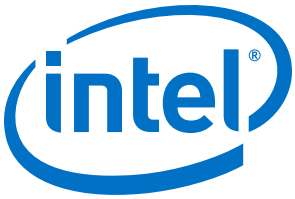


JESD204C Intel Stratix 10 FPGA IP Design Example User Guide

Updated for Intel® Quartus® Prime Design Suite: **19.2**

IP Version: **1.0.0**

Subscribe

Send Feedback

UG-20243 | 2019.08.01

Latest document on the web: **PDF** | **HTML**

Contents

1. About the JESD204C Intel Stratix 10 FPGA IP Design Example User Guide

This user guide provides the features, usage guidelines, and detailed description about the design examples for the JESD204C Intel® FPGA IP using Intel Stratix® 10 devices.

Intended Audience

This document is intended for:

- Design architect to make IP selection during system level design planning phase
- Hardware designers when integrating the IP into their system level design
- Validation engineers during system level simulation and hardware validation phase

Related Documents

The following table lists other reference documents which are related to the JESD204C Intel FPGA IP.

Table 1. Related Documents

Reference	Description
JESD204C Intel FPGA IP User Guide	Provides information about the JESD204C Intel FPGA IP.
Intel Stratix 10 Device Data Sheet	Provides information about the electrical characteristics, switching characteristics, configuration specifications, and timing for Intel Stratix 10 devices
Intel Stratix 10 E-Tile Transceiver PHY User Guide	Provides information about the E-tile Transceiver PHY.

Acronyms and Glossary

Table 2. Acronym List

Acronym	Expansion
LEMC	Local Extended Multiblock Clock
FC	Frame clock rate
ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
DSP	Digital Signal Processor
TX	Transmitter
RX	Receiver
DLL	Data link layer
CSR	Control and status register

continued...

Acronym	Expansion
CRU	Clock and Reset Unit
ISR	Interrupt Service Routine
FIFO	First-In-First-Out
SERDES	Serializer Deserializer
ECC	Error Correcting Code
FEC	Forward Error Correction
SERR	Single Error Detection (in ECC, correctable)
DERR	Double Error Detection (in ECC, fatal)
PRBS	Pseudorandom binary sequence
MAC	Media Access Controller. MAC includes protocol sublayer, transport layer, and data link layer.
PHY	Physical Layer. PHY typically includes the physical layer, SERDES, drivers, receivers and CDR.
PCS	Physical Coding Sub-layer
PMA	Physical Medium Attachment
RBD	RX Buffer Delay
UI	Unit Interval = duration of serial bit
RBD count	RX Buffer Delay latest lane arrival
RBD offset	RX Buffer Delay release opportunity
SH	Sync header
TL	Transport layer

Table 3. Glossary List

Term	Description
Converter Device	ADC or DAC converter
Logic Device	FPGA or ASIC
Octet	A group of 8 bits, serving as input to 64/66 encoder and output from the decoder
Nibble	A set of 4 bits which is the base working unit of JESD204x specifications
Block	A 66-bit symbol generated by the 64/66 encoding scheme
Line Rate	Effective data rate of serial link Lane Line Rate = (Mx Sx N'x 66/64 x FC) / L
Link Clock	The associated parallel data will be 128 bit/132 bit instead of 64 bit/66 bit. Link Clock = Lane Line Rate/132.
Frame	A set of consecutive octets in which the position of each octet can be identified by reference to a frame alignment signal.
Frame Clock	A system clock which runs at the frame's rate, that must be 1x, 2x, or 4x link clock.
Samples per frame clock	Samples per clock, the total samples in frame clock for the converter device.

continued...

Term	Description
LEMC	Internal clock used to align the boundary of the extended multiblocks between lanes and into the external references (SYSREF or Subclass 1).
Subclass 0	No support for deterministic latency. Data should be immediately released upon lane to lane deskew on receiver.
Subclass 1	Deterministic latency using SYSREF.
Multipoint Link	Intra-device links with 2 or more converter devices.
64B/66B Encoding	Line code that maps 64-bit data to 66 bits to form a block. The base level data structure is a block that starts with 2-bit sync header.

Table 4. Symbols

Term	Description
L	Number of lanes per converter device
M	Number of converters per device
F	Number of octets per frame on a single lane
S	Number of samples transmitted per single converter per frame cycle
N	Converter resolution
N’	Total number of bits per sample in the user data format
CS	Number of control bits per conversion sample
CF	Number of control words per frame clock period per link
HD	High Density user data format

2. JESD204C Intel FPGA IP Design Example Quick Start Guide

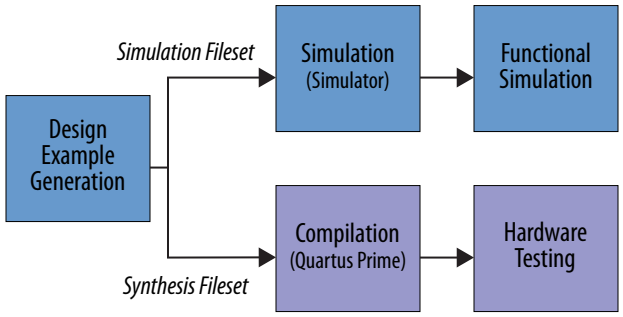
The JESD204C Intel FPGA IP design examples for Intel Stratix 10 devices features a simulating testbench and a hardware design that supports compilation and hardware testing.

The JESD204C Intel FPGA IP provides two preset settings for Intel Stratix 10 E-tile devices in duplex mode.

- JESD204C design example for L=2, M=8, F=12, with data rate of 24.333 Gbps
- JESD204C design example for L=4, M=8, F=4, with data rate of 16.222 Gbps

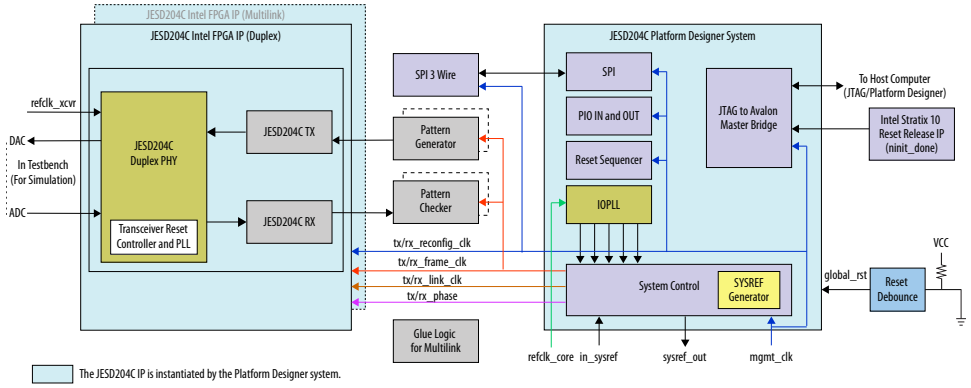
You can generate the JESD204C design examples through the IP catalog in the Intel Quartus® Prime Pro Edition software.

