Design and Performance Analysis of 64 bit Multiplier using Carry Save Adder and its DSP Application using Cadence

Krishna Naik Dungavath
Assistant Professor, Dept. of ECE, PVKKIT,
Anantapuramu,, Andhra Pradesh, India.

Dr. V.Vijayalakshmi
Associate Professor, Dept. of ECE,
Pondicherry Engineering College, Puducherry, India.

Abstract: In this paper we have shown the design and implementation of multiplier in which carry save adder is used as an adder block for the addition of partial products of both multiplier and multiplicand as 64 bits and the product size is of 128 bit. Multiplication is the fundamental arithmetic operation that plays a critical role in several processors and digital signal processing systems. Digital signal processing systems need multiplication algorithms to implement DSP algorithms such as filtering where the multiplication algorithm is directly within the critical path. The Finite Impulse Response (FIR) filter is a digital filter widely used in Digital Signal Processing applications in various fields. The implementation of an FIR requires three basic building blocks i.e. Multiplication, Addition, Unit delay. In a DSP system the multiplier must be fast and must have sufficient precision (bit width) to support the desired application. A high quality filter will in general require more multiplications than one of lesser quality, so throughput suffers if the multiplier is not fast. Hence 64 bit multiplier with carry save adder is designed and the same block which is of 8 bit is implemented in FIR (8-tap) filter. A comparison between array multiplier and multiplier with carry save adder is shown and the proposed technique is efficient in terms of power. A comparison between FIR filter with array multiplier block and FIR filter with multiplier with carry save adder block is shown and the proposed technique is efficient in terms of power and speed. The code is written in Verilog and the simulation and synthesis is carried out in Cadence Encounter tool.

***Keywords*: Cadence Encounter, Verilog, Array Multiplier, Multiplier with Carry Save Adder, FIR Filter with Array Multiplier block, FIR Filter with Multiplier with Carry Save Adder block**

I. INTRODUCTION

The major considerations while designing the digital circuits are speed, power and area. Multiplication is a mathematical operation that at its simplest is an abbreviated process of adding an integer a specified number of times. A basic multiplier can be divided into three parts i) partial product generation ii) partial product addition and iii) final addition. Multiplication plays an important role in Digital Signal Processing (DSP) applications, such as filtering and fast Fourier transform (FFT). Parallel array multipliers are widely used to achieve high speed execution. But these multipliers consume more power. In today"s VLSI system design, Power consumption has become a critical concern. For the design of low-power DSP systems the designers need to concentrate on power efficient multipliers. The impulse response of the filter can be either finite or infinite. The methods for designing and implementing of these two filter classes differ considerably. Finite impulse response (FIR) filters are digital filters whose response to a unit impulse (unit sample function) is finite in duration. This is in contrast to infinite impulse response (IIR) filters whose response to a unit impulse (unit sample function) is infinite in duration. FIR and IIR filters each have advantages and disadvantages. In some applications, the FIR filter circuit must be able to operate at high sample rates, while in other applications the FIR filter circuit must be a low power. Circuit operating at moderate sample rates.The main objective of this project to design power efficient multiplier block and to design high speed and low power FIR filter. The work carried out is described in brief as follows. Section II explains the multiplication of two numbers i.e. array multiplication. Section III represents the architecture of multiplier with carry save adder. Section IV describes the FIR filter with array multiplier block. Section V shows the FIR filter with multiplier with carry save adder block. Section VI consists of experimental results. Section VII concludes this paper

				A3	A2	A1	A0	Inputs
			x	B3	B2	B1	B0	
			C	B0 x A3	B0 x A2	B0 x A1	B0 x A0	
		+	B1 x A3	B1 x A2	B1 x A1	B1 x A0		
		C	sum	sum	sum	sum		
	+	B2 x A3	B2 x A2	B2 x A1	B2 x A0			Internal Signals
	C	sum	sum	sum	sum			
+	B3 x A3	B3 x A2	B3 x A1	B3 x A0				
C	sum	sum	sum	sum				
Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Outputs

Fig.1.Array Multiplication

II. ARRAY MULTIPLICATION

Array multiplier is well known due to its regular structure. Multiplier circuit is based on add and shift algorithm. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial product are shifted according to their bit orders and then added. The addition can be performed with normal carry propagate adder. In array multiplication we need to add, as many partial products as there are multiplier bits.

