Design of High Speed Vedic Multiplier Using Carry Select Adder

[1] C. Durgadevi , [2] M. Renugadevi , [3] C. Sathyasree, [4] R.Chitra
[1,2,3] Students / ECE , Ramco Institute of Technology, Rajapalayam.
[4]Assistant Professor /ECE, Ramco Institute of Technology, Rajapalayam

***Abstract*- The Multiplier is the core element in digital signal processors. High-Speed processors are required for many applications. So, there is a great need for high-speed multiplier circuits. Speed and Area are one of the most important parameters to judge the performance of a multiplier. The goal of this paper is to design Vedic multiplier based on the Urdhva-Tiryakbhyam algorithm to increase the speed of multiplication by using carry select adder (CSLA).CSLA is one of the fast adders which can be used in data path applications to reduce the overall delay involved in addition. However conventional carry select adder is not an area efficient one due to the dual ripple carry adder structure. A new approach of CSLA with D Latch is proposed in the Vedic multiplier to minimize area and delay using Xilinx ISE 14.7 Tool. The synthesis result shows that it has 12% less delay than Vedic multiplier using CSLA.**

Keywords-Carry Select Adder, Ripple carry adder, D Latch, Vedic Multiplier.

I. INTRODUCTION

There is booming in use of portable devices in day to day life and these devices are ultimately demanding for high performance, so the use of Very Large Scale Integration - (VLSI) systems have been increased. To manage these demands, various researchers are developing to improve VLSI systems in terms of area, power, and delay [3]. The reduction in time delay is the prime objective for the high-speed processors. So, there is a great need for high-speed multiplier circuits. The array and Booth multiplication algorithms are the two commonly used multiplication algorithms in the digital hardware. In array multiplier, the partial products are calculated as separately in parallel and thus the time taken for calculation is less. Booth multiplication is another significant multiplication algorithm. High-speed multiplication needs a large booth arrays and exponential operations which sequentially need large partial sum and partial carry registers. Thus a large propagation delay is allied with this case. The Vedic mathematics approach is totally different one which is very close to the way a human mind works. A large amount of work has so far been done in understanding various methodologies (sutras). Vedic mathematics is an ancient mathematical technique[1].

A.Vedic Mathematics

Vedic mathematics is an ancient mathematical technique. Vedic is a word obtained from the word “Veda” and its meaning is “storehouse of all knowledge”. This Vedic mathematics is reconstructed from Vedas by Sri Bharti Krishna Tirathaji between the years 1911 to 1918. The Vedic mathematics has been divided into sixteen different Sutras where each sutra represents the different branch of mathematics[1].These Sutras along with their brief meanings are enlisted below alphabetically:

1. (Anurupye) Shunyamanyat - If one is in ratio, the other is zero.
2. Chalana-Kalanabyham - Differences and Similarities.
3. EkadhikinaPurvena - By one more than the previous one.
4. EkanyunenaPurvena - By one less than the previous one.
5. Gunakasamuchyah - The factors of the sum is equal to the sum of the factors.
6. Gunitasamuchyah -The product of the sum is equal to the sum of the product.
7. NikhilamNavatashcaramamDashatah - All from 9 and last from 10.
8. ParaavartyaYojayet - Transpose and adjust.
9. Puranapuranabyham - By the completion or non completion.
10. Sankalana- vyavakalanabhyam - By addition and by subtraction.
11. ShesanyankenaCharamena - The remainders by the last digit.
12. ShunyamSaamyasamuccaye - When the sum is the same that sum is zero.
13. Sopaantyadvayamantyam - The ultimate and twice the penultimate.
14. Urdhva-Tiryagbhyam - Vertically and crosswise.
15. Vyashtisamanstih - Part and Whole.
16. Yaavadunam - Whatever the extent of its deficiency.

