AN NOVEL VLSI ARCHITECTURE FOR URDHVA TIRYAKBHYAM VEDIC MULTIPLIER USING EFFICIENT CARRY SELECT ADDER

S. Srikanth[1], A. Santhosh Kumar[2], R. Lokeshwaran[3], A. Anandhan [4]

[1,2] Assistant Professor, Department of Electronics & Communication Engg, SNSCT, Coimbatore (India)

[3,4] Department of Electronics & Communication Engineering, SNSCT, Coimbatore (India)

ABSTRACT

In the design of integrated circuits, area plays a vital role because of increasing the necessity of portable systems. CSLA is a fast adder used in many data processing processors for performing fast arithmetic functions. In this paper 4bit, 8bit, 16bit, 32bit, 64bit and 128 bit CSLA have been developed and applied in 4x4 bit, 8x8 bit, 16x16 bit, 32x32 bit, 64x64 bit and 128x128 bit Vedic multiplier respectively.

This paper proposed the design of high speed Vedic Multiplier using the techniques of Ancient Indian Vedic Mathematics that have been modified to improve performance.Vedic method for multiplication which strikes a difference in the actual process of multiplication itself. It enables parallel generation of intermediate products, eliminates unwanted multiplication steps with zeros. The regular CSLAconsumes more areaand hence modified linear CSLA is used which requires less area. For simulating the CSLA's and Vedic multipliers using CSLA's.ISIM simulator is used. Further, the Verilog HDL coding of 128x128 bits multiplication and their FPGA implementation by Xilinx Synthesis Tool on Spartan 3E kit have been done and output has been displayed on LCD of Spartan 3E kit.

Keywords: Binary to Excess-1 Converter (BEC), Carry Select Adder (CSLA), SQRT CSLA, Vedic Multiplier (VM), Urdhva Tiryakbhyam.

1. INTRODUCTION

Multiplication is a crucial arithmetic operation in digital computer systems. Most modern computers directly support multiplication based operations in hardware . Moreover, the performance of processors is significantly affected by the speed of their multipliers. Multiplier has been a basic building block for many algorithms in various computing application. Many high performance algorithms and architectures have been proposed to improve and to accelerate multiplications process. Multiplication involves two basic operations: the generation of partial products and their accumulation. Consequently, there two ways to speed up multiplications: reduce the number of partial products or accelerate their accumulation. Clearly, a smaller number of partial products also reduces the complexity, and, as a

result, reduces the partial products accumulation time. When two n-bit numbers are multiplied, a 2*n*-bit product is produced.

In that case the steps involved in the multiplication process are large and the addition of n partial products also becomes a complicated task. Different schemes of multiplication have been developed to reduce the number of steps required to add the n partial products.The word usage ―Vedic is taken from the word ―Veda which refers to the store-house of knowledge. Vedic Multiplier (VM) architecture differs by lot of means from the Conventional technique of multiplication such as add and shift [12]. Vedic Multiplier has its grounds on ancient Indian Vedic Mathematics. Vedic mathematics is basically upon 16 Sutras (formulae) like Anurupyeshunyamamyat, Chalana-kalanabhyam, EkadhikinaPurvena, EkanyunenaPurvena, Gunakasamuchyah, Gunitasamuchyah, NikhilamNavatashcaramamdashatah, Paraavartyayojavet, Puranapuranabhyam, sankalana- vyavakalanabhyam, Shesanyankenacharamena, Shunyamsaamyasamuccaye, Sopaantyadvayamaantyam, UrdhvaTiryakbhyam, Vyashtisamanstih and yaavadunam that deals with v arious branches of mathematics like arithmetic, algebra, geometry etc [10]. Vedic Multiplier (VM) is reliable and an efficient one compared with Array Multiplier and Booth Multiplier based on area and speed

II BINARY TO EXCESS 1 CONVERTER

The ground idea regarding this work is using BEC instead of RCA with carryin 1 to reduce the area of the Regular SQRT CSLA as well as Modified SQRT CSLA. For replacing the n-bit RCA, an n+1-bit BEC is required. General structure and the basic function of 3-bit BEC are shown in Fig 1 and Table 1, respectively. Boolean expressions for 3-bit BEC is as given below (a),(b),(c) (functional symbols: & AND, ~ NOT, ^ XOR).

