Xilinx Answer 47672
Steps to Update Aurora 64B66B Simplex Tx and Rx Cores to Use Single GTX

Important Note: This downloadable PDF of an Answer Record is provided to enhance its usability and readability. It is important to note that Answer Records are Web-based content that are frequently updated as new information becomes available. You are reminded to visit the Xilinx Technical Support Website and review (Xilinx Answer 47672) for the latest version of this Answer.

Introduction

The purpose of this answer record is to create a Simplex Rx and Simplex Tx Aurora implementation that will use a single transceiver. The CORE Generator software allows you to create a Simplex Rx and Simplex Tx Aurora implementation. As shown in Figure 1, If these cores are integrated as separate designs, then the design would require two transceivers because Aurora core implements independent transceivers for each of them.

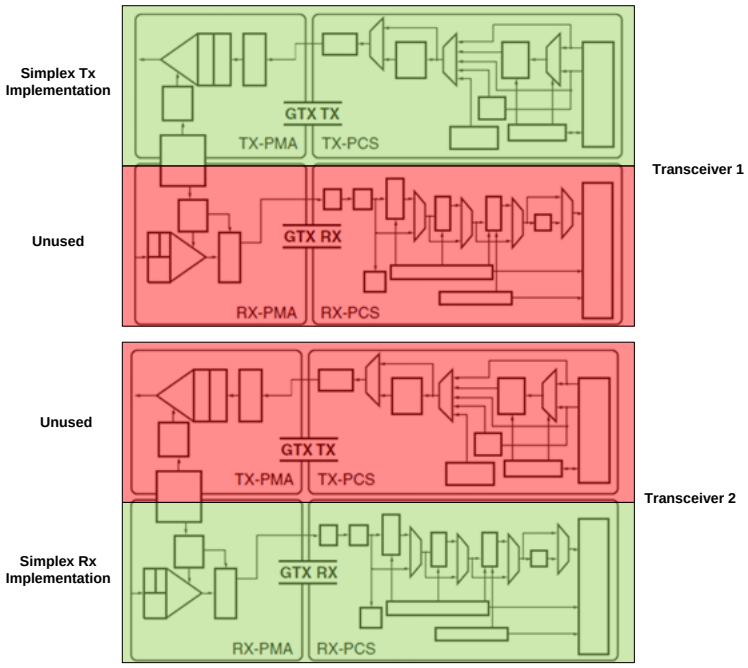

Figure 1 Simplex Rx and Simplex Tx Implementation using CORE Generator

This answer record walks you through the steps that implement the above Simplex Rx and Simplex Tx in a single transceiver as shown in Figure 2. The design thus implemented would have independent controls for the Tx and Rx portion of the transceiver. Any reset applied on the TX/RX side of the transceiver will not propagate or affect the functionality of the corresponding complimentary side.

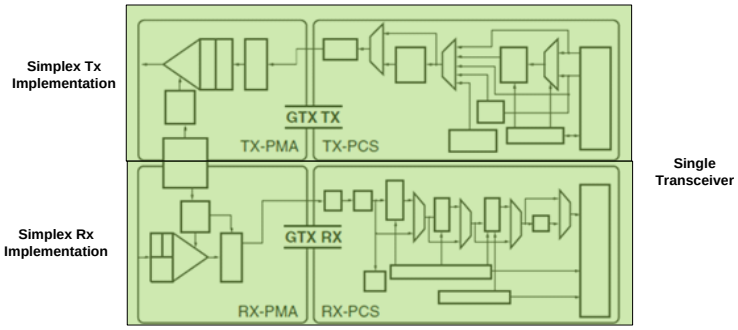

Figure 2 Simplex Rx and Tx implementation that uses single Transceiver (after modifications)

Flowchart

Figure 3 shows the flowchart of generating the Simplex Rx and Simplex Tx block that uses single transceiver.

