IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 6, Issue 4, Ver. II (Jul. - Aug. 2016), PP 40-49
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Design and Realization of practical FIR filter based on CSD and DA algorithms

R Thaviti raju[1], K Chitambararao[2], T Viswanadham[3], B. Chinnarao[4]
[1,]*M.TechScholar,Department.of ECE,AITAM*
[2,]*Assoc.Professor,Department.of ECE,AITAM*
[3,] *Asst.Professor, Department.of ECE, AITAM,* [4,]*Assoc.Professor,Department.of ECE,AITAM*

Abstract: *FIR digital filters find immense applications in mobile communications systems such as channel equalization, channelization, matched filtering and pulse shaping, due to their absolute stability and linear phase properties. In this paper it is proposed to design a practical FIR filter using MATLAB tool to obtain the response. After that the filter will be designed and analyzed based on canonical signed digits and compared with the distributed arithmetic algorithm in order to minimize the power consumption and fast implementation of the filter. The design filter will be simulated and synthesized using Xilinx ISE 13.1 software.*
Keywords: *FIR, CSD, DA, VERILOG HDL*

I. Introduction

Recently many technologies have come out in those technologies electronic technology plays a vital role with development of marvelous speed. As of late, digital signal processing (DSP) is utilized as a part of a considerable measure of uses, for example, computerized set-top box, acoustic pillar formers, computerized adaptable plate, versatile video frameworks/PCs, advanced sound, computerized radio, media and remote interchanges, advanced still and system cameras, discourse handling, transmission frameworks, video pressure, link modems, radar imaging, worldwide situating frameworks, and biomedical sign preparing. The field of DSP has dependably been oversee by the advances in DSP applications and in scaled very-large-scale-integrated (VLSI) innovations. Filtering procedure is utilized as a part of the sign handling area to achieve a coveted recurrence band from a framework as the o/p by giving some contribution to it. The framework that is utilized for the separating operation is known as the channel. In DSP, there are essentially two sorts of channel, IIR and fir channel. The motivation reaction of the IIR channel is of unbounded span where as it is of limited term if there should arise an occurrence of fir channel .The fir channel requires no criticism way and along these lines it has no recursion and subsequently the fir channel is non_recursive. Fir channels detail incorporate greatest middle of the road stop band swell, pass band and stop band edge recurrence. The coefficients of fir channel requires impressive measure of figurings. Along these lines it is by and large performed by utilizing different PC supported configuration apparatuses, for example, channel outline and examination device of MATLAB. So for a continuous applications, for example, separating, combinational multipliers are utilized in view of fast .the vast majority of the equipment many-sided quality is because of multipliers, as channels require expansive no of augmentation, prompting extreme region, postpone and control utilization regardless of the fact that executed in a full exceptionally coordinated circuits now the issue confronted is that how lessen the equipment multifaceted nature of a multiplier. The principle anxiety is on the lessening of multipliers in fir channel the real impediment of higher request need. The higher request forces more equipment necessities, number-crunching operations, territory use and power utilization when outlining the channel. Accordingly, minimizing or diminishing these parameters, is significant objective in advanced channel outline assignment. It is yearning to discover productive calculation that require as couple of math operations as could reasonably be expected, as this in the zone and minimizes the gadget size and vitality utilization. To evacuate the repetitive calculation which prompts more proficient calculations the procedures picked are CSD, DA. This is utilized to streamline the region of high pass fir channel. In CSD structure the channel coefficients are settled. In CSD structure multiplier region get lessened. DA is essentially somewhat serial computational operation that structures and inward result of a couple of vectors in a one direct stride the upside of DA is its productivity of mechanization.

II. Existing Methodology

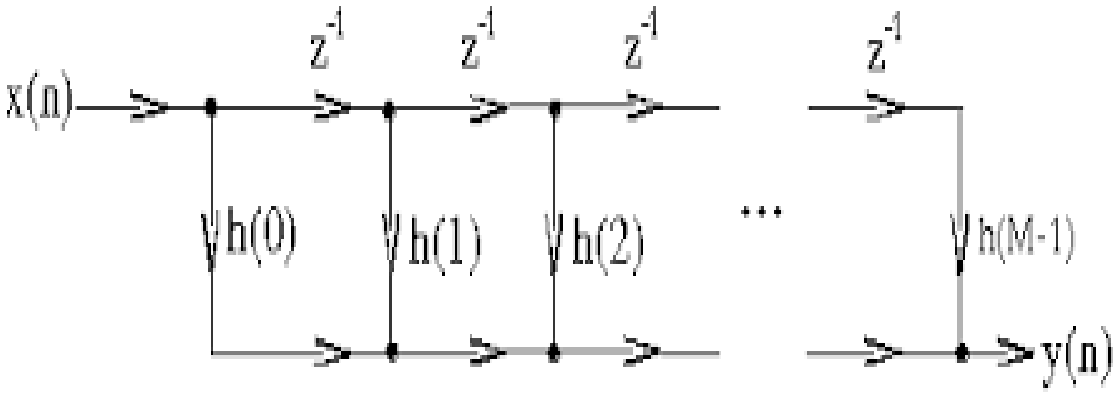


Fig1 Direct form of FIR filter structure

Till now FIR filters are designed and implemented in vlsi domain with out taking real coefficients .Our proposed method is to represent the FIR filters with real coefficient and implement the same in VLSI.Apart from the coefficients will be obtained by using MATLAB command window.

