**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 73361) for the latest version of this Answer.

---

**Introduction**

One of the main reasons users run into link training issues is due to Signal Integrity (SI) issues on the board. A general guideline of things to check has been provided here to help debug issues related to Signal Integrity.

Link training issues do not entirely depend on the PCIe Core. They are equally a function of the board and how the system is connected. Therefore, it is important to make sure that all of the factors affecting the signal integrity on the board should be thoroughly checked (for example, reference clock quality, voltage signal level etc.). There are various parameters that a user could tune to suit their system and check different things to debug their issue; these will be discussed in detail in this document.

**Hardware Review and SI Debug**

The recommended next step is to review the hardware against the required standards to have a successful PCIe design.

**Spec requirement**

Successful board design for the target line rate for PCIe will require that the board satisfy the design requirements specified for the GT. The following are different parameters of the board to be reviewed and confirmed.

1. REFCLK Phase noise
2. Power supply noise

The “Board Design Guidelines” chapter in the Transceiver user guide provides guidelines to be followed in relation to the reference clock, power supply, and related hardware design.

**REFCLK Phase noise**

The Reference Clock feeds the internal HSSIO PLLs; the PLL is needed for both the transmitter and the receiver side. Quality defects on the REFCLK source increase the TX output jitter and reduce the RX jitter tolerance.

For UltraScale/UltraScale+ transceivers, phase noise requirement for the reference clock input to GT can be found in the device data sheet.
Figure 1 UltraScale GTH Transceiver Reference Clock Selection Phase Noise Mask (Ref: DS892, v1.18)

Measuring reference clock phase noise in hardware

Reference clock phase noise can be measured on the board near the decoupling capacitor, which should be close to the FPGA placement on the board. Some oscilloscopes provide “Spectrum analyzer” function to get the measurement of the clock in frequency domain.

Measuring Power Supply Ripple and Noise

If there are data errors observed, it is necessary to confirm the GT power rails “MGTAVCC, MGTAVTT, MGTVCCAUX” are designed as per the guidelines mentioned in GT user guides – UG576, UG578, “Board Design Guidelines” chapter in these user guides provide specific details on selecting the power supplies, decoupling capacitors needed for each of the power supplies. With reference to GT user guides, the total peak-to-peak noise as measured at the input pin of the UltraScale device should not exceed 10 mVpk-pk. Any excessive noise can lead to higher jitter on the transmitter, data recovery issues in the receiver.

When measuring AC noise on a power rail, it is critical to take measurements with proper equipment and techniques. Chapter 1 of (Xilinx Answer 62181) provides guidance on best practices to measure noise on power supplies. Do the measurement of the noise on the GT power rails and confirm the noise is within the required limits for proper operation of GT.

The list below provides the checks to be performed for this section.

1. Review the reference clock data sheet and/or hardware measurement of reference clock, confirm the phase noise specification is below the phase noise mask of the target GT (Ex: UltraScale GTH)
2. Confirm the hardware has taken care of “GTH Transceiver Reference Clock Checklist” specified in GT user guide.
3. Reference clock hardware schematic is as per the requirement of the IO standard of the clock
4. Review the power supply decoupling capacitors on the board are as per the guidelines mentioned in GT user guide
5. Review the power supply noise measurement and confirm that peak-to-peak noise as measured at the input pin of the UltraScale device is less than 10 mVpk-pk.
6. Confirm that Termination Resistor Calibration Circuit on the board is as per the reference circuit in Transceiver user guide and layout guidelines from the user guide are followed.

Debug Approaches

Integrated Debugging Features

From the Vivado 2016.3 release of UltraScale and UltraScale+ PCI Express cores on, core configuration comes with the following three integrated debug options.
• Enable JTAG Debugger
• Enable In system IBERT
• Enable Descrambler of Gen3 Mode

Each of these features help in debugging different link training issues. As a first step, users should enable these features and collect debug data. (Xilinx Answer 72471) describes all of these debug features (targeting Vivado 2019.1) in detail with screenshots to make it easier for users to understand its implementation and usage.

**Transceiver Debug Port**

If the JTAG debugger result shows GT reset FSM has not completed initialization, then enabling “Additional Transceiver Control and Status Ports” can help to add signals related to the GT initialization process. This option is available in the “Basic” mode of the GUI. The figure below shows this option in the IP configuration GUI.

---

**Figure 2 Additional Transceiver Control and Status Ports**

When this option is enabled, the signals listed in Table 0-1 to Table 0-4 will be available for users to access at the IP interface. The signals brought to the IP interface are a generic list of signals to help with debugging of the issues found with different blocks inside GT. The IP example design assumes the scenario where the input signals brought to the IP interface are not controlled. So, it connects them to proper values to ensure that the IP functions correctly based on the other options set in the GUI.

Based on the issue to be debugged, there can be some output signals which are not required to be captured and input signals which are not required to be controlled. Connect the unused control inputs based on the values driven to them in the IP example design top file.

© Copyright 2020 Xilinx
Table 0-1 to Table 0-4 provide a list of the transceiver debug signals enabled with this option.

### Table 0-1 Ports Used for Transceiver Debug

<table>
<thead>
<tr>
<th>Port</th>
<th>Direction (I/O)</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gt_pcieuseratedone</td>
<td>I</td>
<td>1</td>
<td>Connects to PCIEUSERATEDONE on transceiver channel primitives</td>
</tr>
<tr>
<td>gt_loopback</td>
<td>I</td>
<td>3</td>
<td>Connects to LOOPEBACK on transceiver channel primitives</td>
</tr>
<tr>
<td>gt_txprsforceerr</td>
<td>I</td>
<td>1</td>
<td>Connects to TXPRBSFORCEERR on transceiver channel primitives</td>
</tr>
<tr>
<td>gt_txinhibit</td>
<td>I</td>
<td>1</td>
<td>Connects to TXINHIBIT on transceiver channel primitives</td>
</tr>
<tr>
<td>gt_txprbsssel</td>
<td>I</td>
<td>4</td>
<td>PRBS input</td>
</tr>
<tr>
<td>gt_rxprbsssel</td>
<td>I</td>
<td>4</td>
<td>PRBS input</td>
</tr>
<tr>
<td>gt_rxprbscntreset</td>
<td>I</td>
<td>1</td>
<td>Connects to RXPRBSCNTRESET on transceiver channel primitives</td>
</tr>
<tr>
<td>gt_txelecidle</td>
<td>O</td>
<td>1</td>
<td>Connects to TXELECIDLE on transceiver channel primitives</td>
</tr>
</tbody>
</table>

