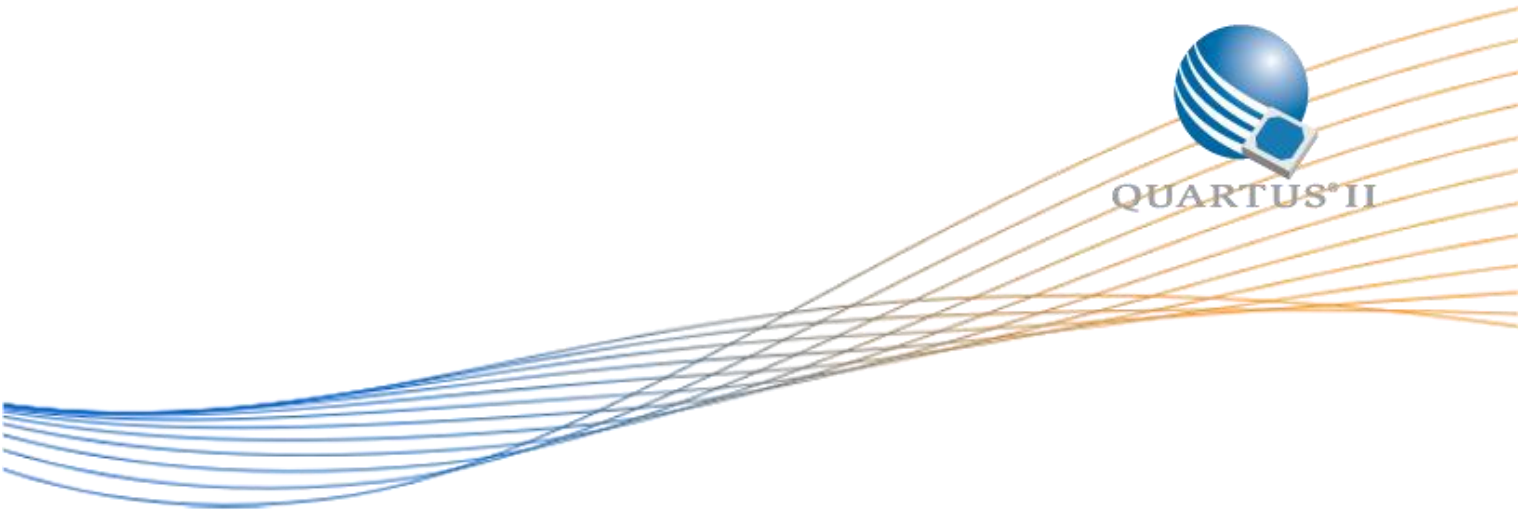


Arria 10 Scalable 10G Ethernet MAC+ Native PHY with IEEE1588v2 Design

Date: 11/11/2016

Revision: 1.0

©2016 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

Table of Contents

Introduction 3
Feature: 3
Design Example Block Diagram 4
Using the reference Design: 6
Requirements: 7
Hardware Test Procedures: 8
Design Example Debug features: 13
Document Revision History 16

Introduction

The intention of creating this design is to provide user reference on how to build an Arria10 10G Ethernet design with IEEE 1588v2 feature using the LL10G MAC IP and A10 Native PHY IP in 10GBaseR 1588 variant. For this design which uses the A10 Native PHY IP instead of the 1G/10GE PHY IP with the native PHY wrapped inside the IP, it serves as a good reference for user who needs a full control over Native PHY to configure for different data rates other than 10G data rate.

This reference design describes a scalable 10G Ethernet design with IEEE1588v2 feature enabled that demonstrates Ethernet operations of the Altera® Low Latency Ethernet 10G MAC and Arria 10 1G/10G Native PHY MegaCore® functions and 10GbaseR 1588 soft FIFO module targeted on Altera Arria 10 SI kit. It provides flexible test and demonstration platforms on which user can control, test, and monitor the Ethernet operations on the TX and RX data paths.

Feature

The design example offers the following features:

- Support 10 Gigabits per second (Gbps) data rate
- Support multi channels Ethernet MAC and Native PHY with 10GBaseR 1588 variants
- Support full control over Native PHY for configuration to different variants.
- Provide packet monitoring system on transmit and receive data paths and report Ethernet MAC statistics counters for transmit and receive data paths.