III.ARCHITECTURE OF MULTIPLIER WITH CARRY SAVE ADDER

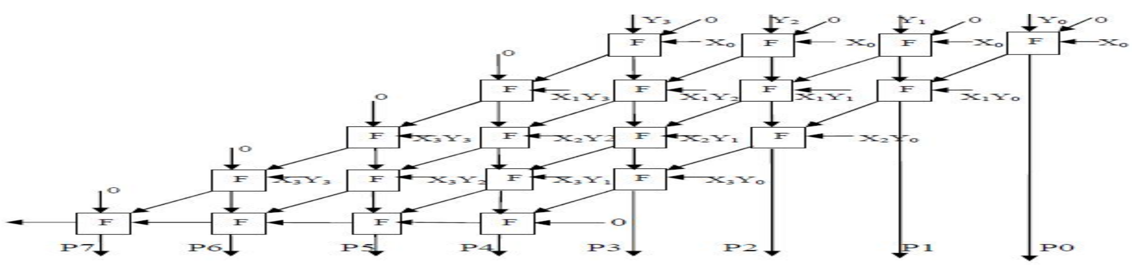


Fig. 2.Multiplier with Carry saves Adder Architecture

In the Carry Save Addition method, the first row will be either Half-Adders or Full-Adders. If the first row of the partial products is implemented with Full-Adders, C_{in} will be considered „0". Then the carries of each Full- Adder can be diagonally forwarded to the next row of the adder. The resulting multiplier is said to be Carry Save Multiplier, because the carry bits are not immediately added, but rather are saved for the next stage. In the design if the full adders have two input data the third input is considered as zero. In the final stage, carries and sums are merged in a fast carry-propagate (e.g. ripple carry or carry look ahead) adder stage.

IV.FIR FILTER

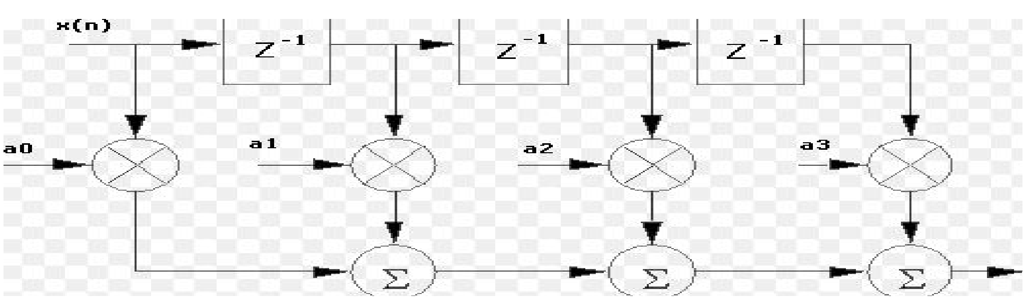


Fig.3. Basic Form of FIR Filter

Filters are signal processing components that are used to process interfered and corrupted signals. They can be classified to two main categories: analog and digital filters. Filters in these two categories are different in terms of cost, speed, accuracy, power consumption and implementation, but they are similar in the sense that they are both used to filter signals. A commonly used method of implementing digital filters is by considering a subset of the filter‟s impulse response. Filter designed this way are called finite impulse response (FIR) filters. The mathematical process used to get the output of a linear system according to its impulse response is the convolution. When a digital signal x[n] is to be processed by a system of impulse response h[n], the output is the result of the following equation

$$y[n] = \sum_{k=0}^{N-1} h[k]x[n-k]$$

The above equation describes how each sample of the output signal is calculated. This is an application of the widely used mathematical operation of the dot product, which consists purely of multiplication and addition. Here multiplication is carried out using array multiplier and addition by the basic adder.

