International Journal of Engineering Sciences & Emerging Technologies, August 2012.
ISSN: 2231 – 6604 Volume 3, Issue 1, pp: 90-97 ©IJESET

DESIGN AND IMPLEMENTATION OF 120 ORDER FIR FILTER BASED ON FPGA

B. Mamatha[1], V.V.S.V.S. Ramachandram[2]
[1]M.Tech Student (Embedded Systems), Dept of ECE,
Pragathi Engineering College, A.P, India.
mamatha.bandaru@gmail.com

[2]Assoc. Professor, Dept of Electronics and Communication Engineering
Pragathi Engineering College,A.P, India
vsrc_2005@yahoo.com

ABSTRACT

Distributed algorithm is suitable for FPGA to do multiply-accumulate operations, which use the abundant memory resources of FPGA to do look-up table operation. We present a method for implementing high speed Finite Impulse Response (FIR) filters using just registered adders and hardwired shifts. We extensively use a modified common sub expression elimination algorithm to reduce the number of adders. We target our optimizations to ALTERA devices we can save up to 50% reduction in the number of slices for fully parallel implementations. We can also achieve power reduction in the total power consumption of the filters. we presented a multiplier less technique, based on the add and shift method and common sub expression elimination for low area, low power and high speed implementations of FIR filters where we observed significant area and power reductions over traditional Distributed Arithmetic based techniques. Our designs perform significantly faster than the existing filters, which use embedded multipliers. The proposed technique efficiently generates adder graphs for the entire set of fixed coefficient by reusing common adder graphs and the partial sums as much as possible in order to reduce the hardware complexity and latency to implement FIR Filter. The multiplier-less method is based on the replacement of multiplications with a minimum number of additions and shifts. The proposed method is performed and compared to a previous one. The results of FPGA implementations on Altera Cyclone II show the increase of the maximum frequency, the decrease of the resources usage and the reduction of the dynamic power with a new proposed FGA algorithm. Another comparison with recent published results has been done and proves the efficiency of the proposed.

KEYWORD: *ALTERA devices, Common Sub Expression Elimination, Embedded Multiplier, Finite-Impulse response filters, FPGA, hardware complexity, look Up Table.*

I. INTRODUCTION

Filters are a basic component of all signal processing and telecommunication systems. Filters are widely employed in signal processing and communication systems in applications such as channel equalization, noise reduction, radar, audio processing, video processing, biomedical signal processing, and analysis of economic and financial data. For example in a radio receiver band-pass filters, or tuners, are used to extract the signals from a radio channel.

Finite impulse response (FIR) filters are the most popular type of filters implemented in software. A digital filter takes a digital input, gives a digital output, and consists of digital components. In a typical digital filtering application, software running on a digital signal processor (DSP) reads input samples from an A/D converter, performs the mathematical manipulations dictated by theory for the

required filter type, and outputs the result via a D/A converter. An analog filter, by contrast, operates directly on the analog inputs and is built entirely with analog components, such as resistors, capacitors, and inductors.

Multiplier block consists of additions, subtractions and shift operations. The Multiplier Block is used to implement a parallel multiplication of a variable x with a set of fixed coefficients. Generation of the minimal cost Multiplier Block from a set of fixed coefficients is known as the multiple constant multiplication (MCM) problems [10]. The complexity of Digital Finite Impulse Response (FIR) filters is dictated by the number of adders/subtractions to implement the coefficient multipliers.

Figure.1 shows the basic block diagram for an FIR filter of length N. The delays result in operating on prior input samples. The h_k values are the coefficients used for multiplication, so that the output at time n is the summation of all the delayed samples multiplied by the appropriate coefficients.

