RapidSilicon

FIR Generator
(Beta Release)

Version 0.1

February 29, 2024

Copyright

Trademarks

All Rapid Silicon trademarks are as listed at www.rapidsilicon.com. Synopsys and Synplify Pro are trademarks of Synopsys, Inc. Aldec and Active-HDL are trademarks of Aldec, Inc. Modelsim and Questa are trademarks or registered trademarks of Siemens Industry Software Inc. or its subsidiaries in the United States or other countries. All other trademarks are the property of their respective owners.

Disclaimers

Contents

IP Summary

Introduction

The FIR Generator IP stands as a versatile tool for crafting Finite Impulse Response (FIR) Filters, offering adaptability in the number of filter taps, coefficients and optimizations. With a customizable input data width, it effortlessly accommodates a wide spectrum of signal widths. This generator meticulously offers optimizations for either performance, enabling parallel computations, or area, enabling serial computations with less hardware, ensuring efficient signal processing across diverse applications. Its flexible design empowers users to tailor FIR filters precisely to their specifications, fostering a high level of customization for advanced signal processing tasks.

Features

- Configurable Input Data Width from 1 - 18.

- Support for up to 120 Coefficients and filter taps.

- Supports loading coefficients from either a source file in .txt or .hex format, or enter individually from the IP Configurator Window.

- Both signed and unsigned operation supported.

- Customizable fractional bits for both coefficients and input data.

- Customizable output data width.

- Support for both Area and Performance optimization.

- Up to 20 bit wide coefficients supported.

- Support for Fixed and Reloadable Coefficients in the filter.

Overview

FIR Generator

The FIR Generator IP (Finite Impulse Response Generator Intellectual Property) is a versatile and high-performance hardware module designed for implementing Finite Impulse Response (FIR) filters in digital signal processing applications. The FIR Generator IP maximizes performance by exclusively leveraging the onboard DSP38 slice for all Multiply-Accumulate (MAC) operations. There are two different types of implementations for the FIR Generator which are detailed below: -

- **Performance Optimization:** This implementation optimizes FIR filtering with parallel processing, utilizing dedicated DSP38 slices for each coefficient that makes up a stage of the FIR Filter. The interconnected slices create a cascaded pipeline, where the output of one slice feeds into the next, maximizing computational efficiency. The design excels in concurrent execution, processing multiple coefficients in parallel at each clock cycle for heightened throughput and real-time signal processing capabilities.

- **Area Optimization:** This implementation optimizes resource usage by employing a single DSP38 slice for any number of coefficients. In this mode, a higher frequency clock is employed to drive the DSP38 slice. The DSP38 slice operates serially on this faster clock, while the internal registers function on the master clock, ensuring that an output is generated on the rising edge of the master clock. While this approach conserves DSP38 slices, it relies on LUTs and registers, impacting FIR Filter performance.

