ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering (ICIET'14)

On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

Resource Efficient Reconfigurable Processor for DSP Applications

P.C.Franklin[1], M.Ramya[2]

Department Of Electronics and Communication Engineering,S.A Engineering College, Chennai, India [1]

Department Of Electronics and Communication Engineering,S.A Engineering College, Chennai, India [2]

ABSTRACT--Reconfigurable processor will configure the architecture based on the application. In general processor consists of data path, control and memory unit. In proposed system CSLA(Carry Select Adder) with BEC(Binary to Excess Converter) and CSLA with D-Latch along with Wallace tree multiplier were developed to enhance the performance of MAC(Multiplication and Accumulation) in data path unit. Wallace tree multiplier and CSLA are used to reduce the size of the MAC unit. Multiplication and addition performed in MAC operation which can be enhanced for FIR(Finite Impulse Response) filters application. In MAC operation 16-bit CSLA with BEC and CSLA with D-latch architectures along with 8-bit Wallace tree multiplier which effectively reduces resource utilization. To make the function faster Wallace tree is replaced by Dadda tree multiplier. Reconfiguration in control unit also done for various functions using CSLA with Dadda tree multiplier. Control unit is designed for controlling the operations of the data path unit. By changing the data path and control unit architecture, resource utilization, power, delay and interconnects are reduced efficiently which mostly supports multimedia and DSP applications.

KEYWORDS- CSLA, RCA, BEC, D-latch, Wallace and Dadda multiplier.

I.INTRODUCTION

Multimedia and DSP applications mainly depend upon the speed and performance. To improve this parameters processor used in DSP and multimedia devices also to be very efficient. Generally processor consists of data path, control and memory unit [1]. Data path unit consist of MAC unit which performs arithmetic and logical functions. In order to using efficient adder and multiplier components in MAC unit performance of the processor is improved. This efficient MAC unit is enhanced for FIR filter applications. Multipliers and adders are applied for FIR filters to eliminate the delay in data transitions [2]. In [3][7] they presented area efficient CSLA using RCA and BEC. In [4] they presented Wallace tree multiplier for low area and less delay. In [5] they presented design of FIR filter using computational sharing multiplier based upon carry select adder technique. This paper proposes two modified design in FIR filter using CSLA and Dadda tree multiplier. . One for FIR filter using CSLA with BEC along with Dadda multiplier and other for FIR filter using CSLA with D-latch along with Dadda multiplier Section 2 gives the design of MAC unit. Section 3 presents the design of adder unit. Section 4 describes the design of multiplier unit. Section 5 represents the Mathematical concepts of FIR filter. Section 6 gives design of FIR filter. Section 7 describes simulation and synthesis results. Section 8 represents conclusion of the project.

II.DESIGN OF MAC UNIT

Multiply and accumulate (MAC) operation that calculates the product of two numbers and adds that product to an accumulator unit. MAC, consisting of a multiplier unit followed by an adder and an accumulator register that stores the result. The output of the register is fed back to one input of an adder unit, so that on each clock cycle, the output of the multiplier is added and stored in register unit is described in fig (1). In MAC adder unit is replaced with carry select adder [3] and multiplier unit is replaced with Dadda multiplier [11]. This modified MAC unit is enhanced for FIR filter applications.

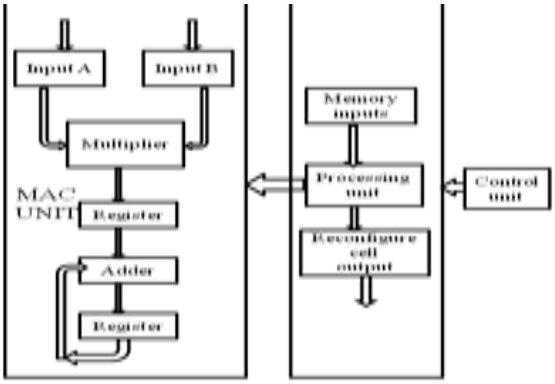

Fig.1. Structure of MAC unit

III. ADDER DESIGN

MAC unit has adder section which uses efficient carry select adder. Carry select consists of two sets of Ripple Carry Adder (RCA) one for carry being zero and another for carry being one and multiplexer which can selects carry input for next stage whether the carry is zero or one.

