

# Design of High Speed Pipelined Merged MAC Using Radix-4 MBA and Carry Select Adder

Aishwarya. E. V.  
M.Tech Student, SENSE  
VIT University  
Vellore, India

Aarthy. M  
Asst. Professor, SENSE  
VIT University  
Vellore, India

**Abstract**— In this paper, we have presented a parallel Multiplier accumulator unit for high speed arithmetic. Accumulation process in a MAC has the largest delay and hence multiplication and accumulation are merged inside the Carry Save Adder (CSA) tree to improve the performance. This CSA tree is based on radix-4 modified booth algorithm which uses 1's complement form and its array is modified for the sign bits. The number of inputs to the final adder is decreased by processing the least significant bits in advance. For the final adder, three adder architectures a regular SQRD CSLA, modified SQRD CSLA and SQRD CSLA with Common Boolean Logic are compared. On the basis of the synthesized results, the adder with minimum delay and gate count (SQRD CSLA with Common Boolean Logic) is chosen. Also, the MAC is pipelined in an optimized way by carrying out the accumulation process of the intermediate sum and carry instead of the final output. The synthesis of the design is performed using Cadence RTL compiler TSMC 45nm standard cell library. The delay for 16-bit MAC based on radix-4 MBA is found to be 5.4ns and hence a 17.84% reduction in delay is achieved due to the reduction of partial products rows and carry select adder at the final stage.

**Keywords**—Radix-4 MBA, Merged MAC, SQRD Carry select adder

## INTRODUCTION

Multiplication and Addition are the basic and frequently used operation in most of the digital systems. The fast growth in digital signal processing system demands high speed processing of data which has been the widely researched area over a couple of decades. Multiply Accumulate (MAC) unit forms the fundamental unit in most of the DSP applications. In a digital system, multiplication operation has longer delay compared to other operations. Improving the speed of the MAC unit plays a vital role in increasing demand in the performance of the DSP systems as it determines the execution time of them. In general, a multiplier has 3 stages. They are partial product generation, partial product reduction and final addition. Modified booth algorithms (MBA) [1] can be used for high speed multiplication. But critical path delay in the multiplier is crucial in reducing the overall delay [2].

Generally, a multiplier uses booth algorithm [3] and full array of full adders. Wallace tree [4] is used instead of full adder array. The multiplier in [5], [6] consists of Booth encoder, Wallace tree and a final adder. The partial product generated undergoes a series of addition operation which affects the speed of it. Hence reducing the number of partial products is a proved effective way. In most of the high speed designs, MBA algorithm is used to reduce the number of

partial products and Wallace tree is used to speed up the addition of partial products. Parallel architectures in [7], [8] increases the speed of MBA.

Multiplier and accumulator are merged in the architecture proposed by Elguibaly [9] in which the accumulation process is combined with the carry save adder tree which compresses the partial products. Hence the critical path delay was reduced because the adders in the accumulation process is eliminated. Since the output of final adder was fed back for accumulation, the output rate needed improvement. Adder block and the accumulator are merged in the architecture proposed in [10] in which two separate N/2 bit adders were used to accumulate N bit MAC. The work in [11] also uses the merging technique using 4-2 compressors. The work in [12] is one of the efficient ways to implement a merged MAC that shows a better delay result by reducing the number of inputs given to the final adder.

From these analysis of architectures, we observe that the delay is improved by using higher radix MBA which eventually shortens the number of multiplication and thereby increases the speed. In this paper, the work done in [12] is modified by using higher radix MBA and also by replacing their final adder with an efficient SQRD carry select adder that uses Common Boolean Logic(CBL).

## I. OVERVIEW OF MAC

In general, a multiplier has three operational steps. The first is the generation of partial products using booth encoding is done from the multiplicand X and multiplier Y. The second step involves partial product compression that adds the partial products into sum and carry. The third step is the final addition wherein final multiplication result is obtained by adding the sum and carry. MAC has an additional step of accumulating the multiplied result.

Mathematically, a MAC is expressed as

$$P = A \times B + Z$$

where A, B and Z are multiplicand, multiplier and previous multiplication result respectively. The proposed MAC architecture in [12] has a hybrid CSA tree which is based on the equation derived. The accumulation step is clearly merged into the addition of partial products. The n-bits A and B are converted into (n+1) bits partial products by booth encoder.

