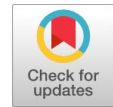# Design of Fast Efficient Radix-16 Sequential Multiplier

B. Gokul, M. Padmaja

Abstract: *Multiplication is an important function in computer arithmetic operations. The multiplication process will be done by the shift-and-add sequential multiplication procedure. Radix-16 sequential multiplier design generates the radix-16 partial products as two low (L) and high (H) components. In order to reduce cycle time, Brent-Kung adder and two radix 16 carry-save adders are used to generate radix-16 partial products. The proposed design of radix-16 sequential multiplier is efficient over previous designs and comparison depicts ADP and PDP of existing method are 11.22% and 8.45% than proposed method. However, the Excess area-Delay product and Excess-power-Delay product is also lowered. The design is carried out in Xilinx ISE 14.5 software and cadence tool for simulation and synthesis results. Fast efficient radix-16 sequential multiplier can be used in many digital signal processing applications.*
Index Terms: Radix-16 Sequential Multiplier, Radix-16 Carry-save Adder, Brent-Kung adder, Excess Area Delay product (EADP), Excess power Delay Product (EPDP).

I. INTRODUCTION

In digital era VLSI (very large scale integration) technology plays a prominent role and moreover IoT (internet of things) and digital signal processors focus mainly on system design. These technologies are extreme target on low area and power hardware applications. In general processors arithmetic unit utilizes hardware multipliers for low power/area properties. Before the utilization of sequential multipliers in digital processors and embedded systems parallel multipliers ruined these types of technologies. And the main intention to override these parallel multipliers with sequential multipliers is less area consumption.

The important factor regarding sequential multiplier is radix and it is taken from previously evaluated multiple value and the multiplicand is depends on and higher digit of multiplier. In most of computer arithmetic architectures utilize the sequential multipliers. In sequential multipliers PPG (partial product generation) and PPR (partial product reduction) are employed for generation of radix in pre-multiple of previous computation and multiplicand of higher radix [1].

In this paper we propose a new conventional design of a sequential multiplier radix-16 with n-bit and product of X×Y

in n-cycles. Sequential multiplier works like a normal multiplication in this we multiply n number of decimal number with n number of decimal numbers and vice-versa binary numbers also. In a sequential multiplier, the multiplication process is divided into sequential steps like in each step certain partial products will be generated, added to an accumulated partial sum and partial sum will be shifted to attain the accumulated sum with partial product of further steps. Sequential multipliers can take a minimum number of clock cycles to enhance the output product.

For redundant bit representation APP (accumulated partial product) are utilized and it gmajaives additional speed to PPR(partial product representation).mainly the PPR and carry save adders are used in this purpose for reduction in number of clock cycles in attained time. In parallel prefix adders we use two types of adders like Brent Kung and kogge-stone adders, Brent Kung adders are used for less number of fan-out and minimum amount of area is utilized. The wiring congestion is less and it leads to less delay without compromising the power performance in a adder. In Kogge -Stone adders are also utilizing minimum number of fan-outs. In this paper Brent Kung adders [3] are utilize and the reason to use those adder is less area and more power efficient. The logic depth of this adder is high that means it uses minimum number of fan-out characteristics.

II. SYSTEM DESIGN

This section describes the modified radix-16 sequential multiplier where the delay is reduced by using parallel computations using Brent Kung adder. In proposed project, first analyzing the existing method operation with its advantages and disadvantages and modifying the existing method to improve some major factors like delay, power etc. The detailed explanation is given in further sections:

A. **Existing Radix-16 Sequential Multiplier**

In existing methodology, a radix 16-bit sequential multiplier consists of elements like CPA (carry propagating adders), CSA (carry save adders), multiplexers and shift registers Fig 1 depicts existing Radix-16 Sequential Multiplier which consists of multiplicand (X) and multiplier (Y) each of n-bit size.

