ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 2, Issue 11, November 2013

# Low Power and Area Efficient Multiplier for MAC

M.Jayaprakash[1], M.Peer Mohamed[2], Dr.A.Shanmugam[3]

**Abstract— A multiplier is one of the key hardware blocks in most digital and high performance systems such as FIR filters, digital signal processors and microprocessors etc [1]. The area and speed of the multiplier is an important issue, increment in speed results in large area consumption and vice versa. Performance of a system depends to a great extent on the performance of multiplier thus multipliers should be fast and consume less area and hardware. This idea forced us to study and review about the Booth's Algorithm, modified Booth's algorithm and its radix-4 forms. Generally as we know multiplication goes in two basic steps, Partial product and then addition [4]. Hence in this paper we have first tried to design different adders and compare their speed and complexity of circuit i.e. the area occupied. And then we have designed Wallace tree multiplier then followed by Booth's Wallace multiplier and have compared the speed and Power consumption in them. While comparing the adders we found out that Ripple Carry Adder had a smaller area while having lesser speed, in contrast to which Carry Select Adders are high speed but posses a larger area. And a Carry Look Ahead Adder is in between the spectrum having a proper tradeoff between time and area complexities after designing and comparing the adders we turned to multipliers. Initially we went for Parallel Multiplier and then Wallace Tree Multiplier. In the mean time we learned that delay amount was considerably reduced when Carry Save Adders were used in Wallace Tree applications. Then we turned to Booths Multiplier and designed Radix-4 modified booth multiplier and analyzed the performance of all the multipliers.**

***Keywords*—** Booth multiplier, digital signal processing (DSP), FIR filters.

## I. INTRODUCTION

As the scale of integration keeps growing, more and more sophisticated signal processing systems are being implemented on a VLSI chip. These signal processing applications not only demand great computation capacity but also consume considerable amount of energy. While performance and Area remain to be the two major design tolls, power consumption has become a critical concern in today's VLSI system design[2]. The need for low-power VLSI system arises from two main forces. First, with the steady growth of operating frequency and processing capacity

per chip, large currents have to be delivered and the heat due to large power consumption must be removed by proper cooling techniques. Second, battery life in portable electronic devices is limited. Low power design directly leads to prolonged operation time in these portable devices. Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore low-power multiplier design has been an important part in low- power VLSI system design. There has been extensive work on low-power multipliers at technology,
physical, circuit and logic levels. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system[3]. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. As a result, a whole spectrum of multipliers with different area- speed constraints has been designed with fully parallel.

In chapter II, the different types of adders are discussed since adder forms the basic building block of the multiplier. In chapter III, different multipliers are compared. The analysis of adders and multipliers is done in chapter IV.

## II. THE ADDERS

The adder topology used in this work are ripple carry adder, carry look ahead adder and carry select adder.

### *A. Ripple Carry Adder (RCA)*

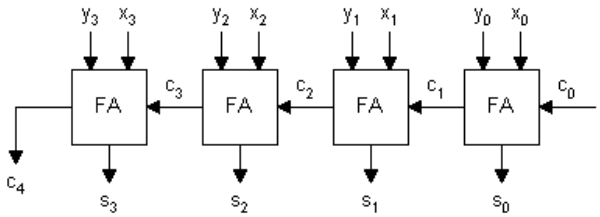



Fig.1. 4-bit Ripple Carry Adder

The ripple carry adder is constructed by cascading full adders (FA) blocks in series. One full adder is responsible for the addition of two binary digits at any stage of the ripple carry. The carryout of one stage is fed directly to the carry-in of the next stage. Even though this is a simple adder and can be used to add unrestricted bit length numbers, it is however

***Manuscript received Oct, 2013.***

*M.Jayaprakash is with the Department of Electronics and Communication at SCAD Institute of Technology, Coimbatore, India.9791903635;*

*M.Peer Mohamed is with the Department of Electronics and Communication at SCAD Institute of Technology, Coimbatore, India. 7845840768*

*Dr.A.Shanmugam is the principal of Bannari Amman Institute of Technology, Sathyamangalam, India.*

not very efficient when large bit numbers are used. One of the most serious drawbacks of this adder is that the delay increases linearly with the bit length[4]. The worst-case delay of the RCA is when a carry signal transition ripples through all stages of adder chain from the least significant bit to the most significant bit, which is approximated by

$$t=(n-1)t_c+t_s$$

where tc = delay through the carry stage of a full adder

ts = delay to compute the sum of the last stage.

