IO CONFIGURATOR
(Beta Release)

Version 0.1

May 9, 2024

Copyright

Trademarks

Disclaimers

Contents

IP Summary

Introduction

IO Configurator is a configurable IP block designed to create IO Models including SERDES, BUFFERS, DELAYS and DDR. SERDES handles serial-to-parallel and parallel-to-serial conversion of multiple FPGA fabric signals to and from a single device I/O based on user clock settings. BUFFERS are used to buffer I/O signals coming into the fabric through IOPAD. DELAYS are used to add delay for the data ports coming into the fabric through BUFFERS. Double Data Rate (DDR) provide interfaces for high-speed data transfers.

Features

- Support four types of BUFFERS. i.e. Single_Ended, Differential, Tri-state and Differential-Tri-state.
- Support Pull-up and Pull-down IO modes to make logic low/high in the absence of an external connection.
- Support multiple IO models including SERDES, BUFFERS, DELAYS and DDR.
- Support SDR data rate for SERDES.
- Support multiple operation modes for SERDES like Dynamic Phase Alignment and Clock Data Recovery.
- Support width from 3 to 10 for Serialization/ Deserialization.
- Support multiple clock sourcing option for SERDES.
- Support user defined input clock frequency for SERDES.
- Support clock forwarding and clock phase for O_SERDES.
- Support Static and Dynamic delay adjustment for DELAYs.
- Support 0 to 63 tap delay values.

Overview

IO Configurator

The IO Configurator IP Core is a versatile tool for configuring IO Primitives. By offering pre-defined and configurable IO models, it eliminates the need for manual design and verification of low-level circuitry. It saves significant development time and effort. This flexibility allows users to tailor the IP core to their specific application needs. It streamlines the integration process and boosting development efficiency.

