AN 830: Intel FPGA Triple-Speed Ethernet and On-Board PHY Chip Reference Design
Contents

1. Intel® FPGA Triple-Speed Ethernet and On-Board PHY Chip Reference Design for Intel® Stratix® 10 Devices ................................................................. 3
   1.1. Features .............................................................................................................. 3
   1.2. Hardware and Software Requirements ................................................................. 3
   1.3. Functional Description .......................................................................................... 4
      1.3.1. Design Components ................................................................................ 5
      1.3.2. Clocking Scheme ...................................................................................... 6
      1.3.3. Reset Scheme .......................................................................................... 7
   1.4. Hardware Testing .................................................................................................. 7
      1.4.1. Test Case—Internal MAC Loopback ............................................................... 7
      1.4.2. Test Case—Avalon Streaming Reverse Loopback .......................................... 14
   1.5. TCL Script ............................................................................................................ 19
      1.5.1. Configuration Script ................................................................................. 19
      1.5.2. Ethernet Packet Generator Script ............................................................... 19
   1.6. Interface Signals .................................................................................................. 20
   1.7. Configuration Registers and Status Registers ....................................................... 20
   1.8. Regenerating Triple-Speed Ethernet Intel FPGA IP .............................................. 24
1. Intel® FPGA Triple-Speed Ethernet and On-Board PHY Chip Reference Design for Intel® Stratix® 10 Devices

The Intel® FPGA Triple-Speed Ethernet and on-board PHY chip reference design demonstrates Ethernet operation between the Triple-Speed Ethernet Intel FPGA IP core and on-board Marvell 88E1111 PHY chip in Intel Stratix® 10 GX FPGA Development Board. In this reference design, the Triple-Speed Ethernet Intel FPGA IP core is connected to the on-board PHY chip through Serial Gigabit Media Independent Interface (SGMII).

1.1. Features

- Single-channel Triple-Speed Ethernet Intel FPGA IP core operating at data rate of 10/100/1000 Mbps.
- Implementation of the SGMII auto-negotiation feature in order to communicate with on-board PHY chip.
- Sequential random burst test is supported in the hardware test and users are allowed to configure the number of packets, payload-data pattern, packet length, source MAC address, and destination MAC address of each burst.
- Support for Ethernet packet transmission and reception through internal MAC loopback path or Avalon® Streaming reverse loopback path.
- Support for packet monitoring on both TX and RX data paths.
- Support for packet statistics report on both MAC transmitter (TX) and MAC receiver (RX).
- Support for System Console user interface. Users can make use of this TCL-based interface to dynamically configure and monitor any registers in this reference design.

1.2. Hardware and Software Requirements

Intel uses the following hardware and software to test the reference design in a Linux system:

- Intel Quartus® Prime Pro Edition software version 19.3
- For hardware testing:
  - Intel Stratix 10 GX L-Tile FPGA Development Board (1SG280LU2F50E2VG)
  - Micro USB cable
  - External Ethernet packet generator (for Avalon Streaming reverse loopback test only)
  - Ethernet Cat5e cable (for Avalon Streaming reverse loopback test only)
1.3. Functional Description

The reference design consists of various components. The following figure shows the design components and the top-level signals of the reference design.