Delay from Carry-in to Carry-out is more important than from A to carry-out or carry-in to SUM, because the carry-propagation chain will determine the latency of the whole circuit for a Ripple-Carry adder. Considering the above worst-case signal propagation path we can thus write the following equation.

For a k-bit RCA worst case path delay is

TRCA-k bit = TFA(x0, y0 c0) + (k-2)* TFA(Cin Ci) + TFA(Cin Sk-1)

*B. Carry Select Adders (CSLA)*

In Carry select adder scheme, blocks of bits are added in two ways: one assuming a carry-in of 0 and the other with a carry-in of 1.This results in two pre computed sum and carry-out signal pairs **(s0i-1:k , c0i ; s1i-1:k , c1i) ,** later as the block's true carry-in (**ck**) becomes known, the correct signal pairs are selected[13]. Generally multiplexers are used to propagate carries.

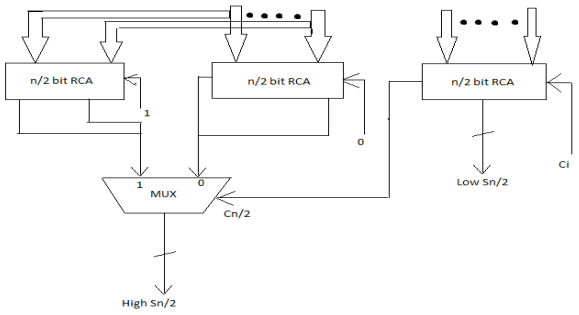



Fig.2. Carry Select Adder

We always go for Carry Select Adder while working with smaller no of bits.The propagation path delay is given by

si-1:k = ck s0i-1:k + ck s1i-1:k _ ci = ck c0i + ck c1i

*C. Carry look ahead Adder*

In ripple carry adders, the carry propagation time is the major speed limiting factor[13]. Accordingly, reducing the carry propagation delay of adders is of great importance. Different logic design approaches have been employed to overcome the carry propagation problem. Carry look-ahead adder is designed to overcome the latency introduced by the rippling effect of the carry bits. The propagation delay occurred in the parallel adders can be eliminated by carry look ahead adder. This adder is based on the principle of looking at the lower order bits of the augends and addend if a higher order carry is generated. This adder reduces the carry delay by reducing the number of gates through which a carry signal must propagate. Carry look ahead depends on two things: Calculating for each digit position, whether that position is going to propagate a carry if one comes in from the right and combining these calculated values to be able to deduce quickly whether, for each group of digits, that group is going to propagate a carry that comes in from the right. The net effect is that the carries start by propagating slowly through each 4-bit group, just as in a ripple-carry system, but then moves 4 times faster, leaping from one look ahead carry unit to the next. Finally, within each group that receives a carry, the carry propagates slowly within the digits in that group.

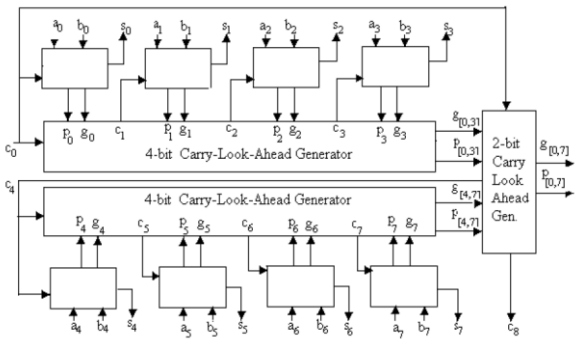



Fig.3. 8-bit Carry Look ahead Adder

This adder consists of three stages: a propagate block/ generate block, a sum generator and carry generator[6]. The generate block can be realized using the expression

$G_i = A_i.B_i$ for i=0,1,2,3

Similarly the propagate block can be realized using the expression

$P_i = A_i \oplus B_i$ for i=0,1,2,3

The carry output of the (i-1) th stage is obtained from

$C_i(cout) = G_i + P_iC_{i+1}$ for i=0,1,2,3

The sum output can be obtained using

$S_i = A_i \oplus B_iC_{i+1}$ for i=0,1,2,3

An 8 bit look ahead adder using two four bit look ahead block is shown in Figure 3 the COUT of the 4-bit CLA is given as the CIN for the second 4-bit CLA.

