set the source MAC address to its own MAC address and the destination address to the MAC address of the other board.
4. On the **Test Menu** page, set the following parameters:

- The packet length and number of packets. You must use the same values for both boards.
- Generator delay for the second board. The delay causes the packet generator running on the second board to pause for the specified duration before it starts sending the first packet, and thus allowing its packet monitor to receive packets sent by the first board.

**Regenerating and Recompiling the Reference Design**

To regenerate the 10GbE hardware demonstration reference design, you must update both the Quartus II project and Nios II software (user interface application).

**Quartus II Project**

 Updating the hardware design includes updates to the following items:

- Backup files
- SOPC Builder system regeneration
- Enable PMA loopback in the serializer/deserializer (SERDES)
- Adding signals in three files manually to reflect the PMA loopback
- Recompile the Quartus II project

To regenerate and recompile the 10GbE hardware demonstration reference design on your computer, perform the following steps:

1. Start the Quartus II software v9.0.
2. Open the Quartus II project for the reference design at `<installation path>/10GbE_Hardware_Demo/hardware/eth_10g_hw_ref_design.qpf`.
3. Setup the reference design library. The hardware reference design uses a Linux version of the reference design. If the hardware reference design needs to be regenerated in the Windows environment, use the following command:

```
unzip <installation path>\hardware\10glib_90sp1\zip\eth_10g_win32.tar.gz
```

Tips: `<10GbE reference design installation path>` and `<installation path>` are not the same directory. `<10GbE reference design installation path>` is the location where you installed the Altera 10GbE reference design source files, and `<installation path>` is the location where you installed the 10GbE hardware demonstration reference design files.

4. Replace the two sub-directories `eth_10g` and `common` under the `<installation path>\hardware\10glib_90sp1` directory with the two corresponding un-zipped directories.
5. Open the SOPC Builder system as follows:
   a. On the Tools menu, click SOPC Builder. SOPC Builder displays the 10GbE hardware demonstration reference design system.
   b. Keep SOPC Builder open while you backup the following files in the <installation path>\hardware directory by changing the *v file name to *.v_bak:
      
      ```
      mv xaui_10g.v xaui_10g.v_bak
      mv eth_10ginst.v eth_10ginst.v_bak
      mv eth_10g_hw_ref_design.v eth_10g_hw_ref_design.v
      ```
   c. Regenerate the SOPC Builder system by clicking on Generate in SOPC Builder.
   d. Click Exit to close SOPC Builder after regeneration is complete.

6. Enable serial loopback in SERDES and enable transmitter polarity inversion. After SOPC Builder generation, SERDES resets to the default settings (no loopback and no Transmit polarity inversion). Enable serial loopback (PMA loopback) as follows:
   a. On the Tools menu click MegaWizard Plug-In Manager, select Edit an existing custom megafunction variation, and click Next.
   b. Select xaui_10_serdes4_reconfig.v and click Next. When the ALTGX MegaWizard interface is displayed, click on the Lpbk (loopback) tab and select Serial loopback. Click on the XAUI tab, select create tx_invpolarity, and click Finish.
   
   The signals rx_seriallpbken (loopback enable) and tx_invpolarity (Tx polarity inversion enable) are added as inputs of xaui_10_serdes4_reconfig.v.
   c. Click Finish to close the MegaWizard interface.

7. Manually connect rx_seriallpbken and tx_invpolarity to the top level entity as follows:
   a. Search for rx_seriallpbken and tx_invpolarity in the three back up files (xaui_10g.v_bak, eth_10ginst.v_bak and eth_10ginst.v_bak).
   b. Using these backup files as reference, add the rx_seriallpbken signal in the three new generated files xaui_10g.v, eth_10ginst.v, and eth_10ginst.v.

8. Verify that the project library is correct and recompile the design by clicking Start Compilation on the Processing menu.

**User Interface Application**

To update the user interface application, perform the following steps:

1. Open a Nios II command shell.

   To start the Nios II command shell, click Start in Windows. On the All Programs submenu, point to Altera and then Nios II EDS <version>, and click Nios II <version> Command Shell.

2. Create a new directory in which to store the .sof and .elf files for the regenerated SOPC Builder system, by typing the following command sequence in the Nios II command shell:
cd <installation_path>/10GbE_Hardware_demo
mkdir tmp

3. Copy the regenerated .sof file to the new directory, by typing the following command:
   cp hardware/eth_10g_hw_ref_design_top.sof tmp/.

4. Change directory to <installation path>/10GbE_Hardware_Demo/software/bsp/10gBSP.

5. Maintain the create-this-bsp file in this directory, and remove all other files and subdirectories.

6. Rebuild the library files for the software application, by typing the following command:
   ./create-this-bsp

7. Change directory to <installation path>/10GbE_Hardware_Demo/software/app/10gDemo.

8. Remove the unnecessary files in this directory by typing the following command:
   make clean_all

9. Rebuild the software application by typing the following command:
   make

10. Copy the .elf file to your new directory by typing the following command:
    cp 10gDemo.elf <installation_path>/10GbE_Hardware_Demo/tmp/.

11. Change directories to <installation path>/10GbE_Hardware_Demo/tmp.

12. To configure the board with the regenerated hardware design, follow the instructions in step 3 in “Configuring the Board With the Hardware Design” on page 31.

13. To download the regenerated software application, follow the instructions in “Programming the FPGA with the Benchmark Application” on page 32.

   The application test bench appears in the nios2-terminal application, as shown in Figure 20 on page 33.

**Testbench Simulation**

You can simulate the 10GbE hardware demonstration reference design using the IP functional simulation model and testbench provided with the reference design files. The testbench files are located in the testbench subdirectory, at <installation path>/10GbE_Hardware_Demo/testbench. The testbench files include a script to run the ModelSim simulator.

To run a simulation using the ModelSim simulator, perform the following steps:

1. Start the ModelSim simulator.

2. Change to the <installation path>/10GbE_Hardware_Demo/testbench/ directory.

3. To compile the IP functional simulation model and run this model with the testbench, type the following command:
   do run_eth_10g_ref_design_tb.tcl
In the ModelSim main window, the ModelSim transcript pane displays messages from the testbench.

**Document Revision History**

Table 14 shows the revision history for this application note.

**Table 14. Document Revision History**

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>December 2009</td>
<td>Updated to include the Arria II GX reference design.</td>
</tr>
<tr>
<td>1.0</td>
<td>July 2009</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>