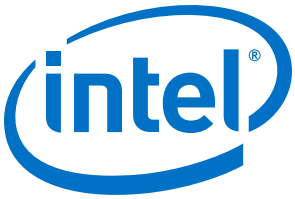


JESD204B Intel® Agilex™ FPGA IP Design Example User Guide

Updated for Intel® Quartus® Prime Design Suite: **19.3**

IP Version: **19.2.0**

Subscribe
Send Feedback

UG-20256 | 2019.09.30
Latest document on the web: **PDF** | **HTML**

Contents

1. About the JESD204B Intel® Agilex™ FPGA IP Design Example User Guide

This user guide provides the features, usage guidelines, and detailed description about the design examples for the JESD204B Intel® FPGA IP using Intel Agilex™ 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 JESD204B Intel FPGA IP design example.

Table 1. Related Documents

Reference	Description
JESD204B Intel FPGA IP User Guide	This document provides information about the JESD204B Intel FPGA IP.
JESD204B Intel FPGA IP Release Notes	This document provides release information for the JESD204B Intel FPGA IP.
Intel Agilex Device Data Sheet	This document describes the electrical characteristics, switching characteristics, configuration specifications, and timing for Intel Agilex devices.

Acronyms and Glossary

Table 2. Acronym List

Acronym	Expansion
LMFC	Local Multiframe 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

continued...

Acronym	Expansion
CSR	Control and status register
CRU	Clock and Reset Unit
ISR	Interrupt Service Routine
FIFO	First-In-First-Out
SERDES	Serializer Deserializer
ECC	Error Correcting Code
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
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 8B/10B encoder and output from the decoder
Data Rate	Effective data rate per lane for serial link Data Rate = Sampling rate per converter x M x N' x (10/8)/L *Note:* Sampling rate in Msps (Mega samples per second); Data rate in Mbps (Megabits per second)
Link Clock	The associated parallel data bus is 40 bits wide. Link Clock = Data Rate/40.
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.
Samples per frame clock	Samples per clock, the total samples in frame clock for the converter device.
LMFC	Local multiframe clock is counter generated from the link clock and depends on the F and K parameters. LMFC Period = (FxK/4) x Link Clock Period; the value of FxK must be divisible by 4.

continued...

Term	Description
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	Inter-device links with 2 or more converter devices.

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. JESD204B Intel FPGA IP Design Example Quick Start Guide

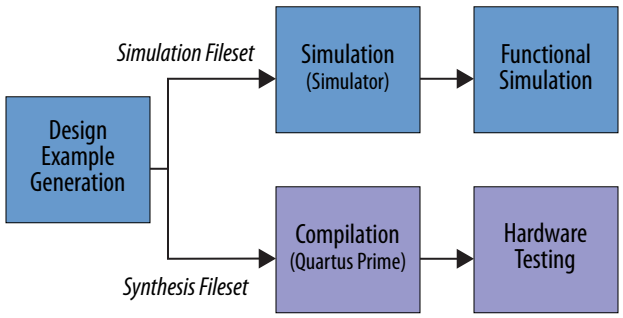
The JESD204B Intel FPGA IP design examples for Intel Agilex devices feature a simulating testbench and a hardware design that supports compilation.

The JESD204B Intel FPGA IP provides two preset settings for Intel Agilex devices in duplex mode.

- JESD204B design example for L=2, M=2, F=2, with data rate of 6.144 Gbps
- JESD204B design example for L=8, M=8, F=8, with data rate of 6.144 Gbps

You can generate the JESD204B design examples through the IP catalog in the Intel Quartus® Prime Pro Edition software.

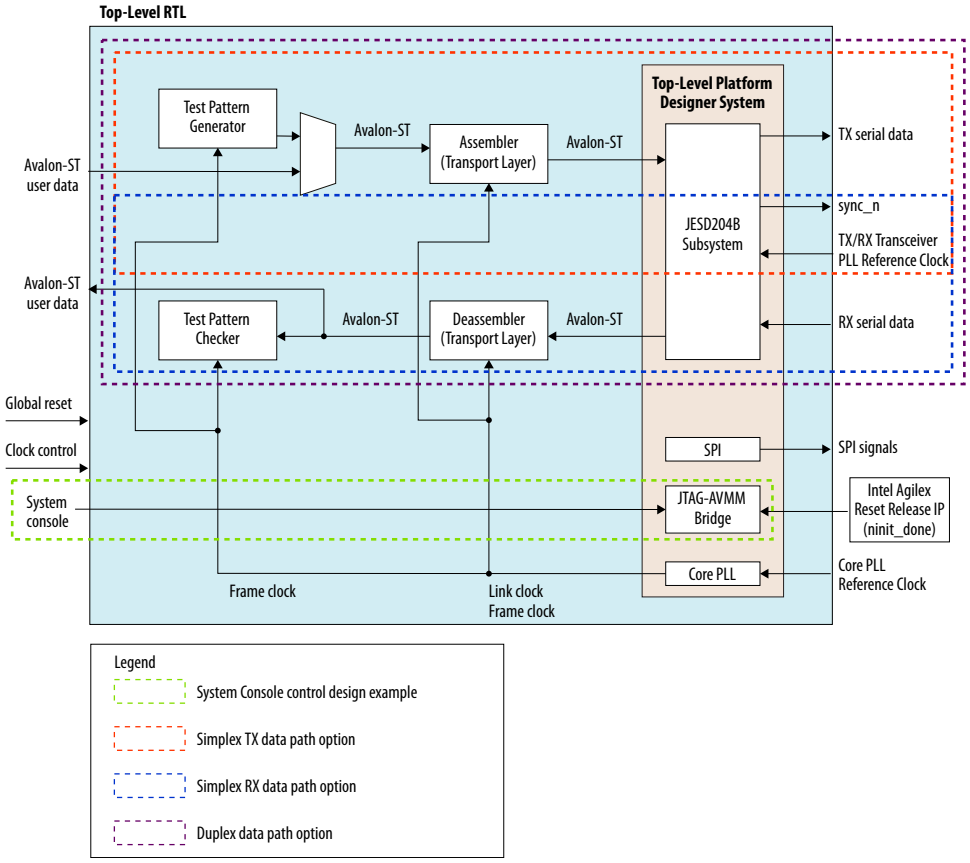
Figure 1. Development Stages for the Design Example



Note: Hardware testing will only be available for Intel Agilex design examples in a future release of the Intel Quartus Prime Pro Edition software.