### Table 0-2 Ports Used for Transceiver Debug

<table>
<thead>
<tr>
<th>Port</th>
<th>Direction (I/O)</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gt_txresetdone</td>
<td>O</td>
<td>1</td>
<td>Connects to TXRESETDONE on transceiver channel primitives</td>
</tr>
<tr>
<td>gt_rxresetdone</td>
<td>O</td>
<td>1</td>
<td>Connects to RXRECLKOUT on transceiver channel primitives</td>
</tr>
<tr>
<td>gt_rxpmarestdone</td>
<td>O</td>
<td>1</td>
<td>Connects to TXPMARESETDONE on transceiver channel primitives</td>
</tr>
<tr>
<td>gt_bxphaligndone</td>
<td>O</td>
<td>1</td>
<td>Connects to TXPHALIGNDONE of transceiver channel primitives</td>
</tr>
<tr>
<td>gt_bxphinitdone</td>
<td>O</td>
<td>1</td>
<td>Connects to TXPHINITDONE of transceiver channel primitives</td>
</tr>
<tr>
<td>gt_txdlysresetdone</td>
<td>O</td>
<td>1</td>
<td>Connects to TXDLYRESETDONE of transceiver channel primitives</td>
</tr>
<tr>
<td>gt_rxphaligndone</td>
<td>O</td>
<td>1</td>
<td>Connects to RXPHALIGNDONE of transceiver channel primitives</td>
</tr>
<tr>
<td>gt_rxtdlysresetdone</td>
<td>O</td>
<td>1</td>
<td>Connects to RXDLYRESETDONE of transceiver channel primitives</td>
</tr>
<tr>
<td>gt_rxsyncdone</td>
<td>O</td>
<td>1</td>
<td>Connects to RXSYNCDONE of transceiver channel primitives</td>
</tr>
<tr>
<td>gt_eyescandataerror</td>
<td>O</td>
<td>1</td>
<td>Connects to EYESCANDATAERROR on transceiver channel primitives</td>
</tr>
<tr>
<td>gt_rxpribsererr</td>
<td>O</td>
<td>1</td>
<td>Connects to RXPRBSERR on transceiver channel primitives</td>
</tr>
</tbody>
</table>

### Table 0-3 Ports Used for Transceiver Debug

© Copyright 2020 Xilinx
Table 0-4 Ports Used for Transceiver Debug

<table>
<thead>
<tr>
<th>Port</th>
<th>Direction (I/O)</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>phy_txeq_preset</td>
<td>O</td>
<td>4</td>
<td>PHY TX Equalization Preset bits</td>
</tr>
<tr>
<td>phy_rst_fsm</td>
<td>O</td>
<td>4</td>
<td>PHY RST FSM state bits</td>
</tr>
<tr>
<td>phy_txeq_fsm</td>
<td>O</td>
<td>3</td>
<td>PHY RX Equalization FSM state bits (Gen3)</td>
</tr>
<tr>
<td>phy_rxeq_fsm</td>
<td>O</td>
<td>3</td>
<td>PHY TX Equalization FSM state bits (Gen3)</td>
</tr>
<tr>
<td>phy_rst_idle</td>
<td>O</td>
<td>1</td>
<td>PHY is in IDLE state</td>
</tr>
<tr>
<td>phy_rst_n</td>
<td>O</td>
<td>1</td>
<td>Synchronized reset generation by sys_clk</td>
</tr>
<tr>
<td>phy_prst_n</td>
<td>O</td>
<td>1</td>
<td>Synchronized reset generation by pipe_clk</td>
</tr>
</tbody>
</table>

The figure below shows the hierarchy of the IP top file in which the “MARK_DEBUG” parameter can be set for the additional transceiver debug signals.
After adding "MARK_DEBUG" to the required signals, synthesize the design and go through the “Set up Debug” flow. All of the signals which have the “MARK_DEBUG” property set are visible in the window “Nets to Debug” as shown in the figure below.

**Figure 3 IP file to set the “MARK_DEBUG” on transceiver debug signals**

**Figure 4 Adding debug nets for transceiver debug**
The figure below shows the directory in which the IP top file is located. Edit this file outside VIVADO using a text editor to set MARK_DEBUG property on the debug signals.

![Figure 5 Directory to look for IP top file](image)

The figure below provides the syntax to set the MARK_DEBUG property to the transceiver debug signals.

```plaintext
(* X_INTERFACE_INFO = "xilinx.com;display_pcie4_uscaleplus:transceiv;
(* mark_debug = "true" *) output wire [15 : 0] phy_txeq_ctrl;
(* X_INTERFACE_INFO = "xilinx.com;display_pcie4_uscaleplus:transceiv;
(* mark_debug = "true" *) output wire [31 : 0] phy_txeq_preset;
(* X_INTERFACE_INFO = "xilinx.com;display_pcie4_uscaleplus:transceiv;
(* mark_debug = "true" *) output wire [31 : 0] phy_rxeq_fsm;
(* X_INTERFACE_INFO = "xilinx.com;display_pcie4_uscaleplus:transceiv;
(* mark_debug = "true" *) output wire [23 : 0] phy_rxeq_fsm;
(* X_INTERFACE_INFO = "xilinx.com;display_pcie4_uscaleplus:transceiv;
(* mark_debug = "true" *) output wire phy_rst_idle;
(* X_INTERFACE_INFO = "xilinx.com;display_pcie4_uscaleplus:transceiv;
(* mark_debug = "true" *) output wire phy_rrst_n;
(* X_INTERFACE_INFO = "xilinx.com;display_pcie4_uscaleplus:transceiv;
(* mark_debug = "true" *) output wire phy_prst_n;
(* X_INTERFACE_INFO = "xilinx.com;display_pcie4_uscaleplus:transceiv;
```

**Figure 6 Syntax to set MARK_DEBUG property**

### Signals to Capture in Vivado ILA

The figures below provide the list of signals to analyze the status of GT initialization during the PCIe link up process.
GT Debug Analysis

When PCIe link up is found to be unsuccessful, enable the “JTAG debugger” option from the IP GUI. (Xilinx Answer 72471) provides a PDF document to explain the steps to be followed when using this feature. The figure below shows that all of the states in the GT reset state machine are in “Green”. This confirms that GT reset state machine has completed all of the required steps to initialize the transceivers at the target line rate.

The answer records below provide guidance to debug issues related to GT.

