IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

Synthesis and Implementation of IIR Filter using VHDL Language

To cite this article: N. T. Gadawe *et al* 2021 *IOP Conf. Ser.: Mater. Sci. Eng.* **1152** 012026

ECS
The Electrochemical Society
Advancing solid state & electrochemical science & technology
18th
239th ECS Meeting with IMCS18
DIGITAL MEETING • May 30-June 3, 2021
Live events daily • Free to register
Register now!

Synthesis and Implementation of IIR Filter using VHDL Language

N. T. Gadawe[1*], **T. A. Fathi**[1], **S. L. Qaddoori**[1] **and R. W. Hamad**[1]

[1]Electronic Engineering Department, Electronics Engineering College, Ninevah University, Mosul, Iraq

[*]E-mail: noor.gadawe@uoninevah.edu.iq

Abstract. VHDL is a hardware description language of different hardware architectures at divers' levels of abstraction. The language is used for design, modelling, and investigation of hardware. In this paper, it is presented hardware implementation architectures of IIR filter using synthesizable VHDL. The MATLAB Filter Design and Analysis Tool (FDA Tool) is usedto calculate filter coefficients. Then, the filter is simulated and implemented on the Xilinx xc3s500efg320-4. A fixed-point number representation is used in the implementation of the filter. The proposed design of the filter includes two architectures: parallel architecture and programmable multiplexing architecture, each type has its advantages and disadvantages. In the results, some figures depending on the size of design (filter length) are introduced in the synthesis report.

1. Introduction

Field programmable gate array (FPGA) possessadequate performance and logic capacity to applysomeDigital Signal Processing (DSP) algorithms efficiently [1].

FPGA has two advantages that make it suitable for DSP, the first one is that its architecture is well appropriated for vastly parallel implementation of DSP purposes with high performance, the second is that designers can make a tradeoff between performance vs. device area by choosing the suitable level of parallelism to employ their tasks. Designers can achieve higher performance and optimization design with low cost [2]

Very High-Speed Integrated Circuit Hardware Description Language (VHDL) is one prevalent programming language that supports Register Transfer Level RTL hardware descriptions [3]. Theessential motivation to utilize VHDL is thatit is a regular, vendor/technology independent language, and is hencereusable and portable [4].

Digital filter is a very significant part of DSP, thereforethe need to implement it has been appeared. The two basic types of digital filtersareinfinite impulse response (IIR)and finite impulse response (FIR). Where, An IIR needs an infinite sum to be achieved. An FIR filter involves a finite number of sample values, where the filter equation convolutionis used to find a finite sum per output sample [5]. Digital filter design is an uncomplicated task because it isimplemented using influential software tools [6].

In [7], Savadi A., and others designed an IIR filter using the convolution method, which is called convolution filter, they showed the effectiveness of the Urdhava-Tiryagbhyamtechnique for multiplication that carries a difference in the real process of multiplication itself. This multiplicationwasachieved in Xilinx 13.4 ISE and applied on vertex-5 FPGA.

In [8] Toledo-Pérez et al, implemented a developed IIR filter on FPGA, and used MATLAB functions to obtain and prove the filter coefficients then transferred these coefficients to an FPGA filter design. The filter was applied within a Basys 3 Artix-7 FPGA Trainer kit of the Xilinx family. In [9] P. Jubair Ahamed, M. Abdul Haseeb, presented hybrid programming methods which use MATLAB and VB in digital filter design. The proposed algorithm decreases the difficulty of computing as well as partlypreserves the multiplier combinationarchitecture, which leads to effective construction in FPGA.

Debarshi D. and Himadri S. D. introduced an enhanced design of reconfigurable IIR filter that isbroadlyutilized in real-time applications in [10]. The designed filter is designed using hardware description language thenverified on Virtex-5 board from Xilinx family.

In this paper, IIR digital filter design using the FDA tool is presented. parallel and programmable multiplexing IIR filter architectures are simulated and implemented on an FPGA kit using ISE14.7i and VHDL.The other sections of the paper is structured as follows: Section 2 describes the digital filter design methodology. IIR filter is synthesized using VHDL in section 3. The simulation and implementation results using a Xilinx FPGA device are given in sections 4 and 5. Finally, section 6 concludes this paper.