Figure 1. Development Stages for the Design Example



2.1. Design Example Block Diagram

Figure 2. JESD204C Design Example High-level Block Diagram



The design example consists of the following modules:

- Platform Designer system
  - JESD204C Intel FPGA IP
  - JTAG to Avalon master bridge
  - Parallel I/O (PIO) controller
  - Serial Port Interface (SPI)—master module
  - Core PLL
  - SYSREF generator
- Pattern generator
- Pattern checker
- IOPLL

Table 5. Design Example Modules

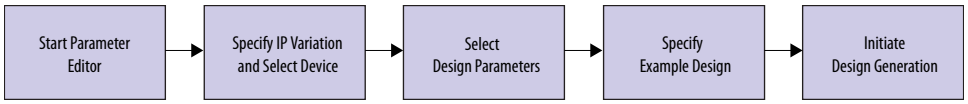
Components	Description
Platform Designer system	The Platform Designer system instantiates the JESD204C IP data path and supporting peripherals.
JESD204C Intel FPGA IP	This Platform Designer subsystem contains the duplex JESD204C IP instantiated together with the PHY.
JTAG to Avalon Master bridge	This bridge provides system console host access to the memory-mapped IP in the design through the JTAG interface.
Parallel I/O (PIO) controller	This controller provides a memory-mapped interface for sampling and driving general I/O ports.
SPI master	This module handles the serial transfer of configuration data to the SPI interface on the converter end.
SYSREF generator	The SYSREF generator uses the link clock as a reference clock and generates SYSREF pulses for the JESD204C IP and the ADC module.
Pattern generator	The pattern generator generates a PRBS or ramp pattern.
Pattern checker	The pattern checker verifies the PRBS or ramp received, and flags an error when it finds a mismatch of data sample.
IOPLL	For Intel Stratix 10 E-tile devices, the design example uses an IOPLL to generate a user clock for transmitting data into the JESD204C IP.

2.2. Hardware and Software Requirements

Intel uses the following hardware and software to test the design examples in a Linux system:

- Intel Quartus Prime Pro Edition software version 19.2
- ModelSim*, Xcelium*, NCSim (Verilog only), or VCS*/VCS MX simulator
- Intel Stratix 10 TX Signal Integrity Development Kit (1ST280EY2F55E2VG) for hardware testing

2.3. Generating the Design



To generate the design example from the IP parameter editor:

1. Create a project targeting Intel Stratix 10 E-tile device family and select the desired device.
2. In the IP Catalog, **Tools ➤ IP Catalog**, select **JESD204C Intel FPGA IP**.
3. Specify a top-level name and the folder for your custom IP variation. Click **OK**.
4. Select a design from the **Presets** library and click **Apply**. When you select a design, the system automatically populates the IP parameters for the design.

   Note: If you select another design, the settings of the IP parameters change accordingly.
5. Under the **Example Design** tab, specify the design example parameters as described in *Design Example Parameters*.
6. Click **Generate Example Design**.

The software generates all design files in the sub-directories. These files are required to run simulation, compilation, and hardware testing.

2.3.1. Design Example Parameters

The JESD204C Intel FPGA IP parameter editor includes a **Design Example** tab for you to specify certain parameters before generating the design example.

Table 6. Parameters in the Example Design Tab

Parameter	Options	Description
Select Design	• System Console Control • None	Select the system console control to access the design example data path through the system console.
Simulation	On, Off	Turn on for the IP to generate the necessary files for simulating the design example.
Synthesis	On, Off	Turn on for the IP to generate the necessary files for Intel Quartus Prime compilation and hardware demonstration.
HDL format (for simulation)	Verilog only	Select the HDL format of the RTL files for simulation.
HDL format (for synthesis)	Verilog only	Select the HDL format of the RTL files for synthesis.
Generate 3-wire SPI module	On, Off	Turn on to enable 3-wire SPI interface instead of 4-wire.

continued...

Parameter	Options	Description
Sysref mode	• One-shot • Periodic • Gapped periodic	Select whether you want the SYSREF alignment to be a one-shot pulse, periodic, or gapped periodic, based on your design requirements and timing flexibility. • One-shot: Select this option to enable SYSREF to be a one-shot signal. Although `sysref_ctrl` default value is 0, you can change the default value to 1 before releasing the JESD204C IP from reset. After the reset deasserts, change the SYSREF so that there's a one shot SYSREF from 0 to 1. • Periodic: SYSREF in periodic mode has 50:50 duty cycle. SYSREF period is E*SYSREF_MULP. • Gapped Periodic: SYSREF has programmable duty cycle of granularity of 1 link clock cycle. SYSREF period is E*SYSREF_MULP. For out-of-range duty cycle setting, the SYSREF generation block should automatically infer 50:50 duty cycle.
Select board	• None • S10 TX SI Devkit (Intel Stratix 10 TX Signal Integrity Development Kit)	Select the board for the design example. • None: This option excludes hardware aspects for the design example. All the pin assignments will be set to virtual pins. • Intel Stratix 10 TX Signal Integrity Development Kit: This option automatically selects the project's target device to match the device on this development kit. You may change the target device with the **Change Target Device** parameter below if your board revision has a different grade of the default targeted device. All the pin assignments are set according to the development kit.
Test pattern	• PRBS-7 • PRBS-9 • PRBS-15 • PRBS-23 • Ramp	Select the patten generator and checker test pattern to either ramp or one of the PRBS patterns. If you select ramp, the logic expects the scrambling seed to be synchronized when the IP is able to decode link up. The first valid octet is loaded as the ramp initial value. Subsequent data must increment up to 0xFF and roll over to 0x00. Ramp pattern checker should check for identical pattern across all lanes. The PRBS options are some of the common sequence generating polynomials.
Enable internal serial loopback (Simulation)	On, Off	Turn on to enable internal serial loopback. If you turn on this option, the RX path takes the serial input from the TX path internally in the FPGA.
Enable command channel pattern (Simulation)	On, Off	Turn on to enable command channel pattern.

2.3.2. Directory Structure

The JESD204C design example directories contain generated files for the design examples.

