IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 3, March 2015.
www.ijiset.com
ISSN 2348 – 7968

High Speed Non Linear Carry Select Adder Used In Wallace Tree Multiplier and In Radix-4 Booth Recorded Multiplier

[1]Anna Johnson [2]Mr.Rakesh S

[1]M-Tech student, ECE Department, Mangalam College of Engineering, Kottayam, India
[2]Assistant Professor, ECE Department, Mangalam College of Engineering, Kottayam, India

Abstract— Carry Select Adder (CSLA) is one of the fastest adders used in many data-processing processors to perform fast arithmetic functions.By gate level modification of CSLA architecture we can reduce area. Based on this modification 16-b square-root CSLA (SQRT CSLA) architecture have been developed. The proposed design has reduced area as compared with the regular SQRT CSLA .This work evaluates the performance of the proposed designs in terms of area and delay through Xilinx ISE 14.7(VHDL). In here both regular and modified carry select adder is used in a wallace tree and booth multiplier. And done FPGA implementation using Spartan-3.

Keywords :
Low Power,SQRT CSLA, Area Efficient, BEC.

I. INTRODUCTION

Design of area- and power-efficient high-speed data path logic systems are one of the most substantial areas of research in VLSI system design. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder. The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position.The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum . However,the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input Cin=0 and Cin=1 then the final sum and carry are selected by the multiplexers (mux).The basic idea of this work is to use Binary to Excess-1 Converter(BEC) instead of RCA with Cin=1 in the regular CSLA to achieve lower area and power consumption. The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure. Wallace tree multiplier includes carry save addition and carry propagation addition. Under the carry propagation addition, both regular and modified csla is used. And also in radix-4 booth recorded multiplier both regular and modified csla is used.

II. LITERATURE REVIEW

Carry select adder is used for fastest addition. Thus in case of a wallace tree multiplier by using carry select adder in case of a carry propagate adder will reduce delay. Thus the speed of operation will be higher. But the area will be higher. By using modified square root carry select adder in Wallace tree multiplier will be having less area than by using regular square-root carry select adder. Regular csla and modified csla is used in radix-4 booth recorded multiplier. Thus the importance is given to the speed of multiplication.

III.EXISTING SYSTEM

The carry-select adder generally consists of two ripple Carry Adders (RCA) and a Multiplexer .Adding two n-bit numbers with a carry-select adder is done with two adders (therefore two RCA).In order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one.

ISSN 2348 – 7968

IV. BEC

As stated above the main idea of this work is to use BEC instead of the RCA with Cin=1 in order to reduce the area and power consumption of the regular CSLA. To replace the n-bit RCA, an n+1 bit BEC is required. A structure and the function table of a 4-b BEC are shown in Fig. 1 and Table I respectively.

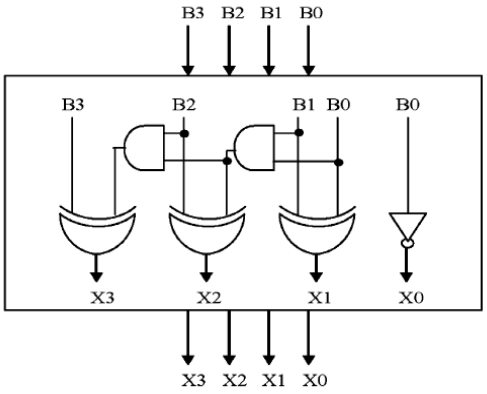

Fig. 1. 4-b BEC.[1]

Table 1. FUNCTION TABLE OF THE 4-b BEC[1]