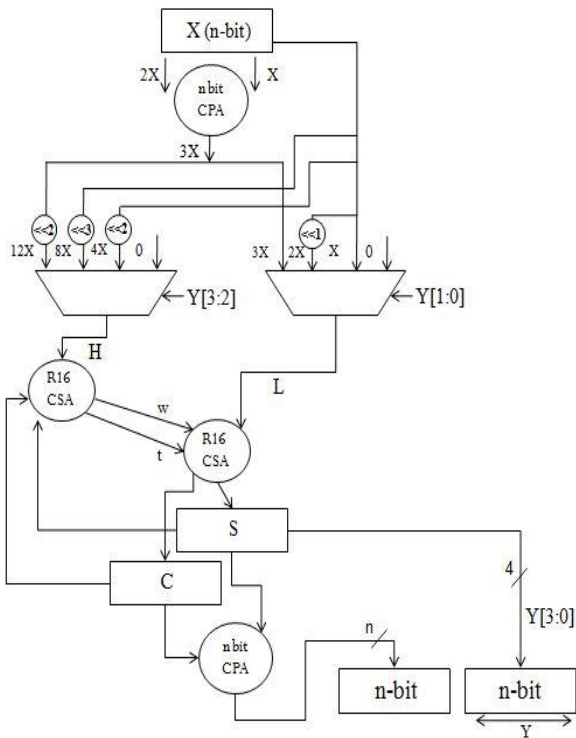# Design Of Fast Efficient Radix-16 Sequential Multiplier

Fig1. Existing Radix-16 Sequential Multiplier [1]

Firstly multiplicand (X) is given to one input of n-bit carry propagate adder and another input is X<<1 i.e., 2X and with third input equal to zero. From carry propagate adder, output W1 is produced whose value is equal to 3X. CPA will generate a partial product of each shifting as P= 4H + L, where H, L belongs to (0, 1, 2, 3) × X. Then in order to collect high and low partial products by shifting method, two 4:1 multiplexers are used. M1 for low partial products and M2 for high partial products.M1 inputs are {0,X,2X,3X} , select line is Y[1:0] and output is L.M2 inputs are {0,4X,8X,12X} , select line is Y[3:2] and output is H. Then the output of M2 i.e., H is given to radix-16 carry save adder with other two inputs as zero in order to produce output as sum (W) and carry (T) which in turn given to second radix-16 CSA with one more input as L.

The outputs of second radix-16 CSA is generated with sum and carry bits and shifted four times in order to get n-bit output value. Products of required n-bit operand addition will be processed in two consecutive levels of carry save adders and collects the bits and least significant bits which will be again shifted to the radix-16 CSA loop and gets iterated. Similarly signed-digit number systems are used to implement the sequential multipliers.

Fig 2 represents the partial products in terms of sum (S) and carry (C) where L and H are outputs from the multiplexers M1 and M2. Further bits from M2 are W and T which in turn returns sum and carry those can be given to the Y [3:2] × X multiplexer. The new accumulated 4 bits is added to the previous already produced partial product as shown in Fig 1.
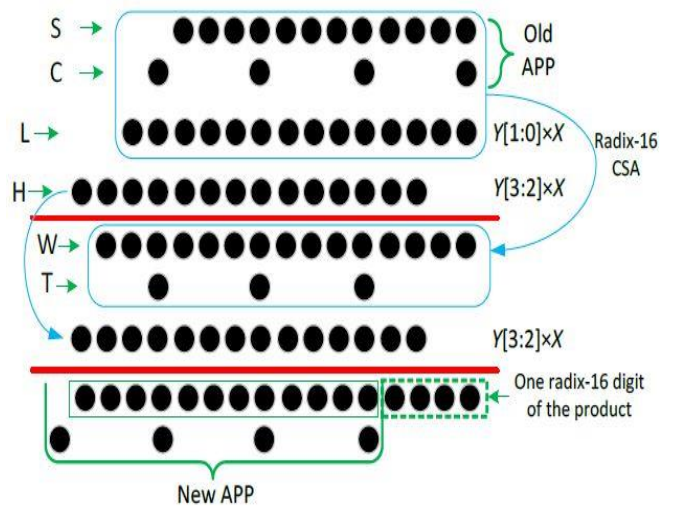

