Implementation and Comparison of Vedic Multiplier using Area Efficient CSLA Architectures

R.Priya
PG Student
M.E Communication Systems
Mepco Schlenk Engineering College, Sivakasi, India

J.Senthil Kumar
Assistant Professor
Department of ECE
Mepco Schlenk Engineering College, Sivakasi, India

ABSTRACT

In the design of Integrated circuits, area plays a vital role because of increasing the necessity of portable systems. Carry Select Adder (CSLA) is a fast adder used in many data-processing processors for performing fast arithmetic functions. From the structure of the CSLA, the scope is reducing the area of CSLA based on the efficient gate-level modification. In this paper 4 bit, 8 bit, 16 bit, 32 bit, 64 bit and 128 bit Regular Linear CSLA, Modified Linear CSLA, Regular Square-root (SQRT) CSLA and Modified SQRT CSLA architectures have been developed and compared the area of these 4 types of CSLA and also applied these 4 bit, 8 bit, 16 bit, 32 bit, 64 bit and 128 bit CSLAs into 4X4 bit, 8X8 bit, 16X16 bit, 32X32 bit, 64X64 bit and 128X128 bit Vedic Multiplier (VM) respectively, then compared the area of Vedic Multiplier based on this adders. However, the Regular CSLA is still area consuming due to the dual Ripple Carry Adder (RCA) structure. For reducing the area of CSLA, it can be implemented by using a single RCA and an add-one circuit instead of using dual RCA. Comparing the Regular CSLA with Modified CSLA, the Modified CSLA has less area. The results and analysis show that the Modified SQRT CSLA provides better outcomes like less area and also the Vedic Multiplier using Modified Linear CSLA provides less area. This project was aimed for implementing high performance optimized FPGA architecture. Modelsim 10.0c is used for simulating the CSLAs and VM using CSLAs and synthesized using Xilinx PlanAhead13.4. Then the implementation is done in Virtex FPGA Kit.

Keywords

Area efficient, Binary to Excess-1 Converter (BEC), Carry Select Adder (CSLA), Field programmable Gate Array (FPGA), Linear CSLA, Square-root CSLA (SQRT CSLA), Vedic Multiplier (VM), Urdhva Tiryakbhyam.

1. INTRODUCTION

Design of area efficient system is one of the most essential parts of research in VLSI. The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position was summed and a carry propagated into the next position. Bedriji proposed [1] that the problem of carry propagation delay is overcome by independently generating multiple radixes carries and using this carries to select between simultaneously generated sums. Akhilash Tyagi introduced a scheme to generate carry bits with block carryin 1 from the carries of a block with block carryin 0 [4]. Chang and Hsiao proposed [3] that a Carry Select Adder scheme using BEC to replace one RCA. Youngioon Kim and Lee Sup Kim introduced a multiplexer based add one circuit was proposed to reduce the area. Yajuan He et al proposed an area efficient Square-root CSLA scheme based on a new first zero detection logic [9]. Ramkumar and Harish proposed [8] Binary to Excess-1 Converter (BEC) technique, which is an efficient and simple gate level modification to reduce the area of SQRT CSLA.

CSLA is used in many computational systems to relieve the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum [1]. However, the CSLA is not area efficient because it uses multiple pairs of RCA to generate partial sum and carry by considering carryin 0 and carryin 1, then the final sum and carry are selected by the multiplexers (Mux). The basic idea of this work is to use BEC instead of RCA with carryin 1 in the Regular CSLA to achieve less area [2], [3] and [4]. The main benefit of BEC comes from the lesser number of logic gates than the n-bit Full Adder (FA).

The word “Vedic” is derived from the word “Veda” which means store-house of knowledge. Vedic Multiplier (VM) architecture is quite different from the Conventional method of multiplication like add and shift [10]. Vedic Multiplier is based on ancient Indian Vedic Mathematics. Vedic mathematics is mainly based on 16 Sutras (formulae) like Anurupye shunyamamyat, Chalana-kalanabhyam, Ekadhikina Purvena, Ekanyunena Purvena, Gunakasamuchyah, Gunitasamuchyah, Nikhilam Navatashcaramam dashatah, Paraavartya yojavet, Puranapuranabhyam, sankalana-vyavakalanabhyam, Shesanyankena charamena, Shunyam saamyasamuccaye, Sopaantyadvayamaantyam, Urdhva Tiryakbhyam, Vyashtisamanstih and yaavadunam dealing with various branches of mathematics like arithmetic, algebra, geometry etc [11]. VM is an efficient one compared with Array Multiplier and Booth Multiplier based on area and speed [12].

Section 2 deals with the area evaluation methodology of the basic adder blocks. The details of BEC are discussed in section 3. The CSLA has been chosen for comparison with the proposed design as it has less area [5], [6]. The area evaluation methodology of the Regular Linear CSLA and Regular SQRT CSLA are presented in section 4. The area evaluation methodology of the Modified SQRT CSLA and Modified Linear CSLA are presented in section 5. The details of Vedic Multiplier using CSLAs are discussed in section 6. The FPGA implementation details and results are analyzed in section 7. Finally this work was concluded in section 8.

2. AREA EVALUATION METHOD OF BASIC ADDER BLOCKS

An XOR gate is implemented by using AND, OR and Inverter (NOT) gates. These gates are performing the operations in parallel. The area evaluation method considers all gates to be made up of AND, OR and Inverter (AOI), each having area equal to 1 unit. The area evaluation is done by counting the total number of AOI gates required for each logic block. Based on this approach, the CSLA blocks of 2:1 Mux, Half Adder (HA) and Full Adder (FA) are evaluated and listed in Table 1.

