DA Algorithm Based Reconfigurable 16-Tap FIR Filter Design Analysis

Agamreet Kaur[1], Rajesh Mehra[2]
[1]ME scholar, [2] Associate Professor
National Institute of Technical Teachers' Training & Research, Chandigarh, India

Abstract

In this paper, 16-Tap FIRlow pass filter has been designed and implemented on FPGA target device. The filter has been designed and analyzed using different folding factors in order to optimize speed and area parameters.A multiplier less Distributed Arithmetic algorithm(DALUT) is used to provide optimized cost effective reconfigurable FIR filter. The proposed filter has been designed and simulated usingMATLAB. Its behavioral simulation and synthesis are performed using ISE simulator and Xilinx Synthesis tool on sparten-3E based on 3s500efg320-5 FPGA device. The synthesis results show area consumption from
And maximum operating frequency from42.093 MHz to 60.680 MHzwith increase in folding factor from 2 to 8.

Keywords - *DALUT, Digital filter,FPGA, VLSI.*

I. INTRODUCTION

Digital signal processing has a broad area of applications in newly evolving technologies. This is due to advancement in VLSI (Very Large Scale Integration). FIR filters realization with high speed and low power consumption are in huge demand. Number of coefficients used to design a filter and its accuracy both involve trade-off. Higher the number of coefficients in complex is the design and more area it consumes. On the other hand, lesser number of coefficients representing a designed filter leads to much lower accuracy and add to ripples.[1] An impulse response of an FIR filter may be expressed as:

$$Y = \sum_{k=1}^{K} C_k x_k \qquad (1)$$

where $C_1, C_2 \ldots\ldots C_K$ are fixed coefficients and $x_1, x_2 \ldots\ldots\ldots x_K$represent the input data words. This method is computationally expensive for hardware implementation as it requires K multiply-and-accumulate (MAC) operations, is logically complex, incorporate more area usage and gives relatively lower throughput. These problems are solved using DALUT Algorithm where the multiply and accumulate operations are replaced by a number of LUTs (look up tables) and adders or summations. [2]

Folding factor plays an important role in filter design. It is transformation technique widely used in digital signal processing to reduce the number of functional blocks in implementing DSP structures. Suppose a functional block performs one unit time processing initially, then after applying folding factor of N, same block is becomes capable ofN unit time processing. It reduces the functional units in the design making it simpler and easy to design. It makes our design cost effective too.

II. DISTRIBUTED ARITHMETIC ALGORITHM

Distributed Arithmetic (DA) is a widely used method for implementing sum-of-products computations without the use of multipliers. The main advantage of DA is its high computational efficiency.[3] DA distributes multiply and accumulate operations across shifters, lookup tables (LUTs), and adders in such a way that conventional multipliers are not required.

We may express each $x_k$as

$$x_k = -b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \qquad (2)$$

where the $b_{kn}$are the bits, 0 or 1, $b_{k0}$is the sign bit.

Now combining equation (1) and (2) in order to express *y* in terms of the bits of x_k; we see

$$Y = \sum_{k=1}^{K} C_k [-b_k + \sum_{n=1}^{N-1} b_{kn} 2^{-n}] \qquad (3)$$

Equation (3) is used to express the inner product.
Interchanging the order of the summations yields:

$$Y = \sum_{n=1}^{N-1} [\sum_{k=1}^{K} C_k b_{kn}] 2^{-n} + \sum_{k=1}^{K} c_k (-b_{k0}) \qquad (4)$$

Where K is number of taps (Inputs) and N is Wordlength of data. It can be stored in look up table (ROM) of size 2^*2^K.

Equiripple filter technique is more efficient as it meets desired specifications with least number of coefficients. Equiripple filter helps in minimizing the maximum ripples in pass band and stop band which is desired in order to design a stable filter [4].

FPGAs have been trending in recent times in consumer market due to their reconfigurable and flexible nature in contrast to ASICs. Millions of transistors are prefabricated on FPGAs which can be customized to obtain desired logic by the user. CLBs in FPGAs are usually comprised of lookup tables(LUTs) and MUXs along with input output lines.[5] FPGAs allow global interconnects which allow the signals to cross the chip without being processed through local switching elements, thus saves time. FPGAs are sometimes hybrid with ASICs leading to three kind of structures viz. fine grained, medium grained and large grained.