Figure 3. Directory Structure for JESD204C Intel Stratix 10 Design Example

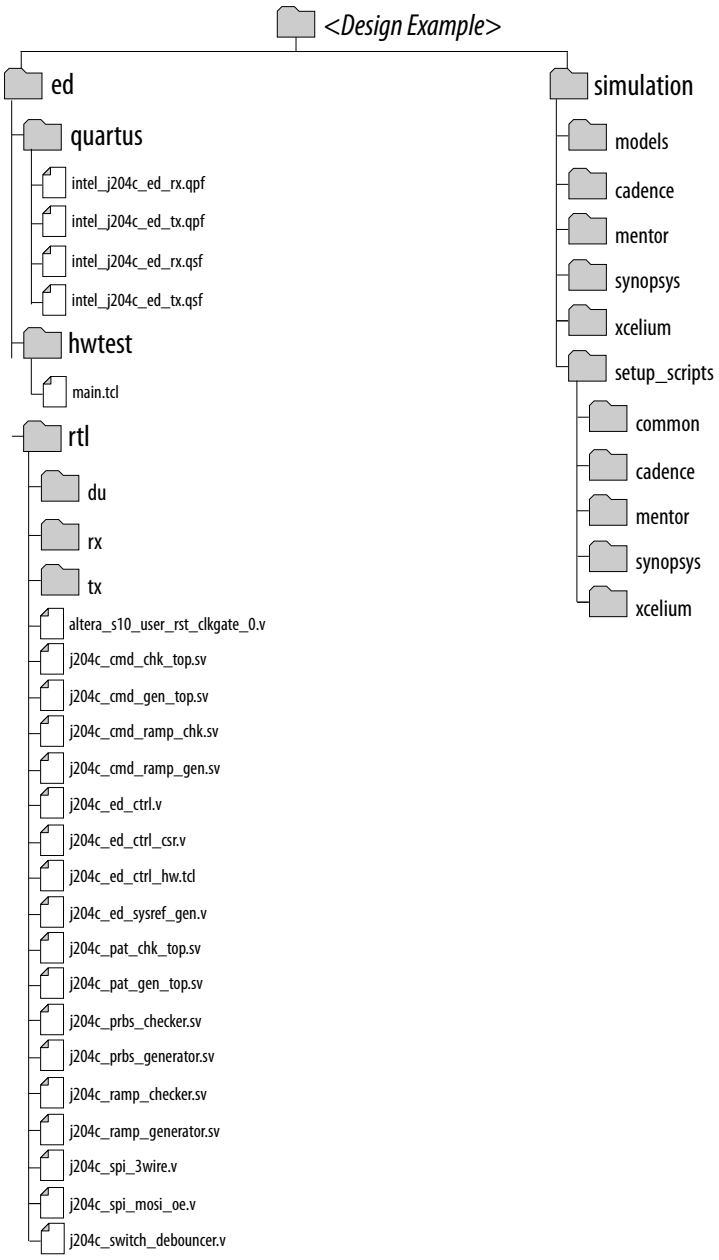


Table 7. Directory Files

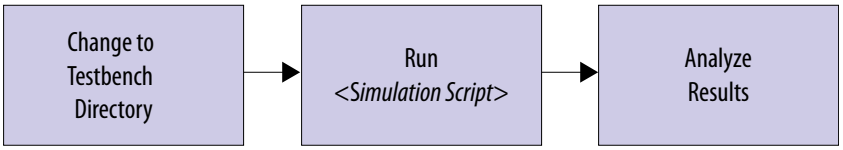
Folders	Files
ed/rtl	• du — intel_j204c_ed_rx_tx.sv — intel_j204c_ed_rx_tx.sdc — J204c_rx_tx_ip.qsys — j204c_rx_tx_ss.qsys — altera_s10_user_rst_clkgate_0.ip — intel_j204c_se_outbuf_1bit.ip • rx — intel_j204c_ed_rx.sv — intel_j204c_ed_rx.sdc — j204c_rx_ip.qsys — j204c_rx_ss.qsys — altera_s10_user_rst_clkgate_0.ip — intel_j204c_se_outbuf_1bit.ip • tx — intel_j204c_ed_tx.sv — intel_j204c_ed_tx.sdc — j204c_tx_ip.qsys — j204c_tx_ss.qsys — altera_s10_user_rst_clkgate_0.ip — intel_j204c_se_outbuf_1bit.ip
ed/simulation/models	• tb_top.sv • j204c_tx/ • j204c_rx/
ed/simulation/cadence	• cadence_sim.sh • tb_top_wave.tcl
ed/simulation/mentor	• modelsim_sim.tcl • tb_top_waveform.do
ed/simulation/synopsys	• vcs — vcs_sim.sh — tb_top_wave_ed.do • vcsmx — vcsmx_sim.sh — tb_top_wave_ed.do
ed/simulation/xcelium	• xcelium_sim.sh • tb_top_wave.tcl
ed/simulation/setup_scripts/cadence	• cds.lib • hdl.var • ncsim_setup.sh • *<cds_libs folder>*

continued...

Folders	Files
ed/simulation/setup_scripts/mentor	• msim_setup.tcl
ed/simulation/setup_scripts/synopsys	• vcs — vcs_setup.sh • vcsmx — vcsmx_setup.sh — synopsys_sim.setup
ed/simulation/setup_scripts/xcelium	• xcelium_setup.sh • cds.lib • hdl.var • *<cds_libs folder>*

2.4. Compiling and Simulating the Design

The design example testbench simulates your generated design.



To simulate the design, perform the following steps:

1. Change the working directory to `<example_design_directory>/simulation/<Simulator>`.
2. In the command line, run the simulation script. The table below shows the commands to run the supported simulators.

Simulator	Command
NCSim	`sh cadence_sim.sh`
ModelSim	`vsim -c -do modelsim_sim.tcl`
VCS	`sh vcs_sim.sh`
VCS MX	`sh vcsmx_sim.sh`
Xcelium Parallel	`sh xcelium_sim.sh`

The simulation ends with messages that indicate whether the run was successful or not.

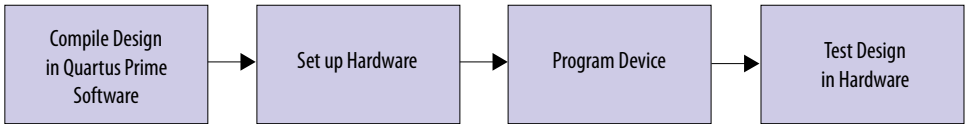
Figure 4. Successful Simulation

The average simulation run time is about 30 minutes.

```
INFO: tx_ready asserted
INFO: rx_ready asserted
Running JESD204C Simulation: L=8, M=8, F=3, DATARATE/L=24333.000000Mbps FCLK_MULP=2 WIDTH_MULP=8
Pattern Checker(s): OK!
Command Channel Pattern Checker(s): OK!
JESD204C Rx Core(s): SH Locked!
JESD204C Rx Core(s): EMB Locked!
JESD204C Tx Core(s): OK!
JESD204C Rx Core(s): OK!
TESTBENCH_PASSED: SIM PASSED!
ucli% quit
                V C S   S i m u l a t i o n   R e p o r t
Time: 205954704000 fs
CPU Time:   1359.710 seconds;       Data structure size:  39.2Mb
```

2.5. Compiling and Testing the Design

The JESD204C Intel FPGA IP parameter editor allows you to run the design example on a target development kit.



