ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 6, June 2015

FPGA Implementation of High Speed FIR Filter Using Carry Select Adder in VHDL

Ankita Sharma[1](P.G. Student, Department of Electronics and Communication Engineering, Bahra University)
Swati Kapoor[2] (Assistant Professor, Department of Electronics and Communication Engineering, Bahra University, Shimla Hills.)

***Abstract*:-In today's world the most widely used operation in digital signal processing is the FIR filtering operation. With increasing complexity of the digital circuits efficient performance is one of the main parameter to be considered while designing an FIR filter and this is the main reason for the popularity of the DSP systems. Efficient performance and area of the system are the two basic parameters which are inversely related to each other. With increase in the time efficiency of the FIR filter the area can be optimized accordingly. As we know that the FIR filter includes three basic operations. They are delaying of the input sequence, multiplication and addition or summation. This paper basically concerns replacing the multiplier by the Booth multiplier and the adder by the carry select adder. This will increase the time efficiency of the FIR filter. System level tools like Xilinx ISE 12.3 are used to design the FIR filter and applications on FPGA. The results obtained are combined with the previously designed FIR filter using different techniques.**

***KEYWORDS:* FIR filter, Booth multiplier, Carry Select Adder, FPGA, VHDL.**

I.INTRODUCTION

Filter is basically a component that separated the desired signal from unwanted signal. Its function also includes enhancing certain parts of the signal. Filters are signal conditioners which function by accepting an input signals, blocking pre-specified frequency components and passing the original signals.With the advent in VLSI technology there is a fast rise to the number of applications of integrated circuits in high-performance computing, tele-communications and consumer electronics. Digital Filters basically include IIR (Infinite impulse response) filters and FIR (finite impulse response filters). FIR filter has a number of good features as compared to IIR. FIR filters have linear phase response which makes it applicable for a number of applications. FIR filter has a number of ways to achieve but with advancement in technology the use of Field programmable gate array FPGA is gaining popularity. The high speed realization of FIR filters with less power consumption and more time efficiency and optimized area is becoming a trend. As the complexity of implementation is increasing with the filter order the realization of such structures is a challenging task. Basically an FIR filter consists of three basic components a delay element, a multiplier and an adder. Various attempts have been made in realizing different FIR filter structures. Multiplication is the strongest operation and multipliers cover a large portion of the chip area and enforce a limitation on the number of maximum processing elements (PEs) that can be accommodated and the highest order of filter that can be realized.[1]. In FIR filtering operation, convolution of two signals takes place and one of the convolving sequences is derived from the input samples whereas the other sequence is derived from the fixed impulse response coefficients of the filter.

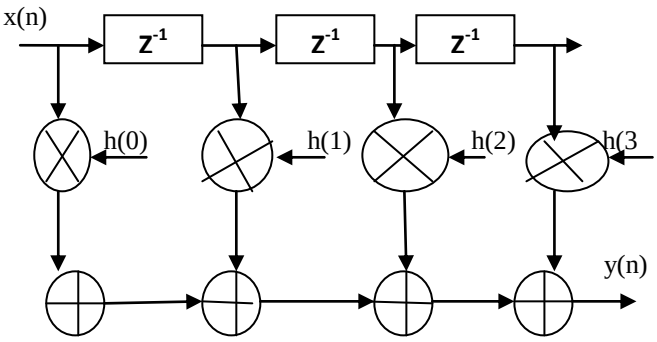


Fig. 1 Basic 3- Tap FIR filter

Fig. 1 represents the basic filter structure for a 3-tap FIR filter. The signal input is represented by the magnitude of the sampled analogue signal. Z^{-1} stores the input value and the input is delayed by one sample period. The C_{ox} blocks represent the coefficients that shape the frequency response of the FIR filter. Three arithmetic circuits multipliers adders and storage elements i.e. the delay elements are needed for the design of FIR filters. Each of these 3 parts are required to realize the structure as a digital system and each of the block can be implemented using different ways. The response of a filter to individual frequency components constituting the input signal represents the frequency response of the FIR filter. The frequency response of the filter can be analyzed in three different zones, the pass band, the stop band and the transition band. The pass band represents the band of frequencies that pass through it and indicate the filter's effect on the frequency components that can pass through it unchanged. The stop band indicates the filter's effect on frequency components that are highly attenuated. Lastly, the transition band indicates an intermediate stage between the pass band and the stop band frequencies. The frequencies in this zone receive some attenuation but are not completely removed from the output signal. Fig. 2 represents the frequency response of the filter.W_p represents the pass band ending frequency and W_s

represents the stop band beginning frequency. Frequencies between Wp and Ws lie within the transition band and are attenuated to some lesser degree. Ripple is also an additional term in filter. It specifies a peak to peak level in decibles and represents how much the filter's amplitude varies within a band. Smaller amount of ripple indicates more consistent response and is more preferable.