These sutras can be applied to any branch of mathematics like algebra, trigonometry, geometry etc. These methods reduce the complex calculations into simpler ones. Because these calculations are performed by the human mind. These mathematical techniques consume lesser power and acquire lower chip area. In this paper, 16X16 Vedic multiplier is implemented using “Urdhva-Tiryakbhyam algorithm”.

II. ANCIENT VEDIC MATHEMATICAL ALGORITHM

By applying sutras, Vedic mathematics resolves the complexity of calculations. It requires less computation time and less hardware for implementation. These sutras are basically used for decimal multiplication here it is incorporated to binary multiplication.

A. Urdhva- Tiryakbhyamsutra (Vertically and Crosswise)

In this paper, implementation of Vedic multiplication technique namely “Urdhva-Tiryakbhyam – Vertically and crosswise” is demonstrated. This technique is more popular for its high speed working as it generates partial products in a parallel manner and then adding partial products simultaneously. Vedic multiplier conciliates this need without increasing power consumption. It has less complexity compared to booth multiplier Vedic multiplier requires less hardware[3].Thus Vedic multiplier gives numerous advantages in terms of area, power, delay and complexity.

B. Example for Vedic Multiplication

Two decimal numbers 234 and 356 are considered. Multiplication of these two numbers (234 x 356) is described with the line diagram for a clear understanding as shown in fig. 1.At first, two numbers shown with the line are multiplied, 2 digits output is generated. One’s place of this generated result is stored as one’s place of the final product and ten’s place of the generated output is hooked up as pre-carry for the next step.

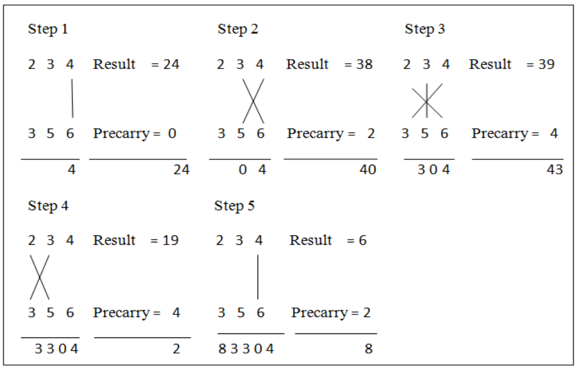


Figure1. Multiplication of two decimal numbers

In this way, the process perpetuated. At which point, there is more than one digit to multiply then multiply those digits shown with lines and accumulate all those generated products. The output of this summation is again stored in final result with forwarding pre-carry to next steps as explained earlier. In this way, the process continues to get the final result of the multiplication.

III. ARCHITECTURE OF VEDIC MULTIPLIER

The Vedic multiplication technique can be used for multiplication of both decimal and binary numbers. In this section, Vedic multiplication technique for binary numbers and their implementation of 2 X 2, 4 X 4, 8 X 8 and 16X16 Vedic multiplier architecture are explained. Implementation of 2 X 2 Vedic multiplier block is prime important in the implementation of 4 X 4,8 X 8 and16X16 Vedic multiplier architecture.

A. 2X2 Vedic Multiplier

Considering two numbers with two bits each and the numbers are A and B where A=a0a1 and B=b0b1 as shown in the below line diagram. Firstly, the Least Significant Bits are multiplied which gives the Least Significant Bit (LSB) of the final product (vertical).The Second step is to take the products in a crosswise manner such as the Least Significant Bit (LSB) of the multiplicand A is multiplied with the next higher bit of the multiplicand B in a crosswise manner. The sum gives a second bit of the final product and the carry is added to the partial product which is obtained by multiplying the Most Significant Bits and it generates the sum and carry. The sum and the carry are the 3rd and 4th bits of the final product [3].

s0 = a0b0 ----- (1)
c1s1 = a1b0 + a0b1 ----- (2)
c2s2 = c1 + a1b1 ----- (3)

The final result is given as c2s2s1s0. A 2×2 Vedic multiplier block is implemented by using two half adders and four two input and gates as shown in the below Figure2.