© Copyright 2020 Xilinx

### Figure 7 Signal list to evaluate GT reset process

<table>
<thead>
<tr>
<th>Signal List</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>/board/PCIe/uscale_plus_0_gttrststatus</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttmgmdone</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttvrxact</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttbcmdkit</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gtt inexpensive</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gtttxnhibit</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gtttxnlock</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttexpreset</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttexpmegadone</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttexprarestatedone</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttexpmegadone</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttexpmegadone</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttexpmegadone</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttexpmegadone</td>
<td>0x000</td>
</tr>
</tbody>
</table>

### Figure 8 Signal list to analyze equalization states during Gen3 link training

<table>
<thead>
<tr>
<th>Signal List</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>/board/PCIe/uscale_plus_0_gttreset</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttreset</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttreset</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttreset</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttreset</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttreset</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttreset</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttreset</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttreset</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttreset</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttreset</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttreset</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttreset</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttreset</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttreset</td>
<td>0x000</td>
</tr>
<tr>
<td>/board/PCIe/uscale_plus_0_gttreset</td>
<td>0x000</td>
</tr>
</tbody>
</table>

Xilinx Answer 73361 – UltraScale and UltraScale+ PCIe Debugging Guide
This section provides guidance to debug when the Reset State Machine output from JTAG debugger shows it is stuck in one of the states.

### IDLE

This is the first FSM state after RST input to the phy_rst_fsm is released. During this state, all of the reset inputs gets initialized.
POWERGOOD

During this state, the RST FSM checks for the GTPOWERGOOD indicator from Transceivers. If the FSM is not moving to this state, the power supply level on MGTAVTT, MGTAVCC and MGTVCCAUX needs to be checked. The figure below shows different signal status to be expected when RST FSM moves to the “POWERGOOD” state.

Figure 10 RST FSM state transition to “POWERGOOD” state

PLL Not Locked

With PCIe Gen3 IP, QPLL is enabled in the GT. PLL must acquire Lock to generate a valid clock. This section provides the checks that can be done to narrow down this issue.

Is the Reference Clock present and with the appropriate frequency?

In most cases, the reference clock to PCIe endpoint is driven from the host machine. So if the host machine is not powered up, it is expected that the reference clock to the PCIe endpoint is not present.

Review the board schematics to confirm that the reference clock used with the PCIe design is mapped to the correct device pin location. If the reference clock is mapped to an incorrect pin and if the clock supplied on the pin makes the VCO in the QPLL operate beyond its rated frequency, QPLL cannot achieve lock.

Is the reference Clock AC coupled and with appropriate swing?

The device data sheet provides the specification for Differential peak-to-peak input voltage, AC coupling capacitor to be used for the reference clock input. The figure below is a snapshot of the GTH Transceiver Clock DC Input Level Specification for Kintex UltraScale GTH (See (DS892) v1.18). Similar data is available in other UltraScale/UltraScale+ data sheets.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>DC Parameter</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>V_{DIFF}</td>
<td>Differential peak-to-peak input voltage</td>
<td>250</td>
<td></td>
<td>2000</td>
<td>mV</td>
</tr>
<tr>
<td>R_{IN}</td>
<td>Differential input resistance</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>Ω</td>
</tr>
<tr>
<td>C_{EXT}</td>
<td>Required external AC coupling capacitor</td>
<td>10</td>
<td></td>
<td>0</td>
<td>nF</td>
</tr>
</tbody>
</table>

Figure 11 GTH Transceiver Clock DC Input Level Specification (Reference (DS892) v1.18)
What is the Reference Clock phase noise?

The reference clock selected for any transceiver type needs to have phase noise levels below the phase noise mask specification provided for that transceiver. The figure below is a snapshot of the Phase noise requirement for Kintex UltraScale GTH transceivers (Reference DS892 v1.18).

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Offset Frequency</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>QPLLREFCLKMASK[1][2]</td>
<td>QPLL0/QPLL1 reference clock select phase noise mask at REFCLK frequency = 312.5 MHz.</td>
<td>10 kHz</td>
<td>–</td>
<td>–</td>
<td>–105</td>
<td>dBC/Hz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100 kHz</td>
<td>–</td>
<td>–</td>
<td>–124</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 MHz</td>
<td>–</td>
<td>–</td>
<td>–130</td>
<td></td>
</tr>
</tbody>
</table>

Figure 12 Phase noise requirement for Kintex UltraScale GTH transceivers (Reference DS892 v1.18)

Was Refclk present when the PLL was reset?

The recommended practice is to apply a reset to the PLL when the reference clock is present and in a stable state. If a PLL reset is helping to achieve PLL lock, review the refclk availability when the PLL reset is pulsed. With UltraScale FPGAs, the reference clock can be forwarded to fabric by using the the ODIV2 output of IBUFDS_GTE3 clock buffer. Get this output clock using the following clocking scheme.

IBUFDS_GTE2.ODIV2 → BUFG_GT.I → BUFG_GT.O → ODDR → Board pin

Monitor both the refclk forwarded using this clocking scheme and the reset input to the PCI express core on Oscilloscope.

Figure 13 shows different signal status to be expected when RST FSM moves to “PLLLOCK and TXPROGDIV” states.

TXPROGDIV

RST FSM moves to the TXPROGDIV state after observing that TXPROGDIVRESETDONE output is asserted HIGH.
RESETDONE

After the userclk inputs to GT are available, phy_rst_fsm will assert TXUSRDRDY/RXUSRDRDY inputs to GT. After the assertion of TXUSRDRDY/RXUSRDRDY, the GT will complete initialization and assert RESETDONE outputs.

TXSYNCSTART

In Gen3 mode, TXBUFFER in the GT is bypassed. In this use case, Tx phase alignment needs to be performed to ensure that TXUSRCLK and TXOUTCLK are phase aligned. During TXSYNCSTART state, all of the signals to initiate the phase alignment process are asserted.

TXSYNCDONE

PHYSTATUS

During this state, phase alignment status is verified. phy_rst_fsm will finish when the phase alignment process has been completed.

PCIe Core Configuration - GT Settings

One of the methods to resolve Signal Integrity (SI) issues resulting in PCIe link up issues is to ensure that the PCIe IP is generated with the required GT settings suitable for the hardware. The IP GUI has some selectable options in the “GT Settings” tab.

Selecting Tx/Rx Equalization

In Gen3 speeds (8 GT/s), the PCIe IP sets DFE as the default RX Equalization. However, in some lower loss channels (<15dB loss) LPM is recommended. The figure below shows options in the PCIe IP to select the right equalizer suitable for the target hardware.

© Copyright 2020 Xilinx
Enable Auto RxEq

When this parameter is set to True, it auto selects the Receiver Equalization (LPM or DFE) mode. Select this option if the card type is add-in card to a backplane system where the card can be inserted in any slot and the channel loss can vary based on the slot in which the card is inserted.