II. FORMULATION OF ALGORITHM AND FILTER DESIGN

Here within the basic filter designing algorithm is discussed. Considering a general system where X[n] be the input to a system and Y[n] be the output to the system. Therefore

Y[n] = X[n] convolution h[n].

$= \sum_{K=-\infty}^{n} X[k]$. h[n-k].....(1)

The above equation describes the convolution where h[n] is the impulse response of a system. The impulse response indicates the response of a filter when the input is an impulse function. Equation (1) indicates that the output of a given system is the result of all the inputs applied to it from past (-∞) to present (n) operated upon by the system. Such systems are called causal systems which do not include future input samples. Such systems are practically realizable. In the above equation we consider the present input n and Y[n] as the output at this instant. Here we multiply the input signal X[k] with the reaction of a system h(n-k) and sum up all the products to get the output. (n-k) is the delay of the system for processing the input. Consider a digital filter with the transfer function H(z).

$H(Z)$=Y(Z)/X(Z) =$(1+bZ^{-1})/ (1 + aZ^{-1}+Z^{-2})$

$Y(Z). (1 + aZ^{-1}+Z^{-2})= X(z). (1+bZ^{-1})$

$Y(z) + aZ^{-1} Y(z)+Z^{-2} Y(Z) = X(Z)+ bZ^{-1} X(Z)$

Taking the inverse Z –Transform we get,

y(n) + a.y(n-1)+ y(n-2) = x(n)+ b.x(n-1)

y(n) = -ay(n-1)- y(n-2)+ x(n)+ bx(n-1)

Therefore at Y(n) we get the sum of external inputs X(n)and X(n-1) as well as the delayed and feedback outputs Y(n-1) and Y(n-2) multiplied by appropriate coefficients. Thus filters which make the use of feedback to get the desired filter implementation are called recursive filters or feedback filters. FIR (finite impulse response) filter can be considered as functional block with X(n) as the input and Y(n) be the output as in the figure

The filter implements the following equation[2]

$Y[n] = \sum_{i=0}^{N-1} H[i].X[n-i]$.....(2)

Where N is the number of coefficients (or taps) of the desired filter, X is the input signal, Y is the output signal, Y[n] is the current output sample and H represents the filter coefficients. The Coefficients of the FIR filter are obtained using Discrete Fourier Transform (DFT) of the required frequency transfer function with some known windowing method[2]. From Equation (2) we infer that the basic FIR filter includes three components adders, multipliers and delay element. In this work replacement of the basic adder by carry select adder and the multiplier unit by the booth multiplier and comparison with the basic FIR filter is made and the results indicate that the new FIR filter structure is more time efficient.

CARRY SELECT ADDER:Carry Select is an alternative way to implement an adder which calculates the (n+1) bit, sum of two n-bit numbers. Carry Select adder is fast and simple with gate level depth of O (n)½. Carry Select Adder consists of 2 Ripple Carry Adders and a multiplexer. Addition of 2 n-bit numbers using carry select adder is done using 2 adders thus requiring 2 Ripple Carry Adder, to calculate the result twice, one time assuming carry being 0 and the other time assuming carry being 1. The main function of the multiplexer is to calculate the correct sum and correct carry after the two results are calculated. For a basic building block structure of carry Select Adder consists of 2 Ripple Carry Adders which are multiplexed together and the carry and sum bits are selected by carry-in. One Ripple Carry Adder assumes a carry-in of 0 and the second assumes a carry-in of 1, selecting the adder which has the correct assumption for the desired result. Thus we can say that the Carry Select Adder achieves high speed of operation at the cost of increased number of devices on a single chip that can be optimized accordingly. Carry Select Adder gets its speed enhancement because multiplexers are usually faster than adders. Carry Select Adder can be cascaded using longer Ripple Carry Adders, matching the increasing length of the adder to the combined delay through the multiplexers. In this 16 –bit Carry Select Adder has been used and 16-bit CSA with uniform block size can be created using three CSA blocks and one ripple carry adder block.