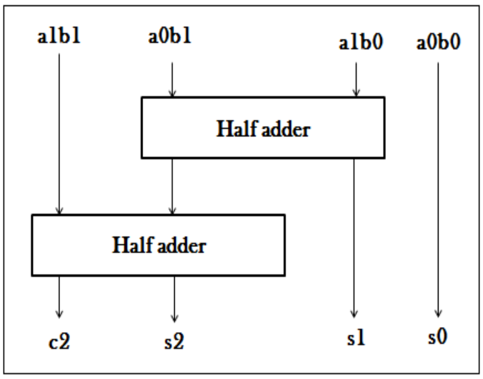


Figure2.Block diagram of 2X2 Vedic multiplier

B. 4X4 Vedic Multiplier

In this section, the 4x4 bit Vedic multiplier is explained. For explaining this multiplier, let us consider two four bit numbers are A and B where A=A3A2A1A0 and B=B3B2B1B0. The final output can be generated as the C3S6S5S4S3S2S1S0. The partial products are calculated in parallel and hence delay obtained is decreased vastly for the increase in the number of bits. Here 2X2 Vedic multipliers are used to implement 4X4Vedic multiplier to generate a partial product. Three Ripple carry adders of 4 bits each is used for addition of generated partial products. The carry output of first two Ripple Carry Adders are performed by OR operation and its output is given to next ripple carry adder. Zero inputs are given to some of the Ripple carry adders wherever required. For clear understanding, observe the block diagrams for 4x4 as shown below Figure 3. The carry generated from the first ripple carry adder is passed on to the next ripple carry adder and there are two zero inputs for the second ripple carry adder. The arrangement of the Ripple Carry Adders is shown in below block diagram which can reduce the computational time such that the delay can be decreased [3].

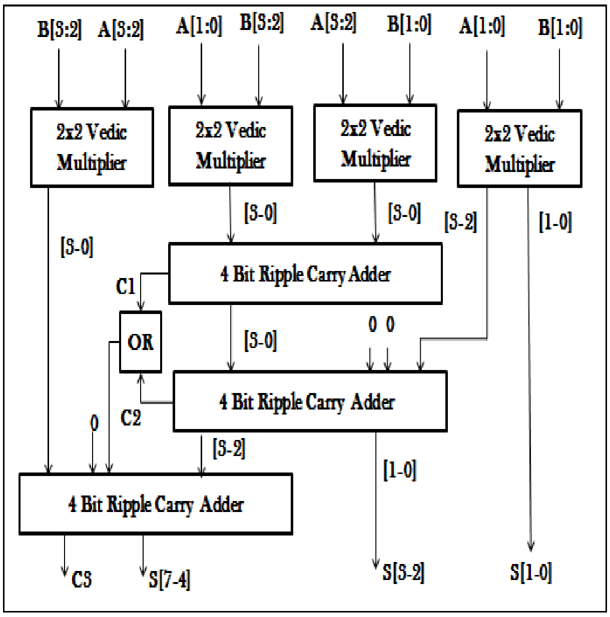


Figure3. Block diagram of 4X4 Vedic multiplier

C. 8X8 Vedic Multiplier

To explain this method, let us consider two 8 bit numbers are A and B, where A=A7A6A5A4A3A2A1A0&B=B7B6B5B4B3B2B1B0. The final output can be obtained as the C3S15S14S13S12S11S10S9S8S7S6S5S4S3S2 S1S0. Implementation of the 8x8 Vedic multiplier is clearly understood from the below block diagram as shown in Figure 4.

