IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 6, Issue 6, Ver. II (Nov. - Dec. 2016), PP 21-28
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Design and Realization of PracticalFIR Filter Using Hybrid Window and CSD Algorithm

K. Chitambararao[1], E.Jaya[2],T.Viswanadham[3],L.Rambau[4]
[1]*Associate Professor, Department of ECE, AITAM, Tekkali, India*
[2]Assistant *Professor,Department of ECE,AITAM,Tekkali, India*
[3]Assistant *Professor,Department of ECE,AITAM,Tekkali, India*
[4]*Assistant Professor,Department of ECE,AITAM,Tekkali, India*

Abstract: *Multiplier-less FIR filter design is the major requirement in VLSI signal processing. In this work the practical FIR low pass filter is designed using hybrid window for various mathematical operations like addition, average, exclusive-or and multiplication. Their frequency responses are obtainedby using Matlab. Multiplication based design of FIR Low Pass Filter (LPF) is realized in direct form structure and implemented in VLSI. This structure consists of adders, multipliers and delay elements. In VLSI multipliers consumpes more powerso that for multiplication of input with coefficients, the Canonical Signed Digit (CSD) algorithm is applied for multiplication process to reduce the power consumption. Apart fromthisthe simulation, synthesis, delay and power reports are analyzed by using Xilinx 13.1 ISE and XPower Estimator 11.1*
Index terms*: CSD algorithm,Double precision format, FIR filter, Hybrid window,Q format.*

I. Introduction

A FIRfilter is a Linear Time Invariant (LTI) system, which exhibits linear phase and inherent stabilityhas been seen in many applications such as equalization of Communication channels, signal detection in radar, sonar, pulse shaping etc.,[1,2,3,4]. To avoid the effect of Gibbs phenomenon, windowing method is used to design FIR filters. Rectangular, Bartlett, Hamming, Hanning, Blackman and Kaiser are the different windowing techniques. Out of all these Blackman window is superior in side lobe level reduction and Kaiser window is best in less transition width[1, 3, 4, 9]. So, the pros of these two windows are taken to design a new hybrid window[1]in this paper. Hybrid window is a new concept which is obtained by the combination of two different windows with different mathematical operations. The characteristics of FIR low pass filter, the windowing technique and the required equations for the hybrid windows are explained further. We used MATLAB tool for implementation of the filters. The output for the frequency response of FIR low pass filter using hybrid window are shown further. Tabular forms for Relative side lobe attenuation are shown further.The VLSI implementation of a linear phase Finite impulse response (FIR) filter with fixed coefficients, large amount of powersare required for multiplications when compared to additions and subtractions involved. The number ofadditions/subtractions used for coefficient multiplications increases the complexity of FIR filters[2]. The CSDcodes minimize the number of adders/subtracters required in each coefficient multiplications and thus reduces the complexity of FIR filters and power required[5 to 8, 10,11].

II. Background

2.1 FIR filter Design Using Hybrid Window Functions

The hybrid window is formed by combining two window functions by some mathematical operations like addition, multiplication, averaging, Exclusive-or[1].
The windows which takenhere are Blackman window and Kaiser Window

- **Blackman window:**

wb(n)=0.42+0.5cos(2*pi*n/N-1)+0.08cos(4*pi*n/N-1)
for |n|≤-(N-1)/2
=0 otherwise (1)

- **Kaiser window:**

wk(n)=Io[alpha(1-(2*n/N-1)2)0.5]/I0[alpha]
for |n|≤-(N-1)/2
=0 otherwise (2)
Where, alpha is adjustable parameter
Io(x)=zeroth- order Bessel function

- **Addition:**

wn(n)=wb(n)+wk(n)=0.42+0.5cos(2*pi*n/N-1)+0.08cos(4*pi*n/N-1)+Io[alpha(1-(2*n/N-1)2)0.5]/I0[alpha] (3)

- **Multiplication:**

wn(n)=wb(n)*wk(n)
=0.42+0.5cos(2*pi*n/N-1)+0.08cos(4*pi*n/N-1)*****Io[alpha(1-(2*n/N-1)2)0.5]/I0[alpha] (4)