III. Proposed Methodology

Block Diagram:

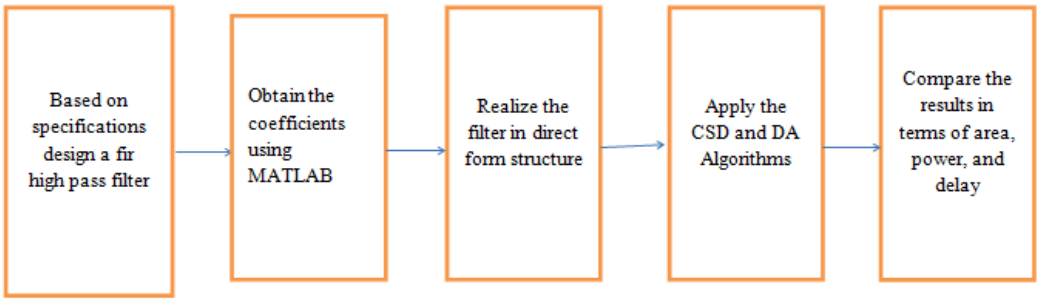
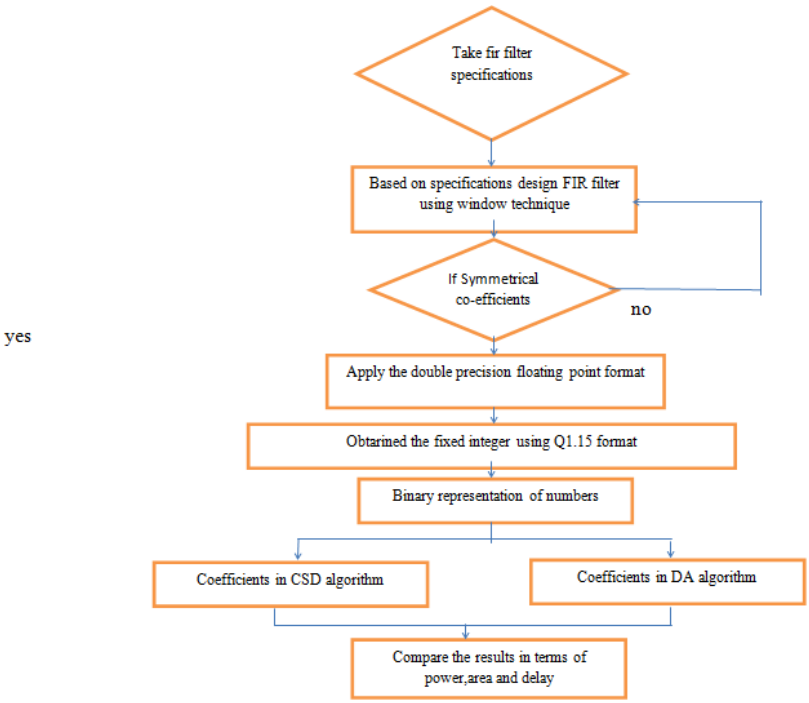


Fig2 Practical FIR filter

Flow Chart:



Realization:

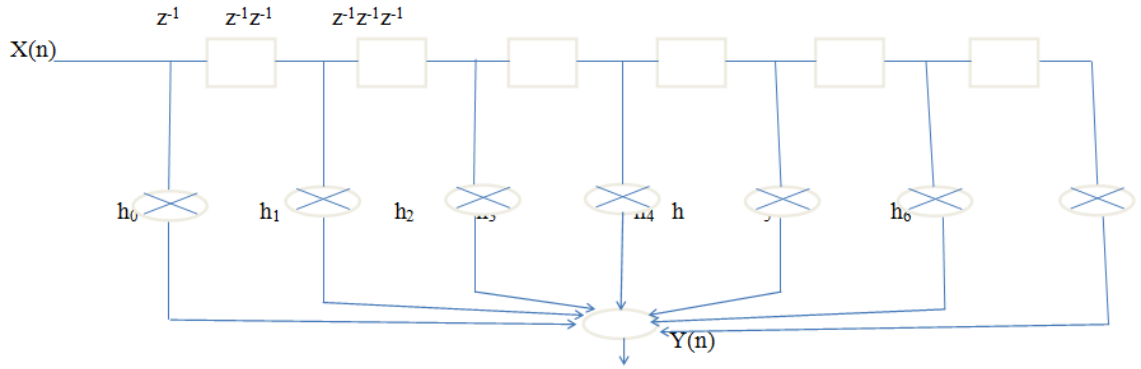


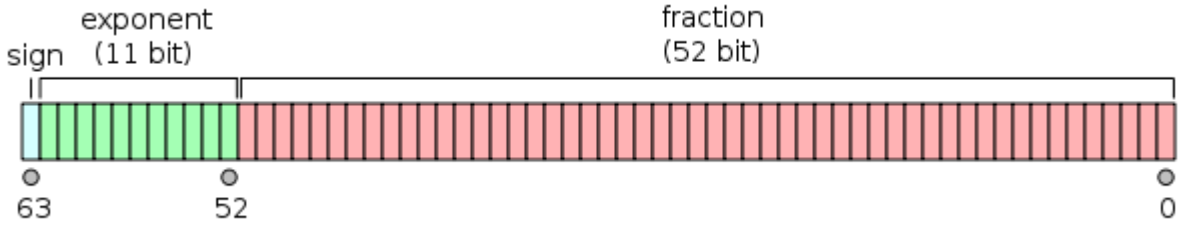
Fig3 Realization filter

Floating Point Format:

Floating point representation works well for numbers with large dynamic range based on the no.of bits. This standard is almost exclusively used across computing platforms and hardware designs that support floating point arithmetic. In this standard a normalized floating point number x is stored in three parts: the sign s , the excess exponent e ,and the significand or mantissa m, and the value of the number in terms of these parts is:

$$x = (-1)^e * 1 * m * 2^{e^{-b}}$$

The format is written with the significand having an indirect integer bit of value 1 (except for special data, see the exponent encoding below). With the 52 bits of the fraction significand become visible in the memory format, the total precision is therefore 53 bits. The bits are laid out as follows:



From the matlab command window the real filter coefficients are 0.0038 ,-0.035, -0.2278,0.610,0.0037,-.0331,-0.2291.these filter coefficients are converted to double precision floating point number.the converted coefficients are 0.003775018138711,-0.033541428579110,-0.227792539932163,0.618026583319437,-0. 229088808118480,-0.033110787270075,0.003737449345339.

i) Q-FORMAT:

Q is a fixed point format where the number of fractional bits (and optionally the number of integer bits) is specified. For example, a Q15 number has 15 fractional bits: a Q1.15 number has 1 integer bit and 14 fractional bits. Q format is commonly used in hardware that does not have a floating-point unit and in applications that require constant resolution.

The $Q_{n,m}$ format of an N bit number sets n bits to the left and m bits to the right of the binary point. In case of signed numbers, the MSB is used for the sign and has negative weight. A two's complement fixed point number in $Q_{n,m}$ format is equivalent to $b=b_{n-1}b_{n-2}b_{n-3}b_{n-4}\ldots\ldots b_2b_1b_0b_{-1}\ldots\ldots b_{-m}$ with equivalent floating point value :$-b_{n-1}2^{n-1}+b_{n-2}2^{n-2}+\ldots\ldots+b_12^1+b_0+b_{-1}2^{-1}+\ldots\ldots b_{-m}2^{-m}$.

A floating point number format is simply converted to $Q_{n,m}$ fixed point format by bringing m fractional bits of the number to the integer part and then dropping the rest of the bits with or without rounding. This conversion translates a floating point number to an integer number with an implied decimal the implied decimal needs to be remembered by the designer for referral in further processing of the number in different calculations:

$$Num\_fixed = round(num\_float*2^m)$$

Or

$$Num\_fixed = fix(num\_float*2^m)$$

The coefficients are converted to double precision floating number into fixed point format.the coefficients are in decimal numbers that are 124,-1099,-7464,20252,-7507,-1085,123.these decimal numbers are converted to binary then hexadecimal number.

IV. Csd Algorithm

The CSD code is a ternary number system with the digit set {1¯ 0 1}, where 1¯ stands for 1. Given a constant, the corresponding CSD representation is unique. CSD representation of a number can be recursively computed using the string property and has two main properties:

(1)The number of nonzero digits is minimal

(2) No two consecutive digits are both nonzero, that is, two nonzero digits are not adjacent.