Table 1. Area count of basic blocks of CSLA

CSLA	AREA COUNT
XOR	5
2:1 Mux	4
HA	6
FA	13

3. BINARY TO EXCESS1 CONVERTER

The main idea of this work is to use BEC instead of RCA with carryin 1 in order to reduce the area of the Regular Linear CSLA as well as Regular SQRT CSLA. To replace the n-bit RCA, an n+1-bit BEC is required. Structure and the function of 3-bit BEC are shown in Fig 1 and Table 2, respectively. The Boolean expressions for 3-bit BEC is shown below (functional symbols: & AND, ~ NOT, ^ XOR**).**

$$X0 = \sim B0 \quad (1)$$

$$X1 = B0 \wedge B1 \quad (2)$$

$$X2 = B2 \wedge (B0 \,\&\, B1) \quad (3)$$

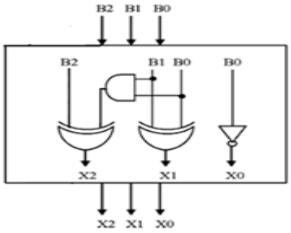


Fig 1: 3-bit Binary to Excess 1 Converter (BEC)

Table 2. Function of 3-bit BEC

B[2:0]	X[2:0]
000	001
001	010
010	011
011	100
100	101
101	110
110	111
111	000

4. AREA EVALUATION METHOD OF 16-BIT REGULAR LINEAR, SQRT CSLA

The structure of the 16-bit Regular Linear CSLA is shown in Fig 2. It has 4 groups. Each group contains same size of dual RCA and Mux. Both sum and carry are calculated for both possible solutions. The Linear CSLA is constructed by chaining a number of equal length adder stages. Here the equal size of inputs is given to each group of the adder. The steps leading to the evaluations are given here. In the Regular Linear CSLA, the group2 has two sets of 4-bit RCA. The selection input of 10:5 Mux is c3. If the c3 = 0, the Mux selects first RCA output otherwise it selects second RCA output. The output of group2 are Sum [7:4] and carryout c7. Then the area count of group2 is determined as follows:

Gate count = 117 (FA + HA + Mux)
FA = 91 (7 * 13)
HA = 6 (1 * 6)
Mux = 20 (5 * 4)

Similarly the estimated area of the other groups in the Regular Linear CSLA are evaluated and listed in Table 3.

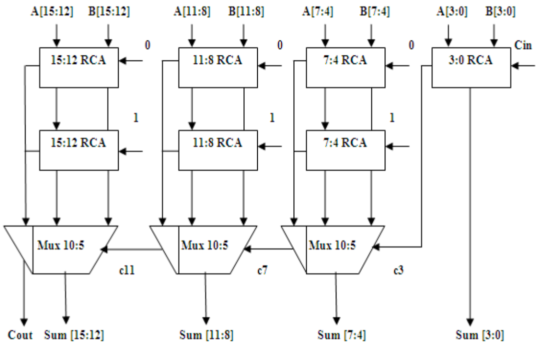


Fig 2: 16-bit Regular Linear CSLA

Table 3. Area count of 16-bit Regular Linear CSLA

GROUP	AREA COUNT
Group1	52
Group2	117
Group3	117
Group4	117

The structure of the 16-bit Regular SQRT CSLA is shown in Fig 3. It has 5 groups of different size RCA. Each group contains dual RCA and Mux. The Linear carry select adder has two disadvantages there are high area usage and high time delay. These disadvantages can be rectified by SQRT CSLA. It is an improved one of Linear CSLA. The time delay of the Linear CSLA can decrease by having one more input into each set of adders than in the previous set. This is called a Square-Root CSLA. Square-Root carry select adder is constructed by equalizing the delay through two carry chains and the block-multiplexer signal from previous stage. The steps leading to the evaluations are given here. In the Regular SQRT CSLA, the group2 has two sets of 2-bit RCA. The selection input of 3:2 Mux is c1. If the c1 = 0, the Mux select first RCA output otherwise it select second RCA output. The output of group2 are Sum [3:2] and carryout, c3. Then the area count of group2 is determined as follows:

Gate count = 57 (FA + HA + Mux)
FA = 39 (3 * 13)
HA = 6 (1 * 6)
Mux = 12 (3 * 4)

Similarly the estimated area of the other groups in the Regular SQRT CSLA are evaluated and listed in Table 4.

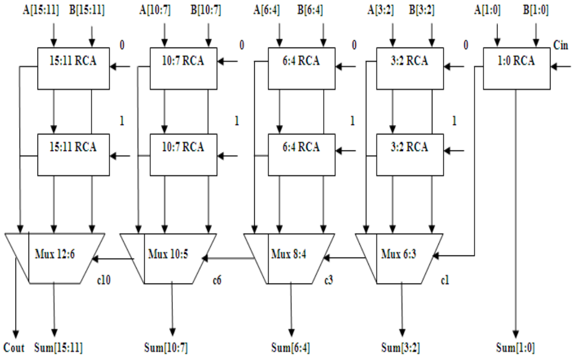


Fig 3: 16-bit Regular SQRT CSLA

Table 4. Area count of 16-bit Regular SQRT CSLA

GROUP	AREA COUNT
Group1	26
Group2	57
Group3	87
Group4	117
Group5	147

5. AREA EVALUATION METHOD OF 16-BIT MODIFIED LINEAR, SQRTCSLA

The structure of proposed i.e., Modified 16-bit Linear and SQRT CSLA using BEC instead of RCA with carryin 1 to optimize the area is shown in Fig 4 and Fig 5 respectively. The 16-bit Modified Linear CSLA has 4 groups. Each group contains same size of RCA, BEC and Mux. In the Modified Linear CSLA, the group2 has one 4-bit RCA which has 3 FA and 1 HA for carryin 0. Instead of another 4-bit RCA with carryin 1, a 5-bit BEC is used which adds one to the output of 4-bit RCA. The selection input of 10:5 Mux is c3. If the c3 = 0, the Mux select RCA output otherwise it select BEC output. The output of group2 are Sum [7:4] and carryout, c7. Then the area count of group2 is determined as follows:

Gate count = 89 (FA + HA + Mux + BEC)
FA = 39 (3 * 13)
HA = 6 (1 * 6)
Mux = 20 (5 * 4)
NOT = 1
AND = 3 (3 * 1)
XOR = 20 (4 * 5)
BEC (5-BIT) = NOT + AND + XOR = 24

