RapidSilicon

IO CONFIGURATOR
(Beta Release)

Version 0.1

May 21, 2024

Contents

IP Summary

Introduction

The IO tile in the FPGA supports the Buffers, Delays, SerDes and DDR primitives which user can instantiate in the design. The IO Configurator allows the user to generate a wrapper with all the necessary IO components connected together to facilitate the ease of design. Using this IP, user can quickly generate the code required to use an IO Primitive in their design. The generated wrapper would include the required IO Buffer and clock generation.

Features

- Support multiple IO models including Buffers, Delays, SerDes and DDR.
- Support four types of Buffers. i.e. Single_Ended, Differential, Tri-state and Differential-Tri-state.
- Support Pull-up and Pull-down resistor to make logic low/high in the absence of an external connection.
- 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 in SerDes.
- Support multiple clock sourcing option for SerDes, Delays and DDR.
- 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 for Delays.

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.

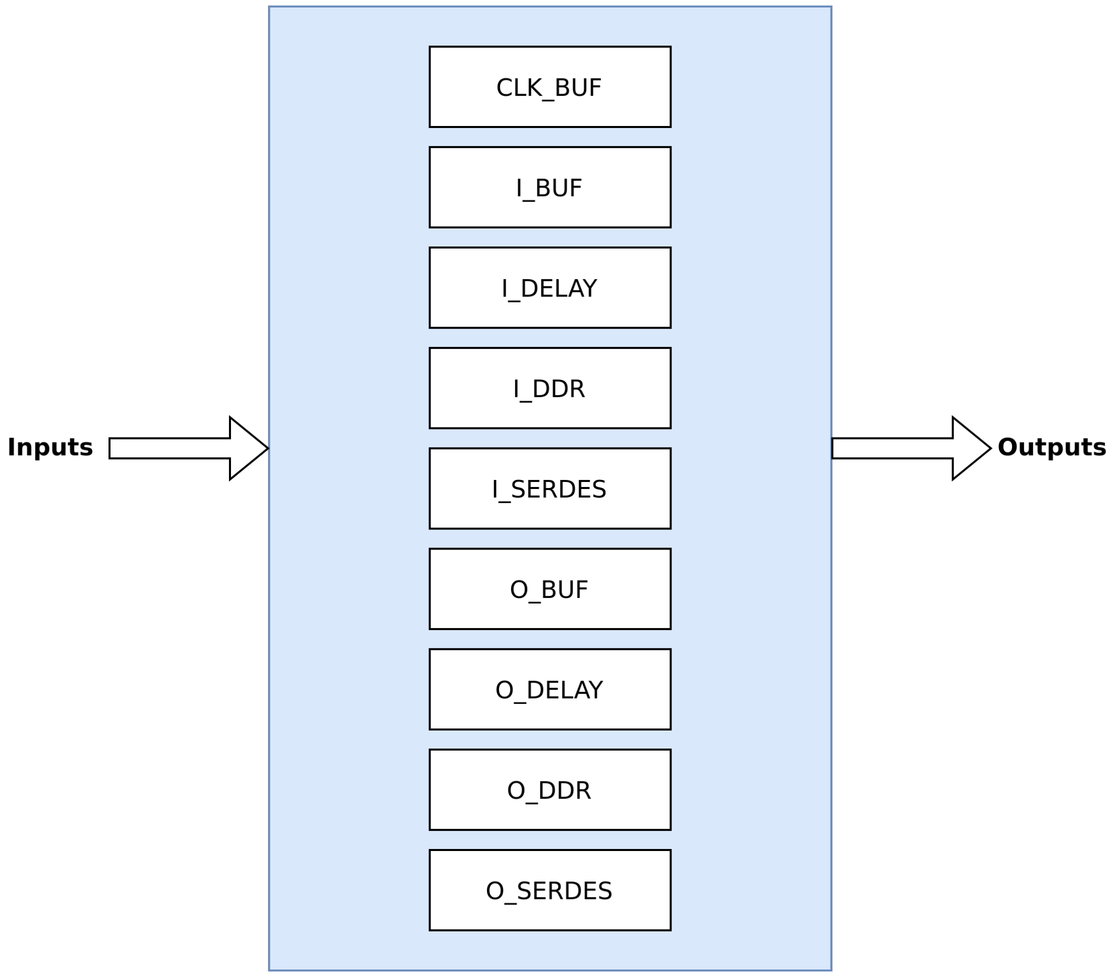

Figure 1: IO Configurator Block Diagram

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 an IO component designed for managing clock signals within an FPGA. It receives a clock signal from an input buffer, and provides a buffered version of that signal for internal use. It may also support Pull-up or Pull-down resistor. These resistors can eliminate the need for external resistors on the circuit board, simplifying the design and reducing component count.