B[3:0]	X[3:0]
0000	0001
0001	0010
⋮	⋮
1110	1111
1111	0000

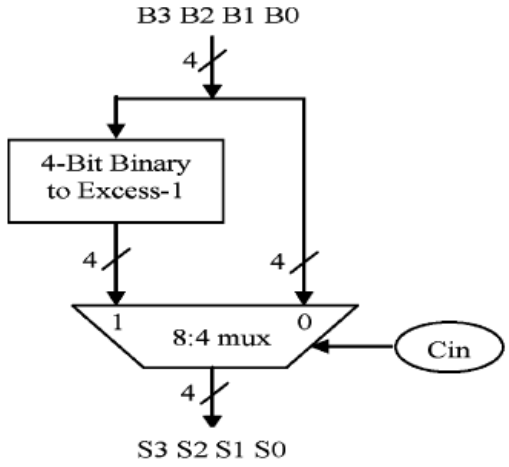

Fig. 2. 4-b BEC with 8:4 mux.[1]

V. REGULAR 16-B SQRT CSLA

The structure of the 16-b regular SQRT CSLA is shown in Fig. 3. It has five groups of different size RCA.

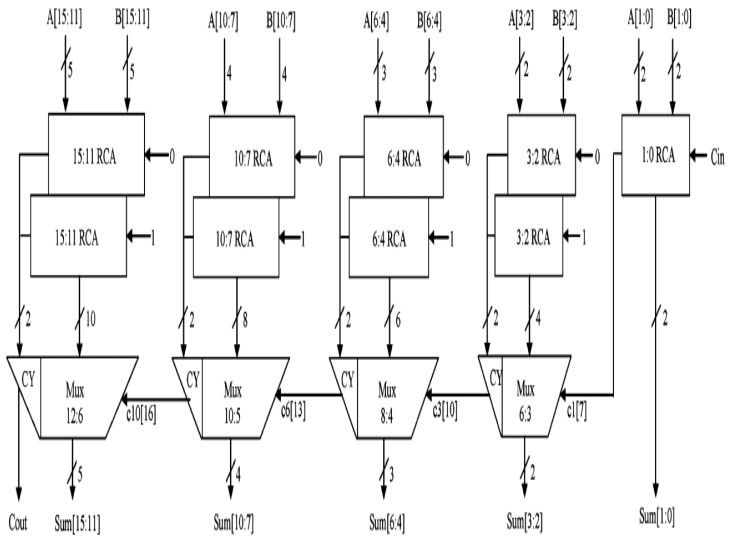

Fig. 3. Regular 16-b SQRT CSLA.[1]

ISSN 2348 – 7968

VI. MODIFIED 16-B SQRT CSLA

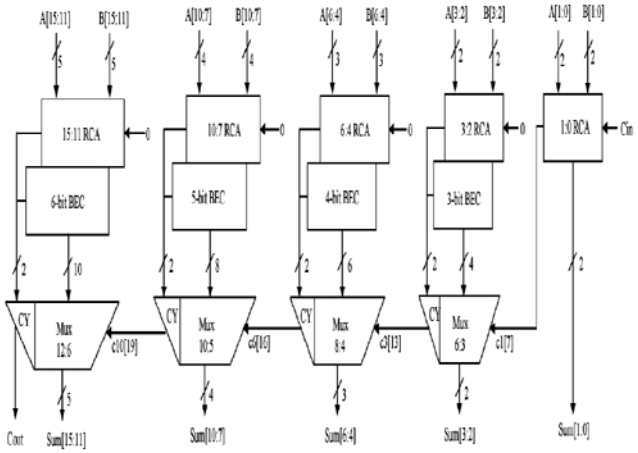

Fig. 4. Modified 16-b SQRT CSLA. The parallel RCA with cin=1 is replaced with BEC[1]

The structure of the proposed 16-b SQRT CSLA using BEC for RCA with cin=1 to optimize the area and power

VII. MODIFICATION DONE IN XOR GATE

In this paper we are considering about the xor gate.In here, existing xor gate having 2 AND gates,2 NOT gates and 1 OR gate.So the expression used in here is (NOT(A) AND B) OR (A AND NOT(B)).This expression is changed to (NOT(A AND B)) AND (A OR B) Thus when drawing the schematic we can understand that one inverter is reduced.When we use this expression in regular and modified 16-bit SQRT CSLA,the area and delay will get reduced in both.