The first property implies a minimal Hamming weight, which leads to a reduction in the number of additions in arithmetic operations. The second property provides its uniqueness characteristic. However, if this property is relaxed, this representation is called the minimal signed digit (MSD) representation, which has as many,nonzero as the CSD representation, but which provides multiple representations for a constant. It enables the reduction of the number of partial products that must be calculated fast and also low-power consumption and low area structure of a multiplier for DSP applications or self-timed circuits. From the practical point of view, the traditional approach to generate the CSD representation. All of these algorithms generate the CSD code recursively from the least significant bit (LSB) to the most significant bit (MSB).

The CSD representation of an integer number is assigned and unique digit representation that contains no adjacent non zero digits. Given an n-digit binary unsigned number X={x0, x1, - - - - - - - , x_{n-1}} expressed as

$$X = \sum_{n=0}^{n-1} x_i.2^i, \quad x_i \in \{0,1\} \qquad 1$$

Then the (n+1)-digit CSD representation Y={y0, y1,_ _ _ _ _ _ _ _ y_n } of X is given by

$$Y = \sum_{i=0}^{n-1} x_i.2^i = \sum_{i=0}^{n} y_i.2^i, \quad y_i \in \{\bar{1},0,1\} \qquad 2$$

The condition that all non-zero digits in a CSD number are separated by zero implies that

$$y_{i+1}.y_i = 0, \qquad 0 \le i \le n-1 \qquad 3$$

From this property, the probability that a CSD n-digit has a non-zerovalue is given by

$$P(|y_i| = 1) = 1/3 + 1/9n[1-(-1/2)^n] \qquad 4$$

As n becomes large, this probability tends to 1/3 while this probability becomes ½ in a binary code.Using this property, the number of additions/subtractions is reduced tominimum in multipliers and as a result, an overall speed –up can be achieved. Encoding 2 is prefer- able since it satisfies the following relation.

$$y_i = y_i^d - y_i^s \qquad 5$$

Where y_i^s represents the sign bit and y_i^d the data bit.This encoding also allows an additional valid representation of 0 when y_i^s =1 and y_i^d =1, which is useful in some arithmetic implementations. In the whole paper, this encoding is used.

CSD representation for binary form:

h(0)=124 =0000 0000 0111 1100=0000 0000 1000 0$\bar{1}$00(CSD form)
h(1)=1099=0000 0100 0100 1011=0000 0100 0101 0$\bar{1}$0$\bar{1}$(CSD form)
h(2)=7464=0001 1101 0010 1000=0010 0$\bar{1}$01 0010 1000(CSD form)
h(3)=20252=0100 1111 0001 1011=0101 000$\bar{1}$ 0010 0$\bar{1}$0$\bar{1}$(CSD form)
h(4)=7507=0001 1101 0101 0011=0010 0$\bar{1}$01 0101 010$\bar{1}$(CSD form)
h(5)=1085=0000 0100 0011 1101=0000 0100 0100 0$\bar{1}$01(CSD form)
h(6)=123 =0000 0000 0111 1011=0000 0000 1000 0$\bar{1}$0$\bar{1}$(CSD form)

Xn(0)*h(0)+xn(1)*h(1)+xn(2)*h(2)+xn(3)*h(3)+xn(4)*h(2)+Xn(5)*h(1)+xn(6)*h(0)

V. Distributive Algorithm

DA is another way of implementing a dot product where one of the arrays has constant elements. The da can be effectively used to implement FIR, IIR and FFT type algorithm.In the case of an FIR filter, the coefficients constitute an array ofconstants in some signed Q format where the tapped delay line forms the array

of variables whichchanges every sample clock. The DA logic replaces the MAC operation of convolution summation of into a bit serial look up table read and addition operation. The architecture of FPGAs, time/area effective designs can be implemented using DA techniques. The DA logic works by first expanding the array of variable numbers in the dot product as a binary number and then rearranging MAC terms with respect to weights of the bits. Let the different elements of arrays of constants and variables are$A_K$and$X_k$respectively. Thelength of both the arrays is K. Then their dot product can be written as:

$$y = \sum_{k=0}^{k-1} A_k x_k$$

Let us assume x_k is an N bit Q1.(N-1) format number:

$$x_k = x_{k0} 2^0 + \sum_{b=1}^{N-1} x_{kb} 2^{-b}$$

The dot product can be written as:

$$y = \sum_{k=0}^{k-1} (-x_{k0} 2^0 + \sum_{b=1}^{N-1} x_{kb} 2^{-b}) A_k$$

$$y = \sum_{k=0}^{k-1} (-x_{k0} 2^0 + x_{k1} 2^{-1} + ... + x_{k(N-1)} 2^{-(N-1)}) A_k$$

Rearranging the terms yields:

$$y = -\sum_{k=0}^{k-1} x_{k0} A_k 2^0 + \sum_{b=1}^{N-1} 2^{-b} \sum_{k=0}^{k-1} x_{kb} A_k$$