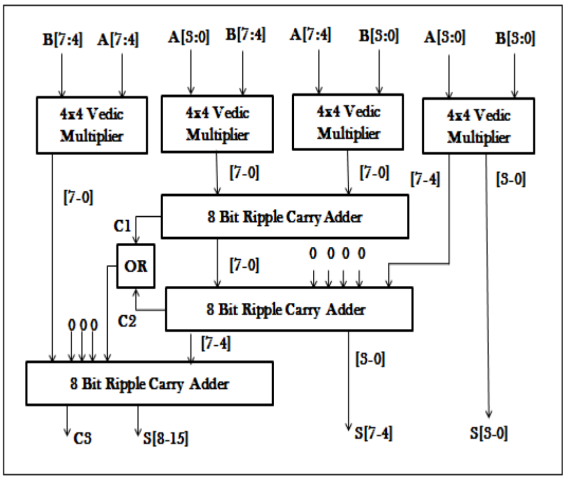


Figure4. Block Diagram of 8X8 bit Vedic Multiplier

Here, four 4X4 Vedic multiplier blocks and three carry select adders of 8 bits each are used. The arrangement of the carry select adders is made in a different way such that it requires less computation time. Some of the carry select adders are given with zero inputs, wherever required. The output of middle multipliers is added using first CSLA. The Output of first CSLA and first Vedic multiplier are added using second CSLA. Carry outputs from first two CSLAs are performed by OR operation and it is given as an input to the third CSLA to generate the final result[3].

D.16X16 Vedic Multiplier

The design of 16×16 block is a similar arrangement of 8×8 blocks in an optimized manner which is shown in Figure 5.The first step in the design of 16×16 block will be grouping the 8 bit (byte) of each 16-bit input.The LSB of two inputs will form vertical and crosswise product terms. Each input byte is handled by a separate 8×8 Vedic multiplier to produce sixteen partial product rows. These partial products rows are added in a 16-bit carry select adder optimally to generate final product bits. The schematic of a 16×16 block is designed by using the 8X8 Vedic multiplier. The partial products represent the Urdhva vertical and cross product terms. Then by using or gate, the final product is obtained.

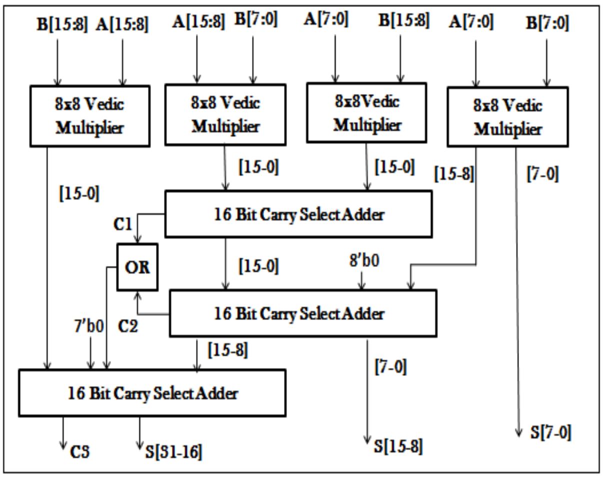


Figure5. Block Diagram of 16X16 bit Vedic Multiplier

IV. CONVENTIONAL CARRY SELECT ADDER

In this conventional method, the RCA and D- latch operation is performed parallelly. For n bit, RCA structure it required n D-latches with enable pin as a CLK. The RCA structure c_{in} is replaced by enable pin , where enable signal is CLK signal. When enable pin en =1 then the RCA structure is calculate for c_{in}=1 that result is stored in D-latch. When en =0 then it will calculate for c_{in} =0 and the D-latch output and full adder output is given to the mux. By using selection line it will give the proper output. Where the enable time period for '1'is very less when compared to the enable pin '0'. Initially, RCA structure will calculate for en=1 and then en =0. The architecture of proposed 16-bit CSLA is shown in Figure 6. It has different five group so different bit size RCA and D-Latch. Instead of using two separate adders in the regular CSLA, in this method only one adder is used to reduce the area, power consumption and delay. Each of the two additions is performed in one clock cycle. This is 16-bit adder in which least significant bit (LSB) adder is ripple carry adder, which is 2 bit wide. The upper half of the adder i.e., most significant part is 14-bit wide which works according to the clock[9].

