ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

Efficient Area Minimization with High Speed and Low Power Multiplier Structural Design for Multirate Filter Design

Ch.D.Vishnupriya [1], K.Neelima[2]

M.Tech, Asst. Professor, Dept of Electronics & Communication Engineering, Pragati Engineering College, Surampalem, AP, India[1]

M.Tech Student, Dept of Electronics & Communication Engineering, Pragati Engineering College, Surampalem, AP, India[2]

ABSTRACT: In Multi-rate Signal processing studies used in Digital Signal processing systems include sample rate conversion. This technique is used for systems with different input and output sample rates. Interpolation and Decimation is very effective and popular in multi rate signal processing applications. This paper proposes a high speed, area and power efficient VLSI architecture for polyphase decimation filter with decimation factor of three (D=3) using Booth multiplier. By using booth multiplier to multiply signed numbers also. Various key performance metrics such as number of slices, maximum operating frequency, number of LUT's, input output bonds, power consumption, setup time, hold time, propagation delay between source and destinations are estimated for the filter of length nine (N=9).The power dissipation is reduced in polyphase decimation filter using Booth multiplier which consumes low-power when compared to the conventional multiplier. The speed is improved by using carry look-ahead adder. It was observed that the proposed scheme provides increase in speed, reduction in area and slight reduction in power dissipation when compared to conventional and BFD multiplier and low complexity.

KEYWORDS: Polyphase decimation filter, Booth multiplier, BFD Multiplier architecture, Area, power dissipation, carry look ahead adder, speed.

I. INTRODUCTION

Multirate (decimation/interpolation) filters are among the essential signal processing components in space-borne instruments where Finite Impulse Response (FIR) filters are often used to minimize nonlinear group delay and finite-precision effects. Cascaded (multi-stage) designs of Multi-Rate FIR (MRFIR) filters are further used for large rate change ratio, in order to lower the required throughput while simultaneously achieving comparable or better performance than single-stage designs. Traditional representation and implementation of MRFIR employ polyphase decomposition of the original filter structure, whose main purpose is to compute only the needed output at the lowest possible sampling rate.

Recently, there has been rapid progress in the area of multirate digital signal processing. The applications of multirate systems include subband coding of video, audio, and speech signals, fast transforms using digital filter banks, wavelet analysis of all types of signals, and many other fields. In multirate systems, decimation and interpolation filters are the most important building blocks. A great amount of literature deals with the theory and design of decimation and interpolation filters.

However, issues concerning the VLSI implementation scheme for multirate filters have not been investigated thoroughly. Since the speed of processing time and the silicon area are the crucial factors in the VLSI implementation] a scalable implementation scheme to flexibly and efficiently implement the multirate FIR filters is presented in this paper.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

The advances in information technology and the increasing requirement of Very Large Scale Integration issues have resulted in a rapid development of several optimization algorithms and techniques. Low power consumption and smaller area are some of the most important criteria for the DSP systems and high performance systems. DSP systems have less sensitivity to component tolerances and environmental changes and the dynamic range of the system can be increased by floating point arithmetic [2]. Multirate processing arises in many fields of digital signal processing.

Multirate signal processing applications includes digital audio tape, transmultiplexers, subband coding, speech processing, Analog voice privacy systems etc [1]. In digital audio, the sampling rate conversions are about 32 kHz to 44.1kHz and 44.1kHz to 48kHz and vice versa. Symbol rate processing, bit rate processing and sample rate processing are some of the signal processing issues in digital communication [3]. Polyphase decomposition is one of the most important techniques used in multirate signal processing. Polyphase structure utilizes FIR filter that leads to very efficient implementation. Because FIR filters are conditionally stable and linear phase filters. Linear phase in the sense that their phase delay and group delay must be a constant. In data communication, if a pulse is smeared then the received signal does not convey the desired information to the intended user. So, linear phase is the most inherent property of FIR filters.