A block diagram of a top level of FIR Filter generated from the FIR Generator IP is shown in Figure 1.
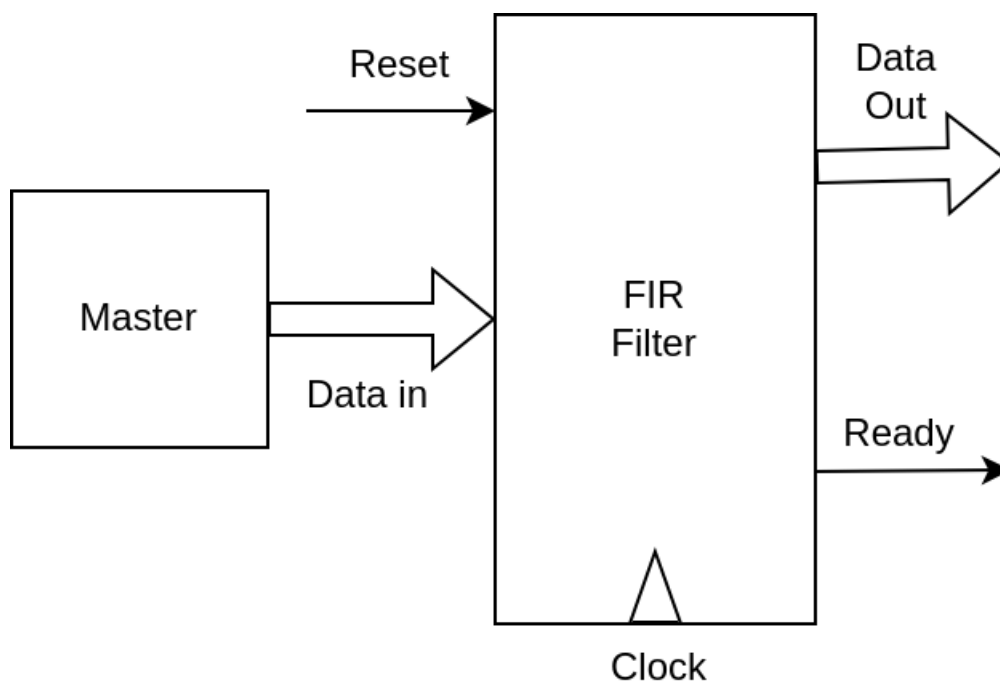

Figure 1: FIR Block Diagram

IP Specification

The classical Finite Impulse Response (FIR) filter performs a convolution sum, combining the filter coefficients and input data within a window defined by the number of filter taps. The filtered output is computed using the formula specified in the equation 1.

$$y[k] = \sum_{n=0}^{N-1} h(n) \cdot x(k-n) \quad k = 0, 1, 2, \ldots \tag{1}$$

The conceptual depiction of the aforementioned process is vividly illustrated in figure 2. In this visual representation, the Z^{-1} elements symbolize the tap delays through which the incoming data traverses. The data then proceeds through a series of multipliers and adders, culminating in the generation of the desired filtered output at the final stage.
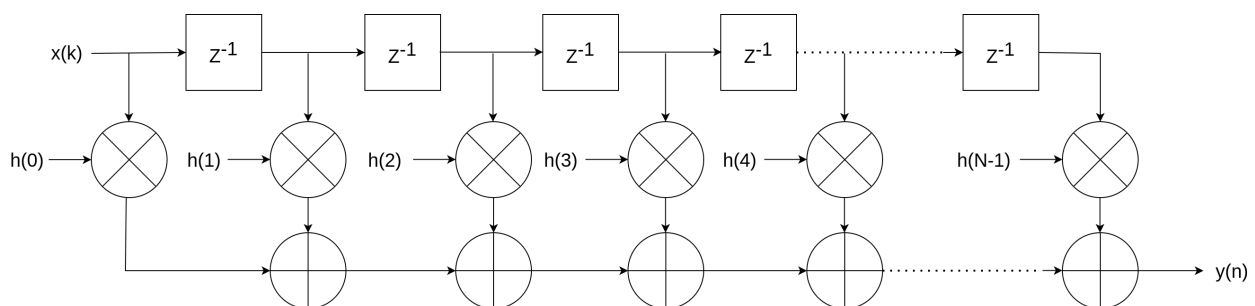

Figure 2: Conventional Tapped Delay FIR Filter Representation

FIR Generator gives two different implementations for this FIR Filter so it suits the requirements of the user, with the inclusion of a **ready** port indicating when the generated FIR Filter is ready to sample the incoming data. Details of both implementations are given below: -

Performance Optimization

In this thoughtfully designed implementation, individual DSP38 slices are exclusively dedicated to each filter tap. This strategic allocation allows for simultaneous computations, enabling the generation of output at peak throughput. The system operates on fixed-point arithmetic, wherein coefficients are provided in either decimal or hexadecimal notation. The FIR Generator seamlessly translates these coefficients into their fixed-form representation, tailoring them to the specific number of fractional bits designated to the coefficients from the IP Configurator.

