International Journal of Electrical and Electronics Engineers
Volume 15, Issue No. 01, Jan-June 2023
ISSN (O) 2321-2055
ISSN (P) 2321 -2045

Synthesis and Implementation of an IIR Filter Using Verilog Language

Vaishali Sharma, R. P. Agarwal

Department of Electronics and Communication Engineering
Shobhit Deemed University ,Meerut, U.P.,India

Abstract

For different hardware designs at different abstraction layers, A language for describing hardware is called Verilog. This language is used for hardware design, modeling, and inquiry. This paper illustrates hardware implementation strategies for IIR filters using synthesizable Verilog. The MATLAB FDA Tool (Filter Design and Analysis Tool) is required to do the calculation of filter coefficients. After that, the filter is implemented and simulated using the Xilinx xc5vlx50t-3ff665. The filter is implemented using a fixed-point number representation. The parallel architecture of the filter is part of its suggested design.

Introduction

Certain digital signal processing (DSP) algorithms can be executed by FPGAs because they have the necessary speed and logic capacity. The architecture of FPGA makes it possible to implement DSP operations in highly parallel with high performance, and designers may trade off performance for device area by choosing the right amount of parallelism for their workloads. Certain digital signal processing (DSP) algorithms allow designers to implement their designs more efficiently and with higher performance. FPGA is well suited for DSP due to two features: in addition, designers may compromise between performance and device area by choosing the right amount of parallelism for their activities, thanks to its architecture, which enables extremely performant, highly parallel implementation of DSP operations. Designers can improve performance and optimize systems for less money. Register Transfer Level (RTL) hardware specifications are supported by many programming languages today, one of which is Verilog. The main reason to use Verilog is because it is a regular language that is independent of vendors and technologies, making it reusable and portable.

The necessity to ultimately put the digital filter into practice has emerged given how important it is to digital signal processing. Digital filters may be divided into two main groups: those with infinite impulse responses (IIR) and those with finite impulse responses (FIR) . For each output sample of a FIR filter, a finite sum is calculated using the filter equation convolution. Because of the availability of powerful software tools, designing digital filters is a simple process. Savadi A. and coworkers created a convolution filter, an IIR filter built with the convolution approach. They demonstrated that the Urdhava-Tiryagbhyam method of multiplication, which makes a tangible difference in practise, is indeed successful. The vertex-5 FPGA was used to perform this multiplication in Xilinx ISE 14.7.1. With the help of MATLAB programmes, Toledo-Pérez et al. were able to acquire and validate the filter coefficients, which they afterwards applied to a filter design for an FPGA to implement. The filter was implemented using the Basys 3 Artix-7 FPGA Trainer kit from the Xilinx series. P. Jubair Ahamed and M. Abdul Haseeb proposed hybrid programming methods for designing digital filters using MATLAB and

VB. Reduced computational complexity and partial preservation of the multiplier combination architecture are two benefits of the proposed approach that facilitate efficient FPGA design.

An improved design of reconfigurable IIR filters, which are widely used in real-time applications, was introduced by Debarshi D. and Himadri S. D. Hardware description language is used to create the filter, which is subsequently tested on a Xilinx Virtex-5 board. In this paper, we introduce the FDA software tool for designing IIR digital filters. Using ISE14.7 and Verilog, a parallel and programmable multiplexing IIR filter design is simulated and then implemented on an FPGA board .

Digital Filter Design Methodology

When applied to a signal in its intermediate form (which may be represented as a floating-point or a fixed-point number), digital filters conduct digital mathematical operations. In its general form, the equation for the difference in a digital filter is:

$$y(n) = \sum_{i=0}^{N} a_i x(n-i) - \sum_{i=1}^{N} b_i y(n-i)$$

(1)

where x(n-i) denotes the most recent input, y(n) denotes the most recent output, y(n-i) denotes the most recent output, N denotes the filter's order, i a's denote the feed-forward coefficients of the filter, which correspond to the filter's zeros, bi's denote the filter's feedback coefficients, and is the set of all possible coefficients for the filter. applications for RF and high-speed transmission, audio equalization, and biomedical sensor signals processing are just a few of the many modern signal processing systems that make use of IIR digital filters. The feedback coefficients of an IIR filter are never all zero. This means that if the filter includes one or more poles, it will always provide an output after being aroused by an impulse provided by the feedback term. Figure 1 illustrates the result is calculated by multiplying a collection of input samples produced from a set of coefficients and adding that to a series of prior outputs. One of the key disadvantages of FIR filters include enormous amount of memory and arithmetic processing needed. As a result, their usefulness decreases in a variety of contexts. However, IIR filters have difficulties in design and stability issues, despite requiring significantly less memory and fewer arithmetic operations. IIR filters may have superior performance and lower system costs despite their complexity and increased design. The filter coefficients are responsible for the filter's behaviour. A generic IIR filter can be calculated using equation (1) as follows:

$$y(n) = a_0 x(n) + a_1 x(n-1) + \ldots + a_N x(n-N) + b_1 y(n-1) + b_2 x(n-2) + \ldots + b_N y(n-N)$$

(2)

As can be seen from the above equation, the coefficients of the filter play a crucial role in digital filter design. The computation involved in computing a digital filter's constant coefficients is substantial and is often carried out with the aid of a software programme. The Filter Design and Analysis (FDA) tool in MATLAB is a powerful resource for developing filters . In MATLAB's signal processing toolbox, you'll find the FDA, a GUI that may be used to create and study filters. The FDA receives input in the form of filter specs. As shown in Figure 2, the FDA tool's filter design window allows users to choose between two filter types (FIR and IIR) and customise every aspect of the filter's design. The low-pass filter described in Table 1 has been chosen and designed for use in a hardware implementation.

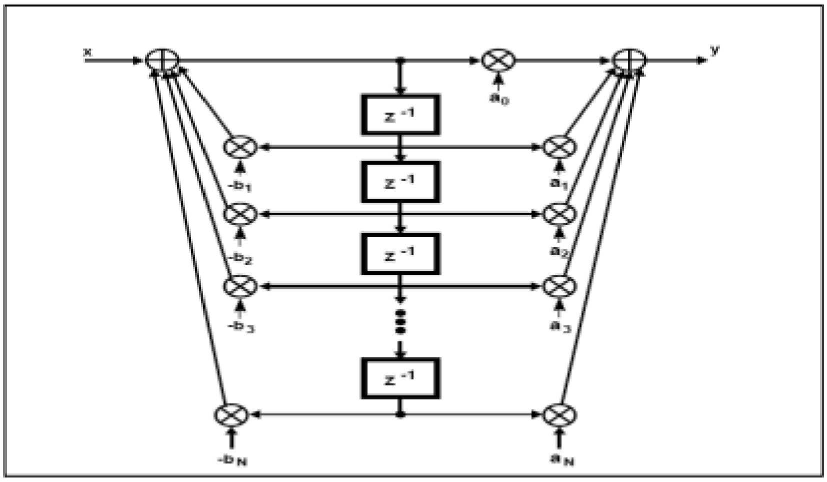
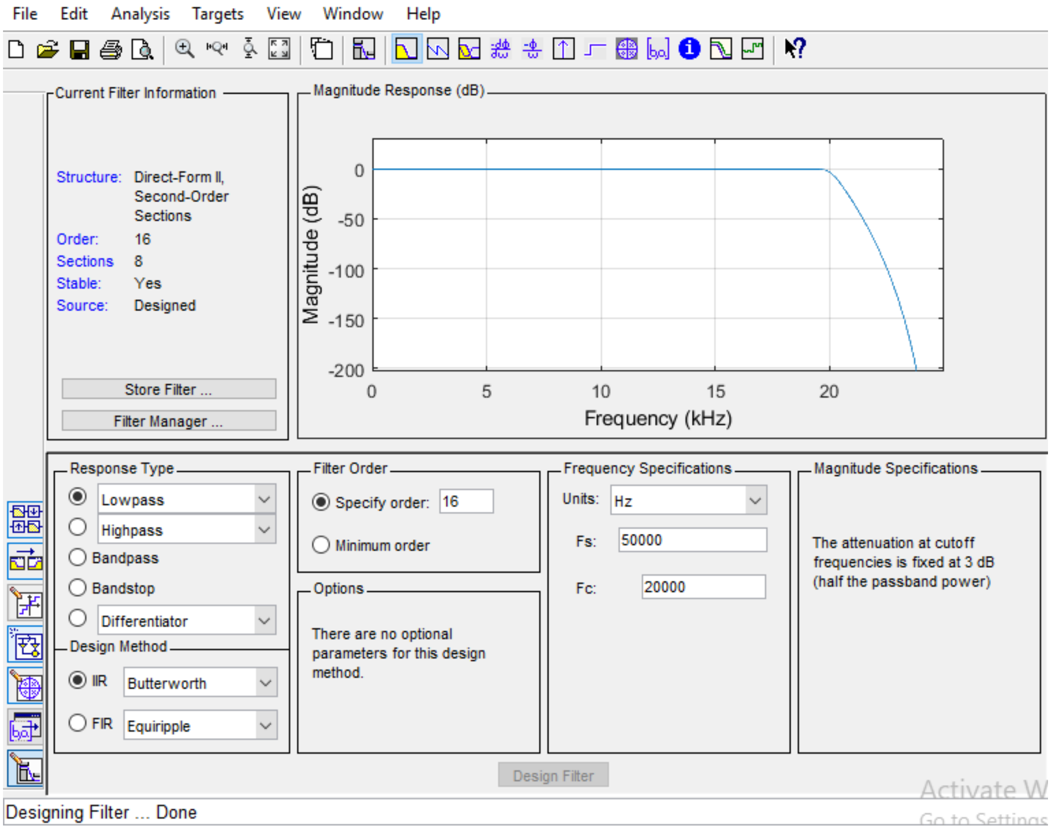


