

FPGA Schematic Implementations and Comparison of FIR Digital Filter Structures

O. Coşkun, and K. Avci

Abstract—In this study, we investigate the FPGA schematic implementations of finite impulse response (FIR) digital filters for three fundamental structures on Altera DE2-115 board without requiring any other software packages such as DSP Builder and Matlab Simulink. First of all, a low pass FIR digital filter is designed by using Matlab filter design and analysis tool (fdatool) program. Then, the designed filter is implemented and simulated on Matlab for a given input signal. After that, for three fundamental structures (namely direct-form, transposed direct-form, and symmetric direct-form) in literature, the designed filter is implemented schematically on Quartus-II software and then each project containing different structure implementation is compiled. Then, the digital filters implemented by each structure are simulated by University Program Vector Wave File (VWF) simulation program on Quartus-II software. Simulation results show that the obtained results are the same as the ones obtained on Matlab, which confirms that the schematic designs are successfully implemented. Moreover, the implemented digital filters are realized and successfully tested on Altera DE2-115 FPGA board. Finally, three fundamental FIR structures for various filter lengths from 11 to 51 are implemented to compare them in terms of total logic elements, total registers, and total memory.

Index Terms—Altera DE2-115, FIR filter, FPGA, Matlab, Quartus-II.

I. INTRODUCTION

DIGITAL filters, which are one of the most important elements in Electronics Engineering applications such as communication, control, and biomedical systems, are used to remove or enhance selected frequency range in a digital signal. Digital filters are classified as finite impulse response (FIR) and infinite impulse response (IIR) according to the duration of impulse response [1]. FIR filters are very popular because they can be designed as always stable and having exact linear phase.

FIR Digital filters can be implemented on the embedded systems such as microcontroller [2], digital signal processor (DSP) [3], and field programmable gate array (FPGA) [4]. An FPGA is an integrated circuit in which hardware structure can be changed after production according to the desired function.

O. COŞKUN, is with Department of Electrical and Electronics Engineering, Abant İzzet Baysal University, Bolu, Turkey, (e-mail: oguzhanoskun1608@gmail.com) 

✉ K. AVCI, is with Department of Electrical and Electronics Engineering, Abant İzzet Baysal University, Bolu, Turkey, (e-mail: avci_k@ibu.edu.tr) 

Manuscript received August 4, 2017; accepted Dec. 18, 2017.

DOI: 10.17694/bajece.369234

FPGA technology is used in a wide spectrum area from consumer electronics to aerospace and defense industries [5]. FPGA implementation of FIR filters can be provided by a hardware description language (VHDL or Verilog) or schematic description. And, there are recently proposed studies in literature on the implementation of FIR filters on FPGA [6-9], but they require some licensed software package.

In this study, we investigate the FPGA schematic implementations of FIR digital filters for three fundamental structures on Altera DE2-115 FPGA board without requiring any other software packages such as DSP Builder and Matlab Simulink. In Section 2, we briefly introduce FIR filters with their fundamental structures and then the software and hardware tools used in this study. We then describe our proposed FPGA schematic implementations in Section 3. The simulation and realization results for the proposed implementations are given in Section 4. Finally, conclusion part is presented in the last section.

II. MATERIALS AND METHODS

In this section, we first briefly describe the theoretical backgrounds of FIR filters and their fundamental structures. And then, the software tools and FPGA board used in this study are introduced.

A. FIR Filters

A causal N-length (or order with N-1) FIR digital filter can be characterized by the transfer function, $H(z)$, which is the z-transform of the impulse response of the digital filter

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{n=0}^{N-1} h(n)z^{-n} = \sum_{n=0}^{N-1} b_{n+1}z^{-n} \quad (1)$$

where $Y(z)$ and $X(z)$ are z-transforms of the input and output signals of the digital filter, respectively. And, $h(n)$ and b coefficients represent the impulse response and filter coefficients of the digital filter, respectively. For an FIR filter, the impulse response values and the filter coefficients are always the same whereas they are different for an IIR filter.

An FIR filter can also be characterized by the difference equation which gives the relationship between the input and output signals of the digital filter in discrete time domain

$$y(n) = b_1x(n) + b_2x(n-1) + \dots + b_Nx(n-N+1) \quad (2)$$

where $x(n)$ and $y(n)$ represent the input and output signals of the digital filter, respectively.

As compared to IIR filters, FIR filters have two main advantages which are about the stability and linear phase characteristics. FIR filters are always stable, because the poles of the transfer function are always on the origin of the z -plane which mean that they are inside the unit circle. And, FIR filters can be easily designed as having exact linear phase by providing that $h(n)$ is symmetric or antisymmetric.

In literature, four general methods, namely optimization, windowing, frequency sampling, and numerical methods are used to design FIR filters to satisfy a prescribed characteristic [1]. By using a software tool such as Matlab ©, FIR filters can be easily designed, i.e. the filter coefficients can be found.

B. Fundamental FIR Filter Structures

FIR digital filters can be implemented by three fundamental structures known as direct-form, transposed direct-form, and symmetric direct-form. Besides these, they can be implemented by some other structures such as cascade, parallel, lattice, etc. [1]. In all digital filter implementation types, three circuit elements are used. These are the multiplier, adder, and delay.

