DESIGN AND IMPLEMENTATION OF HIGH SPEED AND AREA EFFICIENT MAC UNIT

M.Nivedha
B.E (ECE), Associate Professor (ECE) SSN College Of Engineering, Chennai,
Tamil Nadu- 603110. Email: nivedha13065@ece.ssn.edu.in

V.Priyadharshini
B.E (ECE), Associate Professor (ECE) SSN College Of Engineering, Chennai,
Tamil Nadu- 603110, Email: priya24ssn@gmail.com

K.Rathi Meena,
B.E (ECE), Associate Professor (ECE) SSN College Of Engineering, Chennai,
Tamil Nadu- 603110, Email: rathi13072@ece.ssn.edu.in

Mr.C.Thiruvenkatesan
B.E (ECE), Associate Professor (ECE) SSN College Of Engineering, Chennai,
Tamil Nadu- 603110

Abstract — The Multiply-Accumulate unit is the main computational kernel in Digital Signal Processing application. To determine the speed of the entire hardware systems, the Multiply and Accumulate Unit (MAC) always play an important role. The efficient MAC Unit is used to support the variable precisions and parallel functions with high desirability. In this work, 64 Bit MAC Design using area efficient Vedic Multiplier and Square Root Carry-Select Adder (SQRT CSLA) for DSP Processors is implemented. To design a N*N Vedic MAC Design, four N/2*N/2 Vedic Multiplier and Square Root Carry Select Adder are required for an efficient design. Various adders such as Ripple Carry Adder, Carry Save Adder, Square root Carry Select Adder and multipliers such as Booth Multiplier, Wallace Tree Multiplier and Vedic Multiplier are analyzed. Conventional MAC design is implemented using Vedic Multiplier with Ripple Carry Adder (RCA). To reduce the Look-Up Tables (LUTs), Delay and Power, the Vedic Multiplier with Square Root Carry Select Adder (SQRT CSLA) is proposed in this work. The Conventional and Proposed MAC design are coded in Verilog HDL Language, synthesized using Xilinx ISE and simulated using Modelsim XE. Number of LUT Counts, Delay and Power of the conventional MAC and Proposed MAC are compared.

Keywords—Mutiply-Accumulator (MAC), Digital Signal Processing (DSP), Vedic Multiplier, Square Root Carry Select Adder (SQRT CSLA), Carry Select Adder (CSLA), Booth Multiplier, Wallace Tree Multiplier.

1. Introduction

Multiplication is an important fundamental function in arithmetic operations. Multiplication based operations such as Multiply and Accumulate (MAC) and inner product are among some of the frequently used Computation-Intensive Arithmetic Functions(CIAF) currently implemented in many Digital Signal Processing (DSP)applications such as convolution, Fast Fourier Transform (FFT), filtering and in microprocessors in its arithmetic and logic unit. Since multiplication dominates the execution time of most DSP algorithms, there is a need of high speed multiplier. Currently, multiplication time is still the dominant factor in determining the instruction cycle time of a DSP chip. The demand for high speed processing has been increasing as a result of expanding computer and signal processing applications. Higher throughput arithmetic operations are important to achieve the desired performance in many real-time signal and image processing applications. One of the key arithmetic operations in such applications is multiplication and the development of fast multiplier circuit has been a subject of interest over decades. Reducing the time delay and power consumption are essential requirements for many applications.

To design and implement high speed and area efficient MAC unit for digital signal processing applications, the Simulation results of various adders, multipliers units are analyzed and their corresponding power, delay and number of LUTs are measured and compared. From the Comparison of adders, an efficient adder is selected, similarly, an efficient multiplier is selected from the existing multipliers. The MAC UNIT is designed from the selected adder and multiplier and the performance of the proposed MAC UNIT is analyzed and compared with the existing MAC unit.

2. Proposed Adders

2.1 Ripple Carry Adders

It is possible to create a logical circuit using multiple full adders to add N-bit numbers. Each full adder inputs a Cin, which is the Cout of the previous adder. This kind of adder is called a ripple-carry adder, since each carry bit "ripples" to the next full adder. Note that the first (and only the first) full adder may be replaced by a half adder (under the assumption that Cin = 0). The layout of a ripple-carry adder is simple, which allows for fast design time. However, the ripple-carry adder is relatively slow, since each full adder must wait for the carry bit to be calculated from the previous full adder. The gate delay can easily be calculated by inspection of the full adder circuit. Each full adder requires three levels of logic. In a 32-bit ripple-carry adder, there are 32 full adders, so the critical path (worst case) delay is 2 (from input to carry in first adder) + 31 * 3 (for carry propagation in later adders) = 95 gate delays.