- **Averaging:**

wn(n)=(wb(n)+wk(n))/2
=0.21+0.5cos(2*pi*n/N-1)+0.04cos(4*pi*n/N-1)+0.5Io[alpha(1-(2*n/N-1)2)0.5]/I0[alpha] (5)

- **Exclusive-or:**

wn(n)=k*wb(n)+(1-k)wk(n) [3.21]
=k*(0.42+0.5cos(2*pi*n/N-1)+0.08cos(4*pi*n/N-1))+(k-1)*(Io[alpha(1-(2*n/N-1)2)0.5]/I0[alpha])
Where k=0 to 1. (6)

To improve the frequency response of the filter, hybrid window technique can be used. The combination of windowsreduces the ripples in filter response, decrease the stop band attenuation and increases the accuracy compared to filter response using single window.

2.1.1 FIR filter design steps

1.Choose desired frequency response of the filter $H_d(\omega)$
2.Take inverse Fourier transform of $H_d(\omega)$ to obtain the desired response $h_d(n)$.
By definition of inverse Fourier transform

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{j\omega n} H_d(\omega) \quad (7)$$

3.Choose window function w(n)and determine the product of $h_d(n)$ and w(n).
Let product this given by
$h(n)=h_d(n)*w(n)$ (8)
4.The transfer function H(z) of the filter is obtained by taking z-transform of h(n).
Realize the filter by a suitable structure.

2.1.2 Specifications

The table 1 lists the specifications used in this work to design a FIR low pass filter.

Order (N)	15
Cut- off frequency (ω_c)	0.25π rad/sec
Window used	Hybrid
α	7
β	11
eps	0.01

2.2 $Q_{n.m}$ Format for Fixed-point Arithmetic

Most signal processing and communication systems are first implemented in double precisionfloating point arithmetic using tools like MATLAB. While implementing these algorithms themain focus of the developer is to correctly assimilate the functionality of the algorithm. Qn.m is a fixed positional number system for representing floating point numbers. The Qn.mformat of an N bit number sets n bits to the left and m bits to the right of the binary point. In cases of signed numbers, the MSB is used for the sign and has negative weight. A two's complement fixed point number inQn.m format is equivalent to $b=b_{n-1}\ b_{n-2}\ ...b_1b_0b_{-1}b_{-2}...b_m$,with equivalent floating point value:
$-b_{n-1}2^{n-1}+b_{n-2}2^{n-2}+ \ldots +b_12^1+b_0+b_12^1+b_22^2+þ \ldots b_m2^m$ (9)
For example to compute the floating point equivalent of 0 1101 1010 0000000 in signed $Q_{1.15}$ format. Thefields of the bits and their equivalent weights are shown in Figure 1.

-2^0	.	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}

Assigning values to bit locations gives the equivalent floating point value of the Q format fixedpoint number:
0 1101 1010 0000 000= 1/2 +1/4+ 1/16 +1/32 + 1/128 = 0.85156

2.3Canonical Signed Digit Representation

CSD is a radix 2 signed digit coding. It codes constant using signed digits 1, 0 and -1. AnN bit constant C is represented as:

$C = \sum_{i=0}^{N-1} s_i 2^i \; for s_i belongsto \{-1,0,1\}$ (10)

The expression implies that the constant is coded using signed digits 1, 0 or-1, where each digit $s_i$contributes a weight of 2^i to the constant value. TheCSDrepresentation has the following properties:

- No two consecutive bits in CSD representation of a number are non zero.
- The CSD representation of a number uses a minimum number of non zero digits.
- The CSD representation of a number is unique.

CSD representation of a number can be recursively computed using the string property. The numberis observed to contain any string of 1s while moving from the least significant bit (LSB) to the mostsignificant (MSB). The LSB in a string of 1s is changed to$\bar{1}$ that represents -1, and all the other 1s inthe string are replaced with zeros, and the 0 that marks the end of the string is changed to 1. Afterreplacing a string by its equivalent CSD digits, the number is observed again moving from the codeddigit to the MSB to contain any further string of 1s. The newly found string is again replaced by itsequivalent CSD representation. The process is repeated until no string of 1s is found in the number.