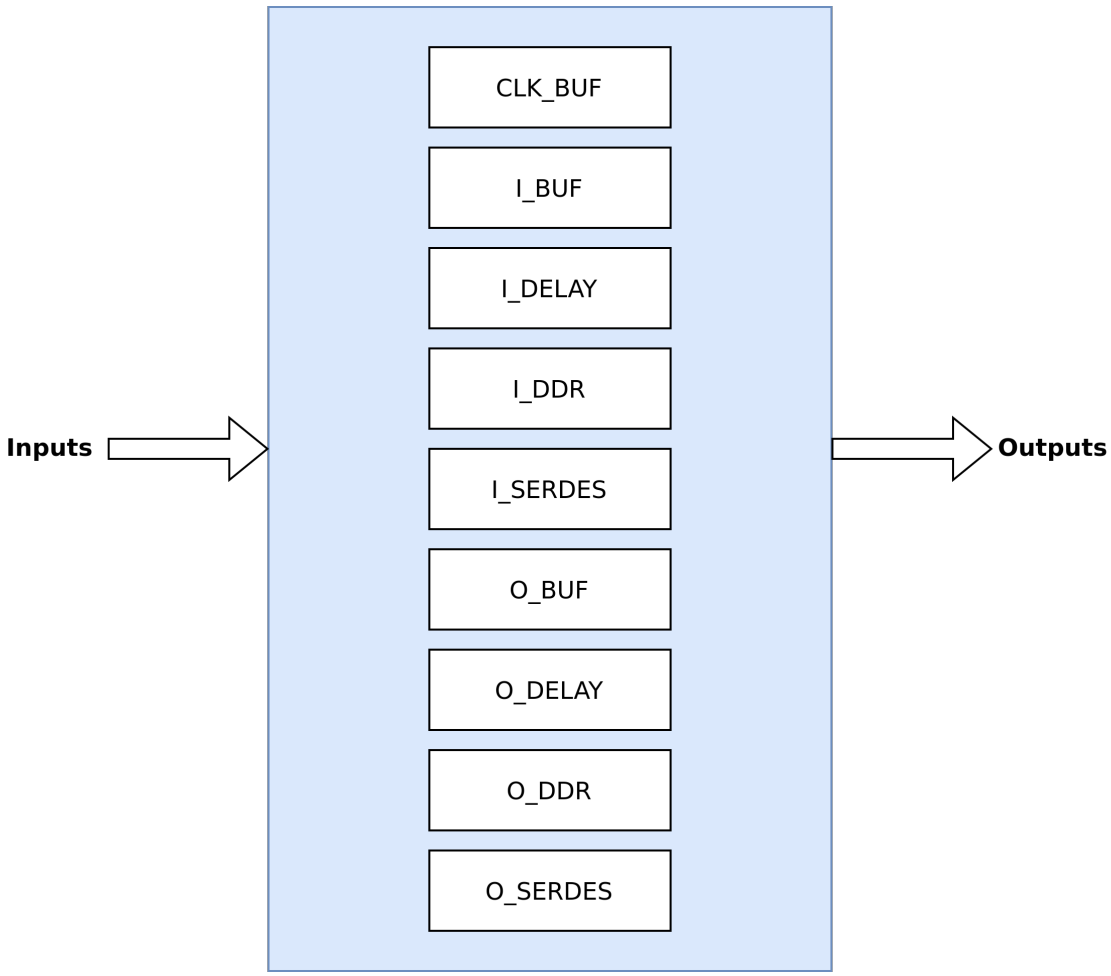

Figure 1: IO Configurator Block Diagram

IP Specification

IO Configurator provides a library of pre-defined and configurable IO models that act as building blocks for common IO functionalities. These models cover a wide range of functionalities: buffers (CLK_BUF, I_BUF, O_BUF) ensure clean and clear clock and data signals, delays (I_DELAY, O_DELAY) provide precise timing adjustments for optimal performance, DDR interfaces (I_DDR, O_DDR) maximize data transfer speeds, and serializers/deserializers (I_SERDES, O_SERDES) efficiently convert data formats for long-distance transmission. Each model is configurable and empowers user to create a customized IO solution that aligns user requirements.

IO Models

Each IO model is described as:

- **CLK_BUF:**
  Clock Buffer is a digital circuit element used to isolate and strengthen a clock signal. It can filter out any electrical noise or glitches present on the original clock signal. It ensures a clean and stable clock for downstream circuitry. User may configure CLK_BUF by providing IO_MODE parameter.

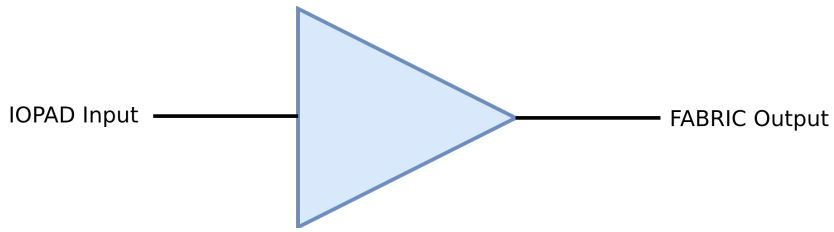

Figure 2: CLK_BUF

- **I_BUF:**
  Input Buffer is a digital circuit element used to isolate and condition an input signal. It acts as a barrier between the external source of the input signal and the internal circuitry of the programmable device. It can filter out any electrical noise or glitches present on the incoming signal, ensuring a clean and stable signal for the internal logic. User may configure I_BUF as a Single Ended or Differential buffer.

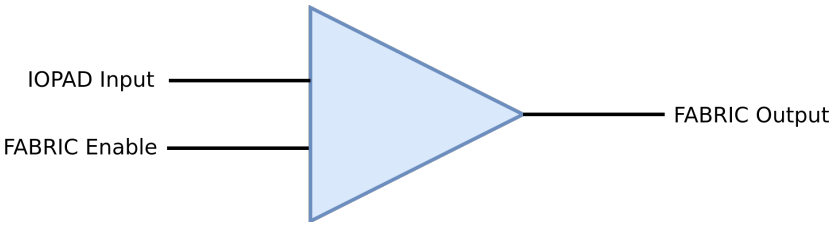

Figure 3: I_BUF

- **I_DELAY:**
  Input Delay is a digital circuit element used to intentionally introduce a controlled time

delay to an incoming input signal. It can be used to adjust the arrival time of the input signal relative to the clock edge to meet the setup and hold time requirements. It can also be used to fine-tune the timing of different data bits within an input signal to achieve proper alignment for processing within the device. User may configure I_DELAY by providing delay type, tap delay value and clock source.

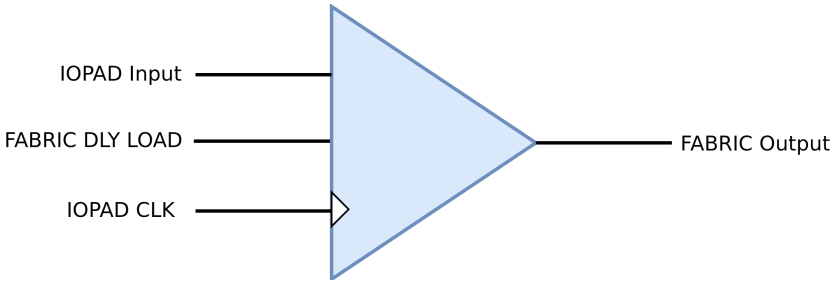

Figure 4: I_DELAY

- **I_DDR:**
  Input Double Data Rate is a type of input primitive used to facilitate high-speed data transfers. The key advantage of an I_DDR interface is its ability to transfer data on both the rising and falling edges of the clock signal. This effectively doubles the data throughput compared to a traditional interface that uses only the rising edge of the clock. The increased data transfer rate often translates to higher power consumption compared to single-data-rate interfaces. User may configure I_DDR by providing clock source and input mode.