DA used in proposed design is fully serial, i.e. process 1 bit at a time.FIR filters, both symmetric and asymmetric, require N+ 1 clock cycles to generate outputand behave as an exception. This is because one extra clock cycle drives the execution and processing of carry bit of preadder. [6].

III. PROPOSED FIR FILTER DESIGN

MATLAB software is used to design FIR Low Pass Equiripple filter of order 15 and density factor 20.

. **Table I. Filter design parameters**

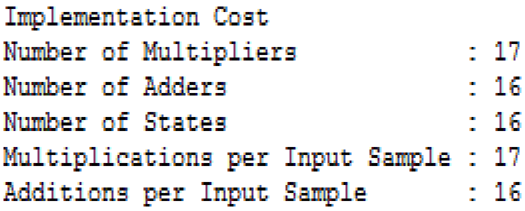
Sampling frequency(F_s)	48000 Hz
$F_{pass(Hz)}$	9600
$F_{stop(Hz)}$	12000
W_{pass}	1
W_{stop}	1

Table I shows designed sampling, pass band and stop band frequencies. Filter weights are kept as 1 in both pass and stop band

Table II. Quantization Parameters

Coefficients	Filter Arithmetic: Fixed point Numerator length:16
Input/ Output	Input word length: 16 Input fraction length: 14 Output word length: 16 Output fraction length: 14
Filter Internals	Rounding mode: nearest (convergent) Overflow mode: Wrap Product word length: 32 Product fraction length: 30 Accumulator word length: 32 Accumulator fraction length: 30

Table II refers to the quantization parameters involved in filter design. Numerator Length of coefficients can be varied. Fixed point filter arithmetic has been used in purposed design. Input and Output word lengths are specified along with their fraction lengths. Fraction lengths are specified to accommodate decimals. In filter internals, rounding mode of filter results is specified s nearest (convergent). Accumulator has word length and fraction length as provided in table. Overflow mode is given as Wrap. [7]



```
Implementation Cost
Number of Multipliers            : 17
Number of Adders                 : 16
Number of States                 : 16
Multiplications per Input Sample : 17
Additions per Input Sample       : 16
```

Figure1. Filter specifications

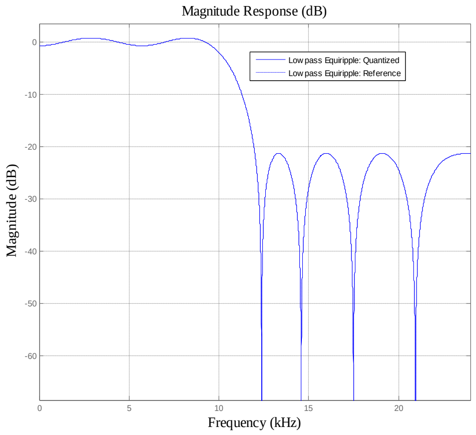


Figure2. Magnitude response

Figure1 depicts implementation cost of designed low pass FIR equiripple filter which incorporates number of multipliers, adders, states, additions and multiplications performed per input sample. Figure2 shows filter magnitude response.[8]