Exact linear phase cannot be achieved with IIR filters. It can only be achieved with FIR filters. Genetic algorithm is used to find the multistage parameter combination for multistage sampling rate conversion FIR filter design [4]. A bit-level optimized algorithm is used for the design of high-speed FIR decimation filters [5]. Efficient FIR filters are implemented by using systolic decomposition [19]. Power dissipation is the factor that changes rapidly and is one of the major challenging issues today. Texas Instruments are making digital circuits which require no power supply.

Particularly for implantable devices like heart artificial transplantation and artificial hearing aids, they are aiming at devices which will work from body temperature. The power dissipation is minimized by reducing the switching activity factor and by minimizing number of operations to be held in the filter structure. The switching activity reduced by adder and counter [8]. The multipliers have slightly less area and power than optimized tree multipliers while keeping similar delay [9]. The reduction of power consumption is obtained by altering multiplicands in software without any hardware modifications [10]. Optimized Wallace tree and pipelining techniques were used to design a power aware booth multiplier [11]. SPST(Spurious Power Suppression Technique) is applied on multipliers for high-speed and lowpower purposes [12]. Low power fixed width multipliers are used to improve the speed, reduce power and area considerably [13]. A low-power structure called bypass zero, feed A directly (BZ-FAD) for shift-and-add multiplier architecture considerably lowers the switching activity [7]. In this paper, FIR filter with power efficient Booth multiplier is preferred here to get reduction in power dissipation.

This paper describes the design of polyphase multiplier with high speed low power Booth multiplier architecture which uses carry look-ahead adder. This provides reduction in power dissipation as well as increase in speed when compared to conventional multiplier.

II. PROPOSED POLYPHASE DECIMATION FILTER

A filter can be realized with several ways such as direct form, cascade form, linear phase and polyphase realizations. When the transfer function of the filter is decomposed into number of sub branches, the process is called polyphase realization [1] [2] [3].

A. Polyphase FIR filter

Finite Impulse Response (FIR) filter is a zero-phase filter with magnitude equal to unity in the pass band and zero in the stop band [3]. FIR system is described by the difference equation shown in "(1)".

$$y(n) = \sum_{k=0}^{M-1} b_k x(n-k) \qquad (1)$$

Where k b be the filter coefficients, $x(n)$& $y(n)$ are input and output sequences. The equivalent system function is given in "(2)".

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{k=0}^{M} b_k z^{-k} \qquad (2)$$

Furthermore, the unit sample response of the FIR system is identical to the filter coefficients. FIR filters can be designed to provide exact linear phase over the whole frequency range and are always BIBO stable independent of the filter coefficients.

A linear-phase FIR filter of order N is either characterized by a symmetric impulse response given in "(3)".

$$\boldsymbol{h(n) = h(n - N)} \qquad \textbf{(3)}$$

or by an asymmetric impulse response of a FIR filter is given in "(4)"

$$\boldsymbol{h(n) = -h(n - N)} \qquad \textbf{(4)}$$

The symmetry (or asymmetry) property of a linear-phase FIR filter can be exploited to reduce the total number of multipliers into almost half of that in the direct form implementations of the transfer function.

In a general case, the transfer function of L-branch polyphase decomposition of the transfer function of order N is given by "(5)".

$$H(z) = \sum_{m=0}^{L-1} z^{-m} E_m(z^L) \qquad (5)$$

$$E_m(z) = \sum_{n=0}^{\lfloor \frac{N+1}{L} \rfloor} h(Ln+m) z^{-n}, \quad 0 \le m \le L-1$$

FIR filter is realized based on polyphase decomposition which leads a parallel structure. To illustrate this approach, a casual FIR transfer function H(z) of length nine is given in "(6)", which is a function of filter coefficients $h(k), 0 \le k \le 8$.

$$\boldsymbol{H(z) = h(0) + h(1)z^{-1} + \ldots\ldots + h(8)z^{-8}} \qquad \textbf{(6)}$$

The above transfer function can be expressed as a sum of two terms, with one term containing the even-indexed coefficients and the other containing the odd-indexed coefficients which is given by

$$H(z) = h(0) + h(2)z^{-2} + h(4)z^{-4} + h(6)z^{-6} +$$
$$h(8)z^{-8} + z^{-1}[h(1) + h(3)z^{-2} + h(5)z^{-4} + h(7)z^{-6}]$$

