This application note describes a reference design that demonstrates the interoperability of the Altera® 10-Gbps Ethernet (10GbE) Media Access Controller (MAC) and XAUI PHY IP cores with a Dual XAUI to small form factor pluggable plus (SFP+) high-speed mezzanine card (HSMC) board.

This application note contains the following sections:

- “Features” on page 1
- “System Architecture” on page 2
- “10GBASE-X Ethernet Subsystem Components” on page 3
- “Hardware and Software Requirements” on page 6
- “Using the Reference Design” on page 7
- “System Register Map” on page 13

For more information about the 10GbE MAC and XAUI PHY IP cores, refer to the 10-Gbps Ethernet MAC MegaCore Function User Guide and the Altera Transceiver PHY IP Core User Guide.

Features

This reference design offers the following features:

- System loopbacks at various points in the data path that control, test, and monitor the 10GbE operations.
- Loopback points that include XGMII and serial physical medium attachment (PMA) interface in the Stratix IV GX FPGA development board, and PMA interface in the Broadcom PHY BCM8727 chip on the Dual XAUI to SFP+ HSMC board.
- External optical loopback test at the HSMC board SFP+ modules.
- Sequential random bursts tests. You can configure the number of packets, payload-data type, and payload size for each burst. A pseudo-random binary sequence (PRBS) generator generates the payload data type in fixed incremental values or in a random sequence.
- Packet statistics for a PRBS generator, monitor, MAC transmitter (TX), and MAC receiver (RX).
- Packet classification of different frame sizes transmitted and received by the MAC.
- Throughput for the traffic received by the traffic monitor.
- Tcl-based System Console that allows you to dynamically configure and monitor any registers in this reference design.

**System Architecture**

Figure 1 shows an overview of the system architecture.

**Figure 1. System Architecture Overview**

This reference design consists of the following components:

- 10GBASE-X Ethernet
- Dual XAUI to SFP+ HSMC board
- PC and System Console
10GBASE-X Ethernet

The 10GBASE-X Ethernet subsystem consists of the Management Data Input/Output (MDIO) IP core, a traffic controller, and the 10GbE MAC and XAUI PHY IP cores.

For more information about the components in the 10GBASE-X Ethernet subsystem, refer to “10GBASE-X Ethernet Subsystem Components”.

Dual XAUI to SFP+ HSMC Board

The Dual XAUI to SFP+ HSMC board has two independent XAUI interfaces and two full duplex SFP+ channels. The HSMC board supports dual-channel PHY through the Broadcom PHY BCM8727 chip. However, this reference design only demonstrates the interoperability on one PHY channel. The HSMC boards connects to the 10GBASE-X Ethernet subsystem through a HSMC connector and communicates using the XAUI PHY protocol.

PC and System Console

The PC running on a Linux or a Windows operating system configures and controls the demonstration with the Tcl-based System Console.

For more information about the System Console, refer to the Analyzing and Debugging Designs with the System Console chapter in volume 3 of the Quartus II Handbook.

10GBASE-X Ethernet Subsystem Components

The following sections discuss the main components in the 10GBASE-X Ethernet subsystem.

MDIO

The MDIO IP core enables you to control the Broadcom PHY BCM8727 chip on the HSMC board. You can access the external PHY registers through a pair of indirect registers to specify read or write operation, register address, port address, and device address.
Traffic Controller

The traffic controller consists of a traffic generator and a traffic monitor. The traffic generator injects client packet bursts into the MAC TX and the traffic monitor receives packet bursts from the MAC RX. The traffic controller connects to the Avalon-ST single-clock FIFO in the 10GBASE-X Ethernet subsystem through an Avalon-ST interface.

You can configure the traffic controller to specify the following:

- Number of packets to be generated by the traffic generator
- Types of payload—either fixed incremental or random data type
- Length of payload—either fixed or random up-to-the-configured length
- Number of packets to be received by the traffic monitor

10GbE MAC and XAUI PHY

The 10GbE MAC and XAUI PHY IP cores have an Ethernet loopback module on the MAC-PHY interface and an Avalon-ST single-clock FIFO buffer on the client-MAC interface. The XAUI PHY IP core implements physical coding sublayer (PCS) and PMA in hard logic.

The following sections discuss about the components in the 10GbE MAC and XAUI PHY IP cores. For more information about other components, refer to the 10-Gbps Ethernet MAC MegaCore Function User Guide.

