ISSN 2277-3061

Systolic Architecture for High Speed FIR Filter Using VLSI Technology

Wavhal Monika Dattatraya[(1)], Wagh Aishwarya Karbhari[(2)] , Sulakhe Manasi Mahesh[(3)], Prof. P.V.Sriniwas Shastry[(4)]

[(1,2,3)] Bachelor of Engineering, Cummins college of Engineering for women, Pune,India

[(4)]Masters in(Electronics and Telecommunication), Cummins college of Engineering for women, Pune,India

ABSTRACT

The tremendous growth of computer and Internet technology wants the data to be processed athigh speed. Low power consumption, high throughput and optimized hardware are the most important design criteria's for VLSI implementation. This project gives an efficient design of high speed FIR filters using systolic architecture. In this paper we have implemented 7th,8th,11th order FIR filter with 8 bit normalized input. To obtain efficient results we have selected B1 design from all the designs of systolic arrays.GF (28) multiplier and XOR adder are used for multiplication and addition in filter. Hamming window technique is used to derive the filter coefficients. The coefficients of filter are found out using Matlab. The FIR filter architecture is effectively synthesized and simulated using Xilinx ISE 8.1i in VHDL and Modelsim simulator. Maximum frequency, timing simulation delay and number of slices were used as performance metrics. The results obtained are compared with the existing results achieved for FIR filter, thus our work proves that the objective of high speed has been achieved successfully with the use of minimum number of slices.

Indexing terms/Keywords

GF (28) Multiplier; Systolic architecture; Galois Field; FIR filter; high speed filter.

Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol. 14, No. 11

www.ijctonline.com, editorijctonline@gmail.com

ISSN 2277-3061

INTRODUCTION

The Finite Impulse Response (FIR) digital filters are widely used in digital signal processing applications due to their stability and linear phase properties. Digital Signal Processing (DSP) is widely used inreal time applications such as video, image processing and wireless communication.Multipliers and adders are the keycomponents of such high performance system such as FIR filter. A system performance can be determinedby the performance of anadder and multiplier becausethey are the processing elements in the system, they are generally most area and time consuming. Therefore optimizing the speed and the area of the adder and multiplier is the major task. Optimizing the speeds of multiplier and adder will enhance the speed of the filter.

Many previous efforts [8]-[10] have been focused on high speed, area optimization and power reduction implementations. One approach byPramod Kumar Meher et.al [10] in July 2008 has proposed the design optimization of one-and two-dimensional fully pipelined computing structures for area-delay-power-efficient-speed implementation of FIR filter by systolic decomposition of distributed arithmetic (DA) based inner product computation. Inner product computation is done by accessing lookup table (LUT) followed by shift accumulation operations of the LUT output. The major difficulty encountered in this scheme is that as the filter size or the number of bits used to represent coefficients increases the memory requirement increases exponentially.

Oscar Gustafson et.al[13] worked on implementation of Low complexity FIR filter using serial arithmetic. The effects of digit size on FIR filters using multiplier block techniques are studied. In digit serial arithmetic words are divided into digits of d bits that process one digit at a time. This provides a tradeoff between area, speed and power consumption.

Another approach by Jongsun Park et.al[8] proposed in April 2003, uses computation sharing multiplier which specifically targets computation reuse in vector-scalar products for high speed design using direct form and transposed direct form as their FIR filter architecture.This paper compared the performance of their implementation with Wallace tree and carry-save multipliers, which shows that sharing multiplier scheme improves speed by 33% and 52% respectively.

H.T.Kung et.al[1], discussed various systolic arrays in his work and mentioned the advantages and features of systolic architectures.

Inthis paperwe have implemented high speed FIR filter using systolic architecture.Evaluating the advantages and disadvantages of all the systolic array designs we have chosen the B1 design proposed by K.Parhi[3]. 8-bit normalised test input signal, generated in Matlab is given to the filter. Hamming window method is used to find the coefficients of the proposed filter. According to the various developments in this particular field, the distinguishing element of our paper is that we have designed FIR filter which uses systolic architecture and Galois Field multiplier which has not been worked on previously.

The rest of the paper isorganized as below: Section (I) deals with the introduction to the proposed FIR filter.Section (II), deals with the fundamentals of Systolic FIR filter and its features. The proposed FIR filter's architecture and its implementationis broadly discussed in Section (III). The various software tools used for the proposed filter design and implementation are discussed in section (IV). The results obtained by implementation of GF multiplier and FIR filter are tabulated and compare with the other filter implementation results in Section (V). Hence conclusion is drawn in Section (VI).