Grouping the same equation differently, the transfer function is re-expressed in the form of sub bands as shown in "(7)".

$$H(z) = E_0(z^3) + z^{-1}E_1(z^3) + z^{-2}E_2(z^3) \qquad (7)$$

$$\text{Where } E_0(z) = h(0) + h(3)z^{-1} + h(6)z^{-2}$$
$$E_1(z) = h(1) + h(4)z^{-1} + h(7)z^{-2}$$
$$E_2(z) = h(2) + h(5)z^{-1} + h(8)z^{-2}$$

Equation (7) is represented schematically in fig.1. The structure has delays, multipliers and accumulators.

B. Transposed Polyphase Decimation Filter

The decimator with a decimation factor D, where D is a positive integer develops an output sequence $y(n)$ with a sampling rate is (1/D)th of the input sequence $x(n)$. This is implemented by keeping every Dth sample of the input sequence and removing D-1 samples between consecutive samples. As a result, all input samples with indices equal to an integer multiple of D are retained at the output and all others are discarded, to generate the output sequence according to the relation given in "(8)".

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

$$y(n) = x(nD) \quad (8)$$

Decimation results in aliasing, to avoid the aliasing effect ant aliasing filter called low pass filter is used before down sampling [17].

The transposed form of the decimation filter is used in this paper to avoid the shift registers used in tapped delay structure. Here, $x(n)$ comes into the filter at the sample rate, fs, and is applied to all of the tap multipliers at the same time. The number of stages in the transposed filter depends on the value of decimation factor and number of coefficients. The number of stages is equal to the number of coefficients divided by decimation factor. If the coefficients are not a multiple of decimation factor then we have to append zeros.

The structure of proposed transposed form of polyphase filter for decimation factor of three is shown in fig.2. In this structure, during each cycle, input $x(n)$ is passed through the tap multiplier. Initially the input sequence $x(n)$ is given through a parallel input serial output (PISO) shift register.

The output from PISO is given to three sub filter section where the input sequences are processed in parallel form by means of serial input parallel output (SIPO) shift register. Each section, the input is multiplied with the coefficients. Since the processing is done in parallel, this structure improves the speed of operation. The coefficients are cycled through their values at the sample rate, but the indexes of the coefficients at any given time are separated by D, the decimation rate.

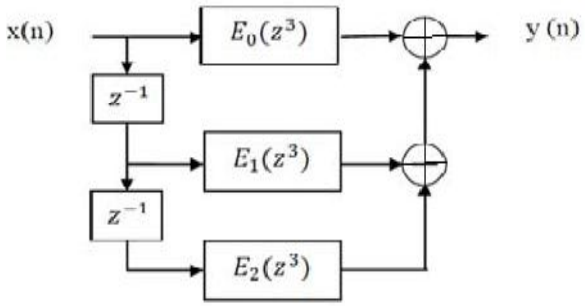


Fig.1 Polyphase FIR filter structure

The coefficient is multiplied with the sample, but the input samples and coefficients changing during each cycle. The function of the accumulator is to accumulate the value of multiplier for every 3(D) cycles. Carry look-ahead adder is used to perform addition in order to increase the speed of operation. The decimated output y(n) is obtained using this polyphase structure.

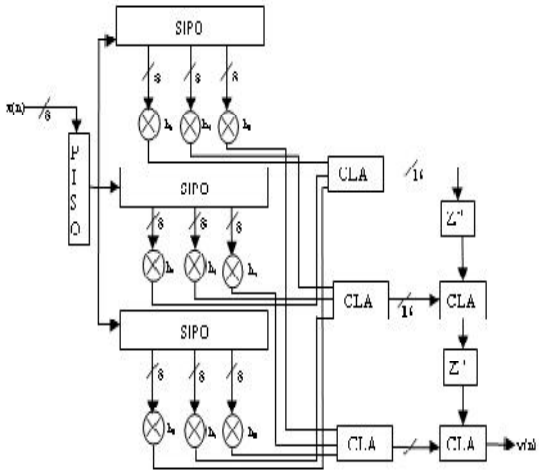


Fig.2. Proposed Transposed Decimation Filter structure

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

III. MULTIPLIER ARCHITECTURES

There are several multipliers available to perform multiplications. Shift and add, BFD multiplication, conventional multiplication and spurious power suppression are some of the techniques used to perform multiplication. In this paper booth multiplier is compared with the conventional and BFD multiplier.