Perform the following steps to compile the design and program the development board:

1. Launch the Intel Quartus Prime software and compile the design (**Processing ➤ Start Compilation**).

   The timing constraints and pin assignments for the design example and the design components are automatically loaded during design example compilation.
2. Connect the development board to the host computer either by connecting a USB cable to the on-board Intel FPGA Download Cable II component or using an external Intel FPGA Download Cable II module to connect to the external JTAG connector.
3. Launch the **Clock Control** application that is included with the development board, and set the clock settings according to the selected data rate.

   Note: Refer to the *Intel Stratix 10 TX Transceiver Signal Integrity Development Kit User Guide* for more information on using the **Clock Control** application.

Table 8. Clock Settings

Clock Name	Clock Control GUI	Clock Frequency
`refclk_xcvr`	OUT8	Select the frequency for the transceiver PLL reference clock in the IP parameter editor.
`refclk_core`	OUT2	Select the frequency for the core PLL reference clock in the IP parameter editor.
`mgmt_clk`	OUT3	100 MHz
`perst_clk`	OUT0	100 MHz

4. If you are performing external loopback test, attach the QSPDD interface to U39B (Bank 8B).
5. Configure the FPGA on the development board with the generated programming file (`.sof` file) using the Intel Quartus Prime **Programmer**.

Related Information

Intel Stratix 10 TX Transceiver Signal Integrity Development Kit User Guide
Provides more information about the clock controller.

3. Detailed Description for the JESD204C Design Example

The JESD204C design example demonstrates the functionality of data streaming using loopback mode.

You can specify the parameters settings of your choice and generate the design example.

The design example is available only in duplex mode.

3.1. System Components

The JESD204C design example provides a software-based control flow that uses the hard control unit with or without system console support.

The design example enables an auto link up in internal and external loopback modes, interoperability with a converter card.

You can either configure your own settings or use one of the two presets provided.

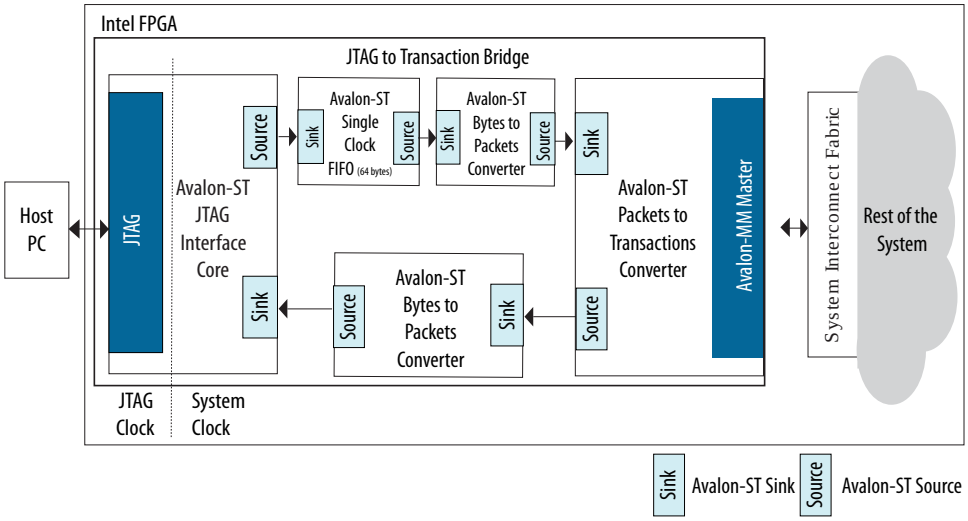
- L=2, M=8, F=12, with data rate of 24.333 Gbps
- L=4, M=8, F=4, with data rate of 16.222 Gbps

3.1.1. JTAG to Avalon Master Bridge

The JTAG to Avalon Master Bridge provides a connection between the host system to access the memory-mapped JESD204C IP through the JTAG interface.

Figure 5. System with a JTAG to Avalon® Master Bridge Core

Note: System clock must be at least 2X faster than the JTAG clock.



Each bridge consists of the following cores, which are available in the Platform Designer.

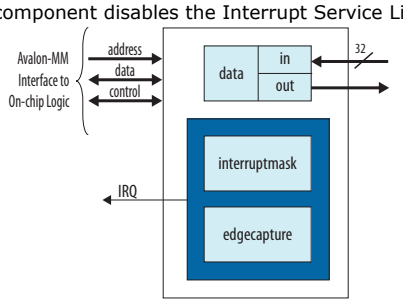
- Avalon-ST Serial Peripheral Interface and Avalon -ST JTAG Interface—Accepts incoming data in bits and packs them into bytes.
- Avalon -ST Bytes to Packets Converter—Transforms packets into encoded stream of bytes, and a similar encoded stream of bytes into packets.
- Avalon-ST Packets to Transactions Converter—Transforms packets with data encoded according to a specific protocol into Avalon-MM transactions, and encodes the responses into packets using the same protocol.
- Avalon -ST Single Clock FIFO—Buffers data from the Avalon -ST JTAG Interface. The FIFO block is only used in the JTAG to Avalon Master Bridge. For the bridges to successfully transform the incoming streams of bytes to Avalon-MM transactions, the streams of bytes must be constructed according to the protocols used by the cores.

3.1.2. Parallel I/O (PIO) Controller

Parallel I/O (PIO) module provides an Avalon-MM slave port interface and general input/output (I/O) ports. The PIO controller drives signals to LEDs.

Figure 6. PIO Core with Input Ports, Output Ports, and IRQ Support

By default, the Platform Designer component disables the Interrupt Service Line (IRQ).



The registers are assigned in the top level HDL file (`io_status` for status registers, `io_control` for control registers).

The tables below describe the signal connectivity for the status and control registers.

Table 9. PIO Controller Registers

Port	Bit	Signal
Out_port	0	USER_LED SPI programming done
	31:1	Reserved
In_port	0	USER_DIP internal serial loopback enable Off = 1 On = 0
	1	USER_DIP FPGA-generated SYSREF enable Off = 1 On = 0
	31:2	Reserved.

3.1.3. SPI Master

The SPI master module is a standard Platform Designer component in the **IP Catalog** standard library. This module uses the SPI protocol to facilitate the configuration of external converters (for example, ADC, DAC, external clock modules) via a structured register space inside the converter device. The SPI master has an Avalon-MM interface that connects to the Avalon master (JTAG to Avalon master bridge) via the Avalon-MM interconnect and can receive configuration instructions from the Avalon master.