SYSTOLIC ARCHITECTURE FOR FIR FILTER

FIR Filter

A finite impulse response (FIR) filter is a filter whose response to any bounded input is of finite durationi.e. we get a bounded output.

Where, x[n] is input signal, N is the order of the filter and y[n] is the output signal

A FIR filter has a number of useful properties which sometimes make it preferable to an infinite impulse response (IIR) filter, that are listed as below:

1. It requires no feedback. They are nonrecursive in nature. The relative errors which occur are not iterative and are not compounded.
2. They are inherently stable, since the output is a sum of a finite number of finite multiples of the input values, so there is no iterative loop in the equation like IIR filters. For every bounded input,the output of FIR filter is bounded.
3. It can easily be designed to be linear phase by making the coefficient sequence symmetric. Linear phase means there is a constant group delay in the filter response.

The primary objective of our proposed architecture is to increase the speed for which the non-recursive nature, stability and constant group delay of FIR filter becomes advantageous properties as compared to an IIR filter for hardware implementation.

ISSN 2277-3061

Systolic architecture

A systolic architecture is a pipelined network arrangement of Processing Elements (PEs), which are called cells. It is a combined form of parallel computing and pipelining, where cells compute the data that arrives as input and store them independently. Systolic architecture represent a network of processing element (PEs) that rhythmically compute and pass data through the stem, the PEs regularly pump data in and out such that regular flow of data is maintained, as a result systolic array features modularity and regularity which are important property for VLSI implementation. The data in systolic array may be passed through coprocessors in combination of host computersthat hosts the PEs and the final result is return to host computer.

The features of systolic array can be listed as below:

1. Synchrony:A systolic array is controlled by a global clock with fixed length ofclock cycles. This global clock also synchronizes this array.
2. Modularity and regularity:Modular processing units which are connected withhomogeneous interconnections and also the computingnetwork can be extended indefinitely.
3. Pipelining ability:The array exhibits a linear rate pipelineability to speed up processing rate, i.e., it shouldachieve an O(N) speedup, in terms of processingrate, where N is the number of ProcessingElements. Here the efficiency of the array ismeasured by the following:Where Ts is the processing time in a single Processor, and Tp is the processing time in the arrayprocessor.
4. Simple and regular design:In integrated-circuit technology, the cost ofdesign grows with the complexity of the system. By using this, cost is reduced and hence it is a cost-effective design.

OUR IMPLEMENTATION

This section describes our proposed system and methodology for implementation of FIR filter based on systolic architecture, using GF multiplier. From all the systolic arrays we chose B 1 design which broadcast input to all processing elements, weights stay and output is collected after shifting the output at each PE byte wise [3]. [3].GF(2^8) multiplier is designed by using isomorphic transformation technique. Coefficients are calculated using Hamming window method. Various order filters like 7, 8 and 11 are implemented and compared. By exploiting the advantages of systolic architecture and GF multiplier, high speed and low area implementation of FIR filter is achieved.

Architecture Of Proposed Filter

From all the systolic arrayswe chose B 1 design which broadcast input to all processing elements, weights stay and output is collected after shifting the output at each PE byte wise[3].Due to broadcast inputs, high throughput is achieved. Hardware utilization factor of B1 is 1 and it produces the output in one clock cycle. It does not require extra bus for an output and it also uses less memory as compared to other designs except F design.

Architecture consists of 7 processing elements and each processing element has a multiplier and a accumulator. Normalized 8 bit input is broadcasted to all the PEs, each GF multiplier, multiplies the input with the filter coefficient. The multiplied output is XOR with the previous multiplied output stored in a register. We get the filter output at every clock cycle.