Similarly the estimated area of the other groups in the Modified Linear CSLA are evaluated and listed in Table 5

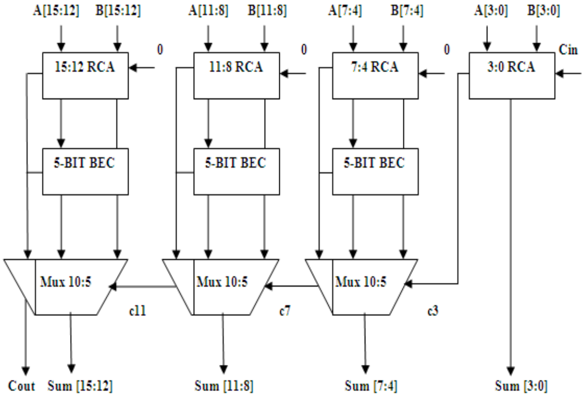


Fig 4: 16-bit Modified Linear CSLA

Table 5. Area count of 16-bit Modified Linear CSLA

GROUP	AREA COUNT
Group1	52
Group2	89
Group3	89
Group4	89

The structure of the 16-bit Modified SQRT CSLA is shown in Fig 5. It has 5 groups of different size of RCA and BEC. Each group contains one RCA, one BEC and Mux. In the Modified SQRT CSLA, the group2 has one 2-bit RCA which has 1 FA and 1 HA for carryin 0. Instead of another 2-bit RCA with carryin 1, a 3-bit BEC is used which adds one to the output of 2-bit RCA. The selection input of 6:3 Mux is c3. If the c3 = 0, the Mux selects RCA output otherwise it selects BEC output. The output of group2 are Sum [3:2] and carryout, c3. Then the area count of group2 is determined as follows:

Gate count = 43 (FA + HA + Mux + BEC)
FA = 13 (1 * 13)
HA = 6 (1 * 6)
Mux = 12 (3 * 4)
NOT = 1
AND = 1
XOR = 10 (2 * 5)
BEC (3-BIT) = NOT + AND + XOR = 12

Similarly the estimated area of the other groups in the Modified SQRT CSLA are evaluated and listed in Table 6.

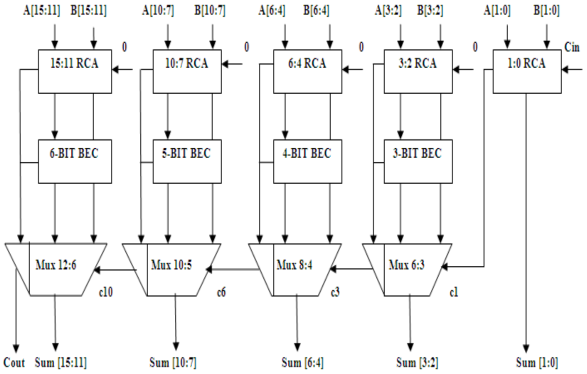


Fig 5: 16-bit Modified SQRT CSLA

Table 6. Area count of 16-bit Modified SQRT CSLA

GROUP	AREA COUNT
Group1	26
Group2	43
Group3	66
Group4	89
Group5	113

Comparing Tables 3 and 4 with Tables 5 and 6, it is clear that the proposed Modified Linear as well as SQRT CSLA save 84 gate and 97 gate areas than the Regular Linear CSLA and Regular SQRT CSLA respectively.

6. VEDIC MULTIPLIER

In this paper Vedic Multiplier is based on Urdhva Tiryakbhyam (multiplication sutra) of ancient Indian Vedic Mathematics. Urdhva Tiryakbhyam means “Vertical and crosswise” respectively. This strikes the difference in the actual process of multiplication itself. This formula enables parallel generation of partial products and eliminates unwanted multiplication steps. Vedic Multiplier has one advantage that is the number of bits increases, area and gate delay increases very slowly as compared to other multipliers. Therefore it is time, power and area efficient. In Fig 6, the digits on the both side of the line are multiplied and added with the carry of the previous step. It generates one of the bits of the result and a carry. This carry is added with the next step multiplication result and hence the process goes on. If more than one line present in one step, all the results are added to the carry of previous step. In each step, least significant bit (LSB) acts as the result bit and all other bits act as carry for the next step. The multiplication of two 2X2 bit and 4X4 bit binary numbers as shown in Fig 6 and Fig 7 respectively.

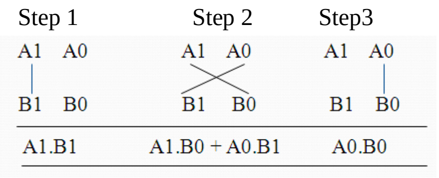


Fig 6: 2X2 bit binary multiplication using Urdhva Tiryakbhyam

$$S0 = A0B0 \tag{4}$$

$$C1S1 = A1B0 + A0B1 \tag{5}$$

$$C2S2 = C1 + A1B1 \tag{6}$$

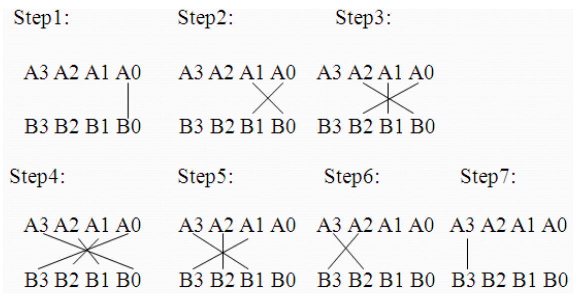


Fig 7: 4X4 bit binary multiplication using Urdhva Tiryakbhyam

$$S0 = A0B0 \tag{7}$$

$$C1S1 = A1B0 + A0B1 \tag{8}$$

$$C2S2 = C1 + A1B1 + A2B0 + A0B2 \tag{9}$$

$$C3S3 = C2 + A3B0 + A0B3 + A1B2 + A2B1 \tag{10}$$

$$C4S4 = C3 + A3B1 + A1B3 + A2B2 \tag{11}$$

$$C5S5 = C4 + A3B2 + A2B3 \tag{12}$$

$$C6S6 = C5 + A3B3 \tag{13}$$

Final result: C6S6S5S4S3S2S1S0

VM consists of 3 stages. The 1st stage consists of Multiplication unit, 2nd stage consists of partial products and carry and the 3rd stage consists of adder and the result of multiplication. In the adder block of 3rd stage, CSLA can be used in this paper.