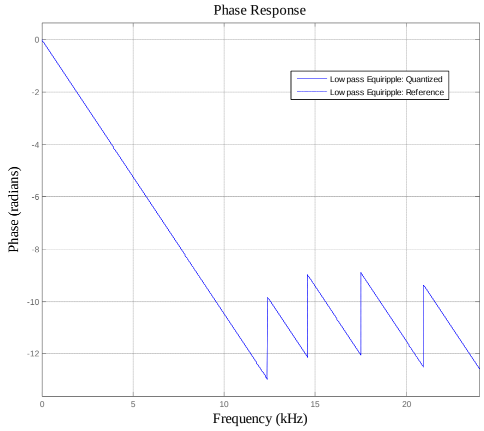


Figure3. Phase response

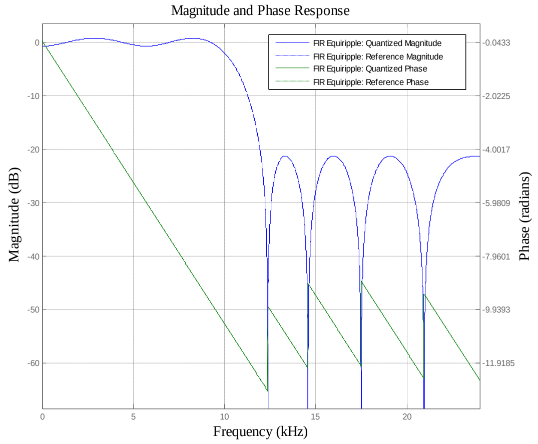


Figure4. Magnitude and Phase Responses.

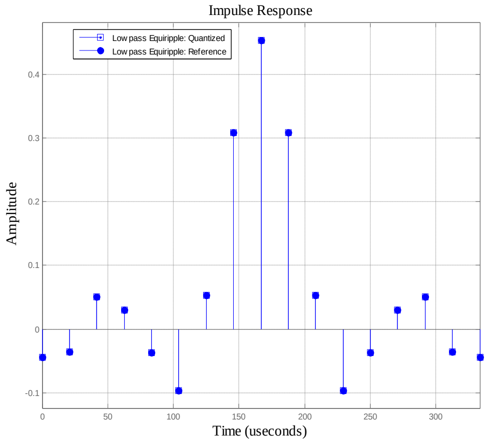


Figure5. Impulse Response

Figure3 shows phase response of designed filter and figure 4 depictscombined magnitude and phase responses of low pass referenceand quantized equiripple filter.Figure 5 describes impulse response of purposed FIR filter. It is observed that reference and quantized FIR low pass equiripple filters impulse responses coincide with each other.