Table 1. Function of 3-bit BEC

B [2 : 0]			X S [2 : 0]		
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

XS0=~B0 - (a)

XS1=B0^B1 -(b)

XS2=B2^(B0&B1) -(c)

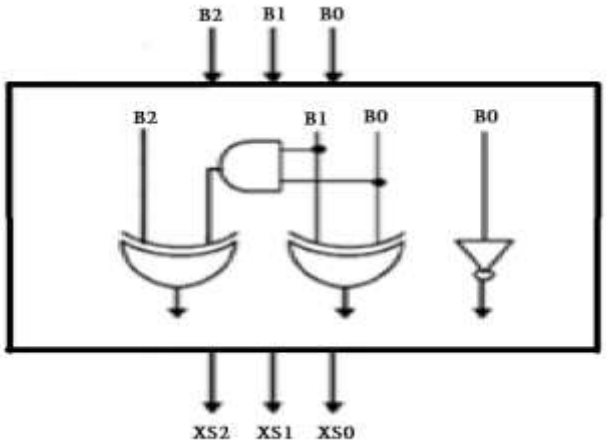

Fig 1. 3-bit Binary to Excess 1 Converter

III PROPOSED METHOD OF CARRY SELECT ADDER

The proposed CSLA is based on the logic formulation givenin (a)–(g), and its structure is shown in Fig. 2(a). It consists ofone HSG unit, one FSG unit, one CG unit, and one CS unit. TheCG unit is composed of two CGs (CG0 and CG1) correspondingto input-carry '0' and '1'. The HSG receives two *n*-bit operands(*A* and *B*) and generate *half-sum* word *s*0 and *half-carry* word*c*0 of width *n* bits each. Both CG0 and CG1 receive *s*0 and *c*0from the HSG unit and generate two *n*-bit full-carry words *c*01 and*c*11corresponding to input-carry '0' and '1', respectively.The logic diagram of the HSG unit is shown in Fig. 2(b). Thelogic circuits of CG0 and CG1 are optimized to take advantageof the fixed input-carry bits. The optimized designs of CG0 and CG1 are shown in Fig. 2(c) and 2(d), respectively.The CS unit selects one final carry word from the two carrywords available at its input line using the control signal *c*in. It selects *c*01 when *c*in = 0; otherwise, it selects *c*11. The CS unitcan be implemented using an *n*-bit 2-to-l MUX. However, we find from the truth table of the CS unit that carry words *c*01and*c*11 follow a specific bit pattern. If *c*01(*i*) = '1', then *c*11 (*i*) = 1, irrespective of *s*0(*i*) and *c*0(*i*), for $0 \leq i \leq n - 1$. This featureis used for logic optimization of the CS unit. The optimized design of the CS unit is shown in Fig. 2(e), which is composedof*n* AND–OR gates. The final carry word *c* is obtained from the CS unit. The MSB of *c* is sent to output as *c*out, and ($n - 1$)LSBs are XORed with ($n - 1$) MSBs of *half-sum* (*s*0) in the FSG [shown in Fig. 2(f) to obtain ($n - 1$) MSBs of *final-sum*(*s*). The LSB of *s*0 is XORed with *c*in to obtain the LSB of *s*