Form Factor Driven Insertion Loss Adjustment

Indicates the transmitter to receiver insertion loss at the Nyquist frequency depending on the form factor selection. Three options are provided:

1. Chip-to-Chip: The value is 5 dB (LPM).
2. Add-in Card: The value is 15 dB and is the default option (DFE).
3. Backplane: The value is 20 dB (DFE).

If the hardware is a fixed setup between a Root Port and End Point design, selecting the right setting for this option can help to ensure GT equalization settings will match the hardware.

PLL Selection

This option is available in UltraScale and UltraScale+ Gen2 only. In Gen2 mode, QPLL is recommended, but CPLL can also be used. Switching between these two PLLs can be attempted when jitter output on the serial line is found to be the probable cause.

In UltraScale and UltraScale+ devices, CPLL is always used at Gen1 rates, Gen 3 rates will require QPLL for better jitter performance.

TX Link Partner Preset (UltraScale and UltraScale+)

During the equalization phase of the Gen3 link-up, it is important to ensure that the link partner sets the Transmitter driver to a preset setting that helps the GT receiver equalizers to adapt to a proper equalization tap setting. For Xilinx UltraScale and UltraScale+ transceivers, “Preset 4” is recommended and “Preset 5” is a “quieter” setting – for shorter links. If the link partner is the Root Port, and bypasses Phase2-3 of EQ training, this will not be honored.

Receiver Detect (UltraScale Only)

Link partner detect can fail if the Link Partner is looking at the falling edge. This is due to a GT characteristic.
The PCIe IP selects the “Default” setting for this option. This setting is suitable for all link partners that do not require receiver detection to be performed on the falling edge.

The “Falling Edge” setting will move RX_CM_SEL to 0V (default 800mV) during detect, then return to qualified 800mV. Choose this option for link partners using both Rising and Falling edge detection.

### PCIe Link Training Debug Analysis

This section provides some of the signals to debug the link training issue when the GT has not completed the initialization process. The figure below shows a signal list to confirm that PCIe link-up is achieved. In the figure, LTSSM can be observed to go through all of the required states from "0x00" to "0x10".

![Signal list to confirm PCIe link-up is achieved](image1)

The figure below shows the expected result for an 8-lane design when the end-point has detected the receiver on the Root Port device on all lanes.

![Signal list to analyze during receiver detection](image2)
The figure below shows a sample result of the LTSSM state transition diagram using the JTAG debugger feature. The state highlighted in orange is the final state which LTSSM has reached. The numbers next to the arrows between different states denote the number of times the state transition has happened.

If the LTSSM has not reached L0, the signal list shown in the figure below can be added to ILA to analyze the issue.
By default, PCIe IP’s are set to use slot clock from the edge connector. If the reference clock input to both End Point and Root Port are driven by a local oscillator on the board, then it is required to not set the "Enable Slot Clock Configuration" option. By doing this, the GT will be setup to support 600ppm offset between the reference clocks. The figure below shows the “Enable Slot Clock Configuration” option with the UltraScale+ PCI Express Integrated Block IP.

If spread spectrum clocking is used, reference clock input to both end point and root port must be driven by the same clock source.

Figure 3-83 through Figure 3-86 of (PG054) illustrate high level representations of different clocking topologies.

The figure below shows an ILA capture when lanes are in reverse order on the End point and Root Port. In this case, lane reversal needs to be enabled at either the root port or end point device. PCIe IP’s does not enable this feature by default.

If possible, inserting a protocol analyzer hardware between the end point and root port can provide a graphical view of the TS1’s exchanged during link training.
Gen3 Transmitter Equalization

Introduction

Transmitter equalization is tuned during the equalization stages of PCIe initialization for Gen3 rate. The Receiver requests presets/coefficients from the Link Partner’s TX to achieve an acceptable BER.

The link protocol captures below show various stages of Link Equalization. Link Equalization comprises of 4 phases; details of each phases are described in the PCI express v3.1a spec in section 4.2.3.
<table>
<thead>
<tr>
<th>31 TS Packets</th>
<th>7089352-7098446</th>
<th>1</th>
<th>x1</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM</td>
<td>Link</td>
<td>Lane</td>
<td>N_FTS</td>
</tr>
<tr>
<td>2.5 Gb/s</td>
<td>1</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

**Notes:**
- **RP moves to Phase 1 and advertises LF (02) and FS (20).**
- **EP moves to Phase 5 and reflects Phase 7 – Hint from RP.
### UltraScale and UltraScale+ PCIe Debugging Guide

#### Table 1: TS Packets

<table>
<thead>
<tr>
<th>TS2</th>
<th>COM</th>
<th>Link</th>
<th>Lane</th>
<th>NPTS</th>
<th>Training Control</th>
<th>Data Rate</th>
<th>Eq Control</th>
<th>TSI Symbols</th>
<th>Time Stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.019 ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.074, 820 532 330 s</td>
</tr>
</tbody>
</table>

#### Table 2: TS Packets

<table>
<thead>
<tr>
<th>TS1</th>
<th>Lane</th>
<th>NPTS</th>
<th>Training Control</th>
<th>Data Rate</th>
<th>Eq Control</th>
<th>TSI Symbols</th>
<th>Time Stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.500 ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.074, 820 532 514 s</td>
</tr>
</tbody>
</table>

**EP moves to Phase 1 and advertises its LF (40) and FS (15)**

---

**EP moves to Phase 2 and requests RP for Phase 4**

---

© Copyright 2020 Xilinx

Xilinx Answer 73361 – UltraScale and UltraScale+ PCIe Debugging Guide

19
Phase 2/3 Bypass

Phase 2/3 bypass refers to a state where the EP cannot request the RP to apply a favorable preset to the RP’s transmitter. The System BIOS has options to control Phase 2/3. The figure below shows the LTSSM state transitions when phase 2/3 bypass is enabled by the Root Port. After LTSSM is moved to state 29 (Equalization phase0), the next state is 0b (Recovery. Receiver lock). So, both the Root port and End point have not exchanged any preset requirement in this case.

Equalization Debug

Link Status2 Register in the configuration space (shown below) provides the status of the equalization phases. If the link trains to Gen1 instead of Gen3, this register can be checked to see which equalization phases were successful.
The figure below provides possible causes for LTSSM to be stuck in different equalization phases.