When loading coefficients from a file, numerical values should be separated by commas or whitespaces. While decimal numbers require no additional notation, hexadecimal numbers necessitate the **0x** prefix. Notably, this implementation imposes no additional logic, ensuring the generated filter operates at the maximum supported frequency, accommodating any configuration of filter taps and sizes. The output port *ready* becomes active immediately upon the removal of the reset signal. This indicates that the filter is ready to process incoming data, generating filtered output on each rising edge of the clock cycle following the specified delay from the filter taps.

It's important to highlight that in this mode, the generated FIR Filter is a fixed-coefficients

filter. This implies that once the IP Generation is completed, the coefficients remain static and cannot be altered thereafter. Any modifications to the coefficients necessitate a repetition of the IP Generation process to incorporate the changes.

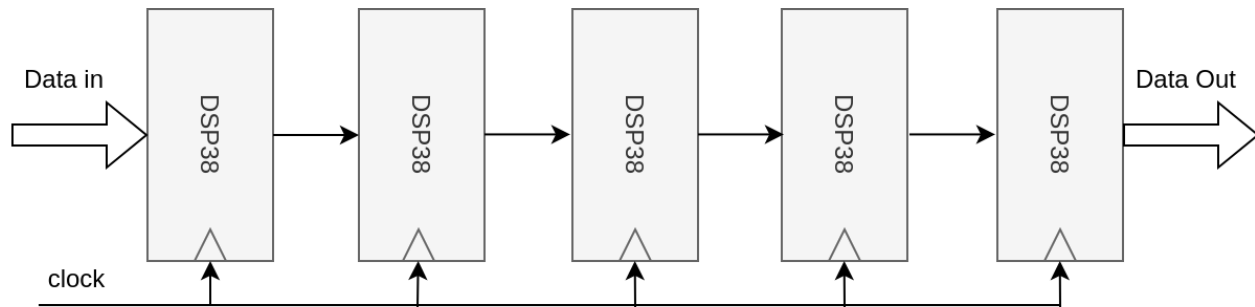For a visual representation of this implementation, refer to figure 3.

Figure 3: Performance Optimized

Area Optimization

This implementation capitalizes on the versatility of the DSP38 slice, leveraging its capability to perform multiply and accumulate operations in a pipelined manner within the same slice. This innovative approach significantly diminishes the need for multiple DSP38 slices, regardless of the specified number of filter taps and stages, a solo DSP38 slice suffices to compute the FIR response, leading to a more resource-efficient design.

To enhance efficiency, a higher clock frequency is introduced, significantly boosting the performance of the DSP38 slice whereas the internal registers operate on the master clock. This strategic design choice ensures that the filter produces an output on each rising edge of the master clock, eliminating potential wait times in the system. The frequency of this faster clock must be equal to the number of filter taps.

In this architecture, incoming data is systematically shifted through a series of registers, guaranteeing that the DSP38 slice consistently operates on the most up-to-date input, aligned with the corresponding coefficient. This inclusion of shift registers makes for some logic being used in the fabric hindering the performance of the generated FIR Filter. What this means is that the range of frequency on which the filter works changes dynamically with the number of filter taps. The range of this input frequency can be easily computed via the equation 2. Note that the ranges come out in MHz.

$$\text{Minimum Frequency} = \left(\max \left(\frac{800}{\text{Number of Filter Taps}}, 8 \right), 2 \right)$$
$$\text{Maximum Frequency} = \left(\min \left(\frac{3200}{\text{Number of Filter Taps}}, 500 \right), 2 \right)$$

(2)

When manually inputting coefficients, the FIR Generator seamlessly employs fixed-point arithmetic. It automatically computes the fixed-point representation for both decimal and hexadecimal numbers. Decimal numbers require no additional notation, while hexadecimal numbers necessitate the **0x** prefix. Users can input coefficients either in decimal or hexadecimal format, separated by commas or whitespaces.