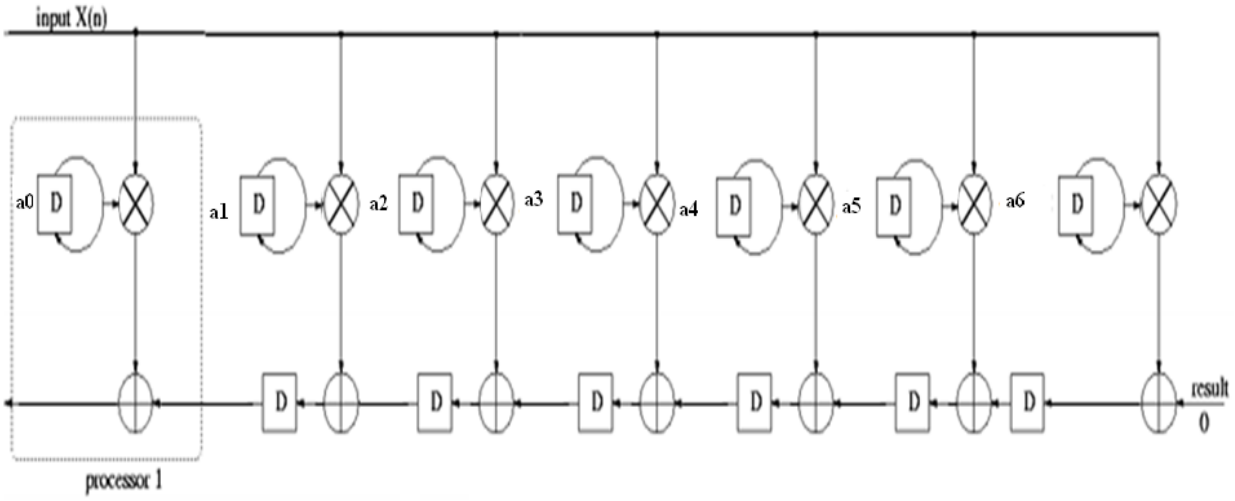

Figure.1 Signal flow graph for 7th order FIR filter

ISSN 2277-3061

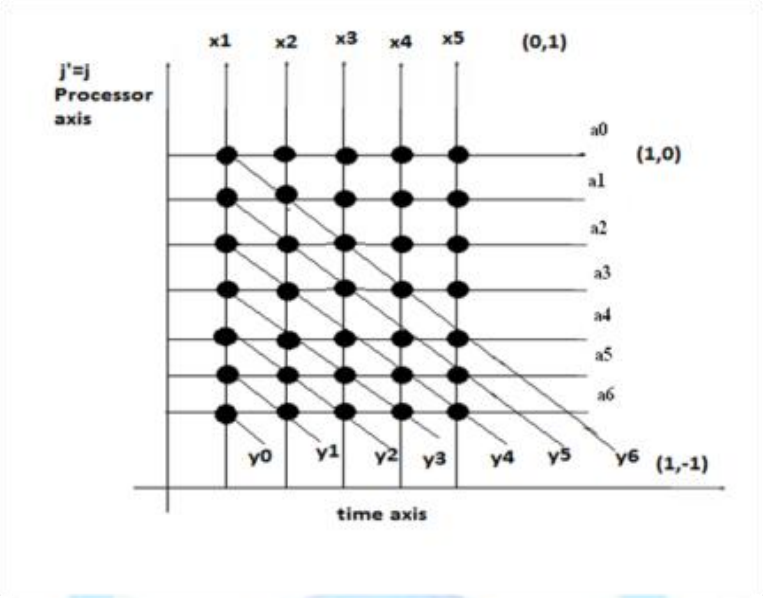

Figure 2.Space-time representation of B1 design

The Space-time representation of B1 design is shown in fig from which we can see that the incoming x value is available at all the processors at the same time. Specifically, the input data is "broadcast" to the processors. Similarly, the weights a_i, appear at the processors at the same spatial co ordinates. Thus $a_i$values stay. And the outputs y_i, appear at the processors at different space and time. Hence the outputs move.

GF Multiplier

Galois Field theory deals with numbers that arerepresented in binary andthey have the properties of a mathematical "field," and are finite in scope. Galois operations comprises of Addition, multiplication and logarithms. The whole Galois field design is purely combinational and it is a design which does not require clock.

The message signal is taken in form of the multiplicand that denotes 8 bit of data. Galois algorithm is implemented on the multiplicand using the generator key irreducible polynomial and a 8 bit multiplier key. Mathematically 8 bit multiplication results in the 16 bit of the result but the Galois technique will result 8 bit resultant for 8 bit multiplication. As for the case of n bit multiplication it will result in n bit result.

We have implemented GF multiplier by using Isomorphic transformation which decomposes $GF(2^8)$ to $GF((2)^2)$ by using different combinations of φ and λ.

In order to accomplish these conversions some irreducible polynomials are required asmentioned below.