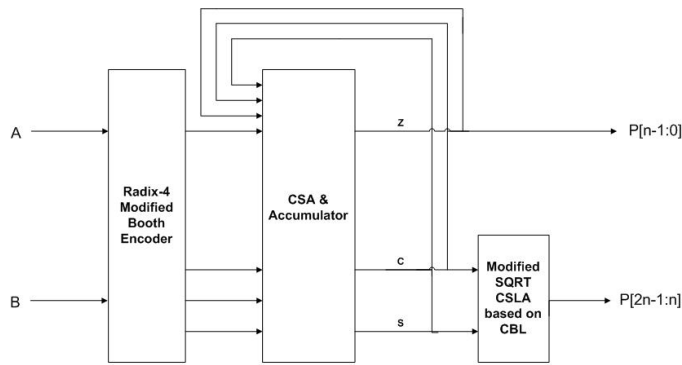


Figure 1. Modified Merged MAC architecture

The accumulation process along with the addition of partial products are carried out in the CSA tree and accumulator as a result of which n-bits  $S$ ,  $C$  and  $Z$  bits are generated. These values are feedback for the accumulation. The  $P[n-1 : 0]$  bits are already generated in CSA tree and  $P[2n-1 : n]$  bits are obtained by adding the  $S$  and  $C$  in the final adder.

II. MODIFIED MERGED MAC

The MAC architecture in [12] is modified by using radix-4 MBA for partial product generation and the final adder with a proposed Sqrt carry select adder based on CBL. This is proposed to further decrease the delay of the MAC. The stages of the MAC in Fig.1 are explained in the following sections.

A. Modified Booth Algorithm(MBA)

The main purpose of using booth encoder is to decrease the number of partial products as compared to basic booth multiplier. Booth multiplication is a technique that allows for smaller, faster multiplication circuits, by recoding the numbers that are multiplied [8]. It is done by reducing the number of partial products by half, by using radix-4 Booth recoding technique. In this, instead of shifting and adding for every column of the multiplier term and multiplying by 1 or 0, we only take every second column, and multiply by  $\pm 1$ ,  $\pm 2$ , or 0, to obtain the same results. Hence, the number of partial products is decreased to half. Booth recoding of the multiplier term is done by considering the bits in blocks of three or triplets, such that each block overlaps the previous block by one bit. Grouping starts from the LSB, and the first block only uses two bits of the multiplier. Fig.2 shows the grouping of bits from the multiplier in the form of triplets. The encoded digit in the multiplier performs a certain operation on the multiplicand  $X$ . The multiplier  $B$  is divided into groups of two bits and the algorithm is applied to each group. Table.1 shows the rules to generate the encoded signals by MBE scheme and Fig.3 shows the corresponding logic diagram. The Booth decoder generates the partial products using the encoded signals as shown in Fig.4.

B. Hybrid CSA Tree

The architecture of the hybrid CSA tree for radix-4 MBA for an  $8 \times 8$  bit operation is shown in Fig.5. The  $S[i]$  and  $C[i]$

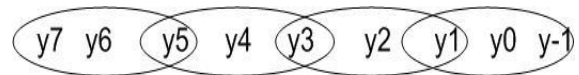


Figure 2. Grouping of multiplier bits

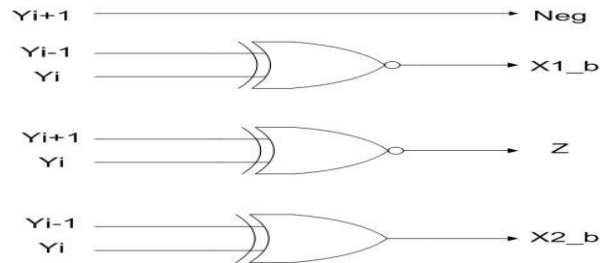


Figure 3. Booth Encoder logic

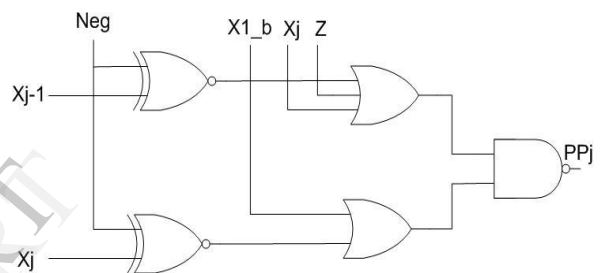


Figure 4. Booth Decoder logic

Table 1. Truth table for Modified Booth Encoding

$Y_{i+1}$	$Y_i$	$Y_{i-1}$	Value	$X1\_b$	$X2\_b$	Neg	Z
0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	1
0	1	0	1	0	1	0	0
0	1	0	2	1	0	0	0
1	0	1	-2	1	0	1	0
1	0	1	-1	0	1	1	0
1	1	1	-1	0	1	1	1
1	1	1	0	1	0	1	1