Figure 1. Block Diagram
## 1.3.1. Design Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
</table>
| Triple-Speed Ethernet Intel FPGA IP core      | • This IP core provides an integrated Ethernet MAC, PCS, and PMA solution for Ethernet applications.  
  • During data transmission, the Triple-Speed Ethernet Intel FPGA IP core transmits Ethernet packets from Avalon Streaming interface to a 1.25-Gbps serial LVDS I/O interface and the Ethernet packets receiving operation is done with the opposite way. |
| Ethernet Packet Generator                     | • This module is a Platform Designer custom component that generates Ethernet packets.  
  • It consists of sub-components such as Ethernet packet generation block, CRC generator, Avalon Memory-Mapped registers, and shift register. |
| Ethernet Packet Monitor                       | • This module is a Platform Designer custom component that verifies the payload of all received packets and collects the statistics of each received packet such as number of bytes received.  
  • It consists of sub-components such as CRC checker and Avalon Memory-Mapped registers. |
| Error Adapter                                 | • This adapter is a Platform Designer custom component that used to connect mismatched Avalon Streaming source and sink interface.  
  • By using this adapter, data source and data sink with different bit width can be connected together. For RX-to-TX Avalon Streaming reverse loopback in this reference design, \( ff\_tx\_err \) is a 1-bit error signal while \( rx\_err \) is a 6-bit error signal.  
  • This adapter can match the error conditions that are handled by the Avalon Streaming source and Avalon Streaming sink. |
| Avalon Streaming Multiplexer                  | • This multiplexer is a Platform Designer custom component that accepts data on its two Avalon Streaming sink interfaces and multiplexes the data for transmission on its Avalon Streaming source interface.  
  • One of the Avalon Streaming sink interface is connected to the Avalon Streaming source interface of Ethernet Packet Generator (for forward MAC loopback) and another Avalon Streaming sink interface is connected to the Avalon Streaming source interface of Error Adapter (for reverse loopback).  
  • The packets on Avalon Streaming source interface of this multiplexer will be transmitted to Triple-Speed Ethernet Intel FPGA IP core. |
| Avalon Streaming Splitter                     | • This splitter is a Platform Designer custom component that accepts data from Triple-Speed Ethernet Intel FPGA IP core through Avalon Streaming sink interface and splits the data on its two Avalon Streaming source interfaces.  
  • One of its Avalon Streaming source interface is connected to the Avalon Streaming sink interface of Ethernet Packet Monitor (for forward MAC loopback) and another Avalon Streaming source interface is connected to the Avalon Streaming sink interface of Error Adapter (for reverse loopback). |
| JTAG to Avalon Master Bridge Intel FPGA IP core| This IP core provides a connection between System Console and Platform Designer system through the physical interface. The System Console initiates Avalon Memory-Mapped transactions by sending encoded streams of bytes through bridge's physical interface. |
| IOPLL Intel FPGA IP core                       | • This IP core generates a 100 MHz PLL output clock \( (clk\_100M) \).  
  • This output clock is the clock source for Platform Designer system. All components in this reference design use this output clock. |
1.3.2. Clocking Scheme

Figure 2. Clocking Scheme
1.3.3. Reset Scheme

Figure 3. Reset Scheme

1.4. Hardware Testing

Related Information
- Getting Started with the Design Store

1.4.1. Test Case—Internal MAC Loopback

To run the hardware test case, follow these steps:

1. Download the reference design from Design Store and restore the design using Intel Quartus Prime software.
2. Launch the Intel Quartus Prime software and open the project file (`top.qpf`).
3. Click **Processing ➤ Start Compilation** to compile the design.
4. After the design is compiled successfully, a programming file (`top.sof`) will be generated and located in the `project_directory/output_files` directory.
5. Set up the Intel Stratix 10 GX L-Tile FPGA Development Board.
   a. Connect the programming cable to the JTAG connection port (CN1).
   b. Connect the power adapter to the power supply input (J27).
6. In the Intel Quartus Prime software, select **Tools ➤ Programmer** to launch the programmer.
7. Download the generated programming file (`top.sof`) to the development board using the **Programmer** application.
8. Reset the Ethernet design by either these methods:
• Press the USER_PB0 push button.
• Toggle the In-System Source and Probes bit[0] from 0 to 1 and back to 0.

*Note:* The design must be reset whenever you begin a new test. The RESET_N pin of the Marvell PHY needs to be kept low for 10 ms because the minimum reset requirement of the Marvell PHY is 10 ms.

9. In the Intel Quartus Prime software, select **Tools ➤ System Debugging Tools ➤ System Console** to launch the system console.

10. In the System Console command shell, change the directory to `project_directory/sc_tcl`.

11. Run the following command in the System Console command shell to start TSE MAC, TSE PCS, and on-board PHY chip configurations:

```
source config.tcl
```

*Note:* Open the `config.tcl` script, which located in `project_directory/sc_tcl` directory, and make sure the LOOP_ENA parameter is set to 1 to ensure the MAC loopback mode is enabled. For more information, refer to Configuration Script on page 19.

12. Run the following command in System Console command shell to start generating and monitoring Ethernet packets:

```
source eth_gen_start.tcl
```

*Note:* Open the `eth_gen_start.tcl` script which located in `project_directory/sc_tcl` directory to set to desired configurations. For more information, refer to Configuration Script on page 19.

The Ethernet Packet Monitor automatically starts after the Ethernet packet is generated. The System Console displays the number of packets with/without error received by the Ethernet Packet Monitor (refer to Figure 7 on page 12).