Figure 2 shows the components in the 10GbE MAC and XAUI PHY IP cores.

Figure 2. Components in the 10GbE MAC and XAUI PHY

10GbE MAC

The 10GbE MAC IP core handles the flow of data through the XAUI PHY IP core. On the transmit path, the MAC accepts client frames and constructs Ethernet frames before forwarding them to the PHY. Similarly, on the receive path, the MAC accepts Ethernet frames via a PHY, performs checks, and removes the relevant fields before forwarding the frames to the client. When you generate the 10GbE MAC IP core with the default settings, the MAC includes memory-based statistics counters.
**XAUI PHY**

The XAUI PHY IP core is set to Hard XAUI by default. You must set Transmitter VOD control setting to 1, Pre-emphasis first post-tap setting to 3, and Receiver static equalizer setting to 0 for this reference design.

For more information about the XAUI PHY IP core, refer to the XAUI PHY IP Core chapter of the *Altera Transceiver PHY IP Core User Guide*.

**Ethernet Loopback**

The Ethernet loopback module verifies the functionality of the 10GbE MAC and XAUI PHY IP cores. You can enable the loopback module through the Avalon-MM interface. The loopback module is disabled by default.

**Avalon-ST Single-Clock FIFO**

The Avalon-ST single-clock FIFO buffer receives and transmits data between the MAC and the client. The buffer is 64 bits wide and 512 bits deep. The buffer operates in store-and-forward mode by default. You can configure the buffer to enable the drop-on-error feature. When you enable the drop-on-error feature, the buffer drops the received packets when an error occurs.

For more information about the Avalon-ST single-clock FIFO buffer, refer to the Avalon-ST Single-Clock and Dual-Clock FIFO Cores chapter of the *Embedded Peripherals IP User Guide*.

**Avalon-MM Pipeline Bridge**

The Avalon-MM pipeline bridge connects the external Avalon-MM master, such as the system controller, to the internal Avalon-MM fabric. It retimes the Avalon-MM signals in both directions.

For more information about the Avalon-MM pipeline bridge, refer to the Avalon Memory-Mapped Bridges chapter in volume 4 of the *Quartus II Handbook*. 
Hardware and Software Requirements

The following sections describe the hardware and software requirements for the reference design.

**Hardware Requirements**

The reference design requires the following hardware:

- Stratix IV GX FPGA development board
- Dual XAUI to SFP+ HSMC board
- USB-Blaster cable
- Windows- or Linux-based system console

**Software Requirements**

The reference design also requires the following features of the Quartus II software version 11.0:

- USB-Blaster or ByteBlaster driver
- Qsys system
- Nios II Embedded Design Suite (EDS)
- ModelSim Simulator
Using the Reference Design

The following sections describe the required hardware and software setup.

**Setting up the Stratix IV GX FPGA Development Board**

Figure 3 shows the Stratix IV GX FPGA development board. The development board has a stop button for testing operations, and reset buttons for the 10GBASE-X Ethernet subsystem and the HSMC board. The Stratix IV GX FPGA development board does not require a specific dual in-line package (DIP) switch setting.

**Figure 3. Stratix IV GX FPGA Development Board**
The top row of green LEDs indicates the test status during and after the test. Table 1 lists the LED indicators.

<table>
<thead>
<tr>
<th>LEDs</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blinking Green</td>
<td>Test in Progress</td>
</tr>
<tr>
<td>Solid Green</td>
<td>Pass</td>
</tr>
<tr>
<td>Turned Off</td>
<td>Fail</td>
</tr>
</tbody>
</table>

For more information about the Stratix IV GX FPGA development board, refer to the Stratix IV GX FPGA Development Kits page.
Setting up the Dual XAUI to SFP+ HSMC Board

Figure 4 shows the Dual XAUI to SFP+ HSMC board. You must install SMA connectors on J13 and J14. The HSMC board connects to the 10GBASE-X Ethernet subsystem through the XAUI PHY and provides an SFP+ connector for connection to any external device. A HSMC connector connects the MDIO bus to the Broadcom PHY BCM8727 chip on the HSMC board.