are the sum and carry of  $i$ th bit.  $Z[i]$  is the addition of the lower  $i$ th bit of partial product that are added in advance.  $S'[i]$ ,  $C'[i]$  and  $Z'[i]$  are the previous results that are fed back for accumulation.  $Pj[i]$  corresponds to the  $i$ th bit of  $j$ th partial product.  $S_i$  is to simplify the sign bit and  $N_i$  is to compensate 1's complement into 2's complement number. The grey squares represent half adders and the white squares represent full adders. The five inputs symbol on the right represents 2-bit carry look-ahead adder. Since the lower bits are added in advance, the number of inputs to the final adder is reduced thereby decreasing the delay.

C. Final Adder

For further reducing the delay, three different Sqrt CSLA architectures are analyzed for power and delay. The better one is replaced with the 4-bit CLA in the previous work [12].

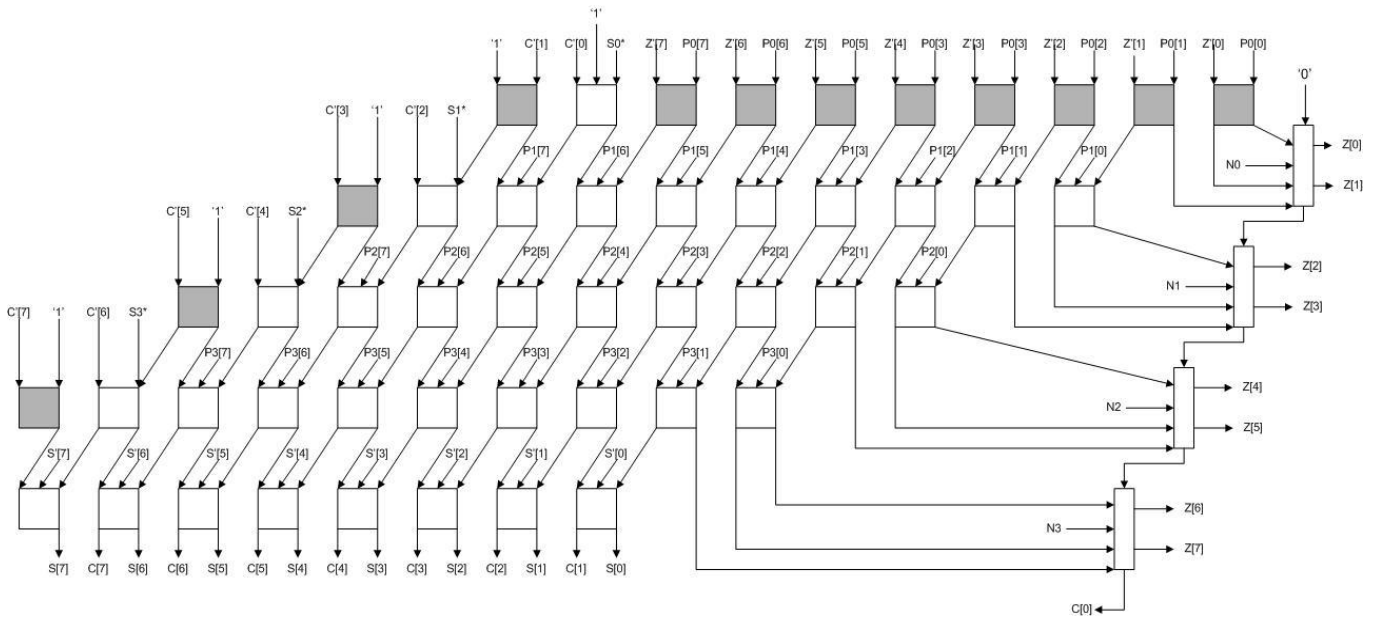


Figure 5. Hybrid CSA Tree for Radix-4 MBA

1) Regular Sqrt CSLA

The regular Sqrt CSLA structure for 16 bits is shown in Fig.6. It consists of five groups of different sized RCA. Every group has two RCA and a MUX. The Sqrt CSLA CSLA is an improved version of the conventional CSLA in terms of area and delay. The Time delay of conventional CSLA is decreased by increasing one more input into each set of adders than in previous set of adders. This modified architecture of increasing inputs is known as Sqrt CSLA. In Sqrt CSLA, the second group has two sets of 3-bit RCA. *c1* is the select line of the 3:2 MUX. If *c1*=0 first RCA output is selected by the MUX else second RCA output is selected.