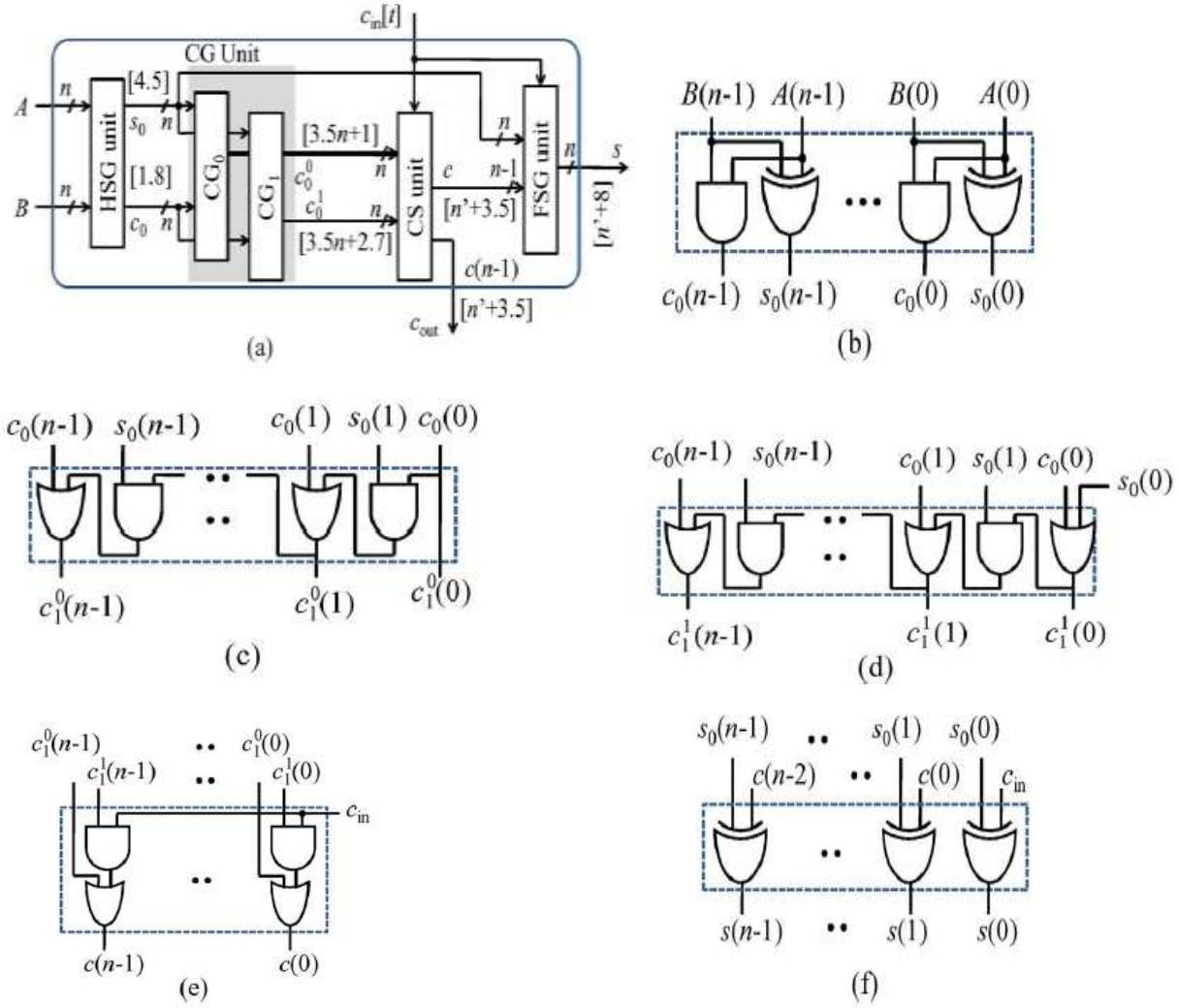

Fig.2 (a) Proposed CS adder design, where *n* is the input operand bit-width, and [*] represents delay (in the unit of inverter delay), *n* = max(*t*, 3.5*n* + 2.7).

(b) Gate-level design of the HSG. (c) Gate-level optimized design of (CG0) for input-carry = 0. (d) Gate-level optimized design of (CG1) for input-carry = 1.

(e) Gate-level design of the CS unit. (f) Gate-level design of the final-sum generation (FSG) unit.

IV VEDIC MULTIPLIER

In this discussion the Vedic Multiplier (VM) is mainly based on UrdhvaTiryakbhyam (multiplication sutra) of ancient Indian Vedic Mathematics. UrdhvTiryakbhyam generally means Vertical and crosswise respectively. This strikes the contrary in the actual multiplication process itself. This formula help proceeding parallel generation of partial products and eliminates unwanted multiplication process steps. Vedic Multiplier has a merit that as the number of bits increases, area and gate delay also increases too slowly compared to other multipliers. Therefore it is

the time, power and area that are efficient. In Fig 3, the digits on the either side of the line are multiplied and added by the carry of the preceding step. It does generate a carry and one of the bits of the results. This carry is to be added with the next step multiplication result and so the process goes on moving. If more than one line is seen in one step, all obtained results are then added to the carry of previous step. In each step bit and all other bits act as carry for the next step and the least significant bit (LSB) acts as the result. The multiplication of two 4X4 bit and 2X2 bit binary numbers are as depicted in Fig 3 and Fig 4 respectively.