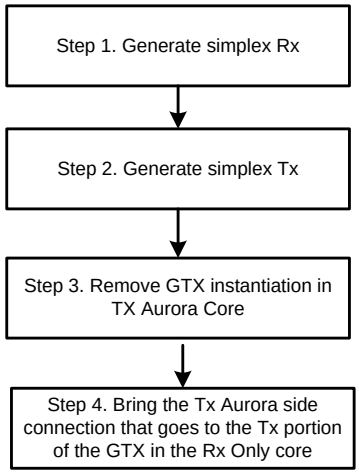

Figure 3 Flowchart to generate Simplex Rx and Simplex Tx that uses single Transceiver

Detailed Description

Step 1. Generate Simplex Rx Core

Use the CORE Generator software to generate Simplex Rx Core.

Step 2. Generate Simplex Tx Core

Use the CORE Generator software to generate Simplex Tx Core.

Following attributes of Simplex Tx core should be the same as Simplex Rx core, except for the “Data flow mode” which will be selected as “Tx-only Simplex”

- Aurora lanes
- GT type
- Line rate
- GT Refclk
- Interface
- Flow control

- User K
- Use ChipScope analyzer
- Lane assignment
- Clock source

Step 3. Remove the GTX connections in the Simplex TX Aurora Core

The basic idea is to remove the GTX instantiation from the Simplex TX Aurora design and connect the required signals to the GTX (TX side) instantiated in the Simplex RX Aurora design.

Figure 4 shows the location of <component_name>_Tx_only_wrapper.v

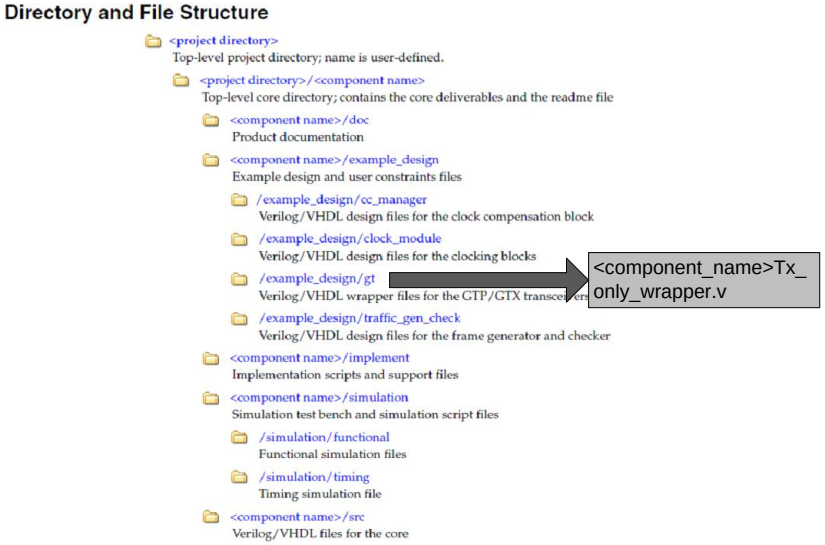

Figure 4 Location and names of the Aurora wrapper of Simplex Tx

The following modification is required in the <component_name>Tx_only_wrapper.v file:

Modification

To remove the GTX instantiation comment out of the following component in <component_name>Tx_only_wrapper.v file <component_name>_MULTI_GT