2. Digital Filter Design Methodology

Digital filters are type of electronic filters, these filters perform digital mathematical operations on a middle form of a signal (could be represented using floating-point or fixed point) [11]. The digital filter difference equation general form is:

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

where the x(n-i)'s is recent or previous filter inputs, y(n) is the recent filter output, the y(n-i)'s are previous filter outputs, N is the filter's order, the a_i 's are the filter's feed-forward coefficients corresponding to the zeros of the filter, the b_i 's are the filter's feedback coefficients corresponding to the poles of the filter, and [12].

IIR digital filter is widely applied in recent signal processing systems such as high-speed telecommunication/RF applications, audio equalization, and biomedical sensor signal processingIIR filters have at least one nonzero feedback coefficients. That is to say, due to the feedback term, the filter will be excited with an impulse and there is always an output if the filter has one or more poles. The output is computed from two sets, a set of input samples and a set of previous outputs, these inputs and outputs are multiplied by a set of coefficients and then accumulated together to yield the output as shown in Figure 1 [12].

The large amounts of memory and arithmetic processing required are one of the major disadvantages of FIR filters. This makes them less used in several applications. Alternatively, IIR filters require much less memory and fewer arithmetic operations, their disadvantages are the design difficulty and they suffer from stability problems. In spite of the design complexity and Although the design is much more, IIR filter usage may led to a higher performance and lower system cost [13]. The filter's behavior is determined by the filter coefficients. From equation (1), a general IIR filter is determined by:

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

As it can be noticed from the above equation, the digital filter design relies on the coefficients of the filter. Constant coefficients calculation of a digital filter includes an extensive amount of computation, these computations are generally accomplished using a software tool. MATLAB has a packed tool for filter design, The Filter Design and Analysis (FDA) tool [14]. The (FDA) is a graphical user interface (GUI) tool available in the signal processing toolbox of MATLAB, it could be used for designing and analyzing filters. Filter specifications are fed as inputs to the FDA. Figure 2 shows the filter design window of the FDA tool, when the filter design is completed from the FDA window, one can select filter type (FIR or IIR) and all specifications of the filter design. For hardware implementation, the lowpass filter has been selected and designed with specification design as shown in Table 1.