The SPI master module controls up to 32 independent SPI slaves. The SCLK baud rate is configured to 20 MHz (divisible by 5).

This module is configured to a 4-wire, 24-bit width interface. If the **Generate 3-Wire SPI Module** option is selected, an additional module is instantiated to convert the 4-wire output of the SPI master to 3-wire.

Note: For multilink implementation, you need to multiplex MISO input.

3.1.4. SYSREF Generator

SYSREF is a critical signal for data converters with JESD204C interface.

For the JESD204C IP, the SYSREF_MULP register creates a SYSREF signal, which is n-integer multiple of the E parameter.

You must ensure E*SYSREF_MULP ≤16. For example, if E=1, the legal setting for SYSREF_MULP must be within 1–16, and if E=3, the legal setting for SYSREF_MULP must be within 1–5.

Note: If you set an out-of-range SYSREF_MULP, the SYSREF generator will fix the setting to SYSREF_MULP=1.

You can select whether you want the SYSREF alignment to be a one-shot pulse, periodic, or gapped periodic through the design example parameter editor.

Table 10. Examples of Periodic and Gapped Periodic SYSREF Counter

E	SYSREF_MULP	SYSREF PERIOD (E*MULP*16)	Programmable Duty cycle	Description
1	1	16	1..15	Gapped Periodic
1	1	16	Auto (8)	Periodic
1	2	32	1..31	Gapped Periodic
1	2	32	16	Periodic
1	3	48	1..47	Gapped Periodic
1	3	48	Auto (24)	Periodic
1	16	256	1..255	Gapped Periodic
1	16	256	Auto (128)	Periodic
2	1	32	1..31	Gapped Periodic
2	1	32	16	Periodic
2	2	64	1..31	Gapped Periodic
2	2	64	Auto (32)	Periodic
2	3	96	1..95	Gapped Periodic
2	3	96	Auto (48)	Periodic
2	8	256	1..255	Gapped Periodic E*SYSREF_MULP <=16
2	8	256	Auto (128)	Periodic
2	9..16	32		Illegal configuration force `sysref_period` = 2*1*16
16	1 Illegal<2..16>	256	1..255	Gapped Periodic E*SYSREF_MULP <=16

Table 11. SYSREF Control Registers

Bits	Description
sysref_ctrl[1:0]	SYSREF type • 'b00: One-hot • 'b01: Periodic • 'b10: Gapped periodic • 'b11: Reserved
sysref_ctrl[6:2]	SYSREF multiplier • 'h0: Illegal (default to 'h1) • 'h1: SYSREF_MULP = 1 • 'h16: SYSREF_MULP = 16 • 'h17 to 'h31: Illegal (default to 'h1)
sysref_ctr[7]	SYSREF select • 0: External SYSREF • 1: Internal SYSREF
sysref_ctrl[15:8]	SYSREF duty cycle for gapped-periodic type. Maximum value = (E*SYSREF_MULP*16)–1
sysref_ctrl[16]	SYSREF phase (for fpga_sysref output) • 0: posedge • 1: negedge
sysref_ctrl[17]	Manual control when SYSREF type is set to One-hot. • Write 1 to set SYSREF to high. • Write 0 to set to low.

3.1.5. Pattern Generator and Checker

The pattern generator and checker are useful for creating data sample stimulus for testing and not compatible with PRBS test mode on the ADC/DAC converter.

Table 12. Supported Pattern Generator

Pattern Generator	Description
PRBS pattern generator	The JESD204C IP supports PRBS pattern generator per data sample. The IP supports the following PRBS patterns: • PRBS23: $X^{23}+X^{18}+1$ • PRBS15: $X^{15}+X^{14}+1$ • PRBS9: $X^{9}+X^{5}+1$ • PRBS7: $X^{7}+X^{6}+1$
Ramp pattern generator	Ramp pattern increments 1 for every sample with the width of N, and rolls over when all bits in the sample are 1. Each converter device transmits an identical ramp pattern. Enable the ramp pattern test through the test_control[2] register.
Command channel ramp pattern generator	The JESD204C IP supports command channel ramp pattern generator per lane. Ramp pattern increments per 6 bits of command words. The starting seed is an increment pattern across lanes.

Table 13. Supported Pattern Checker

Pattern Checker	Description
PRBS pattern checker	The JESD204C PRBS scrambler is self-synchronizing and it is expected that when the IP is able to decode link up, the scrambling seed is already synchronized. PRBS scrambling seed will take up 8 octets to self initialize.
Ramp pattern checker	The JESD204C PRBS scrambler is self-synchronizing and it is expected that when the IP is able to decode link up, the scrambling seed is already synchronized. The first sample received is loaded as the ramp initial value. Subsequent samples within the same converter must increment and roll over when it hits the maximum value in sample width N.
Command channel ramp pattern checker	The JESD204C IP supports command channel ramp pattern checker. The first command word (6 bits) received is loaded as the initial value. Subsequent command words in the same lane must increment up to 0x3F and roll over to 0x00. The command channel ramp pattern checker checks for ramp patterns across lanes.

3.2. Design Example Clock and Reset

The JESD204C design example has a set of clock and reset signals.

Table 14. Design Example Clocks

Clock Name	Direction	Description
`mgmt_clk`	Input	LVDS differential clock with frequency of 100 MHz.
`refclk_xcvr`	Input	Transceiver clock with frequency of datarate/ factor of 33.
`refclk_core`	Input	Core reference clock with the same frequency of `refclk_xcvr`.
`in_sysref`	Input	SYSREF clock with frequency of datarate/ (66*32*E).
`sysref_out`	Output	
`txlink_clk` `rxlink_clk`	Internal	TX and RX link clock with frequency of datarate/ 132.
`txframe_clk` `rxframe_clk`	Internal	• TX and RX frame clock with frequency of datarate/33 (FCLK_MULP=4) • TX and RX frame clock with frequency of datarate/66 (FCLK_MULP=2) • TX and RX frame clock with frequency of datarate/132 (FCLK_MULP=1)
`tx_phase` `rx_phase`	Internal	• TX and RX phase clock with frequency of datarate/132, dutycycle 25% (FCLK_MULP=4) • TX and RX phase clock with frequency of datarate/132 (FCLK_MULP=2) • TX and RX frame clock with frequency of 1 (FCLK_MULP=1)
`txphy_clk` `rxphy_clk`	Internal	Transceiver parallel clock with frequency of datarate/64
`spi_sclk`	Output	SPI baudrate clock with frequency of 20 MHz