Example: Converting 16'b0011 1110 1111 0111 to CSD representation involves the followingrecursion. Find a string while moving from LSB to MSB and replace it with its equivalent CSDrepresentation:

0011111011110111

0011111011111001$\bar{1}$

The newly formed number is observed again for any more string of 1s to be replaced by itsequivalent CSD representation:

0011111011111001$\bar{1}$

0011111100001$\bar{1}$00$\bar{1}$

The process is repeated until all strings of 1s are replaced by theirequivalentCSDrepresentations:

0011111100001$\bar{1}$00$\bar{1}$

0100000$\bar{1}$ 0000$\bar{1}$00$\bar{1}$

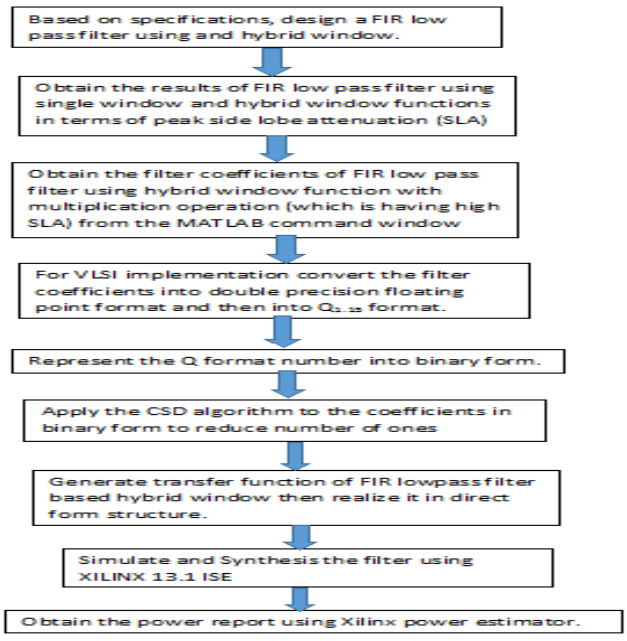
All these steps can be done simultaneously by observing isolated strings or a set of connectedstrings with one 0 in between. All the isolated strings with more than one 0 in between are replacedby their equivalent CSD representations, and for each connected string all the 0s connectingindividual strings are changed to_1, and all the 1s in the strings are all changed to 0. The equivalentCSD representation computed in one step is:

0011111011110111

0100000$\bar{1}$0000$\bar{1}$00$\bar{1}$

III. Proposeddesign

The flowchart in figure 2 explains the proposed design procedure for practical FIR low pass filter using hybrid window and CSD algorithm.



The following table 2 shows FIR filter coefficients withhybrid window (i.e., multiplication operation) of Kaiser and Black man window for N=15 in double precision floating point format, $Q_{1.15}$ format, Binary representation, CSD representation

Table 2. FIR filter coefficients

Filter coefficients	Double Precision Floating Point Format	$Q_{1.15}$ format	Binary representation	CSD representation
h(0)=0.2500	0.249999974297906	8192	10000000000000	10000000000000
h(1)=0.1860	0.185993627075655	6095	1011111001111	$10\bar{1}0000\bar{1}010001\bar{1}$
h(2)=0.0734	0.073385589057126	2405	100101100101	$1010\bar{1}0\bar{1}00101$
h(3)=0.0125	0.012542059857874	411	110011011	$10\bar{1}0100\bar{1}01$
h(4)=-0.0000	0.000002276359643	0	0	0
h(5)= -0.0002	0.000180333843739	6	110	$10\bar{1}0$
h(6)=-0.0000	-0.000007002652330	0	0	0
h(7)=0	0	0	0	0

The following figure 3 shows the architecture of FIR low pass filter realized in direct form structure and the output expression of filter is given in equation 11.