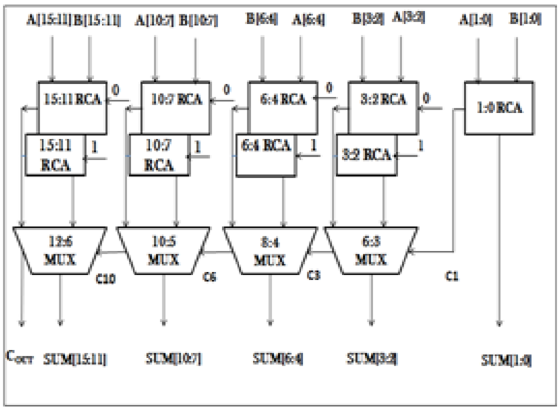


Figure6. Block diagram of conventional carry select adder

Whenever clock goes high addition for carry input one is performed. When the clock goes low then carry input is assumed as zero and sum is stored in adder itself. Carry out from the previous stage i.e., least significant bit adder is used as control signal for the multiplexer to select final output carry and the sum of the 16-bit adder. If the actual carry input is one, then computed sum and carry latch is accessed and for the carry input zero MSB adder is accessed. C_{out} is the output carry.

V. PROPOSED DESIGN

In this proposed design the operation of the Vedic multiplier is similar which explain in previous 16x16 Vedic multiplier. But the function of carry select adder using D-latch is different from normal carry select adder. This method replaces the one RCA and add one circuit by D latch with enable signal. The block diagram of the proposed model is shown in below figure.

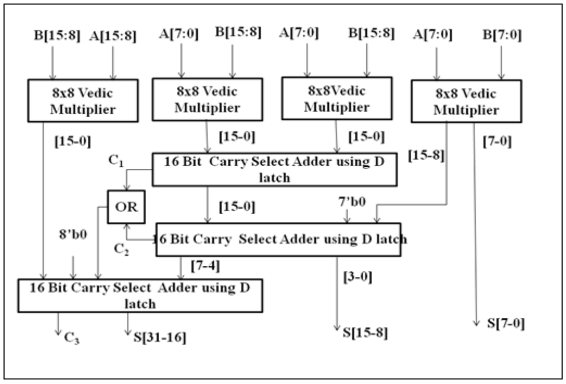


Figure7.Block diagram of Proposed 16x16 Vedic multiplier

A. D-Latch Terminology

Latches are used to store one-bit information. Their outputs are constantly affected by their inputs as long as the enable signal is asserted. In other words, when they are enabled, their content changes immediately according to their inputs[7]. D-latch and its structure is shown in below figure 8.