13. Run the following command to view the TSE MAC statistic counters:

```
source tse_stat_read.tcl
```
Sample Output—MAC Configuration Summary

Starting TSE MAC Configuration System Console

Info: Opened JTAG Master Service

Info: Configure TSE MAC

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSE MAC Rev</td>
<td>0x00001301</td>
</tr>
<tr>
<td>TSE MAC write Scratch</td>
<td>0xaaaaaaaaa</td>
</tr>
<tr>
<td>TSE MAC read Scratch</td>
<td>0xaaaaaaaaa</td>
</tr>
<tr>
<td>Command Config</td>
<td>0x00008033</td>
</tr>
<tr>
<td>MAC Address 0</td>
<td>0x223344550</td>
</tr>
<tr>
<td>MAC Address 1</td>
<td>0x000000001</td>
</tr>
<tr>
<td>Frame Length</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Pause Quanta</td>
<td>0x000000000000</td>
</tr>
<tr>
<td>RX Section Empty</td>
<td>0x000001ff0</td>
</tr>
<tr>
<td>RX Section Full</td>
<td>0x000000010</td>
</tr>
<tr>
<td>TX Section Empty</td>
<td>0x000000000000</td>
</tr>
<tr>
<td>TX Section Full</td>
<td>0x00000000010</td>
</tr>
<tr>
<td>RX Almost Empty</td>
<td>0x0000000000</td>
</tr>
<tr>
<td>RX Almost Full</td>
<td>0x0000000000</td>
</tr>
<tr>
<td>TX Almost Empty</td>
<td>0x0000000000</td>
</tr>
<tr>
<td>TX Almost Full</td>
<td>0x0000000000</td>
</tr>
<tr>
<td>MDIO Address 0</td>
<td>0x0000000000</td>
</tr>
<tr>
<td>MDIO Address 1</td>
<td>0x0000000000</td>
</tr>
<tr>
<td>Register Status</td>
<td>0x000000000000</td>
</tr>
<tr>
<td>TX IPG Length</td>
<td>0x000000000000</td>
</tr>
<tr>
<td>TX Command Status</td>
<td>0x000000000000</td>
</tr>
<tr>
<td>RX Command Status</td>
<td>0x000000000000</td>
</tr>
</tbody>
</table>

Info: Closed JTAG Master Service
**Figure 5. Sample Output—PCS Configuration Summary**

Starting TSE PCS Configuration System Console

---

Info: Opened JTAG Master Service

---

Info: Configure TSE PCS

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSE PCS rev</td>
<td>0x00001301</td>
</tr>
<tr>
<td>TSE PCS write scratch</td>
<td>0x0000aaaa</td>
</tr>
<tr>
<td>TSE PCS read scratch</td>
<td>0x0000aaaa</td>
</tr>
<tr>
<td>TSE PCS if_mode</td>
<td>0x0000000b</td>
</tr>
<tr>
<td>TSE PCS control register</td>
<td>0x00001140</td>
</tr>
</tbody>
</table>

Waiting Link Up.....
Link is established!
Partner Ability:
Copper link interface is up.
Copper operating in Full Duplex mode.
Copper operating Speed 1000Mbps

Info: Closed JTAG Master Service
Figure 6. Sample Output—On-Board PHY Chip Configuration Summary

Starting Marvell PHY Configuration System Console

Info: Opened JTAG Master Service

Info: Configure On Board Ethernet PHY Chip

Configure PHY.
Set PHY SPEED to 1000Mbps
Enable PHY Auto-Negotiation
Enable PHY In Full Duplex Mode
PHY read Control Register = 0x00001140
Advertise PHY 100BASE-TX & 10BASE-TX Full Duplex
PHY read AN Advertisement Register = 0x00000141
Advertise PHY 100BASE-T Full Duplex
PHY read 100BASE-T Control Register = 0x00000000
Set PHY Synchronizing FIFO to maximum
Enable PHY MDIX crossover
Set PHY HWCFG_MODE for SGMII to Copper Without Clock
PHY read Extended PHY Specific Status Register = 0x00000848
PHY Link Up.
PHY Speed and Duplex Resolved.
PHY operating in Full Duplex mode.
PHY operating Speed 1000Mbps
PHY read Status Register = 0x00007969
PHY read Control Register = 0x00001140

Info: Closed JTAG Master Service
Figure 7. Sample Output—Ethernet Packet Generator and Ethernet Packet Monitor Statistics

Starting Ethernet Generator / Monitor System Console

Info: Opened JTAG Master Service

Use Ethernet Generator
Number of packets = 10000
Configuration setting:
- Length : Random
- Packets length : 1500
- Pattern : Random
Random seed 0 = 0x56789abc
Random seed 1 = 0x00001234
Source address 0 = 0x2334560
Source address 1 = 0x00001234
Destination address 0 = 0x2334560
Destination address 1 = 0x00001234
Start Ethernet Generator
Monitor receive done
Number of packets received OK = 10000
Number of packets received error = 0