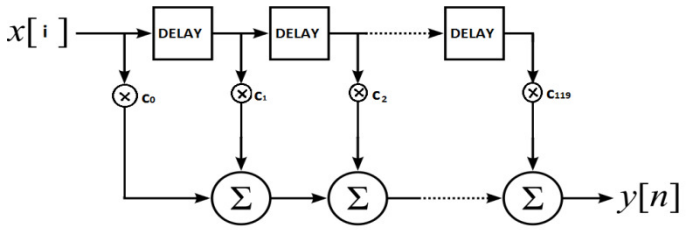

Figure.1. Logical structure of an FIR filter

$$Y(n) = \sum_{i=0}^{119} c(i)x(i) = c(0)x(0) + c(1)x(1) + c(2)x(2) + \ldots\ldots + c(119)x(119) - - - (1.1)$$

c(i) =constant or filter coefficient
x(i) = nth point of input sequences is variable
y(n)= represents the system response

This will be written as

$$Y(n) = \sum_{b=0}^{B-1} 2''' X \sum_{i=0}^{119} f(c(i)x(i)) - - - - - - - - - - (1.2)$$

The longer the filter (more taps), the more finely the response can be tuned with the length, N, and coefficients, float h [N] = {...}, decided upon, the implementation of the FIR filter is fairly straightforward. Running this code on a processor with a multiply-and-accumulate instruction (and a compiler that knows how to use it) is essential to achieving a large number of taps.Distributed algorithm is suitable for FPGA to do multiply-accumulate operations, which use the abundant memory resources of FPGA to do look-up table operation [12].

The traditional algorithm FPGA realizing FIR filters base on multiply-accumulate structure and parallel multiplier structure. But when the filter order is large, both of the two structures of FIR filters will have a significant computational resources and a relative large occupation. According to this problem, a design method of 60-order FIR filter based on distributed algorithm is put forward in this paper.

Because of the limited hardware resources of FPGA, applying directly multiplication will consume large amounts of resources. Distributed algorithm is suitable for FPGA to do multiply-accumulate operations, which use the abundant memory resources of FPGA to do look-up table (LUT: Look up Table) operation [11]. The FIR low-pass filter based on distributed algorithm is implemented on FPGA in this paper, reducing hardware size by using the linear phase FIR filter symmetry, reducing storage space with segmentation of LUT, improving the speed with parallel distributed algorithm structure and pipelining technology.

We present a method for implementing high speed Finite Impulse Response (FIR) filters using just registered adders and hardwired shifts which is discussed in CSD representation in section 3(a). We extensively use a modified common sub expression elimination algorithm to reduce the number of adders. We target our optimizations to ALTERA devices we can save up to 50% reduction in the number of slices for fully parallel implementations. We can also achieve up to 50% reduction in the

total power consumption of the filters. Our designs perform significantly faster than the existing filters, which use embedded multipliers.

II. DISTRIBUTED ALGORITHM

Distributed algorithm is referred to as DA. In the design process of FIR filters, filter coefficients can be obtained by Mat lab. Because the coefficients are constants, then h (i) x (n-i) becomes constant multiplication, only the scaling for input sequences[4].This important characteristic also is the precondition of distributed algorithm. If c(n) is a known constant (such as filter coefficient), x(n) regarded as the nth point of input sequences is variable and y(n) represents the system response (such as the nth point of output sequence after filtering), the N-order inner product of x(n) and y(n) is:

- Intuitive realization structure of DA: the size of the look-up table is decided by the order number of the filter, so the larger the order number is, the greater the demand of LUT units is. Therefore, this structure of DA algorithm isn't suitable for the case that N is large.
- Combination of series and parallel structure: it is a basic transformation of intuitive structure of DA, which copy LUT J (2□J<B) portions, so the system needs B/J clock cycles to get the accumulation results.
- All parallel structure of DA: also called speed optimal algorithms, based on this structure of DA algorithm[4], filtering can be completed in one clock cycle, but hardware resources occupation is larger. The implementation method given in this paper using this all parallel structure and pipelining either improves filter speed or reduces design scale greatly.