For k=3 and N=4, the rearrangement forms the following entries in the ROM:

$$-(x_{00}A_0 + x_{10}A_1 + x_{20}A_2)2^0$$
$$+(x_{01}A_0 + x_{11}A_1 + x_{21}A_2)2^{-1}$$
$$+(x_{02}A_0 + x_{12}A_1 + x_{22}A_2)2^{-2}$$
$$+(x_{03}A_0 + x_{13}A_1 + x_{23}A_2)2^{-3}$$

The DA technique pre computes all possible values of $\sum_{k=0}^{k-1} x_{kb} A_k$

The ROM is P bits wide and 2^k deep and implements a look up table. The value of P is:

$$P = \left\lceil \log_2 \sum_{k=0}^{k-1} |A_k| \right\rceil + 1$$

X_{2b}	X_{1b}	X_{0b}	Contents of ROM
0	0	0	0
0	0	1	A_0
0	1	0	A_1
0	1	1	A_1+A_0
1	0	0	A_2
1	0	1	A_2+A_0
1	1	0	A_2+A_1
1	1	1	$A_2+A_1+A_0$

Table1: ROM for Distributed Arithmetic

All the elements of the vector are stored in the Shift Register. The architecture considers in each cycle the b^{th} bit of all the elements and concatenates them to from the address to the ROM. From the MSB_s value in the ROM is subtracted from a running accumulator, and for the rest of the bit locations values from ROM are added in the accumulator. The size of the accumulator is set to P+N. where a P bit adder adds the current output of the ROM in the accumulator and N bits of the accumulator are kept to the right side to cater for the shift operation

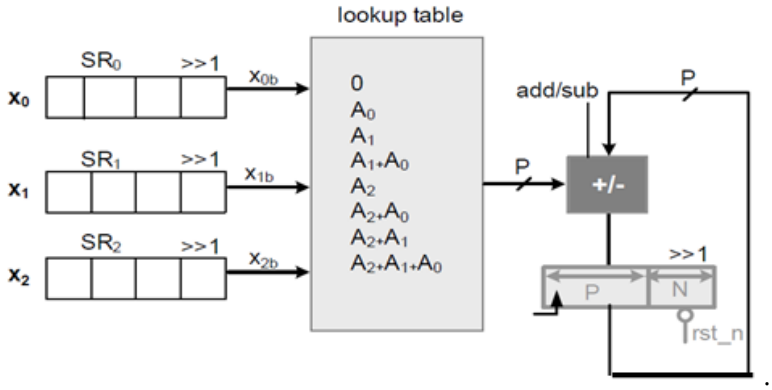


Fig4DA for computing the dot product of integer numbers

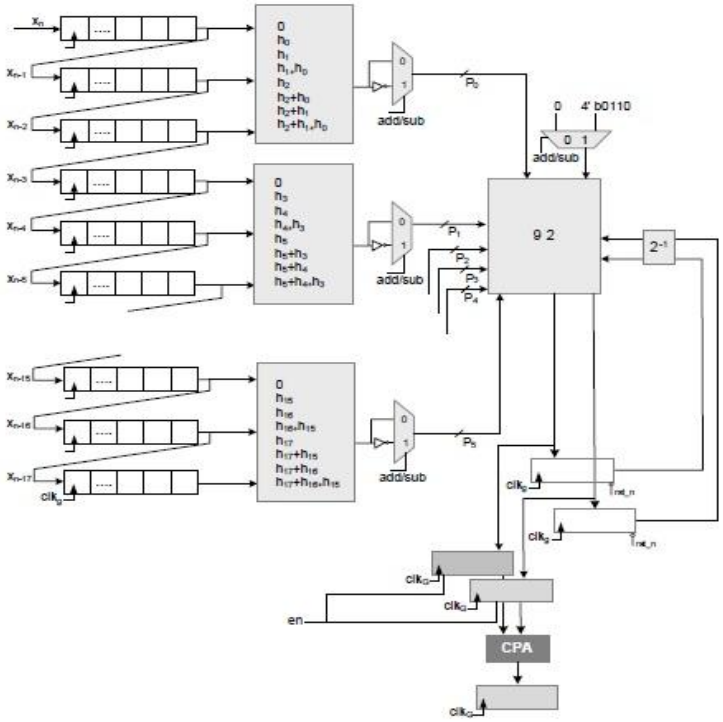


Fig5 DA based parallel implementation of an 18 co-efficient FIR filter

It is obvious from the configuration of a DA based design that the size of ROM increases with an increase in the number of coefficients of the filter. This size is prohibitively large and DA techniques are used to reduce the ROM requirement.