Info: Closed JTAG Master Service
Figure 8. Sample Output—TX and RX MAC Statistic Counters

<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Read Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x48</td>
<td>sFrameTransmitted 00</td>
<td>0x00012710</td>
</tr>
<tr>
<td>0x40</td>
<td>sFrameTransmitted 01</td>
<td>0x00012710</td>
</tr>
<tr>
<td>0x70</td>
<td>sFrameReceived 00</td>
<td>0x00010000</td>
</tr>
<tr>
<td>0x74</td>
<td>sFrameReceived 01</td>
<td>0x00010000</td>
</tr>
<tr>
<td>0x78</td>
<td>sPacketReceived 00</td>
<td>0x005f3b09</td>
</tr>
<tr>
<td>0x7c</td>
<td>sPacketReceived 01</td>
<td>0x005f3b09</td>
</tr>
<tr>
<td>0x80</td>
<td>sPacketFilled 00</td>
<td>0x00010000</td>
</tr>
<tr>
<td>0x84</td>
<td>sPacketFilled 01</td>
<td>0x00010000</td>
</tr>
<tr>
<td>0x88</td>
<td>sPacketFragments 00</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x8c</td>
<td>sPacketFragments 01</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x90</td>
<td>sPacketErrors 00</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x94</td>
<td>sPacketErrors 01</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x98</td>
<td>sPacketErrors 02</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x9c</td>
<td>sPacketErrors 03</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x80</td>
<td>sPacketErrors 04</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x84</td>
<td>sPacketErrors 05</td>
<td>0x00000000</td>
</tr>
</tbody>
</table>

Related Information

- Marvell 88E1111 PHY Configuration Steps
- Intel Stratix 10 GX FPGA Development Kit
1.4.2. Test Case—Avalon Streaming Reverse Loopback

To run the hardware test case, follow these steps:

1. Download the reference design from Design Store and restore the design using Intel Quartus Prime software.
2. Launch the Intel Quartus Prime software and open the project file (top.qpf).
3. Click Processing ➤ Start Compilation to compile the design.
4. After the design is compiled successfully, a programming file (top.sof) will be generated and located in the project_directory/output_files directory.
5. Set up the Intel Stratix 10 GX L-Tile FPGA Development Board.
   a. Connect the external packet generator to the RJ-45 port of the development board (J10) by using Ethernet Cat5e cable.
   b. Connect the programming cable to the JTAG connection port (CN1).
   c. Connect the power adapter to the power supply input (J27).
6. In the Intel Quartus Prime software, select Tools ➤ Programmer to launch the programmer.
7. Download the generated programming file (top.sof) to the development board using the Programmer application.
8. Reset the Ethernet design by either these methods:
   - Press the USER_PB0 push button.
   - Toggle the In-System Source and Probes bit[0] from 0 to 1 and back to 0.
   
   Note: The design must be reset whenever you begin a new test. The RESET_N pin of the Marvell PHY needs to be kept low for 10 ms because the minimum reset requirement of the Marvell PHY is 10 ms.
10. In the System Console command shell, change the directory to project_directory/sc_tcl.
11. Run the following command in the System Console command shell to start TSE MAC, TSE PCS, and on-board PHY chip configurations:
    ```
    source config.tcl
    ```
    
    Note: Open the config.tcl script, which located in project_directory/sc_tcl directory, and make sure the LOOP_ENA parameter is set to 0 to ensure the MAC loopback mode is disabled. For more information, refer to Configuration Script on page 19.

   The System Console displays the copper link connection status and the resolved operating speed and duplex mode of on-board PHY Chip (refer to Figure 11 on page 17).
12. Start to transmit the Ethernet packets from the external packet generator to the development board. Verify the number of packets that successfully loop back to the external packet generator.
13. Run the following command to view the TSE MAC statistic counters:
    ```
    source tse_stat_read.tcl
    ```
Figure 9. Sample Output—MAC Configuration Summary

Starting ISE MAC Configuration System Console

Info: Opened JTAG Master Service

Info: Configure TSE MAC

TSE MAC Rev = 0x00001301
TSE MAC write Scratch = 0x00000000
TSE MAC read Scratch = 0x00000000
Command Config = 0x00000003b
MAC Address 0 = 0x22334456
MAC Address 1 = 0x00000000
Frame Length = 0x00000000
Pause Quanta = 0x00000000
RX Section Empty = 0x00000000
RX Section Full = 0x00000000
TX Section Empty = 0x00000000
TX Section Full = 0x00000000
RX Almost Empty = 0x00000000
RX Almost Full = 0x00000000
TX Almost Empty = 0x00000000
TX Almost Full = 0x00000000
MDIO Address 0 = 0x00000000
MDIO Address 1 = 0x00000000
Register Status = 0x00000000
TX IPG Length = 0x00000000
TX Command Status = 0x00000000
RX Command Status = 0x00000000

