Stratix 10 E-Tile Transceiver Demo Design (Quartus Prime 18.1)

Author: James Murray

Department: European Application Engineering, Customer Experience Group, Intel PSG

Demo Design Scope

This demo design is intended to enable quick and easy transceiver system generation. It can be used to implement a transceiver skeleton designs for transceiver placement and clocking verification. The demo design has ADME enabled so when used with the Quartus Prime Pro Transceiver Toolkit it can be used for hardware transceiver bring-up and signal integrity tuning.

Opening the design and generating IP

- Extract the .zip file and open the XCVR_Top.qpf project file in Quartus Prime Pro software version 18.1.
- Ensure that the option to regenerate all IP is selected in Quartus Settings
    - Quartus Settings > IP Settings > Always regenerate IP
- Synthesise the design. The first time you synthesize the design, it will take a long time. This is because the design includes lots of Quartus generated IP that must be generated.
- When synthesis is complete you can switch off the “Always regenerate IP” option to reduce future compile time
    - Quartus Settings > IP Settings > Never regenerate IP
- You are now ready to begin configuring the demo designs for your system.

Quick Start Guide

The XCVR_Top.vhd file uses a series of generic groups to configure the transceiver architecture.

The following entry will instantiate duplex wrapper 0 containing four PHYs. The datarate will be 10.3Gbps. The “E” denotes this is the E-Tile demo design, and the “NRZ” indicates this is a Non-Return to Zero PHY instance wrapper. The signal integrity pattern mux is enabled. The Reference clock from bank 9A, location 1 on the Right side will be used.

- E_Dx_NRZ0_num:integer:=4; -- Total number of Channels in this group. Enter 1 to 24
- E_Dx_NRZ0_Rate:string:="10G3"; -- "9G8", "10G1", "10G3", "24G3", "25G8", "28G3","USR0"
- E_Dx_NRZ0_EnSIPatterns:boolean:= TRUE;
- E_Dx_NRZ0_refclk_sel:string:="R9A1"; -- Enter REFCLK choice here.

After configuring the generics, run Quartus Prime Pro Analysis and Synthesis, pin out the design, and then fully compile. Standard Stratix 10 device fitting rules apply.

Comprehensive Guide

Demo Design Nomenclature

Throughout the demo design, the following nomenclature is used.

- “E” denotes an E-Tile transceiver IP.
- “Dx” denotes a Duplex channel type.

- “9G8” or “25G8”, etc, denotes data rates of 9.8Gbps or 25.8Gbps. Other data rates are included in the designs
- “NRZ” or “PAM” denote Non Return to Zero or PAM4 modulation types.

For example:

- E_Dx_28G3NRZ.ip – An E-Tile Duplex, NRZ modulated transceiver configured for a datarate of 28.3Gbps.
- E_Dx_51GPAM.ip – An E-Tile Duplex, PAM4 modulated transceiver configured for a datarate of 51Gbps.

This nomenclature extends to the VHDL files too.

- E_Dx_NRZ[n]_Wrap.vhd – A VHDL wrapper file for E-Tile channels
- E_Dx_PAM[n]_Wrap.vhd – A VHDL wrapper file for E-Tile channels

Configuring the XCVR_Top.vhd File Generics

The XCVR_Top.vhd file includes groups of VHDL generics that make it easy to configure the PHYs for your application.

The generics are listed in groups. Each group instantiates a wrapper that contains the number transceiver PHYs entered. The wrapper also includes a data mux to select useful signal integrity patterns

Each group can include any number of supported channels. The following image shows three groups of NRZ generics.

```
E_Dx_NRZ0_num:integer:=1;              -- Total number of Channels ir
E_Dx_NRZ0_Rate:string:="10G3";         -- Enter "9G8", "10G1", "10G3'
E_Dx_NRZ0_EnSIPatterns:boolean:= TRUE; -- Enter TRUE or FALSE
E_Dx_NRZ0_refclk_sel:string:="R9A1";   -- Enter REFCLK choice here.

E_Dx_NRZ1_num:integer:=1;              -- Total number of Channels ir
E_Dx_NRZ1_Rate:string:="25G8";         -- Enter "9G8", "10G1", "10G3'
E_Dx_NRZ1_EnSIPatterns:boolean:= TRUE; -- Enter TRUE or FALSE
E_Dx_NRZ1_refclk_sel:string:="R9A1";   -- Enter REFCLK choice here.

E_Dx_NRZ2_num:integer:=1;              -- Total number of Channels ir
E_Dx_NRZ2_Rate:string:="28G3";         -- Enter "9G8", "10G1", "10G3'
E_Dx_NRZ2_EnSIPatterns:boolean:= TRUE; -- Enter TRUE or FALSE
E_Dx_NRZ2_refclk_sel:string:="R9A0";   -- Enter REFCLK choice here.
```