The circuit for direct-form FIR structure for N length is shown in Fig.1. It is called direct-form because it directly realizes the convolution process of an FIR filter. It is seen from the figure that total number of the delay elements is equal to the filter order ($N-1$), therefore this structure is a canonical type structure [1]. In this structure, the input signal is first delayed by z^{-1} and then multiplied by b coefficients.

The circuit for transposed direct-form FIR structure for N length is shown in Fig.2. It is called transposed direct-form because as opposed to the direct-form the input signal is first multiplied by b coefficients and then delayed by z^{-1} . This structure is also a canonical type structure.

The circuit for symmetric direct-form FIR structure for N length is shown in Fig.3. The number of the multipliers in this structure is half of the ones in other two fundamental structures. This structure can be used only if the filter coefficients are symmetric.

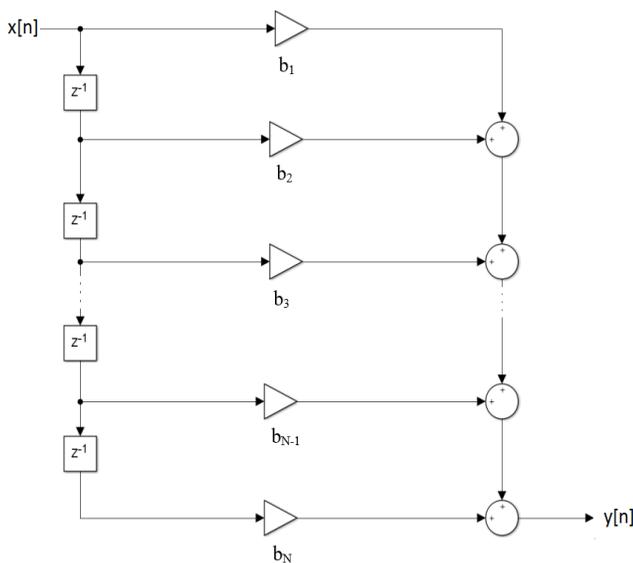


Fig.1. Direct-form FIR structure for N length

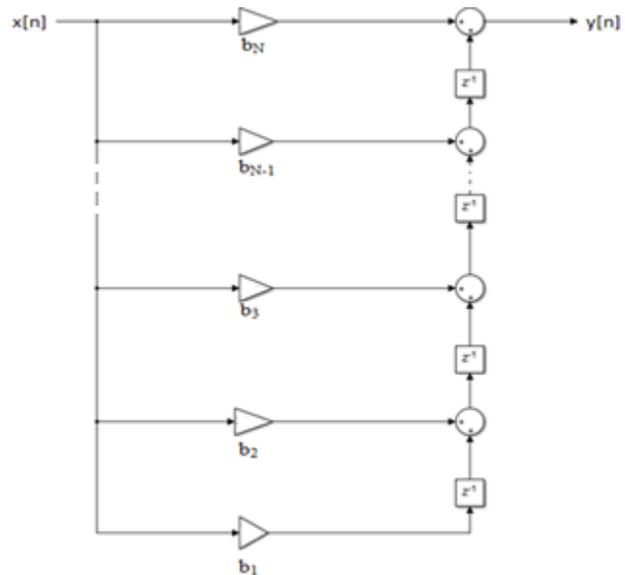


Fig.2. Transposed direct-form FIR structure for N length

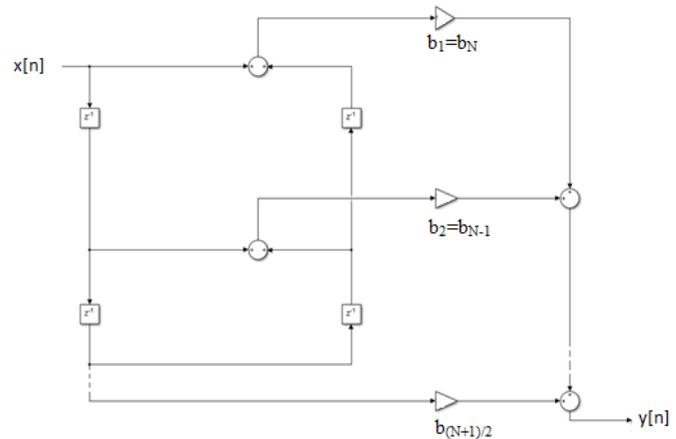


Fig.3. Symmetric direct-form FIR structure for N length

C. Matlab Software

Matlab software produced by the MathWorks Company is one of the most useful software environment for engineers and scientists [10]. In this study, Matlab © 2016a software is used to design and simulate the FIR digital filter to have a reference for the comparison with the FPGA implementation. The digital filters are designed and analyzed in Matlab © software by using a powerful user interface known as fdatool (Filter design and analysis tool).

D. Quartus II Software

The Quartus-II software produced by Altera is a programmable logic device design software, and it enables analysis and synthesis of HDL designs [11]. In this study, Quartus-II is used to implement the digital filters schematically, to compile and simulate the digital filters, and to upload the necessary files to the FPGA board. We use Quartus-II 13.1 64 bit web edition as shown in Fig.4.