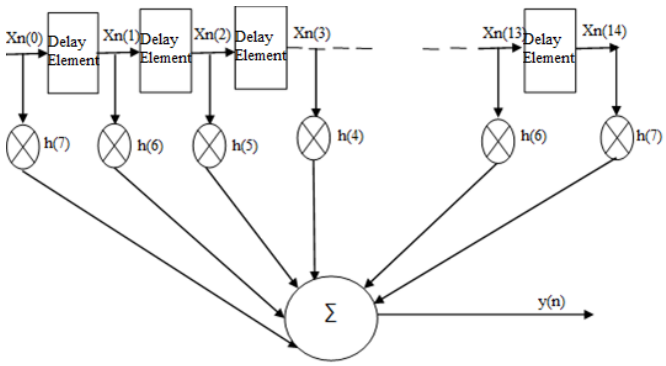


Figure 3. *Architecture of FIR low pass filter*

$Y(n) = x_n(0)*h(7)+x_n(1)*h(6)+x_n(2)*h(5)+x_n(3)*h(4)+x_n(4)*h(3)+x_n(5)*h(2)+x_n(6)*h(1)$
$+x_n(7)*h(0)+x_n(8)*h(1)+x_n(9)*h(2)+x_n(10)h(3)+x_n(11)*h(4)+x_n(12)*h(5)+x_n(13)*h(6)$
$+x_n(14)*h(7)$ (11)

IV. Results and Discussion

MATLAB Simulation Results for 15 tap FIR low pass filter using hybrid windows are shown from figures 4 to 7. Blackman window and Kaiser Window are taken here for generating new hybrid window functions using mathematical operations.Relative Side Lobe Attenuation (SLA)of FIR low pass filter is shown in table3 using different hybrid windows. From table 2, FIR filter using multiplication operator window response gives the highest attenuation compared to other hybrid windows. So, these filter coefficients are taken to realize in direct form structure.The Xilinx Integrated Software Environment (ISE) is used for performing simulation, synthesis and implementation of design. The coding was done by using Verilog language. A 15 tap FIR LPF is designed and implemented with $Q_{1.15}$ format filter coefficients in binary and CSD representations. Simulation, power and delay reports of filter with binary and CSD representations are shown in figures 8 to 15. Table 4 shows the filter implemented using CSD consumes less power when compared to binary representation

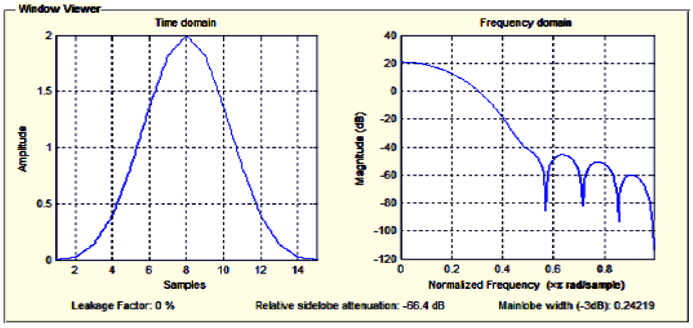


Figure 4. *Response of FIR low pass filter using hybrid window with addition operator*

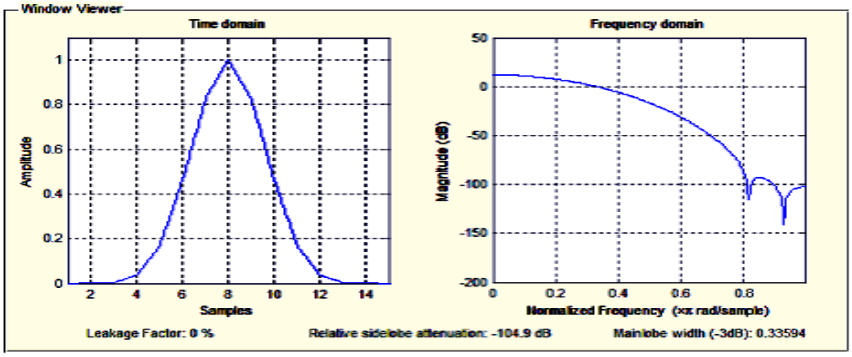


Figure 5. *Response of FIR low pass filter using hybrid window with multiplication operator*