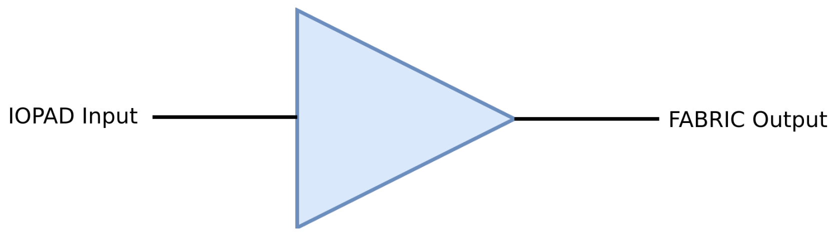

Figure 2: CLK_BUF

- **I_BUF:**
  Input Buffer is an IO component which allows the signals to be received from output of the FPGA. The input buffer may support various voltage standards. These voltage standards must be set using a physical constraint file provided by the user. User may configure I_BUF as a Single Ended or Differential. I_BUF may also support Pull-up or Pull-down resistor. These resistors may replace an external resistor on the board.

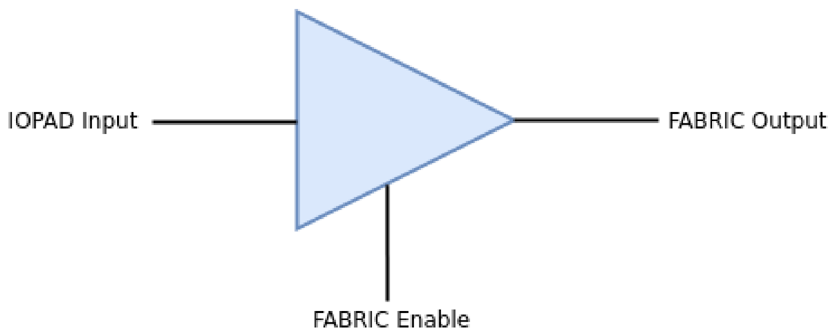

Figure 3: I_BUF

- **I_DELAY:**
  Input Delay is a digital circuit used to introduce a controlled time delay to an incoming input data signal. It can be used to adjust the arrival time of the input signal relative to the clock edge. It supports Static and Dynamic Delay. With Static Delay, user can set a constant delay value and load it while with Dynamic Delay, user can control delay through the Fabric and load it. By providing Tap Delay Value, user may add delay to the input signal. One tap delay value is equal to 51.56 ps. User may provide clock source for I_DELAY.

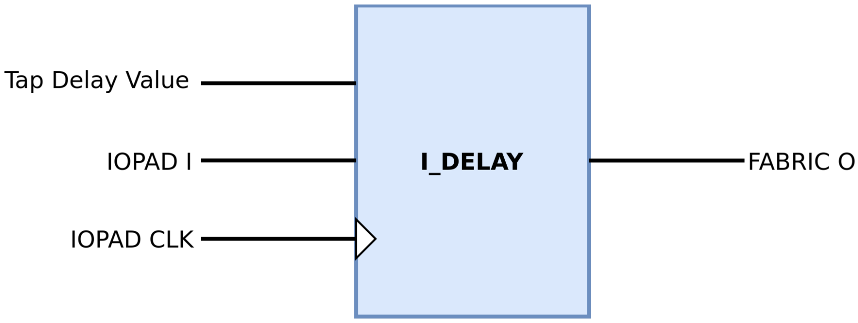

Figure 4: I_DELAY

- **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. User may configure I_DDR by providing clock source and Pull-up and Pull-down resistor to replace on-board resistors.