Design Example Block Diagram (multichannel)

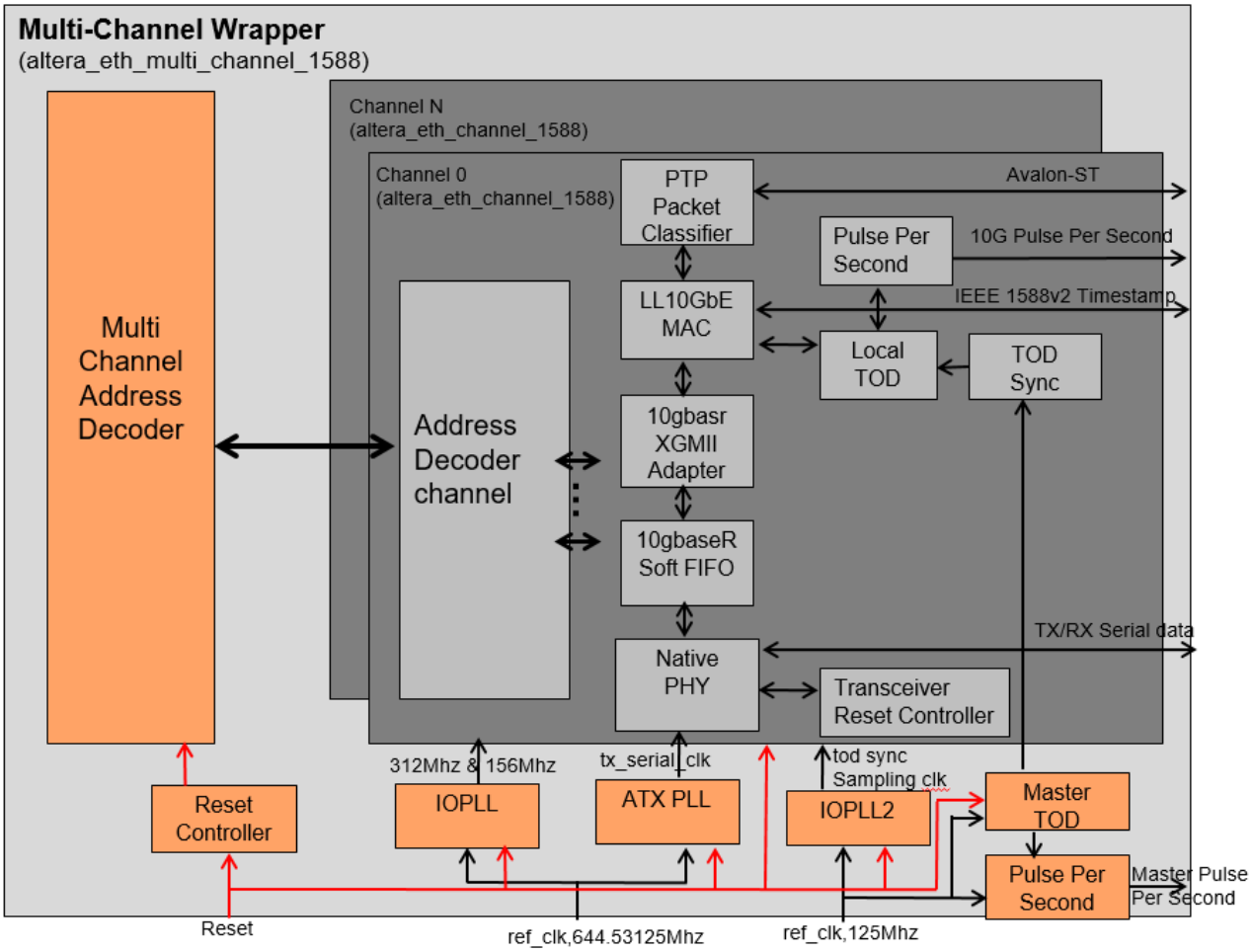


Figure 1: Block Diagram of the Design

Table 1: Design Components of the design

Component	Description
LL 10GbE MAC	The Low Latency Ethernet 10G MAC IP core with the following configuration: • **Speed**: 10G • **Datapath options**: TX & RX • **Enable ECC on memory blocks**: Not selected • **Enable supplementary address**: Selected • **Enable statistics collection**: Selected • **Statistics counters**: Memory-based • All **Legacy Ethernet 10G MAC Interfaces** options: Selected
PHY	A10 Native PHY with 10GBASE-R 1588 variant.
10gbaser XGMII adapter	- Converts 72 bits TX XGMII data interfaces into 64 bits data+8bit control and vice versa for RX XGMII data interfaces. - medium between MAC and 10gbaser phy soft fifo.
10gbaser phy soft fifo	- provides PCS TX & RX latency adjustment value to MAC
Address decoder	Decodes the addresses of the components in each Ethernet channel.
Reset controller	Synchronizes the reset of all design components
Transceiver Reset Controller	The Altera Transceiver PHY Reset Controller IP core. Resets the transceiver.
IOPLL	Generates synchronous 156Mhz & 312Mhz clocks for all design components.
ATX PLL	Generates a TX serial clock for the Arria 10 10G transceiver.
PLL_2	Generates tod sync sampling clocks
FIFO	The Avalon Streaming (Avalon-ST) single-clock FIFO. Buffers the RX and TX data between the MAC IP core and the client.
Master TOD	Provides a master TOD for all channels.
Local TOD	TOD module in each channel.
TOD sync	Module to synch time of day from Master TOD to Local TOD for all channels.
PTP Packet Classifier	Decodes the packet type of incoming PTP packets and returns the decoded information to the Ethernet MAC.
Master PPS	Returns pulse per second (pps) to user for all channels.
Local PPS	Returns pulse per second (pps) to user in each channel.
MDIO	Provides MDIO interface to connect Ethernet MAC to external PHY

Table 2: Parameters for Design Example Customization

Parameter	Description	Default Value
NUM_CHANNELS	Specify the number of channels of 10GbE that will be instantiated in the design example. Range from 1 to 10	2
MDIO_MDC_CLOCK_ DIVISOR	Use this parameter to set the management data input/ output (MDIO) clock divisor. Range from 8 to 64.	32
SHARED_REFCLK_EN	Use this parameter to enable the sharing of reference clock refclk between all channels. 0 : disable sharing 1 : enable sharing	1
FIFO_OPTIONS	Use this parameter to enable the FIFO in between user Avalon-ST and MAC interface. • 0: disable FIFO • 1: enable SC FIFO • 2: enable DC FIFO • 3: enable SC + DC FIFO	1
TSTAMP_FP_WIDTH	Use this parameter to set the timestamp fingerprint width which follows the setting in 1G/10GbE MAC. You must regenerate the MAC IP if this parameter is changed. Enter the new width value in MAC IP regeneration page.	4

Using the reference Design

Requirements

Altera uses the following software & hardware to test the design example:

- Altera Complete Design Suite (ACDS) version 16.0.2
- A10 SI kit clock control (https://www.altera.com/products/boards_and_kits/dev-kits/altera/kit-a10-gx-si.html)
- A10 GX Transceiver Signal Integrity Development Kit for Hardware test
- 1G/10G Finisar SFP+ modules
- ModelSim-SE 10.4d for simulation test
- SFP fiber cable & self-loopback fiber cable

Simulation Test Procedures

To run the simulation on this reference design, follow the steps below:

1. Download and restore the design example from Design Store
2. Invoke Quartus software to generate the simulation model for all IPs Qsys files in the "../platform" folder.
3. Invoke ModelSim® SE 10.4b simulator software.
4. Go to the " ../simulation/ed_sim/mentor" directory
5. In the TCL Console window, type the following commands:
    a. do tb_run.tcl
6. At the end of the simulation, ModelSim simulator will generate statistics of transmitted packets and received packets in the Transcript window.