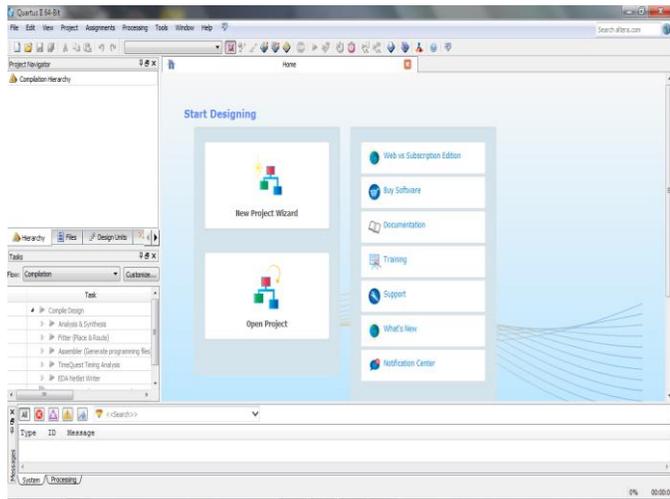


Fig.4. Quartus-II 13.1 web edition

E. Altera DE-2 115 FPGA Board

The DE2-115 board produced by Terasic Company shown in Fig.5 has many features that allow users to implement a wide range of designed circuits, from simple circuits to various multimedia projects [12]. This board is used with Quartus-II software in our study to realize the schematic implemented filter structures.

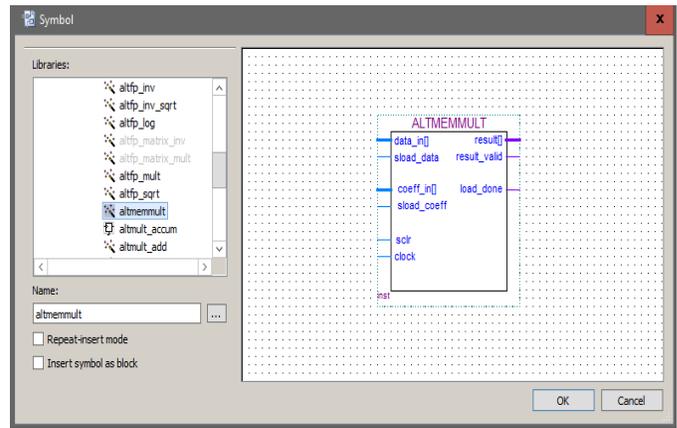


Fig.5. Altera DE2-115 FPGA board

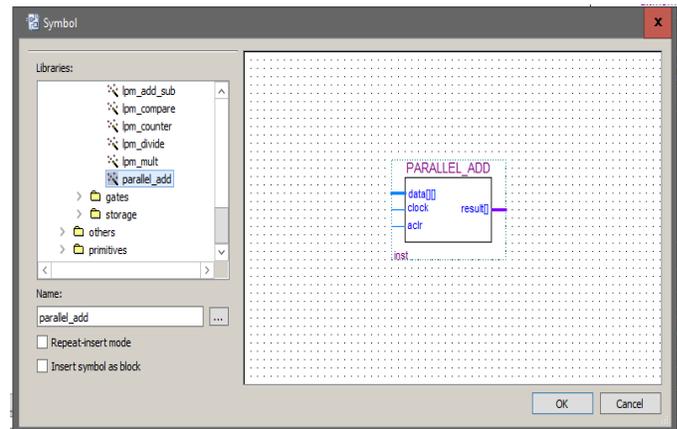
III. SCHEMATIC IMPLEMENTATIONS OF FUNDAMENTAL FIR FILTER STRUCTURES ON QUARTUS II SOFTWARE

In this section, we present the proposed schematic implementations for three fundamental FIR structures.

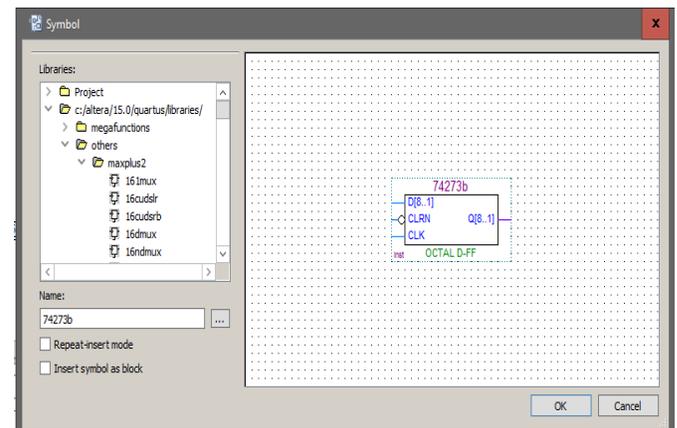
As can be seen from the circuits given by Fig.1, Fig.2, and Fig.3, a digital FIR filter circuit consists of only three digital circuit elements which are multiplier, adder, and delay. Therefore, *altemmult* (Fig.6a), *parallel_add* (Fig.6b), and *74273b D-flip-flop* (Fig.6c) block schemes are chosen from the Quartus II software library for implementing the elements of multiplier, adder, and delay, respectively.