A. BASIC 16-BIT CARRY SELECT ADDER:

CSLA consists of two inputs A and B with 16 bit each. From the figure X its clear that 16 bit input is separately given to RCA. Based on the first RCA carry output it is fed as the carry input to the next RCA and the multiplier, sum considered as a direct output. Similarly the carry and 16 bit sum output is received. CSLA structure consists of two sets of Ripple Carry adders [3][7] . Upper RCA for carry Cin=0 and lower RCA for carry Cin=1 both produces different carry output. Carry output of upper and lower RCA is connected to multiplexer which can select the carry input for the next stage shown in fig (2).

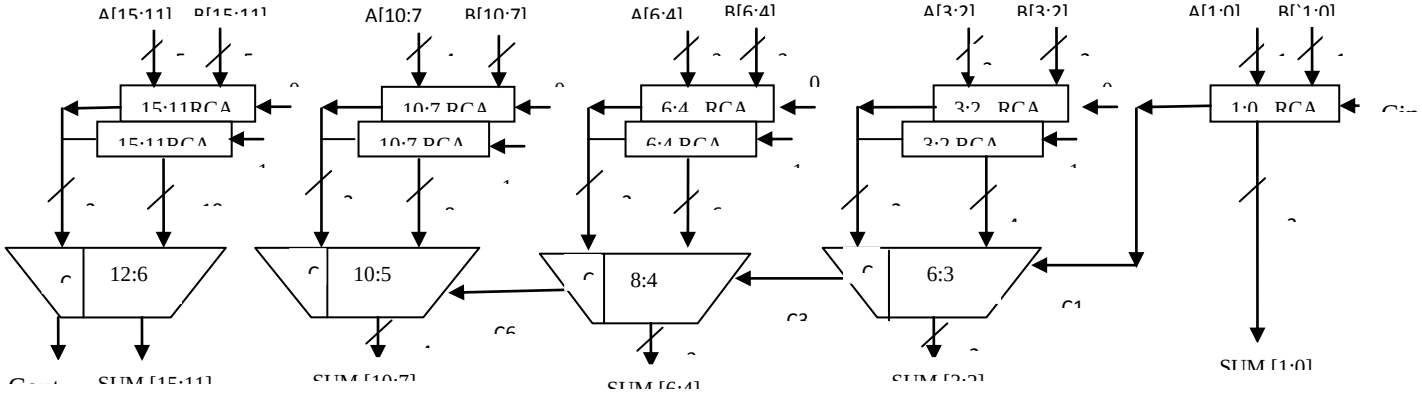

Fig. 2. 16-bit Carry select Adder

B. 16-BIT CSLA USING BEC:

Basic CSLA structure can produces more delay and also resource utilization is high because it uses two sets of RCA for its operation. To avoid this problem CSLA uses Binary to Excess-1(BEC) Code converter for its operation [3]&[7]. One RCA produces carry Cin=0 and another Carry Cin=1.

Here BEC is used instead of RCA with Cin=1 is shown in fig (3). Use of Cin=0 is avoided due to this delay for each operation is reduced. This can occupy less resource as compared with basic CSLA. Remaining operations are of same as that of basic CSLA.