<table>
<thead>
<tr>
<th>cfg_ltssm_state</th>
<th>Possible causes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase0 -&gt; Recovery. Speed</td>
<td>TX/Rx preset hints</td>
</tr>
<tr>
<td>Phase1-&gt; Recovery. Speed</td>
<td>Phase2/3 Bypass Settings</td>
</tr>
<tr>
<td>Phase2-&gt; Recovery. Speed</td>
<td>• Change preset (P5).</td>
</tr>
<tr>
<td></td>
<td>• Change the timeout settings for Phase2.</td>
</tr>
<tr>
<td>Phase3-&gt; Recovery. Speed</td>
<td>Try steps from Phase02 debug</td>
</tr>
</tbody>
</table>

**PHY TXEQ**

The TXEQ module is instantiated in the phy_wrapper. The figure below shows the hierarchy inside the IP where this module is instantiated. It outputs the main cursor, pre-cursor, and post-cursor to be driven to the GT in response to the Preset request from the Root Port. This module gets triggered when LTSSM reaches the phase 3 equalization state. The MAC sends the TXEQ control value to be set to GT which was requested by the Root Port. In response, this module sets the pre-cursor, post-cursor and main cursor values on the GT transmitter. The TXEQ module asserts TXEQ_DONE after setting the GT transmitter port values.
TX Equalization Interface Signals

The following figure shows the TX Equalization interface signals during the speed change. It includes Preset Apply (phy_txeq_ctrl = 2'b01) and Coefficient Query (phy_txeq_ctrl = 2'b11). The speed change is performed in LTSSM Recovery.Speed and LTSSMPolling.Compliance states. The Preset Apply step must be performed after asserting phy_txelecidle and before driving PHY_RATE to the new speed.

© Copyright 2020 Xilinx
The following figures describe the TX Adapt steps. TX Adapt is performed in Phase 3 of the LTSSM Recovery Equalization state for the Upstream Port and Phase 2 of the LTSSM Recovery Equalization state for the Downstream Port.

The following figure shows the New Proposal step where RX proposes a new preset (phy_rxeq_done = 1'b1 and phy_rxeq_adapt_done = 1'b0) upon a TX preset request (phy_rxeq_txpreset with phy_rxeq_ctrl = 2'b10).

The following figure shows the TX Adapt step when receiving Coefficients. It includes Coefficient Apply (phy_txeq_ctrl = 2'b10) and Coefficient Query (phy_txeq_ctrl = 2'b11).

The figure below shows simulation snapshot of transmitter equalization related signals.

© Copyright 2020 Xilinx
### PHY RXEQ

RXEQ instantiated in the phy_wrapper. The figure below shows the hierarchy inside the IP where this module is instantiated. This module is used for sending request for the required link partner’s TX preset/co-eff. This module outputs the following 2 things to the PCIe MAC and signals whether RX equalization is completed or must be requested again.

- Full swing (FS), Low frequency (LF)
- Presets or Co-eff based on select values

This module gets triggered in Phase2 of equalization when the upstream port can either request a new preset or coefficient.
RX Equalization Interface Signals (New Proposal)

The RX equalization process can involve two steps:

- New Proposal
  - RX proposes a new preset (PHY_RXEQ_DONE = 1'b1 and PHY_RXEQ_ADAPT_DONE = 1'b0) upon a TX preset request (PHY_RXEQ_TXPRESET with PHY_RXEQ_CTRL = 2'b10)
- Adapted
RX adapts the preset (PHY_RXEQ_DONE = 1'b1 and PHY_RXEQ_ADAPT_DONE = 1'b1) which TX requests (PHY_RXEQ_TXPRESET with PHY_RXEQ_CTRL = 2'b10).

The following figure shows the New Proposal step where RX proposes a new preset (phy_rxeq_done = 1'b1 and phy_rxeq_adapt_done = 1'b0) upon a TX preset request (phy_rxeq_txpreset with phy_rxeq_ctrl= 2'b10).

The following figure shows the Adaptation step where the RX adapts the preset (phy_rxeq_done =1'b1 and phy_rxeq_adapt_done = 1'b1) which the TX requests (phy_rxeq_txpreset withphy_rxeq_ctrl= 2'b10).

The figure below shows a simulation snapshot of receiver equalization related signals.
RXCDR HOLD

RXCDRHOLD is an input to the GT. This signal needs to be asserted high when RXELECIDLE is asserted high. The PCIe IP has a module to control this signal and it is instantiated in core_top. The figure below shows the hierarchy to this module.

This module implements the logic for holding the CDR logic high in all LTSSM states where RX elec_idle is high. CDR logic is held high in the states mentioned below and until the electrical idle signal is kept asserted.

- LTSSM states -> Recovery.Speed, L1.Entry, L1.Idle or Loopback_Entry_slave
**Link Training Debug Check List**

- Check cfg ltssm state output from the IP to see which state it is stuck on or what is the highest state it can reach before going back to 0. Once a state has been found, check the PCIe Spec to see the requirement in that state to move forward to the next state.
- Run In-System IBERT (EyeScan) to check the link quality.

© Copyright 2020 Xilinx
• Typically, the issue is caused by link integrity problems, incorrect clock, or reset. On a rare occasion it is caused by the user driving incorrect input to the PCIe IP.
• If cfg_ltssm_state shows 0 all of the time, then check the reset FSM state machine in the init_ctrl module to see which states it is stuck on. This module waits for CPLL/QPLL lock as well as the GT initialization process (reset_done, phy_status, etc. from the GT)
• If a Tandem/PR design is used, check your DONE LED to make sure that the Design is loaded properly and is the correct/expected design if loaded from Flash
• If it is a Gen3 IP (Virtex-7, XT, UltraScale, UltraScale+), then check the cfg_link_train_enable signal and make sure it is set to 1.
• Check if the cfg_ltssm_state on ILA shows a value of 10
• If a value of 10 is seen, see if it is stable or going through recovery (this would indicate a link integrity issue).
• If it is stable at 10, and still no device is seen, it indicates that the cfg completions for the reads did not come back or that the device took too long to program (probable 100ms requirement issue)
  o This can be verified by doing a warm boot to see if the system detects the PCIe device
  o If so, your programming time is too long. Increase your Flash programming frequency or use Bitstream Compression. If all fails, check if you can use the Tandem PROM/PCIe solution.
  o Typically, this is an enumeration failure due to slow FPGA programming (missed 100ms enumeration window), the card is being reprogrammed at runtime, or on rare occasions it is an outcome of a certain event which triggered the BIOS to enter “recovery” mode.
• If the ltssm_state is stuck at 0
  o Check if the cfg_link_training_enable is tied to 1. (Driven by the example design)
  o Check if the GT reset FSM is completed and is back to 00. If yes, check if phy_status_rst pin is connected to the PCIe reset_done
  o Check if GT reset states are stuck
• Gen3 Link issues
  o Check the Link Status2 register to see if Equalization phases were attempted
  o Check the AXI JTAG ltssm information to see where the LTSSM failure happened
  o Try Phase2/3 bypass to see if the link comes up
  o Get a Lecroy trace to see what presets are being asked by the link partner and by the Xilinx receiver
  o Try preset 5, or LPM/DFE or RX auto adaptation modules to see if there is any improvement in the link quality
  o Use the descrambler module enabled in the Xilinx PCIe MAC to check for any lane skew present at Gen3
  o AER correctable error registers might potentially help to pinpoint the link issue (if the Gen3 link is unstable)
  o Use the GT transceiver debug ports to potentially discover any link related issues

GT Signals to Inspect

- Assuming the PHY Wrapper reset FSM has not completed. (LTSSM not alive)
  - Check what state the PHY reset FSM is stuck in.
  - Gen1 capable: CPLLLOCK
  - Gen2/3 capable: CPLLLOCK and QPLL1LOCK
  - Gen4 capable: CPLLLOCK lock and QPLL0LOCK
  - TXYSYNCDONE: GT TX Buffer Bypass successfully complete indicator.
  - If no TXSYNCDONE, measure PCLK and TXOUTCLKPCS frequency.