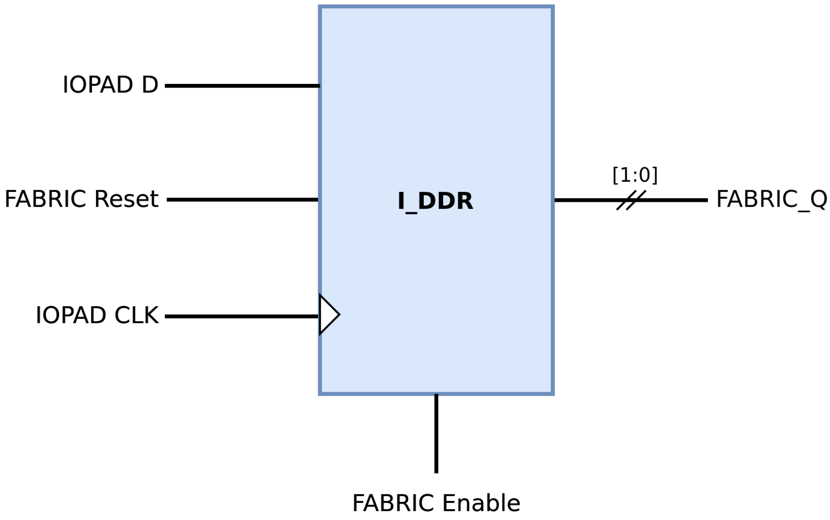

Figure 5: I_DDR

- **I_SERDES:**
  I_SERDES is a digital circuit used for converting an incoming serial data stream into parallel data for FPGA. It converts the data on single clock cycle (SDR). Users have flexibility in choosing the clock source for I_SERDES. It can be provided by an internal Phase-Locked Loop (PLL) for precise timing control, or directly from an external input pin (IOPAD) for interfacing with existing clock signals. User may configure I_SERDES by providing delay adjustment and data width for serialization.

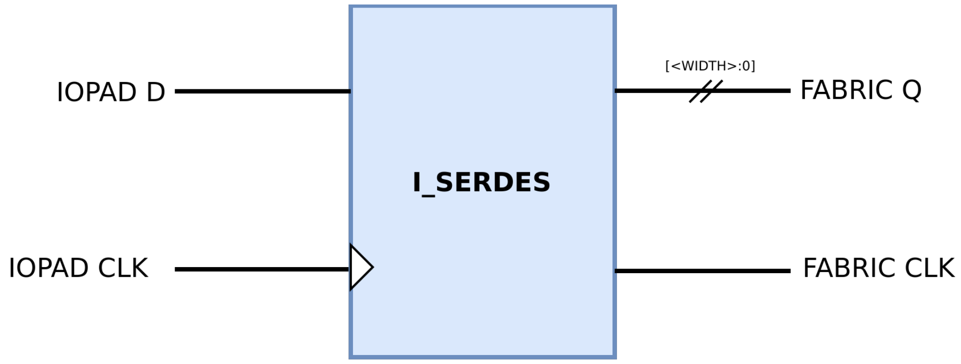

Figure 6: I_SERDES

- **O_BUF:**
  Output Buffer is an IO component designed to drive signals from the internal logic of an FPGA to external devices. The output buffer may support various voltage standards. These voltage standards must be set using a physical constraint file provided by the user. User may configure four types of output buffers.i.e Single Ended, Differential, Tri-State or Differential Tri-State Buffer. O_BUF may also support Pull-up or Pull-down resistor. These resistors may replace an external resistor on the board.

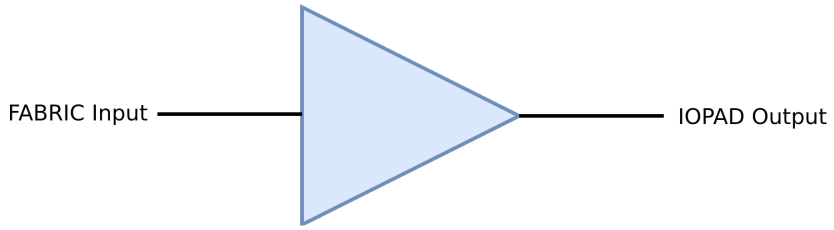

Figure 7: O_BUF

- **O_DELAY:**
  Output Delay is a digital circuit used to introduce a controlled time delay to an outgoing signal from the FPGA. It can be used to adjust the forwarding time of the output signal relative to the clock edge. It supports Static and Dynamic Delay. With Static Delay, user can set a constant delay value and load it while with Dynamic Delay, user can control delay through the Fabric and load it. By providing Tap Delay Value, user may add delay to the output signal. One tap delay value is equal to 51.56 ps. User may provide clock from PLL or IOPAD to O_DELAY.