Info: Closed JTAG Master Service
Figure 10. Sample Output—PCS Configuration Summary

Starting TSE PCS Configuration System Console

Info: Opened JTAG Master Service

Info: Configure TSE PCS

TSE PCS rev = 0x00001301
TSE PCS write scratch = 0x00000000
TSE PCS read scratch = 0x00000000
TSE PCS if_mode = 0x00000000
TSE PCS control register = 0x00001140
Waiting Link Up.....
Link is established!
Partner Ability:

Copper link interface is up.
Copper operating in Full Duplex mode.
Copper operating Speed 10000Mbps

Info: Closed JTAG Master Service
Figure 11. Sample Output—On-Board PHY Chip Configurations

Starting Marvell PHY Configuration System Console

Info: Opened JTAG Master Service

Info: Configure On Board Ethernet PHY Chip

Configure PHY.
Set PHY SPEED to 1000Mbps
Enable PHY Auto-Negotiation
Enable PHY In Full Duplex Mode
PHY read Control Register = 0x00001140
PHY read AN Advertisement Register = 0x00000001
Advertise PHY 1000BASE-T Full Duplex
PHY read 1000BASE-T Control Register = 0x00000e00
Set PHY Synchronizing FIFO to maximum
Set PHY MMR10000_STEP for SMII to Copper Without Clock
PHY read Extended PHY Specific Status Register = 0x00008484
PHY Link Up.
PHY Speed and Duplex Resolved.
PHY operating in Full Duplex mode.
PHY operating Speed 1000Mbps
PHY read Status Register = 0x00007969
PHY read Control Register = 0x00001140

Info: Closed JTAG Master Service
**Figure 12. Sample Output—TX and RX MAC Statistic Counters**

<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Read Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x48</td>
<td>xfFramesTransmittedOK</td>
<td>0x010e43d</td>
</tr>
<tr>
<td>0x4c</td>
<td>xfFramesReceivedOK</td>
<td>0x010e650</td>
</tr>
<tr>
<td>0x70</td>
<td>xfFramesCRCFailedOctets</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x74</td>
<td>AlignErrors</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x78</td>
<td>xfratesTransmittedOK</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x7c</td>
<td>xfratesReceivedOK</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x80</td>
<td>xfratesCRCFailures</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x84</td>
<td>ixifErrors</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x88</td>
<td>ixifOverRunErrors</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x8c</td>
<td>ixifCRCFailures</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x90</td>
<td>ixifInvalidPacket</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x94</td>
<td>ixifLoopbackPacket</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x98</td>
<td>ixifUndersizedPacket</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x9c</td>
<td>ixifUnicastPacket</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0xa0</td>
<td>ixifBroadcastPacket</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0xa4</td>
<td>ixifMulticastPacket</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0xa8</td>
<td>ixifTransmitErrors</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0xac</td>
<td>ixifReceiveErrors</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0xb0</td>
<td>ethframesOk</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0xb4</td>
<td>ethframesCorrupt</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0xb8</td>
<td>ethframesCRCErr</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0xbc</td>
<td>ethframesInvalidFrm</td>
<td>0x00000000</td>
</tr>
</tbody>
</table>

**Related Information**

- Marvell 88E1111 PHY Configuration Steps
- Intel Stratix 10 GX FPGA Development Kit
1.5. TCL Script

Any text editor can be used to edit the TCL scripts which located in project_directory/sc_tcl. However, Intel recommends that you do not modify the following TCL scripts:

- tse_mac_config.tcl
- tse_marvel_phy.tcl
- eth_gen_mon.tcl
- tse_stat_read.tcl

1.5.1. Configuration Script

The configuration script, config.tcl contains the settings and parameters that configure the Triple-Speed Ethernet MAC, Triple-Speed Ethernet PCS and Marvell PHY registers in this reference design.

- Triple-Speed Ethernet MAC configurations can be changed by configuring the MAC registers. For more information about Triple-Speed Ethernet MAC configuration register space, refer to Triple-Speed Ethernet Intel FPGA IP User Guide.
- Triple-Speed Ethernet PCS configurations can be changed by configuring the PCS registers. For more information about Triple-Speed Ethernet PCS configuration register space, refer to Triple-Speed Ethernet Intel FPGA IP User Guide.
- Marvell PHY configurations can be changed by configuring the on-board PHY chip register.