## III. THE MULTIPLIERS

*A.The Wallace Tree Multiplier*

The Wallace tree multiplier is considerably faster than a simple array multiplier because its height is logarithmic in word size, not linear. However, in addition to the large number of adders required, the Wallace tree's wiring is much less regular and more complicated. As a result, Wallace trees are often avoided by designers, while design complexity is a concern to them. Wallace tree styles use a log-depth tree network for reduction. Faster, but irregular, they trade ease of layout for speed. Wallace tree styles are generally avoided for low power applications, since excess of wiring is likely to consume extra power [4][5]. While subsequently faster than Carry-save structure for large bit multipliers, the Wallace tree multiplier has the disadvantage of being very irregular, which complicates the task of coming with an efficient layout. The Wallace tree multiplier is a high speed multiplier. The summing of the partial product bits in

parallel using a tree of carry-save adders became generally known as the "Wallace Tree".

Three step processes are used to multiply two numbers.

1. Formation of bit products.
2. Reduction of the bit product matrix into a two row matrix by means of a carry save adder.
3. Summation of remaining two rows using a faster Carry Look Ahead Adder (CLA).

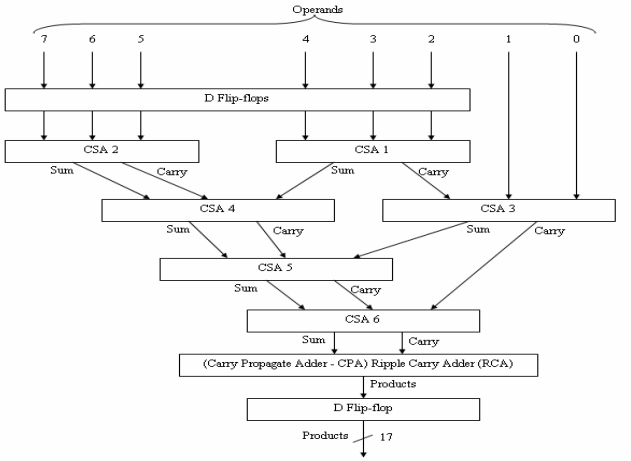



Fig.4. Wallace Tree Multiplier

In order to design an n-bit Wallace tree Multiplier (Generic: =N) an algorithm was derived from the flow diagram developed. The flow diagram shows the intermediate state reductions of the multipliers are being done by Carry save adders and half adders while the final step additions being done by a Carry Look Ahead Adder. After generating the flow diagram for 8-bit × 8-bit we generalized the algorithm for n-bit and hence we designed a Generic Wallace Tree.

*B. The Booth's Multiplier*

Though Wallace Tree multipliers were faster than the traditional Carry Save Method, it also was very irregular and hence was complicated while drawing the Layouts. Slowly when multiplier bits gets beyond 32-bits large numbers of logic gates are required and hence also more interconnecting wires which makes chip design large and slows down operating speed Booth multiplier can be used in different modes such as radix-2, radix-4, radix-8 etc. But we decided to use Radix-4 Booth's Algorithm because of number of Partial products is reduced to n/2.

*Booth Multiplication Algorithm*

One of the solutions realizing high speed multipliers is to enhance parallelism which helps in decreasing the number of subsequent calculation stages. The Original version of Booth's multiplier (Radix – 2) had two drawbacks.

1.The number of add / subtract operations became variable and hence became inconvenient while designing Parallel multipliers.

2.The Algorithm becomes inefficient when there are isolated 1s .