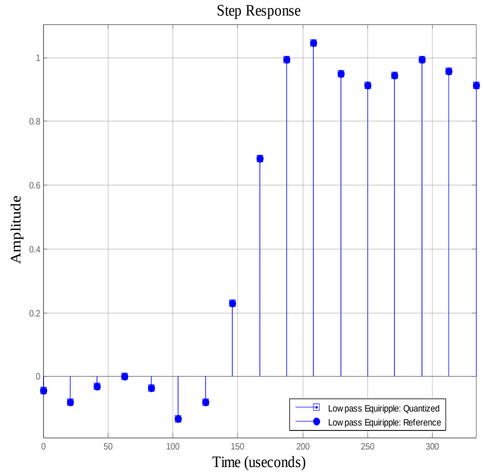


Figure 6. Step Response

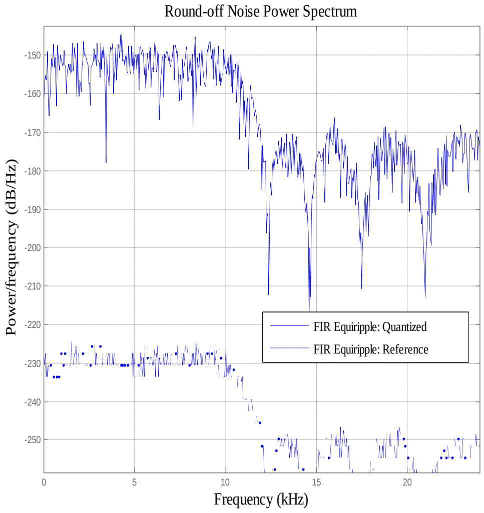


Figure 7. Round-off Noise Power Spectrum

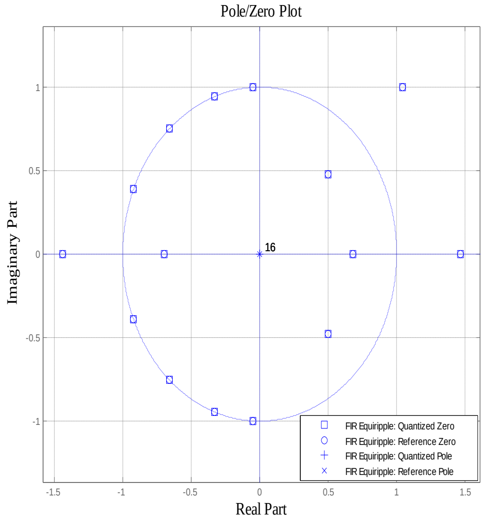


Figure 8. Pole-zero plot

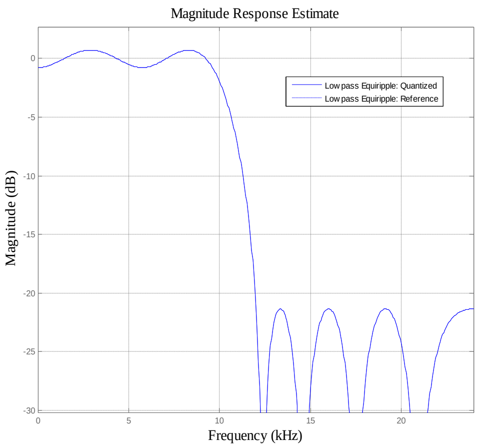
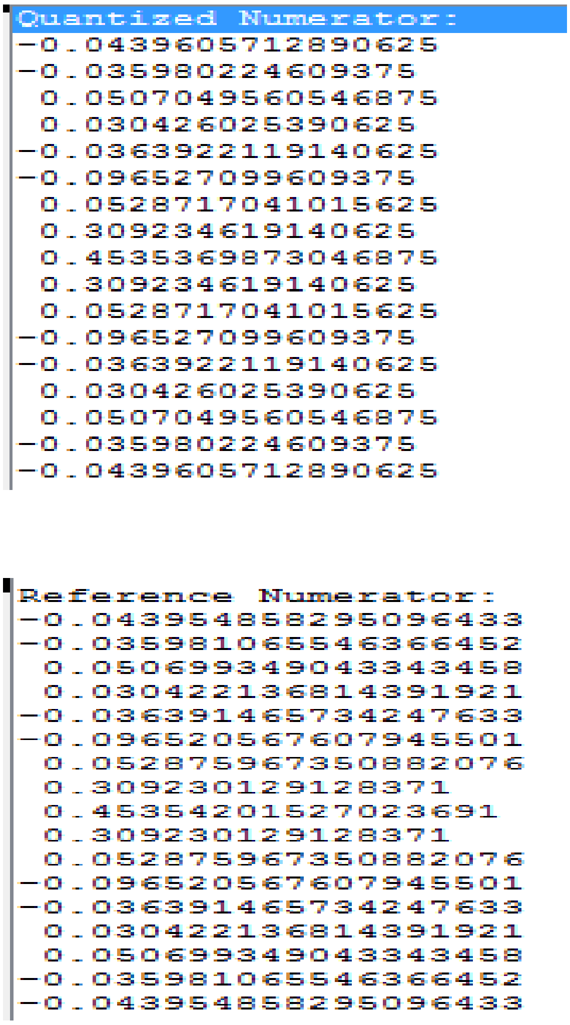


Figure9. Magnitude Response Estimate

Figure 6 depicts step response of a filter i.e. response when input is a step signal.. Figure 7 depicts round off power noise spectrum.Figure 8 represents the pole zero plot of designed equiripple filter. Pole zero plots are used to evaluate frequency response or conversly, frequency response can be used to generate pole-zero plot. Figure 9shows magnitude response estimate of purposed FIR filter design.[9]



```
Quantized Numerator:
-0.0439605712890625
-0.035980224609375
 0.0507049560546875
 0.030426025390625
-0.0363922119140625
-0.096527099609375
 0.0528717041015625
 0.309234619140625
 0.4535369873046875
 0.309234619140625
 0.0528717041015625
-0.096527099609375
-0.0363922119140625
 0.030426025390625
 0.0507049560546875
-0.035980224609375
-0.0439605712890625
```

```
Reference Numerator:
-0.0439548582950964433
-0.0359810655463666452
 0.0506993490433434458
 0.0304221368143919921
-0.0363914657342476333
-0.0965205676079455011
 0.0528759673508820076
 0.309230129128371
 0.45354201527023691
 0.309230129128371
 0.0528759673508820076
-0.0965205676079455011
-0.0363914657342476333
 0.0304221368143919921
 0.0506993490433434458
-0.0359810655463666452
-0.0439548582950964433
```

