AN794: Arria 10 Low Latency Ethernet 10G MAC and XAUI PHY Reference Design

AN-794
2017.02.01
Contents

1 Arria 10 Low Latency Ethernet 10G MAC and XAUI PHY Reference Design ............... 3
  1.1 Design Components .................................................................................................................. 6
  1.2 Design Registers .................................................................................................................... 6
  1.3 Clock and Reset Signals .......................................................................................................... 8
  1.4 Hardware and Software Requirements ....................................................................................... 9
  1.5 Board Setups ............................................................................................................................ 9
  1.6 Testing the Design .................................................................................................................. 11
    1.6.1 Test Commands ................................................................................................................ 11
  1.7 Simulating the Design .............................................................................................................. 12
  1.8 Reference Design Debug Features ........................................................................................... 14
  1.9 Revision History ....................................................................................................................... 15
Arria 10 Low Latency Ethernet 10G MAC and XAUI PHY Reference Design

This reference design demonstrates the Low Latency Ethernet 10G IP solution for Arria® 10 devices.

This design uses Intel's Low Latency Ethernet 10G Media Access Controller (MAC) and XAUI PHY IP cores with a dual XAUI small form factor pluggable plus (SFP+) high-speed mezzanine card (HSMC) board and FPGA mezzanine card (FMC) to high-speed mezzanine card (HSMC) adapter board on Arria 10 FPGA development kit.

The design provides flexible test and demonstration platforms on which you can control, test, and monitor the Ethernet operations using system loopback at various points.

This design offers the following features:

• Loopback points that include XGMII and serial physical medium attachment (PMA) interface in the Arria 10 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 HSMC board SFP+ modules.

• Sequential random bursts tests. You can configure the number of packets, payload-data type, and payload size for each burst.

• Packet statistics for traffic generator, monitor, MAC transmitter (TX) and MAC receiver (RX).

• Packet classification for different frame sizes transmitted and received by the MAC.

• Throughput for the traffic received by the traffic monitor.

• System Console user interface. This TCL-based user interface allows you to dynamically configure and monitor any registers in the reference design.
Figure 1. **System Architecture Overview**

This figure shows the high-level overview of the design's system architecture.

- **Arria 10 FPGA Development Kit**
- **Arria 10 GX FPGA**
- **10G BASE-X Ethernet Subsystem**
- **FMC to HSMC Adapter Board**
- **Dual XAUI to SFP+ HSMC Board**
- **BCMB727**
- **SFP+**
- **SFP+**
- **JTAG**
- **PC and System Console Subsystem**
Figure 2. Ethernet Subsystem Clocking Scheme
This figure shows the clocking scheme for the Ethernet subsystem.

Figure 3. Design Top Level Reset Scheme
This figure shows the reset scheme. Use the master reset signal (csr_rst_n) to reset the MAC, PHY, MDIO, address decoder, reset synchronizer, and synchronizer. This signal connects to hard reset button.

Related Links
Arria 10 Low Latency Ethernet 10G MAC and XAUI PHY Reference Design
Provides the files for the reference design. Click on the tabs for the appropriate version.
1.1 Design Components

The reference design consists of three main subsystems: 10GBASE-X Ethernet, Dual XAUI to SFP+ HSMC board, and PC and System Console.

Table 1. Subsystem Design Components
This table lists the components of the design’s subsystems.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
</table>
| Low Latency Ethernet 10G MAC IP Core | This IP core handles the flow of data through the XAUI PHY IP core.  
  • On transmit path, the MAC accepts client frames and constructs Ethernet frames before forwarding them to the PHY layer.  
  • Similarly on the receive path, the MAC accepts Ethernet frames through the PHY layer, performs checks and removes the relevant fields before forwarding the frames to the client.  
  For this design, the MAC uses the memory-based statistics counters. |
| XAUI PHY IP Core | The XAUI PHY IP core sets to soft XAUI by default. |
| Traffic Controller | The traffic controller consists of:  
  • Traffic generator—injects client packet bursts into the MAC TX core.  
  • Traffic monitor—receives packet bursts from the MAC RX core.  
  The traffic controller connects to the Avalon-ST single-clock FIFO in the Ethernet subsystem through the Avalon Streaming (Avalon-ST) interface. |
| Ethernet Packet Generator | This module consists of Avalon Memory-Mapped (Avalon-MM) registers, Ethernet packet generation block, CRC generator, and shift register. |
| Ethernet Packet Monitor | This module verifies the payload of received packets and collects information from the statistics counters. This consists of Avalon-MM registers and CRC checkers. |
| MDIO IP Core | This IP core enables you to control the Broadcom PHY BCM8727 chip on the Dual XAUI to SFP+ 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. |
| JTAG to Avalon Master Bridge | This IP core provides a connection between the System Console and Qsys system through the physical interfaces. The System Console initiates Avalon-MM transactions by sending encoded streams of bytes through the bridge’s physical interfaces. |
| Reset Controller | This module synchronizes and generates signals as per design requirements. |
| 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. |
| Avalon-ST Adapter | This adapter converts the 32-bit Avalon-ST interface to 64 bits and vice versa. |
| PLL | • Arria 10 ATX PLL takes a 156.25 MHz input clock from the on-board oscillator as the clock source for the XAUI PHY.  
  • Arria 10 fPLL takes a 100 MHz input clock from the on-board oscillator and acts as the clock source for the other components in this design. |