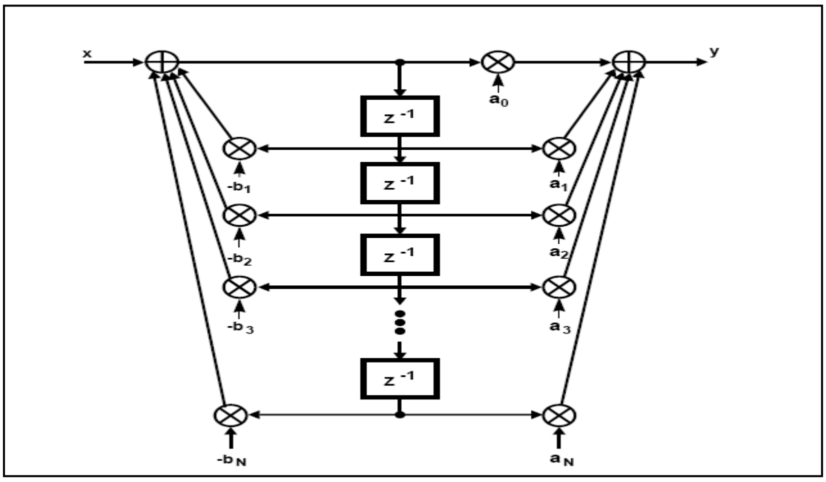
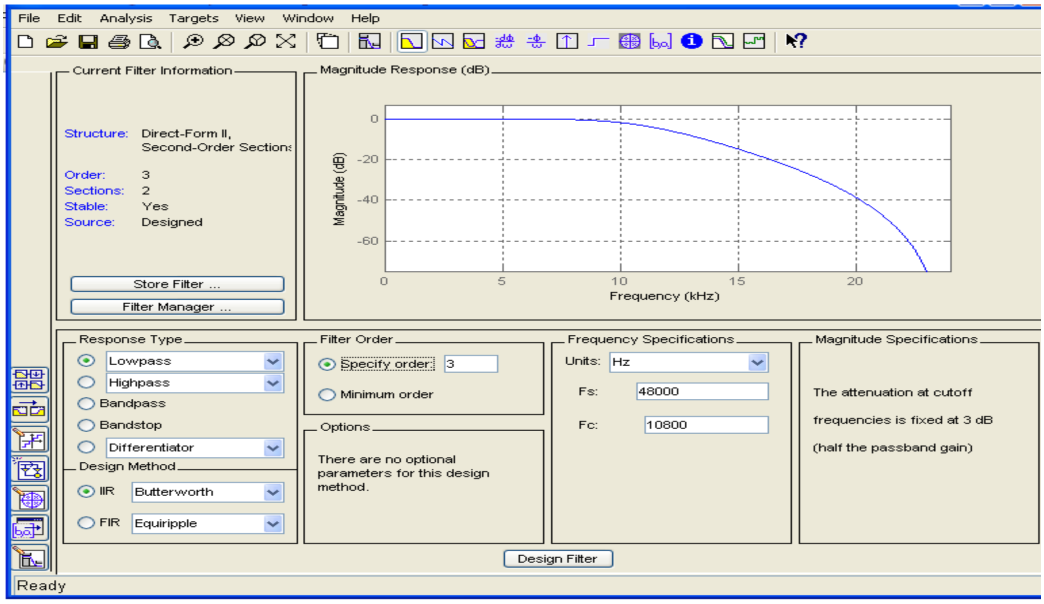


Figure 1. IIR filter (Direct form II)



Structure: Direct-Form II, Second-Order Sections

Order: 3

Sections: 2

Stable: Yes

Source: Designed

● Lowpass

○ Highpass

○ Bandpass

○ Bandstop

○ Differentiator

● IIR Butterworth

○ FIR Equiripple

● Specify order: 3

○ Minimum order

There are no optional parameters for this design method.

Units: Hz

Fs: 48000

Fc: 10800

The attenuation at cutoff frequencies is fixed at 3 dB (half the passband gain)

Figure 2. Filter design using MATLAB FDA tool.

Table 1. The Designed Filter Specifications.

Filter Specifications	Options
Design Method	IIR Butterworth
Response Type	Lowpass Filter
Frequency Specifications	
Sampling Frequency (Fs)	48000Hz
Cutoff Frequency (Fc)	10800Hz
Specify filter order	3
Magnitude Specifications	The attenuation at cutoff frequencies is fixed at 3 dB

3. Synthesis of IIR Filter using VHDL

In general, DSP algorithms are realized in two main number representations: floatingand fixed-point. In practicalDSP applications, the extra precision and range provided by floating-point are not necessary. In other words, floating-point representations tend to more costly and power-consumingbecause of additional hardware difficulty, Consequently, the requirement to lower power consumption anddesign cost led the inventors to utilize the fixed-nonrepresentational place of floating-point [10]. It can be noted that the FDA tool utilizes double-precision floating-point representation for the design calculations. The double-length floating-point representation for the filter coefficients poses huge challenges in terms of resourcesandcost.

To overcome this, the filter coefficientsneed to be quantized to fixed point representations, this can be done by selecting the quantization parameter seton the sidebarbutton of the FDA tool and selecting the suitable word length of the coefficients and input outputs. The filter coefficients are fractional values as shown in Table 2. In the implementation, it is not possible to implement those values exactly due to the use of finite word length representation, by using a large number of bits, the results will be more accurate and near the actual values, of course, this requires additional hardware area and costs. 8-bits representation for the coefficients and input data is selected. The data type used for representing the data and coefficients are "signed" representation, because the results may be negative values.