Reference Clock AC coupling capacitor

- Ensure that the reference clock AC coupling capacitor value is 0.01uF.
- It could lead to start-up issues if cap value is larger than 0.01uF, especially in UltraScale+ devices.
- If having issues upon power-up/configuration, check if the required value is being used.

© Copyright 2020 Xilinx
Reference Clock

- Typically, the PCIe 100 MHz reference clock comes directly from the edge connector to the GT.
- If there is a PLL outside of the Xilinx device, ensure that it meets the PCIe PLL peaking/bandwidth specification at each PCIe rate.
- Some fan-out clock buffers can have a PLL inside them. Ensure that the PLL is bypassed, especially if it does not meet the PCIe specifications.

Debug Questions

If the issue is not resolved, contact Xilinx Support (https://www.xilinx.com/support.html) or post it on the PCIe Forum (https://forums.xilinx.com/t5/PCIe-and-CPM/bd-p/PCIe) with the details of your debug investigation and answers to all relevant questions below.

System Configuration

- Indicate what board you are using: is it a Xilinx Development Board or a Customer Board. If it is a Xilinx development board, please provide the board revision ID.
- Indicate the motherboard description.
  - “What is the link partner? Is it a switch, or a PC?” Who is the manufacturer of that switch or PC? - Which Chipset are you using?
- Did the failure occur in Gen1, Gen2, and/or Gen3?
- Did the failure occur at the RP (Root Port) and/or EP (Endpoint)?
- Describe the channel between the FPGA and the link partner, and estimated channel loss at the desired link speed.
- Is there any passive HW (Interposer for analyzer, retimer) in the path?
  - Connectivity – How is the Xilinx part connected to the system / link partner?
  - (Chip-to-chip on single board, add-in card, backplane, cabling)

Regression

- Did the issue occur in previous Vivado versions too?
- Do other link width configurations show similar behaviour?
- Have you tried with Gen1x1 configuration?
- Do you have a different board that you could try on? If you do, do you see the same issue on that board?
- Have you tried on a different machine?

Clocking
- Did the clock lock?
- What is the clocking architecture? Synchronous or Asynchronous?
- Is SSC enabled?
  - Can you check by disabling SSC?
- What frequency are you using for the reference clock?
- Is the reference clock oscillator driving any other devices / cores / etc?

Design Implementation
- Were there any implementation (synthesis, routing) errors?
- Were there any timing errors?
- Have all of the IP constraints been verified by comparing them to the Example Design XDC file?

Failing Behaviour
- What is the frequency of the error? For example, does it happen immediately or after 1 hour?
- Can the error be cleared? If cleared, does the error come back?
- Is this failure observed on multiple parts?
- Did failure occur immediately after reset?
- Did failure occur immediately, after the first rate change, or after multiple rate changes?
- How long after a successful rate change did it fail?
- Does the issue occur with the Example Design as well or only in your design?

Debug Capability
- Was Lecroy used? If so, provide the Lecroy captures with the details of your analysis of the captures.
- Do you have a high-speed oscilloscope available, to probe clock / power / data lines?
- Do you have a free-running clock available to the FPGA? (A clock that is separate from the PCIe reference clock, and not tied to the PCIe reset)
- Do you have the ability to insert a clean clock in place of on-board reference clock?
- Have you captured the LTSSM graph by enabling the JTAG Debugger feature in the GUI?
- Have you run Eye Scan by enabling the ‘In-system IBERT’ feature in the GUI?

SI Debug Info
- Has it been confirmed whether Clock Jitter, Power and Noise are within the spec?
  - Power integrity measurements
  - REFCLK jitter measurements
  - Channel loss data
  - Eye Scan plots
  - Please confirm whether it is DFE/LPM or AutoRxEq

Known Issues

UltraScale+ PCI Express Integrated Block - Release Notes and Known Issue
https://www.xilinx.com/support/answers/65751.html

UltraScale FPGA Gen3 Integrated Block for PCI Express - Release Notes and Known Issues
© Copyright 2020 Xilinx
Appendix

store_ltssm

- Use to capture continuous transition of the LTSSM signal
- Useful for debug in the following scenarios:
  - The link is unstable and LTSSM is continuously transitioning
  - To check the behavior of LTSSM. For example if it is repeatedly falling out to recovery, looping between detect-polling-configuration etc.
  - Issue with Gen3 link training; check from which equalization phase it falls out to Recovery.Speed
  - Check if phase 2/3 bypass is enabled or not
  - Check if the link is going into power management states
  - Check if LTSSM goes to Hot Reset

store_ltssm Code

- Added by default in UltraScale. Not available by default in UltraScale+
- Ok to copy/paste the code from UltraScale to UltraScale+ without any other modifications

cfg_ltssm / Capture Control

- Make sure to add cfg_ltssm
- Not added by default
- Not used if cfg_ltssm is not added
- Capture Control must be enabled
Basic Capture Setup

- Trigger on an LTSSM state.
- After the trigger, the data will be captured only when the signal selected in the Basic Capture Setup window goes to its Compare Value.
Waveform capture with store_ltsm enabled

<table>
<thead>
<tr>
<th>00</th>
<th>02</th>
<th>04</th>
<th>06</th>
<th>08</th>
<th>0a</th>
<th>0c</th>
<th>0e</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
<th>22</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0000</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0000</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0000</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0000</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

© Copyright 2020 Xilinx
Waveform capture without store_ltssm

IP Directory and Files

Global Vs OOC

- To add mark_debug in the wrapper files in Windows explorer, make sure to select ‘Global’ when generating the output products. If OOC is selected, the changes in the wrapper files will not be picked up during synthesis and implementation.
Comparing Attributes

Step 1: Use report_property to list the attributes on GT#_CHANNEL and GT#_COMMON

Example: write a file that contains all of the properties and values for the common and channel blocks