These problems are overcome by using Radix 4 Booth's Algorithm which can scan strings of three bits with the algorithm given below. The design of Booth's multiplier in this project consists of four Modified Booth Encoded (MBE), four sign extension corrector, four partial product generators (comprises of 5:1 multiplexer) and finally a Wallace Tree Adder. This Booth multiplier technique is to increase speed by reducing the number of partial products by half. Since an 8-bit booth multiplier is used, so there are only four partial products that need to be added instead of eight partial products generated using conventional multiplier. The architecture design for the modified Booths Algorithm is shown in figure 5.

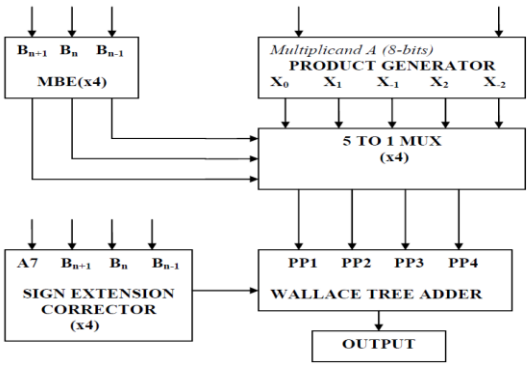



Fig.5. Modified Booth Multiplier Architecture

*Modified Booth Encoder*

Modified Booth encoding is most often used to avoid variable size partial product arrays. Before designing a MBE, the multiplier has to be converted into a Radix-4 number by dividing them into three digits respectively as shown in Booth Encoder Table 1. Prior to convert the multiplier, a zero is appended into the Least Significant Bit (LSB) of the multiplier[15]. The figure 5 shows that the multiplier has been divided into four partitions and hence that mean four partial products will be generated using booth multiplier approach instead of eight partial products being generated using conventional multiplier.

$$Z_n = -2* B_{n+1} + B_n + B_{n-1}$$

Let's take an example of converting an 8-bit number into a Radix-4 number. Let the number be -46 = 1 1 0 1 1 1 0 0. Now we have to append a '0' to the LSB. Hence the new number becomes a 9-digit number, that is 1 1 0 1 1 1 0 0 0. This is now further encoded into Radix-4 numbers according to the following given table. Starting from right we have 0*Multiplicand, -1*Multiplicand, 2*Multiplicand, -1*Multiplicand.

| $B_{n+1}$ | $B_n$ | $B_{n-1}$ | $Z_n$ | Partial Product | 1M | 2M | 3M |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1×Multiplicand | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1×Multiplicand | 0 | 1 | 0 |
| 0 | 1 | 1 | 2 | 2×Multiplicand | 1 | 0 | 0 |
| 1 | 0 | 0 | -2 | -2×Multiplicand | 1 | 0 | 1 |
| 1 | 0 | 1 | -1 | -1×Multiplicand | 0 | 1 | 1 |
| 1 | 1 | 0 | -1 | -1×Multiplicand | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |

Table 1. Modified Booth Encoder's table

Table 1 shows Bn+1, Bn and Bn-1 which are three bits wide binary numbers of the multiplier Bin which Bn+1 is the most significant bit (MSB) and Bn-1 is the least significant bit (LSB). Zn is representing the Radix-4 number of the 3-bit binary multiplier number. For example, if the 3-bit multiplier value is "111", so it means that multiplicand A will be 0.And it's the same for others either to multiply the multiplicand by

-1, -2 and so on depending on 3 digit number. And thing to note is generated numbers are all of 9-bit. From the table 4.1, the M, 2M and 3M are the elect control signals for the partial product generator[7][8]. It will determine whether the multiplicand is multiplied by 0,-1, 2 or -2. M and 2M are designed as an active low circuit which means if let's say the multiplicand should be multiplied by 1 then the M select signal will be set to low "0" whereas If the multiplicand should be multiplied by 2 then the 2M select signal will be set to low "0". The 3M is representing the sign bit control signal and its active high circuit which means if the multiplicand should be multiplied by -1 or -2, then the sign, 3M will be set to high "1"[16].