3.1 The architecture of IIR Filters Parallel Form

In general, an IIR has fewer coefficients, but the compulsory output feedback can perform the employment of the circuit more difficult. If the coefficients are not selected correctly, a stable IIR filter can become unstable. The most straightforward implementation is a parallel structure with fixed-point coefficients. There are several possibilities to implement the structure. One of them is the direct form II as shown in Figure 3. From Figure 3, the hardware implementation requiresa D-flip flop register for delay elements, latches to store coefficients, and different arithmetic operations (composed of multipliers and adders). The filter is compiled in Xilinx ISE14.7 version. The timing simulation is shown in Figure 4. For the entity declaration clk, rst, x are inputs of the filter, y is an output of the filter, x1, x2, x3, and x4 are tap delay lines which refer to x(n), x(n-1), x(n-2), and x(n-3). a0=a3, a1=a2, b1, b2, and b3 are the coefficients of the filter in hexadecimal. y1, y2, y3 are output delay y(n-1), y(n-2), and y(n-3).

Table 2. Coefficients of IIR (low-pass) filter from FDA tool**.**

Coefficients	Decimal Values	8bits Representations
a0	0.13006301	0.12890625
a1	0.39018903	0.390625
a2	0.39018903	0.390625
a3	0.13006301	0.12890625
b1	0.28814049	0.2890625
b2	-0.35476049	-0.35546875
b3	0.02666632	0.2734375