- For a design loaded in Vivado:
  - report_property –file gt_common_attibs.txt [get_cells GT_COMMON_CELLNAME]
  - report_property –file gt_channel_attibs.txt [get_cells GT_CHANNEL_CELLNAME]

- For a design using Vivado Hardware Manager:
  - report_property –file gt_channel_attibs.txt [get_hw_sio_gts GT_CHANNEL_CELLNAME]
  - report_property –file gt_common_attibs.txt [get_hw_sio_commons GT_COMMON_CELLNAME]

Step 2: To compare attributes

Linux:

Diff [file1] [file2]

Windows Command Line:

fc [file1] [file2]
In-System IBERT Free Running Clock

- When ‘In-system IBERT’ is enabled, the core top level wrapper consists of a free_run_clock input for running ‘In-System IBERT’.

```c
// PCIe Core Top Level Wrapper
//
// Core Top Level Wrapper
//
pci_uscale_plus_0 pci4_uscale_plus_0_i {
  // 10 Ports
  //
  .cfg_mgr_debug_access ("b0",
  .cfg_mgr_function_number (8'b0),
  .cfg_vf_pll_func_num (cfg_vf_pll_func_num),

  .free_run_clock (free_run_clock),
}
```

Define pin constraints for a free running clock:

```c
@ set_property IOSTANDARD LVCMOS18 [get_ports sys_rst_n]
@ set_property IOI [get_package_pins -filter {FINFUNC =~ *_PERSTN0_65}] [get_ports sys_rst_n]
@ set_property PACKAGE_PIN A11 [get_ports sys_rst_n]
@ create_clock -name free_run_clk -period 10 [get_ports free_run_clock_p_in]
@ #
@ set_property IOI [get_package_pins -cf_objects {get_bels [get_sites -filter {NAME =~ "COMMON"]}
@ set_property IOI [get_package_pins -cf_objects {get_bels [get_sites -filter {NAME =~ "COMMON"]}
@ #
@ #
```

- The top-level file of the generated example design defines the differential input clock:
For a Single-Ended clock:
Appendix B

Trigger State Machine Language Description

The trigger state machine language is used to describe complex trigger conditions that map to the advanced trigger logic of the ILA debug core. The trigger state machine has the following features:

- Up to 16 states.
Use Case 1:
Use Case 2:

Use Case 3:

© Copyright 2020 Xilinx
Use Case 4:

```
if (u_ila_0_gt欤lock == 1'b1) && (u_ila_0_gt_rasocklock == 1'b1) && (u_ila_0_gt_rasocklock_out == 1'b1) && (u_ila_0_gt_rasocklock == 1'b1) && (u_ila_0_gt_rasocklock_out == 1'b1) && (u_ila_0_gt_rasocklock == 1'b1) && (u_ila_0_gt_rasocklock_out == 1'b1) && (cfg_bteam_vlan == 1'b0)) then begin
  if (cfg_bteam_vlan == 1'b0) then
    case (state)
      state0:
        next
end
```

© Copyright 2020 Xilinx
Note: Refer to the latest version of (UG576) found on Xilinx.com.

### Pin Description and Design Guidelines

#### Table 5-1: GTH Transceiver Quad Pin Descriptions

<table>
<thead>
<tr>
<th>Pins</th>
<th>Dir</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGTREFCLK0P</td>
<td>In/Out (Pad)</td>
<td>Configured as either reference clock input pins or RX recovered clock output pins for the Quad.</td>
</tr>
<tr>
<td>MGTREFCLK0N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MGTREFCLK1P</td>
<td>In/Out (Pad)</td>
<td>Configured as either reference clock input pins or RX recovered clock output pins for the Quad.</td>
</tr>
<tr>
<td>MGTREFCLK1N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MGTHRXP[3:0]/MGTHRXN[3:0]</td>
<td>In (Pad)</td>
<td>RXP and RXN are the differential input pairs for each of the receivers in the GTH transceiver Quad.</td>
</tr>
<tr>
<td>MGHTXP[3:0]/MGHTXN[3:0]</td>
<td>Out (Pad)</td>
<td>TXP and TXN are the differential output pairs for each of the transmitters in the GTH transceiver Quad.</td>
</tr>
<tr>
<td>MGTAVTTRCAL</td>
<td>In (Pad)</td>
<td>Bias current supply for the termination resistor calibration circuit. See Termination Resistor Calibration Circuit.</td>
</tr>
<tr>
<td>MGTRREF</td>
<td>In (Pad)</td>
<td>Calibration resistor input pin for the termination resistor calibration circuit. See Termination Resistor Calibration Circuit.</td>
</tr>
</tbody>
</table>

#### Table 5-1: GTH Transceiver Quad Pin Descriptions (Cont’d)

<table>
<thead>
<tr>
<th>Pins</th>
<th>Dir</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGTAVCC</td>
<td>In (Pad)</td>
<td>MGTAVCC is the analog supply for the internal analog circuits of the GTH transceiver Quad tile. This includes the analog circuits for the PLLs, transmitters, and receivers. Most packages have multiple groups of power supply connections in the package for MGTAVCC. Refer to the package pin definitions to identify in which power supply group a specific GTH transceiver Quad is located. For UltraScale+ FPGAs, the nominal voltage is 1.0 VDC. For UltraScale+ FPGAs, the nominal voltage is 0.9 VDC.</td>
</tr>
<tr>
<td>MGTAVTT</td>
<td>In (Pad)</td>
<td>MGTAVTT is the analog supply for the Transmitter and Receiver termination circuits of the GTH transceiver Quad tile. Most packages have multiple groups of power supply connections in the package for MGTAVTT. Refer to the package pin definitions to identify in which power supply group a specific GTH transceiver Quad is located. The nominal voltage is 1.2 VDC.</td>
</tr>
<tr>
<td>MGTVCCAUX</td>
<td>In (Pad)</td>
<td>MGTVCCAUX is the auxiliary analog QPLL voltage supply for the transceivers. Most packages have multiple groups of power supply connections in the package for MGTVCCAUX. Refer to the package pin definitions to identify in which power supply group a specific GTH transceiver Quad is located. The nominal voltage is 1.8 VDC.</td>
</tr>
</tbody>
</table>
GTH Transceiver Reference Clock Checklist (UG576)

- Provide AC coupling between the oscillator output pins and the dedicated GTH transceiver Quad clock input pins.
- Ensure that the differential voltage swing of the reference clock is within the range specified in the device data sheets.
- Meet or exceed the reference clock characteristics as specified in the device data sheets.
- Meet or exceed the reference clock characteristics as specified in the standard for which the GTH transceiver provides physical layer support.
- Fulfil the oscillator vendor’s requirement regarding power supply, board layout, and noise specification.
- Provide a dedicated point-to-point connection between the oscillator and GTH transceiver Quad clock input pins.
- Keep impedance discontinuities on the differential transmission lines to a minimum (impedance discontinuities generate jitter).