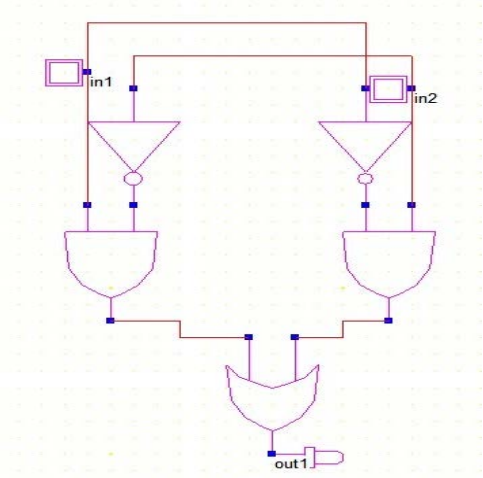

Fig.5.Schematic diagram of existing xor gate

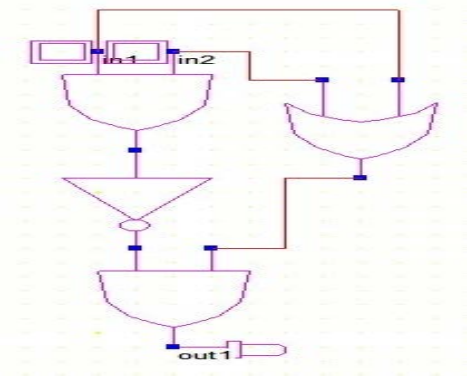

Fig.6.Schematic diagram of modified xor gate

VIII. WALLACE TREE MULTIPLIER

In Digital Signal Processing(DSP) algorithms, multiplier lies in the critical delay path and which determines the performance of the algorithm. The Wallace tree multipliers stimulate VLSI implementation interests reduce the depth of the adder chain thereby minimizing the time complexity. Wallace tree multiplier is a combination of carry save adder and carry propagate adder. Both regular and modified carry select adder is used in case of carry propagation section.Here 8*8 bit wallace tree multiplier is used.

ISSN 2348 – 7968

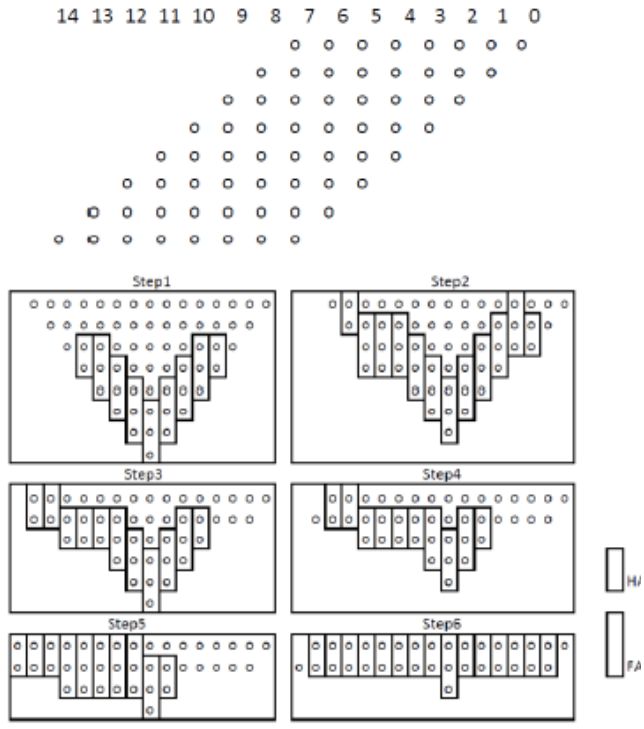

Fig.7.Logic used in Wallace tree multiplier[2]

Thus in case of step6 we can use carry select addition by using both regular and modified carry select adder.

