FPGA Implementation of Efficient FIR Filter

Ranjeeta Yadav, Rohit Tripathi, Sachin Yadav

Abstract: ***Improve the functionality of an FIR Filter by modifying the internal components used to design a filter. These past years have seen some great improvements in the speed, power, and area of the filter. Here, we will, therefore, use an ALU-based algorithm to design our FIR filter. The internal components of the ALU block will be an Adder and a Multiplier. A Floating point Adder and a Floating Point Multiplier will be the basic backbone of the ALU block, which finally will be used to design and implement our FIR filter design. Therefore, the parameters of the area are our main target but we will also see the power consumed by the Filter operation, both static and dynamic power consumed will be seen. The programming language will be written in VERILOG and the simulation and implementation of the design will be done by the help of Xilinx ISE suite version. One important aspect is that there will be 16 input samples and 16 coefficients which will be directly from a 16 tap filter. These coefficients and input values will be generated through MATLAB software.***

Keywords: IEEE, FIR, Multiplier, ALU, Xilinx

I. INTRODUCTION

Filter plays essential role in Digital Signal Processing (DSP). Filters a system that passes certain frequency components and rejects other frequency components. Filters are designed for the specifications of the desired properties of the system. FPGA is a prototype device which is used to implement simpler algorithm.

The undesirable signals or the noise of the signal can be removed by the filter. There are two types of digital filters. 1) FIR (Finite Impulse Response) filter and 2) IIR (Infinite Impulse Response) filter. Both FIR and IIR filters have their own specific areas of working; this completely depends on its characteristics. In most cases, the FIR filters have been used on a larger scale due to its finite character of its inputs being monitored. A finite impulse response (FIR) filter is a filter structure that can be used to implement almost any sort of frequency response digitally. An FIR filter implemented by multipliers, delays and adders to create the filter's output.

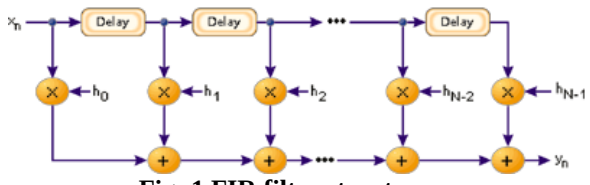

Fig.-1 FIR filter structure

Linear-phase filters delay the input signal but don't distort its phase. They are simple to implement. FIR filter can easily design for "linear phase". They are suited to multi-rate applications. By multi-rate, we mean either "decimation" (reducing the sampling rate), "interpolation" (increasing the sampling rate), or both. Whether decimating or interpolating, the use of FIR filters allows some of the calculations to be omitted, thus providing an important computational efficiency. In contrast, if IIR filters are used, each output must be individually calculated, even if it that output will be discarded (so the feedback will be incorporated into the filter). They have desirable numeric properties. In practice, all DSP filters must be implemented using finite-precision arithmetic, that is, a limited number of bits. The use of finite-precision arithmetic in IIR filters can cause significant problems due to the use of feedback, but FIR filters without feedback can usually be implemented using fewer bits, and the designer has fewer practical problems to solve related to non-ideal arithmetic. They can be implemented using fractional arithmetic. Unlike IIR filters, it is always possible to implement an FIR filter using coefficients with the magnitude of less than 1.0. (The overall gain of the FIR filter can be adjusted at its output, if desired.) This is an important consideration when using fixed-point DSP's because it makes the implementation much simpler.

For an N-tap FIR filter with coefficients h (k), which's output is described by:

y(n)=h(0) x(n) + h(1) x(n-1) + h(2) x(n-2) + ... h(N-1) x(n-N-1),

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n}$$

II. PROPOSED WORK

In this paper use the concept of the ALU and break it down into a much simpler ALU BASED FIR FILTER design by directly using only two types of components to perform multiple operations repeatedly for different operands. We saw the possibility of using a much more efficient form of Adder. i.e. Floating point Adder and a Floating point Multiplier to form the ALU logic and then design the Filter. Most of the time the designing was done using MATLAB but in this approach, we decided to use VERILOG, as it is much simpler to grasp and almost anyone with the prior knowledge of C programming language can work with it. Here is a universally acceptable IEEE standard for representing floating-point numbers for the computing which is the IEEE 754 standard. 32-bit floating point representation is used in this approach for the designing of the ALU based filter. Firstly, 16 tap filter; generate its filter coefficients through MATLAB. The value of the input is defined by "x". The coefficients will be defined by "h" as h(0), h(1), h(2), h(3) and so on.