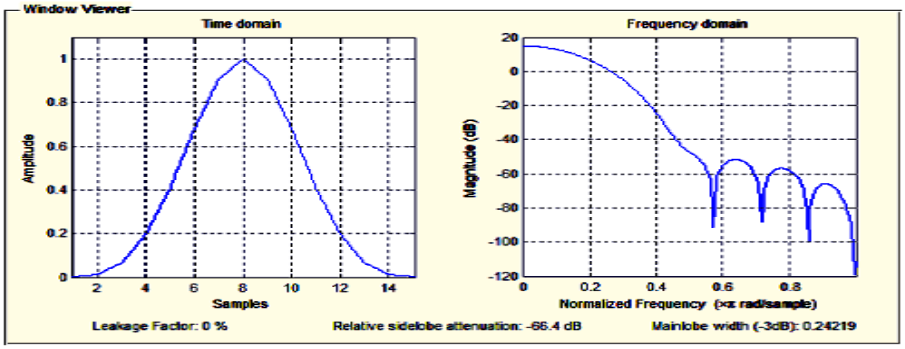


Figure 6. *Response of FIR low pass filter using hybrid window with averaging operator*

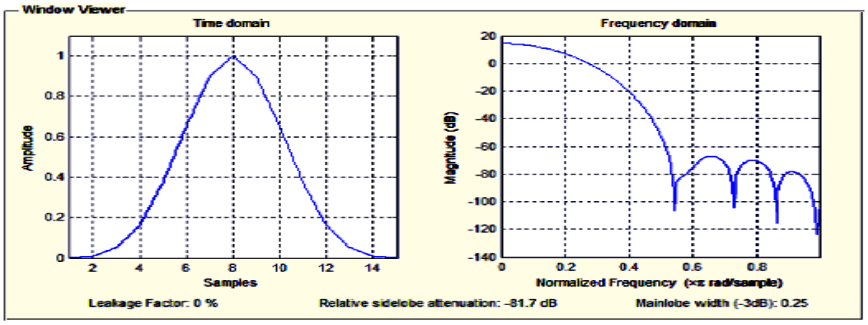


Figure 7. *Response of FIR low pass filter using hybrid window with Ex-or operator*

Table 3. Relative SLA values of FIR LPF for hybrid windows

Mathematical operation	Relative side lobe attenuation
Addition	-81dB
Averaging	-81 dB
Multiplication	-104 dB
Ex-Or	-96.5dB