The demo design includes the following quantities of generic groups:

- Eight groups of NRZ duplex channel types
- Eight groups of PAM4 duplex channel types

Selecting the number of PHYs

You can use as many or as few of the groups in the design as you like. This is done by entering the appropriate integer in the E_Dx_NRZ[n]_num generic string. For example:

- Entering E_Dx_NRZ0_num:integer:=1, instantiates Duplex wrapper 0 containing a single channel NRZ PHY.
- Entering E_Dx_PAM6_num:integer:=4, instantiates Duplex wrapper 6 containing four single channel PAM4 PHYs.

- Entering “0” disables that group and no components are instantiated.

Selecting the Data Rate

The demo design includes some predefined data rates that are selected by entering the appropriate string for the *_Rate generic. For example

E_Dx_NRZ0_Rate:string:="10G3"; -- a 10.3Gbps PHY is instantiated

Predefined data rates include:

- “9G8” - 9.8304Gbps, NRZ, from a 307.2MHz reference clock
- “10G1” - 10137.6Gbps, NRZ, from a 307.2MHz reference clock
- “10G3” - 10312.5Gbps, NRZ, from a 156.25MHz reference clock
- “24G3” – 24.33024Gbps, NRZ, from a 184.32MHz reference clock
- “25G8” – 25.78Gbps, NRZ, from a 156.25MHz reference clock
- “28G3” – 28.25Gbps, NRZ, from a 176.5625 reference clock
- “51G5” – 51.5625Gbps, PAM4, from a 156. 25 reference clock

If you need a different frequency REFCLK to those fixed datarates listed above, you can open the PHY and change it.

User Data Rates

The demo design includes two user PHYs that are selected by entering USR[n] in the *_Rate string. For example:

- E_Dx_NRZ0_Rate:string:="USR1"; -- The predefined data rate you want to use

The string entry is case sensitive.

When using the “USR[1:0]” option, you should open the appropriate PHY and edit these for your own required datarate and reference clock frequency.

For example, to configure the USR1 NRZ PHY for 12.5Gbps from a 250MHz REFCLK you must open and edit the datarate and reference clock frequency in the following IP file.

<project location>\Sources\IP\E_Dx_USR0NRZ.ip

Selecting the Reference Clock

The demo design includes a full complement of E-Tile REFCLK pins for a 5, E-tile 1ST280EY2F55 device fitted to an E-Tile Signal Integrity Development Kit. The REFCLK signals are assigned to pins in the project.

Entering the appropriate transceiver bank and location string into the *_refclk_sel generic string selects which physical REFCLK your PHY will be connected to.

- E_Dx_NRZ0_refclk_sel:string:="R9A1"; -- Enter REFCLK choice here.

Examples of the string nomenclature are:

- REFCLK_GXE**R9A**_CH**4** = **R**ight side, bank **9A**, reference clock **4 = “R9A4”**
- REFCLK_GXB**L8B**_CH**1** = **L**eft side, bank **8B**, reference clock **1 = “R8B1”**

The string entry is case sensitive.

Instead of implementing a mux which would be synthesized in the FPGA fabric that would degrade Tx jitter performance, the REFCLK is chosen by way of generate statements, and the connection made at compile time.

If you use a Stratix 10 device that has a different pinout to this demo design, you should ensure the top level REFCLK pins are assigned correctly for your device.

Using the Signal Integrity Pattern Mux

This feature should only be used for NRZ modulation. It is untested for PAM4.

To enable the signal integrity pattern selector, set the "*_EnSIPatterns" Boolean generic to TRUE. For example

E_Dx_NRZ0_EnSIPatterns:boolean:= TRUE;

This feature enables step, single bit, and clock patterns that can be used to assess channel performance. It can be used to complement the Transceiver Toolkit PRBS patterns and tune channel signal integrity. The Mux is controlled by an In System Source and Probes IP (ISSP).