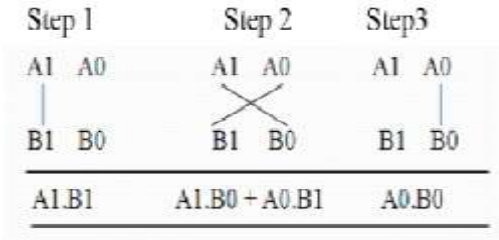

Fig 3: 2X2 bit binary multiplication using Urdhva Tiryakbhyam

S0 = A0B0

C1S1 = A1B0 + A0B1

C2S2 = C1 + A1B1

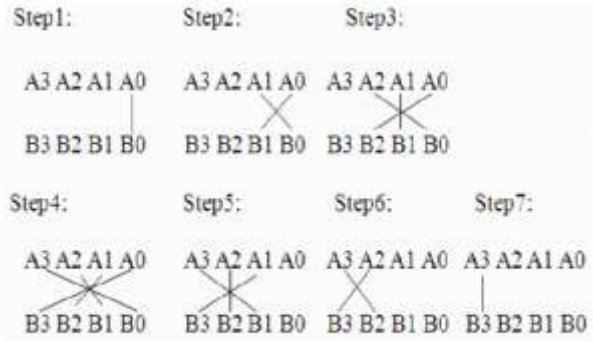

Fig 4: 4X4 bit binary multiplication using Urdhva Tiryakbhyam

S0 = A0B0

C1S1 = A1B0 + A0B1

C2S2 = C1 + A1B1 + A2B0 + A0B2

C3S3 = C2 + A3B0 + A0B3 + A1B2 + A2B1

C4S4 = C3 + A3B1 + A1B3 + A2B2

C5S5 = C4 + A3B2 + A2B3

C6S6 = C5 + A3B3

Final result: C6S6S5S4S3S2S1S0

4.1. Basic block of Vedic Multiplier

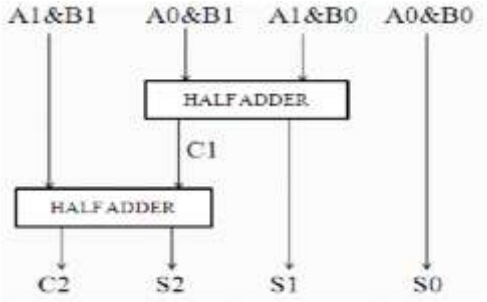

Fig 5 : fundamental block of VM

In Vedic multiplier design, a 2X2 bit block is one of the fundamental block is shown in Fig 5. Also this same fundamental block is taken as a Multiplication unit block of 4X4 bit Vedic Multiplier. Considering the two 2-bit binary numbers A1A0 and B1B0. The obtained result of this 2X2 bit multiplication would be found as 4 bits that is C2, S2, S1 and S0. The least significant bit (LSB) A0 of the multiplicand is vertically multiplied with least significant bit(LSB) B0 of the multiplier, To obtain their product S0 and this S0 is the least significant part of obtained result (S0). Then A1 and B0, and A0 and B1 are multiplied in cross, add the two, get sum1 (S1) and carry1 (C1), the sum bit can be said as the middle part of the result (S1). Then A1 and B1 is multiplied in vertical, and is added with the previous carry (C1) and get S2 as their product and carry2 (C2), the sum bit will be down to the result (S2). Then the carry2 (C2) is considered as the most significant part of the result (S3).