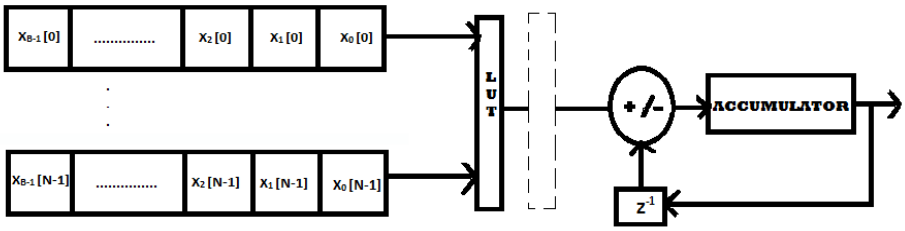

Figure.2. Block diagram of DA algorithm

III. FILTER ARCHITECTURE

3.1 Canonical Sign Digit Algorithm (CSD)

The CSD representation is radix-2 signed digit system with the digit set {1, 0,$\bar{1}$}, where $\bar{1}$ denotes -1.Given a constant, the corresponding CSD representation is unique and has two properties, the first is that the number of non zero digits is minimal and the second is that the product of two adjacent digits is zero; i.e two nonzero digits are not adjacent. An Encoding a binary number such that it contains the fewest number of non-zero bits is called Canonical Sign Digit .A CSD representation is a kind of sum of signed power of two representations. Unlike binary numbers, that is expressed using only 0 and1, but the CSD representation and the SPT representation use 0, 1 and -1.

$$\text{Ex: } 71*X = 1000111_2*X = X \ll 6 + X \ll 2 + X \ll 1 + X \text{ (shift/add operation) -------- (3.1)}$$

$$1000111_2*X = 100100\text{-}1*X = X \ll 6 + X \ll 3 - X \text{ (CSD) ---------- (3.2)}$$

With Ci € {-1, 1, 0}, where (- denotes -1). It is signed digit number system that minimizes the number of non-zero digits .It can reduce the number of partial product additions in a hardware multiplier. They are successful in implementing multipliers with less complexity. Since the complexity of the multipliers is typically estimated through the number of non-zero elements, which can be reduced by using signed digit numbers. Adjacent CSD digits are never both non-zero.
For negative numbers, the numbers of non-zero digits is less for the CSD Representation than the 2's Complement representation. The CSD numbers has the minimum number of non-zero digits and no consecutive nonzero digits [9] .The CSD Representation has fewer nonzero digits than the normal binary expression. Now the multipliers in the digital filters are realized with shifters, adders and

subtractions. The use of CSD expression can reduce the number of adders and sub tractors for example, the normal binary representation would need 3 adders, as 15 is represented as 11112. The number of adders and sub tractors is less than the number of non-zero digits by 1.The CSD Multiplier is based on shifts and adds (or subtracts) instead of conventional multipliers. This results in the area reduction of multiplier of the digital filters. The Complexity of a digital filter design is a function of the number of non-zero design in the filter Coefficients. Encoding the filter Coefficient using the CSD representation reduces the number of partial products as well as the area and the power consumption. Hence, it is useful technique for implementing of FIR Filters with fixed coefficients. Digital FIR filters can take advantage of the CSD representation. The CSD representation is one of the techniques for implementing FIR filters. Instead, the CSD representation permits the minimization of the total number of non-zero bits in all of the filter coefficients that nevertheless maintains acceptable filter performance.

The Coefficient of Low pass FIR Filter is represented in signed and unsigned numbers. Two's complement arithmetic efficiently handles the addition and multiplications of signed numbers. In DSP filtering applications, coefficients are made up of both positive and negative numbers. Depending on the applications data is either positive or negative. Two's complement arithmetic efficiently handles the addition and multiplications of signed numbers. The advantages of two's complement are that we can use the same hardware to add negative numbers and positive numbers and the carry out is discarded. So first the Coefficient firstly convert into two's Complement then CSD Algorithm is applied which convert the representation into maximum number of non-zero terms. There is a flowchart which will convert the two's Complement number into CSD representation. A filter represented by a CSD code is called CSD filter. The multiplication can be easily implemented by using CSD code coefficients. The number of adders/sub tractors required to realize a CSD multiplier is one less than the number of nonzero digits in the CSD code. The Coefficients are multiplied with operands that are supplied to multiplier. Each operand may be multiplied by a coefficient. Conventional multipliers are generally implemented with combinations of shift and add operations. Some of such implementations result in inefficient use of area. The Canonical Signed Digit (CSD) representation of numbers, which further reduces the computational load when used with this Horner's method for multiplication. Here, the multiplication can be performed efficiently with just shift and add operations.