1.2 Design Registers

You can use the registers to edit the design.

Table 2. System Register Map
This table lists all the base addresses for components on the subsystems.
### Table 3. Generator Register Map

This table lists the generator registers for the traffic controller.

<table>
<thead>
<tr>
<th>Byte Offset</th>
<th>Name</th>
<th>Width</th>
<th>R/W</th>
<th>Reset Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>NUMPKTS</td>
<td>32</td>
<td>RW</td>
<td>0x0</td>
<td>Number of packet registers. The total number of packets that the traffic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>generator generates and transmits to the 10GBASE-X Ethernet subsystem</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>components.</td>
</tr>
<tr>
<td>0x04</td>
<td>RANDOMLENGTH</td>
<td>1</td>
<td>RW</td>
<td>0x0</td>
<td>Enables random length packets up to the maximum size defined by the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PKTLENGTH register.</td>
</tr>
<tr>
<td>0x08</td>
<td>RANDOMPAYLOAD</td>
<td>1</td>
<td>RW</td>
<td>0x0</td>
<td>Enables random payload contents.</td>
</tr>
<tr>
<td>0x0C</td>
<td>START</td>
<td>1</td>
<td>R/W</td>
<td>0x0</td>
<td>Write to this register to start the generation of the Ethernet traffic.</td>
</tr>
<tr>
<td>0x10</td>
<td>STOP</td>
<td>1</td>
<td>R/W</td>
<td>0x0</td>
<td>Stops the generation of the Ethernet traffic.</td>
</tr>
<tr>
<td>0x14</td>
<td>MACSA0</td>
<td>32</td>
<td>RW</td>
<td>0x0</td>
<td>Lower 32 bits of the Ethernet frame source address.</td>
</tr>
<tr>
<td>0x18</td>
<td>MACSA1</td>
<td>16</td>
<td>RW</td>
<td>0x0</td>
<td>Upper 16 bits of the Ethernet frame source address.</td>
</tr>
<tr>
<td>0x1C</td>
<td>MACDA0</td>
<td>32</td>
<td>RW</td>
<td>0x0</td>
<td>Lower 32 bits of the Ethernet frame destination address.</td>
</tr>
<tr>
<td>0x2P</td>
<td>MACDA1</td>
<td>16</td>
<td>RW</td>
<td>0x0</td>
<td>Upper 16 bits of the Ethernet frame destination address.</td>
</tr>
<tr>
<td>0x24</td>
<td>TXPKTCNT</td>
<td>32</td>
<td>RO</td>
<td>0x0</td>
<td>The number of packets that the traffic generator transmits. Read this</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>register when the traffic generator is not active (e.g. after testing).</td>
</tr>
<tr>
<td>0x34</td>
<td>PKTLENGTH</td>
<td>—</td>
<td>R/W</td>
<td>0x0</td>
<td>The maximum length of any payload when random-sized packets are enabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Otherwise, this register defines the packet length generated by the traffic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>generator.</td>
</tr>
</tbody>
</table>

### Table 4. Monitor Register Map

This table lists the monitor registers for the traffic controller.
1.3 Clock and Reset Signals

The design uses different clock and reset signals for different components.

**Table 5. Clock and Reset Signals**

<table>
<thead>
<tr>
<th>Signal</th>
<th>Direction</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>refclk1_p</td>
<td>Input</td>
<td>1</td>
<td>100 MHz clock source used as IOPLL reference clock and Avalon-MM management clock</td>
</tr>
<tr>
<td>csr_clk</td>
<td>Input</td>
<td>1</td>
<td>Configuration clock for the Avalon-MM interface, frequency is 100 MHz.</td>
</tr>
<tr>
<td>csr_rst_n</td>
<td>Input</td>
<td>1</td>
<td>Reset Avalon-MM interface.</td>
</tr>
<tr>
<td>tx_312.5_clk</td>
<td>Input</td>
<td>1</td>
<td>312.5 MHz clock for MAC TX data path.</td>
</tr>
<tr>
<td>rx_312.5_clk</td>
<td>Input</td>
<td>1</td>
<td>312.5 MHz clock for MAC RX data path.</td>
</tr>
<tr>
<td>tx_156.25_clk</td>
<td>Input</td>
<td>1</td>
<td>156.25 MHz clock for MAC TX data path.</td>
</tr>
<tr>
<td>rx_156.25_clk</td>
<td>Input</td>
<td>1</td>
<td>156.25 MHz clock for MAC RX data path.</td>
</tr>
<tr>
<td>tx_rst_n</td>
<td>Input</td>
<td>1</td>
<td>Active-low reset for MAC TX data path.</td>
</tr>
<tr>
<td>rx_rst_n</td>
<td>Input</td>
<td>1</td>
<td>Active-low reset for MAC RX data path.</td>
</tr>
</tbody>
</table>