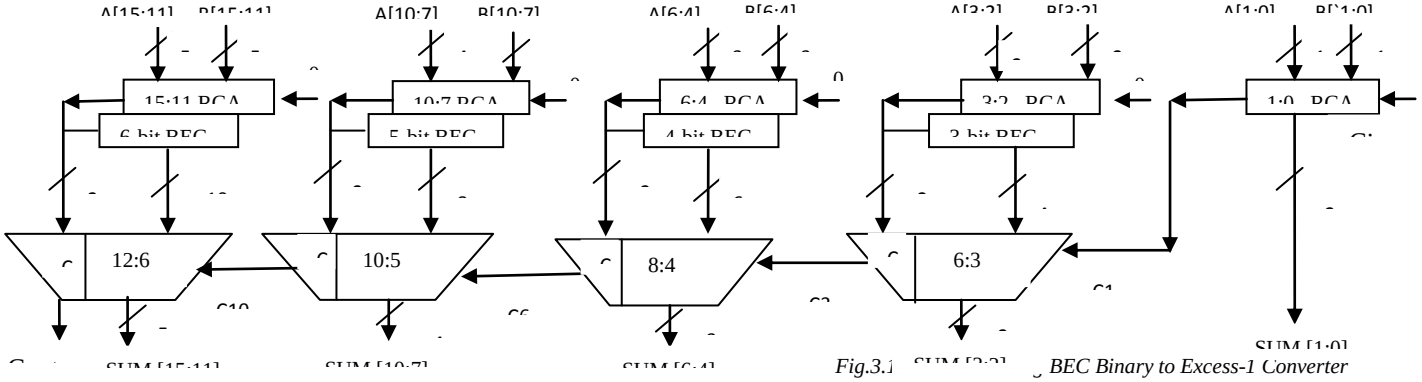

Fig.3. BEC Binary to Excess-1 Converter

The basic function of CSLA with BEC is Shown in fig(4). It consists of 4-bit BEC and 8:4 multiplexer. Multiplexer receives two inputs one for zero and another for one. Zero is direct input for multiplexer and one is BEC output. BEC performs shift by 1 operation. Multiplexer can selects either direct input or BEC output for the next stage.

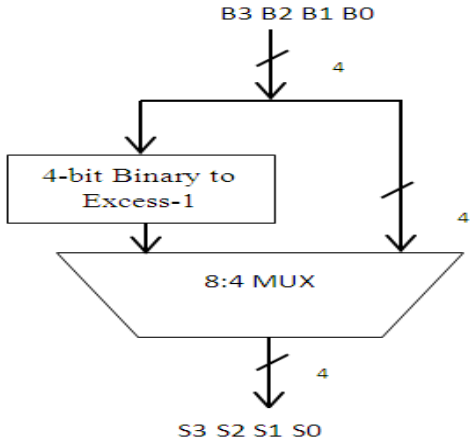

Fig.4. 4-bit BEC operation

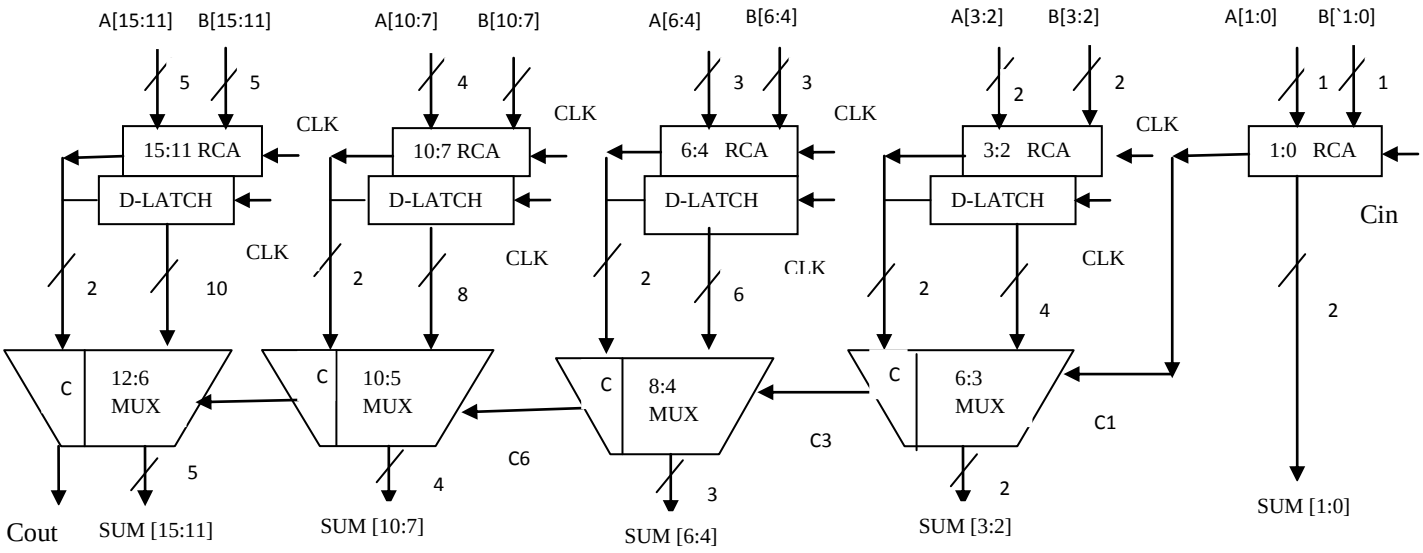

Fig.5.16-bit CSLA using D-latch

A. WALLACE TREE MULTIPLIER

C. 16-BIT CSLA USING D-LATCH

In CSLA with D-latch architecture depends upon clock signal [9]. It can produce carry signal only when the clock input is enable otherwise it will not produce output. So no need to give any separate carry inputs for its operation. When En=1 RCA will calculate the output for Cin=1 and store this value in D-latch. When En=0 then the RCA will calculate the output for Cin=0 and stores in D-latch but the output will not change which will be in same state as that of previous stage which is shown in fig(5). This can produces less delay as compared with CSLA with RCA structure.