2.1. Design Example Block Diagram

Figure 2. JESD204B Design Example High-level Block Diagram



The design example consists of the following modules:

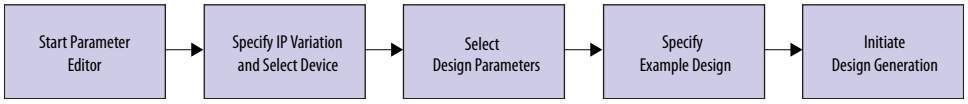
- Platform Designer system
  - JESD204B subsystem
  - JTAG to Avalon master bridge—for System Console Control design example only
  - Parallel I/O (PIO) controller
  - Core PLL
  - Serial Port Interface (SPI)—master module
- Test pattern generator (For duplex and simplex TX data path only)
- Test pattern checker (For duplex and simplex RX data path only)
- Assembler—TX transport layer (For duplex and simplex TX data path only)
- Deassembler—RX transport layer (For duplex and simplex RX data path only)

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 19.3
- ModelSim*, Xcelium*, NCSim (Verilog only), or VCS*/VCS MX simulator

2.3. Generating the Design



To generate the design example from the IP parameter editor:

1. Create a project targeting Intel Agilex device family and select a desired device.
2. In the IP Catalog, **Tools ➤ IP Catalog**, select **JESD204B 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 and compilation.

Related Information

- Presets on page 14
- Supported Configurations on page 15

2.3.1. Design Example Parameters

The JESD204B Intel FPGA IP parameter editor includes a **Design Example** tab for you to specify certain parameters before generating the design example.

Table 5. Parameters in the Example Design Tab

Parameter	Options	Description
Available Example Designs	None (Default)	No design examples selected.
	System Console Control	Design example with System Console control.
Example Design Files	Simulation	Generate simulation fileset.
	Synthesis	Generate synthesis fileset.
Generated HDL Format for Simulation	Verilog (Default)	Verilog HDL format for entire simulation fileset.

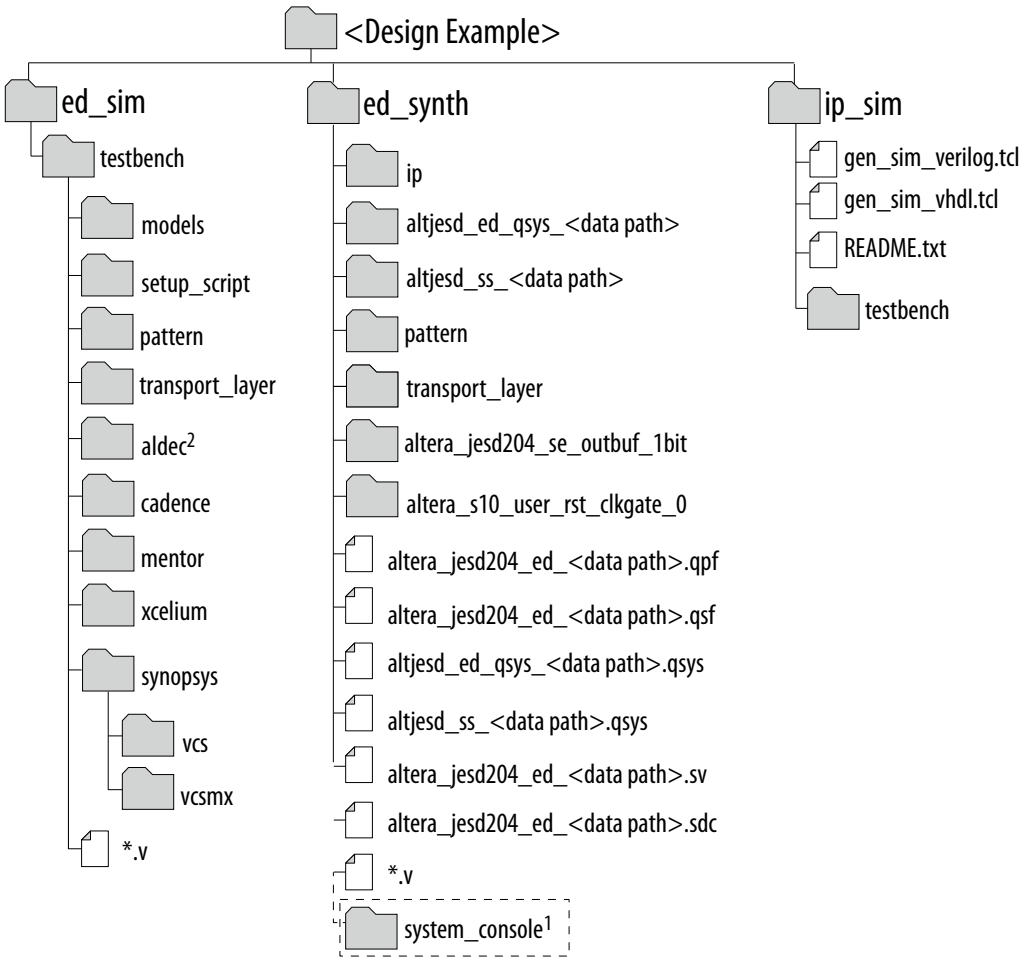
continued...