4.2. 4x4 bit Vedic Multiplier

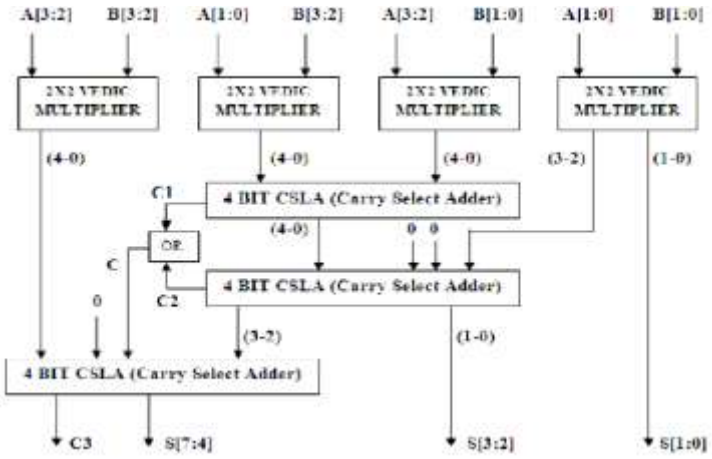

Fig 6: Block diagram of 4x4 Vedic multiplier

The design of 4X4 bit Vedic Multiplier is shown in Fig 6. In this design, the 1st stage is of 2X2 bit block (fundamental block) as a Multiplication unit. The 2nd stage comprises of carry and partial products. Then the 3rd

stage comprises of 4 bit Modified SQRT CSLA and 8 bit result of multiplication. The initial step in the design of 4X4 bit VM is grouping the 2 bit of each 4 bit input. These pairs constitute vertical and crosswise product terms. A separate 2X2 bit VM block does the handling of each input bit. The schematic of a 4X4 bit block is designed using 2X2 bit blocks. The partial products represent the Urdhva vertical and cross product terms.

4.3. 8x8 bit Vedic Multiplier

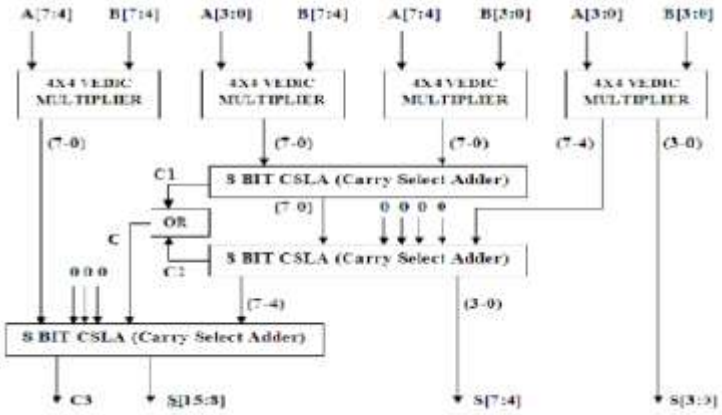

Fig 7: Block diagram of 8x8 bit Vedic Multiplier

The design of 8X8 bit VM is drawn in Fig 7. Here the first stage consists of 4X4 bit block as a Multiplication unit. The second stage is of carry and partial. Then the third stage comprises of 8 bit Modified SQRT CSLA and 16 bit result of multiplication. The primary and initial step in designing of 8X8 bit block VM is to group the 4 bit of each 8 bit input. This pair forms the in cross and vertical product terms.4X4 bit VM block handles each input bit pairs. The schematic depiction of 8X8 bit VM can be designed by 4X4 bit VM blocks.

4.4. 16x16 bit Vedic Multiplier

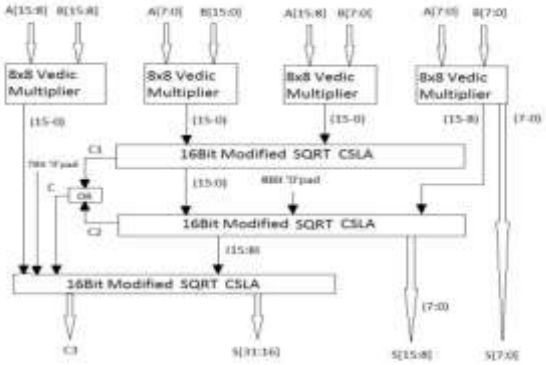

Fig 8: Block diagram of 16x16 bit Vedic Multiplier

The 16X16 bit VM's design drawn is in Fig 8. This design consists of first stage 8X8bit VM blocks, as a Multiplication unit. The second stage is of partial products and carry. Then the third stage is of 16 bit Modified SQRT CSLA and 32 bit result of multiplication. The initial step in designing of 16X16 bit VM is to group the 8 bit of each and every 16 bit input. The above said pairs forms in cross and vertical product terms. Each input bit pairs are handled by a separate 8X8 VM block. The schematic depiction of a16X16 bit VM block can be designed using 8X8 bit VM blocks.