Fig2. Dot notations of partial products [1]

In order to improve design constraints of previous method, some major modifications are carried out in proposed work. In previous method carry propagating adder which was also capable of doing the parallel carry computation but the thing is, it works properly only for the lower bits. For higher bits, parallel prefix adder or tree adders are perfect to compute the n number of carry bits in parallel so CPA of previous method is replaced with the Brent Kung adder. By modifying, the performance is improved in terms of delay thus meeting one of the design constraints.

B. Proposed Radix-16 Sequential Multiplier

In the proposed sequential multiplier operation the architecture is same as existing with change in CPA with the Brent Kung adder (BKA). In Fig 3 depicts the proposed radix 16-bit sequential multiplier consists of BKA (Brent Kung adders), CSA (carry save adders), multiplexers and shift registers. Inputs of proposed Radix-16 sequential multiplier consist of multiplicand (X) and multiplier (Y) each of n bits. Firstly multiplicand (X) is given to one input of n-bit Brent Kung adder and another input is X<<1 i.e., 2X and third input equal to zero. From Brent Kung adder, output W1 is produced of value is equal to 3X.

Brent Kung adder (BKA) used is a 16 bit BKA, internally consisting of a gray cells (GC) and black cells (BC). In 16 bit BKA uses 14 BC and 11 GC and the advantage of using this adder is having a less wiring congestion. Here carry is calculated in parallel thus reducing the cycle time drastically. Then in order to collect high and low partial products by shifting method, two 4:1 multiplexers are used. M1 for low partial products and M2 for high partial products.M1 inputs are {0,X,2X,3X}, select line is Y[1:0] and output is L. M2 inputs are {0,4X,8X,12X} , select line is Y[3:2] and output is H.
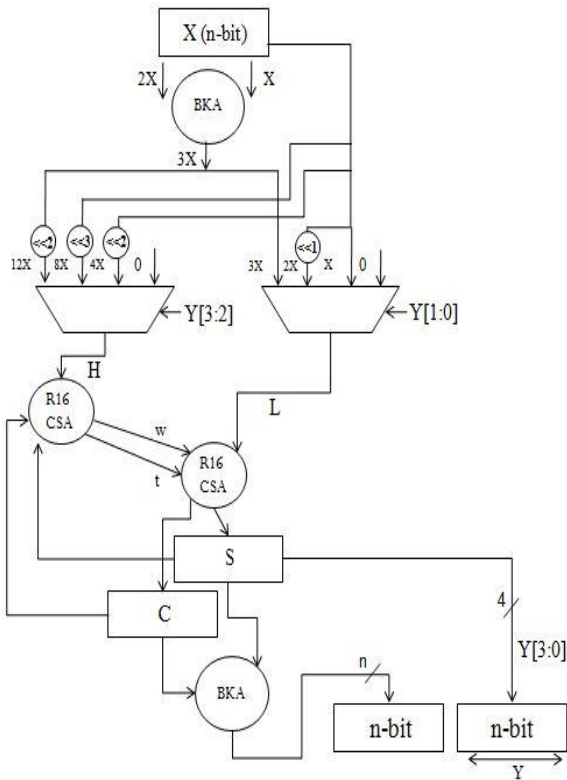
Fig3. Proposed radix-16sequential multipier

Then the output of M2 i.e., H is given to radix-16 carry save adder with other two inputs as zero in order to produce output as sum (W) and carry (T) which in turn given to second radix-16 CSA with one more input as L. The outputs of second radix-16 CSA is generated with sum and carry bits and shifted four times in order to get n-bit output value. Products of required n-bit operand addition will be processed in two consecutive levels of carry save adders and collects the bits and least significant bits which will be again shifted to the radix-16 CSA loop and gets iterated. Similarly signed-digit number systems are used to implement the sequential multipliers.