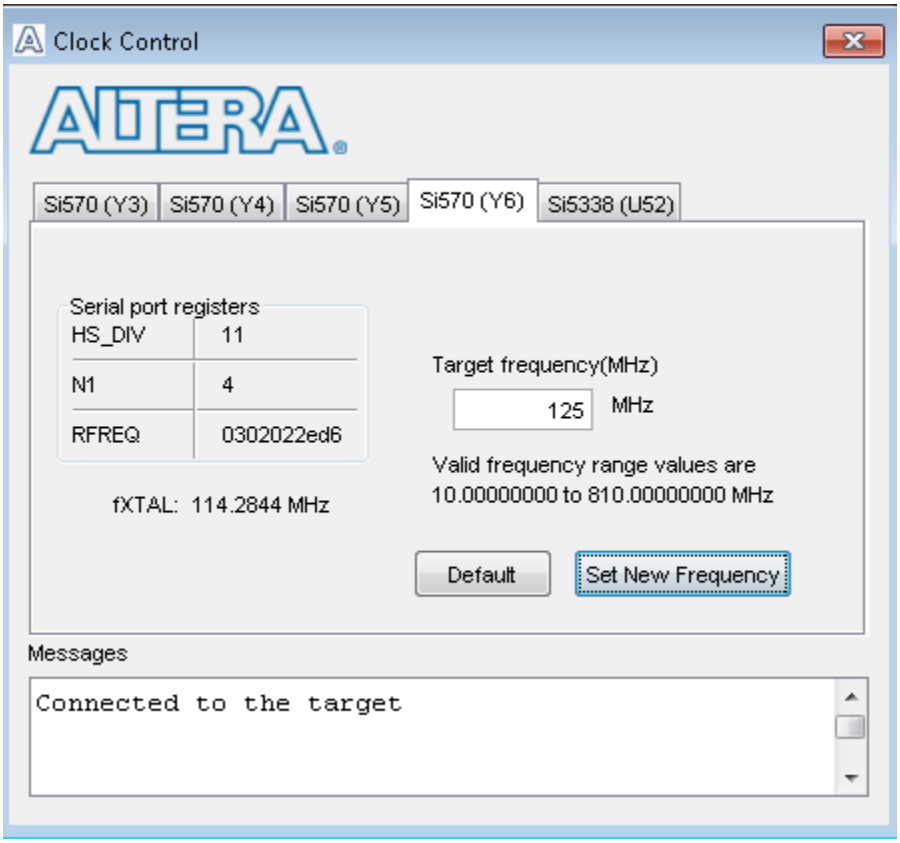
Transcript

```
#
#
# ------------------------
# Channel 0: TX Statistics
# ------------------------
#       framesOK                        = 7
#       framesErr                       = 0
#       framesCRCErr                    = 0
#       octetsOK                        = 522
#       pauseMACCtrlFrames              = 0
#       ifErrors                        = 0
#       unicastFramesOK                 = 7
#       unicastFramesErr                = 0
#       multicastFramesOK               = 0
#       multicastFramesErr              = 0
#       broadcastFramesOK               = 0
#       broadcastFramesErr              = 0
#       etherStatsOctets                = 672
#       etherStatsPkts                  = 7
#       etherStatsUndersizePkts         = 0
#       etherStatsOversizePkts          = 0
#       etherStatsPkts64Octets          = 1
#       etherStatsPkts65to127Octets     = 4
#       etherStatsPkts128to255Octets    = 2
#       etherStatsPkts256to511Octet     = 0
#       etherStatsPkts512to1023Octets   = 0
#       etherStatsPkts1024to1518Octets  = 0
#       etherStatsPkts1519toXOctets     = 0
#       etherStatsFragments             = 0
#       etherStatsJabbers               = 0
#       etherStatsCRCErr                = 0
#       unicastMACCtrlFrames            = 0
#       multicastMACCtrlFrames          = 0
#       broadcastMACCtrlFrames          = 0
#
```

```
#
# ------------------------
# Channel 0: RX Statistics
# ------------------------
#       framesOK                        = 7
#       framesErr                       = 0
#       framesCRCErr                    = 0
#       octetsOK                        = 522
#       pauseMACCtrlFrames              = 0
#       ifErrors                        = 0
#       unicastFramesOK                 = 7
#       unicastFramesErr                = 0
#       multicastFramesOK               = 0
#       multicastFramesErr              = 0
#       broadcastFramesOK               = 0
#       broadcastFramesErr              = 0
#       etherStatsOctets                = 672
#       etherStatsPkts                  = 7
#       etherStatsUndersizePkts         = 0
#       etherStatsOversizePkts          = 0
#       etherStatsPkts64Octets          = 1
#       etherStatsPkts65to127Octets     = 4
#       etherStatsPkts128to255Octets    = 2
#       etherStatsPkts256to511Octet     = 0
#       etherStatsPkts512to1023Octets   = 0
#       etherStatsPkts1024to1518Octets  = 0
#       etherStatsPkts1519toXOctets     = 0
#       etherStatsFragments             = 0
#       etherStatsJabbers               = 0
#       etherStatsCRCErr                = 0
#       unicastMACCtrlFrames            = 0
#       multicastMACCtrlFrames          = 0
#       broadcastMACCtrlFrames          = 0
#
#
# Simulation PASSED
```

Hardware Test Procedures

1. Upon power on the A10 SI board and connect the board to PC via JTAG cable, open the A10 SI kit Clock controller tool to set the clock frequency to 644.53125 and 125 for Y5 and Y6 respectively as shown:



2. Open the programmer and download the design sof file. (Design pre-compiled with 2 channels)

3. Press the Push button, PB0 to hard reset the design.

4. Open System Console tool. Go to /hwtesting/system_console directory and source the "main.tcl" file.

5) Test cases:

Table 3: Hardware Test Cases

Test Case	**Command**		**Example**
1588 test	TEST_1588 <*master channel*> <*slave channel*> <speed>		TEST_1588 0 1 10G
Non 1588 test i)External loopback	TEST_SMA_LB <*channel*> <*speed_ test*> <*burst_size*>		TEST_SMA_LB 0 10G 1000

Table 4: Command Parameters:

Parameter	Valid Values	Description
channel	0 to the number of channel specified for the design	The number of channels for the test. speed_
burst_size	integer	The number of packets to generate for the test.

A: EXTERNAL LOOPBACK TEST:

i) From System Console, type command " TEST_SMA_LB 0 10G 10000" to test channel 0 with 10000 packets.

ii) System console will print out total no of packets successfully received by the channel 0 packet checker and also report out the LL 10G MAC TX & RX statistic counters.

Note: The design is pre-compiled with 2 channels, thus, user can perform similar test for channel 1

B: 1588 TEST:

i) Configure the Channel 0 serial port to channel 1 serial port with SMA cable.

ii) From System Console, type command "TEST_1588 0 1 10G" to set channel 0 as master and channel 1 as slave and starts the 1588 synchronization process in 10G speed.

iii) System console will print out the average delay and offset value calculated.

6. Example of System Console result-

1. Non-1588 test (screenshot)

```
% TEST_SFPP_WO_ENA_ST_LOOPBACK 0 10000
 CONFIGURE CHANNEL 0
                                setting up mac with a basic working config
                                setting 0xC5C4 into rxmac primary address Reg-1
                                setting 0xC3C2C1C0 into rxmac primary address Reg-0
                                enabling: pad and crc stripping in rx mac
                                clearing mac stats registers
                                Disable Avalon ST Loopback


================================================================================
                                B E G I N     C O N F I G U R A T I O N
================================================================================
                                payload length = variable (random) ....
                                payload bytes = random bytes ....
                                burst size =  10000 ....
                                payload length =  1518 ....
                                frame source addres field = F0F1F2F3F4F5 ....
                                frame destination addres field = C5C4C3C2C1C0 ....
                                reseting monitor Packet Counters
                                number of Packets Expected By Monitor = 0x2710
                                burst being injected into device ....
                                -- MONITOR processing frames received .....


                                -- MONITOR Received  Packet#   10000]

                                         -- DONE! - monitor received all expected sum of packets .....

--------------------------------------------------------------------------------
                                -- (MONITOR) GOOD PKTS RECEIVED          = 10000
                                -- (MONITOR) BAD PKTS RECEIVED                 = 0
                                -- (MONITOR) BYTES RECEIVED              = 6385803
                                -- (MONITOR) CYCLES USED                 = 829653
                                -- (MONITOR) THROUGHPUT CALCULATED       = 9.62 Gbps
                                -- (MONITOR) RXBYTECNT_LO32              = 6385803
                                -- (MONITOR) RXBYTECNT_HI32              = 0
                                -- (MONITOR) RXCYCLCNT_LO32              = 829653
                                -- (MONITOR) RXCYCLCNT_HI32              = 0
--------------------------------------------------------------------------------
```