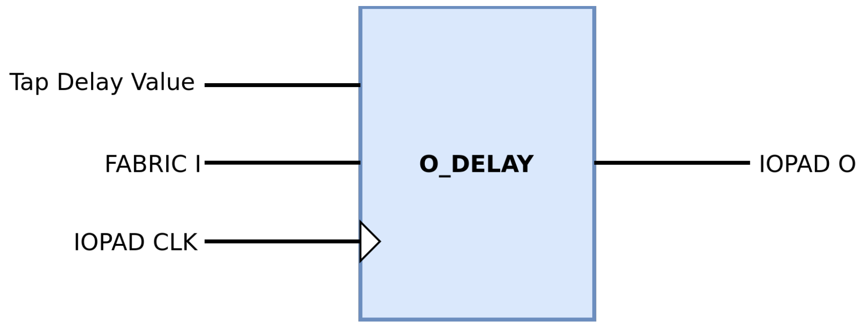

Figure 8: O_DELAY

- **O_DDR:**
  Output Double Data Rate is a type of output primitive to facilitate high-speed data transfers from FPGA to the external world. The key advantage of an O_DDR interface is its ability to transfer data on both the rising and falling edges of the clock signal. User may provide clock from PLL or IOPAD. There is also an option to add Pull-up or Pull-down resistor to replace on-board resistors.

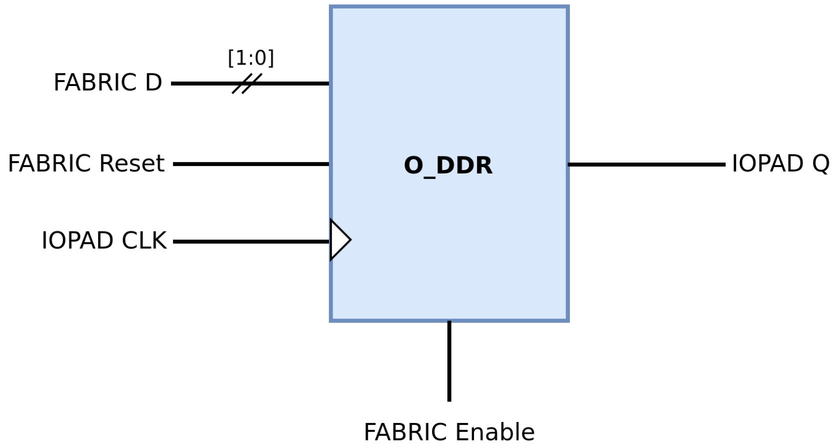

Figure 9: O_DDR

- **O_SERDES:**
  O_SERDES is a digital circuit used for converting parallel data streams generated within FPGA into a serial data stream. It converts the data on single clock cycle (SDR). Users have flexibility in choosing the clock source for O_SERDES. It can be provided by an internal Phase-Locked Loop (PLL) for precise timing control, or directly from an external input pin (IOPAD) for interfacing with existing clock signals. O_SERDES offers additional configuration options like Delay adjustment, clock forwarding and data width for deserialization.

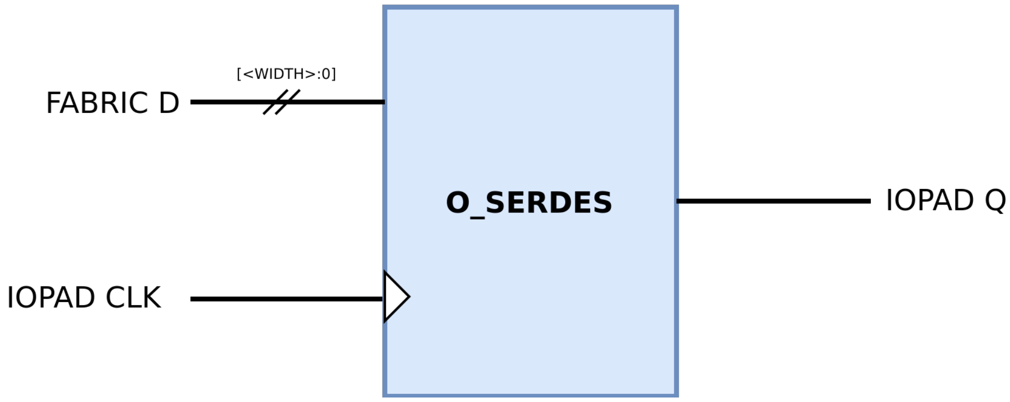

Figure 10: O_SERDES

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	-	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.