(a)



(b)



(c)

Fig.6. a) *altemmult* for multiplier b) *parallel_add* for adder c) *74273b D-flip-flop* for delay element

Using the block schemes as digital circuit elements shown in Fig.6, an FIR digital filter can be implemented schematically for any filter length. The schematic implementations of three different forms for an FIR filter with the filter length $N=5$ are shown in Fig.7, Fig.8, and Fig.9 for direct, transposed direct, and symmetric direct forms, respectively.

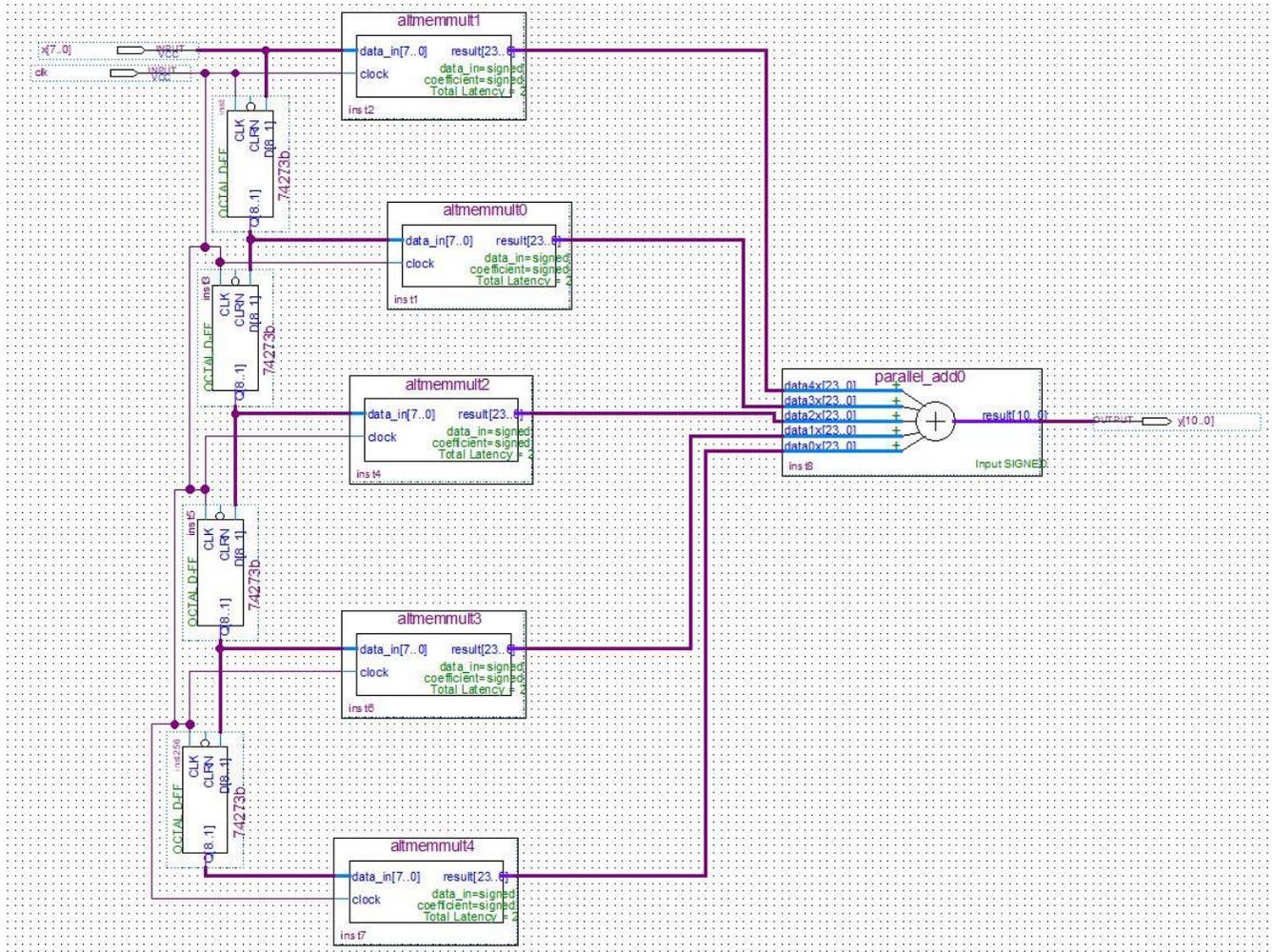


Fig.7. FPGA implementation of direct-form FIR structure for the designed filter with N=5 on Quartus-II