Parameter	Options	Description
	VHDL	VHDL format for generated top-level wrapper file set.
Generated HDL Format for Synthesis	Verilog (Default)	Verilog HDL format for synthesis fileset.
Example Design Customizations	Generate 3-wire SPI module	Check to enable 3-wire SPI interface instead of 4-wire SPI interface.

2.3.2. Directory Structure

The JESD204B design example directories contain generated files for the design examples.

Figure 3. Directory Structure for JESD204B Intel Agilex Design Example



Note:

1. *Directory 'system_console' only generated when 'Data Path Only' design example is generated.*
2. *Intel Quartus Prime Pro Edition software does not support the Aldec Riviera-PRO simulator .*

Table 6. Directory Files

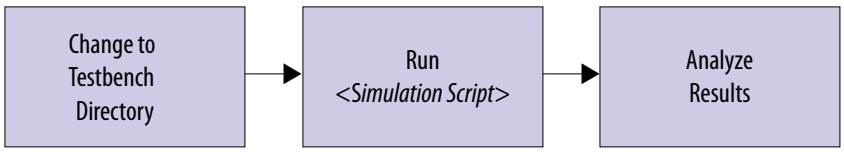
Directory/File	Description
ed_sim	The folder that contains simulation testbench files.
ed_sim/testbench/models	The folder that contains the testbench and source files.
ed_sim/testbench/setup_scripts	The folder that contains the test flow setup scripts.
ed_sim/testbench/pattern	The folder that contains the source files for the pattern generator/checker.
ed_sim/testbench/transport_layer	The folder that contains the source files for the transport layer.
ed_sim/testbench/cadence	The folder that contains the test flow run scripts for NCSim simulator. Also serves as the working directory for the simulator.
ed_sim/testbench/xcelium	The folder that contains the test flow run scripts for Xcelium Parallel simulator. Also serves as the working directory for the simulator.
ed_sim/testbench/mentor	The folder that contains the test flow run scripts for ModelSim simulator. Also serves as the working directory for the simulator.
ed_sim/testbench/synopsys/vcs	The folder that contains the test flow run scripts for VCS simulator. Also serves as the working directory for the simulator.
ed_sim/testbench/synopsys/vcsmx	The folder that contains the test flow run scripts for VCS MX simulator. Also serves as the working directory for the simulator.
ed_synth	The folder that contains design example synthesizable components.
ed_synth/altera_s10_user_rst_clkgate_0	The folder that contains the Platform Designer-generated modules of the Reset Release IP.
ed_synth/altera_jesd204_se_outbuf_1bit	The folder that contains the Platform Designer-generated modules of the bidirectional I/O buffer for the 3-wire SPI interface.
ed_synth/ip	The folder that contains Platform Designer-instantiated IP modules.
ed_synth/altjesd_ed_qsys_<data path>	The folder that contains Platform Designer-generated modules from the altjesd_ed_qsys_<data path>.qsys system.
ed_synth/altjesd_ss_<data path>	The folder that contains Platform Designer-generated modules from the altjesd_ss_<data path>.qsys system.
ed_synth/pattern	The folder that contains the source files for the pattern generator/checker.
ed_synth/transport_layer	The folder that contains the source files for the transport layer
ed_synth/altera_jesd204_ed_<data path>.qpf ed_synth/altera_jesd204_ed_<data path>.qsf	Intel Quartus Prime project and settings files
ed_synth/altjesd_ed_qsys_<data path>.qsys	Platform Designer top level system
ed_synth/altjesd_ss_<data path>.qsys	Platform Designer subsystem
ed_synth/altera_jesd204_ed_<data path>.sv	Top level HDL source file

continued...

Directory/File	Description
ed_synth/altera_jesd204_ed_<data path>.sdc	Top level design constraints file
ed_synth/system_console	The folder that contains all files necessary to run scripts in System Console (See Design Example Files for more details on folder content.)
*.v	Miscellaneous source files
ip_sim	The folder that contains the simulation script to generate the JESD204B IP Verilog/VHDL simulation model.

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>/ed_sim/testbench/<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 run_tb_top.sh
ModelSim	vsim -do run_tb_top.tcl
VCS	sh run_tb_top.sh
VCS MX	sh run_tb_top.sh
Xcelium Parallel	sh run_tb_top.sh