IX. RADIX-4 BOOTH RECORDED MULTIPLIER

In case of radix-4 booth recording ,if n partial products are there then it is reduced to n/2 partial products.

Fig.8.radix-4 booth reduction.

Table .1 Radix-4 Booth recoding

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

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

Fig.9.multiplier bit pair recording[2]

- –2*multiplicand is actually the 2's complement of the multiplicand with an equivalent left shift of one bit position
- +2 *multiplicand is the multiplicand shifted left one bit position which is equivalent to multiplying by 2
- -1 *2's compliment of multiplicand is taken and one left shift is done.
- 1 *multiplicand is placed as such.
- 0 *partial product will be zero.

Thus by using radix-4 booth recording 8 partial products is reduced to 4 partial products and also regular and modified csla used in carry propagate stage reduces delay.

ISSN 2348 – 7968

X.SIMULATIONS

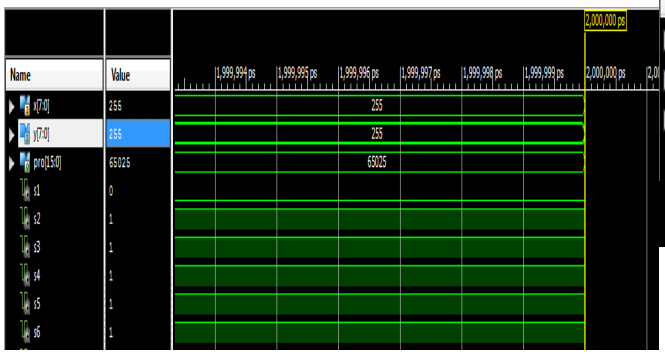

Fig.10. wallace tree multiplier

Cell:in->out	fanout	Gate Delay	Net Delay	Logical Name (Net Name)
IBUF:I->O	11	0.715	1.267	x_1_IBUF (x_1_IBUF)
LUT4:I0->O	3	0.479	1.066	h1/Mxor_sum_Result1 (s1)
LUT4:I0->O	2	0.479	1.040	f8/Mxor_carry_Result1 (c11)
LUT4:I0->O	2	0.479	1.040	h4/carry1 (c22)
LUT4:I0->O	2	0.479	0.915	h10/carry1 (c33)
LUT4:I1->O	3	0.479	0.941	h17/Mxor_sum_Result1 (s43)
LUT3:I1->O	2	0.479	0.768	h26/Mxor_sum_Result1 (s58)
LUT4:I3->O	2	0.479	1.040	f36/Mxor_carry_Result1 (c72)
LUT4:I0->O	2	0.479	1.040	f37/Mxor_carry_Result1 (c73)
LUT3:I0->O	2	0.479	1.040	f38/Mxor_carry_Result1 (c74)
LUT3:I0->O	2	0.479	1.040	f39/Mxor_carry_Result1 (c75)
LUT3:I0->O	2	0.479	1.040	f40/Mxor_carry_Result1 (c76)
LUT3:I0->O	2	0.479	1.040	f41/Mxor_carry_Result1 (c77)
LUT3:I0->O	2	0.479	1.040	f42/Mxor_carry_Result1 (c78)
LUT3:I0->O	2	0.479	1.040	f43/Mxor_carry_Result1 (c79)
LUT4:I0->O	1	0.479	0.681	f45/Mxor_sum_Result1 (PRO_15_OBUF)
OBUF:I->O		4.909		PRO_15_OBUF (PRO<15>)
Total		28.853ns	(12.809ns logic, 16.044ns route) (44.4% logic, 55.6% route)	

