AN 833: Intel® Stratix 10® GX 16-Lane RX JESD204B-ADC12DJ3200 Interoperability Reference Design
Intel® Stratix® 10 GX 16-Lane RX JESD204B-ADC12DJ3200 Interoperability
Reference Design.......................................................................................................3
Reference Design Block Diagram...................................................................................3
Key Features..................................................................................................................4
Submodules in the Top-Level HDL Files.................................................................5
Requirements...............................................................................................................5
  Hardware Requirements..........................................................................................5
  Software Requirements.........................................................................................6
Reference Design Walkthrough.....................................................................................7
  Hardware Set Up....................................................................................................7
  Running the Reference Design..............................................................................8
  Viewing the Results..............................................................................................14
  Rebuilding Design................................................................................................20
Procedures in the main.tcl System Console Script......................................................20
Document Revision History for AN 833: Intel Stratix 10 GX 16-Lane RX JESD204B-
  ADC12DJ3200 Interoperability Reference Design....................................................21
**Intel® Stratix® 10 GX 16-Lane RX JESD204B-ADC12DJ3200 Interoperability Reference Design**

This application note showcases the synchronization of two ×8-lane Intel® Stratix® 10 FPGA JESD204B RX IP cores to interoperate with the 12-bit, 16-lane Texas Instruments (TI) ADC12DJ3200 Evaluation Module (EVM) running at 6.4 gigabits per second (Gbps) per lane connected through FMC+ port A connector.

TI wideband ADC12DJ3200 device is the 12-bit converter which is capable of operating at up to 3.2 gigasamples per second (Gsps) in dual channel mode or 6.4 Gsps in an interleaved single channel mode. This design is programmed to run at the fastest sample rate of 6.4 Gsps in single channel mode, where this mode effectively interleaves the two analog-to-digital converter (ADC) channels together to form a single channel ADC at twice the sampling rate.

Applications such as high-density phased array radar, satellite communications, 5G systems, and medical imaging are driving the need for increased data throughput, higher bandwidth, and lower power. Small package size and lower PCB cost are preferred in these applications.

**Related Links**
- Reference Design Files for AN833
- AN 804: Implementing ADC-Intel Stratix 10 Multi-Link Design with JESD204B RX IP Core
  Provides more information about synchronized ADC-Intel Stratix 10 multi-link implementation.

**Reference Design Block Diagram**

**Figure 1. Reference Design Simplified Block Diagram**

- To IP Core AVS Clock, Reset Sequencer, Transceiver PHY Reset Controller, Frequency Checker, and other system components.
- To IP Cores, RX Transport Layer, Deterministic Latency Measurement.
- To RX Transport Layer, Pattern Checker, Application Layer.

**Notes:**
1. Soft Core Clock, Assert Sequencer, Transceiver IDF Reset Controller, Frequency Checker, and other system components.
2. Soft Core, RX Transport Layer, Deterministic Latency Measurement.
3. S. RX Transport Layer, Pattern Checker, Application Layer.
Key Features

This reference design provides the following key features:

- Two 8-lane simplex RX JESD204B IP cores interoperate with the ADC12DJ3200 EVM through the Intel Stratix 10 Transceiver Signal Integrity Development Kit FMC+ port A running at 6.4 Gbps per lane. The JESD204B IP core has the following parameters:
  - L=8
  - M=8
  - F=8
  - S=5
  - N=12
  - N'=12
  - K=4
  - HD=0
  - SCR=1
  - CS=CF=0

- The LMK04828 clock generator on the ADC12DJ3200 EVM provides 160 MHz for I/O PLL core reference clock and 160 MHz for transceiver CDR reference clock. These clocks are transmitted from a single clock chip through the FMC+ port A to the core reference clock pin and the dedicated transceiver pin at the FPGA. The LMK04828 also provides a clock to LMX2582, where the LMX2582 synthesizer generates a 3,200 MHz ADC sampling clock.

- The I/O PLL on the FPGA generates link clock and frame clock. The IP cores, RX transport layers, and deterministic latency measurement block are sourced from link clock. The frame clock is supplied to RX transport layers, test pattern checkers, and any application layer.

- The LMK04828 clock chip generates continuous SYSREF signal for the RX JESD204B IP cores at the FPGA and the ADC12DJ3200 device.

- The deterministic latency measurement block measures the number of link clock counts between the start of combined SYNC_N deassertion output from the two JESD204B IP cores to the first user data output to ensure latency is deterministic.