$GF((((2)^2)^2)^2)$ to $GF(((2)^2)^2)$: $x^2 + x + \lambda(2)$

$GF(((2)^2)^2)$ to $GF((2)^2)$: $x^2 + x + \varphi(3)$

$GF((2)^2)$ to $GF(2)$: $x^2 + x + 1(4)$

The selection of φ and λ may lead to different gate counts while implementing as combinational logic circuits. In our design we have taken φ and λ as {10} and {1100} respectively.

The advantages of GF Multiplier are as follows:

1. The Galois Field multiplier has small hardware footprint.
2. It has short response latency.
3. Final Implementation of Galois Field multiplier makes use of only XOR gates.
4. It gives direct answer of multiplication. Therefore no registers are required to store partial products.
5. Isomorphic transformation converts the 8 bit number into smaller one bit number, due to this complexity of multiplication operation is reduced.

ISSN 2277-3061

Design of GF Multiplier:

GF(2^2) Multiplication[4]:

Let $k=q\times w$,

where$k= \{k_1\ k_2\}$,

$q= \{q_1\ q_2\}$,

$w =\{w_1\ w_2\}$ are the elements of GF $((2)^2)$.

The formula for computing multiplication in GF$((2)^2)$is as follows:

$k1= q_1.w_1 \oplus q_1.w_0 \oplus q_0.w_0$

$k2= q_1w_1 \oplus q_0w_0$

Hardware implementation of multiplication in GF$(((2)^2)^2)$:

$k= q\times w$

where$k=\{k_1\ k_2\ k_3\ k_4\}$;

$q= \{q_1\ q_2\ q_3\ q_4\}$;

$w= \{w_1\ w_2\ w_3\ w_4\}$

$k_{H..}x + k_L = (q_{H..}x + q_L)(w_{H..}x + w_L)$;

$k= (q_H.w_H)x^2 + (q_H.w_L + w_H.q_L)x + q_L.w_L$

Substituting x^2 term with $(x + \varphi)$yields

$k_Hx + k_L = (q_H.w_L + w_H.q_L + q_H.w_H)x + q_H.w_H\ \varphi + q_L.w_L$

Hardware implementation of multiplication in GF$((((2)^2)^2)^2)$:

$k= q\times w$

where$k=\{k_1\ k_2\ k_3\ k_4\ k_5\ k_6\ k_7\ k_8\}$;

$q= \{q_1\ q_2\ q_3\ q_4\ q_5\ q_6\ q_7\ q_8\}$;

$w= \{w_1\ w_2\ w_3\ w_4\ w_5\ w_6\ w_7\ w_8\}$;

$k_H.x + k_L = (q_H.x + q_L)(w_H.x + w_L)$;

$k= (q_Hw_H)x^2 + (q_H.w_L + w_H.q_L)x + q_Lw_L$

Substituting x^2 term with $(x + \lambda)$ yields,

$k_H.x + k_L = (q_Hw_L + w_H.q_L + q_H.w_H)x + q_H.w_H\ \varphi + q_L.w_L$

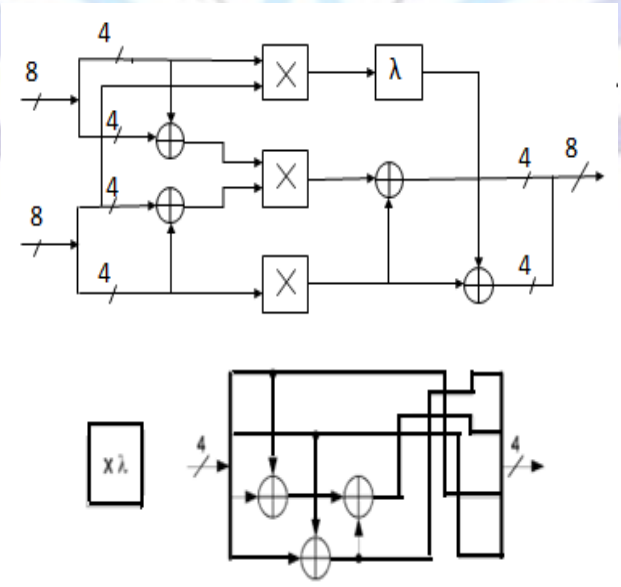

Figure 3.Hardware implementation of multiplication in GF(2^8)

ISSN 2277-3061

Adder

All additions in GF Multiplier are Modulo-2 additions which can be implemented by using XOR gates. Hence XOR gates are used for the addition operation in the FIR filter implementation.