Power Supply and Filtering

- Noise on the GTH transceiver analog power supplies can cause degradation in the performance of the transceivers. The most likely form of degradation is an increase in jitter at the output of the GTH transmitter and reduced jitter tolerance in the receiver. Sources of power supply noise are:
  - Power supply regulator noise
  - Power distribution network
  - Coupling from other circuits
- Each of these noise sources must be considered in the design and implementation of the GTH transceiver analog power supplies. The total peak-to-peak noise as measured at the input pin of the UltraScale device should not exceed 10 mVpk-pk.
### PCB Design Checklist

<table>
<thead>
<tr>
<th>Pins</th>
<th>Recommendations</th>
</tr>
</thead>
</table>
| MGTREFCLKXP  | When configured as an input:  
  - Use AC coupling capacitors for connection to oscillator  
  - For AC coupling capacitors, see Reference Clock Interface, page 322.  
  - Reference clock oscillator output must comply with the minimum and maximum input amplitude requirements for these input pins. See the device data sheets (Ref 4).  
  When configured as an output:  
  - Use AC coupling capacitors for connection to receiving device.  
  - For AC coupling capacitors use 0.01 µF.  
  - For output signal characteristics, see Xilinx UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics (DS892) (Ref 6).  
  If reference pins are not used, leave the associated pin pair unconnected. However, if the IBUFDS/GTE2/4 is instantiated in the design but not used, the associated pin pair should be connected to GND. |
| MGTREFCLKON  |                                                                                                                                                  |
| MGTREFCLKEP  |                                                                                                                                                  |
| MGTREFCLKIN  |                                                                                                                                                  |
| MGTTHRXP[3:0]/MGTTHXN[3:0] | Use AC coupling capacitors for connection to transmitter. The recommended value for AC coupling capacitors is 100 nF.  
  - Receiver data traces should be provided enough clearance to eliminate crosstalk from adjacent signals.  
  - If a receiver will never be used under any conditions, connect the associated pin pair to GND.  
  - If a receiver is not used and not connected to anything under some conditions but might be connected to something and used under other conditions, then for the conditions when the receiver is unused, either do not instantiate the GTH transmitter in the FPGA design or if the GTH transmitter is instantiated, set RXPD[1:0] to 0’b11.  
  - See RX Analog Front End, page 37L. |
| MGTTHTXP[3:0]/MGTTHTXN[3:0] | Transmitter should be AC coupled to the receiver. The recommended value for the AC coupling capacitors is 100 nF.  
  - Transmitter data traces should be provided enough clearance to eliminate crosstalk from adjacent signals.  
  - If a transmitter is not used, leave the associated pin pair unconnected. |
RX Termination for Receiver Detection (UG576)

- Dynamic switching of the receiver termination mode is required when the remote transmitter is using the falling edge for the receiver detection or in a situation where the implementation of the remote transmitter is unknown. The steps to perform dynamic switching of the receiver termination mode are:
  - When the receiver detection is performed by the remote transmitter, set the receiver termination mode to MGTAVTT (RX_CM_SEL[1:0] = 'b00) via a DRP operation.
  - After receiver detection is completed, set the receiver termination mode to programmable mode (RX_CM_SEL [1:0] = 'b11) via a DRP operation.
- The dynamic switching of the termination mode is not required when the transmitter is using the rising edge for the receiver detection. In this case, set the receiver termination mode to programmable (RX_CM_SEL[1:0] = 2'b11). The PCIe IP core can be customized to select different receiver termination schemes.

**LTSSM**

**Note:** Refer to (PG213) for UltraScale+ and (PG156) for UltraScale Integrated Block for PCI Express.

<table>
<thead>
<tr>
<th>Port</th>
<th>Direction</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config_ltssm_state</td>
<td>Output</td>
<td>6</td>
<td>LTSSM State. Shows the current LTSSM state:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>00: Detect.Quiet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>01: Detect.Active</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>02: Polling.Active</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03: Polling.Compliance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>04: Polling.Configuration</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>05: Configuration.Linkwidth.Start</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>06: Configuration.Linkwidth.Accept</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>07: Configuration.Lanenum.Accept</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>08: Configuration.Lanenum.Wait</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>09: Configuration.Complete</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0A: Configuration.Idle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0B: Recovery.RcvrLock</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0C: Recovery.Speed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0D: Recovery.RcvrCfg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0E: Recovery.Idle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10: L0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11: Rx_L0s.Entry</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12: Rx_L0s.Idle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>13: Rx_L0s.FTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>14: Tx_L0s.Entry</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15: Tx_L0s.Idle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>16: Tx_L0s.FTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>17: L1.Entry</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>18: L1.Idle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>19: L2.Idle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1A: L2.TransmitWake</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20: Disabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>21: Loopback_Entry_Master</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>22: Loopback_Active_Master</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>23: Loopback_Exit_Master</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>24: Loopback_Entry_Slave</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>25: Loopback_Active_Slave</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>26: Loopback_Exit_Slave</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>27: Hot_Reset</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>28: Recovery_Equalization_Phase0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>29: Recovery_Equalization_Phase1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2a: Recovery_Equalization_Phase2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2b: Recovery_Equalization_Phase3</td>
</tr>
</tbody>
</table>

Figure 24 LTSSM states with UltraScale Integrated Block for PCI express
Training Sequence 1 and 2 (TS1 and TS2)

- TS1 and TS2 Ordered-Sets are each comprised of up to 16 symbols.
- Symbol 0 is COM which is the K28.5 character. The receiver uses this character to achieve Bit Lock and Symbol Lock.
Electrical Idle Ordered Set
- The Electrical Idle Ordered-Set consists of four symbols: COM-IDL-IDL-IDL = BC, 7C, 7C, 7C.
- The transmitter sends out the electrical idle ordered set before going into the electrical idle state.
- After receiving the electrical idle order set, the link partner prepares the link to go to the electrical idle state.

SKP Ordered Set
- COM, SKP, SKP, SKP = BC, 1C, 1C, 1C
- Transmitted at regular intervals from the transmitter to the receiver.
- Used for Clock Tolerance Compensation

FTS Ordered Set
- COM, FTS, FTS, FTS = BC, 3C, 3C, 3C
- A transmitter sends the defined number of FTS before transitioning to L0 from L0s state
- The number of FTS is agreed during link training and initialization

Revision History

03/31/2020 - Initial release