2.2 Carry Save Adders

If an adding circuit is to compute the sum of three or more numbers it can be advantageous to not propagate the carry result. Instead, three input adders are used, generating two results: a sum and a carry. The sum and the carry may be fed into two inputs of the subsequent 3-number adder without having to wait for propagation of a carry signal.

After all stages of addition, however, a conventional adder (such as the ripple carry or the look ahead) must be used to combine the final sum and carry results.

2.3 Design of Conventional Square Root Carry Select Adder

A full adder is formed by chaining the no of equal length adder stages as in carry bypass approach. In a ripple carry adder every full adder has to wait for the incoming carry to be generated. One way to get this linear dependency is to anticipate the possible values of both the carry to be generated before in advance. Once the real value is known it is selected by simple multiplexer stage. An implementation of this idea is carry select adders which is used in SQRT CSLA. Square root carry select adder is constructed by equalizing the delay through the two carry chains and the multiplexer signal from previous stage. This is an extension of linear carry select adder which improves the delay time greatly. In SQRT CSLA adder the time can be improved, as the time waiting for the carry bit is used to calculate an extra input bit in each stage. Even though there is increase in timing to an extent it has lot of disadvantages.

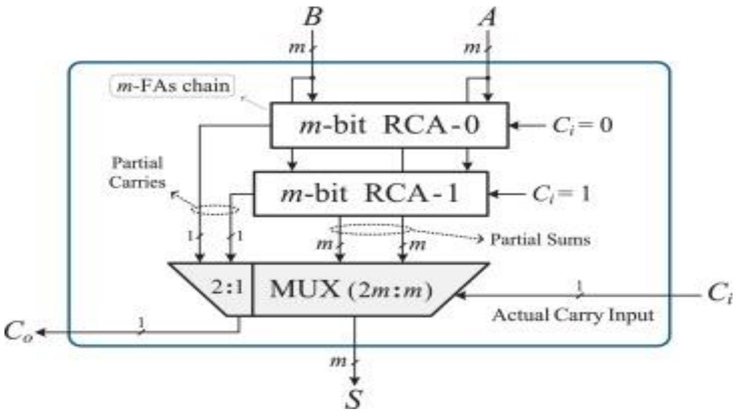


Fig 2.3.1 Conventional Structure of the CSLA

Disadvantages of the conventional SQRT CSLA: Duplication of adders for each set of input bit is a main disadvantage in this kind of adders. Because of this duplication the size of the adder is bigger and takes up more space than standard ripple adder. Since it doubles the amount of calculations done for each bit means that the power consumed is near to twice the amount of ripple carry adder. But the structure can be modified by changing the duplication of adders into a simple circuit. This can be done using binary to excess one converter.

3. Proposed Multipliers

3.1 Booth Multiplier

The Booth multiplier is also known as Recoded booth multiplier, in which the multiplicand is kept as it is and the multiplier is recoded as a recoded multiplier and then the multiplication is done with multiplicand and recoded multiplier. To reduce the number of partial products in the multiplier, the Multiplier uses Radix 2r multipliers, which produces N/r partial products, each of which depends on r bits of the multiplier. Fewer partial products lead to a smaller and faster CSA (Carry Save Adder) array. For example, a radix-4 multiplier produces N/2 partial products. Each radix-4 multiplier produces N/2 partial products. Each partial product is 0, Y, 2Y, or 3Y,depending on a pair of bits of X. Computing 2Y is a simple shift, but 3Y is a hard multiple requiring a slow carry-propagate addition of Y + 2Y before partial product generation begins. Higher-radix Booth encoding is possible, but generating the other hard multiples appears not to be worthwhile for multipliers of fewer than 32 bits.

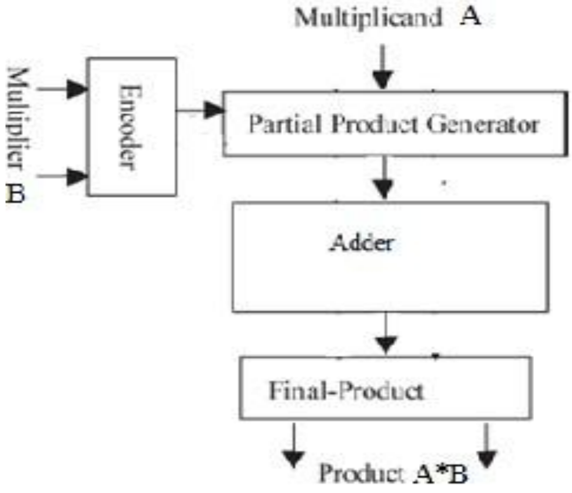


Fig.3.1.1 Booth Multiplier Architecture

One advantage of booth multiplier is, it reduce the number of partial products thus extensively used in long operands and also reduces the number of adders. The main disadvantage of booth multiplier is the complexity of the circuit to generate a partial product in the Booth encoding. And also the high performance of booth multiplier comes with the drawback of power consumption.