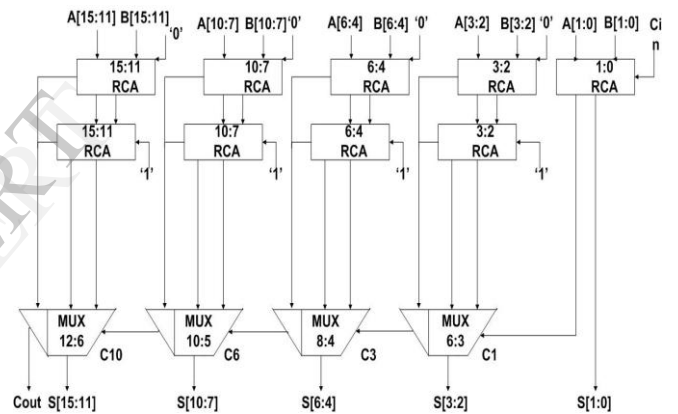


Figure 6. Regular Sqrt CSLA

2) Modified Sqrt CSLA

In the modified Sqrt CSLA, the second RCA in every adder group is replaced by Binary to Excess-1 convertor (BEC). This replaced BEC performs the same operation as that of the replaced RCA with *Cin*=1. The first RCA computes the sum for *Cin*=0. The BEC simply adds 1 to the RCA output. Depending on the select line which is the carry of the previous adder group, the MUX selects the output sum and carry. The block diagram of modified Sqrt CSLA is shown in Fig.7. This structure consumes less power than regular Sqrt CSLA since less number of transistors are involved.

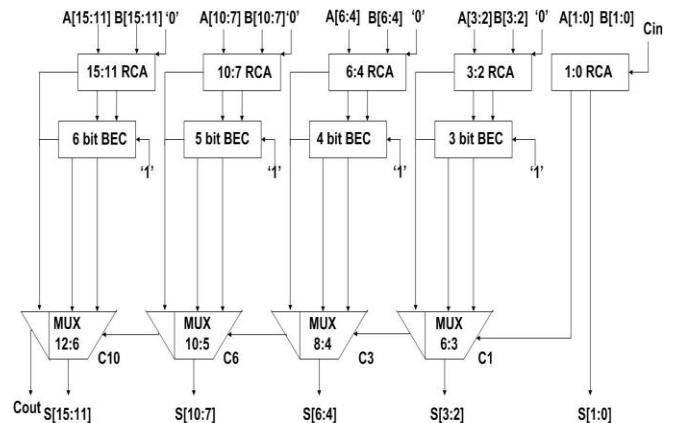


Figure 7. Modified Sqrt CSLA

3) Proposed Sqrt CSLA based on Common Boolean Logic

Common Boolean Logic is the technique used to reduce the duplicate usage of transistors. Designs incorporating the CBL has a significant reduction in area. Hence a new Sqrt adder based on CBL is proposed as it has better area, delay and power. For the full adder, Common Boolean functions are applied. It can be seen that when *Cin*=0, the sum values are the inverse of the sum values obtained when *Cin*=1. So it is sufficient to just invert the sum values obtained from

XOR gate using an inverter. Similarly carry is generated by using AND gate when *Cin*=0 and OR gate when *Cin*=1.