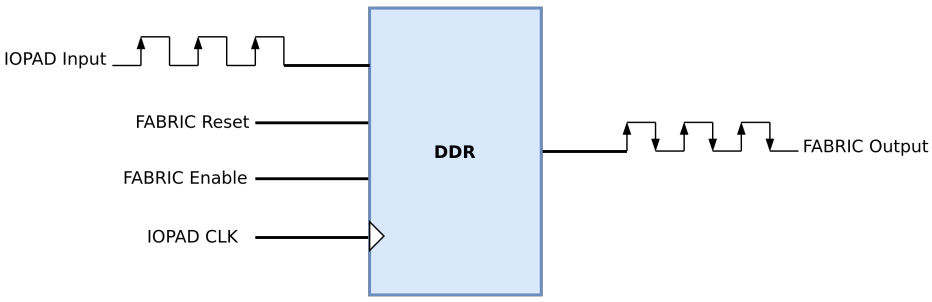

Figure 5: I_DDR

- **I_SERDES:**
  Input Serializer/Deserializer is a digital circuit element used for efficient data transmission between your programmable device and the external world. It performs the crucial task of converting an incoming serial data stream into parallel data for processing by the device. It facilitates this high-speed data transmission, making it ideal for applications like inter-chip communication, networking, and high-speed data acquisition systems. User may configure I_SERDES by providing operational mode, clock source, clock frequency, delay adjustment and data width for serialization.

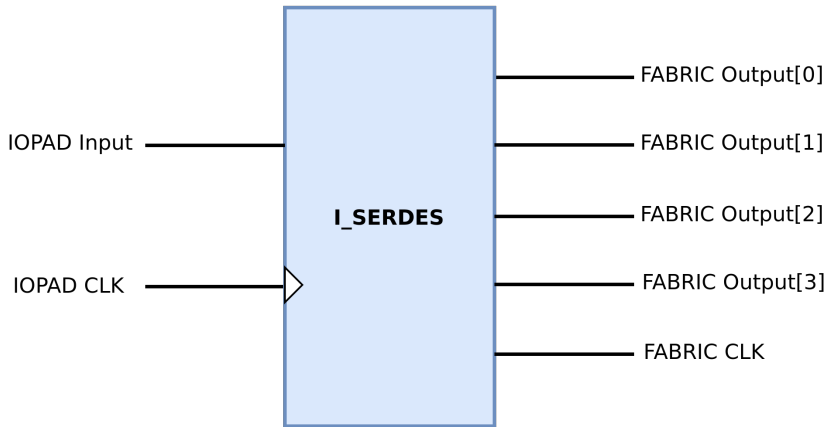

Figure 6: I_SERDES

- **O_BUF:**
  Output Buffer is a digital circuit element used to isolate and strengthen an output signal generated within programmable device. It acts as a barrier between the internal logic of the device and the external circuit that the signal needs to drive. It can filter out any electrical noise generated by the internal circuitry, ensuring a clean and stable signal is transmitted to the external world. User may configure O_BUF by proving output mode and output type (Single Ended, Differential, Tri-state or Differential Tri-State).

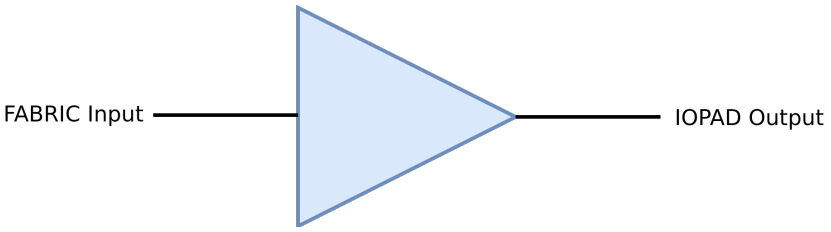

Figure 7: O_BUF

- **O_DELAY:**
  Output Delay is a digital circuit element used to intentionally introduce a controlled time delay to an outgoing signal generated by programmable device. It plays a crucial role in achieving proper timing synchronization within your design. It can be used for fine-tuning the timing of different bits within an output signal. This can be crucial for specific communication protocols where data needs to be transmitted in a specific order or format. It ensures proper alignment of data for the receiving device. User may configure O_DELAY by providing input mode, delay type, tap delay value and clock source.