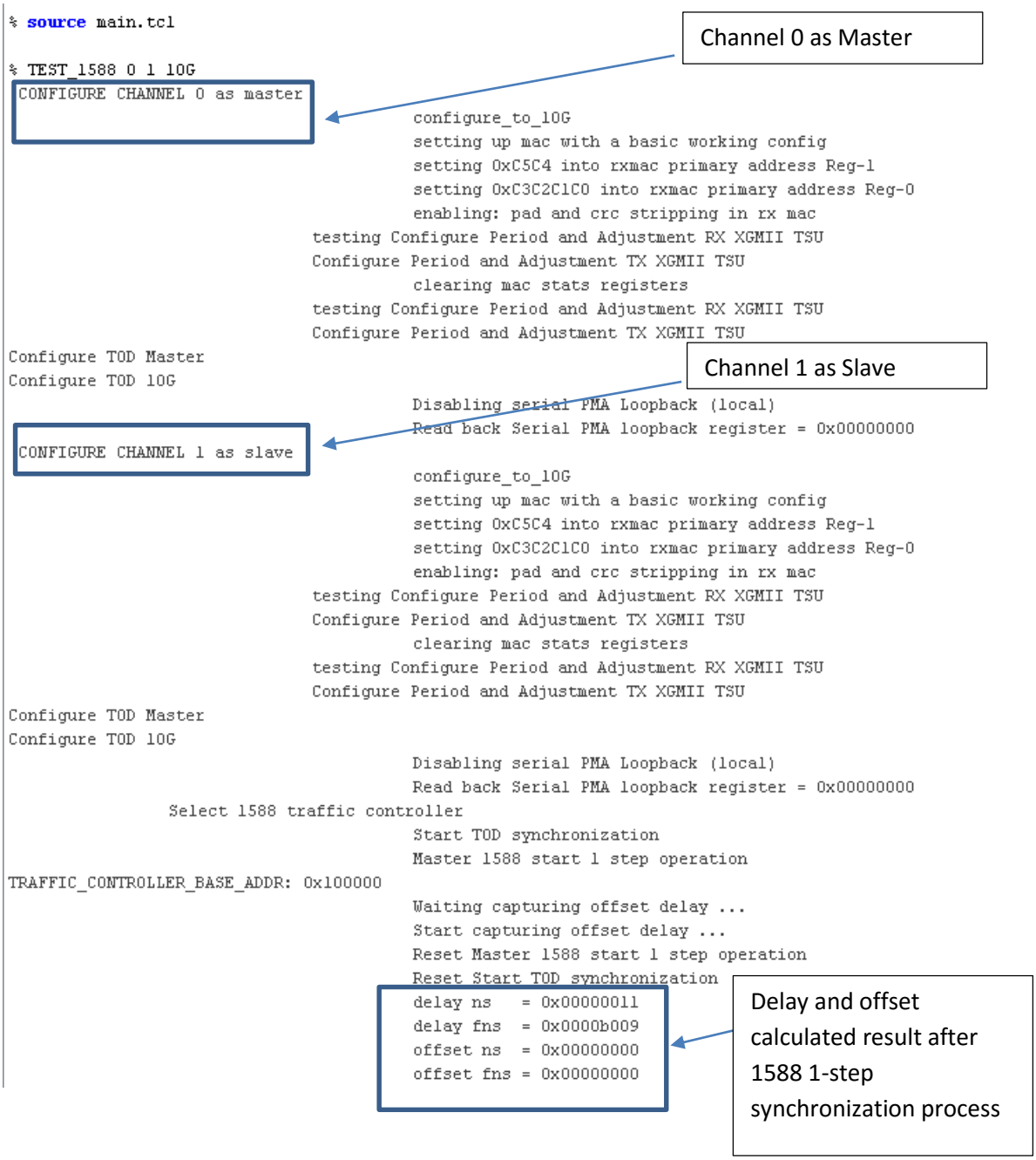
All 10000 good packets received successfully by the monitor module with throughput of 9.62Gbps (running in 10G mode)

```
=================================================================
|  MAC TX STATS REGISTER CHECK
=================================================================
  |# FRAMES_RECEIVED_WITH_ERROR             = 0
  |# UNICAST_FRAMES_WITH_ERROR              = 0
  |# MULTICAST_FRAMES_RECEIVED_WITH_ERROR = 0
  |# BRDCAST_FRAMES_WITH_ERROR                      = 0
  |# FRAMES_RECEIVED_WITH_ONLY_CRCERROR     = 0
  |# VALID_LENGTH_FRAMES_WITH_CRC_ERROR    = 0
  |# JABBER_FRAMES                          = 0
  |# FRAGMENTED_FRAMES                      = 0
  |# INVALID_FRAMES_RECEIVED                = 0
  |# FRAMES_RECEIVED_GOOD                           = 10000
  |# PAUSE_FRAMES_RECEIVED                  = 0
  |# UNICAST_CONTROL_FRAMES                 = 0
  |# MULTICAST_CONTROL_FRAMES               = 0
  |# UNICAST_FRAMES_RECEIVED_GOOD           = 0
  |# MULTICAST_FRAMES_RECEIVED_GOOD         = 10000
  |# BRDCAST_FRAMES_GOOD                    = 0
  |# DATA_AND_PADDING_OCTETS_RECEIVED_GOOD= 6252790
  |# COMPREHENSICE_OCTETS_RECEIVED          = 6432790
  |# FRAMES_WITH_SIZE_64_BYTES             = 376
  |# FRAMES_BETWEEN_SIZE_64AND127_BYTES    = 613
  |# FRAMES_BETWEEN_SIZE_128AND255_BYTES   = 1307
  |# FRAMES_BETWEEN_SIZE_256AND511_BYTES   = 2493