Related Information
Triple-Speed Ethernet Intel FPGA IP User Guide

1.5.2. Ethernet Packet Generator Script

The Ethernet Packet Generator script, eth_gen_start.tcl contains the parameters and settings to configure the Ethernet Packet Generator registers in this reference design.

Table 2. Ethernet Packet Generator Script Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>number_packet</td>
<td>Sets the total number of packets to be generated by the packet generator.</td>
</tr>
<tr>
<td>eth_gen</td>
<td>Enables or disables the packet generator.</td>
</tr>
<tr>
<td>length_sel</td>
<td>Selects fixed or random packet length.</td>
</tr>
<tr>
<td>pkt_length</td>
<td>Sets the fixed packet length. The packet length can be a value between 24 to 9600 bytes.</td>
</tr>
<tr>
<td>pattern_sel</td>
<td>Selects the data pattern for the random packet length.</td>
</tr>
<tr>
<td>rand_seed</td>
<td>Sets the initial random seed for the PRBS generator. This parameter is only valid when you select random packet length.</td>
</tr>
<tr>
<td>source_addr</td>
<td>Sets the source MAC address.</td>
</tr>
<tr>
<td>destination_addr</td>
<td>Sets the destination MAC address.</td>
</tr>
</tbody>
</table>
1.6. Interface Signals

Table 3. Clock and Reset Signals

<table>
<thead>
<tr>
<th>Signal</th>
<th>Direction</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk_clk</td>
<td>Input</td>
<td>1</td>
<td>This is the reference design clock, which derived from the IOPLL Intel FPGA IP.</td>
</tr>
<tr>
<td>reset_reset</td>
<td>Input</td>
<td>1</td>
<td>A single reset signal that used to reset all logic in the reference design. This reset signal is connected to a push button (USER_PB0).</td>
</tr>
<tr>
<td>triple_speed_ethernet_0_pcs_ref_clk_clock_connection_clk</td>
<td>Input</td>
<td>1</td>
<td>The 125 MHz reference clock for the 1.25 Gbps serial LVDS I/O interface. This clock is sourced from a dedicated reference clock source, which is in the same IO bank as triple_speed_ethernet_0_serial_connection_txp_0 and triple_speed_ethernet_0_serial_connection_rxp_0 pins.</td>
</tr>
</tbody>
</table>

Table 4. 1.25 Gbps Serial Interface Signals

<table>
<thead>
<tr>
<th>Signal</th>
<th>Direction</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>triple_speed_ethernet_0_serial_connection_txp_0</td>
<td>Output</td>
<td>1</td>
<td>SGMII serial differential transmit interface. Connect this interface to the on-board PHY chip.</td>
</tr>
<tr>
<td>triple_speed_ethernet_0_serial_connection_rxp_0</td>
<td>Input</td>
<td>1</td>
<td>SGMII serial differential receive interface. Connect this interface to the on-board PHY chip.</td>
</tr>
</tbody>
</table>

Related Information

Triple-Speed Ethernet Intel FPGA IP User Guide

1.7. Configuration Registers and Status Registers

Table 5. System Register Map

<table>
<thead>
<tr>
<th>Base Address</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>Triple-Speed Ethernet Intel FPGA IP core</td>
</tr>
<tr>
<td>0x00000400</td>
<td>Avalon Streaming Multiplexer</td>
</tr>
<tr>
<td>0x00000800</td>
<td>Ethernet Packet Monitor</td>
</tr>
<tr>
<td>0x00000C00</td>
<td>Ethernet Packet Generator</td>
</tr>
</tbody>
</table>
Table 6. Ethernet Packet Generator Configuration Registers Map