Revised Manuscript Received on February 26, 2020.

Ranjeeta Yadav, ABES Engineering College,Ghaziabad, INDIA,Email-ranjeeta29@gmail.com

Dr. Rohit Tripathi, Galgotias University, Greater Noida, INDIA, Email-rohit.tripathi@galgotiasuniversity.edu.in

Dr. Sachin Yadav, GL Bajaj Institute of Technology & Managment, Greater Noida, INDIA, Email- sac.yaduvanshi@gmail.com

We will make these values work with the equation as much times required, and then we will be able to make the filter run and see the results.

Among all the approaches, the most difficult part was to choose the most suitable and simple yet effective type of multiplier and the adder used in the ALU, which took a bit of time to overcome. Finally, we decided to use Floating point type of Adder and Floating point type Multiplier. Once this was decided, we had to learn how to write programming in HDL which took even more time and effort. After a sufficient time of understanding how to write codes in VERILOG language, we started to write a code for a simple adder and multiplier and simulated them. The basic structure of proposed FIR Filter is shown in Fig.-2. Now, after getting familiar with the Xilinx software, we finally started to write codes for Floating point adders and Multipliers and then simulated them in this Software.

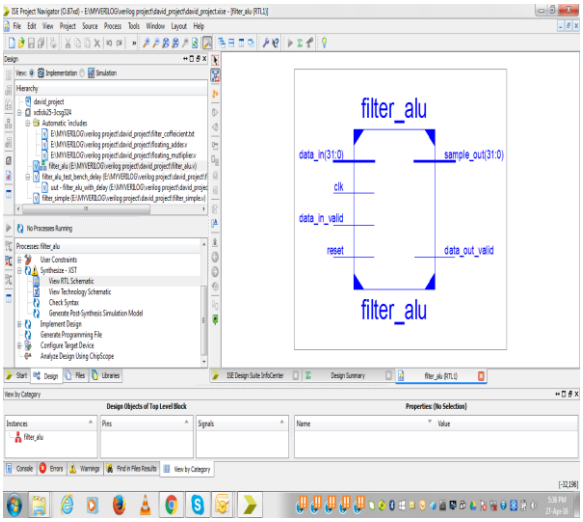

Fig.2 ALU filter RTL Schematic

After successful simulation of these few blocks, we finally wrote the coding/program for the Floating type FIR Filter using the previously used Floating type Adder and Multiplier, but both were called as a function and so did not need to be instantiated again in the final Filter program. Hence, it was made much easier to simulate. We simulated the program and then we took out the RTL view by synthesis of the codes/program and then we implemented the design further which was slower than the synthesis process. Then we were able to see the internal structure of our Filter design. In the end, we were concerned about the power being consumed by our filter during its operation. Therefore, we even found its dynamic and static power value.

III. RESULTS AND OUTCOMES

With proper clock input, we were able to get the output waveform, which could be finally be seen in the output wave sample by further zooming the view of the image. The output values were in hexadecimal form. The values can be changed by changing the Radix in the command options of the software. The values were fixed depending on the 16 tap filter coefficients and input values, as they were also fixed. Our main focus was to try and decrease the Filter area which has been done by using the same adder and multiplier again and again for a different number of times for each different operation Hence, decreasing the area. Although due to the parallel nature of the operation, there was a little delay found, we will not go deep into it in this particular thesis of ours. Input and output waveforms are shown in fig.3 and output values are shown in fig.4.

The input size will be 32 bits in this design, i.e. Single Precision. As we know that the waveform will be in the form of an S-curve, half appearing in the x-axis and half appearing in the y-axis, and changing from a certain point to a negative value but the magnitude will always be the same.