6.1 Basic block of Vedic Multiplier

In the design of Vedic multiplier, a 2X2 bit block is a fundamental block is shown in Fig 8. Also this fundamental block is used as a Multiplication unit block of 4X4 bit Vedic Multiplier.

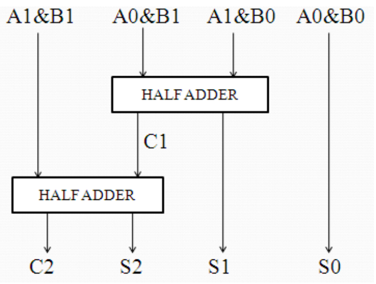


Fig 8: fundamental block of VM

Let us consider the two 2-bit binary numbers A1A0 and B1B0. The result of this 2X2 bit multiplication would be 4 bits that is C2, S2, S1 and S0. The least significant bit A0 of multiplicand is multiplied vertically by least significant bit B0 of the multiplier, get their product S0 and this S0 is the least significant part of result (S0). Then A1 and B0, and A0 and B1 are multiplied crosswise, add the two, get sum1 (S1) and carry1 (C1), the sum bit is the middle part of the result (S1). Then A1 and B1 is multiplied vertically, and add with the previous carry (C1) and get S2 as their product and carry2 (C2), the sum bit is down to the result (S2). Then the carry2 (C2) is taken as most significant part of the result (S3).

6.2 4X4 bit Vedic Multiplier

The design of 4X4 bit VM shown in Fig 9. In this design, the 1st stage consists of 2X2 bit block (fundamental block) as a Multiplication unit. The 2nd stage consists of partial products and carry. Then the 3rd stage consists of 4 bit CSLA (4 bit Regular Linear CSLA , 4 bit Modified Linear CSLA, 4 bit Regular SQRT CSLA and 4 bit Modified SQRT CSLA) and 8 bit result of multiplication. The first step in the design of 4X4 bit VM will be grouping the 2 bit of each 4 bit input. These pairs will form vertical and crosswise product terms. Each input bit pairs are handled by a separate 2X2 bit VM block. The schematic of a 4X4 bit block designed using 2X2 bit blocks. The partial products represent the Urdhva vertical and cross product terms.

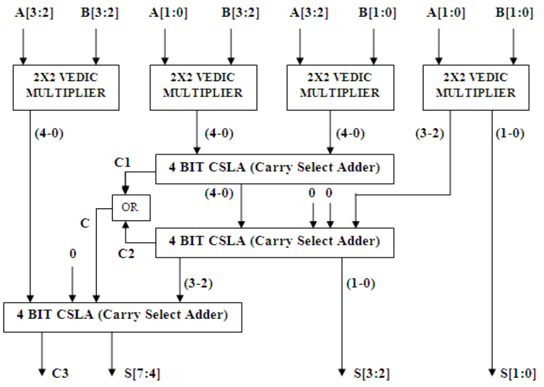


Fig 9: Block diagram of 4X4 bit Vedic Multiplier

6.3 8X8 bit Vedic Multiplier

The design of 8X8 bit VM shown in Fig 10. In this design, the 1st stage consists of 4X4 bit block as a Multiplication unit. The 2nd stage consists of partial products and carry. Then the 3rd stage consists of 8 bit CSLA (8 bit Regular Linear CSLA, 8 bit Modified Linear CSLA, 8 bit Regular SQRT CSLA and 8 bit Modified SQRT CSLA) and 16 bit result of multiplication. The first step in the design of 8X8 bit block VM will be grouping the 4 bit of each 8 bit input. These pair will form the vertical and crosswise product terms. Each input bit pairs are handled by a separate 4X4 bit VM block. The schematic of an 8X8 bit VM designed using 4X4 bit VM blocks.

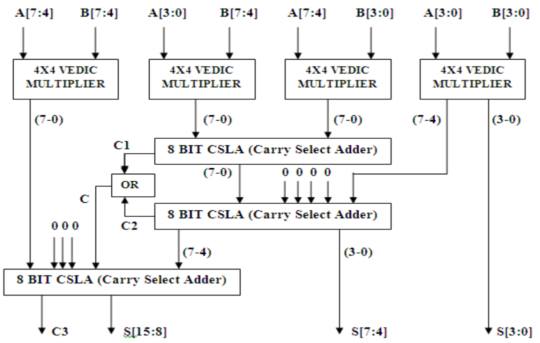


Fig 10: Block diagram of 8X8 bit Vedic Multiplier

6.4 16X16 bit Vedic Multiplier

The design of 16X16 bit VM shown in Fig 11. In this design, the 1st stage consists of 8X8 bit VM blocks as a Multiplication unit. The 2nd stage consists of partial products and carry. Then the 3rd stage consists of 16 bit CSLA (16 bit Regular Linear CSLA, 16 bit Modified Linear CSLA, 16 bit Regular SQRT CSLA and 16 bit Modified SQRT CSLA) and 32 bit result of multiplication. The first step in the design of 16X16 bit VM will be grouping the 8 bit of each 16 bit input. These pairs will form vertical and crosswise product terms. Each input bit pairs are handled by a separate 8X8 VM block. The schematic of 16X16 bit VM block designed using 8X8 bit VM blocks.