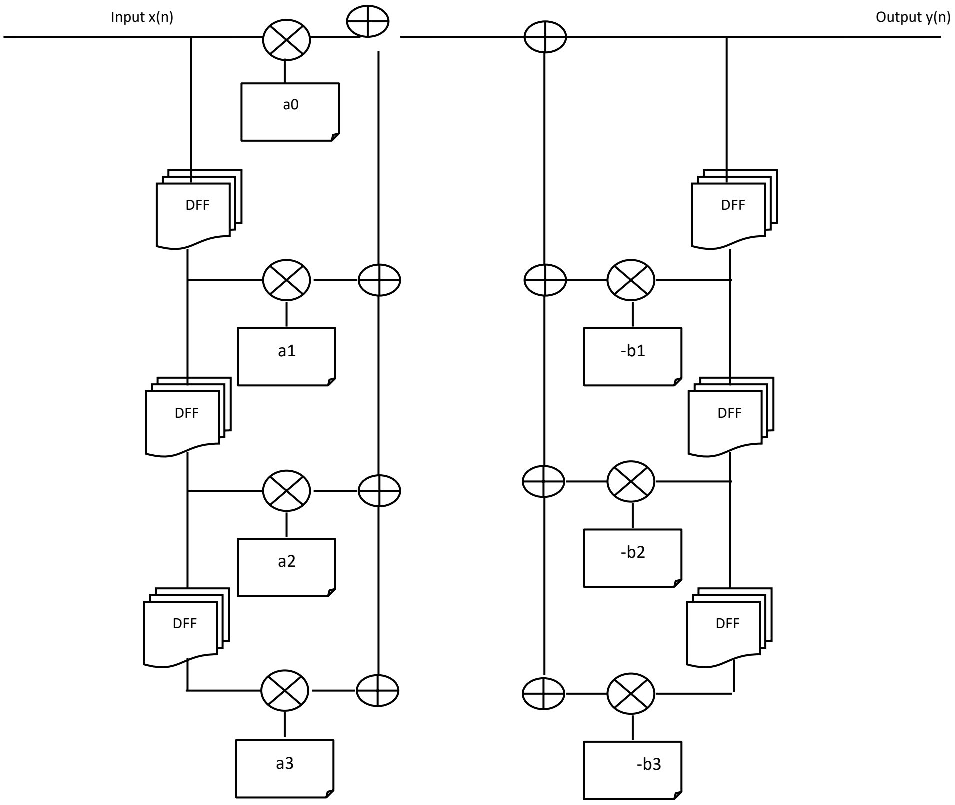


Figure 3. RTL Schematic of parallel form third orderIIR filter

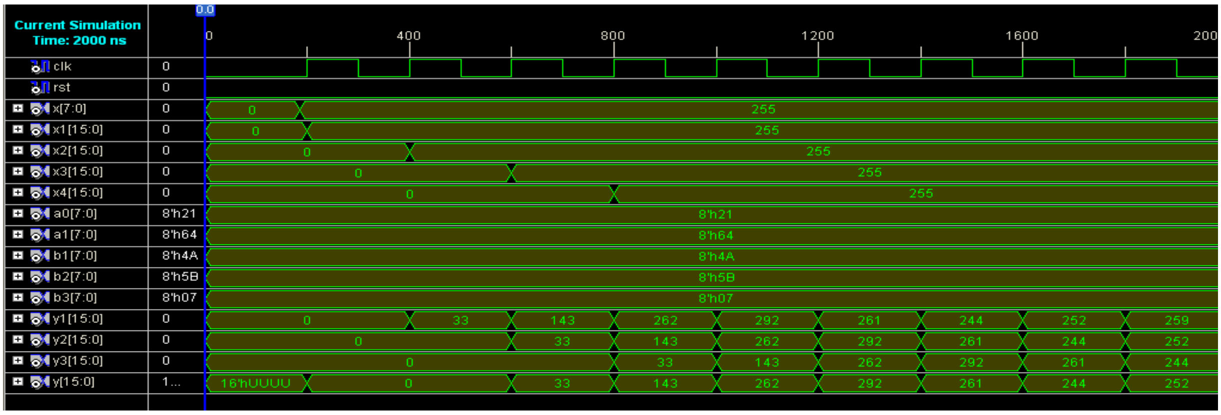


Figure 4. Timing Simulation of parallel formIIR third order

3.2 The architecture of IIR Filters Programmable Multiplexing Form

The implementation of programmable multiplexed IIR filter is built a dataflow structure using one multiplier and one adder which is constituting a multiply and accumulate (MAC) unit for arithmetic operations. The elements are kept in suitable synchronous register banks. The dataflow representation of the implementation architecture for the third-order IIR filter is shown in Figure 5. The timing simulation is shown in Figure 6.