a) Radix-16 Carry Save Adder

In this radix-16 by using carry save adder the delay can be reduced further. In carry save adder, it refrain from directly passing on the carry information until the very last step and 2:1 multiplexer is used explained below in detail.

b) Ripple Carry Adders

Ripple carry adder is a simplest implementation of an 'n' bit adder. Which it includes of full adders connected in a series. The carry out (Cout) of the initial full adder stage is given as carry input (Cin) to the next full adder stage.
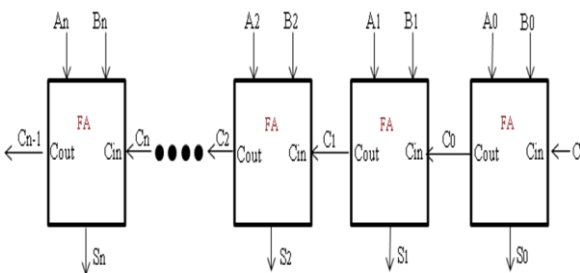

Fig4. Ripple Carry Adder

In the above figure we see the delay in each full adder is carry out (Cout) of the previous stage. Here the sequence of execution is obtained as below
C →S0 →C0 →S1→C1→S2→….. →Cn→Cn-1

In each stage output obtained as a ripple, Hence the name ripple carry adder. The ripple carry adder is simple to design as it implemented by using full adders. The major drawback is delay is obtaining at output, where each stage is dependent on the carry produced by previous stage. As it has to wait until the carries are generated along the way. Even it is having drawbacks it is a basic block to design many future adder designs. And 2:1 multiplexer is used for selecting the MSB and LSB bits of carries in Radix-16 CSA used.

c) Carry Look Ahead Adder

In Ripple carry adder mechanism, we found an important concern is the delay produced because of carry generated in each stage seen in above Fig 4. In this Carry look ahead adder (CLA) it reduces the delay which is simply speeds the operation in the circuit. In Carry look ahead adders logic connections are modified as required with the basic full adder logic stage. Whereas shown in Fig 5 it represents, internal logic block was consisting of an Ex-OR gates and AND gate. CLA calculates one or more carry bits before the sum and which it reduces wait time to calculate the result of the larger value bits. In this adder needs additional hardware but the speed of operation is independent of the number of bits.
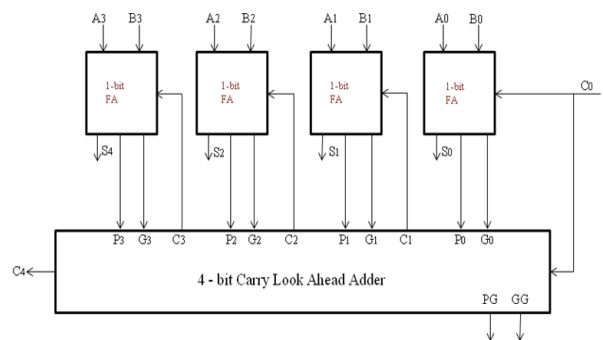

Fig5. Carry Look Ahead Adder

This adder particularly depends on two things

1. Where it calculate for each bit position, whether that position is going to propagate a carry from the right.
2. After these values calculated it combines and able to deduce quickly from each group bit, that group is going to propagate a carry that comes from right.

Reduces the propagation time and fastest addition logic is developed. Advantage of carry look ahead adder is it will speed up the addition operation or computation time. For lower bits this adder will work efficiently coming to the higher bits or larger bits look ahead adder logic will become more complex.

d) Brent Kung Adder[3]

The Brent Kung adder evaluates the prefixes for 2 bit groups. These prefixes are used to find the prefixes for 4 bit groups, which in turn are used to compute the prefixes for 8 bit groups. These prefixes evaluate carry out for every particular bit in stages. These carries are used with the group propagate of the other stage to compute the sum bit of that stage.