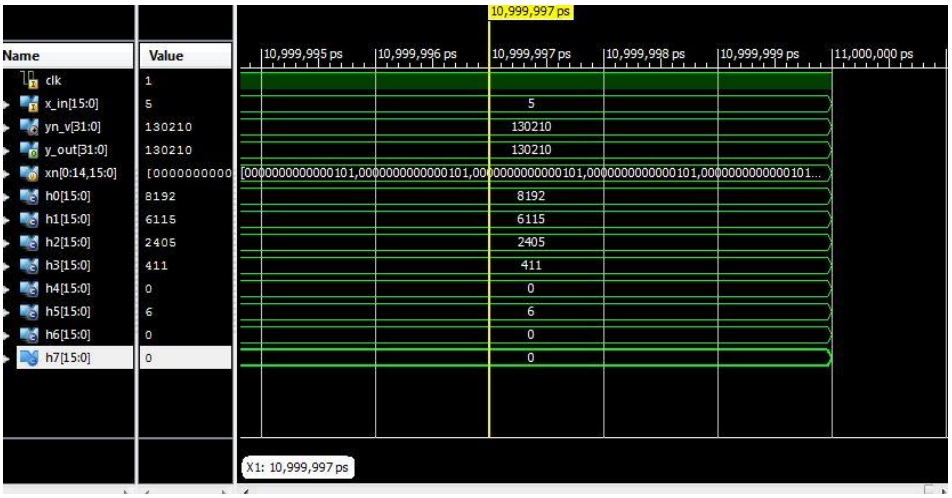


Figure 8. *15 tap FIR LPF output with binary representation*

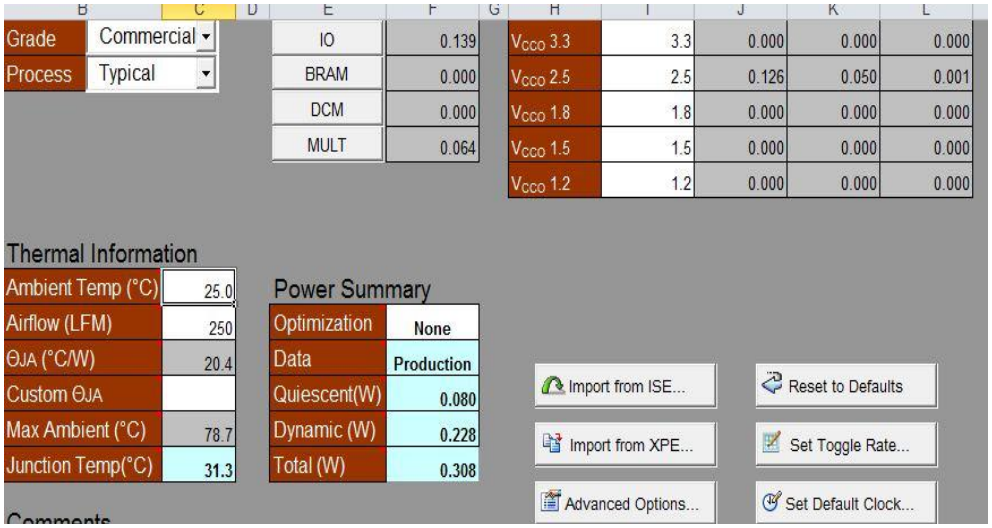


Figure 9. *Power report of 15 tap FIR LPF with binary representation*

```
----------------------------------------
Total                     21.487ns (18.076ns logic, 3.411ns route)
                                   (84.1% logic, 15.9% route)

================================================================================
```

Figure 10. *Delay report of 15 tap FIR LPF with binary representation*

```
Device utilization summary:
---------------------------

Selected Device : 3s500efg320-5

 Number of Slices:                      174  out of   4656     3%
 Number of Slice Flip Flops:             48  out of   9312     0%
 Number of 4 input LUTs:                327  out of   9312     3%
 Number of IOs:                          49
 Number of bonded IOBs:                  49  out of    232    21%
 Number of MULT18X18SIOs:                 8  out of     20    40%
 Number of GCLKs:                         1  out of     24     4%

---------------------------
```