When using this feature with many channels in a design, the number of ISSP IP instances can become too numerous and cumbersome to use. Consider setting the Boolean generic to FALSE for any channels that you don't need the pattern generator for.

The patterns are selected using a 4-bit In System Source and Probes (ISSP) IP source. The encoding is shown below.

- b"0000" transmits x"5555,5555,5555,5555,5555" – Fast clock pattern
- b"0001" transmits x"3333,3333,3333,3333,3333" -- x33 clock pattern
- b"0010" transmits x"F0F0,F0F0,F0F0,F0F0,F0F0" -- F0 clock pattern
- b"0011" transmits x"FF00,FF00,FF00,FF00,FF00" -- FF00 clock pattern
- b"0100"
- b"0101" transmits x"FFFF,FFFF,FF00,0000,0000" – Step pattern
- b"0110" transmits x"8000,0000,0000,0000,0000" – Single bit (impulse) pattern
- b"0111" transmits x"8000,8000,8000,8000,8000" – Multiple impulse pattern
- b"1000" transmits x"8080,8080,8080,8080,8080" – Multiple fast impulse pattern
- b"1001" transmits x"EFFF,FFFF,FFFF,FFFF,FFFF" – Negative impulse pattern
- b"1010" transmits x"0000,1000,0000,FFFF,FFF0" -- Consecutive Step & Impulse pattern
- b"1011" transmits x"FFFF,EFFF,FFFF,0000,000F" – Negative Consecutive Step & Impulse pattern
- b"1100"
- b"1101" transmits a user pattern defined from a 64 bit In System Source and Probes IP
- b"1110" Connects Rx parallel dataout to Tx Parallel data in using a DC FIFO.
- b"1111" transmits Traffic_Tx_parallel_datain

Configuring the Mux using (ISSP)

When the device is configured, open the 4 bit ISSP IP in Quartus Prime and select one of the binary select codes from above.

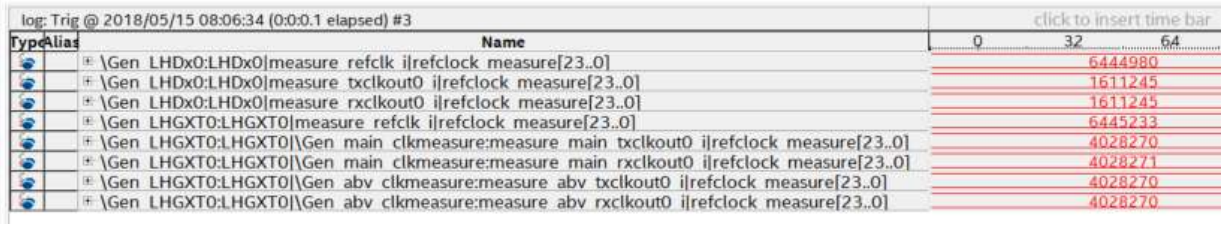
Configuring the User Pattern using (ISSP)

Open the 80-bit ISSP IP in Quartus Prime and configure the data for the pattern you want to transmit. For example, transmitting a step pattern with a random single bit impulse could be implemented with the following code.

x"0000" & x"1000" & x"0000" & x"FFFF" & x"FFF0"

Clock Measurement

The Demo Design includes clock measurement logic that can be used for debug. The REFCLK, tx_clkout[0], and rx_clkout[0] of each group are connected to clock measurement logic. The design assumes a 100MHz sample clock that can be changed in the RTL. To measure the clock you can Signaltap the refclk_measure output bus of the measure_refclk_i instance. You should ensure that the Signaltap bus format is unsigned decimal.



Note: image taken from L-Tile and H-Tile version of this design.