- Frequency Checker monitors to ensure the I/O PLL core reference clock and transceiver CDR reference clock from the EVM clock generator and RX recovered clock frequency generated from CDR are correct.

- The main.tcl script (located at the <project directory>/system_console directory) generated from the JESD204B Example Design (LMF=888, 6.144 Gbps) (Stratix 10 only) preset is enhanced to support multi-link design. Refer to the Procedures in the main.tcl System Console Script table for details about the .tcl procedures.

- A Signal Tap II file is included in this design for debug assistance, such as monitoring the short transport pattern at RX transport layers, checking correct octet ramp pattern at the output of the JESD204B IP cores, and checking the output counter to ensure design is deterministic from one power cycle to another.
Related Links

- Procedures in the main.tcl System Console Script on page 20
- Compiling the Project to Include Signal Tap II File on page 8
  Provides the steps to enable Signal Tap II in the Intel Quartus® Prime project settings.

Submodules in the Top-Level HDL Files

The top-level HDL files of this reference design instantiate the following submodules:

- Top level Platform Designer system
  - JESD204B subsystem includes reset sequencer, two ×8-lane RX JESD204B IP cores, transceiver PHY reset controller, and Avalon® Memory-Mapped (Avalon-MM) pipeline bridge.
  - JTAG-to-Avalon master bridge for the System Console.
  - Link clock and frame clock generated by the core PLL.
  - Serial peripheral interface (SPI) master—Optional component in this design. You can use this component in your custom design if needed.
- RX transport layers for link 0 and link 1
- Test pattern checker
  Note: This test pattern checker is an optional module for this design example and is not suitable to test the ramp pattern transmitted from the ADC12DJ3200 device.
- Deterministic latency measurement
- Frequency checker

Related Links

Intel FPGA JESD204B Design Example User Guide for Intel Stratix 10 Devices
Provides the functional description of each Platform Designer system component and transport layers for RX path.

Requirements

Hardware Requirements

The reference design requires the following hardware tools:

- Intel Stratix 10 Transceiver Signal Integrity Development Kit (1SG280HU2F50E2VGS1).
- TI ADC12DJ3200 Evaluation Module—This design is tested in TI ADC12DJ3200 EVM Rev E3.
- Micro-USB cable and power adaptor—Part of Intel Stratix 10 Transceiver Signal Integrity Development Kit accessories.
- Mini-USB cable and power cable—Part of ADC12DJ3200 EVM accessories.
Software Requirements

The reference design requires the following software:

- TI ADC12DJXXX00 Configuration GUI Software that is compatible with ADC12DJ3200 EVM Rev E3.
- Future Technology Devices International (FTDI) Combined Driver Model (CDM) driver package—D2XX driver.

Related Links

- D2XX Drivers, Future Technology Devices International Ltd. page
  Provides the setup executable D2XX Driver (CDM21228_Setup.exe). Follow the installation instructions to install the driver.
- ADC12DJ3200 12-Bit, Dual 3.2-GSPS or Single 6.4-GSPS, RF-Sampling ADC Evaluation Module, Texas Instruments page
  Provides the ADC12DJ3200 Configuration GUI Software for the latest version of the ADC12DJ3200 EVM.
Reference Design Walkthrough

Hardware Set Up

Figure 2. Intel Stratix 10 Transceiver Signal Integrity Development Kit and ADC12DJ3200 EVM Set Up
1. Set the following MSEL DIP switches to the "1" position to enable **JTAG Only Mode**.
   - SW11: MSEL0 = 1
   - SW10: MSEL1 = 1
   - SW10: MSEL2 = 1
2. Set the SW3-2 to the ON position to disable MAX® V devices in the JTAG chain.
3. Set the remaining DIP switches to the factory default settings. Refer to the **Factory Default Switch Settings** table in the *Intel Stratix 10 GX Transceiver Signal Integrity Development Kit User Guide*.
4. Slot the ADC12DJ3200 EVM module into FMC+ port A on the Intel Stratix 10 Transceiver Signal Integrity Development Kit.
   Ensure the board-to-board FMC+ connection is secure.
5. Connect the micro-USB cable and power adaptor to the Intel Stratix 10 Transceiver Signal Integrity Development Kit.
6. Connect the mini-USB cable and power cable to the ADC12DJ3200 EVM module.