3.2 CSD Coefficient Multiplier:

A fixed-coefficient CSD multiplier can be implemented by coding the multiplier coefficient in a CSD form. CSD multipliers made from 12-bit adder and shifter [2] are implemented for each different coefficient in the CSD FIR filter. This constant multiplier has small area compared to general variable multiplier.

3.3 Reconfigurable CSD Fir Filter Design:

CSD representation permits subtraction, as well as addition, of shifted data in accomplishing multiplication. The feature of redundancy in this representation allows a coefficient implementation to be selected which in general requires fewer adders/sub tractors, and thus yields a faster more compact multiplier is shown in figure 3. However, due to the constraint imposed on CSD coefficient, the quantization level of the coefficients is large, which causes performance degradation. But, if we apply optimization techniques in the integer domain, we can obtain a CSD FIR filter with less degradation.

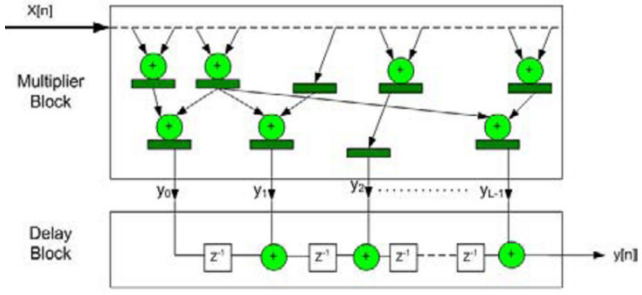

Figure.3. Multiplier Block Using Add And Shift

Canonical representation of a signed digit (CSD) is a method used to reduce cost by representing a signed number using the least number of non-zero digits, thereby reducing the number of multiply operations[2] .CSD is a radix 2 signed-digit representation of coefficients for which the permissible digit set is {-1, 0, 1}. Thus CSD representation permits subs traction, as well as addition, of shifted data in accomplishing multiplication [6]. However, due to the constraint imposed on CSD coefficient [9], the quantization level of the coefficients is large, which causes performance degradation. But, if we apply optimization techniques in the integer domain, we can obtain a CSD FIR filter with less degradation [10].

3.4 Filter Coefficients

This design uses FDA tool [5] in Mat lab to calculate the filter coefficients. Set up sampling frequency, pass-band frequency and cut-off frequency as 4000 kHz, 100 Hz, and 150 Hz separately. Pass-band amplitude and stop-band amplitude are 1dB and 80dB, the low-pass filter designed[13].. In order to improve the filter speed, using shift register module is to transform a serial input into a parallel output [5].

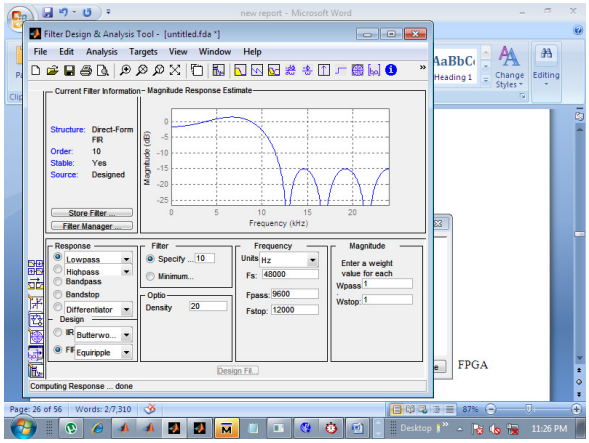

Figure.4. FDA tool window

IV. SIMULATION RESULTS

One of the key objectives is to learn how to implement actual design problem in hardware. The following steps will depict the procedure exactly as done by the author by taking a relevant example:

1. Define the filter specifications such as order, window function, cut-off frequency and sampling frequency.
2. Calculate the filter coefficient using MATLAB FDA Tool.
3. Apply CSD Algorithm on filter Coefficients.
4. Configured the target FPGA Spartan 3E kit for real time debugging.
5. Checked the CSD Filter coefficients on target FPGA Chip.