Finding the Filter Coefficients

For the design of FIR filter, the important step is to find the filter coefficients. The filter coefficients can be calculated in several ways as: window design method, frequency sampling method and paks-McClellan method. The simplest way of designing filter is by using the windowing techniques. Filter coefficients can be obtained using any of the various windows available.

Hamming window method is particularly well-suited for this implementation because of their advantages which are as mentioned below:

It minimizes the side lobe level and offers better selectivity for higher frequencysignals. Here, hamming window is used for the calculation of filter coefficients.

The hamming window impulse response coefficients is given as below:

$$w(n)=0.54-0.46\cos(2\pi n/N-1) \quad (5)$$

We calculated coefficients for filter using MATLAB software for the following specifications:

Cut-off frequency:1KHz (Scalable)

Order of the filter:7, 8, 11

The obtained filter coefficients by hamming function in Matlab were converted to fixed point decimal. These coefficients are converted to binary and stored in memory, which are accessed in the main filter program.

Test Input Signal

Composite signal is given as an input to the FIR filter to test the filter response. The test signal is generated in MATLAB. This signal is sampled and random 10 values are normalized and stored in registers. Chirp command is used to generate this test signal which is a frequency-swept cosine waveform, in which the frequency varies in similar time intervals.

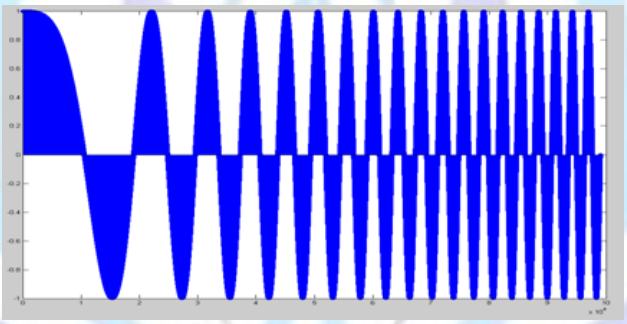

Figure 4.Synthesized Input signal for Simulation

DESIGN ENVIRONMENT

FIR filter is implemented using XILINX ISE 8.1i. Behavioural simulation is done on built code to verify the functionality of the FIR filter. Timing simulation is also performed on MODELSIM simulator.

The GF multiplier is implemented on Virtex2p, Virtex E and Virtex 4 for the purpose of comparison with other multipliers.

The proposed filter was simulated on Virtex4 FPGA device with speed grade 12.MATLAB is used to find filter coefficients and to generate thecomposite input signal.\

RESULTS AND COMPARATIVE STUDY

GF multiplier

GF multiplier is implemented using Xilinx 8.1 synthesis tool and output is observed on Modelsim simulator. We selected Galois Field multiplier for the implementation as it does not give partial product so it helps to increase the speed of the filter. Table 1 shows the comparison of GF multiplier with other multipliers used by Jongsun Park[8] and Laxman.S[15]. From the Table 1 we can conclude that GF multiplier is 45% faster than Wallace Tree multiplier, 52% faster than array multiplier and 24% faster than Booth multiplier.

ISSN 2277-3061

Table 1

Multiplier	Path delay	Device used	Speed grade	Slices
Jongsun Park-[8]	23.39ns	Virtex E	-8	----
Jongsun Park-[8]	16.68ns	Virtex E	-8	----
Laxman S-[15]	19.33ns	Virtex - 2p	-7	132
Laxman S-[15]	12.33ns	Virtex - 2p	-7	114
Our implementation (GF multiplier)	9.26ns	Virtex - 2p	-7	43
	9.14ns	Virtex E	-8	43
	4.32ns	Virtex 4	-12	43

FIR Filter Results

FIR filter was implemented on XILINX synthesis tool with VIRTEX 4 device using speed grade 12 and output is observed on Modelsim simulator.

Table 2. Comparison of Implementation and their Results

Method of implementation	Order of filter	Device used	Frequency (MHz)	Slices	Throughput /slice ($\times 10^6$)
PramodkumarMeher*et.al*[10]	8	Virtex E	62.12	149	3.338
Jongsun Park, Khurram Muhammad *et.al*[8]	11	Virtex E	55	----	----
Jongsun Park, Khurram Muhammad *et.al*[8]	11	Virtex E	40	----	----
Jongsun Park, Khurram Muhammad *et.al*[8]	11	Virtex E	83.33	----	----
(Our Design) FIR filter using systolic architecture and GF(2^8)multiplier	7	Virtex 4	166.67	30	44.3
	8	Virtex 4	166.67	32	41.56
	11	Virtex 4	142.86	44	25.95

The results obtained from timing simulation were compared with existing architectures of FIR filter. The comparison clearly indicate that our proposed FIR filter architecture using Galois Field multiplier utilizes less area and operates on maximum frequency of 166MHz .Our implemented filter achieved 71% increase in speed using very amount of hardware resources compared to Jongsun Park[8] which can be gauged based on throughput per slice parameter in Table 2.