Brent Kung Tree will be using $2\log_2 N - 1$ stage shown in Fig 6. When designing a 16-bit adder the number of stages will be the fan out for each bit stage is limited to 2. This fan out is minimized and the loading on the other stages being reduced and buffers are omitted.
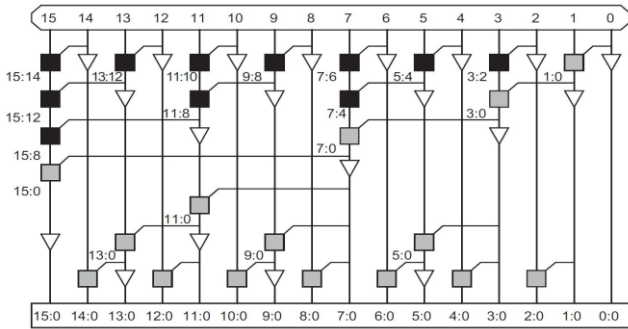

Fig6. 16-bit Brent Kung adder [3]

Considering an example of 16-bit radix-16 sequential multiplier–two inputs A (multiplicand) and B (multiplier).

Assume A = 15 = 0000000000001111

B = 3 = 0000000000000011

Step 1: The multiplicand A is first input to BKA, second input is A<<1 i.e., 0000000000011110 and third input is zero. The output of BKA is shifted bit as the 3X preliminary cycle and shifted the A bit three times which is directed to W1 = 0000000000101101.

Step 2: In order to collect high and low partial products by shifting method, two 4:1 multiplexers are used. M1 for low partial products and M2 for high partial products.M1 inputs are {0, X, 2X, 3X}, select line is B[1:0] and output is L. M2 inputs are {0, 4X, 8X, 12X}, select line is B[3:2] and output is H.

M2 Multiplexer inputs:

12 X = 10110100 (W1<<2)

8 X = 1111000 (A<<3)

4 X = 111100 (A<<2)

0 = 0

M2 Multiplexer inputs:

3 X = 101101(A + 2A)

2 X = 11110 (A<<1)

X = 1111(A)

0 = 0

Step 3: Select lines of multiplexers M1 is B[1:0] and M2 is B[3:2].The output of M2 named as High (H) and M1 output is Low (L). High (H) = 0000 and Low (L) = 101101 this output is given to two Carry save adder (CSA).

Step 4: First CSA receives input H, 0, 0 and output is W, T where W represent sum and T represent carry. Then second CSA receives inputs as L, W, T and generates sum(S) and carry(C).

Step 5: Sum, carry registers are given to the Brent Kung adder (BKA) and 0 as another input and it process the sum, carry in parallel. From the accumulated register, sum bits get shifted four times to produce data as 0000000000101101 and previously carry bits are accumulated with already produced partial products and sends the least significant bits get shifted and again new Y[3:0] is given to multiplexers and the loop iteration continuous.

Final output is 0000000000101101(45)

A (15) $\times$ B (3) = Output (45)

III. EXPERIMENTAL RESULTS

The experimental results of existing and proposed multiplier designs are conducted in two sections:
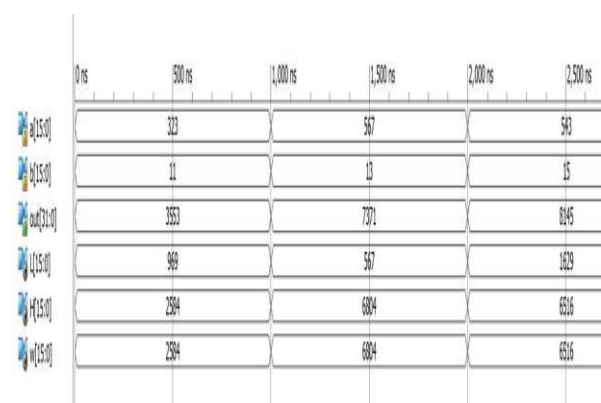
I. Simulation Results:

Fig7. Simulation for existing radix-16 sequential multiplier

Fig 7 depicts the Simulation result for existing radix-16 sequential multiplier. The values are multiplied with A (567) and B (13) which gives an output value of 7371(decimal) in waveform it was represented in binary values.