V.FIR FILTER WITH MULTIPLIER WITH CARRY SAVE ADDER

Here the basic form of FIR Filter structure is considered. The building blocks of FIR filter is multiplier, adder and delay unit. Here in case of multiplier we consider multiplier with carry save adder block. In case of adder we use basic adder for addition. Delay element we are using is D-Flipflop.FIR filter with multiplier with carry save adder block is the new technique which is proposed to improve speed and to reduce power.

VI. RESULTS

The analysis is done using Cadence Encounter tool to simulate and synthesize both Array Multiplier and Multiplier with Carry Save Adder, FIR Filter with Array Multiplier and FIR Filter with Multiplier with Carry Save Adder. The code is written in Verilog HDL to optimize the power of 64 bit multiplier and to optimize the power and speed of FIR filter.

Array multiplier

Simulation waveforms

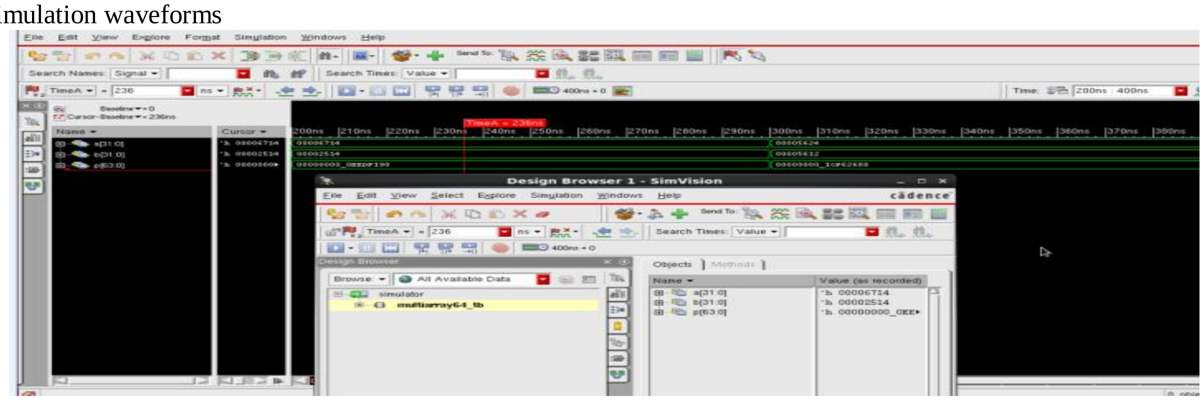
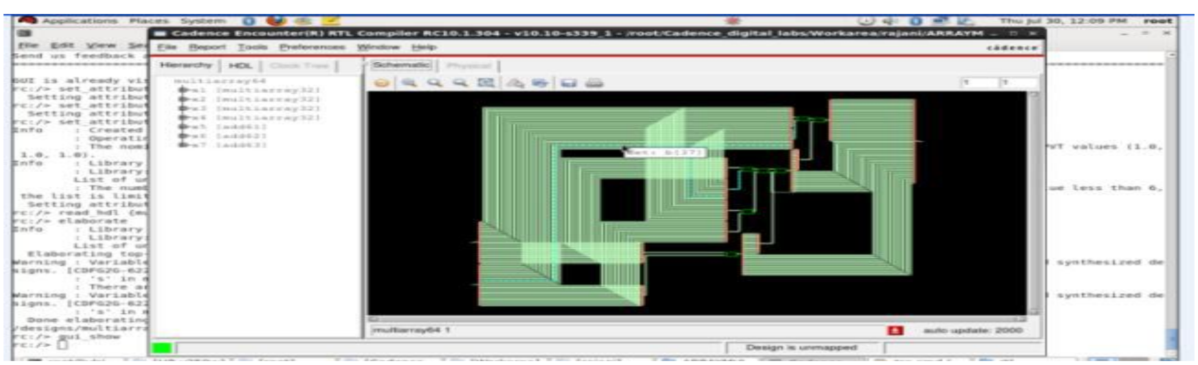


