Hardware Demonstration Design

A hardware demonstration design, targeting the Kintex-7 KC705, Zynq-7000 ZC706, Virtex-7 VC709 or Artix-7 AC701 evaluation platforms, can be generated using the script provided in the download. This is a complete FPGA design, incorporating transmit and receive JESD204 cores which are connected using Transceiver PMA loopback to allow the operation of the JESD204 cores to be explored. This design can be implemented using Xilinx Vivado Design Suite 2016.1, the provided source code and scripts. The design can be downloaded to the targeted FPGA board, controlled and monitored using a Xilinx Vivado Hardware Manager GUI.

The JESD204 Hardware Demo folder contains the following files and folders:

- *jesd204_hwdemo.tcl*. This script is to be used to generate the Hardware Demo for any of the supported platforms.
- *hwdemo_files* folder contains all the required source code and constraint files for the hardware demo project.
- *hw_1* folder which contains a file for setting up the Vivado VIO GUI console.

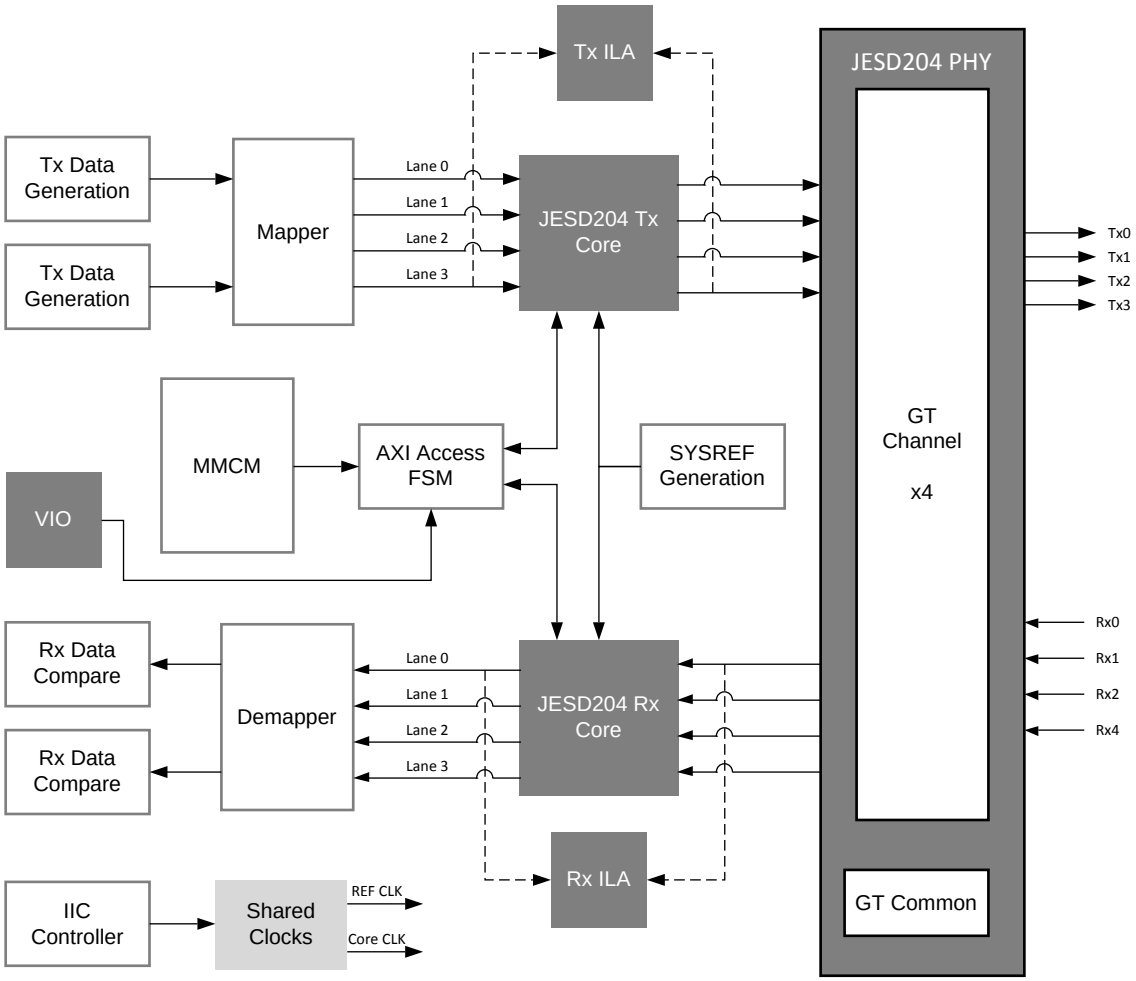

Figure 1 Hardware Demonstration Design

Figure 1 illustrates the hardware demonstration design which consists of the following:

- Transmit JESD204 core *jesd204_0* which has been generated without shared logic.
- Receive JESD204 core *jesd204_1* which has been generated without shared logic.