Fig 8 depicts the Simulation result for proposed radix-16 sequential multiplier. The values are multiplied with A (567) and B (13) which gives an output value of 7371(decimal) in waveform it was represented in binary values.

Inputs A, B given input in binary and output as out in binary

A= 0000001000110111(567)

B =0000000000001101(13)

Output: 00000000000000000111111101000 (7371)

By proposed method simulation report is same as existing shown in Fig 8 within an error rate $\pm 1\%$. Through the synthesis report speed of computation varies from the existing method.
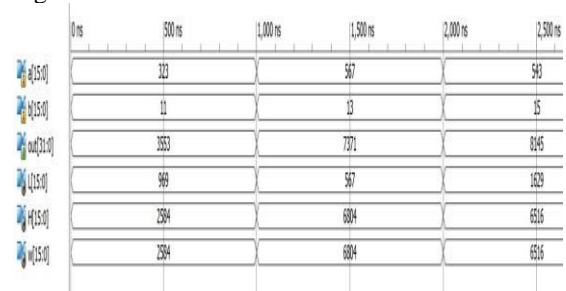

Fig8. Simulation for proposed radix-16 sequential multiplier

II. Synthesis Results:

Here it shows the synthesis result of a proposed radix-16 sequential multiplier in Verilog HDL in Cadence Genus 90nm technology and comparison table shown in table1. The table shows parameters delay, area and power. In proposed method it shows the improvement in Excess Area Delay Product (EADP) and Excess Power Delay Product (PDP). The Excess Area Delay Product (EADP) and Excess Power Delay Product (EPDP) it shows the relation of how much in excess with existing method than the proposed method.

Table1. Parameters comparison table:

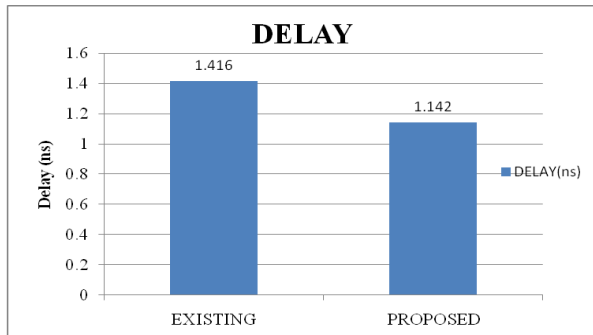|	Area (μm²)	Power (nW)	Delay (ns)	ADP	EADP (%)	PDP	EPDP (%)
Existing	1246.61	86835.14	1.416	1765.19	11.2286	12295.5	8.4588
Proposed	1389.6	9972.19	1.142	1587.08	--	113368.8	--

Fig9. Delay comparison of existing and proposed methods

Where the above result shows ADP of existing method is 11.22% is excess than the proposed method. The PDP of existing method is 8.45% excess than proposed method.

Fig 9 shows the delay comparison of existing and proposed methods .The above result shows the proposed radix-16 sequential multiplier has reduced delay compared with the existing method because of using the Brent Kung adder. Brent Kung adder has a less critical path and speeds up the execution of operation in circuit. So the fast efficient radix-16 sequential adder can be used for the digital signal processing and filter applications.

Table1 represents the comparison of existing and proposed architecture. From the results obtained, delay for proposed architecture has been reduced than that of the existing. So, it increases the speed of operation which implies the architecture is helpful for the digital signal processing applications.