Fig. 4. 64- bit array multiplier waveforms

Synthesis Report



Power Report

```
rc:/> report power
============================================================
  Generated by:              Encounter(R) RTL Compiler RC10.1.304 - v10.10-s339_1
  Generated on:              Jul 30 2015  12:46:55 pm
  Module:                    multiarray64
  Technology library:        slow_normal 1.0
  Operating conditions:      slow (balanced_tree)
  Wireload mode:             enclosed
  Area mode:                 timing library
============================================================

                                    Leakage     Dynamic      Total
       Instance          Cells     Power(nW)   Power(nW)   Power(nW)
-----------------------------------------------------------------------
multiarray64             10050 467693.206 5420443.934 5888137.141
  x1                      2464 113446.301 1152903.687 1266349.989
    x4                     592  26652.809  235474.655  262127.464
      x4                   136   5830.760   39643.409   45474.169
        x1                  28   1058.052    4854.421    5912.474
          FA6                1     94.386     423.490     517.876
          FA4                1     94.329     351.196     445.524
          FA3                1     94.272     295.658     389.930
          FA7                1     94.211     596.757     690.969
          FA5                1     94.016     447.852     541.868
          FA2                1     93.997     276.785     370.782
          FA8                1     93.984     417.067     511.051
```

Multiplier with carry save adder

Simulation waveforms

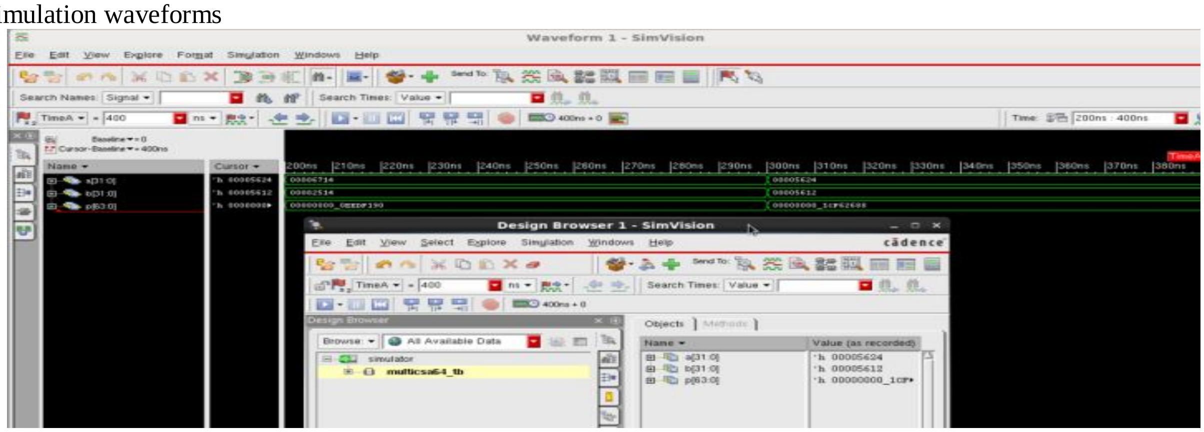
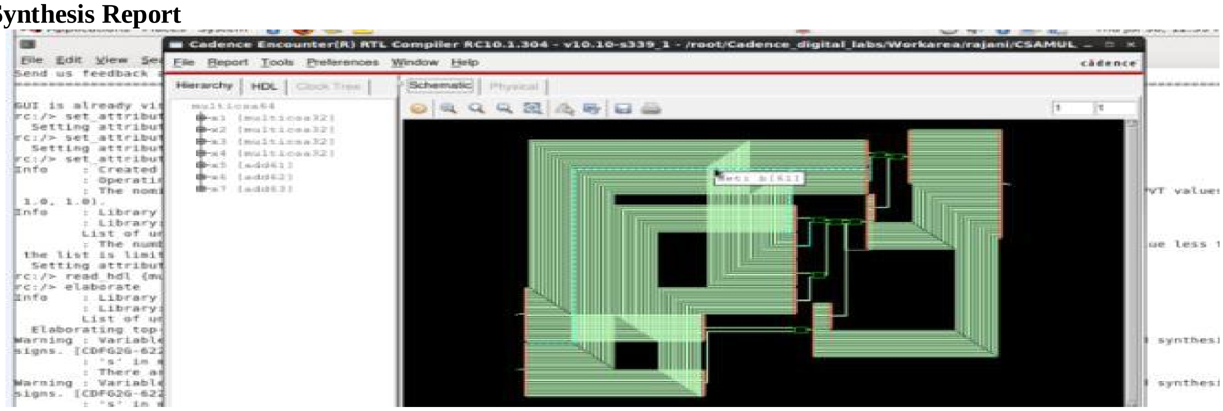