Figure 1. IIR filter (Direct form II)



Structure: Direct-Form II, Second-Order Sections
Order: 16
Sections: 8
Stable: Yes
Source: Designed

Response Type: ● Lowpass ○ Highpass ○ Bandpass ○ Bandstop ○ Differentiator

Design Method: ● IIR Butterworth ○ FIR Equiripple

Filter Order: ● Specify order: 16 ○ Minimum order

Options: There are no optional parameters for this design method.

Frequency Specifications: Units: Hz Fs: 50000 Fc: 20000

Magnitude Specifications: The attenuation at cutoff frequencies is fixed at 3 dB (half the passband power)

Designing Filter ... Done

Figure 2 Filter design using MATLAB FDA tool

Filter Specifications	**Options**
Design Method	IIR Butterworth
Response Type	Lowpass Filter

Frequency Specifications	
Sampling Frequency (Fs)	50000Hz
Cutoff Frequency (Fc)	20000Hz
Specify filter order	16
Magnitude Specifications	The attenuation at cutoff frequencies is fixed at 3 Db

Table 1 The Designed Filter Specifications.

FPGA Implementation of IIR Filters

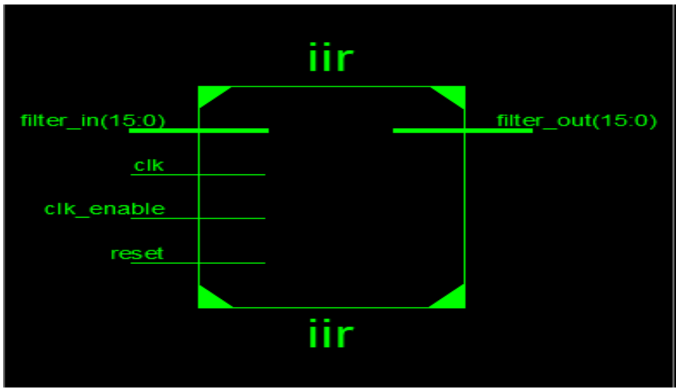
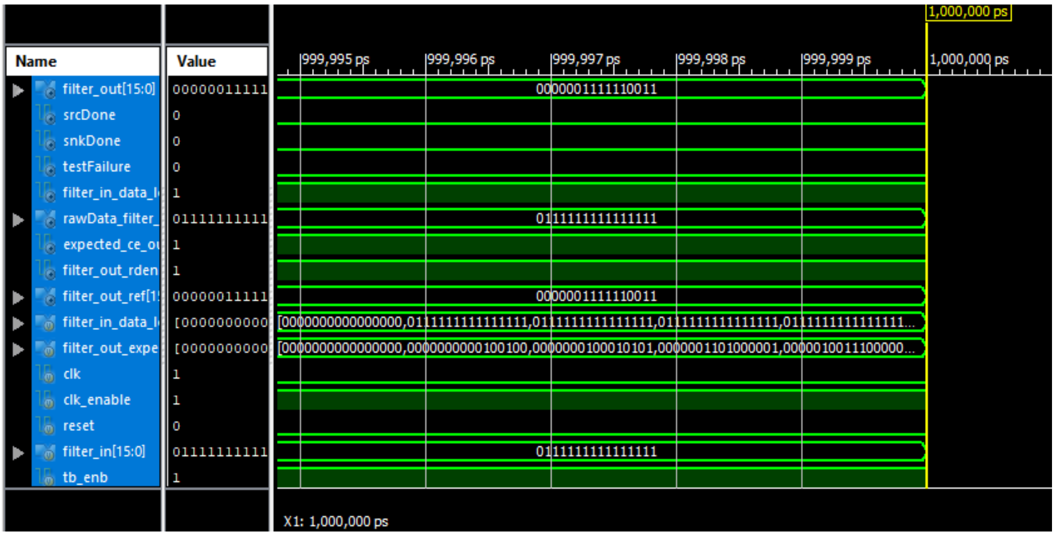