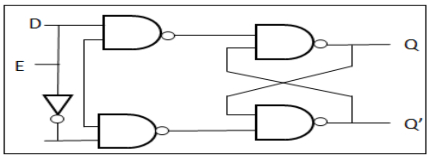


Figure8. Flipflop structure of D Latch

B. Working Principle of Carry Select Adder Using D Latch

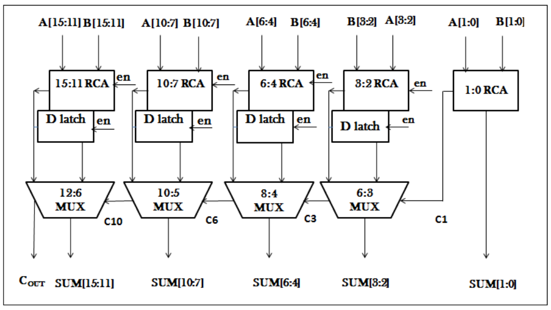


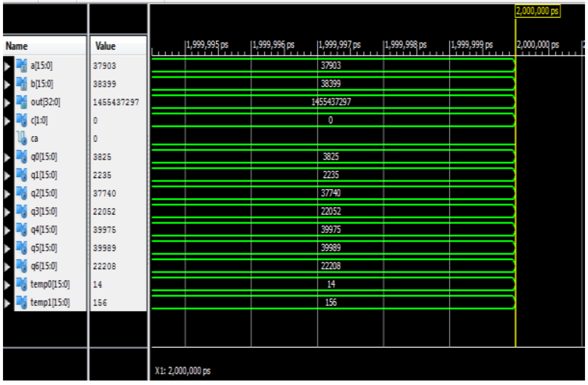
Figure9. Block diagram of carry select adder using D latch

The architecture of proposed 16-bit CSLA is shown in Figure9. It has different five groups of different bit size RCA and D Latch. Instead of using two separate adders in the regular CSLA, in this method, only one adder is used. Each of the two additions is performed in one clock cycle. This is 16-bit adder in which least significant bit (LSB) adder is ripple carry adder, which is 2 bit wide. The upper half of the adder i.e., most significant part is 14-bit wide which works according to the clock. Whenever clock goes high addition for carry input one is performed. When the clock goes low then carry input is assumed as zero and sum are stored in adder itself. From the Figure.9, it can understand that latch is used to store the sum and carry for en=1 and en=0.Carry out from the previous stage i.e., least significant bit adder is used as control signal for the multiplexer to select final output carry and the sum of the 16-bit adder. The group 2 performed the two-bit addition which isa2 with b2 and a3 with b3. The group 2 structure has five D-Latches in which four are used for store the sum2 and sum3 from FA2 and FA3 respectively and the last one is used to store carry.The Multiplexer is used for selecting the actual sum and carry according to the carry is coming from the previous stage. The 6:3 multiplexer is the combination of 2:1 multiplexer. When the clock is low a2 and b2 are added with carry is equal to zero. Because of low clock, the first D-Latch is not enabled[8]

.

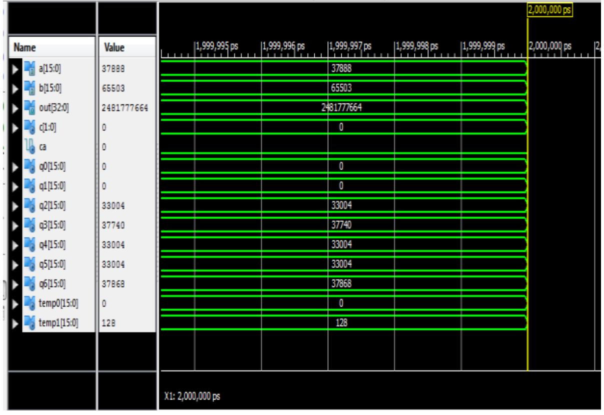
VI. SIMULATION RESULTS

1. 16x16 Vedic Multiplier using Ripple Carry Adder



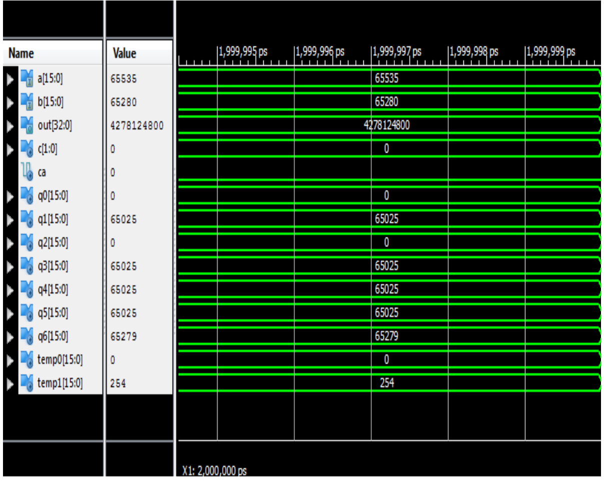
Name	Value
a[15:0]	37903
b[15:0]	38399
out[32:0]	1455437297
c[1:0]	0
ca	0
q0[15:0]	3825
q1[15:0]	2235
q2[15:0]	37740
q3[15:0]	22052
q4[15:0]	39975
q5[15:0]	39989
q6[15:0]	22208
temp0[15:0]	14
temp1[15:0]	156