CONCLUSION

This paper describes design of Systolic architecture, GF multiplier & final FIR implementation for achieving high speed. It is achieved by using an efficient implementation of systolic array structure.

From comparison with other multipliers we observed a significant delay reduction in our GF multiplier which was 24% faster than Booth Multiplier and 52% faster than Array Multiplier. We validated our filter design using Virtex 4 device where we observed 71% increase in speed for 11th order FIR Filter when compared with Jongsun Park[8].

In future we would like to modify our design to obtain further increase in speed by implementing pipelining techniques.

ACKNOWLEDGMENTS

We would like to express our sincere gratitude towards our guide Prof. P.V.S. SHASTRY for their constant encouragement and valuable guidance during the completion of this work.

We would also like to thank Dr. Prachi Mukherjee for her continuous valuable guidance, support, valuable suggestions and his precious time in every possible way in spite of her busy schedule throughout our project activity.

We take this opportunity to express our sincere thanks to all the staff members of Electronics and Telecommunication Department for their help whenever required. Finally, we express our sincere thanks to all those who helped us directly or indirectly in many ways in completion of this work.

REFERENCES

[1] H.T. Kung ,"Why systolic architecture", IEEE transaction (0018-9162/82/0100-0037,January 1982)

[2] P.V. Sriniwas Shastry, Mukul S. Sutaone, "Multiplicative Inverse in GF(2^8) Using Combinational Logic Circuits", IETE National Journal of Innovation and Research, Vol.1, No.1., June 2013.

[3] Keshab K. Parhi , "VLSI Digital Signal Processing Systems"

[4] Edwin NC Mui, "Practical Implementation of Rijndael S- Box Using Combinational Logic"

[5] Xinmiao Shang, Keshab K. Parhi, "On the optimum constructions of composite field for the AES algorithm", IEEE Trans. On Circuits and Systems –II: Express Briefs, Vol 53, No. 10, pp. 1153-1157, 2006.

[6] Soniya1, Suresh Kumar ,"A Review of Different Type of Multipliers and Multiplier-Accumulator Unit" paper published in IJETTCS.

[7] Pramod Kumar Meher, "Systolic and Super-Systolic Multipliers for Finite Field GF(2^m) Based on Irreducible Trinomials", IEEE

[8] Jongsun Park, Khurram Muhammad, and Kaushik Roy, "High-Performance FIR Filter Design Based on Sharing Multiplication", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 11, No. 2, April 2003.

[9] Marcos Martinez-Peiro' and Lars Wanhammar, "High-speed, low-complexity fir filter using multiplier block reduction and polyphase decomposition", ISCAS 2000,IEEE International Symposium on Circuits and Systems, May 28-31, 2000, Geneva, Switzerland.

[10] Pramod Kumar Meher, Shrutisagar Chandrasekaran, Abbes Amira, Senior, " FPGA Realization of FIR Filters by Efficient and Flexible Systolization Using Distributed Arithmetic", IEEE Transactions on Signal Processing, Vol. 56, No. 7, July 2008.

[11] RuiGuo, Lei Wang and Linda S. De Brunner, "A Novel FIR Filter Implementation Using Truncated MCM Technique", IEEE Transaction.

[12] Shahnam Mirzaei, Anup Hosangadi, Ryan Kastner," FPGA Implementation of High Speed FIR Filters Using Add and Shift Method" ,IEEE Transactions.

[13] Kenny Johansson, Oscar Gustafsson, and Lars Wanhammar, "Implementation of Low-Complexity FIR Filters Using Serial Arithmetic", Transactions IEEE.

[14] Patent : Yu Fei Li, Shanghai (CN); Yong Lu, Shanghai (CN); Guang Chang Ye, Shanghai (CN); Fan Zhou, Shanghal (CN) ,Pub. No.: US 2010/0306293 A1

[15] Laxman S, Darshan Prabhu R, Mahesh S Shetty, "FPGA Implementation of Different Multiplier Architectures", IJETAE (ISSN 2250-2459, Volume 2, Issue 6, June 2012).