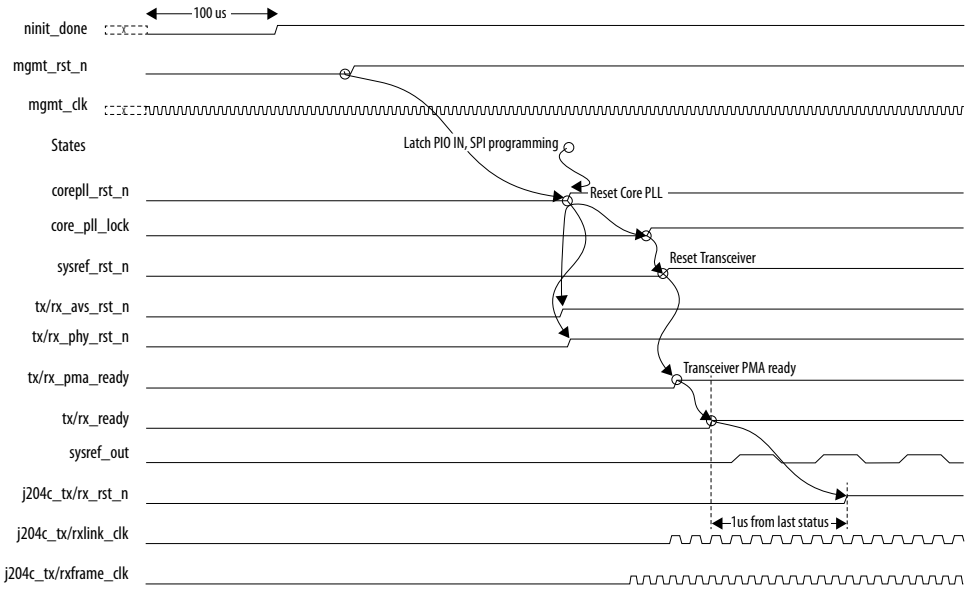
The CSR Avalon-MM reset is an independent domain. When you load the design example into an FPGA device, an internal `ninit_done` event ensures that the JTAG-AVMM bridge is in reset as well as all the other blocks.

The SYSREF generator has its independent reset to inject intentional asynchronous relationship for the `txlink_clk` and `rxlink_clk` clocks. This method is more comprehensive in emulating the SYSREF signal from an external clock chip.

Table 15. Design Example Resets

Clock Name	Direction	Description
`global_rst_n`	Input	Global reset to reset entire controller. For all blocks, except JTAG-Avalon-MM.
`ninit_done`	Internal	Internal reset IP block. For JTAG-Avalon-MM.
`mgmt_rst_n`	Internal	Reset for Avalon-MM configuration access. • `j20c_tx_avs_rst_n` • `j20c_tx_avs_rst_n` • `reconfig_reset` • `spi_rst_n` • `pio_rst_n` • `edctl_rst_n` (The `ed_ctl` block will not be reset by `hw_rst_n` or `global_rst_n`)
`sysref_rst_n`	Internal	Reset for SYSREF generator block in `ed_control` block. Waits for the core PLL to lock before deasserting.
`j204c_tx_phy_rst_n` `j204c_rx_phy_rst_n`	Internal	Reset transceiver. Waits for `tx_pma_ready`, `rx_pma_ready`, `tx_ready`, and `rx_ready` before deasserting.
`j204c_corepll_rst_n`	Internal	Resets the core PLL.
`j204c_tx_rst_n` `j204c_rx_rst_n`	Internal	Resets JESD204C Intel FPGA IP in `txlink_clk`, `rxlink_clk`, `txframe_clk`, and `rxframe_clk` domains. Must wait for all resets to deassert, and `core PLL locked`, `tx_pma_ready`, `rx_pma_ready`, `tx_ready`, and `rx_ready` to assert before deasserting.
`hw_rst_n`	From CSR	This is a software global register-based reset which is OR-ed with `global_rst_n`. This reset resets all the blocks except `ed_control` block.

Figure 7. Timing Diagram for the Design Example Resets



3.3. Design Example Signals

Table 16. System Interface Signals

Signal	Direction	Description
Clocks and Resets		
mgmt_clk	Input	100 MHz clock for system management.
refclk_xcvr	Input	PLL/CDR reference clock for transceiver PHY.
refclk_core	Input	Core PLL reference clock. Applies the same clock frequency as refclk_xcvr.
global_rst_n	Input	Global reset signal from the push button. This reset is an active low signal and the deassertion of this signal is synchronous to the rising-edge of mgmt_clk.
in_sysref	Input	SYSREF signal for JESD204C Subclass 1 implementation.
sysref_out	Output	SYSREF signal for JESD204C Subclass 1 implementation generated by the FPGA device.

Signal	Direction	Description
SPI		
spi_ss_n[LINK-1:0]	Output	Active low, SPI slave select signal.
spi_sclk[LINK-1:0]	Output	SPI serial clock.
spi_sdio[LINK-1:0] *Note:* When **Generate 3-Wire SPI Module** option is enabled.	Input/Output	Output data from the master to external slave. Input data from external slave to master.
spi_miso[LINK-1:0]	Input	Input data from external slave to the SPI master.

continued...

Signal	Direction	Description
Note: When **Generate 3-Wire SPI Module** option is not enabled.		
`spi_mosi[LINK-1:0]` *Note:* When **Generate 3-Wire SPI Module** option is not enabled.	Output	Output data from SPI master to the external slave.

Signal	Direction	Description
ADC/DAC		
`tx_serial_data[LINK*L-1:0]`	Output	Serial I/O to DAC.
`tx_serial_data_n[LINK*L-1:0]`		
`rx_serial_data[LINK*L-1:0]`	Input	Serial I/O from ADC.
`rx_serial_data_n[LINK*L-1:0]`		

Signal	Direction	Description
General Purpose I/O		
`user_led[3:0]`	Output	Indicates the status for the following conditions: • [0]: SPI programming done • [1]: TX link error • [2]: RX link error • [3]: Pattern checker error for Avalon ST data
`user_dip[3:0]`	Input	User mode DIP switch input: • [0]: Internal serial loopback enable • [1]: FPGA-generated SYSREF enable • [3:2}: Reserved

3.4. JESD204C Design Example Control Registers

The JESD204C design example registers use byte-addressing (8 bits).

Table 17. Register Access Type and Definition

This table describes the register access type for Intel FPGA IPs.

Access Type	Definition
RO	Software read-only (no effect on write). The value is hard-tied internally to either '0' or '1' and does not vary.
RO/V	Software read-only (no effect on write). The value may vary.
RC	• Software reads and returns the current bit value, then the bit self-clears to 0. • Software read also causes the bit value to be cleared to 0.

continued...