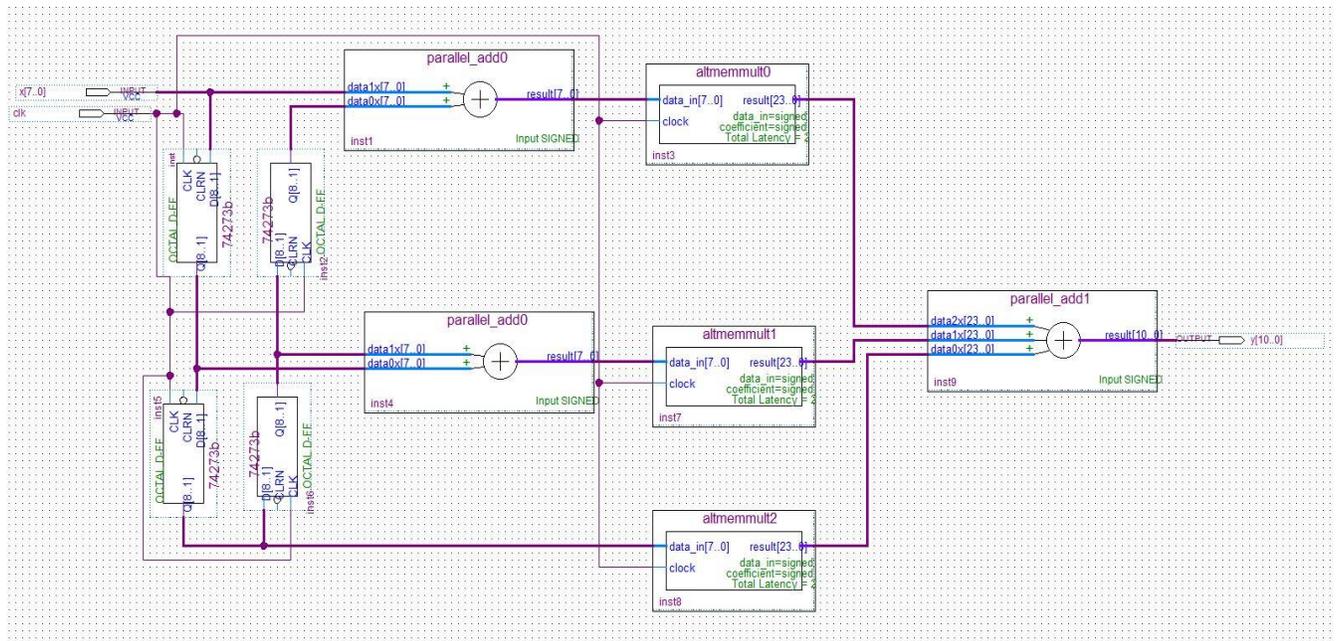


Fig.8. FPGA implementation of transposed direct-form FIR structure for the designed filter with N=5 on Quartus-II