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

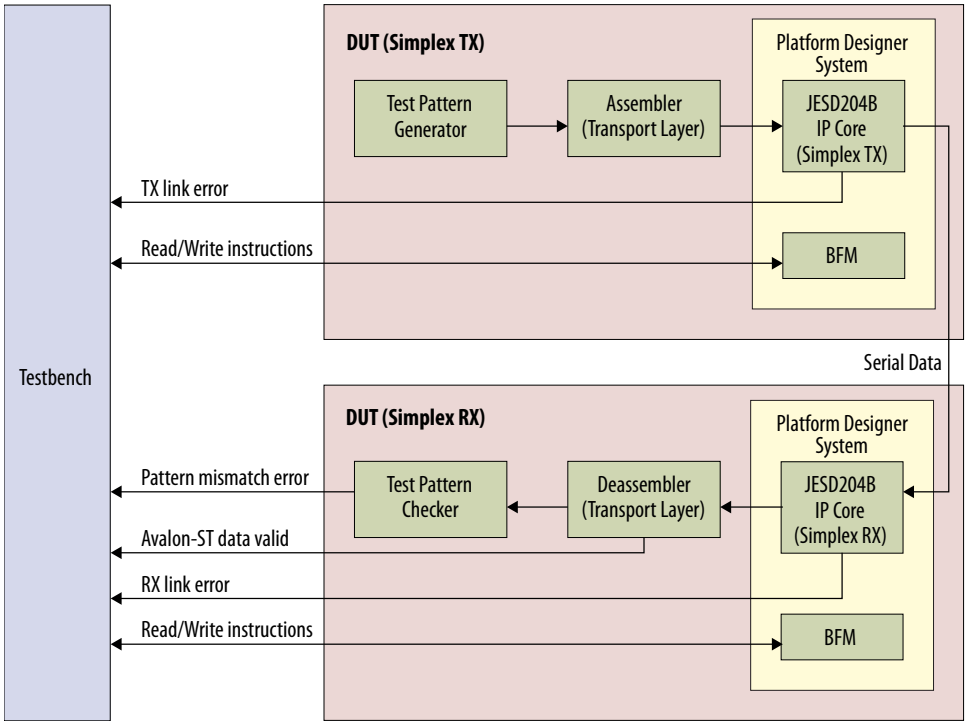
Related Information

Registers in the JESD204B Intel FPGA IP User Guide

2.4.1. Testbench

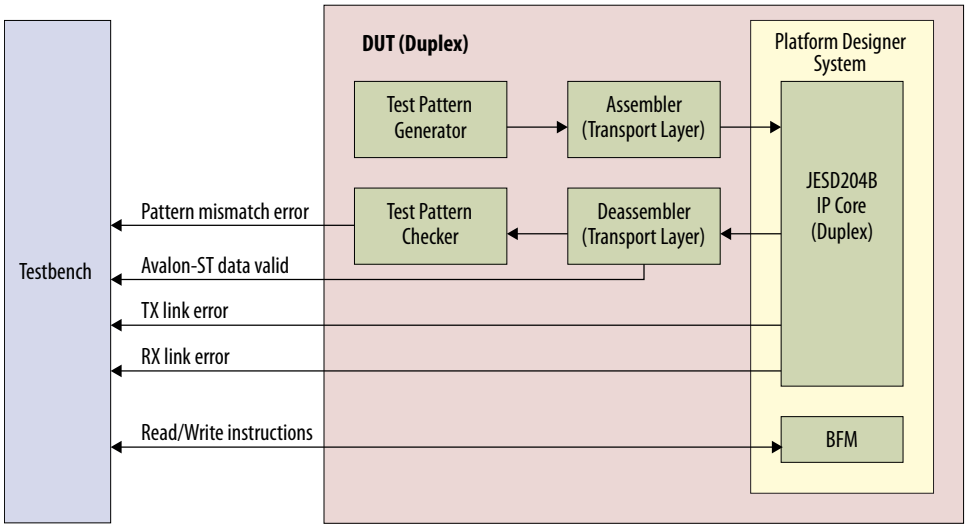
The simulation design-under-test (DUT) is the generated design example which includes a synthesizable pattern generator and checker. The figures below show the testbench block diagram for simplex and duplex options.

Figure 4. Simulation Testbench Block Diagram (Simplex TX or RX)



Note: Both simplex TX and simplex RX design examples generate the same testbench. The testbench instantiates two DUTs: one simplex TX DUT, one simplex RX DUT. The TX serial data output of the simplex TX DUT is connected to the RX serial data input of the simplex RX DUT. The testbench issues separate Avalon Memory-Mapped (Avalon-MM) read/write instructions to the simplex TX and simplex RX DUTs respectively.

Figure 5. Simulation Testbench Block Diagram (Duplex)



The simulation flow replaces the JTAG to Avalon master bridge module in the Platform Designer system of the System Console Control design example with the Avalon-MM master bus functional model (BFM). This BFM enables a testbench to send Avalon-MM read/write commands to the design example registers to mimic the functionality of System Console.

The testbench provided in the simulation flow (`/testbench/models/tb_top.sv`) executes the following steps:

1. Reset DUT.
2. Initialize BFM.
3. Execute Avalon-MM commands to initialize the DUT in the following mode:
   - Internal serial loopback mode (for duplex option only)
   - Pattern generator/checker set to PRBS pattern
4. Wait for DUT to initialize to user mode.
5. Report JESD204B link status.

When simulation ends, the following messages are shown at end.

Table 7. Simulation Messages and Description

Message	Description
`Pattern Checker(s): Data error(s) found!`	Pattern mismatch errors found on the pattern checker
`Pattern Checker(s): OK!`	No errors found on the pattern checker
`Pattern Checker(s): No valid data found!`	No valid data received by pattern checker
`JESD204B Tx Core(s): Tx link error(s) found!`	Link errors reported by JESD204B IP TX
`JESD204B Tx Core(s): OK!`	No link errors reported by JESD204B IP TX
`JESD204B Rx Core(s): Rx link error(s) found!`	Link errors reported by JESD204B IP RX
`JESD204B Rx Core(s): OK!`	No link errors reported by JESD204B IP RX
`TESTBENCH_PASSED: SIM PASSED!`	Overall simulation passed
`TESTBENCH_FAILED: SIM FAILED!`	Overall simulation failed

3. Detailed Description for the JESD204B Design Example

The JESD204B 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 for both Base and PHY variant. You can choose Base only or PHY only variant but the IP would generate the design example for both Base and PHY.

3.1. Presets

Standard presets allow instant entry of pre-selected parameter values in the **IP** and **Example Design** tabs. Select the presets at the lower right window in the parameter editor.

The presets are applicable for JESD204B IP configurations that generate design examples. You can select one of the presets available for your target device to quickly generate a design example without having to set each parameter in the IP tab and verify that the specified parameters match the supported configurations. You can manually change any of the IP and example design parameters in the Platform Designer user interface after selecting a preset. However, you must ensure that your parameter selection falls within the supported configuration ranges detailed in Supported Configurations on page 15 for design example to generate successfully.

Note: Selecting a preset overwrites any pre-existing parameter selections for the IP core under the IP tab.