VI. Simulation Results

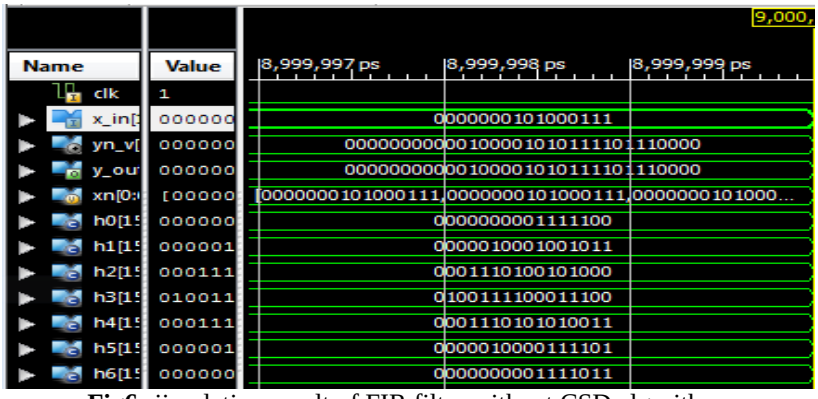


Fig6 siimulation result of FIR filter without CSD algorithm

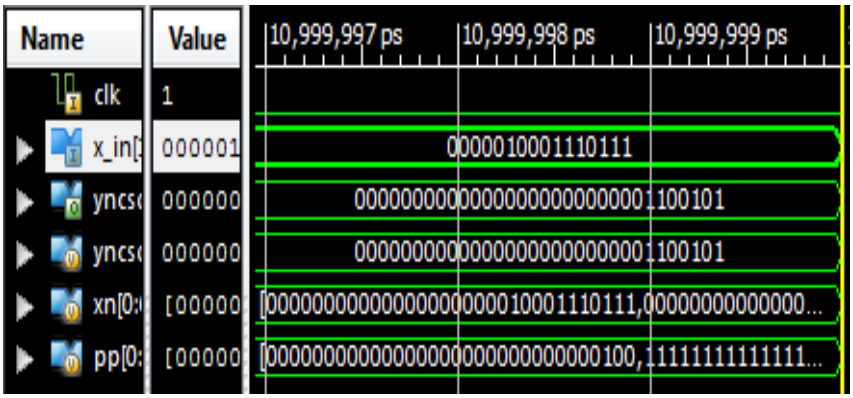


Fig7 siimulation result of FIR filter with CSD algorithm

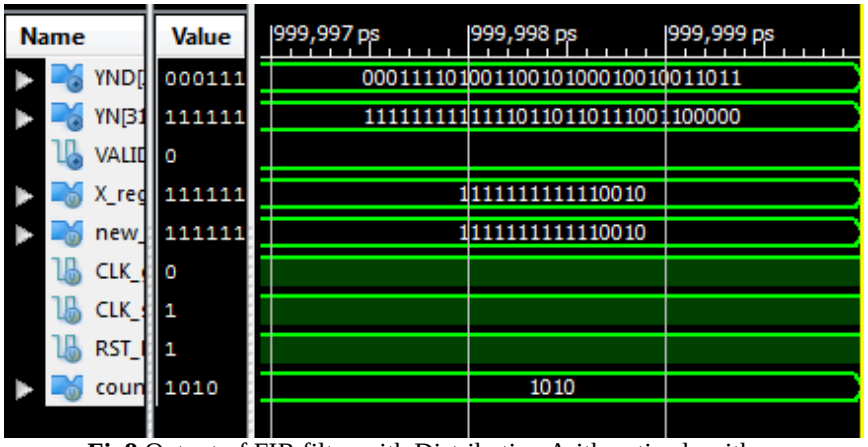


Fig8 Output of FIR filter with Distributive Arithmatic algorithm

VII. Synthesis Report

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	128	9,312	1%
Number of 4 input LUTs	188	9,312	2%
Number of occupied Slices	143	4,656	3%
Number of Slices containing only related logic	143	143	100%
Number of Slices containing unrelated logic	0	143	0%
Total Number of 4 input LUTs	188	9,312	2%

Fig9 Synthesis report of FIR filter without CSD algorithm

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	159	9,312	1%
Number of 4 input LUTs	1,406	9,312	15%
Number of occupied Slices	784	4,656	16%
Number of Slices containing only related logic	784	784	100%
Number of Slices containing unrelated logic	0	784	0%
Total Number of 4 input LUTs	1,413	9,312	15%

Fig10 Synthesis report of FIR filter with CSD algorithm

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	86	9,312	1%
Number of 4 input LUTs	80	9,312	1%
Number of occupied Slices	57	4,656	1%
Number of Slices containing only related logic	57	57	100%
Number of Slices containing unrelated logic	0	57	0%
Total Number of 4 input LUTs	80	9,312	1%