3.2 Wallace Multiplier

A fast process for multiplication of two numbers was developed by Wallace. Using this method, a three step process is used to multiply two numbers. Multiply each bit of one of the arguments, by each bit of the other, yielding results. Depending on position of the multiplied bits, the wires carry different weights. Reduce the number of partial products to two by layers of full and half adders. Group the wires in two numbers, and add them with a conventional adder. The bit products are formed, the bit product matrix is reduced to a two row matrix where sum of the row equals the sum of bit products, and the two resulting rows are summed with a fast adder to produce a final product. In the Wallace tree method, three bit signals are passed to a one bit full adder (“3W”) which is called a three input Wallace tree circuit, and the output signal (sum signal) is supplied to the next stage full adder of the same bit, and the carry output signal thereof is passed to the next stage full adder of the same no of bit, and the carry output signal thereof is supplied to the next stage of the full adder located at a one bit higher position. Wallace tree is reducing the number of operands at earliest opportunity. If you trace the bits in the tree, you will find that the Wallace tree is a tree of carry-save adders arranged as shown. A carry save adder consists of full adders like the more familiar ripple adders, but the carry output from each bit is brought out to form second result vector rather being than wired to the next most significant bit. The carry vector is 'saved' to be combined with the sum later, hence the carry-save moniker.In the Wallace tree method, the circuit layout is not easy although the speed of the operation is high since the circuit is quite irregular.

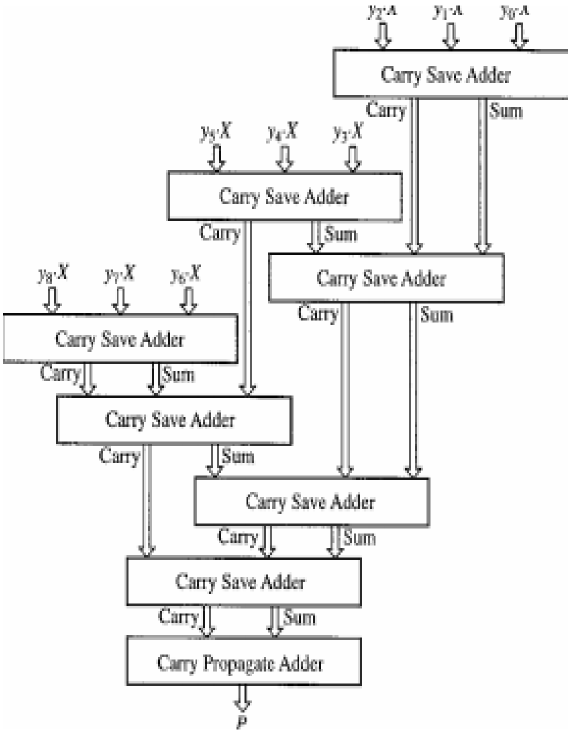


Fig.3.2.1 Wallace tree Multiplier

3.3 Vedic Multiplier

Vedic mathematics is the name given to the ancient Indian system of mathematics that was rediscovered in early twentieth century. Vedic mathematics is mainly based on sixteen principles or word-formulae which are termed as Sutras. A simple digital multiplier (referred henceforth as Vedic multiplier) architecture based on the Urdhva - Triyakbhyam (Vertically and Cross wise) Sutra is presented here in the existing Vedic multiplier. This Sutra was traditionally used in ancient India for the multiplication of two decimal numbers in relatively less time.

Urdhva – Tiryakbhyam is the common formula applicable to all cases of multiplication and also in the division of a huge number by another huge number. It means perpendicularly and diagonally. Existing 16-bit Vedic Multiplier consists of four 8X8 Vedic Multiplier units and three 16 bit Ripple carry adders. This existing 16-bit Vedic Multiplier consumes more area and delay due to usage of three Ripple carry adders. To perform multiplication for 16 bit, four 8X8 Vedic multipliers are necessary. Then 8X8 Vedic Multiplier depends on 4X4 Vedic Multiplier and 4X4 Vedic Multiplier depends on 2X2 Multiplier. The circuit diagram for 2X2, 4X4 and 8X8 Vedic Multipliers are explained in below. In each multiplier, there are three sets of Ripple Carry Adder to perform addition operations. It takes more time to perform the addition operation due to carry propagation.