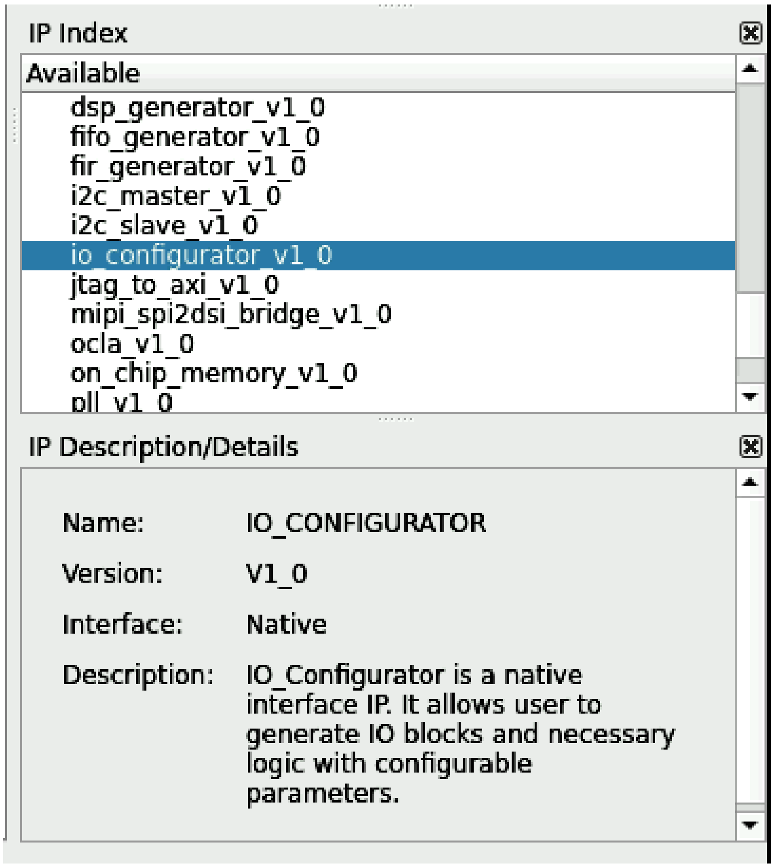

Figure 11: IP List

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.