Fig11 Synthesis report of FIR filter with Distribuive arithmetic algorithm

VIII. Power Analysis

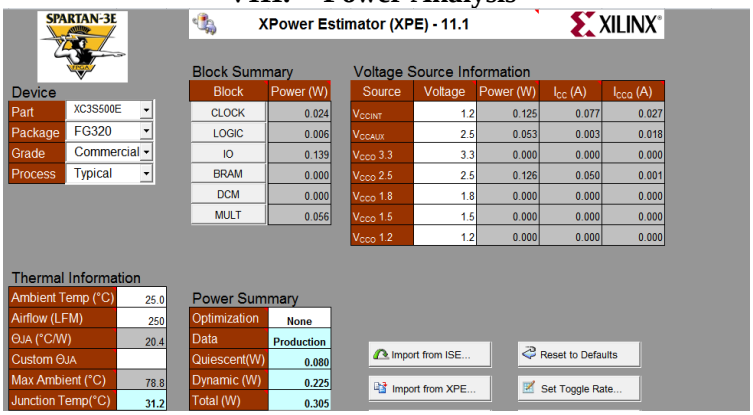


Fig 12 Power estimator of FIR filter without CSD algorithm

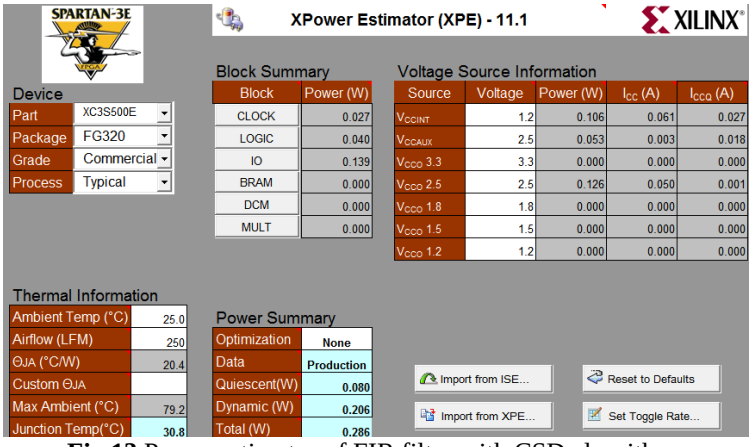


Fig 13 Power estimator of FIR filter with CSD algorithm

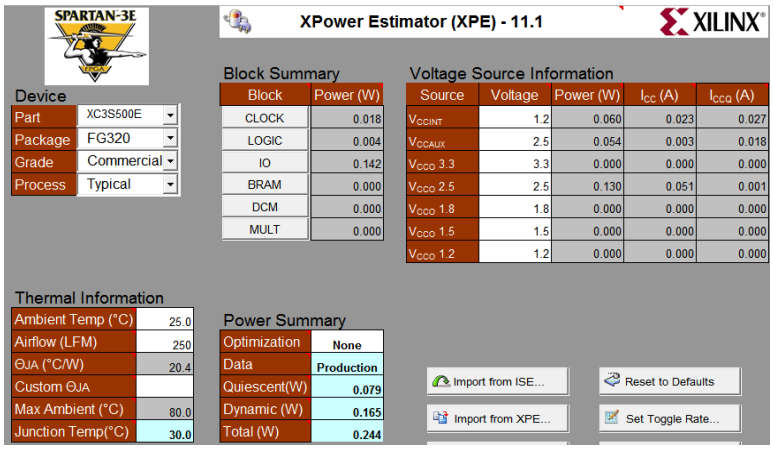


Fig 14 Power estimator of FIR filter Distribuitivearithmatic algorithm

Parameter	FIR	FIR WITH CSD	FIR WITH DA
Area(slices)	128	159	86
Area(LUT's)	188	1406	80
Delay(ns)	18.14	41.68	7.58
Power(mw)	305	286	244
Power*delay	5532.7	11920.48	1849.52

Table2. Comparision of results interms of power,area and delay

The static power can directly obtained from synthesis results but Xilinx software can't provide dynamic power information. For this dynamic power analysis the Xilinx provides plugin support such as XPower Estimator (XPE). XPower Estimator-11.1 is Microsoft Excel spread book, which provides detailed power analysis by using mapping report file generated during the synthesis using Xilinx synthesizer. The dynamic power is 165mW and static power is 79mW. The total power is 244mW whereas the total power consumption for DA based FIRfilter is 244mW.

IX. Conclusion

The FIR filters are extensively used in digital signal processing and can be implemented using programmable digital processors. With the advancement in VLSI technology as the DSP has become increasingly popular over the years, the high speed realization of FIR filter with his power consumption has become much more demanding..In this paer, FIR high pass filter is designed by using hamming window and obtain the coefficients using MATLAB. Moreover the CSD and DA algorithms are applied and compared the results interms of area,power and delay.