You must plug the HSMC board into the Stratix IV GX FPGA development board and install an SFP+ module with a loopback cable in the upper SFP+ slot (CH2). The HSMC board does not require a separate power supply because the board draws power from the Stratix IV GX FPGA development board.
Setting up the System Console to Program and Test the Device

This reference design provides various Tcl commands to test the Stratix IV GX FPGA development board and the Dual XAUI to SFP+ HSMC board in various loopback modes. Table 2 lists the commands in demo.tcl:

### Table 2. Commands to Test the Reference Design

<table>
<thead>
<tr>
<th>Name</th>
<th>Command</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST</td>
<td>LPBK_POINT</td>
<td>SFPP Loopback at SFP+ cable (external)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCMPMA Loopback at BCM8727 PMA (internal)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCMXGS Loopback at BCM8727 XGXS (internal)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALTPMA Loopback at Altera serial PMA (internal)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALTXGMII Loopback at Altera XGMII (internal)</td>
<td></td>
</tr>
<tr>
<td>BURST_SIZE</td>
<td></td>
<td>Any integer Number of packets in the burst. For a multiple burst test, this value increases the size of each successive burst.</td>
<td></td>
</tr>
<tr>
<td>NUM_BURSTS</td>
<td></td>
<td>A number greater than 0 Specifies the intended number of bursts to send through the demonstration board.</td>
<td></td>
</tr>
</tbody>
</table>

To install and set up the files to program and test the Stratix IV GX FPGA development board, follow these steps:

1. Connect the Stratix IV GX FPGA development board with a USB blaster cable.
2. Download and unzip the reference design files.
   - If your PC runs on a Windows operating system, download the 10GMGC_XAUI_HSMC_ACDS11_WIN32.zip file.
   - If your PC runs on a Linux operating system, download the 10GMGC_XAUI_HSMC_ACDS11.tgz file.

You can download the design files for AN638 from the Literature: Application Notes page on the Altera website.

3. Open a Nios II command shell and go to the 10GMGC_XAUI_HSMC_ACDS11 directory.
4. In the command shell, type the following command:
   ```bash
   nios2-configure-sof CHIP.sof
   ```
5. After programming the device, push the reset buttons for the 10GBASE-X Ethernet subsystem and the Dual XAUI to SFP+ HSMC board (Figure 4).
6. Unplug the cable in the CH2 slot on the HSMC board and replug the cable back in. When you replug the cable, ensure that the cable is secured tightly. You should hear a “click” sound when the SFP+ connector locks in the slot.
7. Browse to the DEMO directory.
8. Type the following command to launch the System Console:
   ```bash
   system-console
   ```
9. Open the Quartus II Tcl Console window by pointing to **Utility Windows** on the View menu, and then click **Tcl Console**. In the Tcl Console window, type the following command:

```
source demo.tcl
```

10. Ensure that you are in the **DEMO** directory. Run the various loopback tests using the commands listed in Table 2 on page 10.

Table 3 shows the test examples.

### Table 3. Test Examples

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFPP 20000 5</td>
<td>Runs a test by sending five bursts of 20000, 40000, 60000, and 80000 packets of random sizes (up to 1518 bytes) and random payload contents through the Altera 10GbE MAC and loops them back from the BCM8727 serial PMA. The Altera 10GbE MAC forwards these bursts to the traffic monitor. The received packet#0 in the Tcl Console window indicates that the fiber is possibly loose. Unplug and then replug the fiber back in.</td>
</tr>
<tr>
<td>BCMPMA 15000 5</td>
<td>Runs a test by sending five bursts of 15000, 30000, 45000, and 60000 packets of random sizes (up to 1518 bytes) and random payload contents through the Altera 10GbE MAC and loops them back from the BCM8727 serial PMA. The Altera 10GbE MAC forwards these bursts to the traffic monitor.</td>
</tr>
<tr>
<td>ALTPMA 15000 5</td>
<td>Runs a test by sending five bursts of 15000, 30000, 45000, and 60000 packets of random sizes (up to 1518 bytes) and random payload contents through the Altera 10GbE MAC and loops them back to the XAUI serial PMA. The Altera 10GbE MAC forwards these bursts to the traffic monitor.</td>
</tr>
</tbody>
</table>

Each test generates a `.log` file. View the `.log` file to ensure that the traffic monitor does not receive bad packets. The `.log` file also provides packet classification and statistics by the MAC RX.
Regenerating and Recompiling the Reference Design

You can reproduce the reference design at various levels of the development flow.

When you recompile and regenerate the reference design files, the Quartus II software generates a SRAM Object File (.sof).

The following sections provide guidelines for regeneration and simulation of the 10GBASE-X Ethernet subsystem.