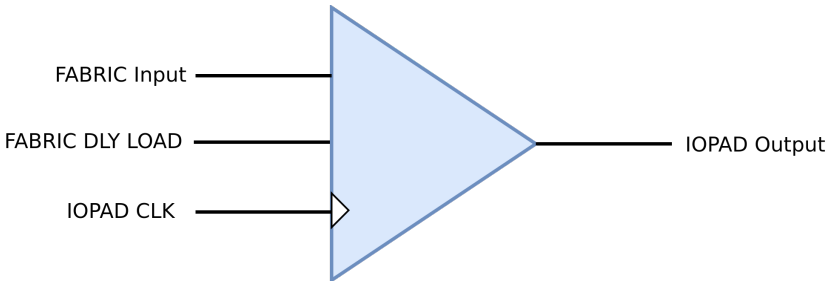

Figure 8: O_DELAY

- **O_DDR:**
  Output Double Data Rate is a type of output primitive to facilitate high-speed data transfers from programmable device to the external world. It offers the key advantage of transmitting data on both the rising and falling edges of the clock signal. This effectively doubles the data throughput compared to a traditional interface that uses only the rising edge for data transmission. This is particularly beneficial for applications where you need to send large amounts of data quickly, such as video output, high-speed data acquisition systems, and real-time signal processing. User may configure O_DDR by providing input mode and clock source.

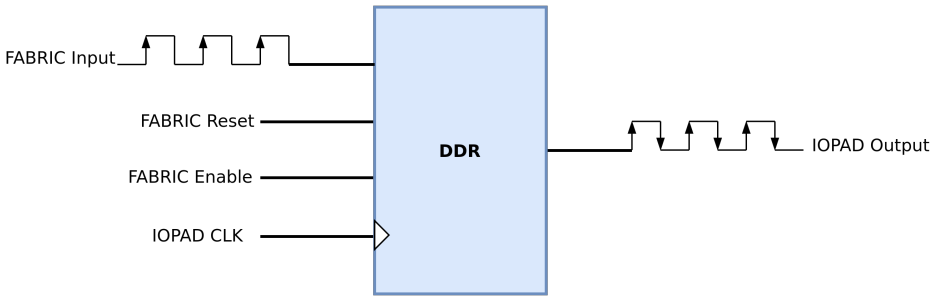

Figure 9: O_DDR

- **O_SERDES:**
  Output Serializer/Deserializer is a digital circuit element used for efficient data transmission from programmable device to the external world. It performs the crucial task of converting parallel data streams generated within programmable device into a serial data stream for transmission over a single wire or differential pair. This is particularly beneficial for long-distance communication as it reduces the number of wires required, simplifying cable design and minimizing signal integrity issues. User may configure O_SERDES by providing input mode, clock source, clock forwarding, delay adjustment, tap delay value and width of deserialization.

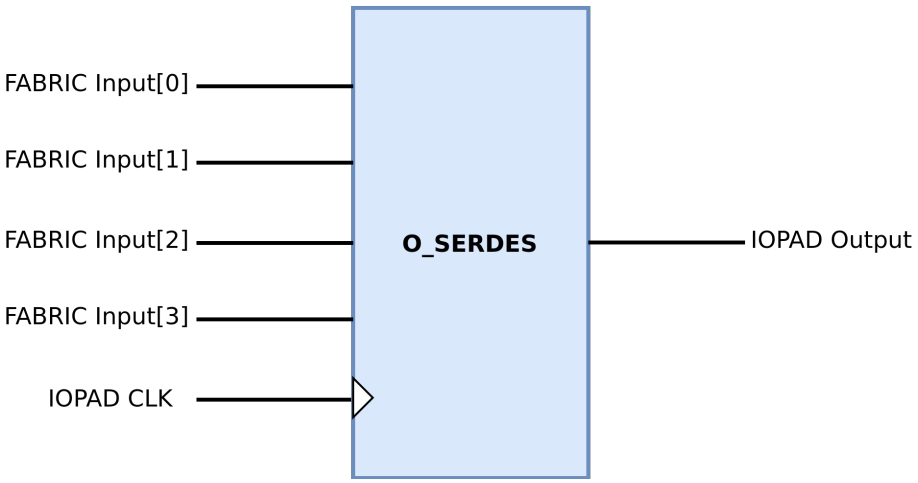

Figure 10: O_SERDES

IP Support Details

The Table 1 gives the support details for IO Configurator.