References

[1]. A. Avizienis, "Signed digit number representation for fast parallel arithmetic," IRE Transactions on Electronic Computers, 1961, vol. 10, pp. 389 400.

[2]. R. Hashemian, "A new method for conversion of a 2's complement to canonic signed digit number system and its representation," in Proceedings of 30th IEEE Asilomar Conference on Signals, Systems and Computers, 1996, pp. 904 907.

[3]. H. H. Loomisand B. Sinha, "High speed recursive digital filter realization," Circuits, Systems and Signal Processing, 1984, vol. 3, pp. 267 294.

[4]. K. K. Parhi and D. G. Messerschmitt, "Pipeline interleaving parallelism in recursive digital filters. Pt I: Pipelining using look ahead and decomposition," IEEE Transactions on Acoustics, Speech Signal Processing, 1989, vol. 37, pp. 1099 1117.

[5]. K. K. Parhi and D. G. Messerschmitt, "Pipeline interleaving and parallelism in recursive digital filters. Pt II: Pipelining incremental block filtering," IEEE Transactions on Acoustics, Speech Signal Processing, 1989, vol. 37, pp. 1118 1134.

[6]. A. V. Oppenheim and R. W. Schafer, Discrete time Signal Processing, 3rd, 2009, Prentice Hall.

[7]. Y. C. Lim and S. R. Parker, "FIR filter design over a discrete powers of two coefficient space," IEEE Transactions on Acoustics, Speech Signal Processing, 1983, vol. 31, pp. 583 691.

[8]. H. Samueli, "An improved search algorithm for the design of multiplier less FIR filters with powers of two coefficients," IEEE Transactions on Circuits and Systems, 1989, vol. 36, pp. 1044 1047.

[9]. J. H. Han and I. C. Park, "FIR filter synthesis considering multiple adder graphs for a coefficient," IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, 2008, vol. 27, pp. 958 962.

[10]. A. G. Dempster. and M. D. Macleod, "Use of minimum adder multiplier blocks in FIR digital filters," IEEE Transactions on Circuits and Systems II, 1995, vol. 42, pp. 569 577.

[11]. R. I. Hartley, "Sub expression sharing in filters using canonic signed digit multipliers," IEEE Transactions onCircuits and Systems II, 1996, vol. 43, pp. 677 688.

[12]. Y. Jang. and S. Yang, "Low power CSD linear phase FIR filter structure using vertical common sub expression," Electronics Letters, 2002, vol. 38, pp. 777 779.

[13]. A. P. Vinod, E. M. K. Lai, A. B. Premkuntar and C. T. Lau, "FIR filter implementation by efficient sharing of horizontal and vertical sub expressions," Electronics Letters, 2003, vol. 39, pp. 251 253.

[14]. A. Hosnagadi, F. Fallah and R. Kastner, "Common sub expression elimination involving multiple variables for linear DSP synthesis," in Proceedings of 15th IEEE International Conference on Application specific Systems, Architectures and Processors, Washington 2004, pp. 202 212.

[15]. P. K. Meher, S. Chandrasekaran and A. Amira, "FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic," IEEE Transactions on Signal Processing, 2008, vol. 56, pp. 3009 3017.

AUTHOR PROFILE:

Mr. R.Thavitirajuis presently pursuing his M.Tech in VLSI system design in Electronics and Communication Engineering Department, AITAM, Tekkali. His areas of interest are Low Power VLSI system design and digital filter optimization. He has attended for one national level workshop. He is membership in GSM IEE. The author may be reached at rtraju1577@gmail.com.

Mr. K. ChitambaraRaois presently working as Associate Professor in Electronics and Communication Engineering Department, AITAM, Tekkali. He completed M.Tech from Sathyabhama University in the specialization of VLSIDesign.and registered in Ph.D in Andhara university (his research area antennas and wave propagation). He has 12years' experience in teaching and research. He published more than 11 research papers in National/ International Journals and Conferences. He is a life member of ISTE, IETE.

Mr. **T. Viswanadham**is presently working as Assistant Professor in Electronics and Communication Engineering Department, AITAM, Tekkali. He completed M.Tech from JNT University in the specialization of VLSI-System Design.andregistered in Ph.D in Andharauniversity(interested research areas are Biomedical signal processing and VLSI). He has 12years' experience in teaching and research. He published 8 research papers in National/ International Journals and Conferences.He is a life member of SEMCE(I).

Mr. B. Chinnaraois presently working as Associate Professor in Electronics and Communication Engineering Department, AITAM, Tekkali. He completed M.Tech from JNTU, Hyd.in the specialization of signal processing and registered in Ph.D in JNTU, Hyd. (his research area image processing). He has 15years' experience in teaching and research. He published more than 30 research papers in National/ International Journals and Conferences. He is a life member of ISTE and member in IEEE.