**Related Links**

- **Factory Default Switch and Jumper Settings, Intel Stratix 10 GX Transceiver Signal Integrity Development Kit User Guide**
  Provides the factory switch settings for the Intel Stratix 10 GX transceiver signal integrity development kit.

**Running the Reference Design**

To run the reference design, follow these steps:
1. Compile the project to include Signal Tap II file.
2. Set up the TI ADC12DJxx00 GUI software and board.
3. Configure the FPGA.
4. Check the basic operation.
5. Execute the Tcl Script File (`.tcl`) code and initialize the JESD204B links.
6. Check for deterministic latency.

**Compiling the Project to Include Signal Tap II File**

The reference design provided does not enable Signal Tap II Logic Analyzer in the Intel Quartus Prime project.
To compile the design that include the Signal Tap II file (.stp) into the Intel Quartus Prime project, follow these steps:
1. To test the reference design targeted for Intel Stratix 10 GX device, download the reference design file to your local project directory.
2. Launch the Intel Quartus Prime Pro Edition software.
3. To prepare the design template in the Intel Quartus Prime Pro Edition software GUI, click File ➤ Open and change the file type to the Quartus Prime Design Template File (*.par). Browse to the <project>.par file and click Ok.
4. Turn on Assignments ➤ Settings ➤ Category ➤ Enable Signal Tap II Logic Analyzer.
5. Browse to the stp1.stp file located at the /stp directory and click Ok.
6. To compile the project, select Processing ➤ Start Compilation.

Setting Up the TI ADC12DJxx00 GUI Software and Board

To set up the TI ADC12DJxx00 GUI software and board, follow these steps:
1. Install the ADC12DJ3200 GUI Configuration software.
2. Extract the files from the .zip file.
3. Run the executable file (setup.exe) and follow the instructions. Use the default installation location (C:\Program Files (x86)\Texas Instruments \ADC12DJxx00 GUI\).
4. Connect the USB cable to PC and turn on the power supply for the ADC12DJ3200 EVM. Launch the ADC12DJ3200 GUI.exe available from your installation directory.
   The USB status becomes green if the ADC12DJ3200 EVM card is detected.

Figure 3. ADC12DJxx00 GUI EVM Tab
5. Replace the original files with the following modified version of the .cfg files using the same file names.
   - ADC12DJxx00_JMODE1.cfg
   - LMK04828_JMODE1.cfg

The original files are in this directory: C:\Program Files (x86)\Texas Instruments\ADC12DJxx00_GUI\Configuration File. Extract the ConfigurationFile.zip file to get the modified version of the .cfg files. You can find the ConfigurationFile.zip file after installing the <project>.par file. Back up the two original files before replacing the two .cfg files provided in the reference design into the same directory.

6. Select Clock Source ➤ On-board. This is the default setting.
7. Select On-board Fclk Selection ➤ Fclk = 3200 MHz.
8. Select Decimation and Serial Data Mode ➤ JMODE1.
9. Click Program Clocks and ADC. Clicking the Program Clocks and ADC overwrites any previous device register settings.
10. Verify the settings for ADC12DJxx00 LMK04828 ➤ Clock Outputs tab as in the following figure.

Figure 4. ADC12DJxx00 GUI LMK04828 Tab

Configuring the FPGA

To configure the FPGA, follow these steps:
1. Before configuring the FPGA, ensure the following:
   a. The Intel FPGA Download Cable II driver is installed on the host computer.
b. The Intel Stratix 10 Transceiver Signal Integrity Development Kit is powered on.

c. No other running application is using the JTAG chain.

2. In the Intel Quartus Prime Programmer, select **Hardware Setup ➤ Stratix 10H SI Dev Kit[USB-1]**.

3. Click **Auto Detect** to display the devices in the JTAG chain. Select device 1SG280HU2S1.

4. Right click and select **Change File**. Choose the appropriate SRAM Object File (.sof) from the `<project directory>/output_files` directory. Click **Open**.

5. Turn on **Program/Configure** for the .sof file.

6. Click **Start** to program the image into the FPGA.

---

**Figure 5. Intel Quartus Prime Programmer**

---

**Checking the Basic Operation**

To check on the basic operation, follow these steps:

1. Make sure the I/O PLL core reference clock (`refclk_core`), transceiver clock data recovery (CDR) reference clock (`refclk_xcvr`), and receiver (RX) recovered clock (`rxphy_clk`) are 160 MHz.