<table>
<thead>
<tr>
<th>Byte Offset</th>
<th>Name</th>
<th>Width</th>
<th>R/W</th>
<th>HW Reset Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>number_packet</td>
<td>32</td>
<td>RW</td>
<td>0x00</td>
<td>Used to specify the number of packets to be generated.</td>
</tr>
</tbody>
</table>
| 0x04        | config_register | 32    | RW   | 0x00           | • Bit 0  
– 0: Fixed packet length  
– 1: Random packet length  
• Bits [14:1]—Specifies the fixed packet length and the valid values are between 24 to 9600 bytes. It is applicable only when you set bit 0 to 0.  
• Bit 15—Specifies the data pattern for random packet length. Set this bit to 0 for incremental data pattern. For random data pattern, set this bit to 1.  
• Bits [31:16]—Reserved. |
| 0x08        | operation     | 32    | RW/RO| 0x00           | • Bit 0—Set this bit to 1 to trigger packet generation. This bit clears after the packet generation is started.  
• Bit 1—Set this bit to 1 to stop the packets generation. The generator will complete its current packet transmission 1st before terminates the packet generation.  
• Bit 2—A value of 1 indicates that the packet generator completes generating the total number of packets specified in the number_packet register. This bit clears each time packet generation triggers.  
• Bit [31:3]—Reserved. |
<table>
<thead>
<tr>
<th>Byte Offset</th>
<th>Name</th>
<th>Width</th>
<th>R/W</th>
<th>HW Reset Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x10</td>
<td>source_addr0</td>
<td>32</td>
<td>RW</td>
<td>0x00</td>
<td>• Used to specify 6-bytes source/destination MAC address.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• source_addr0/destination_addr0 = last four bytes of the address</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Bits [15:0] of source_addr1/destination_addr1 = first two bytes of the address</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Bits [31:16] of source_addr1/destination_addr1 = unused</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• For example, if the source MAC address is 00-1C-23-17-4A-CB, you get the following assignments:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>— source_addr0 = 0x17231C00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>— source_addr1 = 0x0000CB4A</td>
</tr>
<tr>
<td>0x14</td>
<td>source_addr1</td>
<td>32</td>
<td>RW</td>
<td>0x00</td>
<td></td>
</tr>
<tr>
<td>0x18</td>
<td>destination_addr0</td>
<td>32</td>
<td>RW</td>
<td>0x00</td>
<td></td>
</tr>
<tr>
<td>0x1C</td>
<td>destination_addr1</td>
<td>32</td>
<td>RW</td>
<td>0x00</td>
<td></td>
</tr>
<tr>
<td>0x24</td>
<td>packet_tx_count</td>
<td>32</td>
<td>RO</td>
<td>0x00</td>
<td>This register will keep track the number of packets that the generator transmitted successfully. This register will clear if the packet generation is triggered.</td>
</tr>
<tr>
<td>0x30</td>
<td>rand_seed0</td>
<td>32</td>
<td>RW</td>
<td>0x00</td>
<td>• The lower 32 bits of the random seed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Occupies bits 31:0 of the PBRS generator when you set the data pattern to random (bit 15 of the configuration register).</td>
</tr>
<tr>
<td>0x34</td>
<td>rand_seed1</td>
<td>32</td>
<td>RW</td>
<td>0x00</td>
<td>• The middle 32 bits of the random seed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Occupies bits 63:32 of the PBRS generator when you set the data pattern to random (bit 15 of the configuration register).</td>
</tr>
<tr>
<td>0x38</td>
<td>rand_seed2</td>
<td>32</td>
<td>RW</td>
<td>0x00</td>
<td>• The upper 32 bits of the random seed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Occupies bits 91:64 of the PBRS generator when you set the data pattern to random (bit 15 of the configuration register).</td>
</tr>
</tbody>
</table>
Table 7. Ethernet Packet Monitor Configuration Registers Map

<table>
<thead>
<tr>
<th>Byte Offset</th>
<th>Name</th>
<th>Width</th>
<th>R/W</th>
<th>HW Reset Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>number_packet</td>
<td>32</td>
<td>RO</td>
<td>0x00</td>
<td>Total number of packets that the monitor expects to receive.</td>
</tr>
<tr>
<td>0x04</td>
<td>packet_rx_ok</td>
<td>32</td>
<td>RO</td>
<td>0x00</td>
<td>Total number of received good packets.</td>
</tr>
<tr>
<td>0x08</td>
<td>packet_rx_error</td>
<td>32</td>
<td>RO</td>
<td>0x00</td>
<td>Total number of received packets with errors.</td>
</tr>
<tr>
<td>0x0C</td>
<td>byte_rx_count_0</td>
<td>32</td>
<td>RO</td>
<td>0x00</td>
<td>• 64-bit counter that keeps track of the total number of bytes received.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• byte_rx_count_0 represents the lower 32 bits.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• byte_rx_count_1 represents the upper 32 bits.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Read byte_rx_count_0 followed by byte_rx_count_1 in the subsequent cycle to get an accurate count.</td>
</tr>
<tr>
<td>0x10</td>
<td>byte_rx_count_1</td>
<td>32</td>
<td>RO</td>
<td>0x00</td>
<td>• 64-bit counter that keeps track of the total number of cycles the monitor takes to receive all packets.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• cycle_rx_count_0 represents the lower 32 bits.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• cycle_rx_count_1 represents the upper 32 bits.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Read byte_rx_count_0 followed by byte_rx_count_1 in the subsequent cycle to get an accurate count.</td>
</tr>
<tr>
<td>0x14</td>
<td>cycle_rx_count_0</td>
<td>32</td>
<td>RO</td>
<td>0x00</td>
<td>• 64-bit counter that keeps track of the total number of packets specified in the number_packet register.</td>
</tr>
<tr>
<td>0x18</td>
<td>cycle_rx_count_1</td>
<td>32</td>
<td>RO</td>
<td>0x00</td>
<td>• Bit 0—Set this bit to 1 to trigger packets reception. This bit clears after packet reception is started.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Bit 1—Set this bit to 1 to stop packet reception. This bit clears when packet reception starts.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Bit 2—A value of 1 indicates that the packet monitor has received the total number of packets specified in the number_packet register.</td>
</tr>
</tbody>
</table>