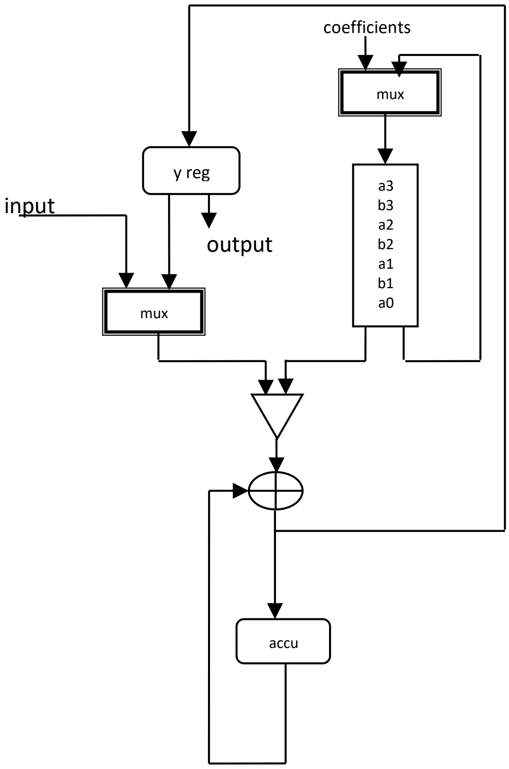


Figure 5. RTL schematic of multiplexing formIIR Filter

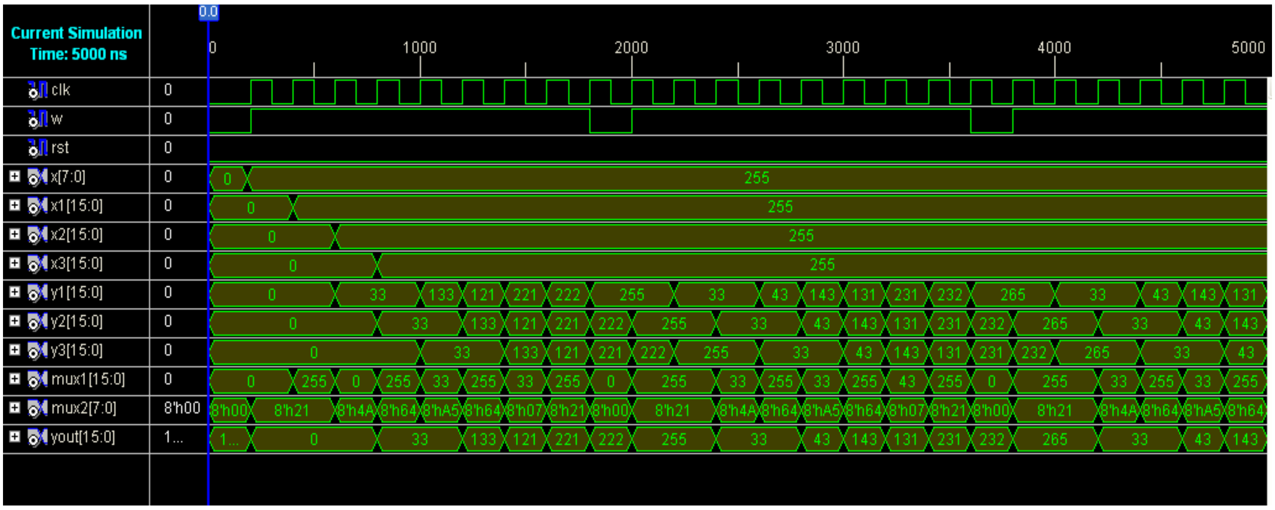


Figure 6. Timing Simulation of programmable multiplexing form

4. FPGA Implementation of IIR Filters

The filters are synthesizedin VHDL using ISE 14.7 tool. The FPGA implementations are realized on SPARTAN3E XC3S500EFG320-4[15]. All implemented filters require 8-bit input samples and their coefficients need 8-bits precision. The major challenge in parallel form is the large number of multipliers utilized, these increased as soon as filter length is increased, compared with programmable multiplexing form which required one multiplier for different filter lengths as shown in Figure (7), but the programmable multiplexing formarchitecture consumed more slices as shown in Figure (8).

Figure 9 demonstrates the relationship between the required number of gates and the filter length, where the number of gatesis increased asthe filter length increased, but this increasing is greater for the parallel form compared with the multiplexing form. The same thing in 4-input (Look Up Tables) LUT utilization as shown in Figure (10). From simulation results, the throughput in parallel form equal maximum frequency/1clk cycles, for multiplexing form the throughput equal maximum frequency/8 clk cycles. therefore, the parallel approach high performance with no constraint in hardware consumed, in contrast, the programmable multiplexed approach is more efficient in terms of resources needed and utilized, but the performance is lower.