   The LMK04828 clock generator from the ADC12DJ3200 EVM module provides the reference clock to I/O PLL and transceiver CDR. A frequency checker module is added to this reference design to verify the I/O PLL core reference clock, transceiver CDR reference clock, and RX recovered clock frequency are correct.

   You can view the measured clock frequency in the SignalTap II file by clicking at the `freq_chk` instance as indicated in the following figure. The frequency values of all measured clocks are displayed in Hz after running analysis or autorun analysis.
2. Verify the RX PHY status by monitoring the status of
   \texttt{rx\_is\_locked\_to\_data[7..0]}, \texttt{rx\_analog\_reset[7:0]},
   \texttt{rx\_digital\_reset[7:0]}, and \texttt{rx\_cal\_busy[7:0]} signals for link 0 and link 1.
   These signals are available under \texttt{rx\_phy} instance in the Signal Tap II file.

<table>
<thead>
<tr>
<th>Signal</th>
<th>Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{rx_is_locked_to_data}</td>
<td>1</td>
</tr>
<tr>
<td>\texttt{rx_analog_reset}</td>
<td>0</td>
</tr>
<tr>
<td>\texttt{rx_digital_reset}</td>
<td>0</td>
</tr>
<tr>
<td>\texttt{rx_cal_busy}</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 7. RX PHY Status**

**Executing the .tcl Code and Initializing the JESD204B Links**

To execute the .tcl code and initialize the JESD204B links, follow these steps:
1. Launch the System Console in the Intel Quartus Prime software, click Tools ➤ System Debugging Tools ➤ System Console.

2. In the Tcl Console, type the following:
   a. `cd` — To change the directory to the working directory that contains the main.tcl script (located at <project directory>/system_console)
   b. `source main.tcl` — To execute the main.tcl script.
   c. `start_basic_test` — To execute the start_basic_test procedure.

Related Links
Procedures in the main.tcl System Console Script on page 20

Checking the Deterministic Latency

To check for deterministic latency, follow these steps:

1. Restart the converter and reprogram the clocks and ADC.
2. Restart and reconfigure the FPGA.
3. Execute the .tcl code to initialize the JESD204B links.
4. Read the RX Buffer Delay (RBD) count by typing the `read_rx_status0` procedure in the Tcl Console and record the value. The RBD count is from the `csr_rbd_count` field in the `rx_status0[10:3]` register (at offset 0x80).
5. Measure and record the number of link counts between the start of combined `SYNC_N` deassertion output from the two JESD204B IP cores to the first user data output, which is the assertion of the `jesd204_rx_link_valid` signal. Ensure the latency is constant for every converter and FPGA power cycle.
6. Repeat step 1 to step 5 for a few times.

Example of the System Console output after executing the `read_rx_status0` procedure:

```bash
% read_rx_status0
master_list_length = 1
RX Status0 (Link 0)= 0x00000009
RX Status0 (Link 1)= 0x00000009
```

RX Status0 = 0x00000009 indicates the following conditions:
- `bit[0]=1`, JESD204B link is out of synchronization (`SYNC_n` deasserted).
- `bit[3]=1`, `csr_rbd_count = 1`

For detailed description of the `csr_rbd_count` field of the `rx_status0[10:3]` register (at offset 0x80), refer to the Addressmap Information for 'altera_jesd204_rx_regmap'.

Related Links
- Setting Up the TI ADC12DJxx00 GUI Software and Board on page 9
  Provides the steps to reprogram the clocks and ADC.
- Configuring the FPGA on page 10
- Executing the .tcl Code and Initializing the JESD204B Links on page 12
• **Addressmap Information for ‘altera_jesd204_rx_regmap’**
  Provides the detailed description of the csr_rbd_count field of the rx_status0[10:3] register (at offset 0x80).