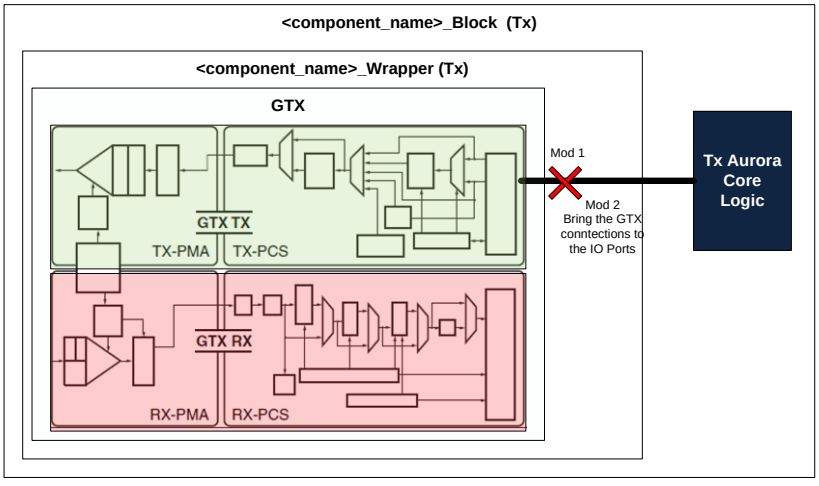

Figure 5 Remove the GTX instantiation in the Simplex Tx design

Step 4. Bring the GTX port connections from Simplex Tx core to Simplex Rx core

In this step, the GTX inputs were brought through Simplex Tx core to Simplex Rx core. GTX outputs required for Simplex Tx core are brought to the Simplex Rx Aurora core top level and are brought to Simplex Tx core. Figure 6 shows the changes done in this step.

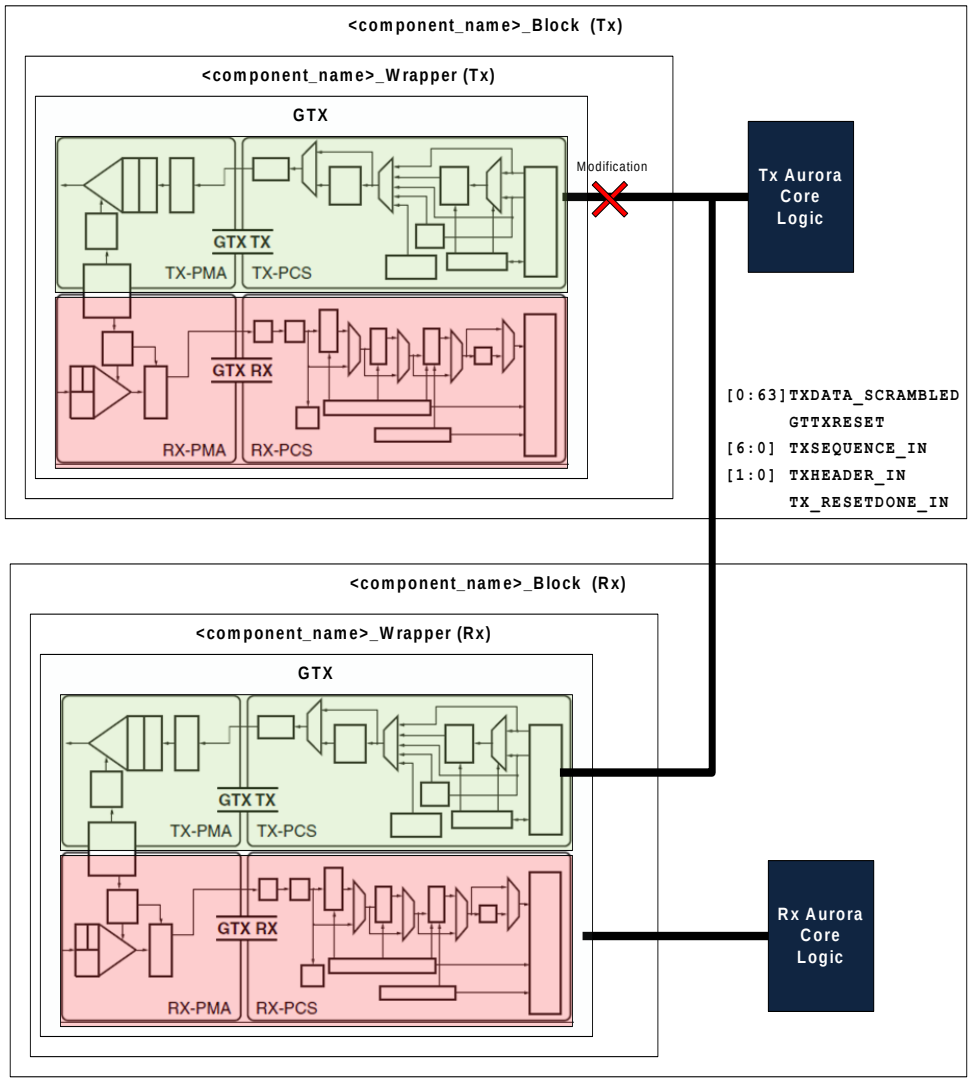

Figure 6 Bringing GTX signals from Simplex Tx core to Simplex Rx core

The following files need to be edited to do this task:

- <component_name>_Tx_only_wrapper.v
- <component_name>_Tx_only_block.v
- <component_name>_Rx_only_block.v
- <component_name>_Rx_only_wrapper.v

The modifications required in all these files will be described in the following sections of this document.

Step 4-1. Edit <component_name>Tx_only_wrapper.v

There are three modifications that need to be done in the <component_name>Tx_only_wrapper file. Figure 4 shows the location of this file.

Modification 1

Remove the following ports in the <component_name>Tx_only_wrapper module in Simplex Tx core. These signals are GTX specific signals and are already available in <component_name>Rx_only_wrapper module.

- `REFCLK1_IN`
- `TXOUTCLK1_OUT`
- `TXHEADER_IN`
- `TXUSRCLK_IN`
- `TX1N_OUT`
- `TX1P_OUT`
- `LOOPBACK_IN`
- `DRPADDR_IN`
- `DRPDI_IN`
- `DRPDO_OUT`
- `DRPRDY_OUT`
- `DRPEN_IN`
- `DRPWE_IN`
- `TXCLK_LOCK`

Modification 2

Add/Edit the following ports in the <component_name>Tx_only_wrapper module in Simplex Tx core.

Add the following signals in the <component_name>Tx_only_wrapper module. These signals are GTX specific signals and will be used to connect to the GTX part of the Simplex RX core:

- `[0:63] TXDATA_SCRAMBLED` - output
- `GTTXRESET` - output
- `[6:0]TXSEQUENCE_IN` - output

Edit the port direction of the following signals from output to input:

- `RESETDONE_OUT` (output -> input)
- `PLLLKDET_OUT` (output -> input)

Modification 3

Edit the following signals and code in the <component_name>Tx_only_wrapper module:

```
assign TXDATA_SCRAMBLED = scrambled_data_i;
```

Table 1 - Code Edits

#	Before Modification	After Modification
1	`assign PLLLKDET_OUT = gt_cplllock_i;`	`//assign PLLLKDET_OUT = gt_cplllock_i;`
2	`assign RESETDONE_OUT = resetdone_i;`	`assign resetdone_i = RESETDONE_OUT;`
3	`assign txsequence_i = txseq_counter_i;`	`assign TXSEQUENCE_IN = txseq_counter_i;`
4	`assign resetdone_i = tx_resetdone_i;`	`//assign resetdone_i = tx_resetdone_i;`
5	`assign gtxreset_i = GTXRESET_IN || !(gt_cplllock_i) ;`	`assign GTTXRESET = GTXRESET_IN || !(PLLLKDET_OUT) ;`
6	`assign gt_cpllreset_i = GTXRESET_IN;`	`// assign gt_cpllreset_i = GTXRESET_IN;`
7	`always @ (posedge TXUSRCLK2_IN) begin tx_hdr_r <= `DLY TXHEADER_IN; end`	`//always @ (posedge TXUSRCLK2_IN) // begin // tx_hdr_r <= `DLY TXHEADER_IN; // end`

Step 4-2. Edit <component_name>Tx_only_block.v

Figure 7 shows the location of the <component_name>Tx_only_block file. There are four modifications that are required in the Aurora block of Simplex Tx.