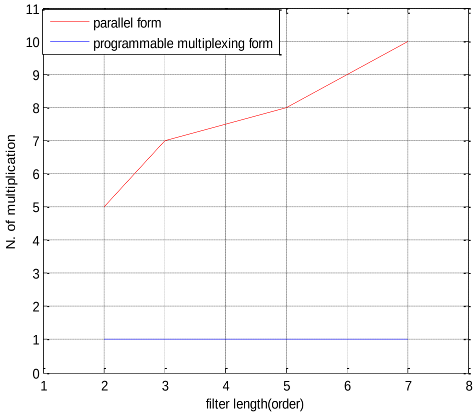


Figure 7. N. of multiplier as function of taps

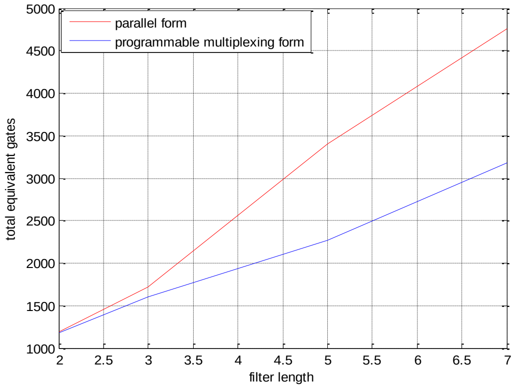


Figure 8. Slices as function of taps

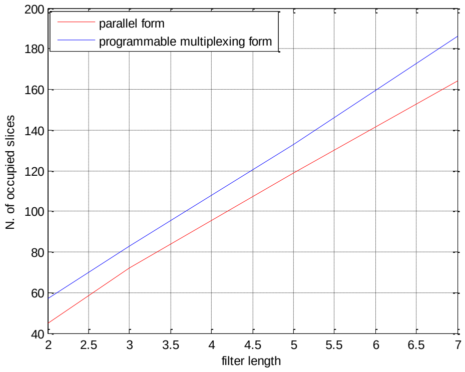


Figure 9. N. of Gates as function of taps

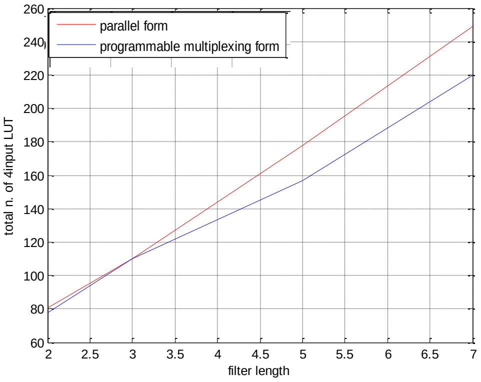


Figure 10. N. of 4-input LUT as a function of filter order

5. Conclusions

Today digital filter design is greatly facilitated by the availability of numerous computer software programs. FDA package is such tool, which used for designing and analyzing filter in this paper. Synthesis, simulated IIR lowpass Butterworth filter using ISE14.7 and VHDL language, then implemented on the FPGA to achieve hardware verification. The target device is spartan3e xc3s500efg320. The research is focused on the architecture part of the design. The proposed design of the filter includes two architectures parallel architecture, and programmable multiplexing architecture, with different filter lengths. The direct approach high performance with no constraint in hardware, in contrast, the programmable multiplexing approach minimizes hardware requirement while yielding less performance. In future works, practical application of IIR filter can be introduced and other architectures of IIR filters can be implemented which can be compared with FIR filter architecture.

6. References

[1] Turney R D, Dick C, David BP and HwangJModeling and implementations of DSP FPGA Solution *XilinxInc* san Jose, CA95124, USA 1-16

[2] WoodR andMcullisterJ2008FPGA based Implementation of Signal Processing Systems*john Wileyand sons* ISBN 978-0-470-03009-7.

[3] HauckS andDehonA2008Reconfigurable Computing, the theory and practice of FPGA based Computation*ElsevierInc* ISBN 978-0-12-370522-8.

[4] PedroniV2004Circuit Design with VHDL*Massachusetts institute of technology (MIT) press* ISBN 0-262-16224-5.

[5] Kunchevo A andYanchev G2008 Synthesis and Implementation of DSP Algorithms in advanced Programmable Architectures *Proceeding InInternational Scientific Conference Computer ScienceISCCS*.

[6] Poučki VM, Žemva A, Lutovac MD and Karčnik 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, Martínez-Prado MA, Rodríguez-Reséndiz J, Arriaga ST andMárquez-Gutiérrez MÁ2017 IIR digital filter design implemented on FPGA for myoelectric signals*In 2017 XIII International Engineering Congress (CONIIN)*IEEE 1-7

[9] JubairAhamed 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 2003Implementing evolution of FIR-filters efficiently in an FPGA *In NASA/DoD Conference on Evolvable Hardware* IEEE26-29

[12] DemirsoyS,DempsterA and KaleI2004Efficient 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] MokhtariN andRahmanianS2006Hardware Implementation Analysis for Digital Filters*In Proceedings of the 14th Iranian Conference on Electrical Engineering (ICEE'06)*.

[14] WanhammarL1999DSP Integrated Circuits*academic press series in engineering* ISBN 0-12-734530-2, UK.