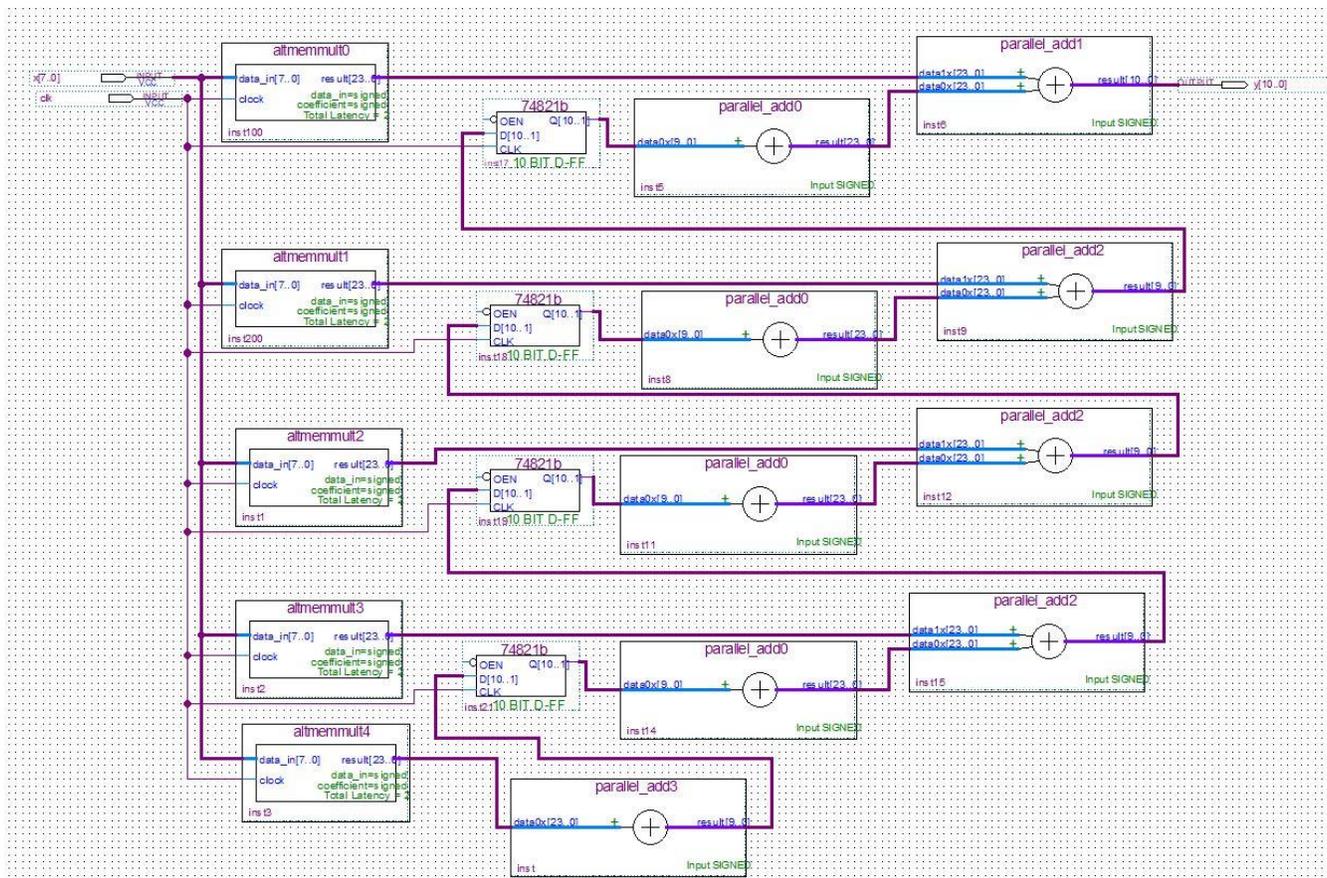


Fig.9. FPGA implementation of symmetric direct-form FIR structure for the designed filter with N=5 on Quartus-II

IV. SIMULATION AND REALIZATION RESULTS

In this section, we present the simulations performed on Matlab and Quartus II environments and then compare with realization performed on the FPGA board.

A. Matlab Simulation Results

To be able to simulate a digital filter and then implement it on an FPGA, we first need to design that filter by using a software program. For this purpose, a lowpass FIR filter based on Kaiser window [13] for beta = 1 is designed by using Matlab fdatool for the filter length N=5 (or filter order = 4), sampling frequency fs=1000 Hz, and cut off frequency fc=100 Hz as seen in Fig. 10.

The five non-integer filter coefficients found from the fdatool are given in Fig.11a. These coefficients are then rounded to be integer as in Fig.11b.

The test signal which will be used as an input to the designed filter for the simulation and realization example is shown in Fig.12a. When this input signal is applied to the designed filter which is also defined by its impulse response in Fig.12b, the output signal is obtained as shown in Fig.12c. It is seen that the steady state filtered value is 1000 in decimal which will be used to evaluate the proposed FPGA implementations of the digital filter.

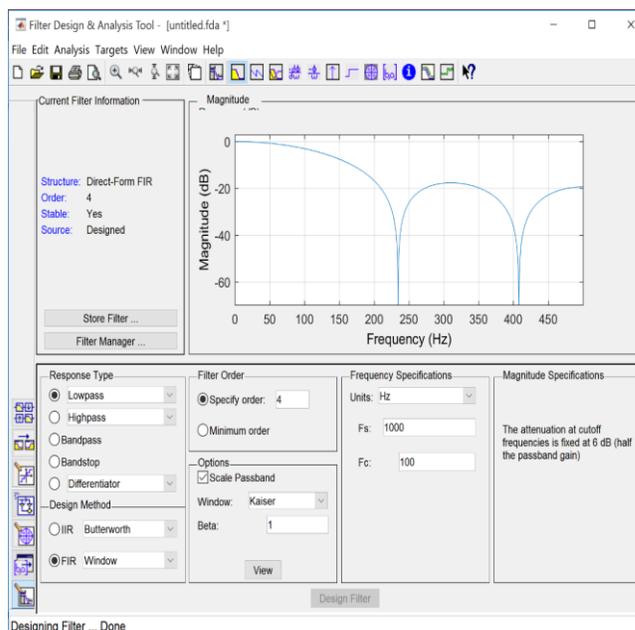


Fig.10. Low pass FIR filter design using Kaiser window [10,11] for window length N=5 (or filter order = 4) and cut off frequency fc=100 Hz

```

-Filter Coefficients-
Numerator:
0.15081345405910787
0.22304157717564707
0.25228993753049012
0.22304157717564707
0.15081345405910787
    
```

(a)

```

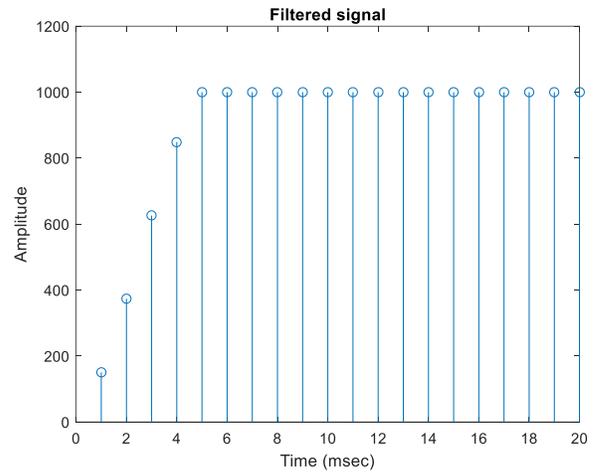
Command Window
New to MATLAB? See resources for Getting Started.
>> b_rounded=round(1000*b)

b_rounded =

    151    223    252    223    151
    
```

(b)

Fig.11. a) Non-integer filter coefficients b) rounded filter coefficients to be integer for the designed filter



(c)

Fig.12. a) input signal, b) impulse response, and c) filtered signal for simulation example of the designed filter for fs=1000 Hz