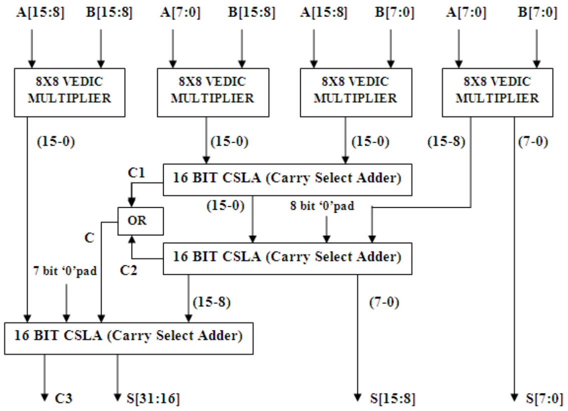


Fig 11: Block diagram of 16X16 bit Vedic Multiplier

6.5 32X32 bit Vedic Multiplier

The design of 32X32 bit VM shown in Fig 12. In this design, the 1st stage consists of 16X16 bit VM blocks as a Multiplication unit. The 2nd stage consists of partial products and carry. Then the 3rd stage consists of 32 bit CSLA (32 bit Regular Linear CSLA, 32 bit Modified Linear CSLA, 32 bit Regular SQRT CSLA and 32 bit Modified SQRT CSLA) and 64 bit result of multiplication. The first step in the design of 32X32 bit VM will be grouping the 16 bit of each 32 bit input. These pairs will form vertical and crosswise product terms. Each input bit pairs are handled by a separate 16X16 bit VM block. The schematic of 32X32 bit VM block designed using 16X16 bit VM blocks.

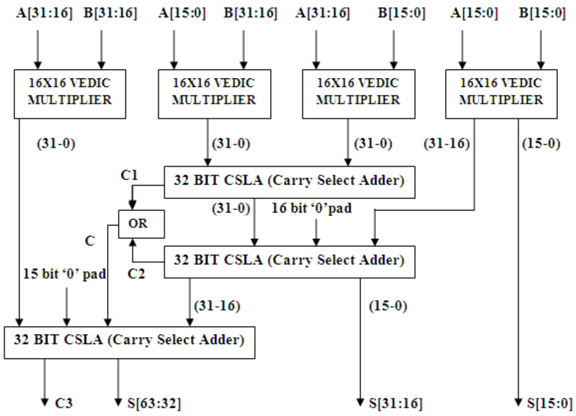


Fig 12: Block diagram of 32X32 bit Vedic Multiplier

6.6 64X64 bit Vedic Multiplier

The design of 64X64 bit VM shown in Fig 13. In this design, the 1st stage consists of 32X32 bit VM blocks as a Multiplication unit. The 2nd stage consists of partial products and carry. Then the 3rd stage consists of 64 bit CSLA (64 bit Regular Linear CSLA, 64 bit Modified Linear CSLA, 64 bit Regular SQRT CSLA and 64 bit Modified SQRT CSLA) and 128 bit result of multiplication. The first step in the design of 64X64 bit VM block will be grouping the 32 bit of each 64 bit input. These pairs will form vertical and crosswise product terms. Each input bit pairs are handled by a separate 32X32 bit VM block. The schematic of a 64X64 bit VM designed using 32X32 bit VM blocks.

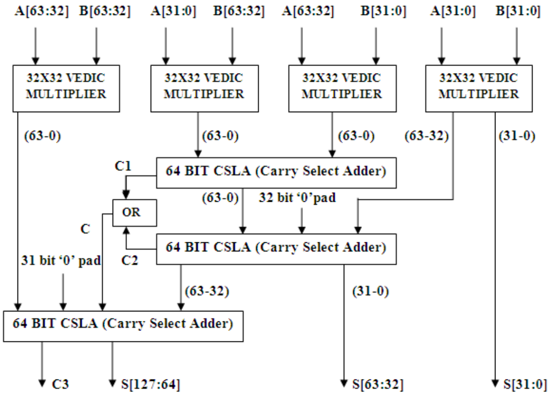


Fig 13: Block diagram of 64X64 bit Vedic Multiplier

6.7 128X128 bit Vedic Multiplier

The design of 128X128 bit VM shown in Fig 14. In this design, the 1st stage consists of 64X64 bit VM blocks as a Multiplication unit. The 2nd stage consists of partial products and carry. Then the 3rd stage consists of 128 bit CSLA (128 bit Regular Linear CSLA, 128 bit Modified Linear CSLA, 128 bit Regular SQRT CSLA and 128 bit Modified SQRT CSLA) and 256 bit result of multiplication. The first step in the design of 128X128 bit VM will be grouping the 64 bit of each 128 bit input. These pairs will form vertical and crosswise product terms. Each input bit pairs are handled by a separate 64X64 bit VM block. The schematic of 128X128 bit VM block designed using 64X64 bit VM blocks.

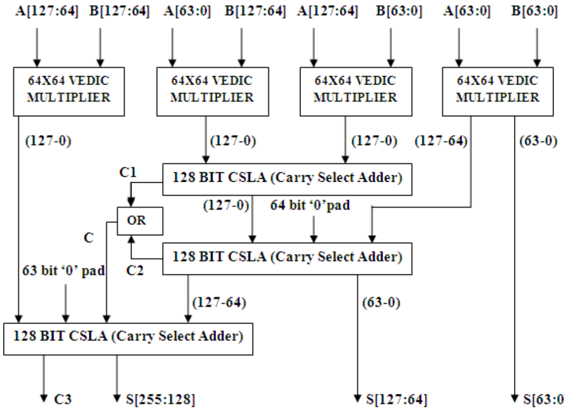
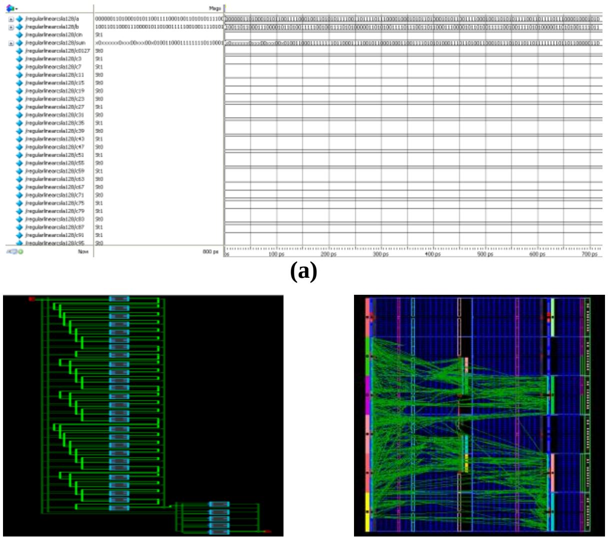