For coefficients loaded from a file, the FIR Generator exclusively supports the .hex file format. In this format, hexadecimal numbers are presented without any prefix and are separated by whitespaces. This streamlined approach ensures compatibility and ease of

use when providing coefficients through manual input or file loading. The activation of the output port *ready* may encounter a delay contingent upon the synchronization of the clocks employed in this design. Once this process is complete, the FIR Filter is prepared to sample the incoming data and generate output on each rising edge of the master clock.

In this mode, it's crucial to emphasize that when manually inputting coefficients, the resulting FIR Filter becomes fixed-coefficients. This signifies that post IP Generation, any adjustments to the coefficients require a complete repetition of the IP Generation process. Conversely, when employing a .hex file to load coefficients, the generated FIR Filter adopts a re-loadable characteristic. This means that coefficients can be modified within the file after IP Generation. However, it's essential to maintain the same total number of coefficients for proper functionality. For a visual representation of this implementation, refer to figure 4.
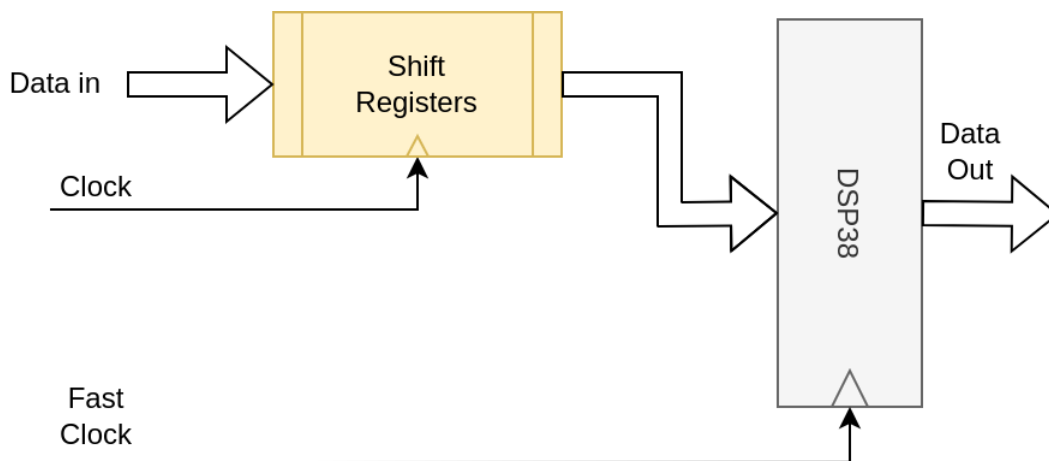

Figure 4: Area Optimized

Output Width and Bit Growth

By default, the FIR Generator IP employs the full precision output data width, defined as the sum of the input data width and the bit growth of the FIR Filter. Bit growth signifies the increase in the output value resulting from numerous multiplications and accumulations occurring within the FIR Filter. This IP incorporates two distinct bit growth mechanisms, each serving a specific purpose, as elaborated below: -

- **True Maximum Bit Growth:** This particular bit growth mechanism is harnessed for Performance mode in scenarios where coefficients are manually inputted and also when loaded from a file. Additionally, it is applied in Area mode when coefficients are manually provided. The calculation of this bit growth involves taking the base 2 logarithm of the sum of the absolute values of all filter coefficients. This approach takes into account the actual value of all the coefficients. This calculated bit growth serves a crucial role in optimizing the accumulator width, thereby conserving valuable resources within the system. This calculation is given by the equation 3.

$$\text{Bit Growth} = \left\lceil \log_2 \left(\sum_{n=0}^{N-1} |a_n| \right) \right\rceil \tag{3}$$

- **Worst Case Bit Growth:** The upper limit of bit growth is determined by adding the coefficient width to the ceiling of the base 2 logarithm of the count of non-zero multiplications required. It's crucial to note that this computation does not account for the specific values of the coefficients. This particular bit growth computation is applied specifically in Area Optimization mode when loading coefficients from a .hex file. This strategy is employed because, in this mode, the actual values of the filter taps remain unknown during compile time. Additionally, coefficients are re-loadable in this mode, and this technique allows the filter to adapt to the associated bit growth. The formula to calculate the worst case bit growth is given by equation 4.