B. Quartus-II Simulation Results

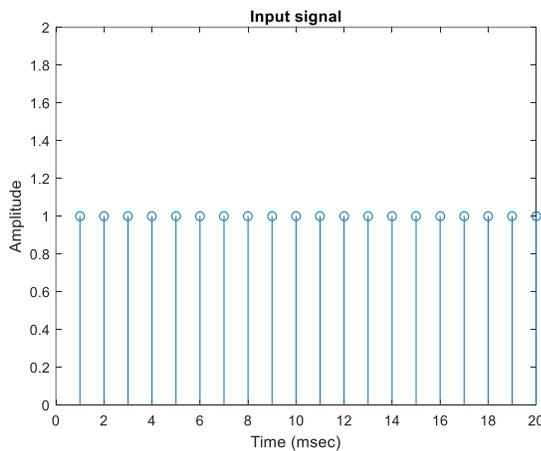
First of all, a Quartus-II project including the schematic implemented direct-form FIR structure given Fig.7 is prepared and then compiled. After that, it is simulated by using *University Program Waveform* in Quartus II. The obtained simulation result is shown in Fig.13.

It is seen from the figure that the output signal has transient values of 0, 151, 374, 626, and 849 and also has a steady state value of 1000. These results are exactly the same as the ones in Matlab simulation given by Fig.12c. Two more Quartus- II projects for the schematic implemented transposed direct and symmetric direct forms of FIR structures given in Fig.8 and Fig.9 are also prepared and then compiled and simulated as in case for the direct-form. It is observed that the simulation results for the transposed direct and symmetric direct forms are also the same as the one given in Fig.13. These simulation results demonstrate that our proposed FPGA schematic implementations for FIR structures given by Fig.7, Fig.8, and Fig.9 are proper implementations.

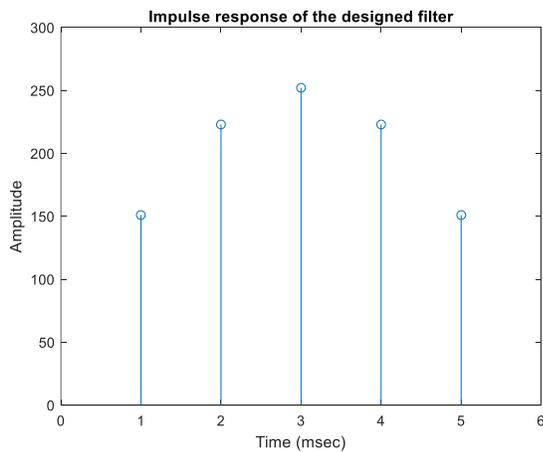
C. Realization on DE-2 115 FPGA Board

To test the schematically implemented FIR structure on FPGA board, first the pins for the input x and output y signals are assigned as shown in Fig.14 by using Assignment Editor in Quartus II. This process results in the switches from SW0 to SW8 on the board to be the input and clock signals, and also in the leds from LEDR0 to LEDR10 to be the output signal.

After making pin assignments, the project for direct-form is compiled again for realization test. Then, by using the programmer interface in Quartus II, the related .sof extended file is loaded to the FPGA board as shown in Fig 15.a. If the loading progress is successful as in Fig 15.b, the realization test can be performed.



(a)



(b)

D. Comparison of Schematic Implementations of FIR Digital Filter Structures for Various Filter Length

In the previous two sections, the simulations and realizations of three different types of the schematic implemented FIR structures for the filter length $N=5$ are successfully carried out. In this section, we perform the same procedures for various filter length ($N=11, 21, 31, 41, \text{ and } 51$) to compare the fundamental structures in terms of the total logic, the total registers, and the total memory bits which are obtained from the compiling all the structures on Quartus-II. Since the circuits for implemented structures become more complex and larger in size for the larger filter length, the circuits are not given here. But we observed that all circuits are successfully simulated and realized separately.

Table I shows the total logic elements used for three fundamental FIR structures for various filter lengths. It is seen that the least total logic elements are obtained for transposed direct-form structure. Also, it can be seen that the number of total logic elements used for all three structures increases as the filter length increases.

TABLE I
DATA FOR THE COMPARISON OF THE STRUCTURES IN TERMS OF THE TOTAL LOGIC FOR VARIOUS FILTER LENGTHS

Structures	N=11	N=21	N=31	N=41	N=51
Direct	150	281	360	439	521
Transposed	80	168	248	328	411
Symmetric	124	242	367	492	608

Table II shows the total register elements used for three fundamental FIR structures for various filter lengths. It is seen that the least total register elements are obtained for direct structure. Also, it can be seen that the number of total register elements used for all three structures increases as the filter length increases.

TABLE II
DATA FOR THE COMPARISON OF THE STRUCTURES IN TERMS OF THE TOTAL REGISTERS FOR VARIOUS FILTER LENGTHS

Structures	N=11	N=21	N=31	N=41	N=51
Direct	80	120	120	120	120
Transposed	80	160	240	320	400
Symmetric	80	160	240	320	400

Table III shows the total memory bits used for three fundamental FIR structures for various filter lengths. It is seen that the least total memory bits are obtained for symmetric direct-form structure for lower filter lengths and for direct structure for higher filter lengths. Also, it can be seen that the number of total memory bits used for all three structures increases as the filter length increases.