A. BOOTH MULTIPLIER:

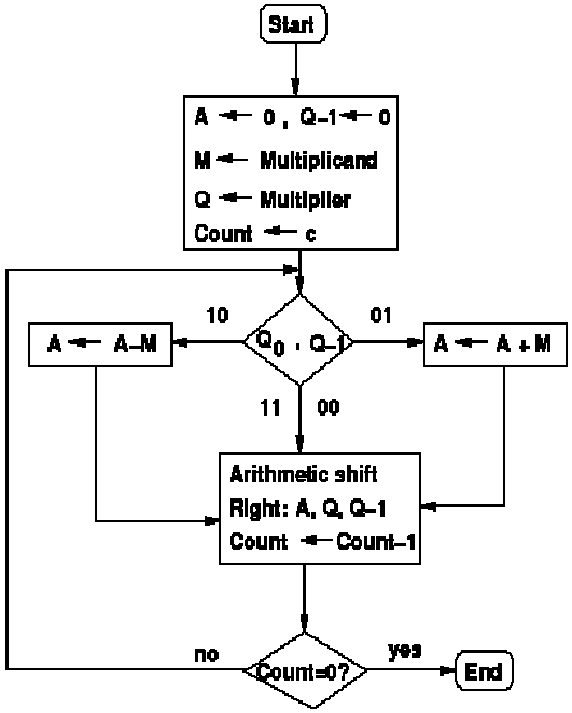


Fig 3 booth multiplier

It is a powerful algorithm for signed-number multiplication, which treats both positive and negative numbers uniformly.

For the standard add-shift operation, each multiplier bit generates one multiple of the multiplicand to be added to the partial product. If the multiplier is very large, then a large number of multiplicands have to be added. In this case the delay of multiplier is determined mainly by the number of additions to be performed. If there is a way to reduce the number of the additions, the performance will get better.

Booth algorithm is a method that will reduce the number of multiplicand multiples. For a given range of numbers to be represented, a higher representation radix leads to fewer digits. Since a k-bit binary number can be interpreted as K/2-digit radix-4 number, a K/3-digit radix-8 number, and so on, it can deal with more than one bit of the multiplier in each cycle by using high radix multiplication. This is shown for Radix-4 in the example below.

Radix – 4 Booth Algorithm

Multiplicand A = ● ● ● ●
Multiplier B = (●●)(●●)

Partial product bits ● ● ● ● $(B_1B_0)_2 A4^0$
● ● ● ● $(B_3B_2)_2 A4^1$
Product P = ● ● ● ● ● ● ● ●

Fig. Radix-4 multiplication in dot notation

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

As shown in the figure above, if multiplication is done in radix 4, in each step, the partial product term $(B_{i+1}B_i)_2$ A needs to be formed and added to the cumulative partial product. Whereas in radix-2 multiplication, each row of dots in the partial products matrix represents 0 or a shifted version of A must be included and added.

Table 1below is used to convert a binary number to radix-4 number. Initially, a "0" is placed to the right most bit of the multiplier. Then 3 bits of the multiplicand is recoded according to table below or according to the following equation:

$\mathbf{Z_i = -2x_{i+1} + x_i + x_{i-1}}$

Example:

Multiplier is equal to 0 1 0 1 1 10

Then a 0 is placed to the right most bit which gives 0

1 0 1 1 10 0 ← 0 added

The 3 digits are selected at a time with overlapping left most bit as follows:

X_{i+1}	X	X_{i-1}	$Z_{i/2}$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	2
1	0	0	-2
1	0	1	-1
1	1	0	-1
1	1	1	0

Table 1 Radix-4 Booth encoder

For example, an unsigned number can be converted into a signed-digit number radix 4:

$(10\ 01\ 11\ 01\ 10\ 10\ 11\ 10)_2 = (-2\ \ 2\ -1\ \ 2\ -1\ -1\ \ 0\ -2)_4$

The Multiplier bit-pair recoding is shown in Table 3.2

0	0	0	+0*multiplicand
0	0	1	+1*multiplicand
0	1	0	+1*multiplicand
0	1	1	+2*multiplicand
1	0	0	-2*multiplicand
1	0	1	-1*multiplicand
1	1	0	-1*multiplicand
1	1	1	-0*multiplicand

Table 2 Radix-4 Booth encoder

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

Here –2*multiplicand is actually the 2s complement of the multiplicand with an equivalent left shift of one bit position. Also, +2 *multiplicand is the multiplicand shifted left one bit position which is equivalent to multiplying by 2.

To enter ± 2*multiplicand into the adder, an (n+1)-bit adder is required. In this case, the multiplicand is offset one bit to the left to enter into the adder while for the low-order multiplicand position a 0 is added. Each time the partial product is shifted two bit positions to the right and the sign is extended to the left.During each add-shift cycle, different versions of the multiplicand are added to the new partial product depends on the equation derived from the bit-pair recoding table above.

Let's see some examples:

```
          000011      (+3)
      X   011101 0    (+29)
            +2 -1 +1
      ------------------
      000000000011
      1111111101
      00000110
      ------------------
1 ←   000001010111    (+87)
```

B. Carry look-ahead adder (CLA)

The carry lookahead adder (CLA) solves the carry delay problem by calculating the carry signals in advance, based on the input signals. It is based on the fact that a carry signal will be generated in two cases: (1) when both bits ai and bi are 1, or (2) when one of the two bits is 1 and the carry-in is 1. Thus, one can write,

$$c_{i+1} = a_i.b_i + (a_i \oplus b_i).c_i$$
$$s_i = (a_i \oplus b_i) \oplus c_i$$

The above two equations can be written in terms of two new signals Pi and Gi, which are shown in Figure 4:

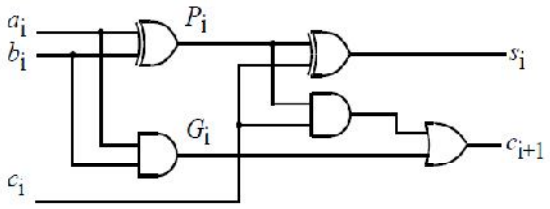


Figure 4: Full adder at stage i with P_i and G_i shown.

$$c_{i-1} = G_i + P_i.c_i$$
$$s_i = P_i \ominus c_i$$

$$G_i = a_i.b_i$$
$$P_i = a_i \ominus b_i$$

Pi and **Gi** are called the carry generate and carry propagate terms, respectively. Notice that the generate and propagate terms only depend on the input bits and thus will be valid after one and two gate delay, respectively. If one uses the above expression to calculate the carry signals, one does not need to wait for the carry to ripple through all the previous stages to find its proper value. Let's apply this to a 4-bit adder to make it clear.

10.15662/ijareeie.2014.0311097

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

Putting I = 0,1,2,3 in Equation 5, we get

$$c_1 = G_0 + P_0.c_0$$
$$c_2 = G_1 + P_1.G_0 + P_1.P_0.c_0$$
$$c_3 = G_2 + P_2.G_1 + P_2.P_1.G_0 + P_2.P_1.P_0.c_0$$
$$c_4 = G_3 + P_3.G_2 + P_3.P_2.G_1 + P_3.P_2.P_1.G_0 + P_3.P_2.P_1.P_0.c_0$$