4.1 Modelsim Output:

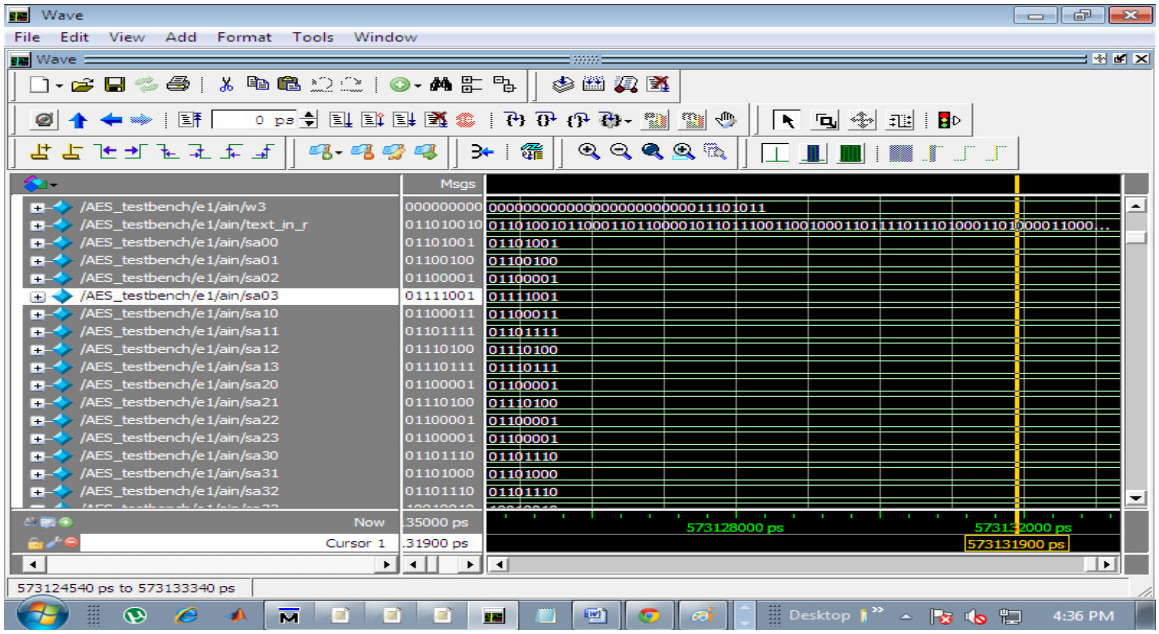

Figure 5. Simulated Output

4.2 Area Utilization Report:

Flow Summary	
Flow Status	Successful - Wed Jul 04 23:14:13 2012
Quartus II Version	11.0 Build 208 07/03/2011 SP 1 SJ Web Edition
Revision Name	medtop
Top-level Entity Name	fir_filter
Family	Cyclone III
Device	EP3C16F484C6
Timing Models	Final
Total logic elements	4,451 / 15,408 (29 %)
Total combinational functions	3,979 / 15,408 (26 %)
Dedicated logic registers	529 / 15,408 (3 %)
Total registers	529
Total pins	55 / 347 (16 %)
Total virtual pins	0
Total memory bits	0 / 516,096 (0 %)
Embedded Multiplier 9-bit elements	0 / 112 (0 %)
Total PLLs	0 / 4 (0 %)