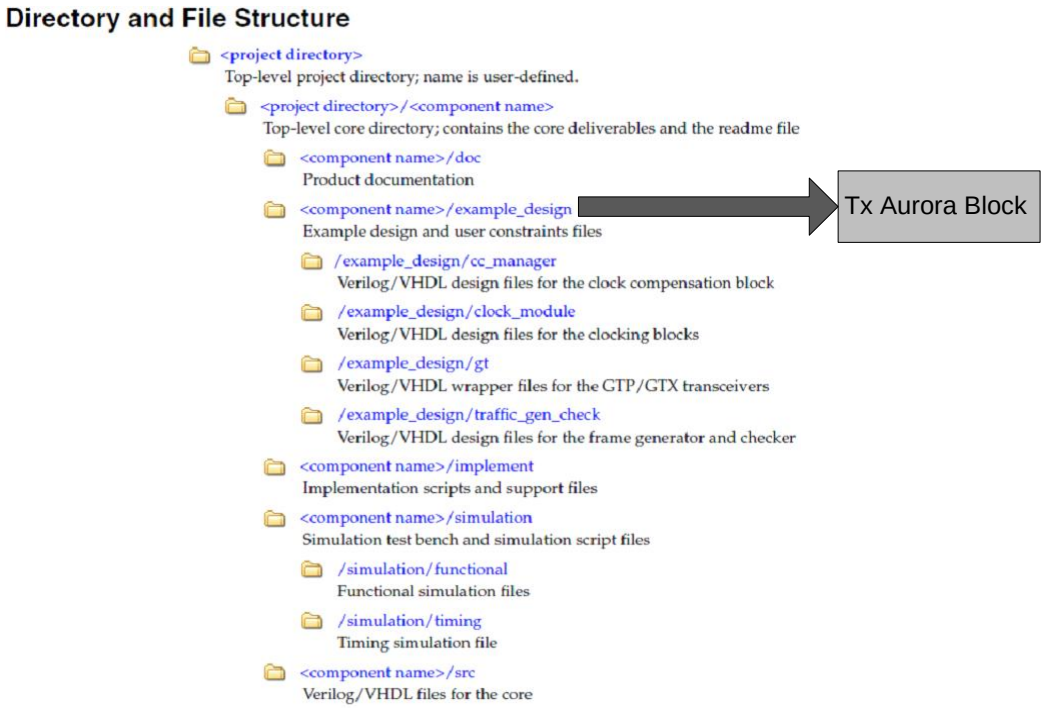

Figure 7 Location and names of the Aurora block of Simplex Tx

Modification 1

Remove the following four GTX signals in the <component_name>Tx_only_block module. These GTX signals are available in Simplex Rx core.

- TXP
- TXN
- GTXQ0
- TX_OUT_CLK
- DRPADDR_IN
- DRP_CLK_IN
- DRPDI_IN
- DRPDO_OUT
- DRPRDY_OUT
- DRPEN_IN
- DRPWE_IN

Modification 2

Add/Edit the following ports in the <component_name>Tx_only_block module in Simplex Tx core.

Add the following signals in the <component_name>Tx_only_block module. These signals are GTX specific signals and will be used to connect to the GTX part of the Simplex RX core:

- [0:63]TXDATA_SCRAMBLED - output
- GTTXRESET – output
- [6:0]TXSEQUENCE_IN – output
- [1:0]TXHEADER_IN – output
- TX_RESETDONE_IN – input

Edit the port direction of MMCM_LOCK signal from output to input:

- MMCM_LOCK (output -> input)

Modification 3

Edit the following signals and code in the <component_name>Tx_only_block module:

```
assign            reset_done_i = TX_RESETDONE_IN;
assign TXHEADER_IN = {tx_char_disp_val_i,tx_char_disp_mode_i};
```