Compliance		IP Resources				Tool Flow		
Device	**Interface**	**Source Files**	**Constraint File**	**Testbench**	**Simulation Model**	**Analyze and Elaboration**	**Simulation**	**Synthesis**
VIRGO	Native	Verilog	-	-	-	Raptor	Raptor	Raptor

Table 1: Support Details

Resource Utilization

The resource utilization of the IO Configurator IP Core depends directly on the selected IO Model.

Ports

Table 2 lists the top interface ports of the IO Configurator.

Signal Name	Width	I/O	Description
CLK_BUF			
IOPAD_I	1	I	Input signal coming from IOPAD
FABRIC_O	1	O	Output Signal for FABRIC
I_BUF			
IOPAD_I_P	1	I	Positive end differential input signal
IOPAD_I_N	1	I	Negative end differential input signal
IOPAD_I	1	I	Input signal from IOPAD
FABRIC_EN	1	I	Input enable signal
FABRIC_O	1	O	Output Signal for FABRIC
I_DELAY			
IOPAD_CLK	1	I	Input clock from IOPAD
IOPAD_PLL_REF_CLK	1	I	PLL reference clock from IOPAD
IOPAD_I	1	I	Input signal from IOPAD
FABRIC_DLY_LOAD	1	I	Delay load input
FABRIC_DLY_ADJ	1	I	Delay adjust input
FABRIC_DLY_INCDEC	1	I	Delay increment / decrement input
FABRIC_DLY_TAP_VALUE	6	O	Delay tap value output
FABRIC_O	1	O	Output Signal for FABRIC
I_DDR			
IOPAD_CLK	1	I	Input clock from IOPAD
IOPAD_PLL_REF_CLK	1	I	Input PLL reference clock from IOPAD
IOPAD_D	1	I	Input data from IOPAD
FABRIC_R	1	I	Active-low asynchrnous reset
FABRIC_E	1	I	Active-high enable
FABRIC_Q	2	O	Output data to FABRIC
I_SERDES			
IOPAD_D	1	I	Input data from IOPAD
IOPAD_CLK	1	I	Input clock from IOPAD
IOPAD_PLL_REF_CLK	1	I	PLL reference clock from IOPAD
FABRIC_RX_RST	1	I	Active-low asycnhronous reset
FABRIC_BITSLIP_ADJ	1	I	Synchronizes incoming data stream
FABRIC_EN	1	I	Input enable
FABRIC_CLK_IN	1	I	Input clock from FABRIC
FABRIC_CLK_OUT	1	O	Output clock
FABRIC_Q	<WIDTH>	O	Output data
FABRIC_DATA_VALID	1	O	Output valid signal
FABRIC_DPA_LOCK	1	O	Delay Phase Alignment lock output
FABRIC_DPA_ERROR	1	O	Delay Phase Alignment error output
FABRIC_DLY_LOAD	1	I	Delay load input

Signal Name	Width	I/O	Description
FABRIC_DLY_ADJ	1	I	Delay adjust input
FABRIC_DLY_INCDEC	1	I	Delay increment / decrement input
FABRIC_DLY_TAP_VALUE	6	O	Delay tap value output
O_BUF			
FABRIC_I	1	I	Input signal from FABRIC
FABRIC_T	1	I	Tri-state input from FABRIC
IOPAD_O	1	O	Output signal to IOPAD
IOPAD_O_P	1	O	Negative end differential output
IOPAD_O_N	1	O	Negative end differential output
O_DELAY			
IOPAD_CLK	1	I	Input clock from IOPAD
IOPAD_PLL_REF_CLK	1	I	PLL reference clock from IOPAD
FABRIC_I	1	I	Input signal from FABRIC
FABRIC_DLY_LOAD	1	I	Delay load input
FABRIC_DLY_ADJ	1	I	Delay adjust input
FABRIC_DLY_INCDEC	1	I	Delay increment / decrement input
FABRIC_DLY_TAP_VALUE	6	O	Delay tap value output
IOPAD_O	1	O	Output Signal for IOPAD
O_DDR			
IOPAD_CLK	1	I	Input clock from IOPAD
IOPAD_PLL_REF_CLK	1	I	PLL reference clock from IOPAD
FABRIC_D	2	I	Input data from FABRIC
FABRIC_R	1	I	Active-low asynchrnous reset
FABRIC_E	1	I	Active-high enable
IOPAD_Q	1	I	Output data
O_SERDES			
IOPAD_CLK	1	I	Input clock from IOPAD
IOPAD_PLL_REF_CLK	1	I	PLL reference clock from IOPAD
FABRIC_D	<WIDTH>	I	Input data from FABRIC
FABRIC_RST	1	I	Active-low asycnhronous reset
FABRIC_LOAD_WORD	1	I	Load word input
FABRIC_CLK_IN	1	I	Input clock from FABRIC
FABRIC_OE	1	I	Output enable signal
IOPAD_CLK_OUT	1	O	Output clock
IOPAD_Q	1	O	Output data
FABRIC_DLY_LOAD	1	I	Delay load input
FABRIC_DLY_ADJ	1	I	Delay adjust input
FABRIC_DLY_INCDEC	1	I	Delay increment / decrement input
FABRIC_DLY_TAP_VALUE	6	O	Delay tap value output