continued...
### Related Information

**Triple-Speed Ethernet Intel FPGA IP User Guide**

#### 1.8. Regenerating Triple-Speed Ethernet Intel FPGA IP

The on-chip termination of the LVDS SERDES IP reference clock within the Triple-Speed Ethernet Intel FPGA IP is turned on by default. The reference clock is pinned out to `REF_CLK` port in this reference design. Because an external termination resistor is present on the Intel Stratix 10 GX FPGA Development Kit at the `REF_CLK` port, the generated QIP file of the Triple-Speed Ethernet Intel FPGA IP is modified to turn off the on-chip termination. When upgrading the reference design to a newer Intel Quartus Prime software version or modification is made to the Triple-Speed Ethernet Intel FPGA IP, the QIP file is regenerated as part of the IP regeneration process. To turn off the on-chip termination, follow these steps to modify the QIP file after the IP regeneration:

1. Locate the QIP file of the Triple-Speed Ethernet Intel FPGA IP at
   `project_directory/platform/ip/qsys_top/qsys_top_eth_tse_0/qsys_top_eth_tse_0.qip`
2. Search the `INPUT_TERMINATION` keyword for the LVDS SERDES IP reference clock:
   ```
   set_instance_assignment -entity
   "qsys_top_eth_tse_0_altera_lvds_core14_191_ofc4bti" -library
   "altera_lvds_core14_191" -name INPUT_TERMINATION DIFFERENTIAL -to inclock
   ```
3. Comment out or remove this assignment.
4. Compile the reference design.

Failure to follow these steps will result in a critical warning:

```
Critical Warning(16643): Found INPUT_TERMINATION assignments found for
"REF_CLK" pin with multiple values. Using value: "OFF"
```

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Changes</th>
</tr>
</thead>
</table>
| 2019.10.18       | • Updated the development kit name from Intel Stratix 10 GX Signal Integrity (SI) development kit to Intel Stratix 10 GX FPGA development kit.  
• Added a new topic—Regenerating Triple-Speed Ethernet Intel FPGA IP.  
• Updated the Intel FPGA Triple-Speed Ethernet and On-Board PHY Chip Reference Design for Intel Stratix 10 Devices topic.  
• Updated the Hardware and Software Requirements topic.  
• Updated the Test Case—Internal MAC Loopback topic:  
  — Updated the steps to run the hardware test case.  
  — Updated Figures:  
    • Sample Output—MAC Configuration Summary  
    • Sample Output—PCS Configuration Summary  
    • Sample Output—On-Board PHY Chip Configuration Summary  
• Updated the Test Case—Avalon-ST Reverse Loopback topic:  
  — Updated the steps to run the hardware test case.  
  — Updated Figures:  
    • Sample Output—MAC Configuration Summary  
    • Sample Output—PCS Configuration Summary  
    • Sample Output—On-Board PHY Chip Configuration Summary  
    • Sample Output—TX and RX MAC Statistic Counters  
• Updated Table: Design Components to update the descriptions for the following components:  
  — Triple-Speed Ethernet IP core  
  — I/O Phase-Locked Loop (PLL) core  
• Updated Figures:  
  — Block Diagram  
  — Clocking Scheme  
  — Reset Scheme  
• Removed Figure: Configuration of U5 OUT8 Frequency in Clock Controller.  
• Updated for latest Intel branding standards. |

| 2019.04.18       | • Updated Figure: Block Diagram.  
• Updated for latest Intel branding standards. |

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>December 2017</td>
<td>2017.12.15</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>