Table 2 - Code Edits

#	Before Modification	After Modification
1	assign txclk_locked_c = !MMCM_NOT_LOCKED ;	//assign txclk_locked_c = !MMCM_NOT_LOCKED ;
2	assign TX_OUT_CLK = raw_tx_out_clk_i;	//assign TX_OUT_CLK = raw_tx_out_clk_i;
3	assign MMCM_LOCK = pll_lock_i;	//assign MMCM_LOCK = pll_lock_i;
4	assign system_reset_c = RESET || MMCM_NOT_LOCKED || !pll_lock_i ;	assign system_reset_c = RESET || MMCM_NOT_LOCKED || !MMCM_LOCK ;
5	aurora_64b66b_v6_2_Tx_only_WRAPPER # (.SIM_GTXRESET_SPEEDUP(SIM_GTXRESET_SPEEDUP)) aurora_64b66b_v6_2_tx_only_wrapper_i (. . .TX1N_OUT(TXN), .TX1P_OUT(TXP), .TXHEADER_IN({tx_char_disp_val_i,tx_char_disp_mode_i}), .TXUSRCLK_IN(SYNC_CLK), .REFCLK1_IN(GTXQ0), .TXOUTCLK1_OUT(raw_tx_out_clk_i), .PLLLKDET_OUT(pll_lock_i), .LOOPBACK_IN(3'b000), .TXCLK_LOCK(txclk_locked_c), .DRPADDR_IN(DRPADDR_IN), .DRP_CLK_IN(DRP_CLK_IN), .DRPDI_IN(DRPDI_IN), .DRPDO_OUT(DRPDO_OUT), .DRPRDY_OUT(DRPRDY_OUT), .DRPEN_IN(DRPEN_IN), .DRPWE_IN(DRPWE_IN), . .);	aurora_64b66b_v6_2_Tx_only_WRAPPER # (.SIM_GTXRESET_SPEEDUP(SIM_GTXRESET_SPEEDUP)) aurora_64b66b_v6_2_tx_only_wrapper_i (. . //.TX1N_OUT(TXN), //.TX1P_OUT(TXP), //.TXHEADER_IN({tx_char_disp_val_i,tx_char_disp_mode_i}), //.TXUSRCLK_IN(SYNC_CLK), //.REFCLK1_IN(GTXQ0), //.TXOUTCLK1_OUT(raw_tx_out_clk_i), .PLLLKDET_OUT(MMCM_LOCK), //.LOOPBACK_IN(3'b000), //.TXCLK_LOCK(txclk_locked_c), //.DRPADDR_IN(DRPADDR_IN), //.DRP_CLK_IN(DRP_CLK_IN), //.DRPDI_IN(DRPDI_IN), //.DRPDO_OUT(DRPDO_OUT), //.DRPRDY_OUT(DRPRDY_OUT), //.DRPEN_IN(DRPEN_IN), //.DRPWE_IN(DRPWE_IN), .TXDATA_SCRAMBLED(TXDATA_SCRAMBLED), .GTTXRESET(GTTXRESET), .TXSEQUENCE_IN(TXSEQUENCE_IN), . .);

Step 4-3. Edit <component_name>Rx_only_block.v file

Figure 6 shows the location of the <component_name>Rx_only_block file.