- JESD204 PHY core *jesd204_phy_0* which has been generated with shared logic included in core.
- Hardware demonstration top level wrapper, instantiating the JESD204 cores top level, JESD204 PHY core's top level and the additional modules listed below:
  - Shared clocking module
  - Transmit data generation and Mapper modules
  - Receive Demapper and data compare modules
  - Vivado Logic Analyser ILA and VIO modules
  - Additional logic to set up, control and monitor the JESD204 cores

Overview of the Hardware Demonstration Design

The hardware demonstration design incorporates 4-lane transmit and receive JESD204 cores. The transceiver line rates and reference clock frequencies for each design can be seen in Table 1. The demonstration uses the PMA loopback mode of the transceivers to allow standalone operation without the need of external connections.

Device	Transceiver Line Rate (Gb/s)	Reference Clock Frequency (MHz)
Kintex-7 KC705	10.0	250
Zynq-7000 ZC706	10.0	250
Artix-7 AC701	6.144	153.6
Virtex-7 VC709	10.0	250

Table 1 Transceiver Setup

Generating the Hardware Demonstration Design

All the files required to create the hardware demonstration design are located in the file available for download. The following steps must be carried out to successfully generate the hardware demo project.

1) Unzip and open the hardware demo folder.
2) This folder contains the *jesd204_hwdemo.tcl* file which is a Vivado tcl script.
   a) Linux
      i) From terminal console, navigate to the hardware demo folder
      ii) Without opening the Vivado GUI, run *vivado –mode tcl –source jesd204_hwdemo.tcl*.
   b) Windows
      i) Open a Windows Command prompt and navigate to the hardware demo folder
      ii) From the command prompt and without opening the Vivado GUI, run *vivado –mode tcl –source jesd204_hwdemo.tcl*.
3) When asked, enter the number of the Hardware Demo to be generated. The following options are available:
   a) Kintex-7 KC705
   b) Virtex-7 VC709
   c) Zynq-7000 ZC706
   d) Artix-7 AC701

4) This implement script will perform the following actions:
   a) Create a new Vivado project
   b) Add all the required IP and source files to the project.
   c) Synthesize the design
   d) Implement the design
   e) Generate a bitstream
5) Once the above operations are complete, open the Vivado project which is located in *<board name>_hwdemo/<board name>_hwdemo.xpr* using the Vivado GUI. This can also be done by using the '*start_gui*' command in the terminal console, once the bitstream has been generated.
6) Once the project has been created and the hardware manager has been opened, the *hw_1* folder must be copied into the following directoy: *<board name>_hwdemo/<board name>_hwdemo.hw.* This will setup the VIO console in the Vivado Hardware Manager.

Running the Hardware Design Using Vivado Lab Tools

Prerequisites for Running the Demonstration

- KC705 or ZC706 or VC709 or AC701 Evaluation Platform
- Vivado 2016.1 toolset
- Micro-USB cable for USB JTAG connection

Configuring the FPGA

- Connect USB port on host PC to USB-JTAG port on the evaluation board (micro-USB)
- Power up the evaluation board
- Open the Vivado Hardware Manager from the Flow Navigator in the opened project
- Open a new hardware target
- Select the targeted FPGA, and click on next. Leave all other options on the default values[1].
- Right click on the device that should have appeared in the GUI and select '*program device*' from drop down menu.
- Locate the bitfile (the location of the file should be selected by default) and click ok. This action will program the device.

Once the configuration is complete, the JESD204 cores should come up synchronized[2].

Setting Up the VIO GUI

Once the device has been configured the ILAs and VIO probes will appear in the Vivado GUI under the device (hw_ila_1, hw_ila_2 and hw_vio_1). To add the probes to the VIO window

- Highlight all the VIO probes under *hw_vio_1* and right click. This will open a menu, select '*Add Probes to VIO Window*'
- Signals should appear in the console and if the *hw_1* folder was copied into the correct location, LEDs and buttons should already be setup.
- Reorder the signals in the VIO console to match that of Figure 2.

[1] At this point, the *hw_1* folder provided in the download must be copied into the <board name>_hwdemo/<board name>_hwdemo.hw directory to ensure that the VIO GUI is setup correctly.
[2] A 6 second delay in the initial start-up sequence is required in the Kintex-7, Viretx-7 and Zynq-7000 designs due to the time required for the Si5324 to LOCK to the correct clock frequency.