IV. MULTIPLIER DESIGN

Multipliers are more energy consuming elements in processor design. In order to select the efficient multiplier components it is possible to reduce the delay as well as resource utilization. In this project describes two multipliers Wallace tree and Dadda multiplier. Both techniques are used for reduction of partial product stage in multiplication operation.

Wallace tree is an implementation of adder tree designed mainly for reducing propagation delay for each stage operations. It has follows three stages such as partial product generation stage, compression and reduction. The fig(6) shows the operation of Wallace tree multiplier. Here uses (8×8) Wallace tree multiplier[4].Multiply each bit of the argument by each bit of the other; Which can generates 8 set of partial products in row order. Depending on position of the multiplier bits the wires carry different weights. Reduce the number of partial products by layer of full adder and half adder. In this full adder is implemented using 3:2 compression technique and half adder is implemented using 2:2 compression technique. Group the wires into two and three columns respective of half and full adder and add them using carry propagation adder. It can uses minimum number of carry propagation adder for final reduction stage. In carry propagation method carry of previous stage is added with the sum of next stage. This algorithm can be mainly developed for reducing the propagation delay for each stage compared with existing multipliers techniques.

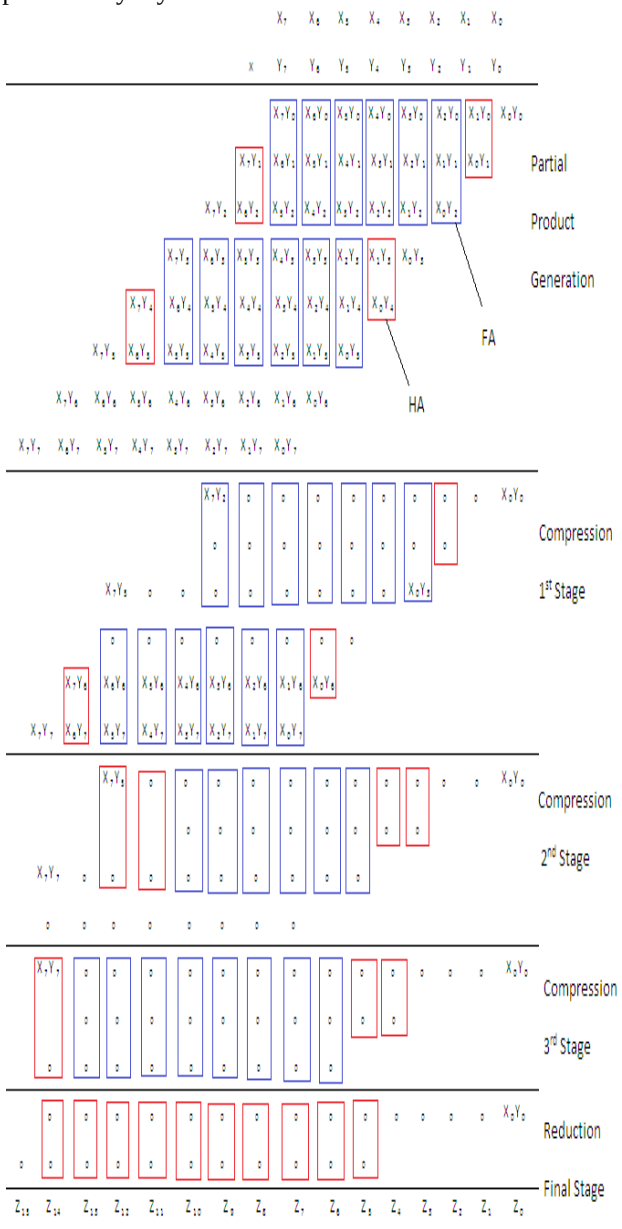

Fig.6. (8×8) Wallace Tree Multiplier

B. DADDA TREE MULTIPLIER

In Wallace tree method the partial products are reduced as soon as possible. In Dadda's tree multiplier does the minimum reduction at each level to perform reduction in the same number of levels as required by Wallace tree multiplier is shown in fig (7). Dadda tree algorithm[8] follows three levels like Wallace tree multiplier such as generation of partial products, compression and reduction Unlike Wallace tree multiplier, Dadda multiplier algorithm requires more carry look ahead adder at final reduction level thus the operation is faster and delay is less as compared with Wallace tree algorithm. There are four reduction stages takes place such as h = 8,6,4,3 and 2.