2. 16x16 Vedic Multiplier using Carry Select Adder



Name	Value
a[15:0]	37888
b[15:0]	65503
out[32:0]	2481777664
c[1:0]	0
ca	0
q0[15:0]	0
q1[15:0]	0
q2[15:0]	33004
q3[15:0]	37740
q4[15:0]	33004
q5[15:0]	33004
q6[15:0]	37868
temp0[15:0]	0
temp1[15:0]	128

3. Proposed Vedic Multiplier



Name	Value
a[15:0]	65535
b[15:0]	65280
out[32:0]	4278124800
c[1:0]	0
ca	0
q0[15:0]	0
q1[15:0]	65025
q2[15:0]	0
q3[15:0]	65025
q4[15:0]	65025
q5[15:0]	65025
q6[15:0]	65279
temp0[15:0]	0
temp1[15:0]	254

VII. COMPARISION TABLE

Design	Width(n)	Area (No of LUTs)	Delay (ns)
Vedic multiplier using RCA	16 bits	539	14.351
Vedic multiplier using CSLA	16 bits	553	14.013
Proposed design	16 bits	546	12.343

VIII. CONCLUSION

In this Paper the Proposed High speed carry select adder using D-latch based Vedic multiplier architecture is designed to reduce the delay of CSLA architecture than the RCA based CSLA architecture. The functionality verification of the design is carried out by using ISE Simulator and the synthesis is also carried out by the XILINX ISE 14.7. From the table, it is concluded that, the proposed D-Latch based Vedic multiplier design is having less delay when compare to the RCA based architecture.

REFERENCES

[1] Charishma, V. and Ganeshkumar, G. (2012) “Design of High Speed Vedic Multiplier using Vedic Mathematics Techniques’, International Journal of Scientific and Research Publications Vol.2, Issue 3, pp.1-5.

[2] Basant Kumar Mohanty and Sujit Kumar Patel (2014) ‘Area–Delay–Power Efficient Carry-Select Adder’, IEEE Transactions on Circuits And Systems-II Express Briefs, Vol. 61, No. 6, pp.418-422.

[3] Gokhale, G.R. and Gokhale, S.R. (2015) ‘Design of Area and Delay Efficient Vedic Multiplier Using Carry Select Adder’, IEEE International Conference on Information Processing, pp.295-300.

[4] JyogiLeelavathi and Raghu Ramprasad, Y. (2016) ‘Design and Implementation of Delay Efficient Carry Select Adder Using D-Latch’, International Journal of Advanced Scientific Technologies in Engineering and Management Sciences, Vol.2,Issue.7,pp.1-5.

[5] Ramkumar, B and Kittur, H.M (2012) "Lower-power and Area-Efficient Carry-Select Adder",IEEE trans. Very large scale integr.(VLSI) syst., vol. 20, no.2, pp.371-375.

[6] Manju,S and Sornagopal,V(2013) “An efficient SQRT architecture of carry select adder design by common Boolean logic”, in Proc. VLSI ICEVENT, pp.1–5.

[7] Manikandan S.A eta.,(2015)”Power efficient Carry select adder using D latch” IRJET , vol .01, issue .03, pp.2055-2058.

[8] Gudise Anil Babu and Pushpalatha C.H,(2015)” Implementation of High Speed Adder using Dlatch”vol.3,issue.12,pp.162- 172.

[9] Prashanti ,P and Rajendra Naik,B(2013) ”Design and Implementation of High speed carry select adder”IJETT,l.4,issue.9, pp.3985-3990