[](http://www.opencores.org/)

LPFFIR IP Core  
Specification

*Author: Vladimir Armstrong*

*vladimirarmstrong@opencores.org*

**Rev. 1.1**

**April 2, 2019**

*This page has been intentionally left blank.*

**Revision History**

| **Rev.** | **Date** | **Author** | **Description** |
| --- | --- | --- | --- |
| 1.0 | 01/27/19 | Vladimir Armstrong | First Draft |
| 1.1 | 03/25/19 | Vladimir Armstrong | Added AXI-Stream Interface |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contents**

Introduction 1

Specifications 2

Architecture 3

Application 5

IO Ports 8

IO Waveforms 9

Appendix A 10

Impulse Response 10

Pole Zero Plot 11

Magnitude and Phase Response 12

Structure 13

Appendix B 14

Appendix C 16

Full adder Boolean algebra expressions 16

Full adder simplified Boolean algebra expressions 17

Index 18

## 

Introduction

Lowpass filter with finite impulse response (LPFFIR) IP core is characterized by one passband and one stopband, each specified by passband edge frequency and stopband edge frequency. The LPFFIR ideal filter gain is 6 in the passband and ideal attenuation in the stopband is zero, the filter design specifications include tolerance limits by which the ideal gains in the passband can be attenuated by value and ideal stopband can be gained by value. The LPFFIR tolerance scheme with edge frequencies and tolerance limits is shown in Figure 1.



Figure 1 LPFFIR tolerance scheme.

## 

Specifications

The LPFFIR Figure 1 specifications are shown in Table 1

Table 1 LPFFIR specifications.

|  |  |  |
| --- | --- | --- |
|  | Passband | Stopband |
| Ideal filter | 6 gain | 0 attenuation |
| Edge frequencies |  |  |
| Tolerance |  |  |

Architecture

The Figure 2 architecture is a realization of Figure 10 DSP structure with AXI-Stream (AXIS) protocol [4] wrapper. The LPFFIR core is made up of addition and delay () elements. The addition element function is implemented by Full Adder (FA) module and Ripple Carry Adder (RCA) module with hierarchy of Figure 3. The delay () element is implemented by Flip Flops (FF) in a series.

**

Figure 2 LPFFIR block diagram.

**

Figure 3 Module hierarchy block diagram.

The RCA module adds two 16-bit and one 1-bit binary number inputs A, B, and (CI) respectively and outputs one 16-bit and one 1-bit binary numbers S and (CO) respectively. The Figure 4 shows how multiple 1-bit add function FA modules are used to create 16-bit add function of RCA module.



Figure 4 RCA module block diagram.

The FA module adds three 1-bit binary number inputs A, B, and (CI) and outputs two 1-bit binary numbers S and (CO) as gate diagram shown in Figure 5 which is an implementation of [Full adder simplified Boolean algebra expressions].



Figure 5 FA module gate diagram.

Application

Application example of LPFFIR IP core is Discrete-Time Processing of Continuous-Time Signals[1] with block diagram of Figure 6 and frequency-domain illustration of Figure 7, if the input is bandlimited and the sampling frequency is high enough to avoid aliasing, then the overall system behaves as an LTI continuous-time system with the output is related to the input through an equation of the form

where effective continuous-time frequency responds

Using relation to convert from effective continuous-time filter specification to the discrete-time filter specification results an equation of the form



Figure 6 Discrete-time filtering of continuous-time signals system application.

The continuous-time overall system of Figure 6 with following requirements

1. Sample period shall be
2. The passband gain shall be 6.
3. The attenuated tolerance at the passband shall be 1.56 in the frequency band .
4. The gain tolerance at the stopband shall be 1.605 in the frequency band .

The mapping between the continuous-time and discrete-time frequencies only affects the passband and stopband edge frequencies and not the tolerance limits on frequency response magnitude [2].

The discrete-time block of Figure 6 with following requirements

1. The passband gain shall be shall be 6.
2. The attenuated tolerance at the passband shall be 1.56 in the frequency band .
3. The gain tolerance at the stopband shall be 1.605 in the frequency band .



Figure 7 Frequency-domain illustration of discrete-time filtering of continuous-time signals.

IO Ports

| **Port** | **Width** | **Direction** | **Description** |
| --- | --- | --- | --- |
| aclk\_i | 1 | Input | The global clock signal. All signals are sampled on the rising edge of ACLK. |
| aresetn\_i | 1 | Input | The global reset signal. ARESETn is active-LOW. |
| **AXI-Stream RX interface (AXIS\_RX)** | | | |
| rx\_tlast\_i | 1 | Input | TLAST indicates the boundary of a packet. |
| rx\_tvalid\_i | 1 | Input | TVALID indicates that the master is driving a valid transfer. A transfer takes place when both TVALID and TREADY are asserted. |
| rx\_tready\_o | 1 | Output | TREADY indicates that the slave can accept a transfer in the current cycle. |
| rx\_tdata\_i | 16 | Input | TDATA is the primary payload that is used to provide the data that is passing across the interface. |
| **AXI-Stream TX interface (AXIS\_TX)** | | | |
| tx\_tlast\_o | 1 | Output | TLAST indicates the boundary of a packet. |
| tx\_tvalid\_o | 1 | Output | TVALID indicates that the master is driving a valid transfer. A transfer takes place when both TVALID and TREADY are asserted. |
| tx\_tready\_i | 1 | Input | TREADY indicates that the slave can accept a transfer in the current cycle. |
| tx\_tdata\_o | 16 | Output | TDATA is the primary payload that is used to provide the data that is passing across the interface. |