TABLE III
DATA FOR THE COMPARISON OF THE STRUCTURES IN TERMS OF THE TOTAL MEMORY BITS FOR VARIOUS FILTER LENGTHS

Structures	N=11	N=21	N=31	N=41	N=51
Direct	22528	32848	33008	33168	33328
Transposed	22528	43008	63488	83968	104448
Symmetric	12288	22528	32768	43008	53248

V. CONCLUSION

In this study, FPGA schematic implementations for three different fundamental structures of FIR digital filters are investigated. To have a reference result for the comparisons, we first provide a simulation example on Matlab for the filter length $N=5$ for an input signal. Then, three FIR structures are schematically implemented on Quartus-II for $N=5$.

After that, three FIR structures for $N=5$ are simulated using the universal waveform tool in Quartus-II. It is shown that the simulation results of Quartus-II are the same as those of Matlab, which means our proposed schematic implementations are proper successful implementations.

Then, our schematic implementations are tested on Altera DE2-115 FPGA board. The realization results also demonstrate that our proposed implementations are successful.

Moreover, we implement three fundamental FIR structures for various filter lengths from 11 to 51 to compare them in terms of total logic elements, total registers, and total memory by compiling them on Quartus-II. We conclude that the least total logic elements and registers are obtained for transposed direct-form and direct-form structures, respectively. As for the least total memory bits, the best results are obtained for symmetric direct-form structure for lower filter lengths and for direct structure for higher filter lengths.

ACKNOWLEDGMENT

This study is supported by the Scientific Research Projects Unit of Abant Izzet Baysal University with the project number 2016.09.03.1009.

Also, this study is partly presented in ICENS 2016 and published as an abstract [14].

REFERENCES

- [1] A. Antoniou, *Digital Signal Processing: Signal, Systems, and Filters*. New York: McGraw-hill, 2005.
- [2] B. K. A. Ramu, "Implementing FIR and IIR Digital Filters Using PIC18 Microcontrollers", Microchip Application Notes, pp.1-46, 2002.
- [3] C. L. Hu, "Design and Verification of FIR Filter Based on Matlab and DSP", International Conference on Image Analysis and Signal Processing (IASP), 9-11 Nov., Hangzhou, China, pp. 286-289, 2012.
- [4] I. Thingom, P. Khundrakpam, "FPGA implementation of FIR filter using RADIX-2", International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 23-25 March, Chennai, India, pp. 1524-1528, 2016.
- [5] S. Karataş, E. Saritas, *Her yönüyle FPGA ve VHDL*. Palme Yayıncılık, 2003.
- [6] V. H. Deshmukh, A. Mishra, A.S. Bhalchandra, "FIR Filter Design on Chip Using VHDL" IPASJ International Journal of Computer Science. Vol. 2, No. 7, July, pp 12-16, 2014.
- [7] R. M. Narsale, D. Gawali, A. Kulkarni, "FPGA Based Desing & Implementation of Low Power FIR Filter for ECG Signal Processing" IJSETR International Journal of Science, Engineering and Technology Research. Vol.3, No. 6, June, pp 1673-1677, 2014.
- [8] K. Aboutabikh, I. Haidar, A. Garib, "Design and Implementation of a Digital FIR LFP with Variable Pass-Band for ECG Signal using FPGA" International Journal of Advanced Research in Computer and Communication Engineering. Vol.4, No. 9, September, pp 275-281, 2015.
- [9] K. Aboutabikh, N. Aboukerdah, "Design and Implementation of a Multiband Digital Filter Using FPGA to Extract the ECG Signal in the Presence of Different Interference Signals" Computers in Biology and Medicine, Vol. 62, pp.1-13, 2015.
- [10] https://www.mathworks.com/products.html?s_tid=gn_ps
- [11] <https://www.altera.com/downloads/download-center.html>
- [12] www.terasic.com.tw/_sub/de2-115

- [13] J. F. Kaiser, Nonrecursive digital filter design using I_0 -sinh window function. *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS'74)*, San Francisco, Calif, USA, April, 20-23, 1974.
- [14] O. Coşkun, K. Avci, "Schematic Implementations of FIR Digital Filters on Altera FPGA", 2016 Book of Abstracts: 2nd International Conference on Engineering and Natural Sciences (ICENS 2016), 24-28 May, Srajevo, Bosnia and Herzegovina, pp. 664, 2016, (Abstract)

BIOGRAPHIES



OĞUZHAN COŞKUN was born in Bursa, Turkey, in 1995. He received the B.S. degree in electrical and electronics engineering from Abant İzzet Baysal University, Turkey in 2016. Currently, he is a graduate student at the same department since February 2017. His current research interests are digital filters and FPGA implementation.



KEMAL AVCI was born in Adiyaman, Turkey, in 1980. He received the B.S., M.S., and Ph.D. degrees in electrical and electronics engineering from the University of Gaziantep, Turkey in 2002, 2004, and 2008, respectively. Also, from 2003 to 2008, he worked as a research assistant at the same department. Currently, he works as an assistant professor since 2008 in electrical and electronics engineering in Abant İzzet Baysal University, Turkey. His main research interests are design and implementation of digital filters and filter banks, two dimensional digital filters, audio & music signal processing, and switched capacitor networks.