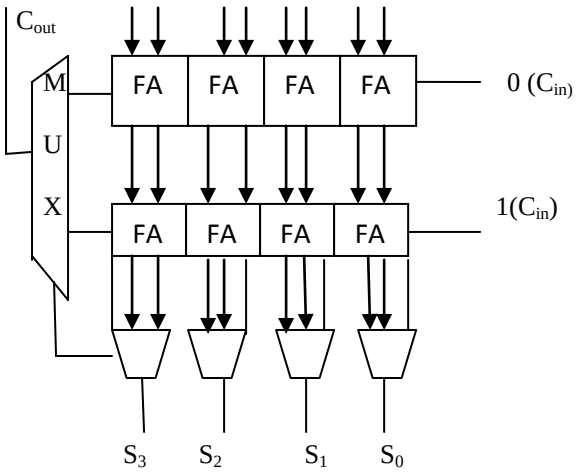


Fig. 3 Building Block of Carry Select Adder

For A 16-bit Carry Select Adder we use three building blocks as above and one Ripple Carry Adder.

MULTIPLIERS

Multiplication and Addition of two binary numbers, dominate the execution time. To reduce the power consumption, it is important to reduce the number of operations performed, thus reducing the dynamic power which is a major part of total power consumption. Multiplication has 3 basic steps, Partial product generation, partial product reduction and final addition. There are a number of multipliers that can be used. They can be explained as follows:

A.COMBINATIONAL MULTIPLIER

This multiplier is used for multiplication of 2 signed and unsigned numbers. In this each bit of the multiplier is multiplied against the multiplicand and the product is associated in accordance with the position of the bit within the multiplier and then the resulting products are then added. The basic drawback of the combinational multiplier is that it is very slow and covers a lot of area.

B. WALLACE TREE MULTIPLIER

This multiplier is used for the multiplication of two integers. It reduces the number of partial products and uses a carry select adder for addition of partial products. It includes three steps. Firstly, the multiplication of each bit of multiplier with same bit position of the multiplicand is done. Depending upon the multiplier bits generated the partial products have different weights and then the reduction of the number of partial products to 2 using layers of full and half adders. Lastly, the final addition which calculates the result.

C. MULTIPLY ACCUMULATE UNIT (MAC)

MAC affects the speed of the processor. Inputs of the MAC(X and Y) are fetched from memory location and fed to the multiplier block which will perform the multiplication of X and Y and will give it to the adder which will accumulate the result and then store the result in memory location.

D. BOOTH MULTIPLIER

Booth multiplication is an algorithm that multiplies two signed binary numbers in two's complement notation. It was developed by Donald Booth in 1950.With advancement in technology we need multipliers with high speed, low power consumption, regularity of layout and reduced area. In booth multiplier there will be a reduction in the multiplicand multiples. Booth Multiplier carries out extremely fast multiplication and thus can be used in FIR filters.

III. FILTER IMPLEMENTATION

With the advent in technology two basic constraints are considered for designing of digital circuits. These two constraints include the area constraint and the timing constraint. Area constraint includes reduction in the area of the chip. The timing constraint indicates the efficiency of the circuit. The lesser is the execution time greater is the efficiency of the filter. In this paper timing constraints are taken into consideration. The adder is replaced by the carry select adder which is fast as compared to ripple carry and carry look ahead adder and gives the results which reduces the delay of the system and increases the time efficiency of the filter. For area reduction booth multiplier is used which will optimize the overall area of the chip.

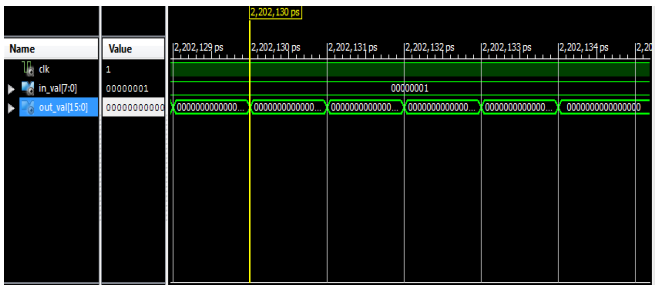
IV. RESULTS AND COMPARISON

Design equipped to 16-bit carry select adder is accomplished to structural description via VHDL hardware description language using Xilinx ISE V.12.3 software synthesized and implemented on FPGA in VIRTEX IV family. The table shows the various timing constraints attained after the comparison of the basic FIR filter with that of the new designed FIR filter.