$$\text{Bit Growth = coefficient width} + \lceil \log_2(\text{number of non-zero multiplications})\rceil \quad (4)$$

- **Truncation:** The FIR Generator IP additionally offers flexibility with custom output data width, achieved by selectively truncating the least significant bits (LSBs) of the previously computed full precision data width. The fractional bits in the now truncated output is calculated via the formula given in the equation 5, where

    - OFB: Output Fractional Bits
    - IFB: Input Fractional Bits
    - CFB: Coefficient Fractional Bits
    - FPW: Full Precision Output Width
    - ODW: Output Data Width

$$\text{OFB = IFB + CFB} - \max\left(0, (\text{FPW}) - \text{ODW}\right) \quad (5)$$

Standards

The FIR Generator soft IP supports the native interface with the standard *data_in*, *ready* and *data_out* ports along with the supporting clock and reset signals in the port list.

IP Support Details

The Table 1 gives the support details for FIR Generator.

Compliance		IP Resources					Tool Flow		
Device	Interface	Source Files	Constraint File	Testbench	Simulation Model	Software Driver	Analyze and Elaboration	Simulation	Synthesis
GEMINI	Native	Verilog	-	Verilog	VVP	Iverilog	Raptor (Surelog)	Raptor (Icarus)	Raptor

Table 1: IP Details

Port List

Table 2 lists the interface ports of the FIR Generator.

Signal Name	I / O	Description
data_in {input_width}	I	Data Input
data_out {output_width}	O	Data Output
rst	I	Reset
clk	I	Master Clock
ready	O	Signal indicating readiness of FIR
fast_clk	I	Accelerated Clock for Area Optimized Mode only

Table 2: FIR Generator Interface

Resource Utilization

The parameters for computing the maximum and minimum resource utilization are given in Table 3, remaining parameters have been kept at their default values.

FPGA Device				
Configuration			**Resource Utilization**	
	Options	**Configuration**	**Resources**	**Utilized**
Minimum Resource Performance Optimized	Input Width	18	DSP	1
	Number of Coefficients	1		
	Coefficients File	FALSE		
	Signed	FALSE		
	Truncated Output	FALSE		
	Options	**Configuration**	**Resources**	**Utilized**
Maximum Resource Performance Optimized	Input Width	18	DSP	120
	Number of Coefficients	120		
	Coefficients File	FALSE		
	Signed	TRUE		
	Truncated Output	FALSE		
	Options	**Configuration**	**Resources**	**Utilized**
Minimum Resource Area Optimized	Input Width	18	DSP	1
	Number of Coefficients	2	Registers	76
	Coefficients File	FALSE	LUTs	138
	Signed	FALSE		
	Truncated Output	FALSE		
	Options	**Configuration**	**Resources**	**Utilized**
Maximum Resource Area Optimized	Input Width	18	DSP	1
	Number of Coefficients	120	Registers	2206
	Coefficients File	FALSE	LUTs	3645
	Signed	TRUE		
	Truncated Output	FALSE		

Table 3: Resource Utilization

Parameters

Table 4 lists the parameters of the FIR Generator.

Parameter	Values	Default Value	Description
INPUT WIDTH	1 - 18	18	Input Data Width
COEFFICIENTS	Up to 120 coefficients	-	Coefficients for the filter taps
COEFFICIENTS FILE	0 / 1	0	Grab coefficients from a .txt or .hex file or enter manually
FILE PATH	\<path to the coefficients file\>		Absolute Path of the .txt / .hex is located with coefficients
OPTIMIZATION	Area / Performance	Area	Area vs Speed Optimization
NUMBER OF COEFFICIENTS	1 - 120	4	Number of Filter Taps
COEFFICIENT WIDTH	1 - 20	20	Bit width of Coefficients
COEFFICIENT FRACTIONAL BITS	1 - 20	0	Fractional bits of coefficients
OUTPUT DATA WIDTH	1 - 38	2	Custom output data width
TRUNCATED OUTPUT	0 / 1	0	Toggle for custom output data width
SIGNED	0 / 1	0	Signed input data and coefficients
IP TYPE	-	FIRG	Type of Peripheral
IP VERSION	-	\<ip_version\>	Version of Peripheral
IP ID	-	\<date_and_time\>	Date and Time of the generated Peripheral