**Viewing the Results**

**Link Up Test Results**

**Figure 8. Successful Link Up Indicated in the System Console**

This figure illustrates the expected output from the **Tcl Console** of the **System Console** if the link up is successful.

```tcl
# start_basic_test
---- Read Initial Status----
master_list_length = 1
PIO Status = 0x00000000f
master_list_length = 1
Reset Done!
Info: Closed JTAG Master Service

Waiting for reset seq active low...
Reset seq active low Link and frame reset held
Waiting for reset seq active low...
Reset seq active low master_list_length = 1
 Setting tx_test & rx_test registers...
 Set test mode to keep test pattern
 Waiting for reset seq active low...
Reset seq active low Link and frame reset released
Waiting for reset seq active low...
Reset seq active low master_list_length = 1
RX syncyster_referenceeward before write = 0x00000000
RX syncyster_referenceeward before write = 0x00000000
RX syncyster_referenceeward after write = 0x00000002
RX syncyster_referenceeward after write = 0x00000002

SYSTERF Continuous Detection!
Info: Closed JTAG Master Service

master_list_length = 1
Error status registers cleared for link 0 & link 1
RX Error0 Status (Link 0)= 0x00000000
RX Error1 Status (Link 0)= 0x00000000
RX Error0 Status (Link 1)= 0x00000000
RX Error1 Status (Link 1)= 0x00000000
master_list_length = 1
Status: 0xb (Masked value)
Info: Bit 0 - Core PLL Locked
Info: Bit 1 - RX MVR Ready (Link 0)
Info: Bit 2 - Rx Link Error (Link 0)
Info: Bit 3 - RX MVR Ready (Link 1)
Info: Bit 4 - Rx Link Error (Link 1)
RX Status0 (Link 0): 0x0 (Masked value)
RX Status1 (Link 1): 0x0 (Masked value)
Info: Bit 0 - SYNC_R
RX_TEST (Link 0) : PASS
RX_TEST (Link 1) : PASS
```
When the link up is successful, you should observe the following conditions:

- USER_LED0–USER_LED3 (D12–D15) illuminate
- USER_LED4 (D16) turned off

**Figure 9. On-board User LEDs**

![Intel® Stratix® 10 Transceiver Signal Integrity Development Kit User LEDs](image)

<table>
<thead>
<tr>
<th>On-board User LED</th>
<th>Signal</th>
<th>Indication when LED Illuminates</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED D12</td>
<td>rx_frame_rst_n</td>
<td>The transport layers and test pattern checkers are out of reset.</td>
</tr>
<tr>
<td>LED D13</td>
<td>rx_link_rst_n</td>
<td>The IP cores, transport layers, and deterministic latency module are out of reset.</td>
</tr>
<tr>
<td>LED D14</td>
<td>alldev_laneAligned</td>
<td>All lanes are aligned for two JESD204B IP cores receiver.</td>
</tr>
<tr>
<td>LED D15</td>
<td>rx_dev_sync_n_out</td>
<td>The receivers at link 0 and link 1 have successful received K28.5 characters.</td>
</tr>
<tr>
<td>LED D16</td>
<td>rx_link_error</td>
<td>The interrupt is triggered at any of the JESD204B RX IP cores.</td>
</tr>
</tbody>
</table>

**Table 2. On-board User LEDs Indication**

**Short Transport Layer and Ramp Test Mode Test Results**

**Verifying Short Transport Test Patterns at Receiver Transport Layer**

JESD204B block in the ADC12DJ3200 device defines the short transport test pattern for N’=12 test mode to verify that the transport layer test patterns in transmitter and receiver are operating correctly.

To verify the short transport test patterns at the receiver transport layers of the FPGA, follow these steps:

1. In the Intel Quartus Prime Pro Edition software, click **Tools ➤ Signal Tap II Logic Analyzer**.
2. Check the JTAG chain configuration. Select the hardware and device correctly.
3. In the Instance Manager, click **rx_tprt ➤ run Analysis/Autorun Analysis**.

The output of RX transport layers for both links are grouped in 16 groups in the Signal Tap II waveform. Each group has 60 bits. You may further split the 60-bit buses into 12-bit buses as indicated in the following figure to match with the following table.
Splitted each group of the RX transport layer output into 12-bit bus.

Ensure both link 0 and link 1 of receiver output transport layers are matched with the following table.