Access Type	Definition
RW	• Software reads and returns the current bit value. • Software writes and sets the bit to the desired value.
RW1C	• Software reads and returns the current bit value. • Software writes 0 and has no effect. • Software writes 1 and clears the bit to 0 if the bit has been set to 1 by hardware. • Hardware sets the bit to 1. • Software clear has higher priority than hardware set.
RW1S	• Software reads and returns the current bit value. • Software writes 0 and has no effect. • Software writes 1 and sets the bit to 1. • Hardware clears the bit to 0 if the bit has been set to 1 by software. • Software set has higher priority than hardware clear.

Table 18. Design Example Address Map

Component	Address
J204C Design Example Control	0x0102_0400 – 0x0102_04FF
MM Bridge	0x0000_0000 – 0x007F_FFFF
PIO Control	0x0102_0020 – 0x0102_002F
PIO Status	0x0102_0040 – 0x0102_004F
Reset Sequence 0	0x0102_0100 – 0x0102_01FF
Reset Sequence 1	0x0102_0200 – 0x0102_02FF
SPI Control	0x0102_0000 – 0x0102_001F
J204C Reconfig	0x0200_0000 – 0x021F_FFFF
J204C TX IP (Link 0)	0x000C_0000 – 0x000C_03FF
J204C RX IP (Link 0)	0x000D_0000 – 0x000D_03FF

Table 19. Design Example Control and Status Registers

These registers are in the `mgmt_clk` domain.

Register	Bit	Name	Access	Reset	Description
`rst_ctl` Offset = 0x00	31:0	`rst_assert`	RWS	0x0	Reset control. Use this to inject manual reset. Write 1 to assert reset. Write 0 again to deassert reset.
`rst_sts0` Offset = 0x04	31:0	`rst_status`	ROV	0x0	Reset status. Signals which can be monitored in the system to stagger reset sequence should be staggered. Core `pll_locked` status for Link 0 .
`rst_sts1` Offset = 0x08	31:0	`rst_status`	ROV	0x0	Reset status. Signals which can be monitored in the system to stagger reset sequence should be staggered. Link 0 `tx_pma_ready` and `rx_pma_ready` status for 16 lanes.

continued...

Register	Bit	Name	Access	Reset	Description
rst_sts2 Offset = 0x0c	31:0		ROV	0x0	Reserved.
rst_sts_detected0 Offset = 0x10	31:0	rst_sts_set	RW1C	0x0	Any reset status for Link 0, which is a pulsed indicator and needs to be latched so that it can be detected by software.
rst_sts_detected1 Offset = 0x14	31:0		RW1C		Reserved.
sysref_ctl Offset = 0x40	31:0	sysref_control	RWS	0x081	SYSREF control. Use this register to enable different patterns. SYSREF control can be mapped to enable different SYSREF patterns such as periodic, gapped periodic, and one-shot.
sysref_sts Offset = 0x44	31:0	sysref_status	ROV	0x0	SYSREF status. Log final value of sysref period and duty cycle used.
tst_ctl Offset = 0x80	31:0	tst_control	RWS	0x0	Test control. Use this register to enable different test mode. Test control can be mapped to enable different pattern generator, test command channel. [1:0] = prbs_test_ctl [2] = ramp_test_ctl
tst_sts0 Offset = 0x84	31:0	tst_status	RW1C	0x0	Test status of IP Link 0. Signals which can be monitored in the system to stagger reset sequence should be staggered.
tst_sts1 Offset = 0x88	31:0		RW1C	0x0	Reserved.
tst_err0 Offset = 0x8c	31:0	tst_error	RW1C	0x0	Test checker errors of IP Link 0. When error condition is set, use this register to log the errors until it is serviced by software. [0] = Pattern checker error [1] = tx_link_error [2] = rx_link_error [3] = Command pattern checker error
tst_err1 Offset = 0x90	31:0		RW1C	0x0	Reserved.

3.5. Hardware Test for System Console Control Design Example

Perform the following instructions to run the hardware test for the design example using the system console control in the Intel Quartus Prime software.

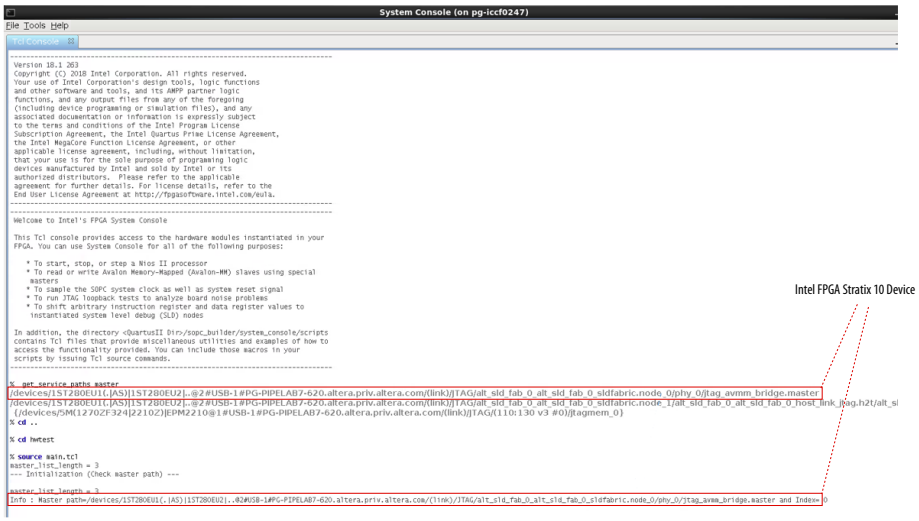
Note: This hardware test assumes that you configured your design in duplex mode for system console control. Make your own modifications if you're using simplex mode design.

1. Launch the System Console tool from Intel Quartus Prime (**Tools ➤ System Debugging Tools ➤ System Console**).
2. In the **TCL Console** command prompt, type `get_service_paths master` to print a list of devices connected to your JTAG chain.
3. Open the `main.tcl` script located in the `ed/hwtest/` directory in any text editor of your choice and locate the following line.

```
set master_index [expr {$master_list_length - <your offset>}]
```

4. Adjust the `master_index` offset as necessary to reflect your JTAG chain configuration such that the `master_index` always points to the Intel Stratix 10 device and save the file.
5. In the **TCL Console** command prompt, navigate to the `ed/hwtest/` directory (`cd../ed/hwtest/`) and execute the `main.tcl` script (`source main.tcl`). Your **TCL Console** window should resemble the following figure.

Figure 8. Source main.tcl



6. Type `start_basic_test` at the command prompt to execute the link setup and test procedure.

   This procedure executes a set of instructions to set up sysref on TX and RX as periodic , check PRBS pattern, configure the JESD204C IP PHY internal serial loopback mode and report link status.

   The following figure illustrates the expected result from a successful link setup and test for a variant of L=4, M=8, and F=4 with a data rate of 16.22 Gbps.