Fig 14: Block diagram of 128X128 bit Vedic Multiplier

7. FPGA IMPLEMENTATION RESULT

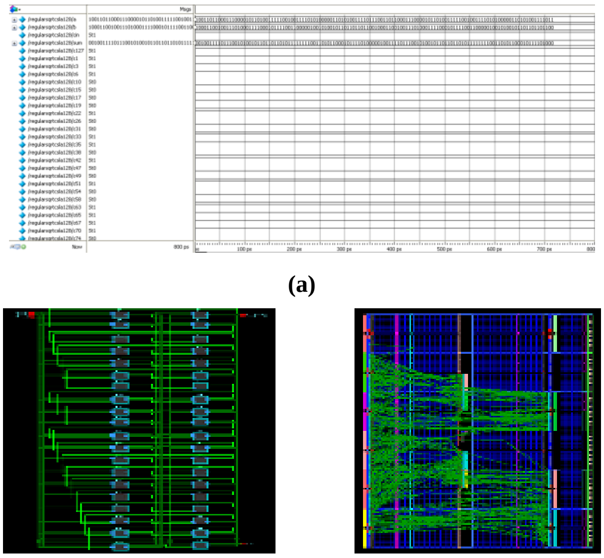
This work has been developed using Verilog-HDL. It was simulated using Modelsim Altera 10.0c and synthesized using Xilinx PlanAhead 13.4. This design was implemented in Virtex kit. Fig 15(a), 15(b), 15(c), Fig 16(a), 16(b), 16(c), Fig 17(a), 17(b), 17(c) and Fig 18(a), 18(b), 18(c) show the Simulation, RTL schematic view and FPGA Editor results of Regular 128 bit Linear CSLA, Regular 128-bit SQRT CSLA, Modified 128-bit Linear CSLA and Modified 128-bit SQRT CSLA respectively. Table 7 exhibits the area count of all types of CSLA structures. The gate reduction in the area as a function of the bit size is shown in Fig 19. It is clear that the area of the 16-bit, 32-bit, 64-bit and 128-bit proposed Modified SQRT CSLA has less area when compared with the other CSLAs. The Fig 20(a), 20(b), 20(c), Fig 21(a), 21(b), 21(c), Fig 22(a), 22(b), 22(c) and Fig 23(a), 23(b), 23(c) show the Simulation, RTL schematic view and FPGA Editor results of VM using Regular 128 bit Linear CSLA, VM using Regular 128-bit SQRT CSLA, VM using Modified 128-bit Linear CSLA and VM using Modified 128-bit SQRT CSLA respectively. Table 8 exhibits the area count of all types of VM using CSLAs structures. The gate reduction in the area as a function of the bit size is shown in Fig 24. From the result analysis, it is clear that the area of the 4x4 bit, 8X8 bit, 16X16 bit, 32X32 bit, 64X64 bit and 128X128 bit Vedic Multiplier using Modified Linear CSLA has area efficient when compared with the other VM using CSLAs.



(a)

(b) (c)

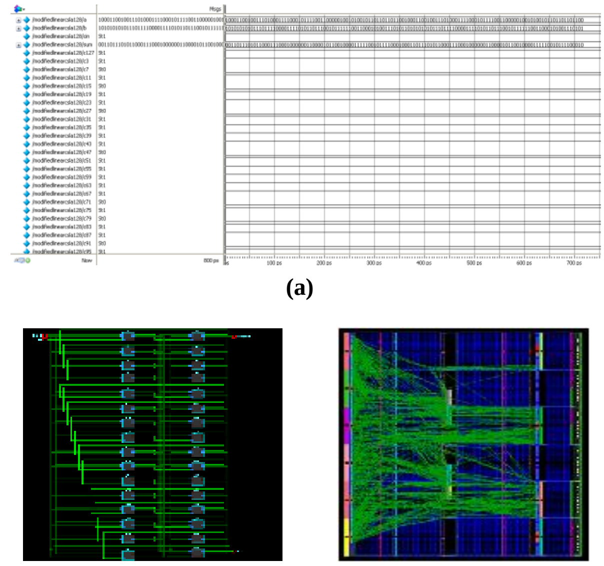
Fig 15: Result of 128 bit Regular Linear CSLA (a) Simulation waveform (b) RTL schematic view and (c) FPGA Editor



(a)

(b) (c)

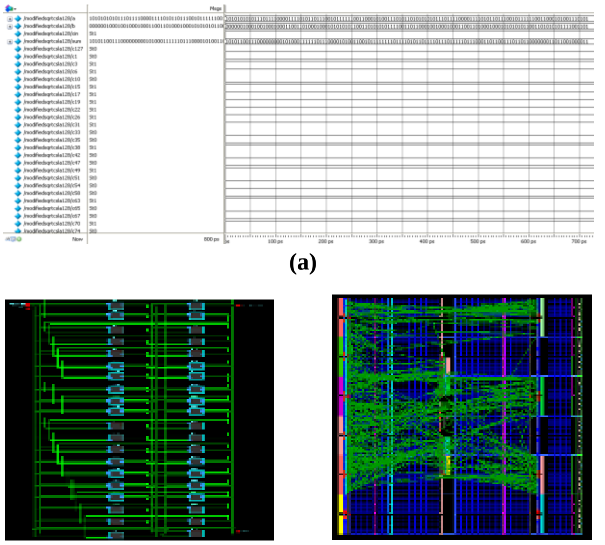
Fig 16: Result of 128 bit Regular SQRT CSLA (a) Simulation waveform (b) RTL schematic view and (c) FPGA Editor



(a)

(b) (c)

Fig 17: Result of 128 bit Modified Linear CSLA (a) Simulation waveform (b) RTL schematic view and (c) FPGA Editor



(a)

(b) (c)