Figure.3 block schematic of low pass filter(lpf)



Name	Value
filter_out[15:0]	00000011111
srcDone	0
snkDone	0
testFailure	0
filter_in_data_l	1
rawData_filter_	01111111111
expected_ce_ou	1
filter_out_rden	1
filter_out_ref[1	00000011111
filter_in_data_l	[0000000000
filter_out_expe	[0000000000
clk	1
clk_enable	1
reset	0
filter_in[15:0]	01111111111
tb_enb	1

Figure. 4 simulation result of iir filter

SIMULATION RESULTS

WEBPACK project navigator 14.7, developed by Xilinx, has been used to synthesise and simulate an IIR filter. A digital filter is implemented using a direct-form approach. Xilinx 14.7 ISE (Integrated Software Environment) was used to create an IIR filter in a VIRTEX 5 device in the xc5vlx50t package operating at speed grade 3.

REFERENCES

[1] Turney R. D., Dick C., David BP, and Hwang J. Modeling and implementations of DSP FPGA solutions by Xilinx Inc., San Jose, CA 95124, USA 1-16

[2] Wood, R., and Mcullister, J. (2008). FPGA-Based Implementation of Signal Processing Systems, John Wiley & Sons, ISBN 978-0-470-03009-7.

[3] Hauck S. and Dehon A. (2008), Reconfigurable Computing: The Theory and Practise of FPGA-Based Computation, Elsevier Inc., ISBN 978-0-12-370522-8.

[4] Pedroni V. (2004), Circuit Design with VHDL, Massachusetts Institute of Technology (MIT) Press, ISBN 0-262-16224-5

[5] Kunchevo A. and Yanchev G. (2008), Synthesis and Implementation of DSP Algorithms in Advanced Programmable Architectures, Proceeding of the International Scientific Conference on Computer Science (ISCCS).

[6] Pouki VM, emva A, Lutovac MD, and Karnik T. (2008 Chebyshev IIR filter sharpening implemented on FPGA transfer 100 1–11.

[7] Savadi A.,Yanamshetti R., and Biradar S. (2016Design and implementation of 64-bit IIR filters using Vedic multipliers Procedia Computer Science 85: 790–797

[8] Toledo-Pérez DC, Martnez-Prado MA, Rodrguez-Reséndiz J, Arriaga ST, and Márquez Gutiérrez M2017 IIR digital filter design implemented on an FPGA for myoelectric signals In the 2017 XIII International Engineering Congress (CONIIN), IEEE 1–7

[9] Jubair Ahmed P. and Abdul Haseeb M2020 Implementation of Digital IIR Filter Design Based on Field Programmable Gate Array Materials Today: Proceedings,ISSN 2214-7853

[10] Datta D. and Dutta HS 2021: High Performance IIR Filter Implementation on FPGA, Journal of Electrical Systems and Information Technology, 8(1):1–9.

[11] Vinger KA and Torresen J. 2003. Implementing the evolution of FIR filters efficiently in an FPGA In the NASA/DoD Conference on Evolvable Hardware, IEEE 26–29

[12] Demirsoy S, Dempster A, and Kale I (2004) Efficient implementation of digital filters using novel reconfigurable multiplier blocks (REMB) In Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems, and Computers, IEEE (1) 461-464

[13] Mokhtari N. and Rahmanian S. (2006), Hardware Implementation Analysis for Digital Filters In Proceedings of the 14th Iranian Conference on Electrical Engineering (ICEE'06).

[14] WanhammarL1999DSP Integrated Circuitsacademic press series in engineering ISBN 0-12- 734530-2, UK