  |# FRAMES_BETWEEN_SIZE_512AND1K_BYTES    = 2789
  |# FRAMES_BETWEEN_SIZE_1KND1518_BYTES    = 2422
=================================================================
|  MAC RX STATS REGISTER CHECK
=================================================================
  |# FRAMES_RECEIVED_WITH_ERROR             = 0
  |# UNICAST_FRAMES_WITH_ERROR              = 0
  |# MULTICAST_FRAMES_RECEIVED_WITH_ERROR = 0
  |# BRDCAST_FRAMES_WITH_ERROR                      = 0
  |# FRAMES_RECEIVED_WITH_ONLY_CRCERROR     = 0
  |# VALID_LENGTH_FRAMES_WITH_CRC_ERROR    = 0
  |# JABBER_FRAMES                          = 0
  |# FRAGMENTED_FRAMES                      = 0
  |# INVALID_FRAMES_RECEIVED                = 0
  |# FRAMES_RECEIVED_GOOD                           = 10000
  |# PAUSE_FRAMES_RECEIVED                  = 0
  |# UNICAST_CONTROL_FRAMES                 = 0
  |# MULTICAST_CONTROL_FRAMES               = 0
  |# UNICAST_FRAMES_RECEIVED_GOOD           = 0
  |# MULTICAST_FRAMES_RECEIVED_GOOD         = 10000
  |# BRDCAST_FRAMES_GOOD                    = 0
  |# DATA_AND_PADDING_OCTETS_RECEIVED_GOOD= 6252790
  |# COMPREHENSICE_OCTETS_RECEIVED          = 6432790
  |# FRAMES_WITH_SIZE_64_BYTES             = 376
  |# FRAMES_BETWEEN_SIZE_64AND127_BYTES    = 613
  |# FRAMES_BETWEEN_SIZE_128AND255_BYTES   = 1307
  |# FRAMES_BETWEEN_SIZE_256AND511_BYTES   = 2493
  |# FRAMES_BETWEEN_SIZE_512AND1K_BYTES    = 2789
  |# FRAMES_BETWEEN_SIZE_1KND1518_BYTES    = 2422
```