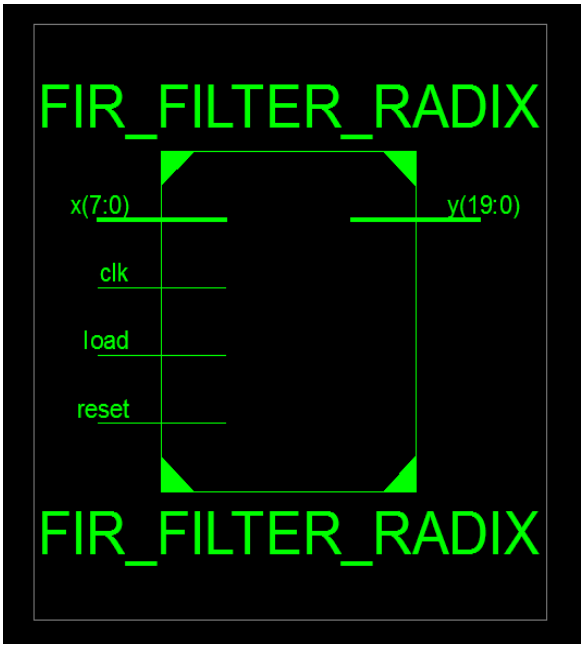
Implementation Observations:

The implementation of the proposed algorithm is illustrated in the form of various pictorial views obtained during the process of FPGA Implementation. Figure presents the Synthesis report of integrated Interpolator and Decimator module which was constructed from a gate level net list to the model of a circuit described in Verilog HDL. Figure shows the RTL view of proposed algorithm whereas Figure depicts the technical schematic view of targeted FPGA device.

IV. RESULTS AND DISCUSSION

In this paper, we have presented how to make FIR filters applicable for multiple sampling rates with the help of interpolator and decimator. For reducing the complexity we had used the low complexity FIR filter design with the help of programmable shifting method. The proposed technique in this paper is able to reduce the complexity of decimator filter and interpolator filter circuits in DSP processing where multiple sampling rates are required.

The polyphase decimation filter is designed and verified using Xilinx 14.3 and ModelSim 6.3g. The Xilinx simulation results are as follows. The RTL (Register Transfer Logic) schematic layout, technology schematic layout and design summary of Multi rate FIR filter is shown in below. 6. The result is the output of the multiplier is 16 bits wide. The below fig shows the detailed RTL schematic of Multiplier architecture.