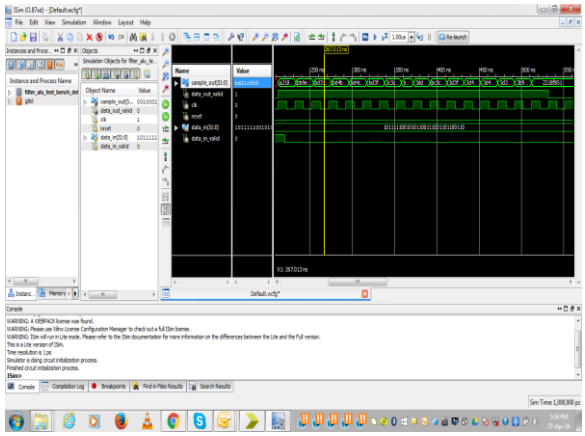

Fig.-3 Input and Output sample waveform

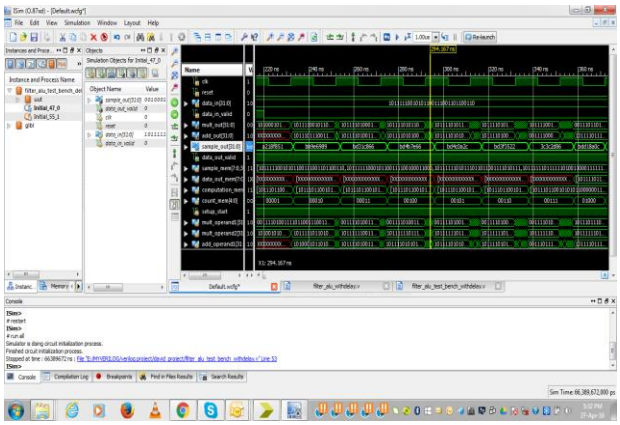

Fig.-4 Zoomed in view of the sample output values

There will be a formation of mirror image after the 8th sample, the 9th will be the same magnitude with the 8th and so on, but the sign will be different. So we can see that the result of the equation of the FIR Filter will run till the 8th cycle in a normal way which is easy, but when it reaches the ninth cycle, the computation will change. The first memory location of the Comparison memory in the VERILOG program of the filter will be re-used by overwriting the new values. The adders will keep adding the values and the multiplier will multiply the values for the next greater sample rate and this will continue until 16 total cycles as we have used only 16 tap Filter.

Only the Area was decreased in this work but at the end we also found out the total power consumed to run the FIR Filter, which is the static and the dynamic power combined together. Therefore, the total power used was found to be 0.047 in this analysis. The area consumption was decreased by approximately 16.4 %. The power used by FIR filter is shown in Fig.-5.

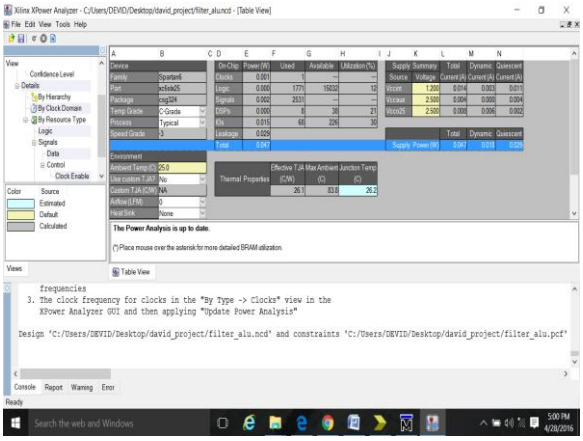

Fig.-5 Power used by the FIR filter

IV. CONCLUDING REMARKS

In this paper, to improve the quality of the inner processes involved in the work. Similarly, in the field of Very Large Scale Integration (VLSI), various new steps have been developed within these few passing years which have made it possible for the various engineers to improve the quality of the products by improving the inner components. And talking about FIR Filters, they are the building blocks of the Digital world.

It is now been seen that the FIR Filter designed in this work has a less Area consumption and that with the help of two simple types of adder and multiplier, how we can change the characteristics so rapidly, We were also able to learn about the various programming languages, even though not completely every one of them but mainly the VERILOG and the MATLAB, though the MATLAB was used less the VERILOG played an important role in this work So we can see that by just knowing how to write the programming in HDLs such as VERILOG, how useful it can be to use it in designing various digital circuits. If there is more time given to learn this programming language, a lot of further studies can be done with much ease. It is essential to know this as it is vital in the world of the fast growing Digital world of improving Technologies and Devices

We also came to know that Xilinx ISE suite is a very user-friendly platform to work on. Although the processes involved in it are very large, it is still very much effective and simpler for working on for filter designing, Implementation, and it's Synthesis.