Table 4: Parameters

Note:

- *FILE PATH is only available when COEFFICIENTS FILE is selected.*

- *COEFFICIENTS is only available when COEFFICIENTS FILE is not selected.*

- *For Area Optimization when loading coefficients from a hex file, the minimum number of filter taps is 2, while for all other cases, it is 1.*

- *The INPUT FRACTIONAL BITS play no role in determining the output width but is used for truncation and rounding purposes.*

- *For Performance optimization, the sum of INPUT WIDTH and COEFFICIENTS WIDTH is limited to 20 but no such limit exists for the implementation of Area optimization.*

- *OUTPUT DATA WIDTH only becomes available when TRUNCATED OUTPUT is selected.*

- *IP TYPE, IP VERSION and IP ID cannot be selected from the UI but can be seen at the top of the generated Verilog file.*

Design Flow

IP Customization and Generation

FIR Generator IP core is a part of the Raptor Design Suite Software. A customized FIR IP can be generated from the Raptor's IP configurator window as shown in Figure 5.
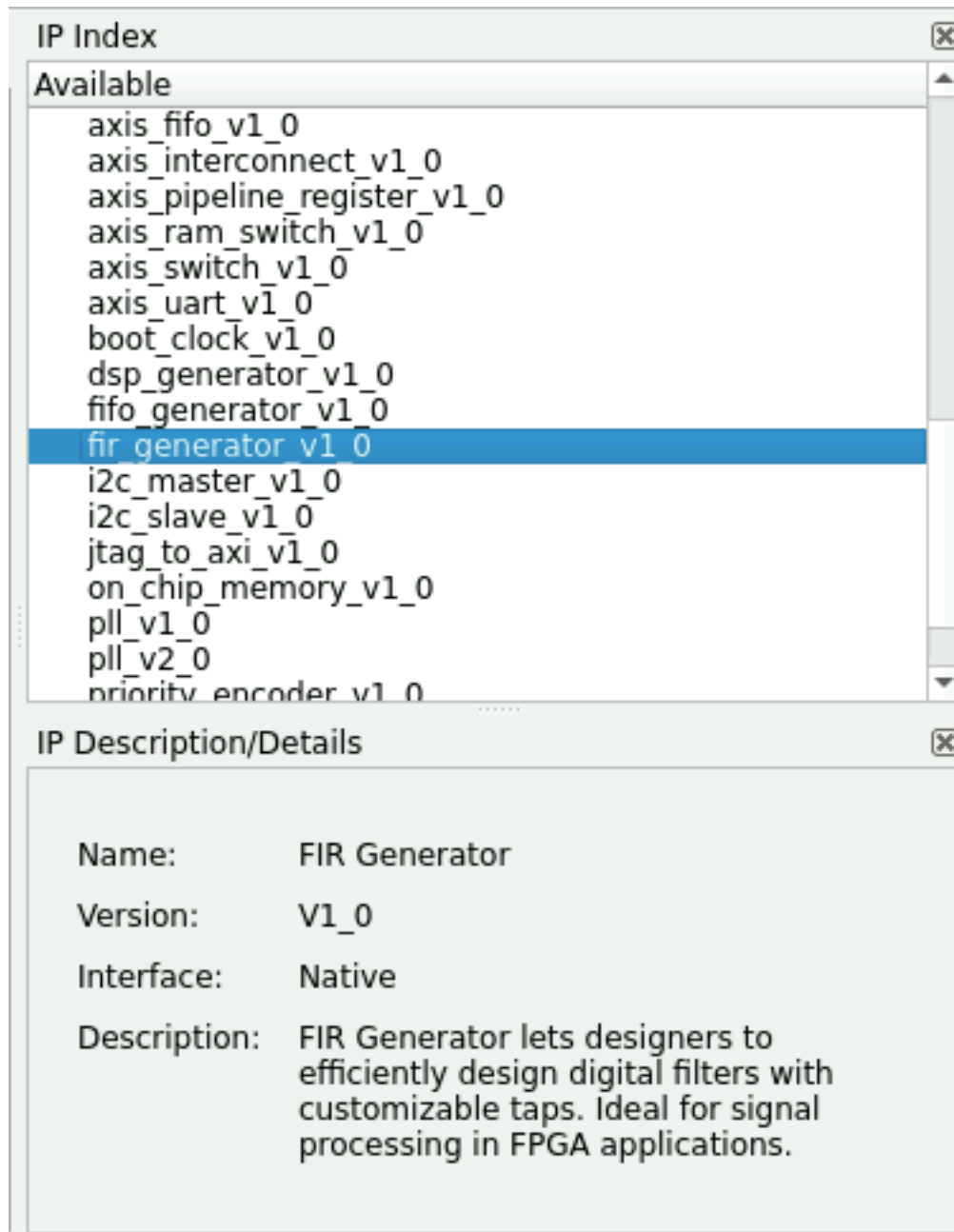