Fig.11. delay of Wallace tree multiplier

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

Selected Device : 3s50pq208-5

 Number of Slices:                       93  out of    768    12%
 Number of 4 input LUTs:                163  out of   1536    10%
 Number of IOs:                          32
 Number of bonded IOBs:                  32  out of    124    25%
```

Fig.12.Area of Wallace tree multiplier

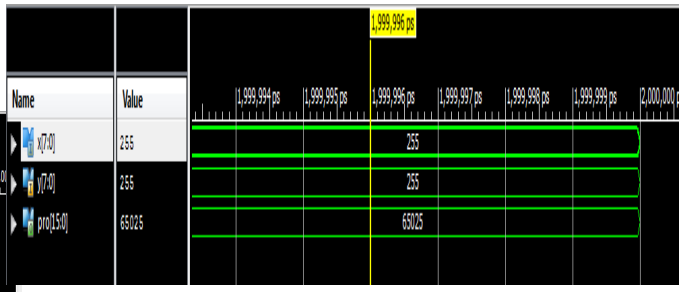

Fig.13.wallace tree multiplier using regular csla

Cell:in->out	fanout	Gate Delay	Net Delay	Logical Name (Net Name)
IBUF:I->O	2	0.715	1.040	a_8_IBUF (a_8_IBUF)
LUT4:I0->O	2	0.479	0.804	F4/A1/cout1 (F4/c<1>)
LUT3:I2->O	2	0.479	0.915	F4/A2/ap1/y1 (s1<9>)
LUT4:I1->O	3	0.479	1.066	B3/e_4_and000011 (B3/e_3_and0000)
LUT4:I0->O	1	0.479	0.740	M12/y1_SW0 (N78)
LUT4:I2->O	7	0.479	1.076	M12/y1 (c<9>)
LUT4:I1->O	1	0.479	0.851	M18/y1_SW0 (N76)
LUT4:I1->O	3	0.479	0.771	M18/y1 (c<12>)
MUXF5:S->O	3	0.540	0.771	M21/y_f5 (c<15>)
MUXF5:S->O	4	0.540	0.949	M24/y_f5 (c<18>)
LUT4:I1->O	1	0.479	0.851	M28/y1_SW0 (N74)
LUT4:I1->O	8	0.479	0.980	M28/y1 (c<21>)
LUT4:I2->O	1	0.479	0.000	M33/y12 (M33/y11)
MUXF5:I0->O	7	0.314	1.076	M33/y1_f5 (c<24>)
LUT4:I1->O	1	0.479	0.851	M39/y1_SW0 (N72)
LUT4:I1->O	1	0.479	0.681	M39/y1 (cout_OBUF)
OBUF:I->O		4.909		cout_OBUF (cout)
Total		26.189ns	(12.766ns logic, 13.423ns route) (48.7% logic, 51.3% route)	

Fig.14.delay of wallace tree multiplier using regular csla

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

Selected Device : 3s50pq208-5

 Number of Slices:                       99  out of    768    12%
 Number of 4 input LUTs:                173  out of   1536    11%
 Number of IOs:                          32
 Number of bonded IOBs:                  32  out of    124    25%
```