Table 8. Preset Settings

JESD204B IP Parameters	Preset 1 JESD204B Example Design (LMF = 222, 6.144 Gbps)	Preset 2 JESD204B Example Design (LMF = 888, 6.144 Gbps)
Wrapper Options	Both Base and PHY	Both Base and PHY
Data Path	Duplex	Duplex
JESD204B Subclass	1	1
Data Rate	6144 Mbps	6144 Mbps
PCS Option	Enabled Hard PCS	Enabled Hard PCS
Bonding Mode	Non-bonded	Non-bonded
PLL/CDR Reference Clock Frequency	153.6 MHz	153.6 MHz
Enable Bit Reversal and Byte Reversal	No	No

continued...

JESD204B IP Parameters	Preset 1 JESD204B Example Design (LMF = 222, 6.144 Gbps)	Preset 2 JESD204B Example Design (LMF = 888, 6.144 Gbps)
Enable Transceiver Dynamic Reconfiguration	No	No
L	2	8
M	2	8
Enable manual F configuration	No	Yes
F	2	8
N	16	12
N’	16	12
S	1	5
K	16	32
Enable Scramble (SCR)	No	No
CS	0	0
CF	0	0
High Density User Data Format (HD)	0	0
Enable Error Code Correction (ECC_EN)	No	No

3.2. Supported Configurations

The design examples only support a limited set of JESD204B IP parameter configurations. The IP parameter editor allows you to generate a design example only if the parameter configurations matches the following table.

Note: If you are not able to generate a design example that fully matches your desired parameter settings, choose the closest allowable parameter values for generation. Modify the post-generated design parameters manually in the Intel Quartus Prime software to match your desire parameter settings. Refer to the *JESD204B Intel FPGA IP User Guide* for more details on the rules and ranges that govern each IP and transport layer parameter. Refer to *Customizing the Design Example* for more information about customizing the design example.

Table 9. Supported JESD204B IP Core Parameter Configurations

Table lists the parameters for the JESD204B IP. The JESD204B IP parameters are governed by various rules and ranges that are described in the *JESD204B Intel FPGA IP User Guide*. Please refer to the *JESD204B Intel FPGA IP User Guide* for more details on the legal parameter values. The value ranges given below should be considered as a subset of the allowable values described in the *JESD204B Intel FPGA IP User Guide*.

JESD204B IP Parameters	Values
Wrapper Options	Both Base and PHY
Data Path	• Receiver • Transmitter • Duplex
JESD204B Subclass	1
Data Rate	Any valid value(1)

continued...

JESD204B IP Parameters	Values
PCS Option	• Enabled Hard PCS • Enabled Soft PCS
Bonding Mode	• Bonded • Non-bonded
PLL/CDR Reference Clock Frequency	Any valid value
Enable Bit Reversal and Byte Reversal	Any valid value
L	• 1 • 2 • 4 • 6(2) • 8
M	• 1 • 2 • 3(3) • 4 • 8 • 16 • 32
Enable manual F configuration	• No • Yes only for the following configuration: L=8, M=8, F=8, S=5, N'=12, N=12
F	• Auto calculated • Manual F configuration only allowed for the following configuration: L=8, M=8, F=8, S=5, N'=12, N=12
N	Integer, range 12 – 16
N'	• 16 • 12 only for the following configurations: — L=8, M=8, F=8, S=5, N=12 — F=3, N'=12, N=12
S	Any valid value
K	Any valid value
Enable Scramble (SCR)	Any valid value
CS	Integer, range 0 – 3
CF	0
High Density User Data Format (HD)	• 0 • 1 only for F=1
Enable Error Code Correction (ECC_EN)	Any valid value

(1) Refer to *JESD204B Intel FPGA IP User Guide* for more details on maximum and minimum data rates for your target device.

(2) L=6 is only allowed when F=1

(3) M=3 is only allowed for L=6

Related Information

JESD204B Intel FPGA IP User Guide

3.3. System Components

The JESD204B 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.

3.3.1. JESD204B Subsystem in Platform Designer

The JESD204B subsystem instantiates the following modules:

- JESD204B Intel FPGA IP
- Reset sequencer
- Avalon-MM bridge

JESD204B IP

The generated design example is a self-contained system with its own JESD204B IP core instantiation that is separate from the IP core that is generated from the **IP** tab. The JESD204B IP base core and PHY layer connect to System Console through the Avalon-MM interconnect. The JESD204B IP core uses three separate Avalon-MM ports:

- Base core TX data path—For accessing the TX CSR
- Base core RX data path—For accessing the RX CSR
- PHY layer—For accessing the transceiver PHY CSR

The structure of the design example varies depending on the values of these JESD204B IP core parameters:

- Data path:
  - Duplex—Both TX and RX data paths and CSR interfaces present
  - TX only—Only TX data path and CSR interface present
  - RX only—Only RX data path and CSR interface present