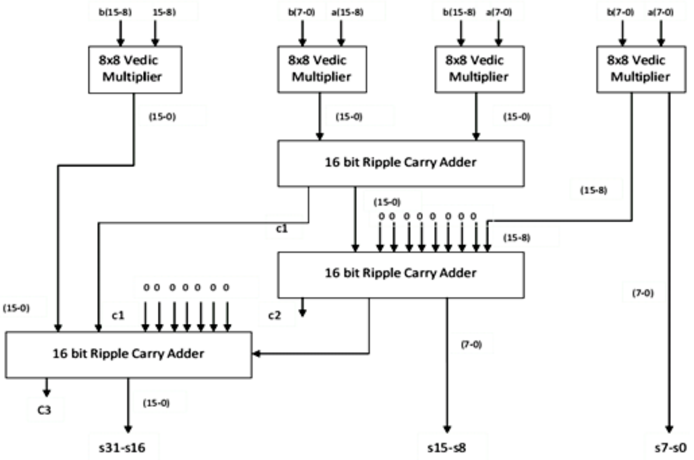


Fig 3.3.1 Block diagram of 16-bit Vedic Multiplier

8X8 Vedic Multiplier

In existing Vedic Multiplier shown in fig, four 8X8 Vedic multipliers are used to perform 16-bit multiplication. 8X8 Vedic Multipliers can be easily implemented by using four 4X4 Vedic multipliers and three 8 bit Ripple carry adders. In this multiplier, more area and delay caused due to the three 8 bit Ripple carry adders. This 8 bit multiplication is performed in a parallel manner.

4X4 Vedic Multiplier

The design of 8X8 Vedic Multiplier is done by using four 4X4 Vedic Multiplier. Further, 4X4 Vedic Multiplier is designed using four 2X2 Vedic Multipliers and three 4 bit Ripple carry adders are shown. Because of three Ripple Carry Adders in 4X4 Vedic multiplier, this architecture consumes more area and delay.

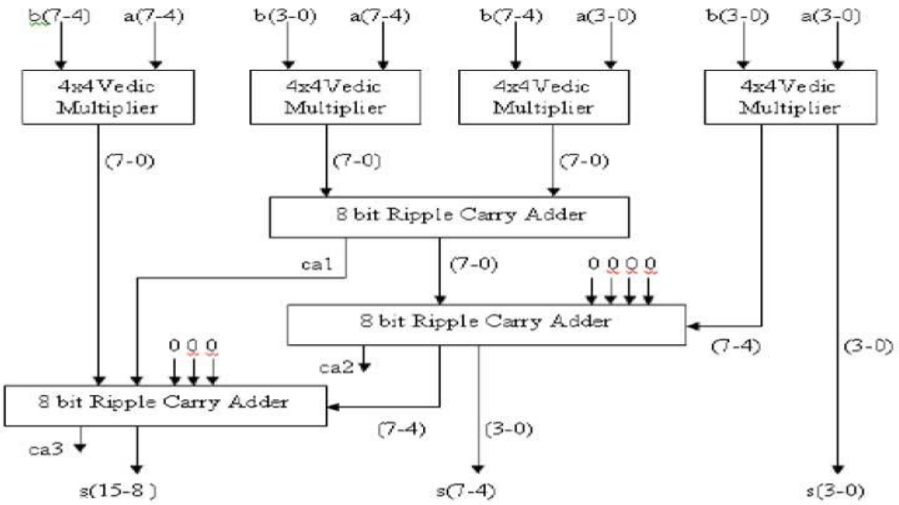
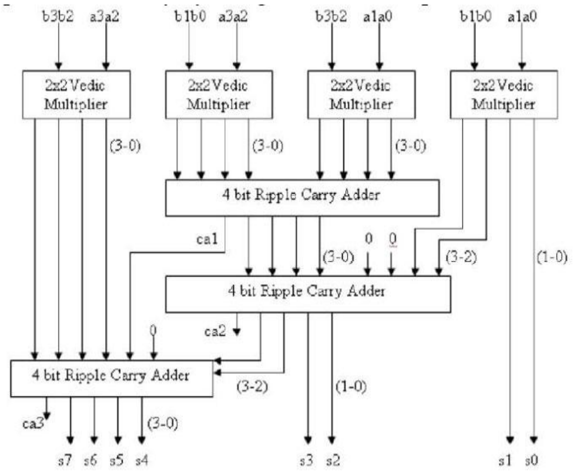


Fig.3.3.2 Block diagram of 8X8 Vedic Multiplier



.