Fig 18: Result of 128 bit Modified SQRT CSLA (a) Simulation waveform (b) RTL schematic view and (c) FPGA Editor

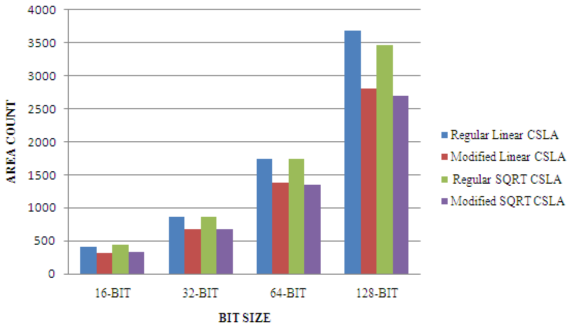
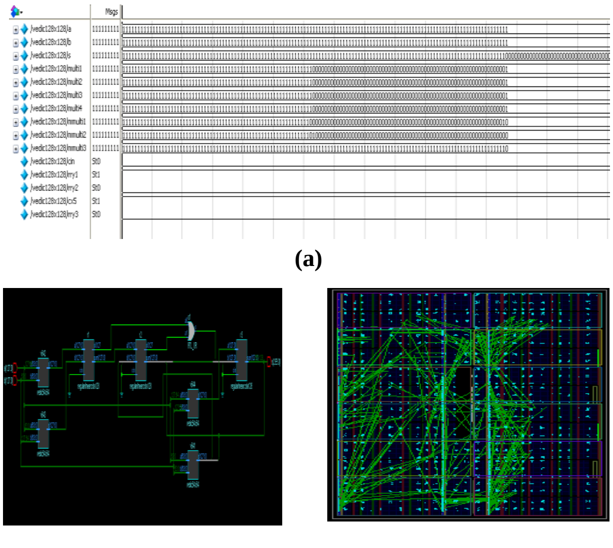


Fig 19: Comparison of CSLAs based on Gate count (Area)

Table 7. Comparison of CSLAs based on Area (gate count)

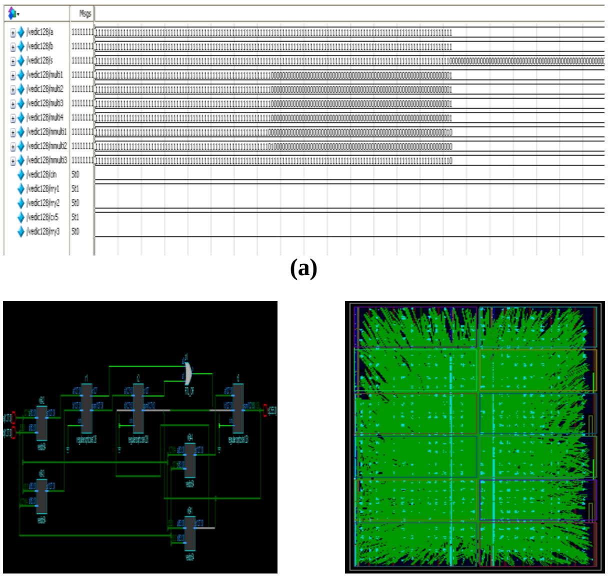
Bit size	Types	GATE COUNT	
		Regular CSLA	Modified CSLA
16 bit	LINEAR	403	319
	SQRT	434	337
32 bit	LINEAR	871	675
	SQRT	868	674
64 bit	LINEAR	1742	1387
	SQRT	1736	1348
128 bit	LINEAR	3679	2811
	SQRT	3472	2696



(a)

(b) (c)

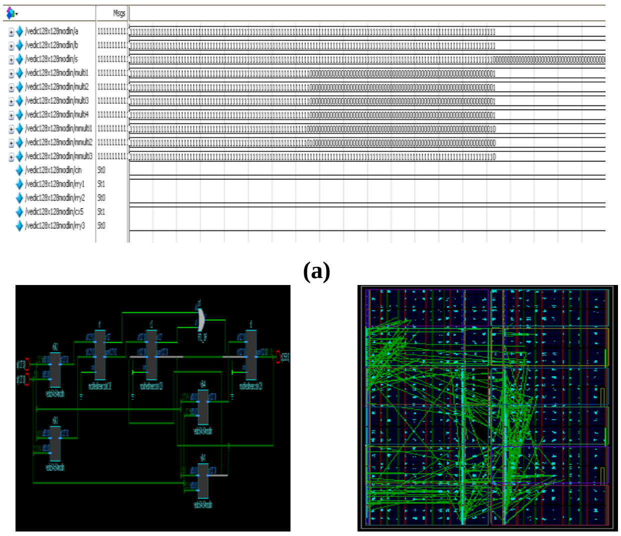
Fig 20: Result of 128X128 bit VM using 128 bit Regular Linear CSLA (a) Simulation waveform (b) RTL schematic view and (c) FPGA Editor



(a)

(b) (c)

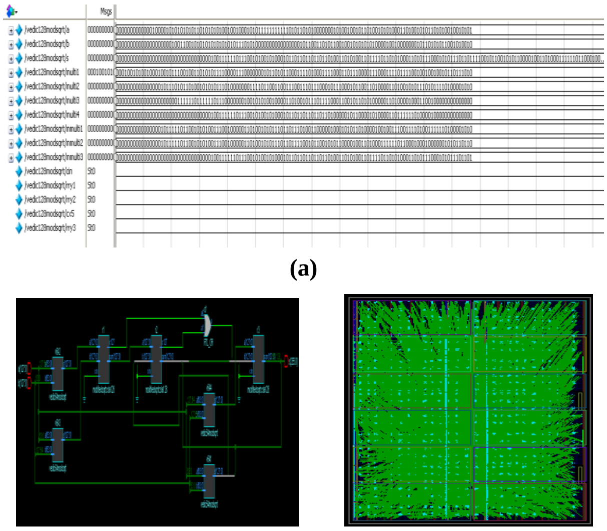
Fig 21: Result of 128X128 bit VM using 128 bit Regular SQRT CSLA (a) Simulation waveform (b) RTL schematic view and (c) FPGA Editor