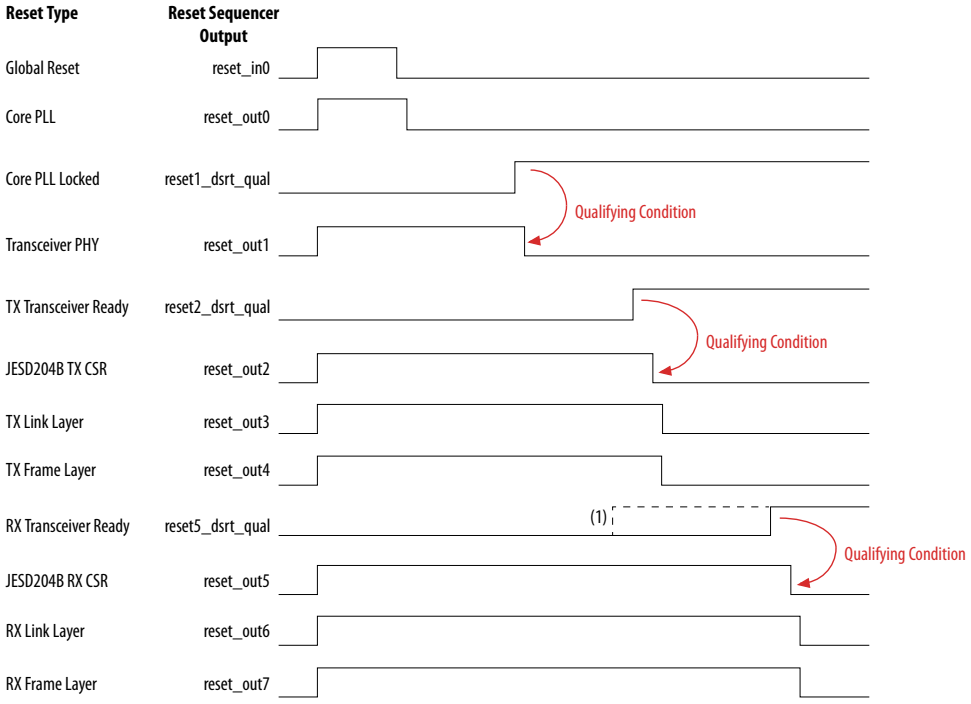
Reset Sequencer

The reset sequencer is a standard Platform Designer component in the **IP Catalog** standard library. The reset sequencer generates the following system resets to reset various modules in the system:

1. Core PLL reset—resets the core PLL
2. Transceiver reset—resets the JESD204B IP core PHY module
3. TX/RX JESD204B IP core CSR reset—resets the TX/RX JESD204B IP core CSRs
4. TX/RX link reset—resets the TX/RX JESD204B IP core base module and transport layer
5. TX/RX frame reset—resets the TX/RX transport layer, upstream and downstream modules

The reset sequencer has hard and soft reset options. The hard reset port connects to the global reset input pin in the top level design. The soft reset is activated via Avalon-MM interface by TCL scripts (System Console control). When you assert a hard or soft reset, the reset sequencer cycles through all the various module resets based on a pre-set sequence. The figure below illustrates the sequence and also shows how the reset sequencer output ports correspond to the modules that are being reset.

Figure 6. Reset Sequence



Note:

(1) In the event that the RX transceiver ready (reset5_dsrt_qual) asserts before the TX transceiver ready (reset2_drst_qual), the RX CSR, RX link layer, and RX frame layer will remain in reset until TX CSR, TX link layer, and TX frame layer are out of reset.

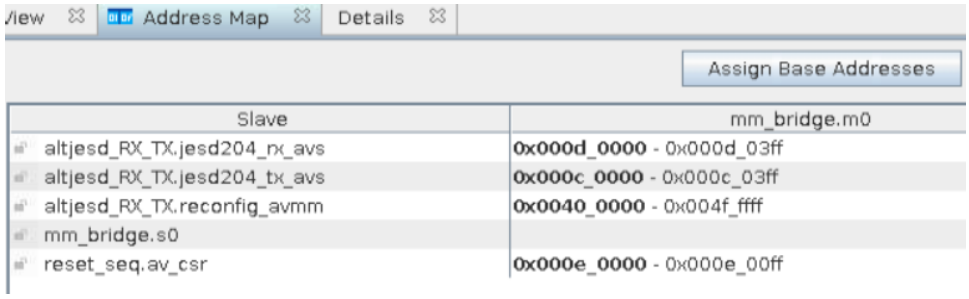
Avalon-MM Bridge

All the Avalon-MM submodules in the JESD204B subsystem are connected via Avalon-MM interconnect to a single Avalon-MM bridge. This bridge is the single interface for Avalon-MM communications into and out of the subsystem.

JESD204B Subsystem Address Map

Access the address map of the submodules in the JESD204B subsystem by clicking on the **Address Map** tab in the Platform Designer window.

Figure 7. JESD204B Subsystem Address Map



JTAG to Avalon Master Bridge

The JTAG to Avalon master bridge is a standard Platform Designer component in the IP Catalog standard library. This module provides a connection between a host system and the Platform Designer system via the respective physical interfaces; JTAG on the host system end and Avalon-MM on the Platform Designer system end. Host systems can initiate Avalon-MM transactions by sending encoded streams of bytes via JTAG interface. The module supports reads and writes, but not burst transactions.

3.3.3. Parallel I/O

Parallel I/O (PIO) modules provide general input/output (I/O) access from the Avalon master (JTAG to Avalon master bridge). There are two sets of 32-bit PIO registers:

- Status registers—input from the HDL components to the Avalon master
- Control registers—output from the Avalon master to the HDL components

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 10. Signal Connectivity for Status Registers

Bit	Signal
0	Core PLL locked
1	TX transceiver ready (for duplex and simplex TX data path only)
2	RX transceiver ready (for duplex and simplex RX data path only)
3	Test pattern checker data error (for duplex and simplex RX data path only)
4	TX link error (for duplex and simplex TX data path only)
5	RX link error (for duplex and simplex RX data path only)

Table 11. Signal Connectivity for Control Registers

Bit	Signal
0	RX serial loopback enable (for duplex data path only)
30	Global reset
31	SYSREF

3.3.4. Core PLL

The core PLL module generates the clocks for the FPGA core fabric. An IOPLL module is instantiated as core PLL.

The core PLL uses an external clock input as its reference clock to generate two derivative clocks from a single VCO:

- Link clock
- Frame clock

Table 12. Core PLL Ouputs

Clock	Formula	Description
Link Clock	Serial data rate/40	The link clock clocks the JESD204B IP core link layer and the link interface of the transport layer.
Frame Clock	Derived based on settings; refer to Table 13 on page 21.	The frame clock clocks the transport layer, test pattern generators and checkers, and any downstream modules in the FPGA core fabric.

For the frame clock, when the **F** parameter is 1, 2 or 3, the resulting frame clock frequency easily exceeds the capability of the core PLL to generate and close timing. The top level RTL file, (`altera_jesd204_ed_<data path>.sv`), defines the frame clock division factor parameters, `F1_FRAMECLK_DIV` (for cases with F = 1) and `F2_FRAMECLK_DIV` (for cases with F = 2). F = 3 uses a constant division factor of 2. This factor enables the transport layer and test pattern generator to operate at a divided factor of the required frame clock rate by widening the data width accordingly.