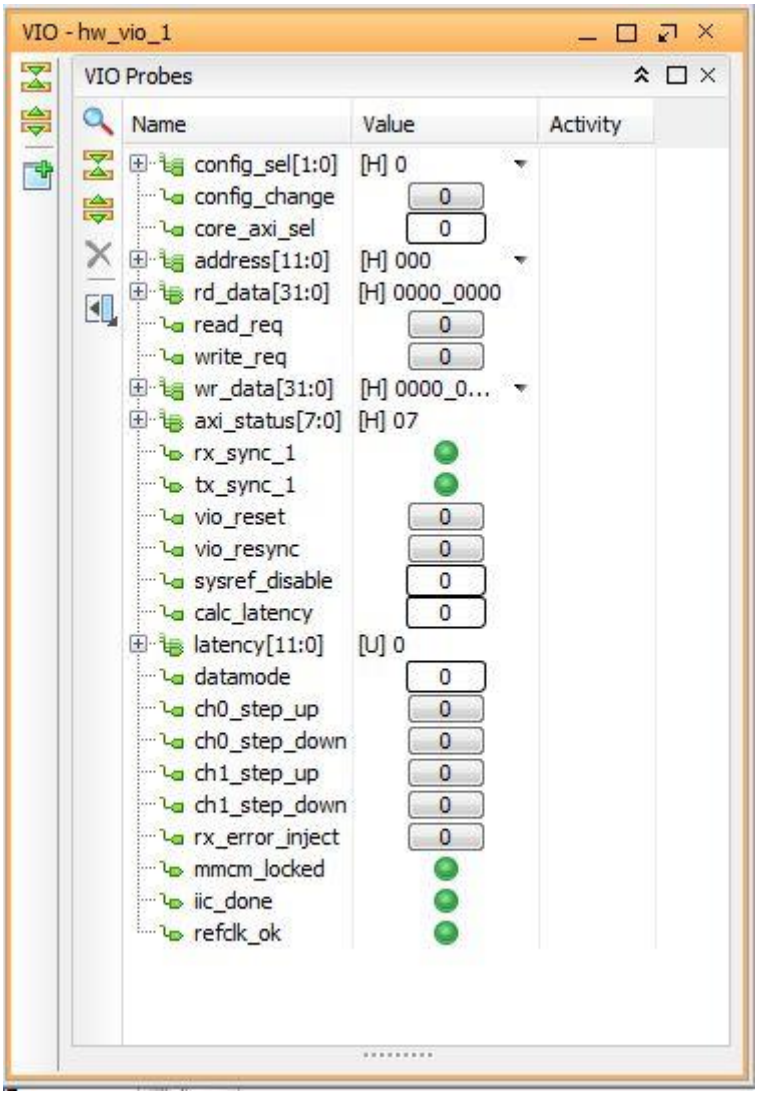

Figure 2 VIO GUI Setup

Operating the Hardware Demonstration

Read/Write Access to JESD204 Core Control and Status Registers Using VIO Console

The VIO Console allows read/write access to all of the JESD204 cores Control and Status registers.

To Perform a Register Read Access

1. Using the *core_axi_sel* toggle button select
    a. '0' to read from the receive JESD204 core
    b. '1' to read from the transmit JESD204 core
2. Enter the required register address value (in hexadecimal) in the '*address'* section in the VIO console.
3. Press the '*read_req*' button from the VIO console (notice the activity on the '*axi_status'* probe).
4. The data read from the addressed core is reflected in the '*rd_data*' probe in the VIO console.

To Perform a Register Write Access

1. Using the *axi_ctrl* toggle button select
    a. '0' to write to the receive JESD204 core
    b. '1' to write to the transmit JESD204 core
2. Enter the required register address value (in hexadecimal) in the '*address'* section in the VIO console.
3. Enter the required register write value (in hexadecimal) in the '*wr_data'* section in the VIO console.
4. Press the '*write_req*' button from the VIO console (notice the activity on the '*axi_status'* probe).
5. The requested write operation is carried out.

Soft Reset/Reset Pushbutton

A *soft* reset can be performed by the *'vio_reset'* pushbutton in the Control VIO console. This performs a full reset of the JESD204 cores and the transceivers.

The same action can also be performed by using a specific pushbutton on the device.

- Kintex-7: North SW2 pushbutton
- Virtex-7: North SW3 pushbutton
- Zynq-7000: Left SW7 pushbutton
- Artix-7: North SW3 pushbutton

Reconfiguring with the Preset Configurations

The demonstration provides a preset configuration mechanism, allowing any of four preset configurations to be written to the JESD204 cores.
To perform a configuration change:

1. Enter the required configuration on *config_sel* (0 to 3) in the VIO Console.
2. Press the '*config_change'* pushbutton in the VIO Console.

The control registers of the TX and RX JESD204 cores are configured with matching settings according to those defined in Table 2[3].