MAX TX statistic counter shows all 10000 good packets have been transmitted out from the IP core.

MAX RX statistic counter shows all 10000 good packets have been received successfully by the IP.

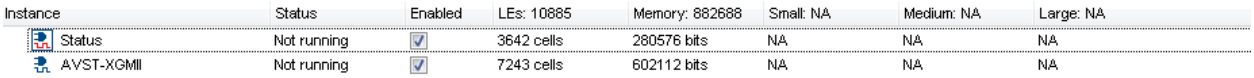
2 Non-1588 test (screenshot)



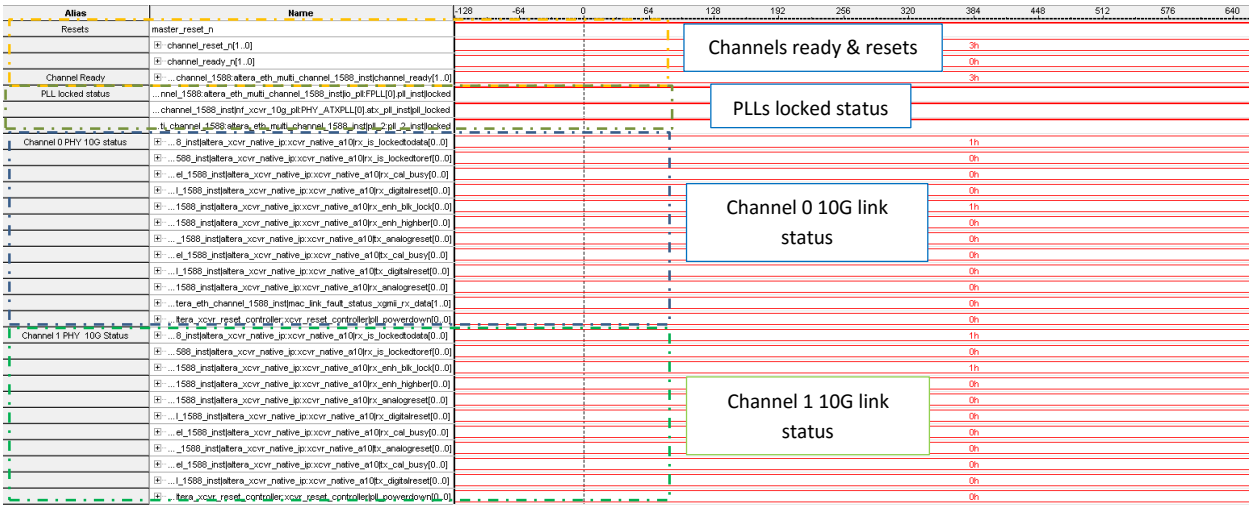
Design Example Debug features

This design has included 1 STP file (stp1.stp) to enable user for debug the hardware design bring up issue. There are 2 instances in the stp file:

- status – monitor design channel ready, master reset , channel reset , PHY status and external clock frequency checker signals.
- AVST-XGMII – monitor the packet condition at client Avalon-ST and XGMII interface

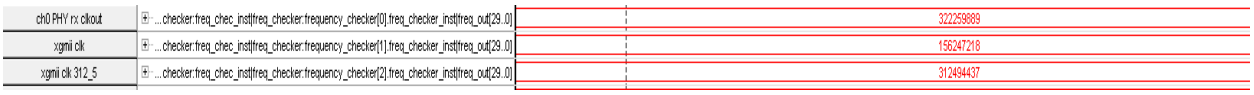


a. status instance:



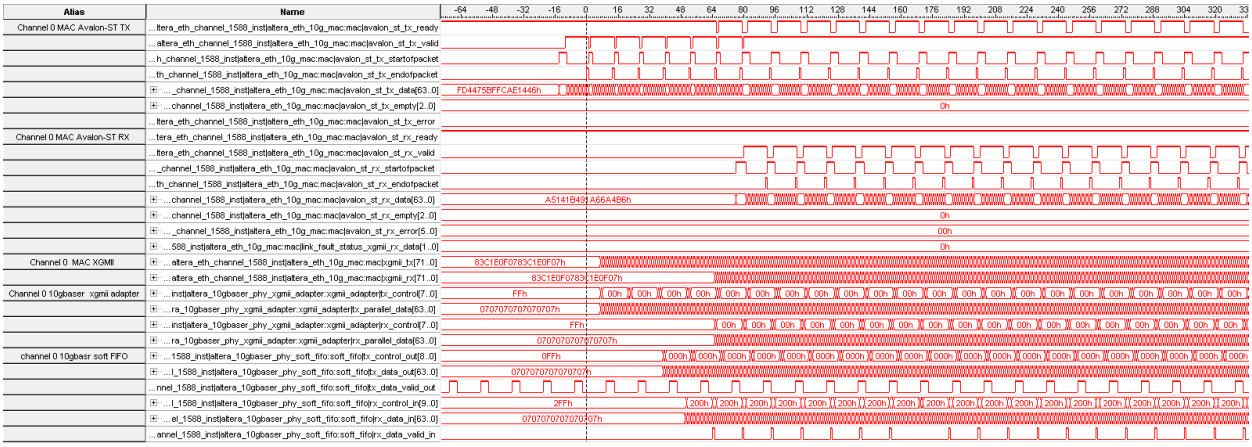
The stp screenshot shows that both channel 0 and 1 are ready with resets de-asserted. Both Channel 0 & 1 PHY are UP with the rx_is_lockedtodata and rx_enh_blk_lock signals are high.

Besides, there is 1 clock frequency checker module included in the design to check for the PHY rx clockout , 10G 156Mhz core clock and 312Mhz core clock frequencies as shown in the stp signals below:



With the adapters module enabled in the MAC IP, a synchronous core clocks of 156Mhz and 312Mhz are required. The native phy rx_clkout should be 322Mhz with the PCS-PMA width set to 32 bits. User needs to make sure that the design has been provided with correct 10G reference clock source: ref_clk_10g = 644Mhz in order to get the correct clock frequencies as mentioned above.

b. AVST-XGMII instance:



Screenshots b) consists of xgmii (10G) and Avalon-ST (client) interfaces data path signals to monitor and debug the packets condition during transmission and reception time.

Note that the STP file provided consists only the channel 0 and channel 1 signals for issue debug. Similar signals can be added for other channels (channel 2 and onwards) if user scales up the design to more than 2 channels by setting the “NUM_CHANNELS “ parameter in the top level wrapper file (altera_eth_top.sv).

Document Revision History

Table 4 shows the revision history for this document.

Table 4: Document Revision History

Date	Version	Changes
11 Novermber 2016	1.0	Initial release