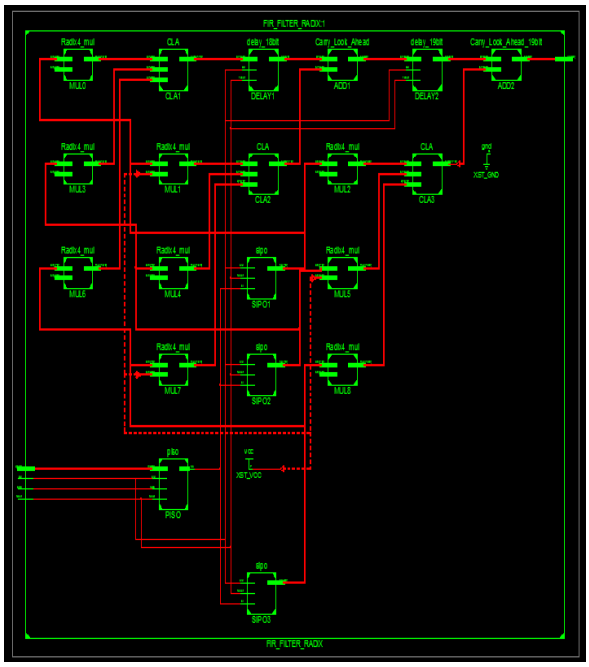


Block diagram

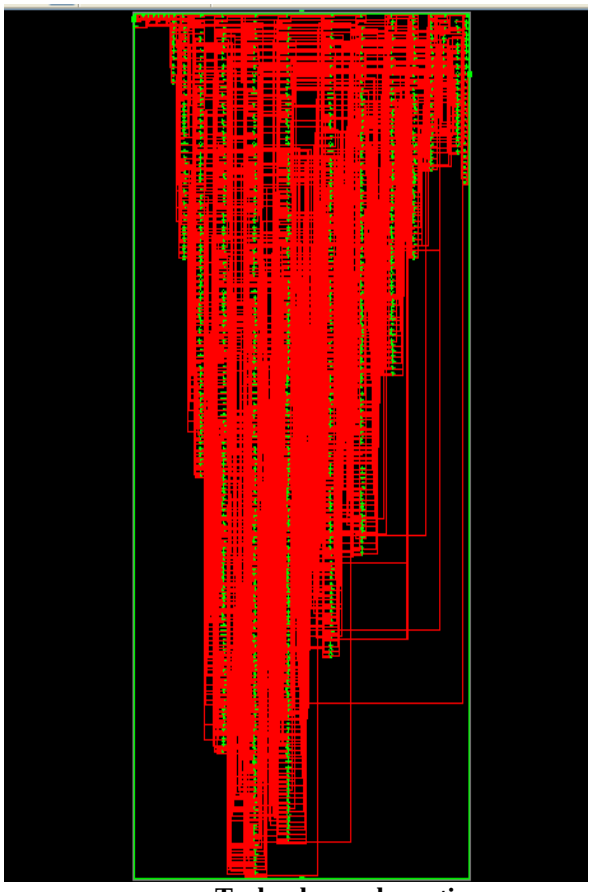
ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014



RTL schematic



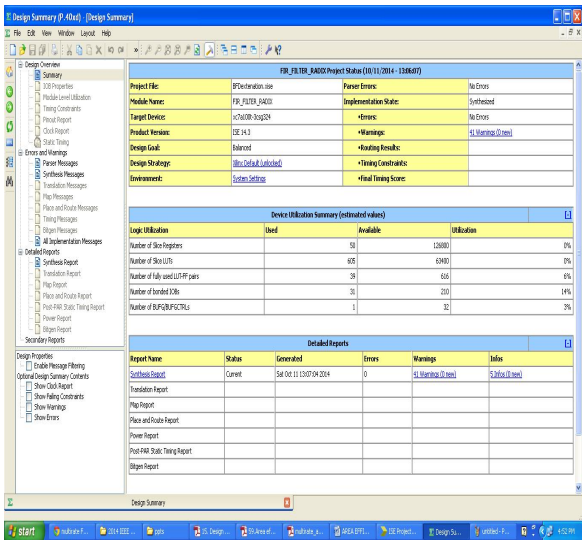
Technology schematic

10.15662/ijareeie.2014.0311097

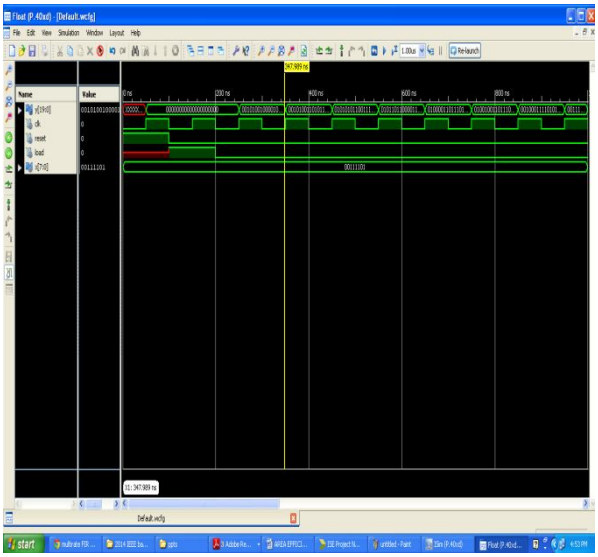
ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014



Design summary



Output waveform

V. CONCLUSION

In this paper, a polyphase decimation filter with high speed , area and power efficient Booth multiplier architecture has been proposed. Various key performance metrics such as number of slices, maximum operating frequency, number of LUT's, input- output bonds, power consumption, setup time, hold time, propagation delay between source and destinations are estimated for the filter of length nine (N=9).Power dissipation is reduced slightly in polyphase decimation filter using this low-power multiplier architecture when compared to the conventional shift and add multiplier and BFD multiplier.