Table 2: List of IO ports of LPFFIR AXIS module

IO Waveforms



Figure Discrete-time processing of continuous-time signals

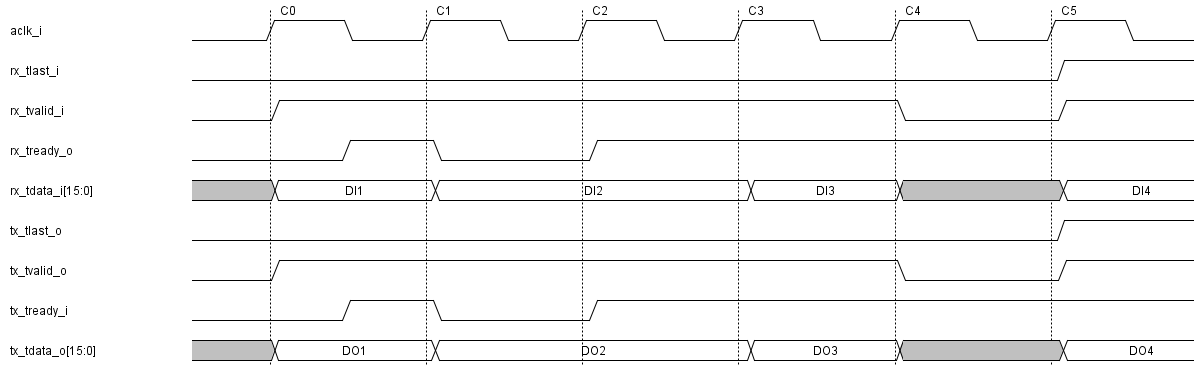


Figure Timing diagram AXIS\_RX/TX signals



Structure

The LPFFIR uses a direct form structure for a FIR linear-phase system. The DSP theory [3] is used for derivation and structure is shown in Figure 10.

### Impulse Response

0

-2

-1

0

1

2

3

4

5

6

7

0

1

1

1

1

1

1

0

0

### Pole Zero Plot

*Imaginary Part*

*Real Part*

### Magnitude and Phase Response

### Structure

Note: *M* is an odd integer.



Figure 10 Direct form structure for a FIR linear-phase system.



Expected Behavior

The LPFFIR expected behavior is generated from MATLAB simulation. The simulation source code and result plot are shown in Figure 11 and Figure 12 respectively.

Figure 11 MATLAB simulation source code.

1. % FIR difference equation of lowpass filter
2. b **=** **[**1**,** 1**,** 1**,** 1**,** 1**,** 1**];** a **=** **[**1**];**
3. % Response
4. n **=** **[**0**:**7**];**
5. h **=** impz**(**b**,**a**,**8**);**
6. **[**H**,**w**]** **=** freqz**(**b**,**a**,**100**);**
7. magH **=** abs**(**H**);** phaH **=** angle**(**H**);**
8. % Plot
9. subplot**(**4**,**1**,**1**);** stem**(**n**,**h**);**
10. title**(**'Impulse Response'**);** xlabel**(**'n'**);** ylabel**(**'h(n)'**)**
11. subplot**(**4**,**1**,**2**);**zplane**(**b**,**a**);**grid
12. title**(**'Pole-Zero Plot'**)**
13. subplot**(**4**,**1**,**3**);**plot**(**w**/**pi**,**magH**);**grid
14. xlabel**(**'Frequency in \pi units'**);** ylabel**(**'Magnitude'**);**
15. title**(**'Magnitude Response'**)**
16. subplot**(**4**,**1**,**4**);**plot**(**w**/**pi**,**phaH**/**pi**);**grid
17. xlabel**(**'Frequency in \pi units'**);** ylabel**(**'Phase in \pi units'**);**
18. title**(**'Phase Response'**)**