## IV. ANALYSIS OF ADDERS AND MULTIPLIERS

### A. *Analysis of Adders*

We compared 3- different adders Ripple Carry Adders, Carry Select Adders and the Carry Look Ahead Adders. The basic purpose of our experiment was to know the time and power trade-offs between different adders whish will give us a clear picture of which adder suits best in which type of situation during design process. Hence below we present both the theoretical and practical comparisons of all the three adders whish were taken into consideration.

| Adder Name | Delay for n-bit | Area for n-bit | Area Delay Product |
|---|---|---|---|
| Ripple Carry Adder (RCA) | 2n | 7n | $14n^2$ |
| Carry Select Adder (CSLA) | $2.8(n)^{1/2}$ | 14n | $39.6(n)^{3/2}$ |
| Carry Look ahead Adder (CLA) | $4\log_2 n$ | 4n | $16n\log_2 n$ |

Table 2. Theoretical Area Delay product

| Adder Name | Delay for 8-bit |
|---|---|
| Ripple Carry Adder (RCA) | 20.2 ns |
| Carry Select Adder (CSLA) | 12.4 ns |
| Carry Look ahead Adder (CLA) | 13.4 ns |

Table 3. Simulated Adders Delay

As can be seen above we have stated the theoretical comparison of area required and both the theoretical and simulated value of time required. The values stated above are the values for 8-bit adders. So analyzing the above facts we reached at the following conclusions about different adders and intelligent use of them in different circumstances according to the space time trade-off. The results can be summarized as follows.

1. Regarding the circuit area complexity in the adder architectures, the ripple-carry adder (RCA) in the first class is the most efficient one, but the carry select adder (CSLA) in the fourth class with highest complexity is the least efficient one.

2.Considering the circuit delay time, carry select adder (CSLA) is the fastest one for every n-bit length, so has the shortest delay. Otherwise, ripple carry adder (RCA) is the slowest one, due to the long carry propagation.

We defined a term area-delay product which gave us the clear picture of the space-time tradeoff. It is worthy to note that while we consider all the adders discussed above ripple carry adders and carry select adders are the two sides of the spectrum. Because, while ripple carry adders have a smaller area and lesser speed, in contrast to which carry select adders have high speed (nearly twice the speed f ripple carry adders) and occupy a larger area. But carry look ahead adder (CLA) has a proper balance between both the area occupied and time required. Hence among the three, carry look ahead adder has the least area delay product. Hence we should use carry look ahead adders when it comes to optimization with both area and time. For an instance, the last stage of the wallace tree adder in booth multiplier is a carry look ahead adder.

### B. *Analysis of Multipliers*

The different multipliers are compared on the basis of their speed and power parameters. We used Xilinx ISE version 10.2 for our simulation of different multipliers and knowing their delays. We analyzed Array Multipliers, Wallace Tree Multipliers and Booth Multiplier (Radix-2 and Radix-4) and analyzed their speed and power consumption using the above.

| Parameters | Values |
|---|---|
| Number of Slices | 96 |
| Number of 4-input LUTs | 178 |
| Number of Bonded Input | 32 |
| Number of Bonded Output | 32 |
| Power | 124mW |

Table 4. Array Multiplier

| Parameters | Values |
|---|---|
| Number of Slices | 72 |
| Number of 4-input LUTs | 130 |
| Number of Bonded Input | 32 |
| Number of Bonded Output | 32 |
| Power | 114mW |

Table 5. Booth Multiplier (Radix-2)

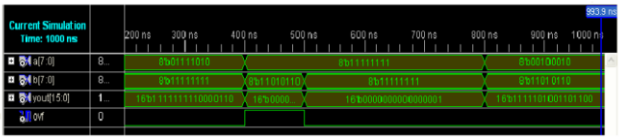
| Parameters | Values |
|---|---|
| Number of Slices | 69 |
| Number of 4-input LUTs | 125 |
| Number of Bonded Input | 32 |
| Number of Bonded Output | 32 |
| Power | 81mW |

Table 6. Wallace Tree Multiplier

| Parameters | Values |
|---|---|
| Number of Slices | 96 |
| Number of 4-input LUTs | 178 |
| Number of Bonded Input | 32 |
| Number of Bonded Output | 32 |
| Power | 76mW |

Table 7. Booth Multiplier (Radix-4)

If we compare the above values among each other we can observe that the Array Multiplier is the worst case multiplier consuming highest amount of power. Then comes the Radix – 2 booth multiplier which consumes lesser power than array multiplier. The Wallace Tree multiplier and Booth Multiplier Radix-4 have nearly same amount of delay while Radix-4 Booth consuming lesser power than the other. Hence we reach to a conclusion that Booth Radix-4 Multiplier is best for situations requiring Low power Applications.