Advantages:

The advantages of this work is that it reduced complexity which in turn reduces the area, power dissipation ,gives higher throughput rate, higher processing speed, fast computation, LFSR can rapidly transmit a sequence that indicates high-precision relative time offsets and many more.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

REFERENCES

[1] Sean Caffee, " Continuously variable fractional rate Decimator" , Xilinx, xapp936 (v1.1), march 5, 2007.
[2] John G.Proakis, Dimitris G.Manolakis, "Digital signal rocessing , principles, algorithms and applications", Fourth edition, Prentice Hall of India, 2006.
[3] P.P.Vaidyanathan, "Multirate systems and filter banks", ISBN 978-81- 7758-942-9, Pearson, Fifth Impression, 2011.
[4] Der-Feng Huang, Shau-Ren Hung, "The Optimum esign of Multistage Multirate FIR Filter for Audio Signal Sampling Rate Conversion via a Genetic Algorithm Approach" IEEE 2009.
[5] Anton Blad and Oscar Gustafsson "Bit-Level Optimized FIR Filter Architectures for High-Speed Decimation pplications", IEEE 2008
[6] AbdSamad Benkrid, Member, IEEE, and Khaled Benkrid, Senior Member, IEEE. "Novel Area-Efficient FPGA Architectures for FIR Filtering With Symmetric Signal Extension", IEEE Transactions on Very large Scale Integration (VLSI) Systems, VOL. 17, NO. 5, MAY 2009.
[7] M. Mottaghi-Dastjerdi, A. Afzali-Kusha, and M. Pedram, "BZ-FAD A Low-Power Low-Area Multiplier Based on Shift-and-AddArchitecture",volume 7,No.2,February 2009.
[8] C.N.Marimuthu, Dr.P.Thangaraj and Aswathy ramesan, "Low power Shift and Add Multiplier design", International Journal of Computer Science and Information Technology, Volume 2, Number 3, june 2010.
[9] Zhijun Huang and Milos D. Ercegovac, "High-Performance Left-to- Right Array Multiplier Design", Computer Science Dept., University of California Los Angeles, CA 90095.
[10] Kyungate Han, "Data Wordlength Reduction for Low-Power Signal Processing Software", final report, May13,2004.
[11] Hanho LEE, "Power-Aware Scalable Pipelined Booth Multiplier", IEICE TRANS Fundamentals vol.E88-A, N0.11, Nov 2005.
[12] K.-H.Chen and Y.-S. Chu. "A low-power multiplier with the spurious power suppression technique," IEEE Trans.Very Large Scale Integr.(VLSI) Syst.,vol.15,no.7,pp.846-850,Jul 2007.
[13] J.S.Wang, C.N. Kuo, and T.H.Yang, "Low-power fixed-width array multipliers," in Proc.IEEE Symp. Low Power Electron. Des.,2004, pp 307-312.
[14] V.P.Nelson, H.T.Nagle, B.D. Carroll, and J.I. David, "Digital Logic Circuit Analysis & Design", Englewood Cliffs, NJ: Prentice-Hall, 1996.
[15] N.-Y.Shen and O.T.-C.Chen, " Low power multipliers by minimizing switching activities of partial products," in Proc.IEEE Int.Symp.Circuits Syst.,May 2002,vol.4,pp.93-96.
[16] O.T.Chen,S.Wang, and Y.-W. Wu, "Minimization of switching activities of partial products for designing low-power multipliers",IEEE Trans.Very Large Scale Integr.(VLSI) Syst., vol.11,no.3, pp.418-433, Jun. 2003.
[17] Robert Bregovic, Tapio Saramaki, Ya Jun Yu, and Yong Ching Lim, "An Efficient Implementation of Linear-Phase FIR Filters for a Rational Sampling Rate Conversion", IEEE 2006.
[18] Alvin Joseph J. Tang Joy Alinda Reyes, "Comparative Analysis of Low Power Multiplier Architectures" 2011 Fifth Asia Modelling Symposium, IEEE 2011.
[19] Pramod Kumar Meher, Shrutisagar Chandrasekaran, Abbes Amira, "FPGA realization of FIR filters by Efficient and Flexible Systolization using Distributed Arithmetic", IEEE Transactions on Signal Processing, VOL.56, NO.7, July 2008.