Table 3. Short Transport Test Patterns for N'=12 Modes (Length = 1 Frame)

<table>
<thead>
<tr>
<th>Link</th>
<th>Octet</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Nibble</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>8 lanes of link 0</td>
<td>DA0</td>
<td>0xF01</td>
<td>0xF02</td>
<td>0xF03</td>
<td>0xF04</td>
<td>0xF05</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DA1</td>
<td>0xE11</td>
<td>0xE12</td>
<td>0xE13</td>
<td>0xE14</td>
<td>0xE15</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DA2</td>
<td>0xD21</td>
<td>0xD22</td>
<td>0xD23</td>
<td>0xD24</td>
<td>0xD25</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DA3</td>
<td>0xC31</td>
<td>0xC32</td>
<td>0xC33</td>
<td>0xC34</td>
<td>0xC35</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DA4</td>
<td>0xB41</td>
<td>0xB42</td>
<td>0xB43</td>
<td>0xB44</td>
<td>0xB45</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DA5</td>
<td>0xA51</td>
<td>0xA52</td>
<td>0xA53</td>
<td>0xA54</td>
<td>0xA55</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DA6</td>
<td>0x961</td>
<td>0x962</td>
<td>0x963</td>
<td>0x964</td>
<td>0x965</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DA7</td>
<td>0x871</td>
<td>0x872</td>
<td>0x873</td>
<td>0x874</td>
<td>0x875</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 lanes of link 1</td>
<td>DB0</td>
<td>0xF01</td>
<td>0xF02</td>
<td>0xF03</td>
<td>0xF04</td>
<td>0xF05</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DB1</td>
<td>0xE11</td>
<td>0xE12</td>
<td>0xE13</td>
<td>0xE14</td>
<td>0xE15</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DB2</td>
<td>0xD21</td>
<td>0xD22</td>
<td>0xD23</td>
<td>0xD24</td>
<td>0xD25</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DB3</td>
<td>0xC31</td>
<td>0xC32</td>
<td>0xC33</td>
<td>0xC34</td>
<td>0xC35</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DB4</td>
<td>0xB41</td>
<td>0xB42</td>
<td>0xB43</td>
<td>0xB44</td>
<td>0xB45</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DB5</td>
<td>0xA51</td>
<td>0xA52</td>
<td>0xA53</td>
<td>0xA54</td>
<td>0xA55</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DB6</td>
<td>0x961</td>
<td>0x962</td>
<td>0x963</td>
<td>0x964</td>
<td>0x965</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DB7</td>
<td>0x871</td>
<td>0x872</td>
<td>0x873</td>
<td>0x874</td>
<td>0x875</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Verifying Ramp Pattern at Output Data of DLL to the Input Receiver Transport Layer

You can run the ramp test mode after running the short transport pattern test. In this mode, the JESD04B link layers operate normally, but the transport layers are disabled. After the initial lane alignment sequence (ILAS), each lane transmits an identical octet stream that increments from 0x00 to 0xFF and repeats.

To verify the ramp pattern at output data of DLL to the input receiver transport layer, follow these steps:

1. Change the command in line 7 of the ADC12DJxx00_JMODE1.cfg file to:
   ```
   0x0205 0x04 // Set JTEST to 4 gives ramp test mode.
   ```
2. Reprogram the clocks and ADC.
3. Reconfigure the FPGA.
4. Type the `start_basic_test` procedure in the System Console to execute the `.tcl` script to initialize the JESD204B links.
5. In the Instance Manager, click `rx_link ➤ run Analysis`.
6. If you want to check the JESD204B link up process, set a trigger condition to the `dev_sync_n` signal. The signal tap waits for trigger condition to occur. The trigger condition should occur once you execute the `start_basic_test` procedure.

**Figure 11.** Octet Ramp Pattern Captured at the Output of DLL
SYSREF+/-, thus the periodic SYSREF signal is required to achieve deterministic latency. The SYSREF period from LMK04828 is configured to run at a frequency equals to the Local Multi-Frame Clock (LMFC) frequency before the SYSREF signal is supplied to the ADC and FPGA. The SYSREF pulse restarts the LMFC counter on the JESD204B IP cores and converter device, and realigns the LMFC counter to the LMFC boundary.

To ensure the deterministic latency in the reference design, follow these steps:

1. Check the FPGA SYSREF single detection.

   For detailed description of the registers in the JESD204B RX IP core, refer to the Addressmap Information for 'altera_jesd204_rx_regmap'.

   Passing criteria: The value of csr_sysref_singledet and csr_sysref_lmfc_err should be zero.

   **Figure 12.** csr_sysref_singledet and csr_sysref_lmfc_err Observed from the .tcl Console

   syncn_sysref_ctrl[2] represents csr_sysref_singledet. This bit is zero when SYSREF is sampled.

   syncn_sysref_ctrl[1] represents csr_sysref_alwayson. This bit is one indicating SYSREF continuous detection.

   rx_err0[1] for link 0 and link1 is zero when SYSREF period matches the LMFC counter.