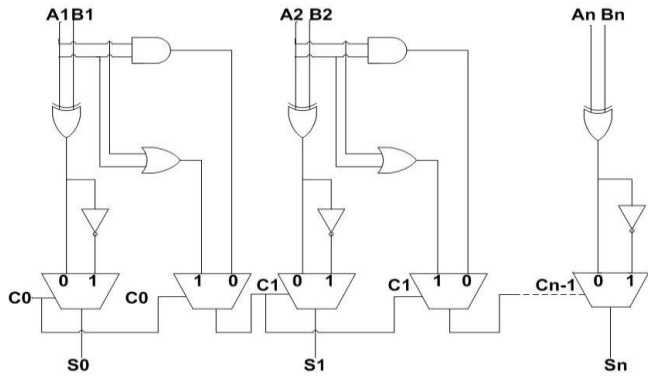


Figure 8. Internal structure of Sqrt CSLA using CBL

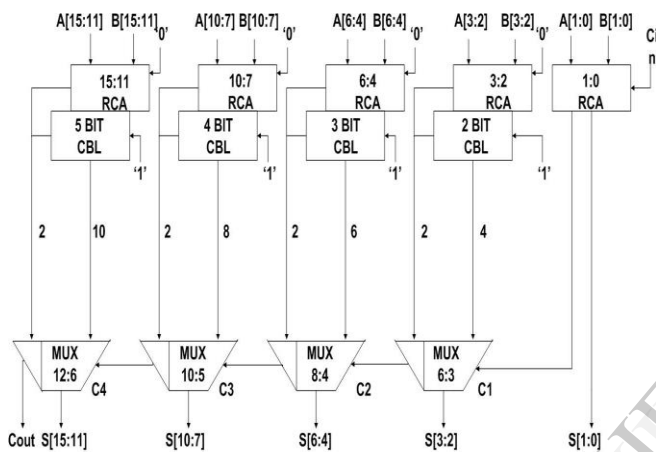


Figure 9. Proposed Sqrt CSLA using CBL

In this way, the sum and the carry are obtained in a parallel way the sum and the carry are obtained in a parallel way. The Binary to Excess-1 converter circuit is replaced by common Boolean Logic. This proposed adder is faster than the modified Sqrt CSLA. The structure of the Sqrt CSLA based on CBL is shown in Fig.8. There is area delay trade off in. Hence the proposed Sqrt CSLA outperforms other CSLA. Fig.9 shows the block diagram of Sqrt CSLA based on CBL.

**D. Pipelining in MAC**

Pipelining scheme is applied to increase the operational speed of the design. In this work, a two stage pipelining is done as shown in Fig.10. The first stage includes the booth partial product generation and the CSA tree. The second stage consists of the final adder. Now the two stages can perform independently.

**III. RESULTS AND ANALYSIS**

The 16x16 bits MAC was simulated in Altera Modelsim 6.5b Edition and the functionality is verified. The simulation outputs of MAC with and without pipelining are shown in Fig.11 and Fig.12 respectively.

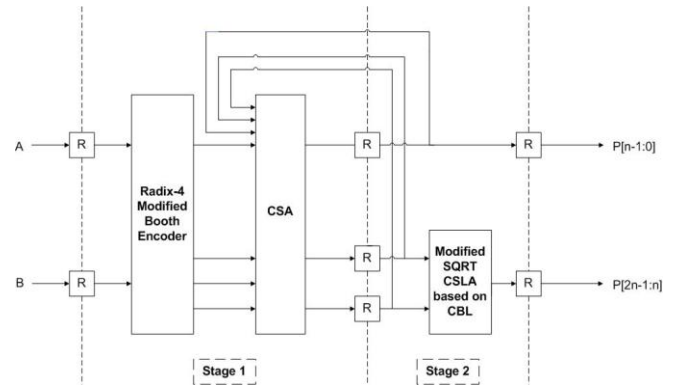


Figure 10. Pipelined architecture of Merged MAC

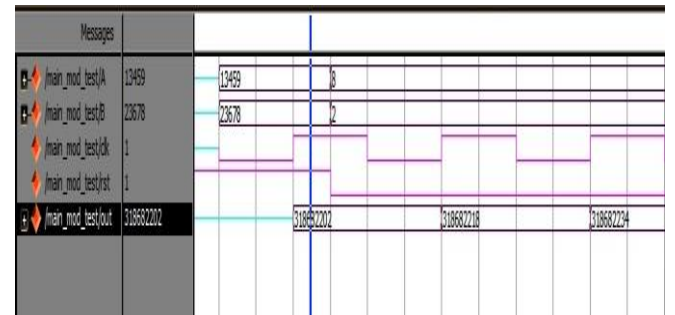


Figure 11. Simulation of Modified Merged MAC



Figure 12. Simulation of Pipelined Modified Merged MAC

It can be seen from Fig.12 that the first output is obtained after one clock cycle and the following outputs at consecutive clock cycles. Partial product generation and the S, C and Z are obtained in the first clock cycle and the final output is obtained in the second clock.