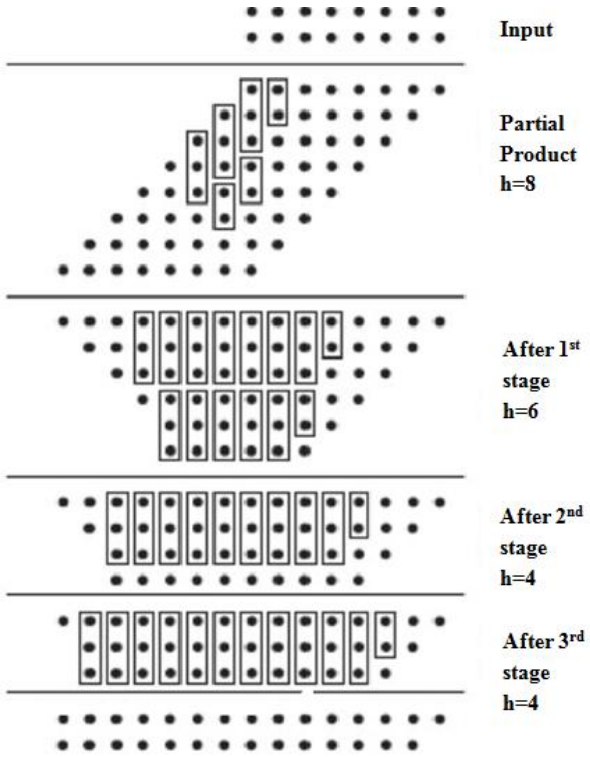

Fig.7. (8×8) Dadda Tree Multiplier

.

V. MATHEMATICAL CONCEPTS USED IN FIR FILTER DESIGN

Filters are very important part of signal processing applications. Filters are used for signal separation and for signal restoration. In general filtering is described by simple convolution operation

y(n) = x(n)*f(n) = $\sum_{k=0} f(k)x(n-k)$

$= \sum_{k=0} x(k)f(n-k)$ (1)

The straight forward way of implementing LTI (Linear Time Invariant) FIR filter is finite convolution of input series x(n) with impulse response coefficients which is given by

y(n) = x(n)*f(n) (2)

$=\sum_{k=0}^{L-1} f(k)x(n-k)$ (3)

Here L is the length of FIR filter, L is the length of FIR filter, h(n) is filter impulse response coefficients, x(n) is input sequence and y(n) is output of FIR filter. The above equations can also expressed in Z domain as

Y(z) = x(z) H(z) (4)

Where H(z) is the transfer function of FIR filter. X(z) is input filter coefficient. Y(z) is output filter coefficient.

VI. DESIGN OF FIR FILTER

FIR filters are used in signal processing applications. Filter structure consists of delay element, adder and multiplier elements. The adder is replaced using carry select adder and multiplier is replaced using Dadda tree multiplier is shown in fig (8).

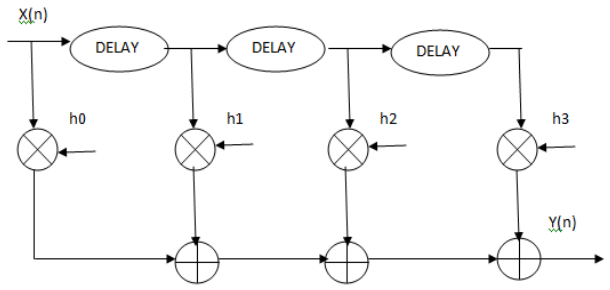

Fig.8. 4-tap FIR filter

There are two FIR filter structures are developed one is CSLA with BEC along with Dadda multiplier and another is CSLA with D-latch along with Dadda tree multiplier. Here X(n) is input coefficient and Y(n) is output filter coefficient. Both can produces less delay as well as consume less resource for its operation.