Figure10. Filter coefficients

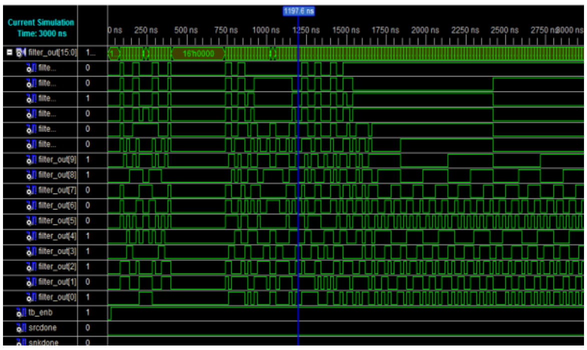


Figure11. Proposed filter simulation in ISE.

Figure 10 represents filter coefficients values.Figure 11 shows group delay and is constant at a value 8 throughout all the frequencies. Figure12 depicts proposed filter simulation in ISE.

IV. HARDWARE SYNTHESIS

Table II refers to resource utilization of FIR filter design in FPGA implementation on Sparten 3e. Area resources can be no. of slices, no. of flip flop slices, no. of LUTs, no. of bonded Input/ Output, minimum period (in ns) and maximum frequency of operation. [10] Table III gives bar chart representation of resource utilization with comparison among three folding factors, viz. 2, 4 and 8.Table III represents the maximum frequency in MHz corresponding to different Folding Factors.[11]

Table III. Performance Evaluation Based on SPARTEN 3E

Parameters	Folding Factor=2		Folding Factor=4		Folding Factor=8	
	Used/ Available	% Utilisation	Used/ Available	% Utilisation	Used/ Available	% Utilisation
Number of slices	722/ 4656	15%	438/ 4656	9%	264/ 4656	5%
Number of slice flip flops	360/ 9312	3%	261/ 9312	2%	198/ 9312	2%
Number of four input LUTs	1268/ 9312	13%	811/ 9312	8%	487/ 9312	5%
Number of bonded IOBs	35/ 232	15%	760/ 232	15%	35/ 232	15%
Minimum period(ns)	23.757		18.603		16.480	
Maximum Frequency (MHz)	42.093		53.756		60.680	

Figure12. Area comparison of FIR filters with different Folding Factors.

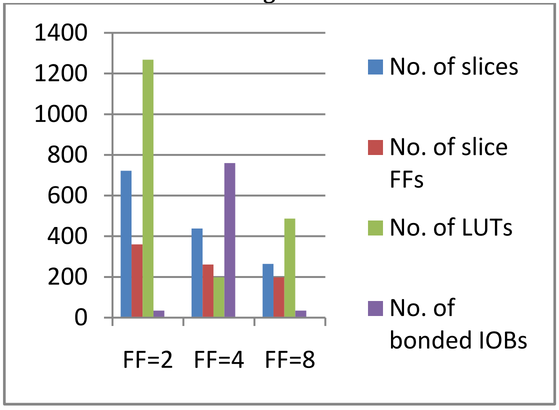
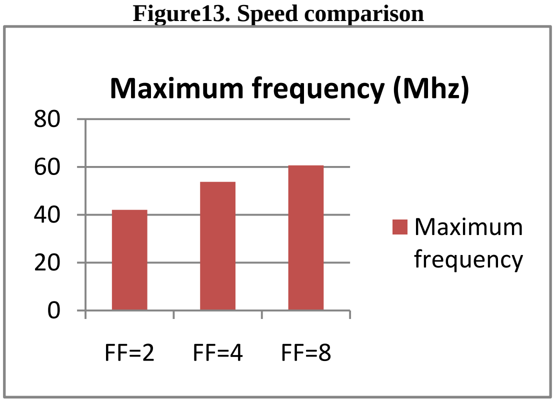


Figure13. Speed comparison



V. RESULTS AND CONCLUSION

Implementation on Sparten 3e FPGA yielded much lesser area resource utilisation. Maximum operating frequency is increasedunder similar operating conditions with increase in folding factor. Thus we conclude that the proposed design is area efficient and cost effective design and with increase in folding factor in FIR filter design, speed of operation also increases which is attractive feature in FPGA implementation world.