Note: For JESD204B IP design examples, `F1_FRAMECLK_DIV` is set to 4 and `F2_FRAMECLK_DIV` is set to 2.

These examples show how to derive the frame clock frequency:

Example 1: The actual frame clock for a serial data rate of 10 Gbps and F = 1 is:

```
(10000/(10 × 1)) / F1_FRAMECLK_DIV = 1000 / 4 = 250 MHz
```

Example 2: The actual frame clock for a serial data rate of 6 Gbps and F = 3 is:

```
(6000/(10 × 3)) / 2 = 200 / 2 = 100 MHz
```

Frame Clock and Link Clock Relationship

The frame clock and link clock are synchronous. For the derived F mode, the ratio of `link_clk` period to `frame_clk` period is given by this formula:

```
link_clk period to frame_clk period ratio = 32xL/(MxSxN')
```

Table 13. $f_{TXframe}$ and $f_{RXframe}$ for Different F Parameter Settings

- f_{TXlink} is the TX link clock frequency
- f_{RXlink} is the RX link clock frequency

F Parameter	$f_{TXframe}$(txframe_clk frequency)	$f_{RXframe}$(rxframe_clk frequency)
1	f_{TXlink}x(4/F1_FRAMECLK_DIV)	f_{RXlink}x(4/F1_FRAMECLK_DIV)
2	f_{TXlink}x(2/F2_FRAMECLK_DIV)	f_{RXlink}x(2/F2_FRAMECLK_DIV)
3	f_{TXlink}x(2/3)	f_{RXlink}x(2/3)
4	f_{TXlink}	f_{RXlink}
8	f_{TXlink}/2	f_{RXlink}/2

Note: The IOPLL is generated with the **Use Nondedicated Feedback Path** option being disabled (default setting). You can turn on the **Use Nondedicated Feedback Path** option in the IP parameter editor to utilize the clock resources efficiently after the design example is successfully generated. Refer to the *Clock Feedback Modes* section of *Clocking and PLL User Guide* for more information about this option.

Related Information

Clock Feedback Modes in the Intel Agilex Clocking and PLL User Guide

3.3.5. 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.

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.

For more details on the SPI master module, refer to the *JESD204B Intel FPGA IP User Guide*.

Related Information

JESD204B Intel FPGA IP User Guide

3.3.6. Transport Layer

The transport layer in the design example consists of an assembler at the TX path and a deassembler at the RX path. The transport layer for both the TX and RX path is instantiated in the top level RTL file, not in the Platform Designer project.

Note: When the simplex TX data path option is selected, only the assembler is instantiated in the design example. When the simplex RX data path option is selected, only the deassembler is instantiated in the design example. When the duplex data path option is selected, both assembler and deassembler is instantiated in the design example.

The transport layer provides the following services to the application layer (AL) and the data link layer (DLL):

- Assembler at the TX path:
  - Maps the conversion samples from the AL (through the Avalon-ST interface) to a specific format of non-scrambled octets, before streaming them to the DLL.
  - Reports AL error to the DLL if it encounters a specific error condition on the Avalon-ST interface during TX data streaming.
- Deassembler at the RX path:
  - Maps the descrambled octets from the DLL to a specific conversion sample format before streaming them to the AL (through the Avalon-ST interface).
  - Reports AL error to the DLL if it encounters a specific error condition on the Avalon-ST interface during RX data streaming.

The transport layer has many customization options and you can modify the transport layer RTL to customize it to your specifications. Furthermore, for certain parameters like L, F, and N, the transport layer shares the CSR values with the JESD204B IP core.

For more details on the implementation of the transport layer in RTL and customization options, refer to the *JESD204B Intel FPGA IP User Guide*.

Related Information

JESD204B Intel FPGA IP User Guide

3.3.7. Test Pattern Generator

Note: This module is only available in the design example when the duplex or simplex TX data path option is selected.

The test pattern generator generates either a parallel PRBS, alternate checkerboard, or ramp wave, and sends it to the transport layer during test mode. The test pattern generator is implemented in the top level RTL file, not in the Platform Designer project.

Related Information

JESD204B Intel FPGA IP User Guide

3.3.8. Test Pattern Checker

Note: This module is only available in the design example when the duplex or simplex RX data path option is selected.

The test pattern checker checks either a parallel PRBS, alternate checkerboard, or ramp wave from the transport layer during test mode and outputs an error flag if there are any data mismatches. The test pattern checker is implemented in the top level RTL file, not in the Platform Designer project.

Related Information

JESD204B Intel FPGA IP User Guide

3.4. Design Example Clock and Reset

The main reference clocks for the design example are `refclk_core` and `refclk_xcvr`. These clocks must be supplied from a single external source (i.e `refclk_core` and `refclk_xcvr` must be synchronous to one another). The `refclk_core` is the reference clock for the core PLL and the `refclk_xcvr` is the reference clock for the TX/RX transceiver PHY. The core PLL generates the `link_clk` and `frame_clk` from `refclk_core`.

The `link_clk` clocks the JESD204B IP core link layer and link interface of the transport layer. The `frame_clk` clocks the transport layer, test pattern generator and checker modules, and any downstream modules. An external source supplies a clock called the `mgmt_clk` to clock the Avalon-MM interfaces of Platform Designer components.

Table 14. Design Example Clocks

Clock	Description	Source	Modules Clocked
`refclk_core`	Reference clock for core PLL	External	Core PLL
`refclk_xcvr`	Reference clock for TX PLL and RX transceiver PHY	External	TX PLL and RX transceiver PHY
`link_clk`	Link layer clock	refclk_core	JESD204B IP core link layer, transport layer link interface
`frame_clk`	Frame layer clock	refclk_core	Transport layer, test pattern generator and checker, downstream modules
`mgmt_clk`	Control plane clock	External	Avalon®-MM interfaces

Table 15. Design Example Resets

Signal	Direction	Description
`global_rst_n`	Input	Global reset to reset entire controller. For all blocks, except JTAG-Avalon-MM.
`mgmt_rst_in_n`	Internal	Reset for Avalon-MM configuration access.