Fig. 5 64- bit multiplier with carry save adder waveforms

Synthesis Report



Power Report

```
rc:/> report power
============================================================
  Generated by:           Encounter(R) RTL Compiler RC10.1.304 - v10.10-s339_1
  Generated on:           Jul 30 2015  01:07:55 pm
  Module:                 multicsa64
  Technology library:     slow_normal 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================
```

Instance	Cells	Leakage Power(nW)	Dynamic Power(nW)	Total Power(nW)
multicsa64	15426	451914.303	5400644.307	5852558.610
x4	3808	109519.181	1141260.016	1250779.197
x3	928	25664.906	227295.061	252959.966
x1	220	5578.215	38730.980	44309.195
x2	49	995.817	4842.706	5838.522
x7	1	29.599	322.573	352.172
x12	1	29.492	175.083	204.575
x9	1	29.322	126.245	155.567
x4	1	29.264	170.670	199.934

FIR Filter with Array multiplier

Simulation waveforms

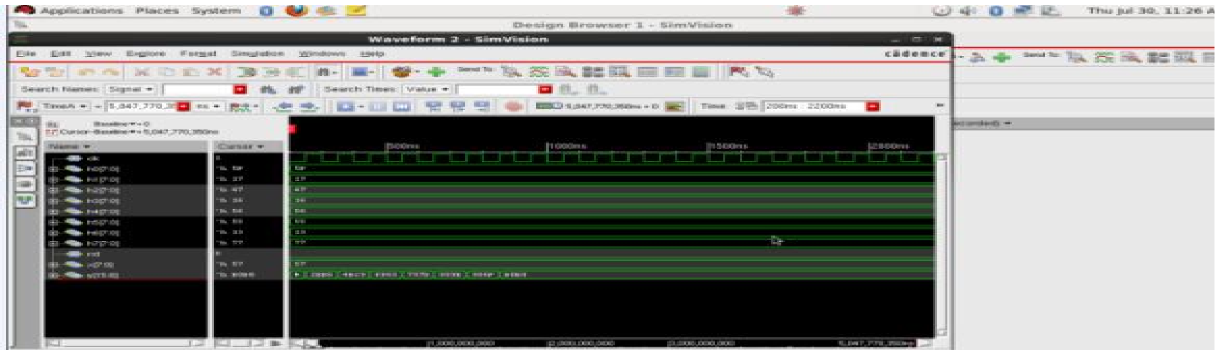
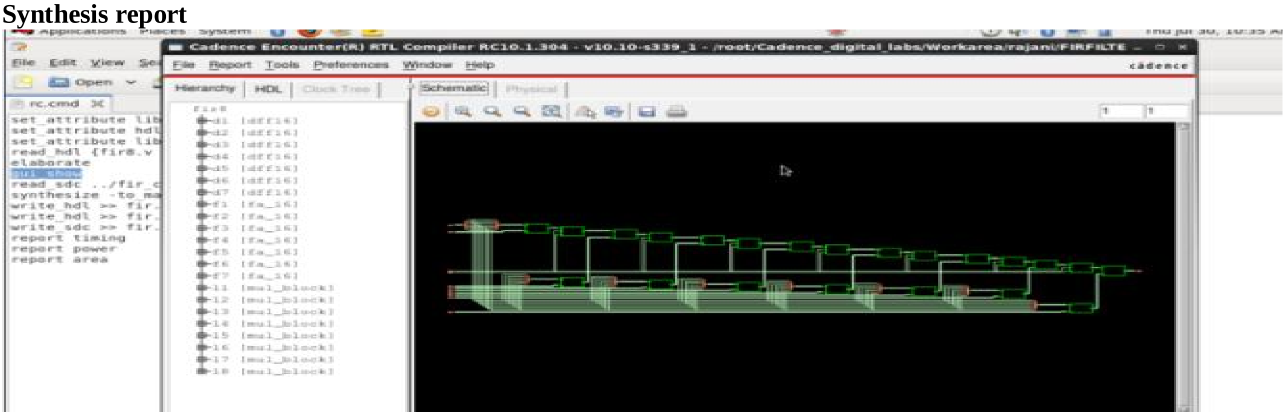