VII. SIMULATION AND SYNTHESIS RESULTS

We perform the simulation and synthesis and summarize the results of all adders and multipliers. Functional verification of all the adders and multipliers are performed and these modified architectures are applied in 4-tap FIR filter finally results are summarized.

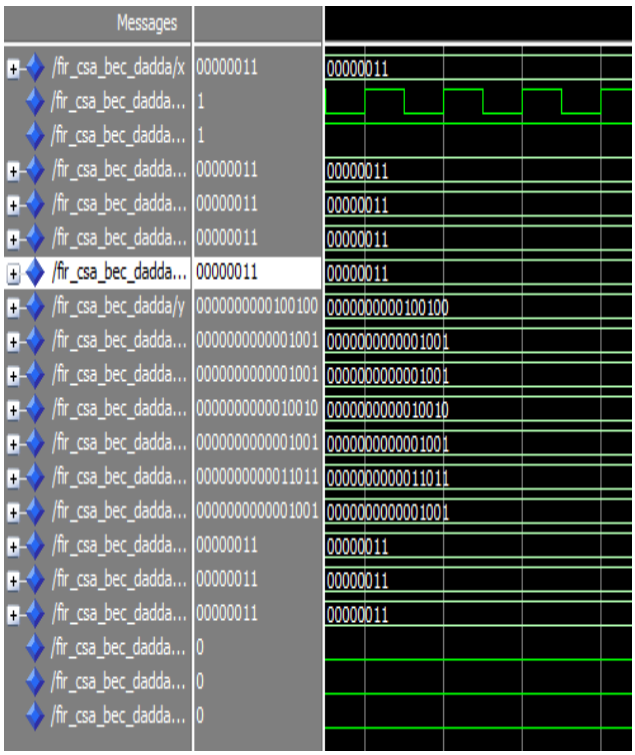

Fig.9. Simulation output for FIR filter using CSLA using BEC and Dadda tree multiplier

Fig.9 shows the output for 4 tap FIR filter using CSLA with BEC and Dadda tree multiplier. Here X is 8-bit input coefficient that is multiplied with 8 bit multiplier filter coefficients h0, h1, h2 and h3 produces 16-bit output. Both are sum together and produce filter output Y. Here multiplier uses Dadda tree multiplier and adder unit uses CSLA with BEC.

Fig.10 shows the output for 4 tap FIR filter using CSLA with D-latch and Dadda tree multiplier. Here X is 8-bit input coefficient that is multiplied with 8 bit multiplier filter coefficients h0, h1, h2 and h3 produces 16-bit output. Both are sum together and produce filter output Y. Here multiplier uses Dadda tree multiplier and adder unit uses CSLA with D-latch.

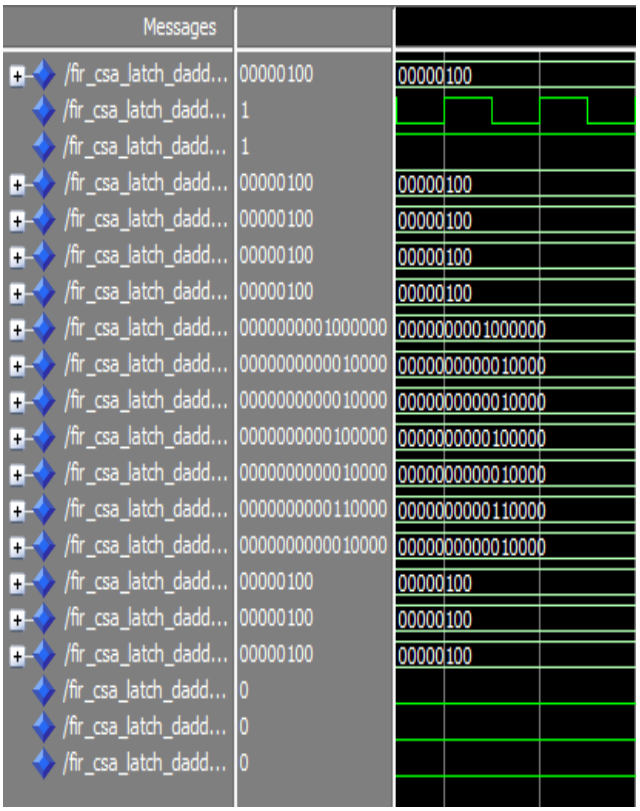

Fig.10. Simulation output for FIR filter using CSLA using D-latch and Dadda tree multiplier

COMPARISON OF ADDER AND MULTIPLIER ARCHITECTURES

After observation of simulation waveforms, synthesis is performed for calculation of delay and area and comparison of adder and multiplier architectures are made in terms of area and delay and listed in the below table. From the comparison it's clear that the area and delay is very much less in proposed adder and multiplier techniques. These modified units are used in MAC which has to be enhanced for FIR filter applications.