continued...
<table>
<thead>
<tr>
<th>Signal</th>
<th>Direction</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xgmii_156_25_clk</td>
<td>Output</td>
<td>1</td>
<td>156.25 MHz output clock from fPLL.</td>
</tr>
<tr>
<td>mac_312_5_clk</td>
<td>Output</td>
<td>1</td>
<td>312.5 MHz output clock from fPLL.</td>
</tr>
<tr>
<td>pll_ref_clk</td>
<td>Input</td>
<td>1</td>
<td>Reference clock for ATX PLL, fPLL, and XAUI PHY.</td>
</tr>
</tbody>
</table>

### 1.4 Hardware and Software Requirements

Intel uses the following hardware and software to test the design.

**Hardware**
- Arria 10 GX FPGA development kit (revision D)
- Dual XAUI to SFP+ HSMC board
- FMC to HSMC adapter board
- USB Download Cable
- SFP+ module with loopback cable

**Software**
- Quartus® Prime version 16.1 (for hardware testing)
- ModelSim simulator

**Related Links**

**Arria 10 Low Latency Ethernet 10G MAC and XAUI PHY Reference Design**
Provides the files for the reference design. Click on the tabs for the appropriate version.

### 1.5 Board Setups

Before you run the design, you need to set up the boards.
The development board has a stop button for system console testing operations, and reset buttons for the Ethernet subsystem and the Dual XAUI to SFP+ HSMC board.

You need the FMC to HSMC adapter board to connect the Arria 10 FPGA development board to the Dual XAUI to SFP+ HSMC board.
• Install jumpers at J13 and J14 as shown in the figure above.
• You must plug the adapter board and HSMC board into the FMCB of the Arria 10 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 it draws power from the Arria 10 FPGA development board.

1.6 Testing the Design

To make sure the design runs well, you must test the design on hardware.

The perform the hardware test, follow these steps:
1. Download and restore the design.
2. Launch the Quartus Prime software and open the project file (top.qpf).
3. Click Processing ➤ Start Compilation to compile the design.
4. Configure the FPGA using the generated configuration file (top.sof).
5. When configuration completes, open the Clock Control application (arria10GX_10ax115sf45_fpga_v15.1.2\examples\board_test_system \ClockController.exe) and change the frequency for U14 CLK2 to 156.25 MHz.
6. Reset the Ethernet system and HSMC board using the push button. You must reset the system whenever you begin a new test.
7. On the Quartus Prime software, click Tools ➤ System Debugging Tools and launch the System Console.
8. In the System Console command shell, change the directory to "system_console" directory.
9. Run the following command to initialize the design:
   `source demo.tcl`
10. Run the required tests using the provided test commands listed in Test Commands on page 11.

1.6.1 Test Commands

This reference design provides various Tcl commands to test the Arria 10 FPGA development board and the Dual XAUI to SFP+ HSMC board in various loopback modes.

<table>
<thead>
<tr>
<th>Command</th>
<th>Mode/Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPBK_POINT</td>
<td>SFPP</td>
<td>Loopback at SFP+ cable</td>
</tr>
<tr>
<td></td>
<td>BCMMPMA</td>
<td>Loopback at BCM8727 PMA</td>
</tr>
<tr>
<td></td>
<td>BCNXGXS</td>
<td>Loopback at BCM8727 XGXS</td>
</tr>
</tbody>
</table>

...continued...
<table>
<thead>
<tr>
<th>Command</th>
<th>Mode/Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTPMA</td>
<td>ALTPMA</td>
<td>Loopback at Intel serial PMA</td>
</tr>
<tr>
<td>BURST_SIZE</td>
<td>Any integer</td>
<td>Number of packets in the burst.</td>
</tr>
<tr>
<td>NUM_BURSTS</td>
<td>Any number greater than 0</td>
<td>Specifies the intended number of bursts.</td>
</tr>
</tbody>
</table>

Each test generates a log file in text file format. View the log to ensure that the traffic monitor does not receive bad packets. The log also provides packet classification and statistics by the MAC TX and RX.