3.5. Design Example Signals

Table 16. System Interface Signals

Signal	Clock Domain	Direction	Description
Clocks and Resets			
`refclk_core`	—	Input	Reference clock for FPGA core modules.
`refclk_xcvr`	—	Input	Reference clock for transceiver PHY.
`mgmt_clk`	—	Input	Reference clock for all peripherals connected via Avalon-MM interconnect.
`global_rst_n`	`mgmt_clk`	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`.

Signal	Clock Domain	Direction	Description
Serial Data			
rx_serial_data[LINK*L-1:0] rx_serial_data_n[LINK*L-1:0]	refclk_xcvr	Input	Differential high speed serial input data. The clock is recovered from the serial data stream.
tx_serial_data[LINK*L-1:0] tx_serial_data_n[LINK*L-1:0]	refclk_xcvr	Output	Differential high speed serial output data. The clock is embedded in the serial data stream.

Signal	Clock Domain	Direction	Description
JESD204B			
sysref_out	mgmt_clk	Output	*SYSREF* signal for JESD204B Subclass 1 implementation.
sync_n_out	link_clk	Output	Indicates a SYNC_N from the receiver. This is an active low signal and is asserted 0 to indicate a synchronization request or error reporting.
tx_link_error	link_clk	Output	Error interrupt from JESD204B IP core indicating TX link error
rx_link_error	link_clk	Output	Error interrupt from JESD204B IP core indicating RX link error

Signal	Clock Domain	Direction	Description
Avalon- ST User Data			
avst_usr_din[LINK*TL_DATA_BUS_WIDTH-1:0]	frame_clk	Input	TX data from the Avalon-ST source interface. The TL_DATA_BUS_WIDTH is determined by the following formulas: • If F = 1, TL_DATA_BUS_WIDTH = F1_FRAMECLK_DIV*8*1*L*N/N_PRIME • If F = 2, TL_DATA_BUS_WIDTH = F2_FRAMECLK_DIV*8*2*L*N/N_PRIME • If F = 3, TL_DATA_BUS_WIDTH = 2*8*3*L*N/N_PRIME • If F = 4, TL_DATA_BUS_WIDTH = 8*4*L*N/N_PRIME • If F = 8, TL_DATA_BUS_WIDTH = 8*8*L*N/N_PRIME
avst_usr_din_valid[LINK-1:0]	frame_clk	Input	Indicates whether the data from the Avalon-ST source interface to the transport layer is valid or invalid. • 0—data is invalid • 1—data is valid
avst_usr_din_ready[LINK-1:0]	frame_clk	Output	Indicates that the transport layer is ready to accept data from the Avalon-ST source interface. • 0—transport layer is not ready to receive data • 1—transport layer is ready to receive data
avst_usr_dout[LINK*TL_DATA_BUS_WIDTH-1:0]	frame_clk	Output	RX data to the Avalon-ST sink interface. The TL_DATA_BUS_WIDTH is determined by the following formulas:

continued...

Signal	Clock Domain	Direction	Description
			• If F = 1, TL_DATA_BUS_WIDTH = F1_FRAMECLK_DIV*8*1*L*N/N_PRIME • If F = 2, TL_DATA_BUS_WIDTH = F2_FRAMECLK_DIV*8*2*L*N/N_PRIME • If F = 3, TL_DATA_BUS_WIDTH = 2*8*3*L*N/N_PRIME • If F = 4, TL_DATA_BUS_WIDTH = 8*4*L*N/N_PRIME • If F = 8, TL_DATA_BUS_WIDTH = 8*8*L*N/N_PRIME
avst_usr_dout_valid[LINK-1:0]	frame_clk	Output	Indicates whether the data from the transport layer to the Avalon-ST sink interface is valid or invalid. • 0—data is invalid • 1—data is valid
avst_usr_dout_ready[LINK-1:0]	frame_clk	Input	Indicates that the Avalon-ST sink interface is ready to accept data from the transport layer. • 0—Avalon-ST sink interface is not ready to receive data • 1—Avalon-ST sink interface is ready to receive data
avst_patchk_data_error [LINK-1:0]	frame_clk	Output	Output signal from pattern checker indicating a pattern check error.

Signal	Clock Domain	Direction	Description
SPI			
spi_MISO *Note:* When **Generate 3-Wire SPI Module** option is not enabled.	spi_SCLK	Input	Input data from external slave to the master.
spi_MOSI *Note:* When **Generate 3-Wire SPI Module** option is not enabled.	spi_SCLK	Output	Output data from the master to the external slaves.
spi_SDIO *Note:* When **Generate 3-Wire SPI Module** option is enabled.	spi_SCLK	Input/ Output	Output data from the master to external slave. Input data from external slave to master
spi_SCLK	mgmt_clk	Output	Clock driven by the master to slaves, to synchronize the data bits.
spi_SS_n[2:0]	spi_SCLK	Output	Active low select signal driven by the master to individual slaves, to select the target slave. Defaults to 3 bits.

3.6. JESD204B Design Example Status and Control Registers

The JESD204B design example registers use byte-addressing (32 bits).

Refer to the *JESD204B Registers* section in the *JESD204B Intel FPGA IP User Guide*.

Related Information

Registers in the JESD204B Intel FPGA IP User Guide

4. Document Revision History for the JESD204B Intel Agilex FPGA IP Design Example User Guide

Document Version	Intel Quartus Prime Version	IP Version	Changes
2019.09.30	19.3	19.2.0	Initial release.