Figure.6. Area utilization Report

4.3 Performance Report:

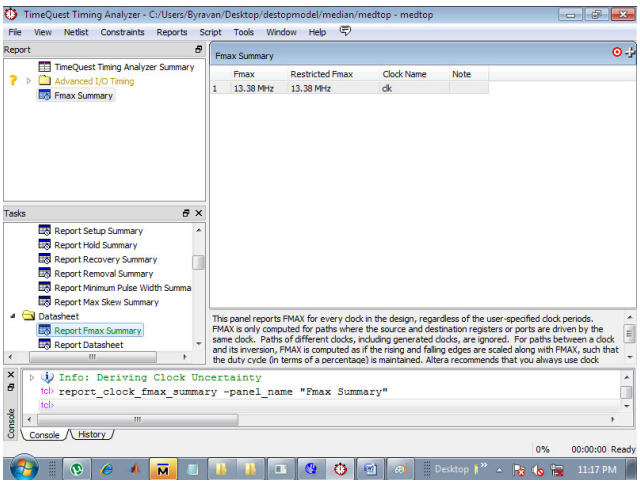

Figure.7. Fmax.Summary Report of Slow Corner.

4.4 Synthesis Report:

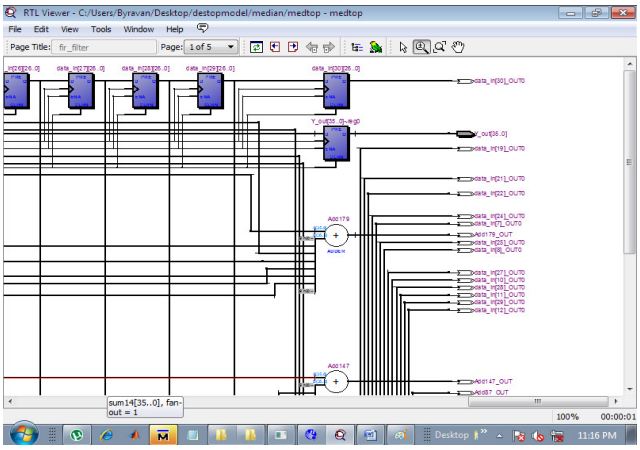

Figure.8. RTL Schematic Report.

4.5 Map Viewer:

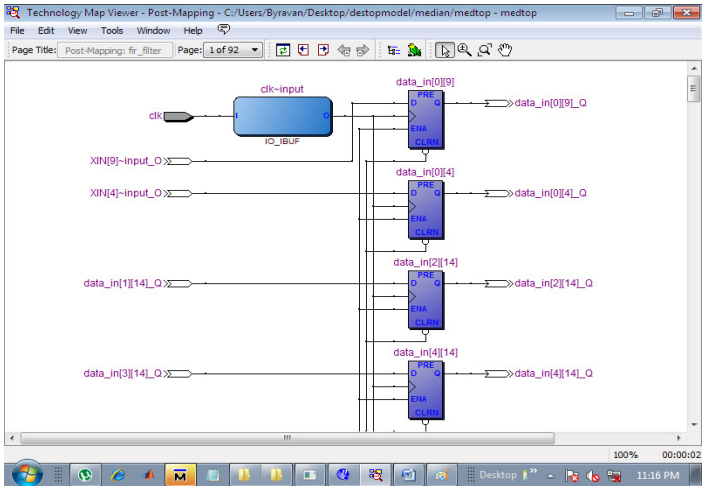

Figure 9. Technology Map Viewer

Table.1. Comparison of Resources Usage and the data rate

Parameter	Existing fir filter	Proposed system
Number of logical elements used	4962	4551
Number of combinational elements used	4487	3979
Power consumed	69.35mW	69.33mW
Data rate	778MBPS	820MBPS

V. CONCLUSION

By researching the distributed algorithm and its several structures, a design method of 60-order low-pass FIR filter based on parallel structure of DA and pipelining technology is not an efficient one when the filter order is increased. But CSD based FIR filter which use FPGA as the hardware platform and only shifters and adders are used .Simulation results show that this filter system has a good performance, the filter speed is higher and the resource occupation is fewer.