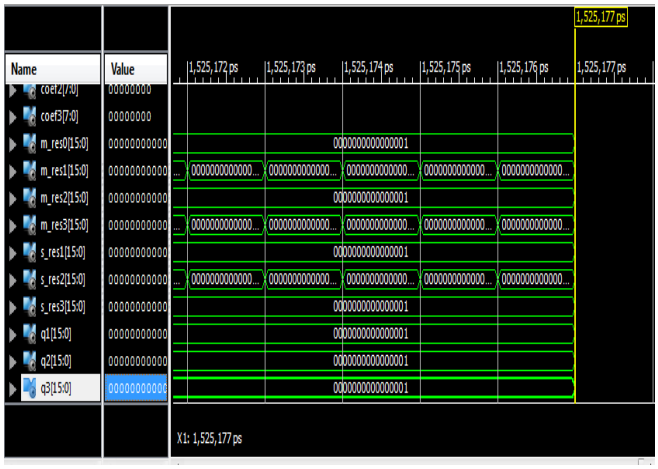
TABLE I.SLICES, DELAY,LUT'S and FFSs COMPARISON[3]

Filter	Delay(ns)	Slices	LUT's	FFs
Basic Filter	5.8	264	213	369
Implemented Filter	4	373	406	413

SIMULATION OF BASIC FILTER



SIMULATION OF THE DESIGNED FILTER



REFERENCES

[1] Shrutisagar Chandrasekran and Pramod Meher,"*FPGA* Realization FIR filters by Efficient and Flexible Systolization using Distributed Arithmetic",*IEEE Transactions on Signal Processing, January 2007.*

[2] Fabio Fabian Daitx and Vagner S. Rosa,"VHDL Generation of Optimized FIR filters", *International Conference on Signals, Circuits and Systems,* vol. 978, no.1, pp.4424-4628, july.2008.

[3] J.G. Proakis and D.G.Manolakis,"*Digital Signal Processing: Principles, Algorithms and Applications.* Upper Saddle River, NJ:Prentice Hall, 1996.

[4] A. Antoniou,"*Digital Filters: analysis, design and applications.* New York: McGraw-Hill, 1993.

[5] K.K.Parhi,"*VLSI Digital Signal Processing Systems: Design and Implementation.* New York: John Wiley & Sons, Inc, 1999.

[6] B.K. Mohanty and P.K.Meher, " Cost Effective novel flexiblecell-level systolic architechture for high throughput implementation of 2-D FIR filters", *IEEE Proceedings Computer and Digital Techniques,* vol. 143, no. 5, pp. 436-439, Nov. 1996.

[7] H. Yoo and D.V.Anderson,"Hardware-Efficient Distributes Arithmetic Architechture for higher order Digital Filters", in Proc. *IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '05),*vol 5. Mar 2005, pp.v/125-v/128.

[8] Xilnx, Inc., 2100 Logic Drive, San Jose, CA 95124-3400. [Online]. Available: www.xilinx.com.

[9] D.J. Allred, H.Yoo, V. Krishnan, W. Huang, and D.V. Anderson,"LMS adaptive filters using Distributed arithmetic for high throughput," *IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 52, no. 7, pp. 1327-1337, July 2005*.

Ankita Sharma[1] born in Himachal Pradesh, India is currently pursuing her regular M.Tech in Low Power VLSI design from Bahra University, Shimla Hills. She Completed her regular B.Tech in Electronics and Communication Engineering from I.E.T.Bhaddal, Ropar, Punjab, affiliated to Punjab Technical University in 2012

Swati Kapoor[2] born in Himachal Pradesh, India is currently working as an Assistant Professor, department of electronics and communication Engineering at Bahra University, Shimla Hills. She Completed her B.Tech from I.E.T.Bhaddal, Punjab affiliated to Punjab Technical University. She completed her M.tech in Embedded System in 2013 from CDAC,Mohali. She has her specialization in Embedded systems and has published a paper on"Modular Design of an automated analyzer"in International Journal of Computer Applications and "Design and interface of Optical Assembly for an automated analyzer in international Journal of Scientific Research.