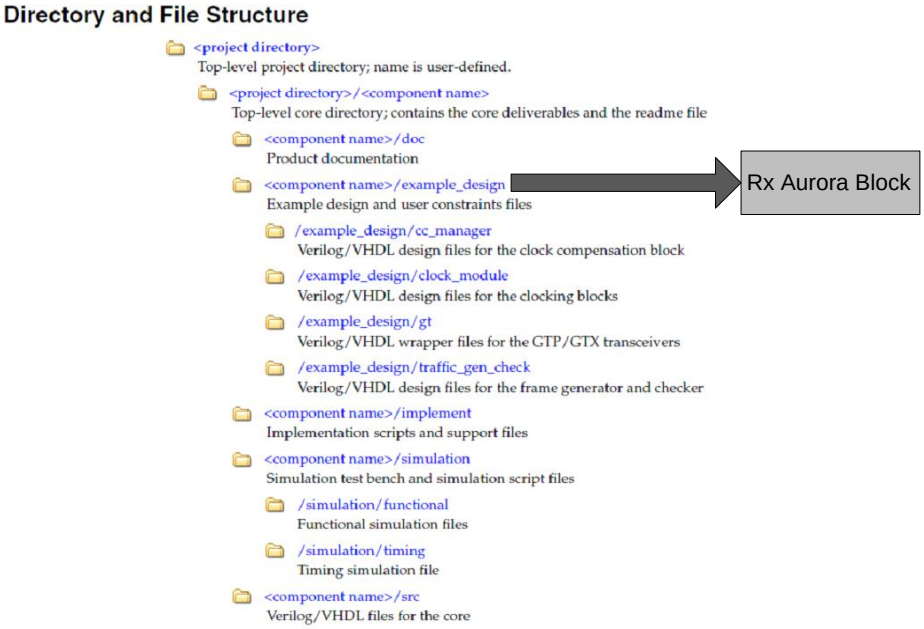

Figure 8 Location and names of the Aurora block of Simplex Rx

There are two modifications that are required in the Aurora block of Simplex Rx:

Modification 1

Add the following GTX signals in the <component_name>Rx_only_block module:

- `TXP` - output
- `TXN` - output
- `GTTXRESET` - input
- `[6:0]TXSEQUENCE_IN` - input
- `[1:0]TXHEADER_IN` - input
- `[0:63]TXDATA_IN` - input
- `TXUSRCLK_IN` - input
- `TXUSRCLK2_IN` - input
- `TXRESETDONE_OUT` – output

Modification 2

Edit the following code in the <component_name>Rx_only_block module:

```
aurora_64b66b_v6_2_Rx_only_WRAPPER  #
    (
        .SIM_GTXRESET_SPEEDUP(SIM_GTXRESET_SPEEDUP)
    )
    aurora_64b66b_v6_2_rx_only_wrapper_i
```

```
    (
        .
        .
        .
        .TX1N_OUT(TXN),
        .TX1P_OUT(TXP),
        .TXHEADER_IN(TXHEADER_IN),
        .TXDATA_IN(TXDATA_IN),
        .GTTXRESET(GTTXRESET),
        .TXRESETDONE_OUT(TXRESETDONE_OUT),
        .TXSEQUENCE_IN(TXSEQUENCE_IN),
        .TXUSRCLK_IN(TXUSRCLK_IN),
        .TXUSRCLK2_IN(TXUSRCLK2_IN),
        .
        .
    );
```

Step 4-4. Edit <component_name>Rx_only_wrapper.v file

The location and the names of the Tx Aurora block are shown in Figure 9.

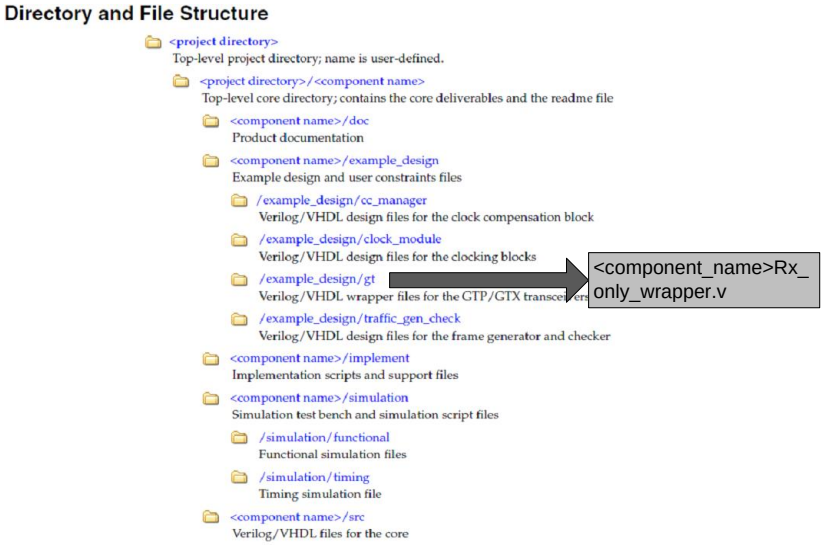

Figure 9 Location and names of the Aurora wrapper of Simplex Rx

There are two modifications that needs to be done in the <component_name>Rx_only_wrapper file.