FUTURE SCOPE

This work is just the foundation laid down as this area of work is limitless in nature. There are many ways we can work on it further. We can work on its power consumption which can be tried to decrease more by using various efficient techniques. We can work on its speed also in the future and even try to work with 32 taps, 64 tap Filters which will be much tunable in its parametric efficiency. Hence, it can be improved much further in its Area reduction, Power consumption and increasing speed. As long as there is advancement in the field of science and technology, there will always be a need to develop more and more simple and easy to execute processes and to finish the work faster and efficiently so that the progress has to be rapid and more easily obtainable. Hence, this field of work is always going to move forward and never stop growing in its future progress.

REFERENCES

1. Hardware Implementation of Parallel FIR Filter Using Modified Distributed Arithmetic by Gourishankar Das , Krishanu Maity , Suman Sau 2018 2nd IEEE International Conference on Data Science and Business Analytics (ICDSBA) Pages 643 - 646
2. Efficient FPGA Implementation Architecture of Fast FIR Algorithm Using Han-Carlson Adder Based Vedic Multiplier by Payal Paliwal , Janki Ballabh Sharma in International Conference on Inventive Research in Computing Applications (ICIRCA 2018) IEEE Xplore ISBN:978-1-5386-2456-2
3. Design and Implementation of FPGA Based 32 Bit Floating Point Processor for DSP Application by Anand Burud , Pradip Bhaskar in 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) Pages: 1 - 5
4. HDL implementation of digital filters using floating point vedic multiplier by Prashant S. Howal , Kishor P. Upla , Mehul C. Patel in 2017 IEEE International Conference on Circuits and Systems (ICCS) Pages: 274 - 279
5. Comparative study of 16-order FIR filter design using different multiplication techniques by Anubhuti Mittal , Ashutosh Nandi, Disha Yadav in IET Circuits Devices Syst., 2017, Vol. 11 Iss. 3, pp. 196-200
6. A. Chandra, S. Chattopadhyay, "Design of hardware efficient FIR filter: A review of the state of the art approaches", international journal, Elsevier, 2015
7. B. Somanathan Nair, "Digital Signal Processing: Theory, Analysis and Digital-filter Design", Prentice-Hall of India, New Delhi, 2004.
8. J. G. Proakis, "Digital Signal Processing: Principles, Algorithms and Applications", Prentice Hall of India, New Delhi, 1997.
9. M. Jhamb, Garima, and H. Lohani, "Design, implementation andperformance comparison of multiplier topologies in power-delayspace,"Engineering Science and Technology, an International Journal,vol. 19, no. 1, pp. 355 – 363, 2016.

AUTHORS PROFILE

Ms. Ranjeeta Yadav, has 15 years of teaching experience in the field of Electronics and Communication Engineering. She has a master's degree in Signal Processing and pursuing Ph.D. in VLSI design. Her area of interest is Microcontroller, Embedded and VLSI Design. She has presented and published many research articles in various international journals and Conferences.

Dr. Rohit Tripathi, Dr. Tripathi have earned his PhD from IIT Delhi in 2017. He has extensive experience in research and development. He is editorial board member of two reputed international journals, USA and UK. He has visited 12 countries on his research assignments in last 5 years with multiple funding of govt. agencies. He has published 8 research papers in SCI indexed international journals and conferences within India or abroad whereas 24 Scopus index research papers. His area is IoT based wireless system, VLSI, IDMA and photovoltaic application systems. He has 15 years of teaching and research experience. He has won number of research and teaching excellence awards from different organization in his career.

Dr. SachinYadav, Doctorate in Computer Science & Engineering, has 20 years of teaching experience in the field Computer Science Engineering. He is an active researcher having hands on experience in NS-2. Dr. Yadav has presented and published more than 20 research papers in various international conferences and journals. Dr. Yadav is an active member of Computer Society of India. He acts as a member of Technical Program Committee in various international conferences. His research area includes Theory of Computing, Overlay Networks & Advance network architecture. He has guided several B. Tech and M.Tech students for projects and dissertations. Currently, one Ph.D scholar got her Ph.D and one Ph.D Scholar is pursuing under AKTU, Lucknow, INDIA under his supervision.