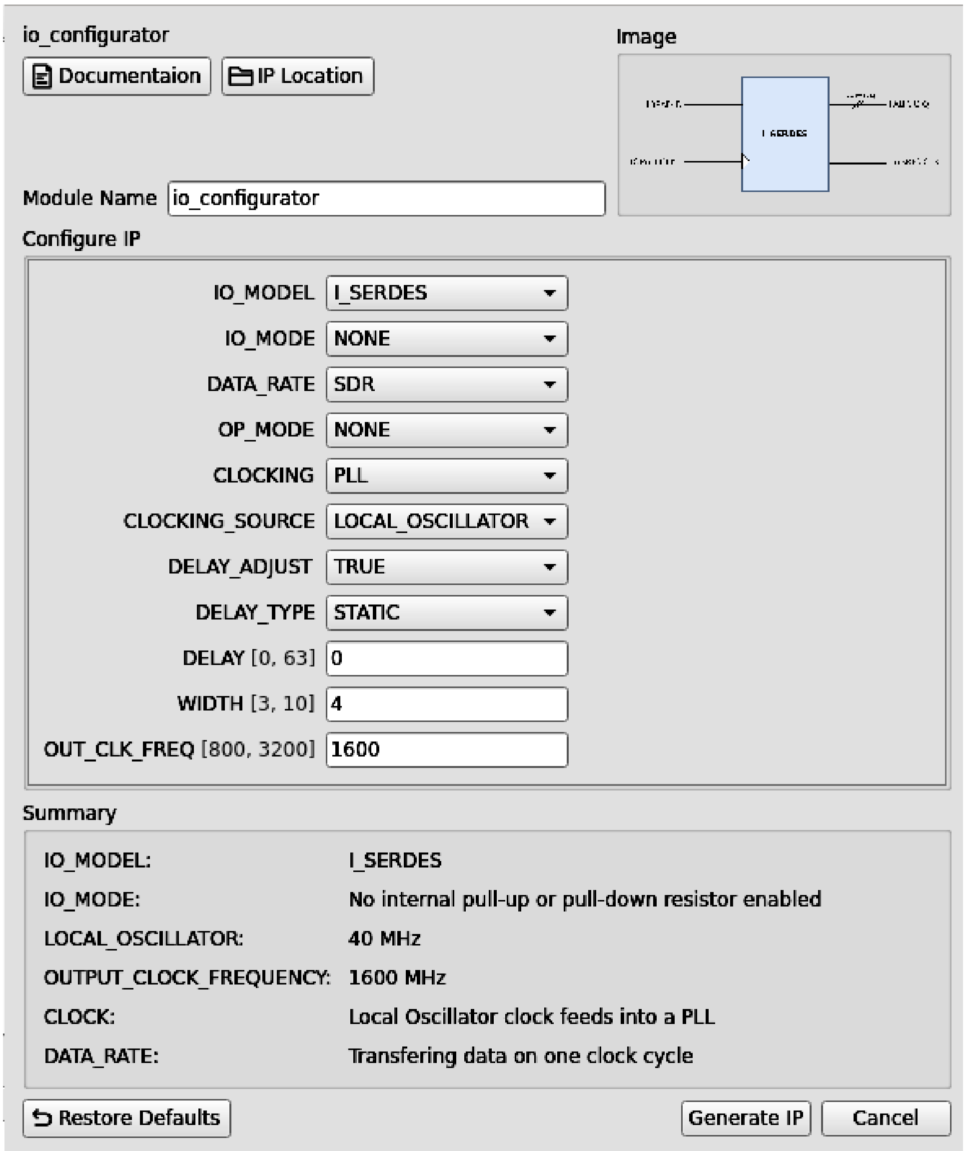

Figure 12: IP Configuration

Constraint File

Raptor Design Suite supports two types of constraint files to customize design. To create constraint file: Project > Project Settings > Design Constraints > Create File.

- **Physical Constraints (*.pin):** This file defines how design signals are mapped to specific physical pins on the FPGA device. To create (*.pin) file, select pin as a File type and specify File name of the physical constraint file as shown in Figure 13.