Timing

The design has minimal constraints for timing. It includes constraints for the PHY Avalon Memory Mapped Reconfig Interface clock and REFCLKs too.

```
#**************************************************************
# Create Clock
#**************************************************************
create_clock -period 100MHz          -name reconfig_clk [get_ports {reconfig_clk[0]}]

# E-Tile Left Bank 8B
create_clock -period 156.25MHz       -name REF156M_L8B0 [get_ports {REFCLK_L8B0}]
create_clock -period 307.2MHz        -name REF307M_L8B1 [get_ports {REFCLK_L8B1}]

# E-Tile Left Bank 8C
create_clock -period 156.25MHz       -name REF156M_L8C1 [get_ports {REFCLK_L8C1}]
create_clock -period 176.5625MHz     -name REF176M_L8C4 [get_ports {REFCLK_L8C4}]

# E-Tile Right Bank 9A
create_clock -period 176.5625MHz     -name REF176M_R9A0 [get_ports {REFCLK_R9A0}]
create_clock -period 156.25MHz       -name REF156M_R9A1 [get_ports {REFCLK_R9A1}]

# E-Tile Right Bank 9B
create_clock -period 156.25MHz       -name REF156M_R9B0 [get_ports {REFCLK_R9B0}]
create_clock -period 322.265625MHz   -name REF322M_R9B1 [get_ports {REFCLK_R9B1}]

# E-Tile Right Bank 9B
create_clock -period 322.265625MHz   -name REF322M_R9C0 [get_ports {REFCLK_R9C0}]
create_clock -period 156.25MHz       -name REF156M_R9B1 [get_ports {REFCLK_R9B1}]
```

If you fail to constrain your design, the channels may not appear in the Transceiver Toolkit or you may see other typical timing related problems.

Transceiver Toolkit

The design has ADME enabled so should work with the Quartus Transceiver Toolkit. All PRBS generators, checkers, and PMA settings will work with this design.

To start the Transceiver Toolkit select it from the Tools Menu in Quartus Prime (with the device programmed loaded) using the appropriate version of the Quartus Prime Pro software. When the Transceiver Toolkit is open, load the SOF in the System Console. You should then see your channels and can control them using the Transceiver Toolkit GUI.

Stratix 10 Helper TCL script

Written by Peter Schepers, this Stratix 10 Helper TCL script should also compatible with this design. This tcl file contains several procedures to do additional configuration and status updates for Stratix 10 devices.

http://www.alterawiki.com/wiki/File:ttk_helper_s10tx.tcl

When the Transceiver Toolkit System Console is open, you can source the ttk_helper_s10tx.tcl script located in the scripts folder. This script contains several useful routines for advanced debug and setup. For instance, calibration, full register access, etc.

Trouble Shooting

The design uses standard Quartus Stratix 10 fitting rules documented in the Stratix 10 E-Tile PHY IP user guides.

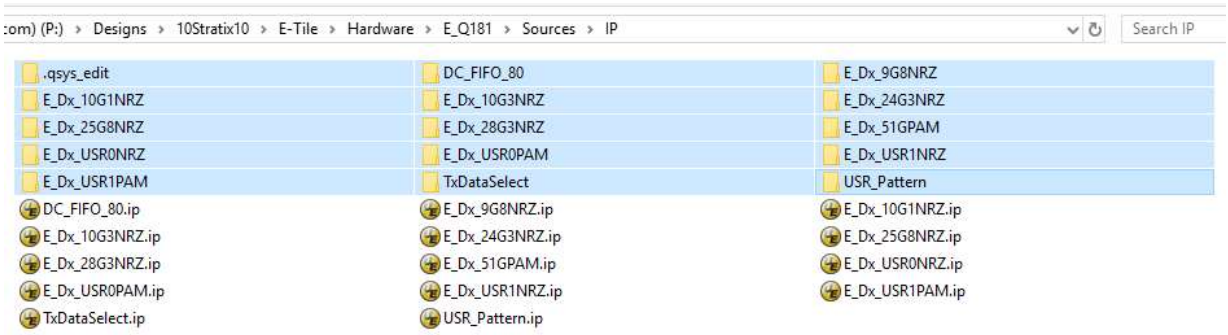
Generic String Entry is Case and Input sensitive

The generic string entries are case and entry sensitive. Failing to enter the exact required string can cause missing PHYs and no fit errors. For instance incorrectly entering the Rate string as “10g3” or “10G” instead of “10G3” will cause problems.

Refer to the RTL comments for correct entry.

IP Regeneration

When switching to a different device part number Intel recommend deleting all of the IP folders such that only the top level *.ip file remains. You should also delete the design database.



You can easily regenerate all IPs again by changing the Quartus settings.

- Ensure that the option to regenerate all IP is selected in Quartus Settings
    - Quartus Settings > IP Settings > Always regenerate IP

The transceivers are not visible in the Quartus Transceiver Toolkit.

Check that the reset to all IP is not asserted

Check that the Avalon Memory Mapped clock is present and connected

Check that the REFCLK is properly connected

Check that your AVMM clock is properly timing constrained

Revision Control

V1.0, 17 October 2018. Initial release