Fig.6. 8 tap FIR Filter waveforms

Synthesis report



Power Report

```
rc:/> report power
============================================================
  Generated by:             Encounter(R) RTL Compiler RC10.1.304 - v10.10-s339_1
  Generated on:             Jul 30 2015  10:39:12 am
  Module:                   fir8
  Technology library:       slow_normal 1.0
  Operating conditions:     slow (balanced_tree)
  Wireload mode:            enclosed
  Area mode:                timing library
============================================================

                                          Leakage    Dynamic     Total
           Instance                 Cells Power(nW)  Power(nW)   Power(nW)
--------------------------------------------------------------------
fir8                                 1094 63378.123 899520.571 962898.694
  l2                                  107  6068.052  55985.454  62053.506
    csa_tree_a..129_60_groupi          43  4906.592  51190.698  56097.291
  l3                                  107  6062.640  58155.847  64218.487
    csa_tree_a..129_60_groupi          43  4901.290  53353.571  58254.861
  l7                                  107  6056.133  63044.190  69100.323
    csa_tree_a..129_60_groupi          43  4889.470  58144.914  63034.384
  l4                                  107  6053.633  62972.471  69026.104
```

Timing Report

```
  add0099/Z[15]
f7/s[15]
y[15]                            out port                        +0    3881 F
(fir_constraints.g_line_14)      ext delay                    +1000    4881 F
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                      capture                              10000 R
------------------------------------------------------------------------------
Cost Group   : 'clk' (path_group 'clk')
Timing slack :    5119ps
Start-point  : x[1]
End-point    : y[15]
```