Table 2: Port List

Parameters

Table 3 lists the parameters of the IO Configurator.

Parameter	Values	Default	Description
IO_MODEL	CLK_BUF, I_BUF, I_DELAY, I_DDR, I_SERDES, O_BUF, O_DELAY, O_DDR, O_SERDES	CLK_BUF	IO Primitive
IO_TYPE	SINGLE_ENDED, DIFFERENTIAL, TRI_STATE, DIFF_TRI_STATE	SINGLE_ENDED	Type of IO
IO_MODE	NONE, PULLUP, PULLDOWN	NONE	Pullup/Pulldown resistor enabing
DELAY	0 - 63	0	Tap delay value
DELAY_ADJUST	TRUE, FALSE	TRUE	Delay adjustment for input/output
DELAY_TYPE	STATIC, DYNAMIC	STATIC	Delay Type for input/output
DATA_RATE	SDR	SDR	Data rate for SERDES
OP_MODE	NONE, DPA, CDR	NONE	Operation mode for SERDES
CLOCKING	RX_CLOCK, PLL	RX_CLOCK	Clock source for IO Model
CLOCKING_SOURCE	LO-CAL_OSCILLATOR, RX_IO_CLOCK	LO-CAL_OSCILLATOR	Clock source for PLL
WIDTH	3 - 10	3	Width of Serialization/Deserialization
REF_CLK_FREQ	5 - 1200	50	Reference clock frequency in MHz
OUT_CLK_FREQ	800 - 3200	1600	Output clock frequency in MHz
CLOCK_FORWARDING	TRUE, FALSE	FALSE	Clock forwarding option for O_SERDES
CLOCK_PHASE	0, 90, 180, 270	0	Clock phase for O_SERDES

Table 3: Parameters

Design Flow

IP Customization and Generation

IO Configurator IP core is a part of the Raptor Design Suite Software. A customized memory can be generated from the Raptor's IP configuration window as shown in figure 11.

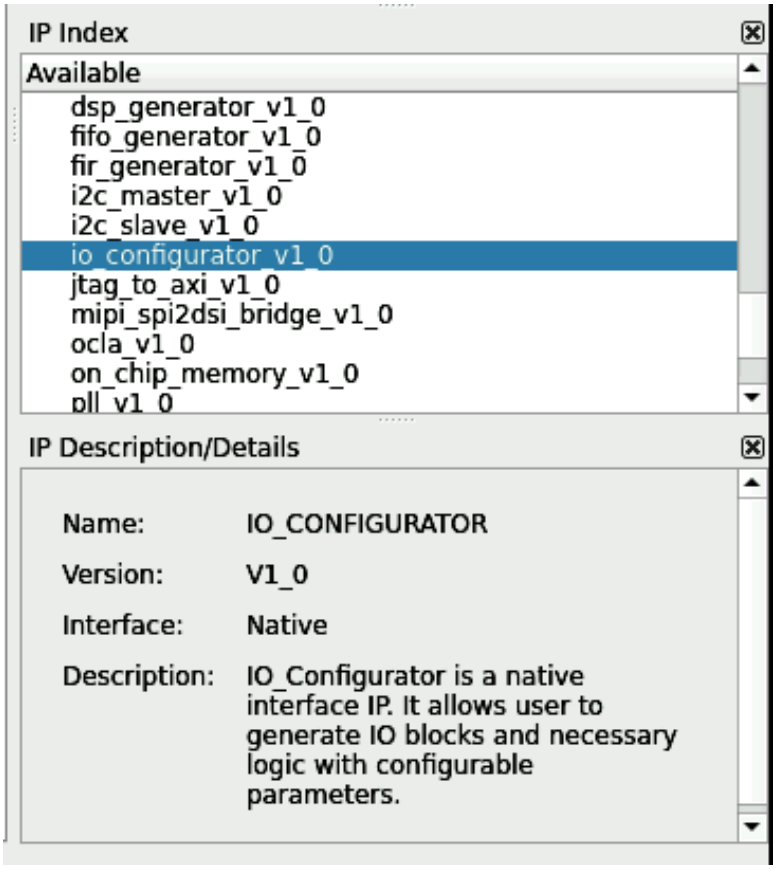

Figure 11: IP List

Parameters Customization

From the IP configuration window, the parameters of the IO Configurator can be configured and it's features can be enabled for generating a customized IP core that suits the user application requirements. All parameters are shown in Figure 12. In Figure 12, the module name specifies the name of both the Verilog file and the top-level IP name that will be generated based on configured parameters.