Fig.3.3.3 Block diagram of 4X4 Vedic Multiplier

2X2 Vedic Multiplier

All 2X2, 4X4 and 8X8 multipliers depend on three Ripple carry adders to generate multiplication results. Even using parallel operation for multiplication, the addition operation could take more time and consume large area to provide multiplication result. So, the Vedic multiplier requires large area and delay to perform multiplication operation.

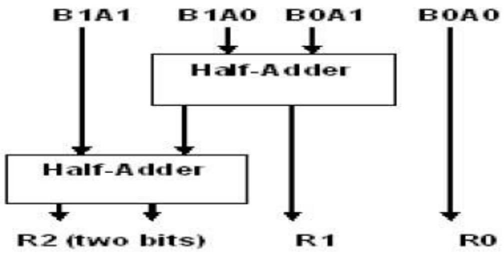


Fig.3.3.4 Block diagram of 2X2 Vedic Multiplier

Vedic multiplier is highly suitable for high speed complex arithmetic circuits which are having wide application in VLSI and signal processing. Vedic multiplier is superior area and speed wise compared to other multipliers. It enables parallel generation of partial products and eliminates unwanted multiplication steps. Multiplier architecture is based on generating all partial products and their sums in one step.

4.Proposed Sqrt Carry Select Adder

The existing system BEC- based CSLA involves less logical resources than conventional CSLA but it has marginally higher delay as the BEC unit wait until the RCA unit calculate the n-bit sum and carry corresponding Cin =0. The BEC method therefore increases the data dependence in the CSLA.

The proposed structure consists of SQRT CLSA is shown in figure 4.1

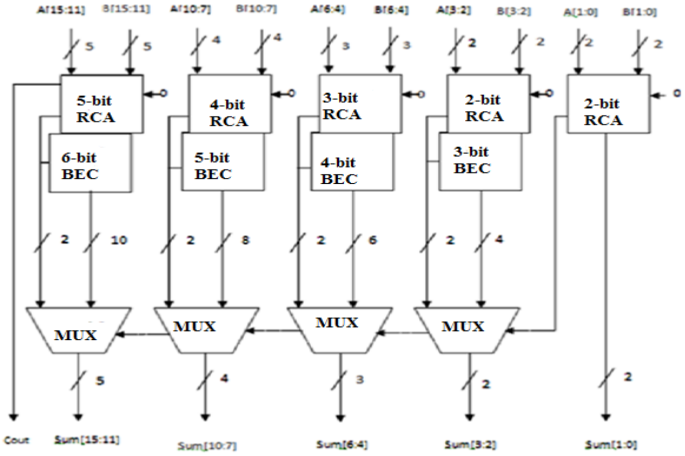


Fig.4.1 SQRT CSLA

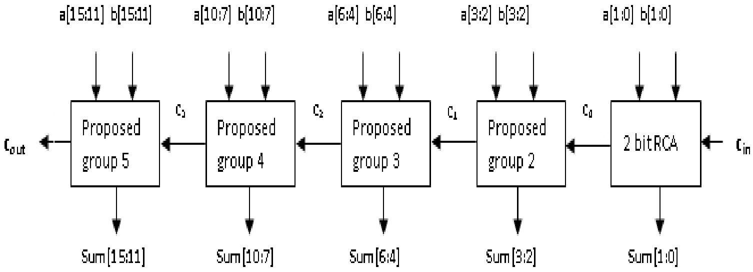


Fig.4.2 Proposed SQRT Carry Select Adder

Every group structures have RCA, BEC and Multiplexer circuits, hence most essential components to design group structures of SQRT CSLA are Full Adders (FAs), Half Adders (HAs), Logic Gates (AND, EX-OR and NOT) and Multiplexers. For instance, group-2 and group-3 structures of 16-bit SQRT CSLA circuits are illustrated. In this structures, combination of FA and HA gives the results of RCA and it is followed by BEC circuit which indicated in dotted line of fig 4.1s. Finally, Multiplexors are used to provide final sum outputs. Carry input (Cin) is given to the selection input of first group of Multiplexers. Remaining groups get the Carry inputs from previous groups. Hence, final stage of SQRT CSLA only cause little CPD than traditional RCA circuit is shown in the fig.4.2 .