Main Configuration Settings				
Config.	F (Octets per Frame)	K (Frames per Multiframe)	Multiframe Size	Scrambling
0	2	16	32	ON
1	1	32	32	ON
2	2	16	32	OFF
3	1	32	32	OFF

Table 2 Preset Configurations

Force Resync

After the TX and RX cores have synchronized (SYNCB deasserted by the Rx core), they can be forced to re-synchronize by pressing the '*vio_resync*' button. This forces SYNCB Low at the transmit core, which in turn forces a complete link resynchronization.

SYSREF Disable

Setting this value to '1' inhibits the TX and RX cores receiving any SYSREF pulses. If a resync is forced with no SYSREF, the cores do not regain SYNC.

TX Data Mode

The '*datamode*' toggle button controls the TX Data Generation mode

- 0 = Simple Digital Pattern (RX data compare enabled when this mode is selected)
- 1 = Analog - Sine wave with variable frequency. (Rx data compare is disabled when this mode is selected). The data is generated from a fixed lookup table containing pre-generated 16-bit data sample values for a full 360° sine wave.

TX Analog Channel Frequency Control

The frequency of the generated sine wave can be controlled independently for the two TX data generation channels (when *analog* data mode is selected). This is achieved by increasing/decreasing the step value used to index into the sine wave lookup table, increasing or decreasing the rate at which the sampled data traverses the sine wave.

A single toggle of the *Ch0_step_up* or *Ch1_step_up* pushbutton increases the step value for the associated channel, doubling the frequency of the sine wave. A single toggle of the *Ch0_step_down* or *Ch1_step_down* pushbutton decreases the step value for the associated channel, halving the frequency of the sine wave.

Viewing Analog Signal Data

The Vivado Hardware Manager allows you to setup triggers and view the various probes at the defied moment. For simplicity we will explain how to trigger an event and view the analog signals.

1. Select *datamode* = 1 to generate an analog signal from the transmitter.
2. From the GUI right click on *hw_ila_1* (contains the RX probes) and select *Run Trigger*. This will plot the probes at that specific moment.
3. A new window should open with a waveform including all the probes.
4. To view the analog signal select the following signals
    a. All four *rx_tdata* probes

[3] If the system is reset by using the vio reset button or the push button on the device, by default, the cores return to Configuration 0.

b. All four *gt_rxdata* probes

5. Right click and under *Wave Form Style* select Analog.

You should now be able to see the analog signal entering the GT (*gt_rxdata*) and the exiting the Rx JESD204 core (*rx_tdata*).

The same actions can be performed for the *hw_ila_2* which contains the Tx probes.

Digital Pattern Error Injection

To allow verification of the RX data comparison, errors can be injected on the RX channel 0 data after the Demapper module.

- Select *Data Mode* = 0 (Digital Mode)
- Trigger a capture on hw_ila_1 (see Viewing Analog Signal Data).
- Observe RX error counts are zero.
- From the VIO window select *rx_error_inject*.
- Observe error(s) have been detected on RX Ch0 by retriggering hw_ila_1.

Latency Test

It is possible to view the latency from the moment the data enters the Tx JESD204 core and exits the Rx core. The latency is calculated by inserting a known pattern (0xB5 in our case) into the Tx JESD204 core and then waiting to receive it on the Rx data output. To enable this test, the '*calc_latency*' button must to be set to '1'. Once activated, the latency will appear in the *'latency'* output in the VIO console. The latency is calculated in core clock cycles. While the '*calc_latency*' VIO button is high, the latency is refreshed every 68 msec.

Status LEDs

The user GPIO LEDs on the evaluation boards are used to provide additional visual status indications. The LEDS are assigned as shown in Table 3 for the KC705 and VC709 devices, Table 4 for the ZC706 device and Table 5 for the AC701 device.

GPIO LED	Function	Description
0	Rx SYNC	ON = SYNCB deasserted by Rx core
1	Tx SYNC	ON = SYNCB deasserted at Tx core
2	Rx ARESETN	ON = Rx core out of reset
3	Tx ARESETN	ON = Tx core out of reset
4	IIC DONE	ON = IIC programming complete
5	MMCM LOCK	ON = MMCM locked for local clock generation
6	Rx aclk ALIVE	Flashing – divide of Rx core clock
7	Tx aclk ALIVE	Flashing – divide of Tx core clock

Table 3 GPIO LEDs for Kintex and Virtex Devices

GPIO LED	Function	Description
0	Tx aclk ALIVE	Flashing – divide of Tx core clock
1	Tx SYNC	ON = SYNCB deasserted at Tx core
2	Rx SYNC	ON = SYNCB deasserted at Rx core

Table 4 GPIO LEDs for Zynq Device

GPIO LED	Function	Description
0	Rx aclk ALIVE	Flashing – divide of Rx core clock
1	Rx ARESETN	ON = Rx core out of reset
2	Tx ARESETN	ON = Tx core out of reset
3	Tx SYNC	ON = SYNCB deasserted at Tx core

Table 5 GPIO LEDs for Artix-7 Device