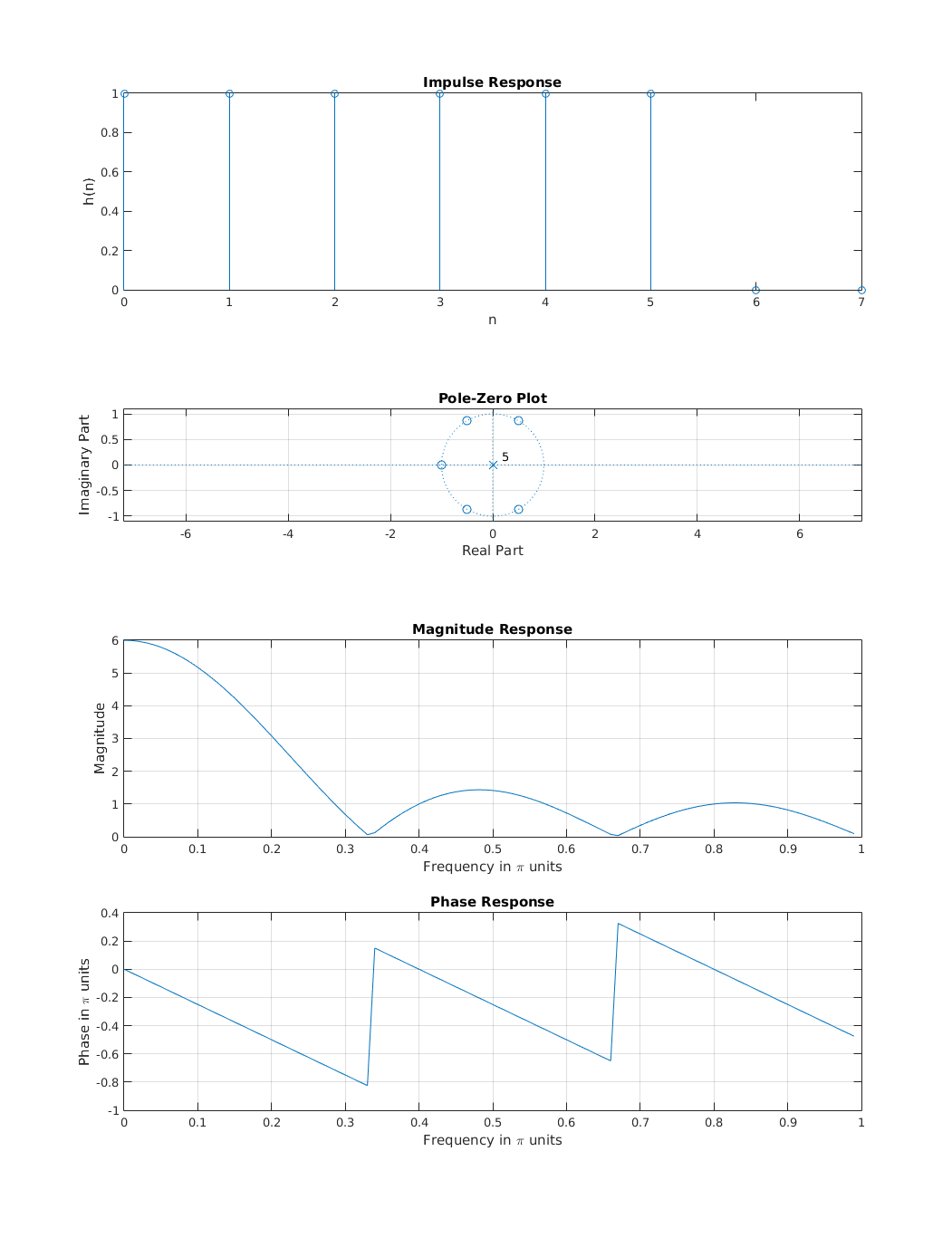


Figure 12 MATLAB simulation result plot.



### Full adder Boolean algebra expressions



The full adder Boolean expressions are derived from truth Table 3.

Table 3 Full adder truth table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | | | **Output** | |
| **A** | **B** | **CI** | **CO** | **S** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

### Full adder simplified Boolean algebra expressions

The K-map of Table 4 and Table 5 are used for simplifying Boolean algebra expressions of full adder.

Table 4 Full adder K-map of CO.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | 0 | 0 | 1 | 0 |
|  | 0 | 1 | 1 | 1 |

Table 5 Full adder K-map of S.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | 0 | 1 | 0 | 1 |
|  | 1 | 0 | 1 | 0 |

2. Oppenheim, A. V., & Ronald, W. S. (2009). Discrete-Time Processing of Continuous-Time Signals. In *Discrete-Time Signal Processing 3rd Edition* (pp. 197-172). Upper Saddle River, NJ: Pearson
3. Oppenheim, A. V., & Ronald, W. S. (2009). Filter Specifications. In *Discrete-Time Signal Processing 3rd Edition* (pp. 494-496). Upper Saddle River, NJ: Pearson
4. Oppenheim, A. V., & Ronald, W. S. (2009). Structures for Linear-Phase FIR Systems. In *Discrete-Time Signal Processing 3rd Edition* (pp. 403-405). Upper Saddle River, NJ: Pearson.
5. ARM publications (2010). *AMBA 4 AXI4-Stream Protocol Specification Version 1.0* (ARM IHI 0051).