Fig.15.area of wallace tree multiplier using regular csla

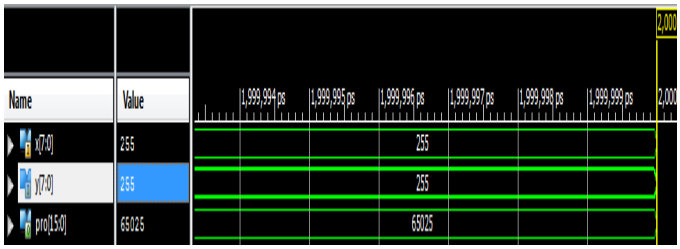

Fig.16.wallace tree multiplier using modified csla

ISSN 2348 – 7968

Cell:in->out	fanout	Gate Delay	Net Delay	Logical Name (Net Name)
IBUF:I->O	17	0.715	1.313	y_1_IBUF (y_1_IBUF)
LUT2:I1->O	2	0.479	1.040	p1_2_and00001 (p1<2>)
LUT4:I0->O	2	0.479	1.040	f8/Mxor_carry_Result1 (c<11>)
LUT4:I0->O	2	0.479	1.040	h4/carry1 (c<22>)
LUT4:I0->O	2	0.479	0.915	h10/carry1 (c<33>)
LUT4:I1->O	2	0.479	0.915	h17/Mxor_sum_Result1 (s<43>)
LUT3:I1->O	2	0.479	1.040	h26/Mxor_sum_Result1 (s<58>)
LUT3:I0->O	2	0.479	0.915	csla/M5/y11 (csla/N5)
LUT3:I1->O	2	0.479	0.804	csla/M4/y11 (csla/N11)
LUT3:I2->O	2	0.479	0.915	csla/M7/y1 (csla/c<6>)
LUT3:I1->O	2	0.479	0.915	csla/M9/y11 (csla/N4)
LUT3:I1->O	2	0.479	0.804	csla/M8/y11 (csla/N0)
LUT3:I2->O	5	0.479	0.842	csla/M11/y1 (csla/c<9>)
LUT3:I2->O	2	0.479	1.040	csla/M13/y1_SW0 (N36)
LUT4:I0->O	1	0.479	0.000	csla/M13/y21 (csla/M13/y2)
MUXF5:I1->O	1	0.314	0.681	csla/M13/y2_f5 (PRO_14_OBUF)
OBUF:I->O		4.909		PRO_14_OBUF (PRO<14>)

Total 26.868ns (12.644ns logic, 14.224ns route)
(47.1% logic, 52.9% route)

Fig.17.delay of wallace tree multiplier using modified csla

Device utilization summary:

Selected Device : 3s50pq208-5

Number of Slices:	97	out of	768	12%
Number of 4 input LUTs:	170	out of	1536	11%
Number of IOs:	32			
Number of bonded IOBs:	32	out of	124	25%

Fig.18.area of wallace tree multiplier using modified csla

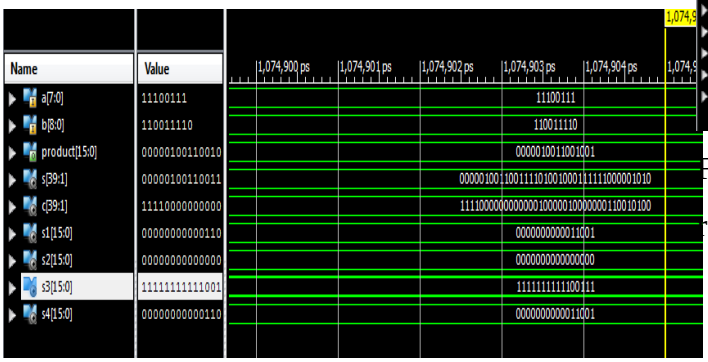

Fig.19.radix-4 booth recorded multiplier

Cell:in->out	fanout	Gate Delay	Net Delay	Logical Name (Net Name)
IBUF:I->O	26	0.715	1.716	a_1_IBUF (a_1_IBUF)
LUT4:I1->O	17	0.479	1.438	h4/temp2_add0000<4>11 (h4/N12)
LUT4:I0->O	1	0.479	0.704	h1/s1<4>104_SW0 (N56)
LUT4:I3->O	1	0.479	0.740	h1/s1<4>104 (h1/s1<4>104)
LUT3:I2->O	2	0.479	0.804	h1/s1<4>136 (s1<4>)
LUT3:I2->O	4	0.479	0.802	h7/Mxor_sum_Result1 (s<3>)
LUT4:I3->O	2	0.479	1.040	h32/carry1 (c<28>)
LUT4:I0->O	2	0.479	1.040	h33/Mxor_carry_Result1 (c<29>)
LUT3:I0->O	2	0.479	1.040	h34/Mxor_carry_Result1 (c<30>)
LUT3:I0->O	2	0.479	1.040	h35/Mxor_carry_Result1 (c<31>)
LUT3:I0->O	2	0.479	1.040	h36/Mxor_carry_Result1 (c<32>)
LUT3:I0->O	2	0.479	1.040	h37/Mxor_carry_Result1 (c<33>)
LUT3:I0->O	2	0.479	1.040	h38/Mxor_carry_Result1 (c<34>)
LUT3:I0->O	2	0.479	1.040	h39/Mxor_carry_Result1 (c<35>)
LUT3:I0->O	2	0.479	1.040	h40/Mxor_carry_Result1 (c<36>)
LUT3:I0->O	2	0.479	0.768	h41/Mxor_carry_Result1 (c<37>)
LUT4:I3->O	1	0.479	0.976	h43/Mxor_carry_xor0000_Result1 (h43/carry_xor0000)
LUT4:I0->O	1	0.479	0.681	h43/Mxor_sum_Result1 (product_15_OBUF)
OBUF:I->O		4.909		product_15_OBUF (product<15>)