The synthesis of the adder designs and the MAC are done using Cadence RTL compiler TSMC 45nm standard cell library and analyzed for delay and power. The reports of adders and MAC are compared in Table.2 and Table.3 respectively. It can be from Table.2 that the Proposed 16 bits



Table 2. Delay, power report of CSLA

Design for 16 bits	Delay(ns)	Power(nW)
Regular SQRT CSLA	1.61	593976.5
Modified SQRT CSLA	1.87	581342.0
Proposed SQRT CSLA with CBL	1.54	391413.7

Table 3. Delay, power report of MAC

Design	Delay(ns)	Power(nW)
MAC Based on Radix-2 MBA with CLA [12]	6.73	1157298.07
MAC Based on Radix-4 MBA with 16 bit SQRT CSLA using CBL	5.4	1414300.062

SQRT CSLA on CBL shows reduced delay and power. Hence it is used to replace the final 4 bits CLA in work [12] and the results showed a 17.84% reduction in delay in the modified MAC.

#### IV. CONCLUSION

A new MAC architecture for high speed digital systems was designed and analyzed in this work. Since the accumulation process that has the largest delay is merged into the partial product reduction stage, the overall performance has been improved. In addition to it, the Radix-4 MBA and the final carry select adder improves the efficiency of the overall design. Further improvements over area and power can be done by modifying the existing high speed design.

#### REFERENCES

- [1] O. L. MacSorley, "High speed arithmetic in binary computers," Proc. IRE, vol. 49, pp. 67-91, Jan. 1961.
- [2] A. R. Omondi, Computer Arithmetic Systems. Englewood Cliffs, NJ: Prentice-Hall, 1994.
- [3] A. D. Booth, "A signed binary multiplication technique," Quart. J. Math., vol. IV, pp. 236-240, 1952.
- [4] C. S. Wallace, "A suggestion for a fast multiplier," IEEE Trans. Elec- tron Comput., vol. EC-13, no. 1, pp. 14-17, Feb. 1964.
- [5] A. R. Cooper, "Parallel architecture modified Booth multiplier," Proc.Inst. Electr. Eng. G, vol. 135, pp. 125- 128, 1988.
- [6] N. R. Shanbag and P. Juneja, "Parallel implementation of a 44-bit multiplier using modified Booth's algorithm," IEEE J. Solid-State Circuits, vol. 23, no. 4, pp. 1010- 1013, Aug. 1988.
- [7] G. Goto, T. Sato, M. Nakajima, and T. Sukemura, "A 54 54 regular structured tree multiplier," IEEE J. Solid-State Circuits, vol. 27, no. 9, pp. 1229-1236, Sep. 1992.
- [8] J. Fadavi-Ardekani, "M N Booth encoded multiplier generator using optimized Wallace trees," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 1, no. 2, pp. 120-125, Jun. 1993.
- [9] F. Elguibaly, "A fast parallel multiplier-accumulator using the modi- fied Booth algorithm," IEEE Trans. Circuits Syst., vol. 27, no. 9, pp. 902-908, Sep. 2000.
- [10] A. Fayed and M. Bayoumi, "A merged multiplier-accumulator for high speed signal processing applications," Proc. ICASSP, vol. 3, pp. 3212-3215, 2002.
- [11] P. Zicari, S. Perri, P. Corsonello, and G. Cocorullo, "An optimized adder accumulator for high speed MACs," Proc. ASICON 2005, vol. 2, pp. 757-760, 2005.
- [12] Young-Ho Seo, Dong-Wook Kim, ,, "A New VLSI Architecture of Parallel Multiplier-Accumulator Based on Radix-2Modified Booth Algorithm"" IEEE transactions on very large scale integration (VLSI) systems, vol. 18, no. 2, February2010.
- [13] Samiappa,Sakthikumar,S. Salivahanan, V. S. Kanchana Bhaaskaran, V. Kavnilavu, B. Brindha and CVinoth, "A Very Fast and Low Power Carry Select Adder Circuit",978-1-4244-8679-3/11.2011 IEEE.
- [14] P. Devi and A. Girdher, "Improved Carry Select Adder with Reduced Area and Low Power Consumption", International Journal of Computer Applications (0975- 8887) vol. 3 - No.4, June, 2010.