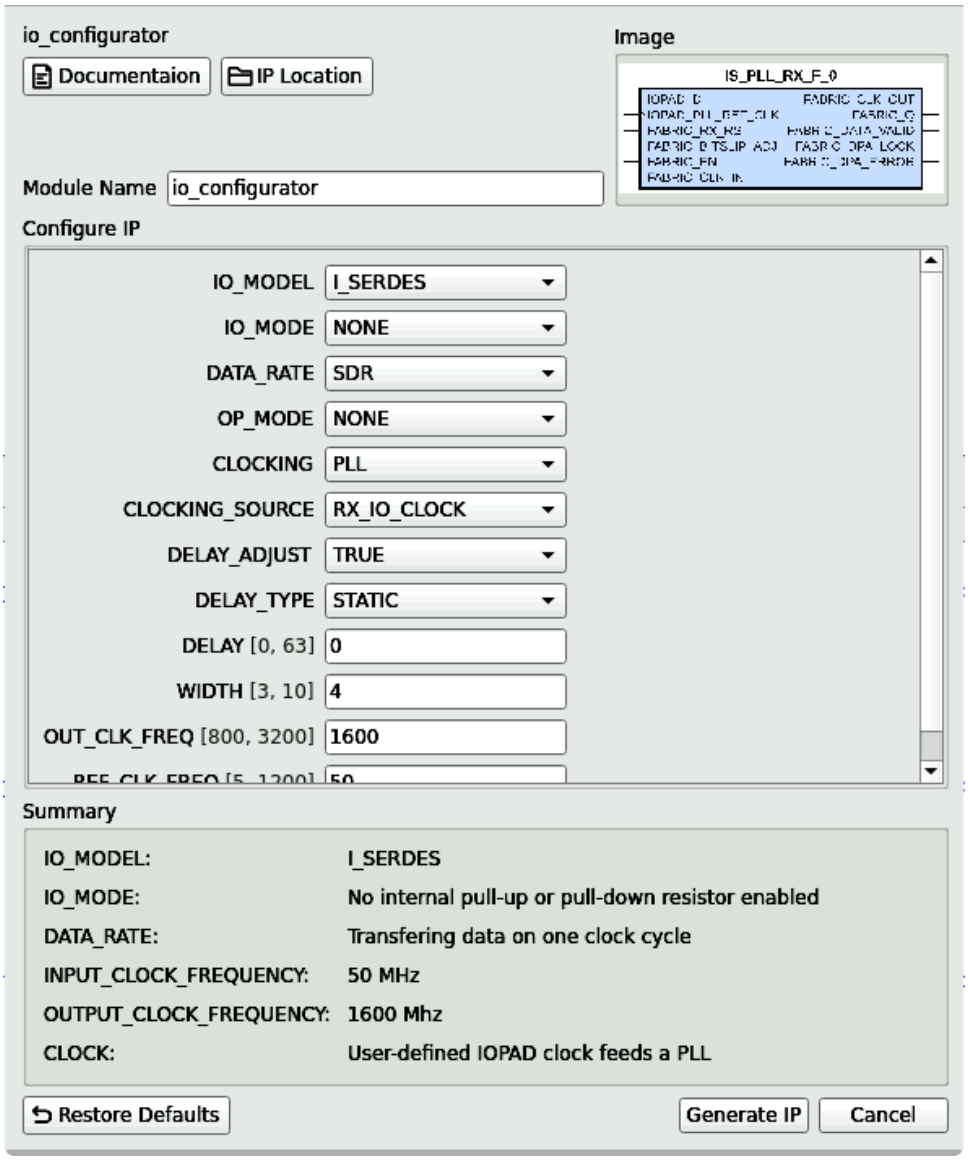

Figure 12: IP Configuration

Testbench

Test

The testbench provided with IO Configurator is for O_SERDES. In this test, random data is generated and fed to O_SERDES design. 100 MHz clock is provided to design for simulation. Output results of this test are shown in Figure 13.