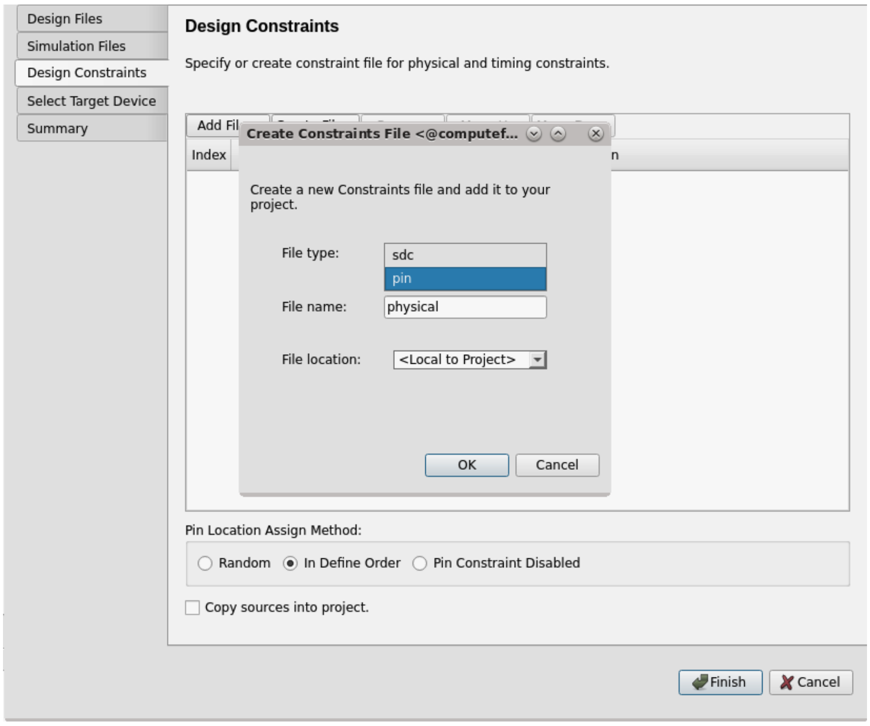

Figure 13: Physical Constraint File

To configure (*.pin) file, go to: Tools > Pin Planner. New IO Ports tap will be displayed as shown in Figure 14.

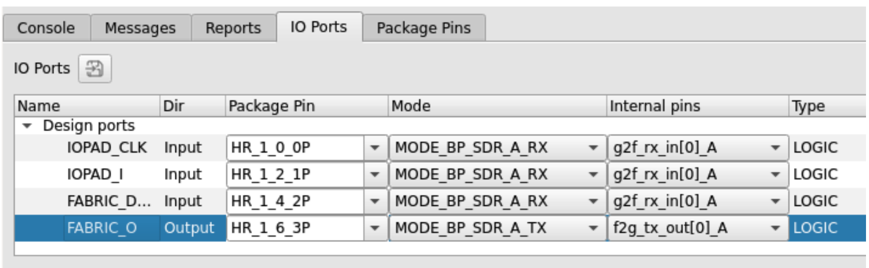

Figure 14: IO Ports

- **Timing Constraints (*.sdc):** This file specifies critical timing requirements for user design. To create (*.sdc) file, select sdc as a File type and specify File name of the timing constraint file as shown in Figure 15.

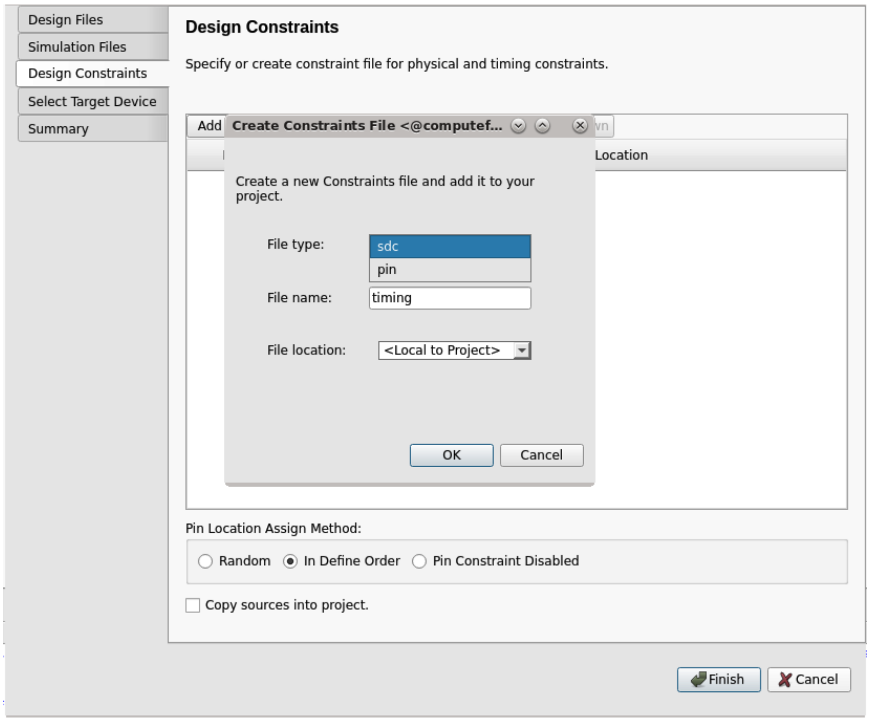

Figure 15: Timing Constraint File

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 16.