Generating the 10GBASE-X Ethernet Subsystem

To generate the 10GBASE-X Ethernet subsystem, follow these steps:

You must complete steps 1 to 11 before you simulate the 10GBASE-X Ethernet subsystem.

1. Start the Quartus II software and open the CHIP.qpf from the 10GMGC_XAUI_HSMC_ACDS11 directory.
2. On the Tools menu, click Qsys and open the ETH10G.qsys file in the ETH10G_TOP directory.
3. Double click ETH10G_TOP on the System Contents tab to open the parameter editor.
4. On the CONFIGURATION tab, select MDIO, XAUI PHY and Avalon-ST Single Clock FIFO.
5. On the MAC tab, select Supplementary Address, CRC on Transmit Path and Statistic Collection under the Resource Optimization Options. Set the Statistic Counters to Memory-based.
6. On the MDIO tab, set the MDC DIVISOR value to 64.
7. For both TX Single Clock FIFO and RX Single Clock FIFO on the SC FIFO tab, select USE STORE AND FORWARD and set the FIFO depth to 512.
8. On XAUI tab, set the following parameters to the respective values:
   - Transmitter VOD control setting—1
   - Pre-emphasis first post-tap setting—3
   - Receiver static equalizer setting—0
9. Click Finish to close the parameter editor.
10. On the Generation tab in Qsys, select Verilog simulation model and make sure that the Create HDL design files for synthesis option is turned on.
11. Click Generate to generate ETH10G.v along with other files necessary for the simulation and hardware compilation of the design.
Running Simulation on the 10GBASE-X Ethernet Subsystem

To run simulation on the 10GBASE-X Ethernet subsystem, follow these steps:

1. Start the ModelSim simulator software.
2. Go to the 10GMGC_XAUI_HSMC_ACDS11/SIM directory.
3. In the Tcl Console window, type the following commands:
   
   do compile.tcl
   do runsim.tcl

4. At the end of the simulation, the ModelSim simulator provides a summary of packets received with CRC error. CRC errors appear when the simulation fails.

Recompiling the Design

To recompile the design, follow these steps:

1. Start the Quartus II software and open CHIP.qpf.
2. On the Processing menu, click Start Compilation. The Quartus II software generates a .sof file after the compilation.
4. In the Task window, click Update Timing Netlist. The timing report should not show any design path in violation; however, you can ignore any timing path violations from SignalTap II Logic Analyzer nets.

System Register Map

Table 4 lists the base addresses for various components in the subsystems.

<table>
<thead>
<tr>
<th>Subsystem/Component</th>
<th>Base Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>10GBASE-X Ethernet</td>
<td>0x00000000</td>
</tr>
<tr>
<td>10GbE MAC</td>
<td>0x00000000</td>
</tr>
<tr>
<td>XAUI PHY</td>
<td>0x00040000</td>
</tr>
<tr>
<td>Avalon-ST Single-Clock FIFO</td>
<td>0x00010400</td>
</tr>
<tr>
<td>Ethernet Loopback</td>
<td>0x00010200</td>
</tr>
<tr>
<td>Controller (Traffic/Ethernet MDIO)</td>
<td>0x01000000</td>
</tr>
<tr>
<td>Generator</td>
<td>0x01000000</td>
</tr>
<tr>
<td>Monitor</td>
<td>0x01040000</td>
</tr>
<tr>
<td>Ethernet MDIO</td>
<td>0x00010000</td>
</tr>
</tbody>
</table>
Table 5 and Table 6 list the register map for the traffic generator and traffic monitor in the traffic controller.