FIR Filter with multiplier with carry save adder

Simulation waveforms

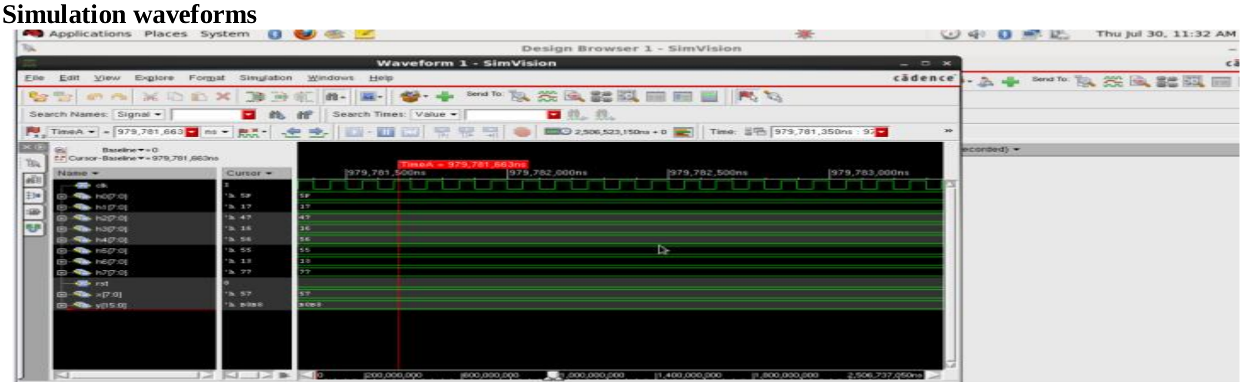


Fig. 7 . 8 tap FIR Filter with multiplier with carry save adder

Synthesis Report

Power Report

```
rc:/> report power
============================================================
  Generated by:           Encounter(R) RTL Compiler RC10.1.304 - v10.10-s339_1
  Generated on:           Jul 30 2015  10:49:31 am
  Module:                 FIRCSA8
  Technology library:     slow_normal 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                       Leakage     Dynamic      Total
  Instance    Cells  Power(nW)   Power(nW)   Power(nW)
-----------------------------------------------------
FIRCSA8        1998  75702.432  771585.021  847287.453
  13            220   7596.349   43471.760   51068.110
    x2           49   1488.086    6337.839    7825.925
      x15         1     44.879     118.203     163.082
      x14         1     44.478      83.537     128.015
      x16         1     44.408     124.398     168.806
      x9          1     44.291     121.716     166.007
```

Timing Report

```
y[15]                              out port                 +0     4328 F
(fir_constraints.g_line_14)        ext delay             +1000     5328 F
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                        capture                        10000 R
---------------------------------------------------------------------------
Cost Group    : 'clk' (path_group 'clk')
Timing slack  :    4672ps
Start-point   : x[0]
End-point     : y[15]
```

The power consumption of 64 bit conventional multiplier and proposed multiplier is shown in the table.

Table.1. Total, Power comparison of different multipliers.

S.No	Multiplier	Total Power (nW)
1.	Conventional Multiplier	5888137.1412
2.	Proposed Multiplier	5852558.610

The power consumption and timing performance of 8 tap Conventional FIR filter and proposed FIR filter is shown in
The table

Table.2.Total Power and Timing comparison of different FIR filters

S.No	FIR Filter	Total Power(nW)	Time(Ps)
1.	Conventional FIR Filter	962898.694	5119
2.	Proposed FIR Filter	847287.453	4672

VII. CONCLUSION

This paper presents two different multipliers and two different FIR filters that are modeled using verilog. The proposed multiplier is more efficient in power than the conventional multiplier. The proposed FIR filter is more efficient in power and timing performance than the conventional FIR filter. The simulation and synthesis reports are obtained using the Cadence tool.

REFERENCES

[1] A text book on “CMOS VLSI DESIGN, A Circuits and Systems Perspective”,4th Edition by Neil H.E.Weste, 2011.
[2] S. Smith, The Scientist and Engineer's Guide to Digital Signal Processing, San Diego: California Technical Publishing, 1997.
[3] Nik Ghazali Nik Daud, Forkful Ridzuan Hashim, **“**Hybrid Modified Booth Encoded Algorithm-Carry Save Adder Fast Multiplier”, IEEE 2014.
[4] Maroju SaiKumar,**”**Design and Performance Analysis of Various Adders using Verilog”, International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 2, Issue. 9, September 2013.
[5] A. Arun,“Design of Novel FIR Filter Using Add and Shift Multiplier and Carry Save Adder”, IJCSEC-International Journal of Computer Science and Engineering Communications, Vol.2 Issue.3, May 2014.
[6] Hesham Altwaijry, “FIR Filter Design Using The Signed-Digit Number System and Carry Save Adders – A Comparison”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No.12, 2013.
[7] N. Jhansi,” Design and Analysis of High Performance FIR Filter using MAC Unit”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3, Issue 11, November 2014.
[8] Mr. Pravin Y.Kadu1,” High Speed and Low Power FIR Filter Implementation Using Optimized Adder And Multiplier Based On Xilinx FPGA”, IORD Journal of Science & Technology E-ISSN: 2348-0831 Volume 1, Issue III (MAR-APR 2014).
[9] Ravi, A.Satish, “A New Design for Array Multiplier withTrade off in Power and Area”, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, May 2011.
[10 Bahram Rashidi, Bahman Rashidi and Majid ourormazd “ Design and Implementation of Low Power Digital FIR Filter based on low power multipliers and adders on xilinx FPGA” IEEE 2011.
[11] V.Vijayalakshmil, R.Seshadd, Dr.S.Ramakrishnan3 “Design and Implementation of 32 Bit Unsigned Multiplier Using CLAA and CSLA” IEEE 2013.