2. Check the SYSREF captured.

   Passing criteria: If the SYSREF is sampled correctly, the LMFC counter should be reset. Thus, the RBD_count value should only drift within 1–2 link clocks from one power cycle to another power cycle. In this test, the RBD_count is consistently 1 across 5 power cycle tests. It means the /R/ character is consistently received at 1 LMFC count before the next LMFC boundary for 5 power cycle tests.
3. Check the latency from the start of combined `SYNC_n` deassertion output from the two JESD204B IP cores to the first user data output. Passing criteria: You should observe consistent latency from the start of combined `SYNC_n` deassertion to the assertion of the `jesd204_rx_link_valid` signal. In this design test, you should consistently observe 67 link cycles clock from one power cycle to another.

Figure 14. Measured Latency from the Start of Combined `SYNC_n` to the First User Data Output

4. Ensure the data latency is fixed during user data phase. Passing criteria: The ramp pattern should be in perfect shape with no distortion.

Figure 15. Ramp Pattern in Perfect Shape
**Rebuilding Design**

You need to regenerate the platform designer system if you want to modify the system, such as IP components settings and interfaces.

To regenerate the HDL files for the top level platform designer system, follow these steps:

1. Open `altera_jesd204_ed_qsys_RX.qsys`.
2. In the platform designer GUI, click **Generate ➤ Generate HDL**.
   
   *Note:* If you want to modify the JESD204B IP cores, Transceiver PHY Reset Controller, and other component settings, right click on the `altera_jesd204Subsystem_RX` component and select **Drill into Subsystem**. After the modification, save the system. Go back to the top level platform designer system file and generate the HDL files.

3. To rerun design in hardware, follow the steps in the *Running the Reference Design* section.

**Related Links**

*Running the Reference Design* on page 8

Provides the steps to rerun the reference design.

### Procedures in the main.tcl System Console Script

**Table 4. Procedures in the main.tcl System Console Script**

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start_basic_test</td>
<td>—</td>
<td>Main procedure that performs global reset, set checker test mode, enable SYSREF continuous detection mode, clear JESD204B IP cores error status registers, and report link status.</td>
</tr>
<tr>
<td>reset</td>
<td>—</td>
<td>Perform global reset.</td>
</tr>
<tr>
<td>force_link_frame_reset</td>
<td>0, 1</td>
<td>0: Deassert link and frame reset. 1: Assert and hold link and frame reset.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: Link and frame clock domains should be held in reset while writing to the JESD204B IP control and status register (CSR).</td>
</tr>
</tbody>
</table>
| set_testmode               | alt, ramp, prbs | Set pattern checker to alternate pattern.  
                            |                  | ramp: Set pattern checker to ramp pattern.  
                            |                  | prbs: Set pattern checker to pseudo-random binary sequence (PRBS) pattern. |
| sysref_con                 | —      | Enable SYSREF continuous detection with SYSREF single detection. |
| rbd_offset                 | integer| Adjust RBD offset value to eliminate RX lane deskew error if needed. Refer to the JESD204B IP Core User Guide for more details. |

*continued...*
**Procedure** | **Value** | **Description**
--- | --- | ---
read_status_pio | — | Read the status of the parallel I/O (PIO) registers. The PIO status has the following configurations:
• Bit 0—Core PLL locked
• Bit 1—RX transceiver ready (Link 0)
• Bit 2—RX link error (Link 0)
• Bit 3—RX transceiver ready (Link 1)
• Bit 4—RX link error (Link 1)

*Note:* Use the `read_err_status` procedure to report error in description.

read_err_status | — | Read the JESD204B IP error status registers. For detailed description of the status registers, refer to the Addressmap Information for 'altera_jesd204_rx_regmap'.

clear_err_status | — | Clear the JESD204B IP error status registers.

read_rx_status0 | — | Read JESD204B IP rx_status0 register. For detailed description of the status registers, refer to the Addressmap Information for 'altera_jesd204_rx_regmap'.

wait_seconds | integer | Set to wait for integer seconds.

eval_test | — | Report the link status.

**Related Links**
Programmable RBD Offset, JESD204B IP Core User Guide
Provides more information on adjusting the RBD offset value.

**Document Revision History for AN 833: Intel Stratix 10 GX 16-Lane RX JESD204B-ADC12DJ3200 Interoperability Reference Design**

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
</table>