4.5. 32x32 bit Vedic Multiplier

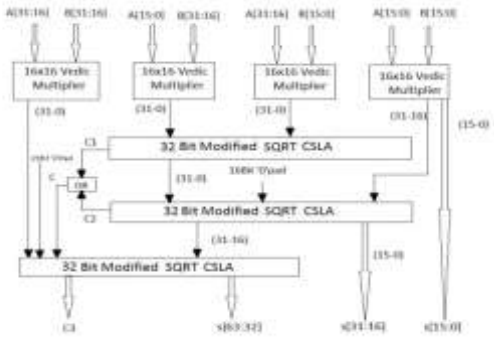

Fig 9: Block diagram of 32x32 bit Vedic Multiplier

Designing of 32X32 bit VM is as shown in Fig 9.Here the first stage is of 16X16 bit VM blocks as a Multiplication unit. The second stage comprises of carry and partial products. Then the third stage is of 32 bit Modified SQRT CSLA and 64 bit result of multiplication. The primary and initial step in the design of 32X32 bit VM is grouping the 16 bit of each 32 bit input. These pairs forms vertical, crosswise product terms. Separate 16X16 bit VM block handles each input bit pairs. The schematic depiction of 32X32 bit VM block can be designed using 16X16 bit VM blocks.

4.6. 64x64 bit Vedic Multiplier

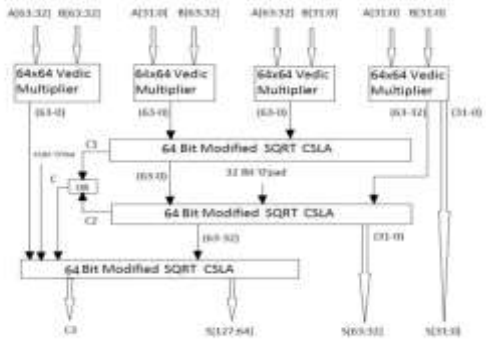

Fig 10: Block diagram of 64x64 bit Vedic Multiplier

Designing of 64X64 bit VM drawn in Fig 10.Here the first stage is of 32X32 bit VM blocks as a Multiplication unit. The second stage is of carry and partial products. Then the third stage comprises of 64 bit Modified SQRT CSLA and 128 bit results of multiplication. The primary and initial step in designing of 64X64 bit VM block is to group the 32 bit of each 64 bit input. These pairs forms vertical in cross and vertical product terms. Separate 32X32 bit VM block handles each input bit pairs. The schematic depiction of a 64X64 bit VM can be designed using 32X32 bit VM blocks.

4.7. 128x128 bit Vedic Multiplier

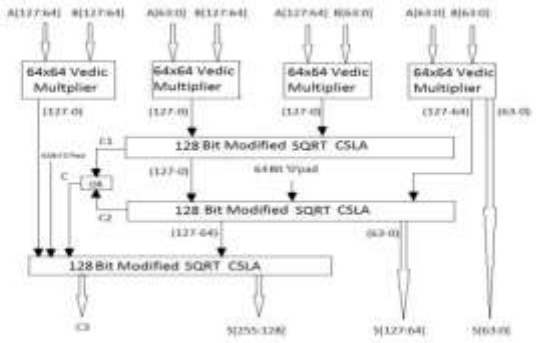

Fig 11: Block diagram of 128x128 bit Vedic Multiplier

Designing of 128X128 bit VM drawn in Fig 11.Here the 1st stage is of 64X64 bit VM blocks, as a Multiplication unit. The second stage is of carry and partial products. Then the third stage is of 128 bit Modified SQRT CSLA and 256 bit result of multiplication. The primary and initial step in designing of 128X128 bit VM is to group the 64 bit of each 128 bit input. These pair forms in cross and vertical product terms. Separate 64X64 bit VM block handles each and every input terms. The schematic depiction of 128X128 bit VM block can be designed using 64X64 bit VM blocks.