Booth Multiplier Output

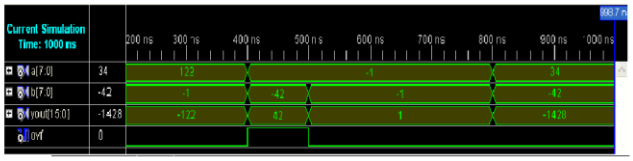



Fig.6. Output of Booth Multiplier (Radix-8)

## V. CONCLUSION AND FUTURE WORK

We studied about different adders among compared them by different criteria like Area, Time and then Area-Delay Product etc. so that we can judge to know which adder was best suited for situation. After comparing all we came to a conclusion that Carry Select Adders are best suited for situations where Speed is the only criteria. Similarly Ripple Carry Adders are best suited for Low Power Applications. But Among all the Carry Look Ahead Adder had the least Area-Delay product that tells us that, it is suitable for situations where both low power and fastness are criteria such that we need a proper balance between both. Coming to Multipliers we studied different Multipliers starting from Array Multiplier to Wallace Tree, Booth Multipliers. We found that parallel multipliers are much better than the serial multipliers due to less area consumption and hence the less power consumption. Comparing Radix-2 and Radix-4 booth multipliers we found that radix-4 consumes less power than radix-2, because radix-4 uses almost half number of iterations than radix-2.We saw Wallace tree having nearly same delay as of radix-4 multipliers where as consuming a little more power than the former. The Power efficiency of the circuits can be increased by proper recording schemes.

## REFERENCES

[1] K.H. Tsoi, P.H.W. Leong, "Mullet - a parallel multiplier generator," fpl, pp.691-694, International Conference on Field Programmable Logic and Applications, 2005.

[2] S. Tahmasbi Oskuii, P. G. Kjeldsberg, and O. Gustafsson, "Transition activity aware design of reduction-stages for parallel multipliers," in Proc. 17th Great Lakes Symp. On VLSI, March 2007, pp. 120–125.

[3] Ayman A. Fayed, Magdy A. Bayoumi, "A Novel Architecture for Low-Power Design of Parallel Multipliers," wvlsi, pp.0149, IEEE Computer Society Workshop on VLSI 2001.

[4] M. 0. Lakshmanan, Alauddin Mohd Ali, "High Performance Parallel Multiplier Using Wallace-Booth Algorithm," IEEE International Conference on Semiconductor Electronics, pp. 433-436, 2002

[5] Animul islam, M.W. Akram, S.D. pable ,Mohd. Hasan, "Design and Analysis of Robust Dual
Threshold CMOS Full Adder Circuit in 32 nm Technology", International Conference on Advances
in Recent Technologies in Communication and Computing,2010.

[6] Deepa Sinha, Tripti Sharma, k.G.Sharma, Prof.B.P.Singh, "Design and Analysis of low Power 1-bit
Full Adder Cell",IEEE, 2011.

[7] Nabihah Ahmad, Rezaul Hasan, "A new Design of XOR-XNOR gates for Low Power application",International Conference on Electronic Devices,Systems and Applications(ICEDSA) ,2011.

[8] R.Uma, "4-Bit Fast Adder Design: Topology and Layout with Self-Resetting Logic for Low Power VLSI Circuits", International Journal of Advanced Engineering Sciences and Technology, Vol No. 7,Issue No. 2, 197 – 205.

[9] David J. Willingham and izzet Kale, "A Ternary Adiabatic Logic (TAL) Implementation of a Four-Trit Full-Adder,IEEE, 2011.