VI. FUTURE SCOPE

Compared with the traditional filters, this filter is more popular in digital signal processing. This work can be extended by increasing the order of filter for accurate outputs i.e., in some applications like medicine more accurate filters are used.

REFERENCES

[1] "A New Algorithm for Elimination of Common Sub expressions" R. Pasco, P.Schaumont, V. Denuder, S. Ferndale, *Member, IEEE*, and D. D Urac˘Kova IEEE Transactions on Computer Aided and Integrated Circuit's Vol.18 No.1 January 1999.
[2] "FPGA Implementation Of High Speed FIR Filters Using Add And Shift Method" IEEE Transactions On Communications 2006 Shahnam Mirzaei, Anup Hosangadi, Ryan Kastner University Of California, Santa Barbara.
[3] "Common Sub-Expression Elimination" March 23, 2007 RESEARCH CENTRE FOR INTEGRATED MICROSYSTEMS UNIVERSITY OF WINDSOR Payman Samadi.
[4]. "FIR Filter Implementation By Systolization Using DA-Based Decomposition" Proc. Of Int. Conf. On Advances In Computer Science 2010 Shreeharsha K G, Rekha Bhandarkar Rekha Bhandarkar,
[5] "FPGA BASED FIR FILTER" International Journal of Engineering Science and Technology Vol. 2 (12), 2010 SUVARNA JOSHI1 Computer Engg. Department, Pune, BHARATI AINAPURE, Pune.
[6] "Low Power And Fast DCT Architecture Using Multiplier-Less Method" M. El Aakif, S. Belkouch, N. Chabini, M. M. Hassani
Embedded Systems And Digital Control Laboratory, National School Of Applied Sciences, University Of Cadi Ayyad, Av. Abdelkarim Khattabi, Guéliz, Marrakech, Morocco 2011 FTFC.
[7] "A Novel Low Power, Low Area Array Multiplier Design for DSP Applications" 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN 2011) N.Ravi, Y.Subbaiah, Dr.T.Jayachandra Prasad, Dr.T.Subba Rao,
[8] "Design And Simulation Of 60-Order Filter Based On FPGA" 2011 Third International Conference On Intelligent Human-Machine Systems And Cybernetics Jinming Li, Min Zhao, Tingling Wang,National Key Lab For Electronic Measurement And Technology Key Laboratory Of Instrumentation Science & Dynamic Measurement North University Of China.
[9]. "A New Algorithm For FIR Digital Filter Synthesis For A Set Of Fixed Coefficients" J. L. Mazher Iqbal S. Varadarajan*, Sri Venkateswara University Tirupati, India* European Journal Of Scientific Research ISSN 1450-216X Vol.59 No.1 (2011), Pp.104-114INDIA.
[10] "Design And Implementation Of Low Power Digital FIR Filter Based On Low Power Multipliers And Adders On Xilinx FPGA" IEEE Transactions' On 2011 Bahram Rashidi Bahman Rashidi Majid Pourormazd University.
[11]"FPGA based Design and Implementation of Higher Order FIR Filter using Improved DA Algorithm" Purna Ramesh ,G.Nagarjuna,G.Siva Raam,A.P, India *International Journal of Computer Applications (0975 – 8887) Volume 35– No.9, December* ***2011***.
[12]. **"**Distributed Arithmetic for FIR Filter implementation on FPGA" Shi School of Automation, Hangzhou Dianzi University
HangZhou, ZheJiang, China. 2011 IEEE.

[13] http://www.mathworks.com/help/toolbox/signal/fdatool.html

AUTHORS

B. Mamatha completed her Intermediate in Sri Kakatiya junior College, received her bachelor degree in engineering ECE from Sri Sarathi institute of Engineering & Technology and presently doing her Master degree in Embedded Systems from JNTU Kakinada University in Pragathi Engineering College, Surampalem.

V.V.S.V.S. Ramachandram received his bachelor degree in ECE engineering from JNTU Hyderabad, and his master degree in Embedded Systems from Vinayaka University. He is an Associate Professor of ECE in Pragathi Engineering College, Surampalem.