Figure 9. Successful Test in the System Console

```
Error status registers cleared
TX Error Status= 0x00000000
RX Error Status= 0x00000000
Tst_err0 Status= 0x00000000

CORE_PLL_LOCKED: 0x1
TX_PMA_READY: 0xf
RX_PMA_READY: 0xf
TX_XCVR_READY: 0xf
RX_XCVR_READY: 0xf
RX_CDR_LOCKED: 0xf
SH_LOCKED: 0xf
EMB_LOCKED: 0xf
TX_ERR_STATUS: 0x00000000
RX_ERR_STATUS: 0x00000000
tst_err0: 0x00000000
    Info: Bit 0 - PATTERN CHECKER ERROR
    Info: Bit 1 - TX LINK ERROR
    Info: Bit 2 - RX LINK ERROR
    Info: Bit 3 - CMD PATTERN CHECKER ERROR

PMA_load_status : PMA configuration loading done
Info: Lane 0 is passing
Info: Lane 1 is passing
Info: Lane 2 is passing
Info: Lane 3 is passing

HW_TEST : PASS
TX Error Status= 0x00000000
RX Error Status= 0x00000000
Tst_err0 Status= 0x00000000
```

Table 20. Procedures in the main.tcl System Console Script

The table describes useful procedures in the **main.tcl** that may be helpful in debugging.

Procedure	Values	Description
`get_service_paths`	{master}	Reports all devices that are connected to the JTAG chain. Use this information to set the master index to point to the Intel Stratix 10 device.
`get_master_index`	N/A	Sets the targeted device master index. Use `get_service_paths` master to determine the offset of the Intel Stratix 10 device in the JTAG chain, and edit the offset in this procedure accordingly.
`start_basic_test`	N/A	Procedure that executes a set of instructions to set up SYSREF on TX and RX as periodic, check PRBS pattern, configure the JESD204C IP PHY internal serial loopback mode, configure PMA and to report link status.
`reset`	N/A	Global reset
`xcvr_lpbk`	{0,1}	0: Disables internal serial loopback 1: Enables internal serial loopback
`testmode`	{ramp, prbs7, prbs9, prbs15, prbs23 }	ramp: Sets pattern generator and checker to ramp pattern prbs7-23: Set pattern generator and checker to the respective PRBS patterns
`sysref_type`	{onehot, periodic, gperiodic }	Sets the SYSREF type
`eval_test`	N/A	1. Loads status. 2. Checks lane by lane.

continued...

Procedure	Values	Description
		3. Shows the TX and RX registers values: • rst_sts0: — Bit 0 – Core PLL Locked • rst_sts1: — Bit 0-15 – TX PMA READY — Bit 16-31 – RX PMA READY • tx_status2: — Bit 0-15 – TX XCVR READY • rx_status2: — Bit 0-15 – RX XCVR READY — Bit 16-31 – RX CDR LOCKED • rx_status4: — Bit 0-15 – SH LOCKED • rx_status5: — Bit 0-15 – EMB LOCKED • tst_err0: — Bit 0 – PATTERN CHECKER ERROR — Bit 1 – TX LINK ERROR — Bit 2 – RX LINK ERROR — Bit 3 – COMMAND PATTERN CHECKER ERROR
`read_err_status`	N/A	Reads JESD204C IP error status registers.
`clear_err_status`	N/A	Clears JESD204C IP error status registers
`read_rx_status0`	N/A	Reads JESD204C IP `rx_status0` register.
`read_tx_status0`	N/A	Reads JESD204C IP `tx_status0` register.
`read_rx_syncn_sysref_ctrl`	N/A	Reads JESD204C IP `syncn_sysref_ctrl` register.
`wait_seconds`	{integer}	Waits for {integer} seconds.
`wait_minutes`	{integer}	Waits for {integer} minutes.
`run_load_PMA_configuration`	N/A	Loads transceiver calibration presets.
`load_adaptation_PMA_configuration`	{0,1}	• 0: Disables internal serial loopback. • 1: Enables internal serial loopback. 1. Runs set operation mode. Checks if loopback and PRBS settings match the design requirements. 2. Polls registers 0x207, 0x80 = operation passed, and 0x81 = operation failed. 3. Runs load PMA configuration and load PMA configuration status check (0x40144 and 0x40143) if you turn on the **Enable PMA configuration loading** parameter in the parameter editor. 4. Starts calibration. Checks if loopback mode, load recipe enable, and PRBS settings match the design requirements. If PRBS is disabled, ensure that there is data sent from the core before starting calibration.. 5. Polls registers 0x207, 0x80 = operation passed, and 0x81 = operation failed. 6. Runs `check_cal_stat`. • Sets registers 0x203, 0x202, 0x201, and 0x200 to 0x97000001 • Polls registers 0x207, 0x80 = operation passed, and 0x81 = operation failed. • Reads register 0x204. 0x80 indicates successful calibration.

3.5.1. Board Connectivity

If you are performing hardware testing on the selected Intel development kits, generate the design example with the appropriate target development kit selected.

Refer to the instructions in Generating the Design on page 8.

Note: Running the hardware test with the design generated as-is is only possible when the JESD204C Intel FPGA IP is configured in duplex data path mode (i.e. with both TX and RX data paths present). Make your own modifications to the design to run the hardware test if generating a simplex data path design.

Table 21. Intel Stratix 10 Signal Integrity Development Kit Board Connectivity

The generated design has pre-assigned pins that target the relevant boards. The table describes the board connectivity of key design ports for all supported target development kits.

Port Name	Port Description	Board Component	Component Description
global_rst_n	Global reset	U13C	User PB0 push-button
refclk_core	Core PLL reference clock input	U3	Si5341 clock generator (OUT2)
refclk_xcvr	Transceiver reference clock input	U3	Si5341 clock generator (OUT8)
mgmt_clk	Control clock	U3	Si5341 clock generator (OUT3) (100 MHz)
perst_clk	Control clock	U3	Si5341 clock generator (OUT0) (100 MHz)—supplied through SMA cable from the J5/J6 port or to the J30/J32 SMA port
tx_serial_data	TX serial data	U32-1 and U75-1	Intel Stratix 10 E-tile banks - 8B (QSFPDD1x2)
rx_serial_data	RX serial data	U32-1 and U75-1	Intel Stratix 10 E-tile banks - 8B (QSFPDD1x2)
user_led[0]	GPIO- SPI programming done	U39G	Banks 2 L/M/N (G31)
user_dip[0]	GPIO- internal serial loopback enable	U39G	Banks 2 L/M/N (G23)

Related Information

Intel Stratix 10 TX Transceiver Signal Integrity Development Kit User Guide

4. Document Revision History for the JESD204C Intel Stratix 10 FPGA IP Design Example User Guide

Document Version	Intel Quartus Prime Version	IP Version	Changes
2019.08.01	19.2	1.0.0	Initial release.