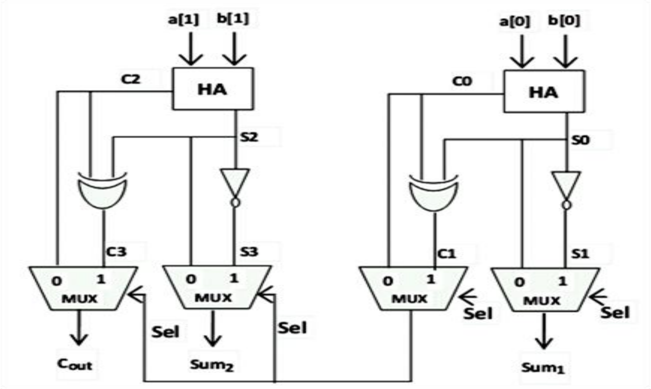


Fig 4.3 Proposed Group2 structure for proposed SQRT CSLA

In this, the complexity of BEC circuits and multiplexer circuits are realized and re-constructed to increase the performance in terms silicon area and power consumption. Redundant logic function of each group structures are identified and eliminated to reduce the hardware complexity. Hence, the developed adder circuit is named as “Reduced Complexity SQRT CSLA. The circuit diagram of reduced complexity SQRT CSLA for 4-bit addition Similarly, we can extend and compress the circuit of for group-5 structure and group-2, group- 3 structures. When compared to the traditional group structures of BEC based SQRT CSLA, developed group structures of reduced complexity SQRT CSLA reduces the gate count value significantly. Theoretically, 38% of gate counts are reduced in reduced complexity SQRT CSLA than traditional SQRT CSLA adder circuits. Further, the performance of reduced complexity SQRT CSLA is compared with Compressor based adder circuits .

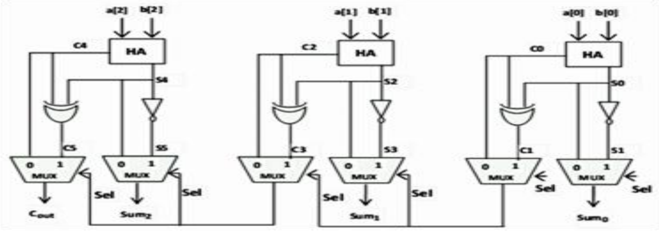


Fig4.4 Proposed Group3 structure for proposed SQRT CSLA

Both compressors based digital adder and reduced complexity SQRT CSLA adder is incorporated into the addition part of Bi-Recoder multiplier independently. The performance of reduced complexity SQRT CSLA based Bi-Recoder is better than the performance of compressors adder based Bi-Recoder due to less hardware complexity of reduced complexity SQRT CSLA. Both compressors based digital adder and reduced complexity SQRT CSLA adder is incorporate addition part of Bi-Recoder multiplier independently. The performance of reduced complexity SQRT CSLA based Bi-Recoder is better than the performance of reduced complexity SQRT CSLA. Hence, this circuit is named as SQRT CSLA. Divided each and every group can operate instantly at same time. Therefore the resultant partial products of the bi-recoder multiplier are added using the sqrt CSLA. The sqrt CSLA are the fastest adders. The carry output of one stage is given as a carry input to the next stage.This proposed SQRT CSLA with vedic multiplier MAC is compared with existing Vedic Multiplier with Ripple Carry Adder.

5. RESULTS AND DISCUSSION

The design of the adders, multipliers and MAC unit are implemented and simulated by using VHDL code in Xilinx ISE 10.2 and verified using Modelsim 6.3c. After synthesis, we run design implementation, which converts logical design into a physical file format that is downloaded to the selected target device, Field Programmable Gate Array (FPGA). The outputs of the adders, multipliers and MAC unit are compared for better performance in power, area and speed. The compared results are given below in Table 5.1.1, 5.1.2 & 5.1.3. and presented in BAR CHART as shown in fig. 5.1, 5.2 & 5.3

TABLE 5.1.1 COMPARISON TABLE OF ADDERS

ADDER	AREA(NO. OF LUTs)	DELAY (ns)	POWER (W)
Ripple Carry Adder	31	27.35	0.065
Carry Save Adder	68	41.25	0.064
Square Root Carry Select Adder	30	24.45	0.066

TABLE 5.1.2 COMPARISON TABLE OF MULTIPLIERS

MULTIPLIER	AREA(NO. OF LUTs)	DELAY (ns)	POWER (W)
Booth Multiplier	765	19.27	0.059
Wallace Tree Multiplier	687	52.304	0.063
Vedic Multiplier	686	36.67	0.070

TABLE 5.1.3 COMPARISON TABLE OF MAC

MAC UNIT	AREA(NO. OF LUTs)	DELAY (ns)	POWER (W)
Vedic Multiplier with Rca	2968	6.919	18.94
Vedic Multiplier with Sqrt Csla	2352	3.403	18.39