Figure 5: IP list

Parameters Customization

From the IP configuration window, the parameters of FIR Generator can be configured and IP features can be enabled for generating a FIR IP core that suits the user application requirement as shown in Figure 6. After IP Customization, the generated IP is made available to the user to be used in applications.
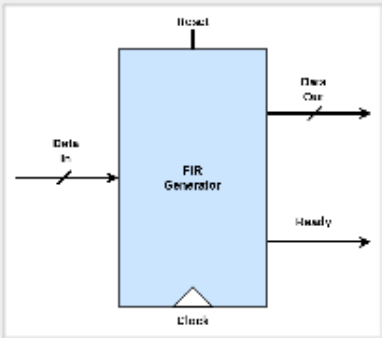

Figure 6: IP Configuration

Synthesis and PR

Raptor Suite is armed with tools for Synthesis and the generated post-synthesis net-lists can be viewed and analyzed from within the Raptor. The generated bit-stream can then be uploaded on an FPGA device to be utilized in hardware applications.

Test Bench

The FIR IP, developed in Verilog HDL, can be efficiently stimulated through various industry-standard methods. These methods encompass using simple Verilog test benches or employing more sophisticated approaches such as stimulating the FIFO through an operating system or via bare-metal firmware. The included test bench for this IP is Verilog-based and can be customized to align with the specific configuration of the generated FIR IP. After the generation of the IP, the source files and the simulation files are made available to the user along with the steps to simulate it via the bundled simulator by clicking on the "Simulate IP" button as shown in figure 7a. The waveform can then be viewed in the integrated wave-viewer by clicking the "View waveform" as shown in the figure 7b.

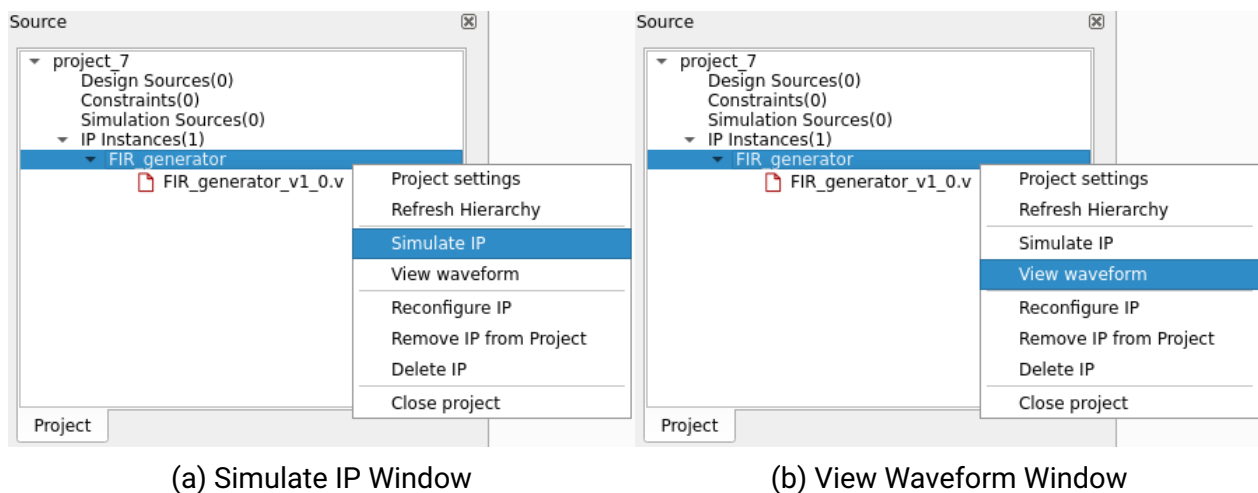(a) Simulate IP Window   (b) View Waveform Window