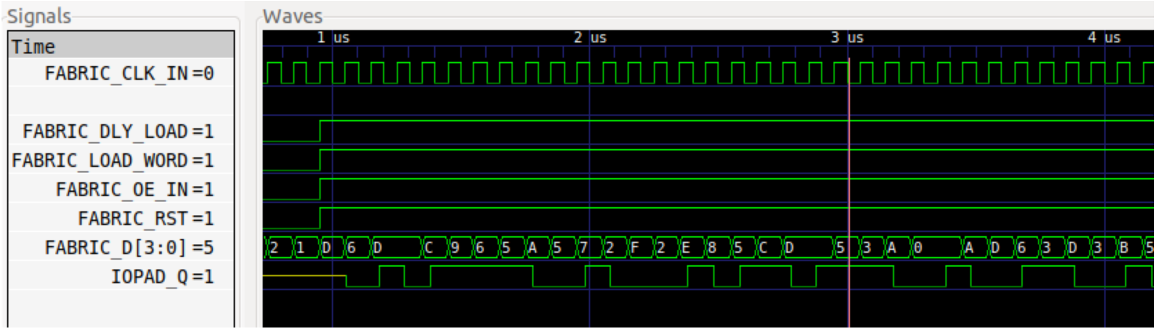

Figure 16: Simulation Results

Simulation

To run simulation, go to Simulate IP option as shown in Figure 17.

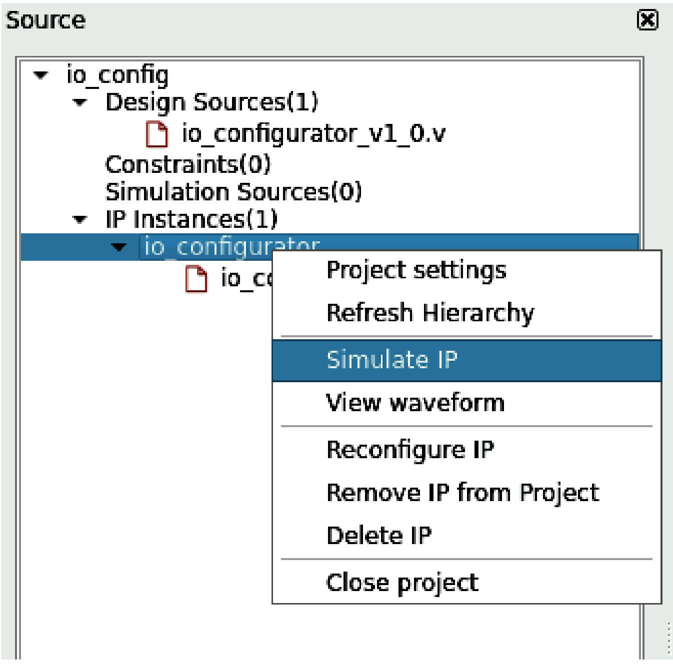

Figure 17: Simulate IP

Waveform

To view waveform, go to View waveform option as shown in Figure 18.

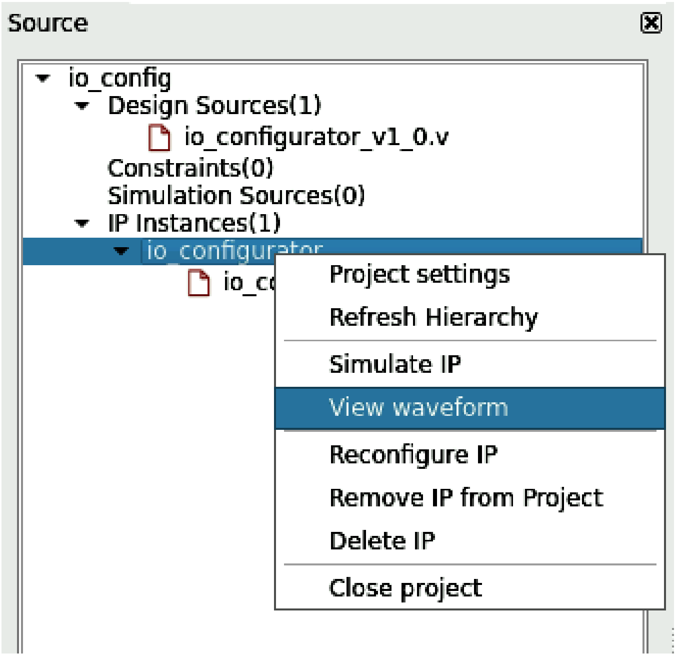

Figure 18: View Waveform

Revision History

Date	Version	Revisions
May 21, 2024	0.1	Initial version IO Configurator User Guide