BAR CHART REPRESENTATION

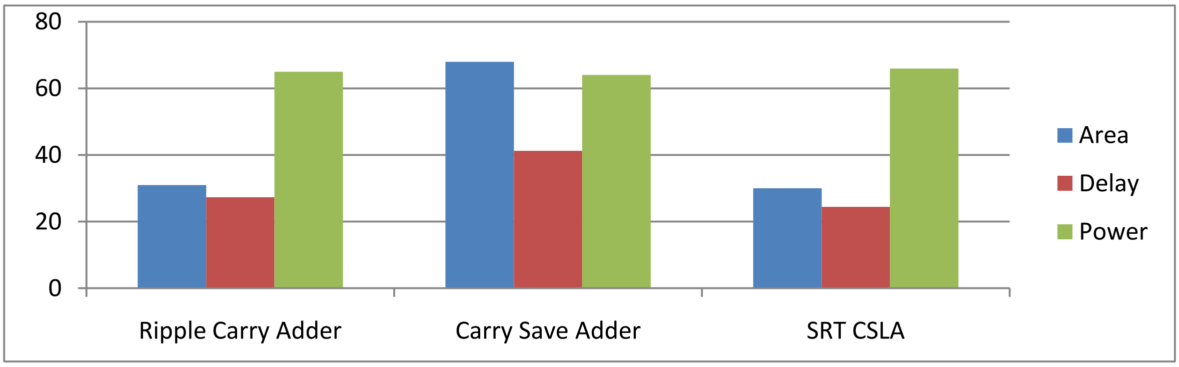


FIG 5.1 PERFORMANCE ANALYSIS OF ADDERS

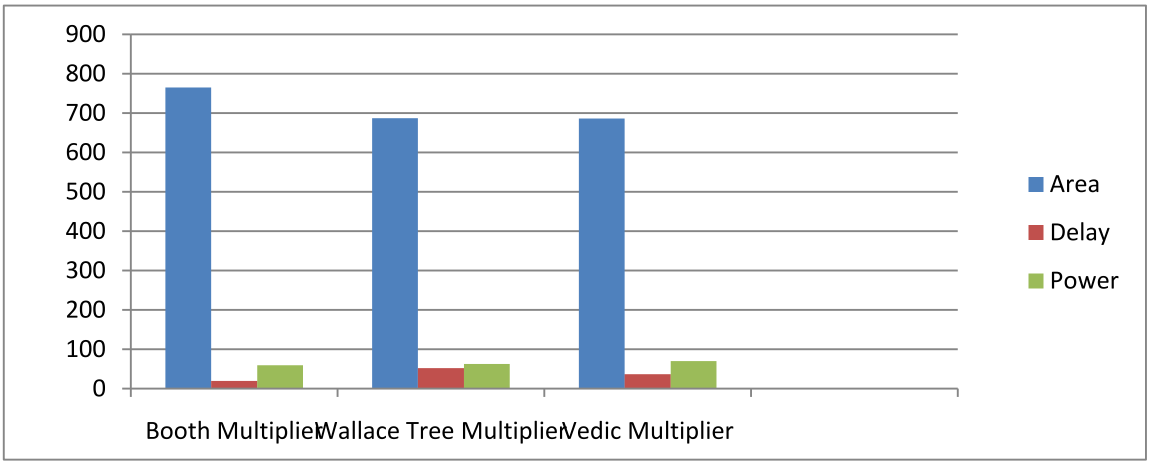


FIG 5.2 PERFORMANCE ANALYSIS OFMULTIPLIERS

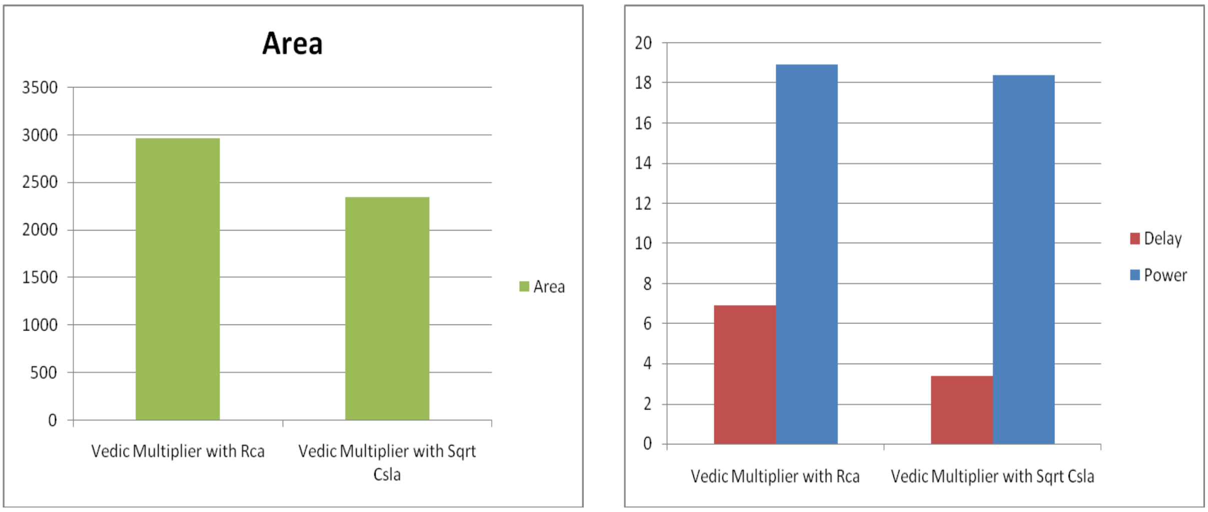


FIG 5.3 PERFORMANCE ANALYSIS OF MAC UNIT

6. CONCLUSION

The implementation of all the multipliers in VHDL code is used to easily understand the different designing parameters effectively. The designs have been synthesized easily with the support of Xilinx IST. In this work, different types of adders (Ripple Carry Adder, Carry Save Adder, Square Root Carry Select Adder) has been designed and evaluated on power , delay and area parameter. The area and delay of Square Root Carry Select Adder is the lowest among other adder types but its power dissipation is high. The compared results shows that the Vedic multiplier has a slightly higher power compared to Booth multiplier and Wallace Tree multiplier. This is due to the tradeoff that the power increases with the reduction of number of LUTs. A Simple approach is proposed in this project work to reduce the power consumption and area of the proposed MAC UNIT. The reduced number of LUTs of this work offers the great advantage in the reduction of power and area of the proposed MAC unit.Vedic Multiplier with Square root CSLA is compared with Vedic Multiplier with RCA in terms of area, delay and power. The number of LUTs, POWER and DELAY is low comparatively than the latter.

REFERENCES

[1] Abdelgawad, A. and Bayoumi, M. (2007) “High speed and area efficient Multiply Accumulate Unit (MAC) for Digital Signal Processing applications”, Proc. IEEE Int. symp. Circuits Syst. (ISCAS), pp. 3199-3202.

[2] Marimuthu, C.N. and Thiangaraj, P. (2008) “Low Power High Performance Multiplier”, ICGST-PDCS, Vol.8.

[3] Raghunath, R.K.J. et al. (1997) “A compact carry-save multiplier architecture and its applications”,Proc. IEEE 40th Midwest Symp. Circuits and Systems, Vol.2, pp. 794-797.