VI. FUTURE SCOPE

Folding factor may lead to decrease in complexity of implementation; however it needs now some extra memory to store the temporary data. Reason being that now every function block is handling N data as compared to the 1 unit data initially, and it needs to be distinguished from data produced from original operations. [12] Therefore, more number of registers are needed during implementation. Furthermore, there are multiple switching operation paths, and their proper working is ensured by using more number of multiplexers and switching elements which adds to cost.[13]

ACKNOWLEDGEMENT

Authors are highly thankful to Prof. Shyam Sundar PattnaikDirector, NITTTR, Chandigarh for constant encouragement and support during this research work.

REFERENCES

[1] Ramesh .R, Nathiya .R, “Realization of FIR filter using modified distributed arithmetic architecture”, Signal & Image Processing: International Journal (SIPIJ) Vol.3, No.1, pp. 13–17, February 2012.

[2] Mehra R., Kaur R. (2011) Reconfigurable Area and Speed Efficient Interpolator using DALUT Algorithm**.** In: Meghanathan N., Kaushik B.K., Nagamalai D. (eds) Advances in Networks and Communications. CCSIT 2011. Communications in Computer and Information Science, vol 132, pp. 5–12, Springer, Berlin, Heidelberg.

[3] MATLAB User’s Guide, “Filter Design HDL Coder 2”, chapter 3, pp. 29-145,2007.

[4] Rajesh Mehra, Lajwanti Singh, “Cost Analysis and Simulation of Decimator for Multirate Applications”, International Journal of Computers and Technology, volume 11, pp. 2175-81, 2013.

[5] Rajesh Mehra, Ravinder Kaur, “FPGA based Efficient Interpolator design using DALUT Algorithm”, NeTCoM 2010, CSCP 01, pp. 51–62, 2011

[6] ShyhJye Jou, Kai-Yuan Jheng*, Hsiao-Yun Chen and AnYeu Wu, “Multiplierless Multirate Decimator I Interpolator Module Generator”, IEEE Asia-Pacific Conference on Advanced System Integrated Circuits, pp. 58-61, Aug-2004.

[7] Rajesh Mehra, Swapna Devi, “FPGA Based Design of High Performance Decimator using DALUT Algorithm”, ACEEE International Journal on Signal and Image Processing, Volume 1, pp. 9-13, 2010.

[8] Amir Beygi, Ali Mohammadi, Adib Abrishamifar. “An Fpga-Based Irrational Decimator For Digital Receivers” in 9th IEEE International Symposium on Signal Processing and its Applications, pp. 14, ISSPA-2007.

[9] Kanu Priya, Rajesh Mehra. “Area Efficient Design of FIR Filter Using Symmetric Structure”, International Journal of Advanced Research in Computer and Communication Engineering, Volume 1, Issue 10, December 2012.

[10] Rajesh Mehra, Swapna Devi, “Optimized Design of Decimator for Alias Removal in Multirate DSP Applications”, Proceedings of the 10th WSEAS International Conference on Wavelet Analysis and Multirate Systems, pp: 100-103, 2010.

[11] Zhao Yiqiang; Xing Dongyang; Zhao Hongliang; “Optimized Design of Digital Filter in Sigma-Delta AID Converter”, International Conference on Neural Networks and Signal Processing, pp. 502 – 505, 2008.

[12] Nerurkar, S.B.; Abed, K.H.; “Low-Power Decimator Design Using Approximated Linear-Phase N-Band IIR Filter”, IEEE Transaction on signal processing, vol. 54 , pp. 1550 – 1553,2006.

[13] Mathworks, “Users Guide Filter Design Toolbox”,March-2007.

[14] D.J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. Anderson, “A Novel High Performance Distributed Arithmetic Adaptive Filter Implementation on an FPGA”, in Proc. IEEE Int.