**Note:** Make sure to reset using the push buttons after each test completes.

### Table 7. Test Commands

<table>
<thead>
<tr>
<th>Test</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFP+ Loopback Test (External loopback at the SFP+ cable)</td>
<td>TEST SFPP 10000 1</td>
</tr>
<tr>
<td>BCM8727 PMA Loopback Test</td>
<td>TEST BCMTPMA 10000 1</td>
</tr>
<tr>
<td>BCM8727 XGXS Loopback Test</td>
<td>TEST BCMXGXS 10000 1</td>
</tr>
<tr>
<td>PMA Serial Loopback Test</td>
<td>TEST ALTPMA 10000 1</td>
</tr>
</tbody>
</table>

### 1.7 Simulating the Design

The design provides a testbench for you to verify the design in loopback mode.

The simulation script uses **QUARTUS_ROOTDIR** environment variable to access Intel simulation libraries. You must set the **QUARTUS_ROOTDIR** to point to the Quartus Prime installation path after installation.

**Note:** If you can't find this environment variable, set the variable manually.

### Figure 6. Testbench Block Diagram

To run the simulation, follow these steps:
1. Download and restore the design.
2. Navigate to the a10_llmac_xauli_project\testbench directory.
3. In the TCL Console window, type the following command:
   
   ```
do tb_run.tcl
   ```
4. When the simulation completes, the Transcript window displays the statistics of the transmitted packets and received packets generated by the simulator.

**Figure 7. TX Statistics**

```
# -----------
# TX Statistics
# -----------
# framesOK    = 50
# framesErr   = 0
# framesCRCErr= 0
# octetsOK    = 29384
# pauseMACCtrlFrames = 0
# ifErrors    = 0
# unicastFramesOK = 50
# unicastFramesErr = 0
# multicastFramesOK = 0
# multicastFramesErr = 0
# broadcastFramesOK = 0
# broadcastFramesErr = 0
# etherStatsOctets = 30284
# etherStatsPkts  = 50
# etherStatsUndersizePkts = 0
# etherStatsOversizePkts = 0
# etherStatsPkts64Octets = 4
# etherStatsPkts65to1270octets = 4
# etherStatsPkts128to2550octets = 3
# etherStatsPkts256to511Octet = 16
# etherStatsPkts512to1023Octets = 12
# etherStatsPkts1024to1518Octets = 11
# etherStatsPkts1519toXOctets = 0
# etherStatsFragments = 0
# etherStatsJabbers = 0
# etherStatsCRCErr = 0
# unicastMACCtrlFrames = 0
# multicastMACCtrlFrames = 0
# broadcastMACCtrlFrames = 0
```
Figure 8. RX Statistics

```
# --------
# RX_Statistics
# --------
# framesOK     = 50
# framesErr    = 0
# framesCRCErr = 0
# octetsOK     = 29384
# pauseMACCtrlFrames = 0
# ifErrors     = 0
# unicastFramesOK = 50
# unicastFramesErr = 0
# multicastFramesOK = 0
# multicastFramesErr = 0
# broadcastFramesOK = 0
# broadcastFramesErr = 0
# etherStatsOctets = 30284
# etherStatsPkts  = 50
# etherStatsUndersizePkts = 0
# etherStatsOverSizePkts = 0
# etherStatsPkts64Octets = 4
# etherStatsPkts65to127Octets = 4
# etherStatsPkts128to255Octets = 3
# etherStatsPkts256to511Octets = 16
# etherStatsPkts512to1023Octets = 12
# etherStatsPkts1024to1518Octets = 11
# etherStatsPkts1519toXOctets = 0
# etherStatsFragments = 0
# etherStatsJabbers = 0
# etherStatsCRCErr  = 0
# unicastMACCtrlFrames = 0
# multicastMACCtrlFrames = 0
# broadcastMACCtrlFrames = 0
```

Related Links
Arria 10 Low Latency Ethernet 10G MAC and XAUI PHY Reference Design
Provides the files for the reference design. Click on the tabs for the appropriate version.

1.8 Reference Design Debug Features
This design includes a Signal Tap II (STP) file (top.stp) to help you self-debug if you encounter any design issues on the hardware.

The STP file contains 2 instances:
• Status – monitors the design channel’s ready, reset, and PHY statuses.
• XGMII – monitors the packet condition at XGMII and Avalon-ST interfaces.

**Figure 9. Status Instance**
The figure shows that the design channel is ready with resets deasserted.

**Figure 10. XGMII Instance**
The figure shows the XGMII (10G) and Avalon-ST (client) interface data path signals to monitor and debug packet conditions during transmission and reception.

1.9 Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>February 2017</td>
<td>2017.02.01</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>