[4] Elguibaly, F. (2000) “A fast parallel multiplier-accumulator using the modified Booth algorithm”, IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing, vol. 47, pp. 902-098.

[5] Senthilpari, C. Ajay Kumar Singh, and Diwadkar, K. (2007) “Low power and high speed 8x8 bit Multiplier Using Non-clocked Pass Transistor Logic”, IEEE, Vol.9/07, pp.1-4244-1355.
[6] Tam Anh Chu, (2002) “Booth Multiplier with Low Power High Performance Input Circuitary”, US Patent, B1.6.393.454 .
[7] Wallace, C.S. (1964) “A suggestion for a fast multiplier”, IEEE Transactions on Electronic Computers, Vol.13, No.1, pp. 14–17.50
[8] Raghunath, R.K. Et al. (1997) “A compact carry-save multiplier architecture and its applications,” Proc. IEEE 40th Midwest Symp. Circuits and Systems, vol. 2, pp. 794-797.
[9] Ohsang Kwon, Nowka, K and Swartzlander, E.E. (2000) “A 16-bitx16-bit MAC design using fast 5:2 compressors,” Proc. Of IEEE International Conference on Application-Specific Systems, Architectures, and Processors, pp. 235 –243.
[10] Ayman Fayed, Walid Elgharbawy, and Magdy Bayoumi, (2004) “A merged Multiply accumulate for high-speed signal processing application,” ICASSP IEEE.
[11] Law, C.F, Rofail, S.S, and Yeo, K.S. (1999)“A Low-Power 16×16-Bit Parallel Multiplier Utilizing Pass-Transistor Logic” IEEE Journal of Solid State circuits, Vol.34, No.10, pp. 1395-1399.
[12] Wallace, C.S. (1967) “A Suggestion for a fast multipliers,” IEEE Trans. Electronic Computers, vol. 13, no.l, pp 14-17.
[13] Tiwari, Gankhuyag, G. and Kim, C.M. and Cho, Y.B. (2008) “Multiplier design based on ancient Indian Vedic mathematics”, Proc. Int SoC Design Conf., pp.65-68.
[14] Ramkumar. B and Kittur, H.M, (2012) “Low-power and area-efficient Carry select adder,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.20, no. 2, pp. 371–375.