[10] Padma Devi, Ashima Girdher and Balwinder Singh, "Improved Carry Select Adder with Reduced Area and Low Power Consumption", International Journal of Computer Application,Vol 3.No.4, June 2010 .

[11] B.Ramkumar, Harish M Kittur, P.Mahesh Kannan, "ASIC Implementation of Modified Faster Carry Save Adder", European Journal of Scientific Research ISSN 1450-216X Vol.42 No.1, pp.53-58,2010.

[12] Y. Sunil Gavaskar Reddy and V.V.G.S.Rajendra Prasad, "Power Comparison of CMOS and Adiabatic Full Adder Circuits", International Journal of VLSI design & Communication Systems (VLSICS) Vol.2, No.3, September 2011

[13] Mariano Aguirre-Hernandez and Monico Linares-Aranda, "CMOS Full-Adders for Energy-Efficient Arithmetic Applications", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol.19, No. 4, April 2011.

[14] Ning Zhu, Wang Ling Goh, Weija Zhang, Kiat Seng Yeo, and Zhi Hui Kong, "Design of Low-Power High-Speed Truncation-Error-Tolerant Adder and Its Application in Digital Signal Processing", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 18, No. 8, August 2010.

[15] Sreehari Veeramachaneni, M.B. Srinivas, "New Improved 1-Bit Full Adder Cells", IEEE, 2008.

[16]Young-Ho Seo and Dong-Wook Kim,"A new VLSI architecture of parallel multiplier-accumulator based on radix-2 modified Booth algorithm", in IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 18, no. 2, pp.201-208, February 2010.

[17]C.N. Marimuthu and P. Thangaraj, "Low power high performance multiplier", Proc. ICGST-PDCS,vol. 8, pp.31–38, December 2008.

Jayaprakash. M, HOD in charge and Assistant Professor in the Department of ECE in SCAD Institute of Technology, Coimbatore, was born on 05-02-1981. He was graduated at Bharathiar University in B.E., Electrical and Electronics Engineering in 2002. He has done M.E. in Applied Electronics in 2004 at Anna University and currently pursuing Ph.D., under guidance of Dr.A.Shanmugam, Principal, Bannari Amman Institute of Technology. In addition to this he has completed the course Advanced PLC programming and PLC based DCS. He was working as an Assistant Professor in the Department of ECE in Park College of Engineering and Technology. He has more than nine years of teaching experience. He has carried out 4 projects, has undergone three Training Programmes, and has presented 5 papers in seminars and conferences. He has been holding many responsibilities. He has great interest in VLSI Design and Embedded Systems.

Peer Mohamed.M, Assistant Professor in the Department of ECE in SCAD Institute of Technology, Coimbatore, was born on 16-03-1989. He was graduated at Anna University in B.E., Electronics and Communication Engineering in 2010. He has done M.E. in VLSI Design in 2013 at Government College of Technology, Coimbatore. He worked as a Programmer at Elysium Technologies Pvt Ltd. He has carried out 10 projects, has undergone two Training Programmes and has presented 6 papers in seminars and conferences. He has great interest in VLSI Design and DSP.

**Dr.A.Shanmugam** received the BE Degree in PSG College of Technology in 1972, Coimbatore and ME Degree from College of Engineering, Guindy, Chennai in 1978 and Doctor of Philosophy in Electrical Engineering from Bharathiar University, Coimbatore in 1994.From 1972–76, he worked as Testing Engineer in Testing and Development Centre, Chennai. He was working as a Lecturer in Annamalai University in 1978. He was the Professor and Head of the Department of Electronics and Communication Engineering at PSG College of Technology, Coimbatore during 1999 to 2004. He authored a book entitled "Computer Communication Networks' which is published by ISTE, New Delhi, 2000.He is currently the Principal of Bannari Amman Institute of Technology, Sathyamangalam. He is on the editorial board of International Journal Artificial Intelligence in Engineering & Technology (ICAIET), University of Malaysia, International Journal on "Systemics, Cybernetics and Informatics (IJSCI)" Pentagram Research Centre, Hyderabad, India. He is member of the IEEE, the IEEE computer society.