Modification 1

Add the following GTX signals in the <component_name>Rx_only_wrapper module:

- `TX1N_OUT` - output
- `TX1P_OUT` - output
- `[1:0]TXHEADER_IN` - input
- `[63:0]TXDATA_IN` - input
- `TXUSRCLK_IN` - input
- `TXUSRCLK2_IN` - input
- `TXRESETDONE_OUT` - output
- `[6:0]TXSEQUENCE_IN` - input
- `GTTXRESET` – input
- `Reg [1:0] tx_hdr_r`

Modification 2

Edit the following signals and code in the <component_name>Rx_only_wrapper module:

```
assign txsequence_i = TXSEQUENCE_IN;
assign TXRESETDONE_OUT = tx_resetdone_i;
always @ (posedge TXUSRCLK2_IN)
    begin
        tx_hdr_r   <= `DLY TXHEADER_IN;
    end
```

Edit the following port connections in GTX ports in the <component_name>Rx_only_ wrapper module:

```
.*_TXUSERRDY_IN  (TXCLK_LOCK)
.*_TXHEADER_IN (tx_hdr_r)
.*_TXSEQUENCE_IN (txsequence_i)
.*_GTTXRESET_IN (GTTXRESET)
.*_TXDATA_IN  (TXDATA_IN)
.*_TXUSRCLK_IN  (TXUSRCLK_IN)
.*_TXUSRCLK2_IN    (TXUSRCLK2_IN)
.*_GTXTXN_OUT (TX1N_OUT)
.*_GTXTXP_OUT (TX1P_OUT)
```

Edit <component_name>Rx_only_gtx.v file

The location and the names of the Tx Aurora block are shown in Figure 10.

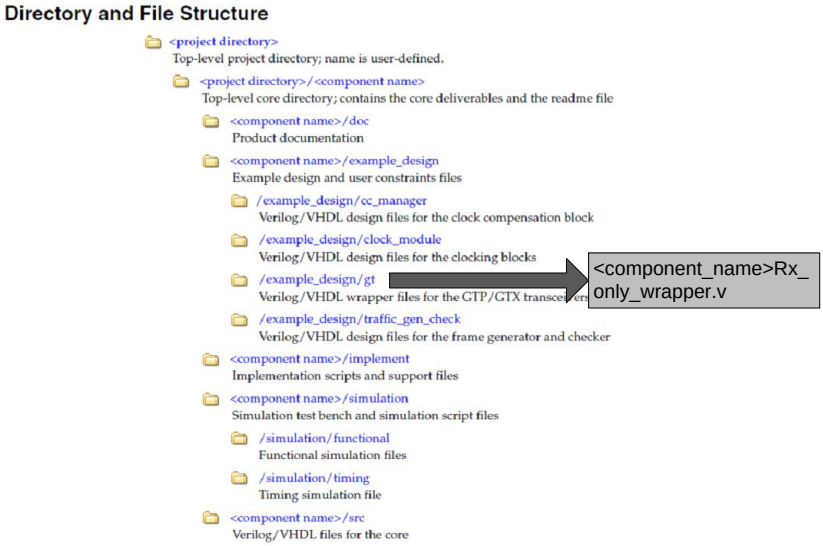

Figure 10 Location and names of the <component_name>Rx_only_gtx.v of Simplex Rx

Modification

TX_DATA_WIDTH parameter value needs to be changed to “64”.

Conclusion

Simplex Tx and Simplex Rx cores can be combined together to share a single transceiver and to keep them operating independent of each other. This document helps to re-route the port connections of GTX from Simplex Tx core to be connected through Simplex Rx core.

If this document does not help to update the design to integrate the Simplex Tx and Simplex Rx cores, please create a WebCase with Xilinx Technical Support.

References

1. LogiCORE IP Aurora 64B/66B v7.1 User Guide (UG775)

Revision History

06/19/2012 - Initial release