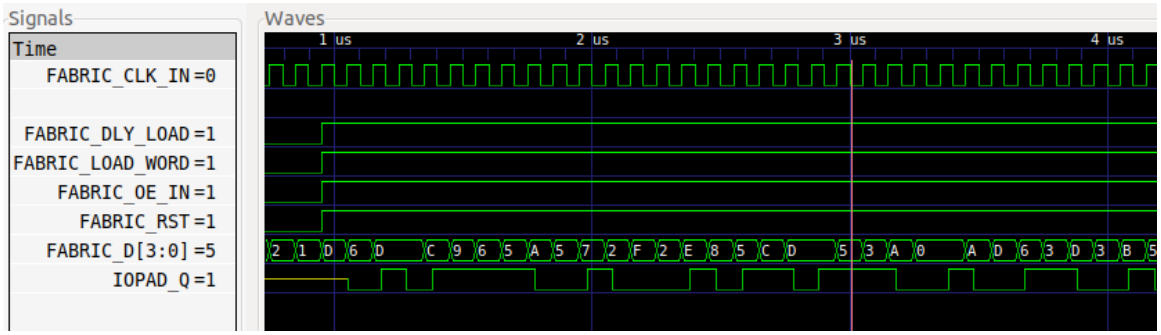

Figure 13: Simulation Results

Simulation

To run simulation, go to Simulate IP option as shown in Figure 14.

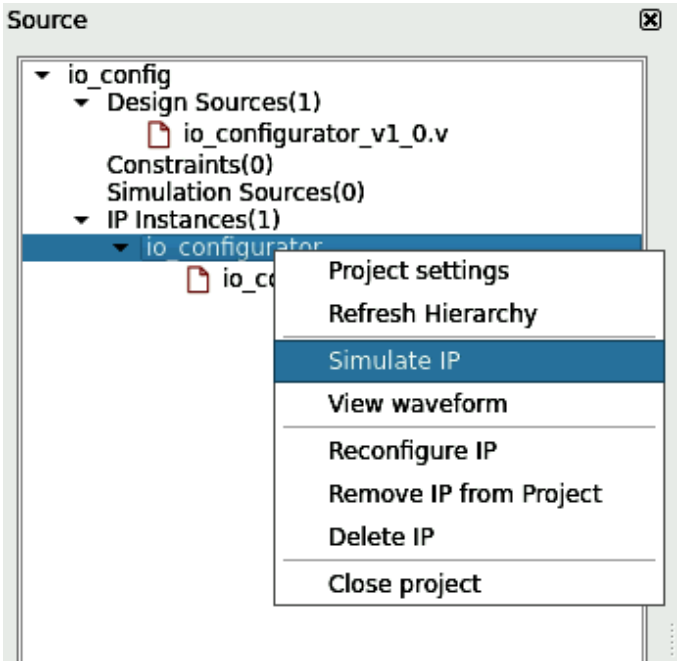

Figure 14: Simulate IP

Waveform

To view waveform, go to View waveform option as shown in Figure 15.

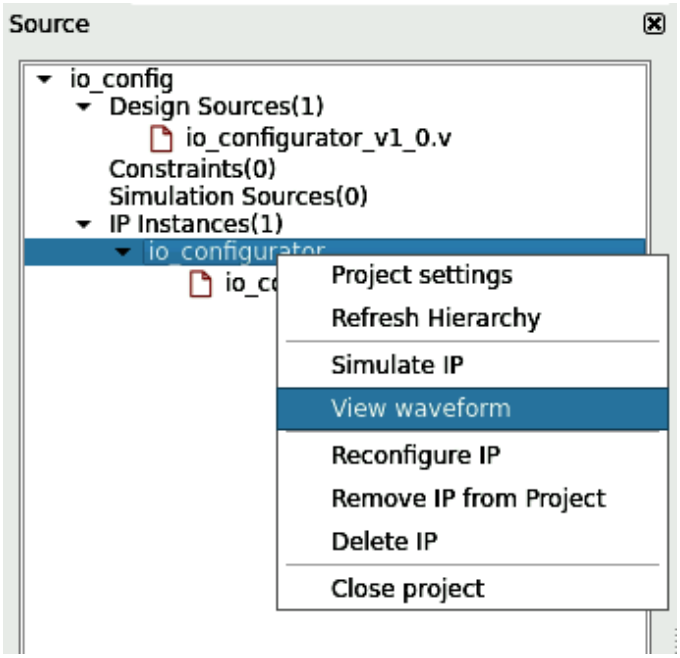

Figure 15: View Waveform

Revision History

Date	Version	Revisions
May 9, 2024	0.1	Initial version IO Configurator User Guide