V DEVICE UTILIZATION SUMMARY

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of 4 input LUTs	55,225	66,560	82%
Number of occupied Slices	33,278	33,280	99%
Number of Slices containing only related logic	33,278	33,278	100%
Number of Slices containing unrelated logic	0	33,278	0%
Total Number of 4 input LUTs	55,225	66,560	82%
Number of bonded IOBs	513	633	81%

Fig 12: Device Utilization Summary of Vedic Multiplier

VI POWER REPORT AND TIME DELAY

On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	55225	66560	83
Signals	0.000	55359	---	---
IOs	0.000	513	633	81
Leakage	0.338			
Total	0.338			

	Total	Dynamic	Quiescent
Supply Power (W)	0.338	0.000	0.338

```
Maximum combinational path delay: 256.815ns
```

Fig 13: Power Report and Time delay of Vedic Multiplier

VII SIMULATION RESULTS

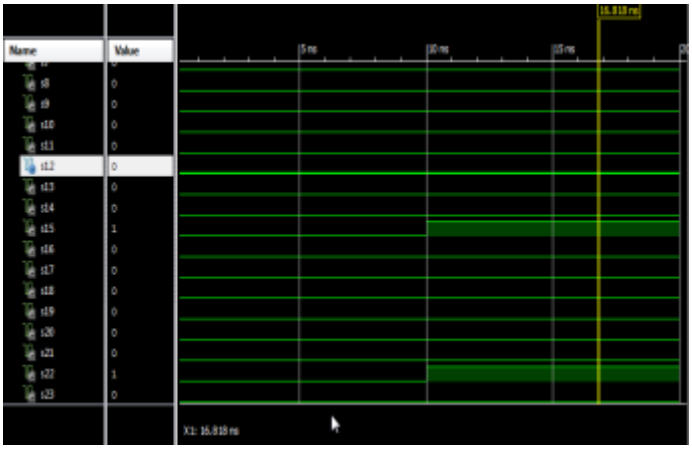

Fig 14: Simulation Results of Vedic Multiplier

The Vedic Multiplier using proposed CSLA is designed on Verilog. Xilinx project navigator 14.5 is used for synthesis. Simulation results are shown in Fig 14.

VIII CONCLUSION

In this paper we have designed the Vedic multiplier using the proposed CSLA which results in the reduction of the area and the time delay. The proposed design speeds up the addition and offer the great advantage in reducing the delay. Thus the area and the time delay is reduced in the proposed Vedic multiplier than the other existing methods.

REFERENCES

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

[2] 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*

[3] Koren Israel, "*Computer Arithmetic Algorithms," 2nd Ed, pp. 141-149*,*Universities Press*, 2001.

[4] L. Sriraman, T.N. Prabakar, *"Design and Implementation of Two Variable Multiplier Using KCM and Vedic Mathematics,"* 1st Int. Conf. on Recent Advances in Information Technology, Dhanbad, India, 2012, *IEEE Proc., pp. 782-787.*

[5] M. Ramalatha, K. Deena Dayalan, S. Deborah Priya, *"High Speed Energy Efficient ALU Design using Vedic Multiplication Techniques,"* Advances in Computational Tools for EngineeringApplications, 2009, *IEEE Proc., pp 600-603*.

AUTHORS PROFILE

	**Srikanth.S**received his B.E degree in electronics and communication engineering from S.N.S College of technology, Coimbatore and also received M.E degree from Sri Ramakrishna Engineering College; Coimbatore.He already published one journal related to VLSI Design. His area of interest is VLSI signal processing and Computer Architecture.
	**SanthoshKumar.A** is working as an assistant professor in S.N.S College of technology, Coimbatore and also his area of interests are VLSI Design and Signal Processing and Computer Networks.
	**LOKESHWARAN.R** pursuing B.E. 3rd year Electronics and Communication Engineering at SNS College of Technology, Coimbatore. His area of interest are circuit designing, automation and VLSI Design.
	**ANANDHAN.A** pursuing B.E. 3rd year Electronics and Communication Engineering at SNS College of Technology, Coimbatore. His area of interest are PCB Design, automation, VLSI Design and Signal Processing.