Total 31.761ns (13.767ns logic, 17.994ns route)
(43.3% logic, 56.7% route)

Fig.20.delay of radix-4 booth recorded multiplier

Device utilization summary:

Selected Device : 3s50pq208-5

Number of Slices:	115	out of	768	14%
Number of 4 input LUTs:	202	out of	1536	13%
Number of IOs:	33			
Number of bonded IOBs:	33	out of	124	26%

Fig.21.area of radix-4 booth recorded multiplier

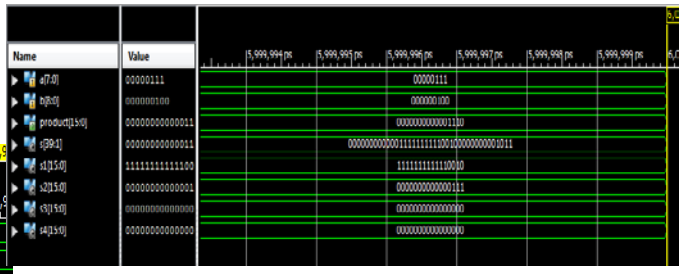

Fig.22. radix-4 booth recorded multiplier using regular csla

Timing Summary:

Speed Grade: -5

Minimum period: No path found
Minimum input arrival time before clock: No path found
Maximum output required time after clock: No path found
Maximum combinational path delay: 24.388ns

Fig.23.delay of radix-4 booth recorded multiplier using regular csla

ISSN 2348 – 7968

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

Selected Device : 3s50pq208-5

 Number of Slices:                        117  out of    768    15%
 Number of 4 input LUTs:                  205  out of   1536    13%
 Number of IOs:                            33
 Number of bonded IOBs:                    33  out of    124    26%
```

Fig.24. area of radix-4 booth recorded multiplier using regular csla

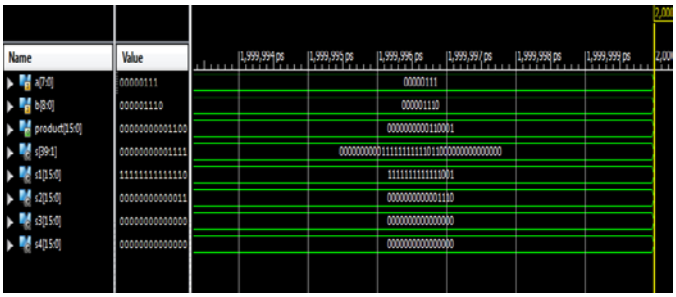

Fig.25. radix-4 booth recorded multiplier using modified csla