(a)

(b) (c)

Fig 22: Result of 128X128 bit VM using 128 bit Modified Linear CSLA (a) Simulation waveform (b) RTL schematic view and (c) FPGA Editor



(a)

(b) (c)

Fig 23: Result of 128X128 bit VM using 128 bit Modified SQRT CSLA (a) Simulation waveform (b) RTL schematic view and (c) FPGA Editor

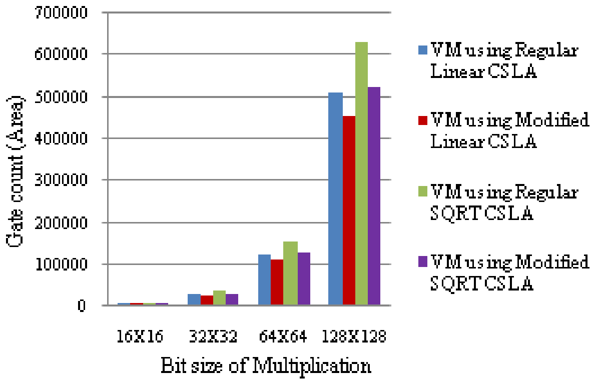


Fig 24: Comparison of Vedic Multiplier using CSLAs based on Area (gate count)

Table 8. Comparison of Vedic Multiplier using CSLAs based on Area (gate count)

Size of Multipli cation	GATE COUNT (AREA)			
	VM using Regular Linear CSLA	VM using Modified Linear CSLA	VM using Regular SQRT CSLA	VM using Modified SQRT CSLA
4X4 bit	221	221	314	272
8X8 bit	1,392	1,308	1,857	1,563
16X16 bit	6,778	6,190	8,731	7,264
32X32 bit	29,726	26,786	37,529	31,079
64X64 bit	1,24,131	1,11,306	1,55,325	1,28,361
128X128 bit	5,07,562	4,53,658	6,31,717	5,21,533

8. CONCLUSION

This paper presented a simple approach to reduce the area of CSLA architectures; comparison of CSLAs based Area and applying these CSLAs into Vedic Multiplier and also comparing the area of VM. The reduced number of gates of this work offers the great advantage in the reduction of area. The area (gate count) of the 128 bit Regular SQRT CSLA is significantly reduced by 207 gates when compared with 128 bit Regular Linear CSLA and also the comparison between 128 bit Regular Linear CSLA and 128 bit Modified Linear CSLA, the area of the 128 bit Modified Linear CSLA is reduced by 868 gates. Then the area of the 128 bit Modified SQRT CSLA is reduced by 115 gates than the area of the 128 bit Modified Linear CSLA. Finally the comparison between the area of 128 bit Regular SQRT CSLA and the area of the 128 bit Modified SQRT CSLA, the area of 128 bit Modified SQRT CSLA is reduced by 776 gates than the 128 bit Regular SQRT CSLA. Totally from the result analysis the Modified SQRT CSLA has less area. Then applied these 4 types of CSLA into Vedic Multiplier's adder block, and compared the area of VM based on these 4 types of CSLA. From the results the area of VM using Modified Linear CSLA has less area. The Modified CSLA architecture is therefore, less area, simple and efficient for VLSI hardware implementation.

9. REFERENCES

[1] Bedrij, O. J. 1962, Carry-select adder, IRE Trans. Electron. Computer, pp. 340–344.

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

[3] Ceiang, T. Y., and Hsiao, M. J. 1998, Carry-select adder using single ripple carry adder Electron. Lett, vol. 34, no. 22, pp. 2101–2103.

[4] Kim, Y., and Kim, L. S., 2001, 64-bit carry-select adder with reduced area Electron. Lett. vol. 37, no. 10, pp. 614–615.

[5] Rabaey, J. M., 2001, Digital Integrated Circuits—A Design Perspective. Upper Saddle River, NJ: Prentice-Hall.

[6] He, Y., Chang, C. H., and Gu, J., 2005, An area efficient 64-bit square Root carry-select adder for low power applications, in proc. IEEE Int. Symp. Circuits Syst., vol. 4, pp. 4082-4085.

[7] Cadence, 2008, Encounter user guide, Version 6.2.4.

[8] Ramkumar, B., and Harish M Kittur, 2012, Low Power and Area Efficient Carry Select Adder, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, pp.1-5.

[9] Padma Devi, Ashima Girdher and Balwinder Singh, 1998, Improved Carry Select Adder with Reduced Area and Low Power Consumption, International Journal of Computer Applications, Vol.3, No.4, pp. 14-18.

[10] Pushpalata Verma, K. K., Mehta, 2012, Implementation of an Efficient Multiplier based on Vedic Mathematics Using EDA Tool, International Journal of Engineering and Advanced Technology (IJEAT).

[11] Ganesh Kumar, G., Charishma, V., 2012, Design of High Speed Vedic Multiplier using Vedic Mathematics Techniques, International Journal of Scientific and Research Publications.

[12] Vaijyanath Kunchigi, Linganagoda Kulkarni and Subhash Kulkarni, 2012, High Speed and Area Efficient Vedic Multiplier.

[13] Akhilesh Tyagi, 1993, A Reduced-Area Scheme for Carry-Select Adders, IEEE Transactions on Computers, Vol.42, No.10, pp.1163-1170.

[14] Edison, A. J., and Manikanda babu, C. S., 2012, An Efficient CSLA Architecture for VLSI Hardware Implementation" International Journal for Mechanical and Industrial Engineering, Vol. 2. Issue 5.

[15] Sreenivasulu, P., Srinivasa Rao, P., Malla Reddy, and Vinay Babu, A., 2012, Energy and Area efficient Carry Select Adder on a reconfigurable Hardware, International Journal of Engineering Research and Applications, Vol. 2, Issue 2, pp.436-440.

[16] Sarabdeep Singh and Dilip Kumar, 2011, Design of Area and Power efficient Modified Carry Select Adder, International Journal of Computer Applications (0975 – 8887) Volume 33– No.3.