Figure 7: IP Source Window

The included testbench provides a random input of the size of the number of coefficients which are then computed inside the generated FIR IP. During simulation, the testbench meticulously computes the responses of the FIR filter based on the provided coefficients and input data. The results are systematically captured and logged in a VCD (Value Change Dump) file format, ensuring a thorough record of the IP's output data. The utilization of VCD files enhances the testbench's effectiveness in providing a granular view of the IP's operation, facilitating efficient debugging and verification within the broader digital design and verification workflow.

A simulation run with Coefficients: 4, 5, 6, 5, 4, 7 and an input data of 1, 2, 3, 4, 5, 6 is shown in the figures below, which the figure 8 showing the data output of the FIR Filter in Performance mode.

Figure 8: FIR Filter in Performance Mode
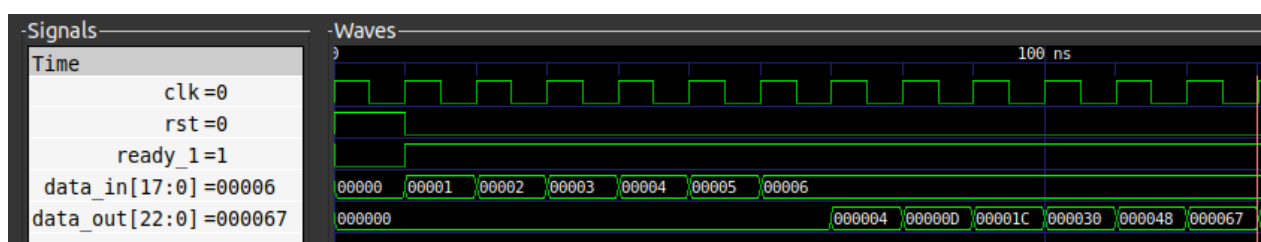
And the figure 9 showing the output of the filter in Area mode, with the fast_clock exposed.

Figure 9: FIR Filter in Area Mode
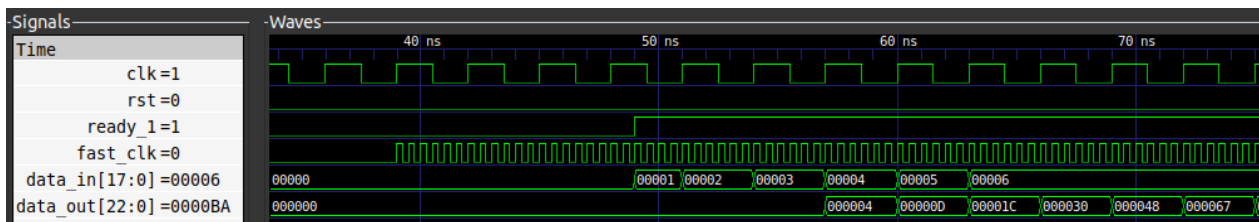
Release

Release History

Date	Version	Revisions
February 29, 2024	0.1	Initial version FIR Generator User Guide Document