TABLE 1
COMPARISON OF ADDER ARCHUTECTURES

Parameters	**Basic CSLA**	**CSLA using BEC**	**CSLA using D-latch**
Number of gates used	30	28	18
Destination paths	662	437	365
Delay(ns)	7.195	7.370	5.984

From the above mentioned table delay and resource utilization is less in CSLA with BEC and CSLA with D-latch compared with basic CSLA.

TABLE 2
COMPARISON OF MULTIPLIER UNITS

Parameters	**Wallace Tree Multiplier**	**Dadda Tree Multiplier**
Number of gates used	64	56
Destination paths	8867	5943
Delay(ns)	11.531	9.377

From the above mentioned table delay and resource utilization is less in Dadda multiplier compared with Wallace tree multiplier.

TABLE 3

COMPARISON OF FIR FILTER STRUCTURES

Parameters	**Number of gates used**	**Delay(ns)**
FIR filter using CSLA-BEC and Wallace multiplier	333	16.957
FIR filter using CSLA-D-latch & Wallace tree multiplier	312	15.889
FIR filter using CSLA-BEC and Dadda tree multiplier	308	13.566
FIR filter using CSLA-D-latch and Dadda tree multiplier	272	12.999

From the above table it is clear that delay and resource utilization is less in FIR filter with CSLA-BEC,D-latch

and Dadda multiplier compared with FIR filter with CSLA-BEC,D-latch and Wallace tree multiplier.

VIII. CONCLUSION

Area efficient MAC unit for data path unit is designed and are implemented in VHDL using Xilinx 10.1 ISE tool and the results are compared in terms of delay and area. Using MAC unit two FIR filter structures are developed one for CSLA with BEC along with Dadda tree multiplier another for CSLA with D-latch along with Dadda tree multiplier. The improved MAC unit is therefore high speed and efficient for VLSI hardware implementation.

REFERENCES

[1] SohanPurohith, Sai Rahul Chalamacheti, Martin Margala and WimVanderbauwhede, 'Throughput/Resource Efficient Reconfigurable processor for Multimedia Applications', *IEEE* Transactions on VLSI Systems, Vol.21,No.7,2013.

[2] A. Senthilkumar, A.M. Natarajan, S.Subha "*Design and Implementation of Low Power Digital FIR Filters relying on Data Transition Power Diminution Technique"* DSP Journal,Volume 8, pp. 21-29, 2008.

[3] Ram Kumar.B, Harish M Kittur , *'Low Power And Area Efficient CSA', IEEE* transactions on VLSI Systems,Vol.20, No.2,2012

[4] Thapliyal.H, Gobi.N, Kumar.K.K.P, Srinivas.M.B , *'Low Power Hierarchical Multiplier and Carry Look Ahead Architecture', IEEE* International Conference on Computer Systems and Applications,2006.

[5]Karunakaran.S, Kasthuri.N , *'VLSI Implementation of FIR filter using computational sharing multiplier based on high speed carry select adder'*, American Journals of Applied Sciences,Vol.9.No.12,2012

[6] Oklobdzija. V. G , *'High-Speed VLSI Arithmetic Units: Adders and Multipliers, in Design of High-Performance Microprocessor Circuits',* Book edited by A.Chandrakasan, *IEEE* Press,2000.

[7] B. Ramkumar, H.M. Kittur, and P. M. Kannan, *'ASIC implementation of modified faster carry save adder', Eur. J. Sci. Res.*, vol. 42, no. 1, pp.53–58,2010.

[8] P.Samundiswary, K.Anitha , '*Design and analysis of CMOS based Daddamultiplier'IJCEM*, vol16, issue 6,2013.

[9] LaxmanShanigarapu, Bhavana P. Shrivastava, *'Low-Power and High Speed Carry Select Adder',* IJSRP,volume 3, Issue 8,2013.