Table 5. Generator Register Map

<table>
<thead>
<tr>
<th>Byte Offset</th>
<th>Name</th>
<th>Width</th>
<th>Description</th>
<th>R/W</th>
<th>Reset Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>NUMPKTS</td>
<td>32</td>
<td>Number of packet registers. The total number of packets that the traffic generator generates and transmits to the 10GBASE-X Ethernet subsystem components.</td>
<td>RW</td>
<td>0x0</td>
</tr>
<tr>
<td>0x04</td>
<td>RANDOMLENGTH</td>
<td>1</td>
<td>Enables random length packets up to the maximum size defined by the PKTLENGTH register.</td>
<td>RW</td>
<td>0x0</td>
</tr>
<tr>
<td>0x08</td>
<td>RANDOMPAYLOAD</td>
<td>1</td>
<td>Enables random payload contents.</td>
<td>RW</td>
<td>0x0</td>
</tr>
<tr>
<td>0x0C</td>
<td>START</td>
<td>1</td>
<td>Write to this register to start the generation of the ethernet traffic.</td>
<td>R/W</td>
<td>0x0</td>
</tr>
<tr>
<td>0x10</td>
<td>STOP</td>
<td>1</td>
<td>Stops the generation of the ethernet traffic.</td>
<td>R/W</td>
<td>0x0</td>
</tr>
<tr>
<td>0x14</td>
<td>MACSA0</td>
<td>32</td>
<td>Lower 32-bits of the ethernet frame source address.</td>
<td>RW</td>
<td>0x0</td>
</tr>
<tr>
<td>0x18</td>
<td>MACSA1</td>
<td>16</td>
<td>Upper 16-bits of the ethernet frame source address.</td>
<td>RW</td>
<td>0x0</td>
</tr>
<tr>
<td>0x1C</td>
<td>MACDA0</td>
<td>32</td>
<td>Lower 32-bits of the ethernet frame destination address.</td>
<td>RW</td>
<td>0x0</td>
</tr>
<tr>
<td>0x20</td>
<td>MACDA1</td>
<td>16</td>
<td>Upper 16-bits of the ethernet frame destination address.</td>
<td>RW</td>
<td>0x0</td>
</tr>
<tr>
<td>0x24</td>
<td>TXPKTCNT</td>
<td>32</td>
<td>The number of packets that the traffic generator transmits. Read this register when the traffic generator is not active (for example, after testing).</td>
<td>RO</td>
<td>0x0</td>
</tr>
<tr>
<td>0x34</td>
<td>PKTLENGTH</td>
<td>—</td>
<td>The maximum length of any payload when random-sized packets are enabled. Otherwise, this register defines the packet length generated by the traffic generator.</td>
<td>R/W</td>
<td>0x0</td>
</tr>
</tbody>
</table>
Table 6. Traffic Monitor Register Map

<table>
<thead>
<tr>
<th>Byte Offset</th>
<th>Name</th>
<th>Width</th>
<th>Description</th>
<th>R/W</th>
<th>Reset Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>RXPKTCNT_EXPT</td>
<td>32</td>
<td>Number of packets that the traffic monitor expects.</td>
<td>RW</td>
<td>0xffffffff</td>
</tr>
<tr>
<td>0x04</td>
<td>RXPKTCNT_GOOD</td>
<td>32</td>
<td>Number of good packets received by the traffic monitor.</td>
<td>RO</td>
<td>0x0</td>
</tr>
<tr>
<td>0x08</td>
<td>RXPKTCNT_BAD</td>
<td>32</td>
<td>Number of packets received with CRC error.</td>
<td>RO</td>
<td>0x0</td>
</tr>
<tr>
<td>0x0C</td>
<td>RXBYTECNT_LO32</td>
<td>32</td>
<td>Lower 32-bits of the counter for bytes that the traffic monitor receives.</td>
<td>RO</td>
<td>0x0</td>
</tr>
<tr>
<td>0x10</td>
<td>RXBYTECNT_HI32</td>
<td>32</td>
<td>Upper 32-bits of the counter for bytes that the traffic monitor receives.</td>
<td>RO</td>
<td>0x0</td>
</tr>
<tr>
<td>0x14</td>
<td>RXCYCLCNT_LO32</td>
<td>32</td>
<td>Lower 32-bits of the counter for cycles that the traffic monitor uses to receive the expected number of packets.</td>
<td>RO</td>
<td>0x0</td>
</tr>
<tr>
<td>0x18</td>
<td>RXCYCLCNT_HI32</td>
<td>32</td>
<td>Upper 32-bits of the counter for cycles that the traffic monitor uses to receive the expected number of packets.</td>
<td>RO</td>
<td>0x0</td>
</tr>
<tr>
<td>0x1C</td>
<td>RXCTRL_STATUS</td>
<td>10</td>
<td>Monitor configuration and status register. Bit[0]: initialize all counters when 1'b1 Bit[1]: reserved Bit[2]: Read-only – set when the traffic monitor received all expected packets.</td>
<td>RW/RO</td>
<td>0x0</td>
</tr>
</tbody>
</table>

Document Revision History

Table 7 shows the revision history for this document.

Table 7. Document Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 2011</td>
<td>1.0</td>
<td>Initial release.</td>
</tr>
<tr>
<td>June 2011</td>
<td>1.1</td>
<td>Updated for Qsys version 11.0</td>
</tr>
</tbody>
</table>