Figure 11. *Synthesis report of 15 tap FIR LPF with binary representation*

Figure 12. *15 tap FIR LPF output with CSD representation*

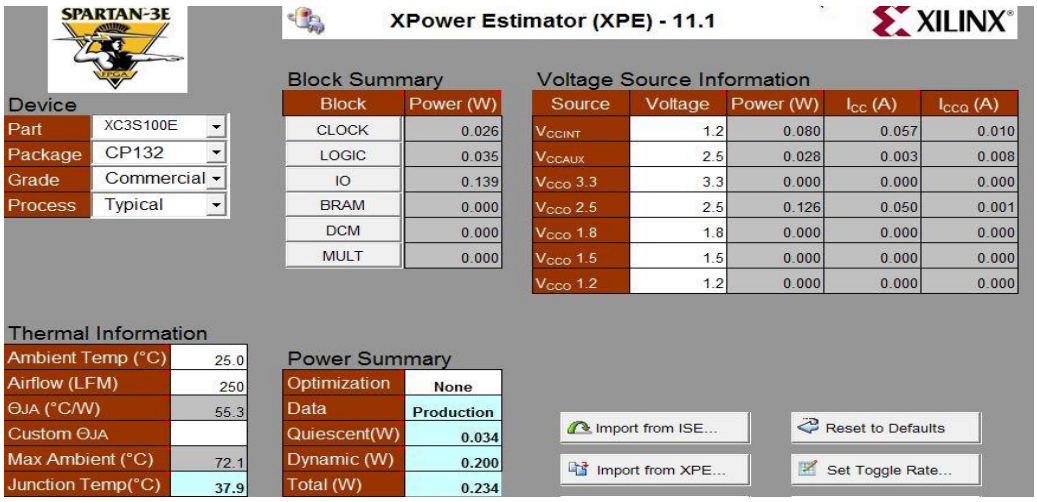


Figure 13*. Power report of 15 tap FIR LPF with CSD representation*

```
----------------------------------------
Total                      52.765ns (34.841ns logic, 17.924ns route)
                                    (66.0% logic, 34.0% route)

================================================================================
```

Figure 14. *Delay report of 15 tap FIR LPF with CSD representation*

```
Device utilization summary:
---------------------------

Selected Device : 3s1500fg320-4

 Number of Slices:                      603  out of  13312     4%
 Number of Slice Flip Flops:            187  out of  26624     0%
 Number of 4 input LUTs:               1103  out of  26624     4%
    Number used as logic:              1038
    Number used as Shift registers:      65
 Number of IOs:                         49
 Number of bonded IOBs:                 49  out of    221    22%
 Number of GCLKs:                        1  out of      8    12%
```

Figure 15. *Synthesis report of 15 tap FIR LPF with CSD representation*

Table 4. Performancecomparision in terms of power consumption

Method of Filter coefficient representation	Binary	CSD
Power consumption	308 mw	234 mw

V. Conclusion and Future Scope

With the advancement in VLSI technology as the Digital signal processing has become increasingly popular over the years, the high speed realization of FIR filter with less power consumption has become much more demanding.In this work, the FIR LPF is designed using hybrid windows for various mathematical operations like addition, averaging, ex-or and multiplication.The highest side lobe attenuation was obtained and achieved better magnitude response for the multiplication based design of FIR low pass filter . Moreover multiplication based design of filter is realized in direct form structure and the CSD algorithm is applied for the multiplication process and obtained the less power consumption compared to binary representation of filter coefficients. The simulation, synthesis and power reports are analyzed using Xilinx 13.1 ISE and XPower Estimator. In multiplication process more number of partial products will be generated. The future scope of this work is to use any partial products reduction technique to reduce further power consumption.

References

[1]. Eppili Jaya, K.ChitambaraRao, JayalaxmiAnem "FIR filter design using new Hybrid Window functions", IJSETR, Vol.4 Issue 6, JUNE – 2015, ISSN: 2278-7798.

[2]. Eppili Jaya, K.ChitambaraRao "Power, Area and Delay Comparision of Different Multipliers". IJSETR, Volume 5, Issue 6, June 2016, ISSN: 2278-7798.

[3]. Muralidhar, P. V., Nataraj, D., LokeshRaju, V., &Naik, S. K. (2010, July). Implementation of different FIR high pass filters using fractional Kaiser window. In Signal Processing Systems (ICSPS), 2010 2nd International Conference on (Vol. 2, pp. V2-651). IEEE.

[4]. Muralidhar, P. V., AS SrinivasaRao, and S. K. Nayak. "Spectral Interpretation of Sinusoidal Wave using Fractional Fourier Transform Based FIR window Functions." International Review on Computers and Software 4.6 (2009): 652-657.

[5]. Hewlitt, Reid M., and E. S. Swartzlantler. "Canonical signed digit representation for FIR digital filters." Signal Processing Systems, 2000. SiPS 2000. 2000 IEEE Workshop on. IEEE, 2000.

[6]. Guo, Rui, and Linda S. DeBrunner. "A novel fast canonical-signed-digit conversion technique for multiplication." 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2011.

[7]. Ruiz, Gustavo A., and Mercedes Granda. "Efficient canonic signed digit recoding." Microelectronics Journal 42.9 (2011): 1090-1097.

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

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

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

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