```
                                  Gate     Net
Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
----------------------------------------  ------------
 IBUF:I->O            26   0.715   1.716  a_1_IBUF (a_1_IBUF)
 LUT4:I1->O           17   0.479   1.438  h4/temp2_add0000<4>11 (h4/N12)
 LUT4:I0->O            1   0.479   0.851  h1/s1<4>78 (h1/s1<4>78)
 LUT4:I1->O            2   0.479   0.804  h1/s1<4>137 (s1<4>)
 LUT3:I2->O            2   0.479   1.040  h7/Mxor_carry_Result1 (c<3>)
 LUT4:I0->O            2   0.479   0.768  h21/Mxor_sum_Result1 (s<17>)
 LUT4:I3->O            2   0.479   0.915  h33/F1/A0/cout1 (h33/F1/c)
 LUT3:I1->O            3   0.479   0.941  h33/F1/A1/cout1 (h33/c<0>)
 LUT3:I1->O            1   0.479   0.740  h33/M3/y_SW0 (N22)
 LUT3:I2->O            2   0.479   0.915  h33/M3/y (h33/c<3>)
 LUT3:I1->O            2   0.479   0.915  h33/M5/y11 (h33/N3)
 LUT3:I1->O            2   0.479   0.804  h33/M4/y11 (h33/N0)
 LUT3:I2->O            4   0.479   0.838  h33/M7/y1 (h33/c<6>)
 LUT3:I2->O            2   0.479   1.040  h33/M9/y1_SW0 (N20)
 LUT4:I0->O            1   0.479   0.000  h33/M9/y21 (h33/M9/y2)
 MUXF5:I1->O           1   0.314   0.681  h33/M9/y2_f5 (product_14_OBUF)
 OBUF:I->O                 4.909          product_14_OBUF (product<14>)
----------------------------------------
Total                     27.054ns (12.644ns logic, 14.410ns route)
                                   (46.7% logic, 53.3% route)
```

Fig.26.delay of radix-4 booth recorded multiplier using modified csla

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

Selected Device : 3s50pq208-5

 Number of Slices:                        204  out of    768    15%
 Number of 4 input LUTs:                  115  out of   1536    13%
 Number of IOs:                            33
 Number of bonded IOBs:                    33  out of    124    26%
```

Fig.27.area of radix-4 booth recorded multiplier using modified csla

XI.COMPARISON TABLE

MULTIPLIER	DELAY(ns)	AREA
WALLACE TREE MULTIPLIER	28.853	320
WALLACE TREE MULTIPLIER USING REG CSLA	26.189	336
WALLACE TREE MULTIPLIER USING MOD CSLA	26.868	331
BOOTH MULTIPLIER	37.761	383
BOOTH MULTIPLIER USING REG CSLA	24.388	388
BOOTH MULTIPLIER USING MOD CSLA	27.054	385

XII. CONCLUSION

A simple approach is proposed in this paper to reduce the area and delay of SQRT CSLA architecture. The reduced number of gates of this work offers the great advantage in the reduction of area and also the delay. The compared results show that the modified SQRT CSLA has a slightly larger delay, but the area of the 16-b modified SQRT CSLA are significantly reduced respectively. The delay of the proposed design show a decrease for 16-b sizes which indicates the success of the method .The modified CSLA architecture is therefore, low area, simple and efficient for VLSI hardware implementation. Thus the modification in XOR gate can further reduces the area and no delay is observed by change in xor gate.Thus using this regular and modified CSLA in Wallace tree

ISSN 2348 – 7968

multiplier and radix-4 booth recorded multiplier shows great advantages.High speed multiplication is the speciality of using this CSLA.

XIII. REFERENCES

[1] B. Ramkumar, Harish M Kittur "Low power and Area efficient carry select adder,"IEEE Trans,Vol.20,Feb 2012.

[2] en.wikipedia.org/wiki/Wallace_tree

BIOGRAPHY

Ms. Anna Johnson received her BTech degree in Electronics and Communication Engineering from Mangalam College of Engineering, Ettumanoor in 2013 and pursuing MTech in VLSI And Embedded system in Mangalam College Of Engineering, Ettumanoor. She has attended 1 International conferences and National conferences.

Mr. Rakesh S ,Assistant professor at Mangalam College of Engineering, Ettumanoor. He done his M Tech in VLSI Design. He published a paper at IEEE conference on solid state circuits.