μm² = micrometer, nW= nano watts, ns= nano seconds

IV. CONCLUSION

In existing work previously carry propagation adder having the similar functionality of processing the parallel execution of bits. It is well suited to lower bits when coming to the higher bits parallel prefix adders are proper. In this proposed architecture more concentrated to reduce the delay. Comparison based on figure.4.4 these results are helpful for speeding up the computation in the circuit.

On performing synthesis, the proposed method gives the reduced delay which helps for speed of operation, By using proposed multiplier a saving of 11.22% EADP and 8.45% EPDP is achieved than the proposed method. By using Brent Kung parallel adder, delay is reduced which helps in improvement for speed of operation with an error rate of $\pm 1\%$

V. FUTURE SCOPE

The proposed system is designed and simulated in 90nm technology library. It can be implemented on FPGAs.
- By changing the design process.
- By implementing the different parallel prefix adders less delay, area and power may achieve.
- By changing technology.

REFERENCES

1. Amanollahi, S., & Jaberipur, G. (2017). Fast Energy Efficient Radix-16 Sequential Multiplier. *IEEE Embedded Systems Letters, 9*, 73-76.
2. R. Brent and H. Kung, "A regular layout for parallel adders", IEEE Transaction on Computers, vol. C-31,n o.3,p p. 260-264,M arch 1982.
3. Gundi, N. D. (2015). Implementation of 32 bit Brent Kung Adder using complementary pass transistor logic (Doctoral dissertation, Oklahoma State University).
4. S.Amanollahi and G.Jaberipur, "Architecture-Level Design Space Exploration for Radix-16 Sequential Multipliers", The CSI Journal on Computer Science and Engineering, vol. 13, no. 1, 2015.
5. Baran, D., Aktan, M., & Oklobdzija, V. G. (2011, May). Multiplier structures for low power applications in deep-CMOS. In 2011 IEEE International Symposium of Circuits and Systems (ISCAS) (pp. 1061-1064). IEEE.
6. M.D.Ercegovac, and T. Lang, "Fast Multiplication without Carry-Propagate Addition" IEEE Transactions on Computer, vol. 39, no. 11, 1385-1390, 1990.
7. N.Honarmand,M.R.Javaheri, N.Sedaghati-Mokhtari and A.Afzali-Kusha, "Power Efficient Sequential Multiplication Using Pre-computation" Proceedings of IEEE International Symposium on Circuits and Systems, 2006, pp. 270 9-2712.2
8. M. Mottaghi- Dastjerdi, A. Afzali-Kusha, and M. Pedram, "BZ-FAD: A Low-Power Low-Area Multiplier based on Shift-and-Add Architecture," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 17, no. 2, pp. 302 -306, 2009
9. Abid, Z., El-Razouk, H., El-Dib, D.A. 'Low power multipliers based on new hybrid full adders', Microelectron. J., 2008, 39, pp. 1509–1515.
10. Antelo, E., Montuschi, P., and Nannarelli, A. Improved 64-bit Radix-16 Booth Multiplier Based on Partial Product Array Height Reduction. IEEE Transactions on Circuits and Systems I: Regular Papers, 64(2), 409–418.
11. Yezerla, S. K., and Rajendra Naik, B. Design and estimation of delay, power and area for Parallel prefix adders. 2014 Recent Advances in Engineering and Computational Sciences (RAECS).
12. Saxena, P. Design of low power and high speed Carry Select Adder using Brent Kung adder. 2015 International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA).
13. Lynch, T., and Swartzlander, E. A spanning tree carry look ahead adder. IEEE Transactions on Computers, 41(8), 931–939.

AUTHORS PROFILE

B Gokul from Vijayawada, Andhra Pradesh, India. He received his B.Tech degree in the year of 2015 from NRI Institute of technology, Agiripalli (A.P), India. Currently he is pursuing M.Tech in VLSI & ES from VR Siddhartha engineering college, Vijayawada (A.P), India. His research interests on digital design circuits

Dr M.Padmaja is working as professor in V R Siddhartha Engineering College, Vijayawada (A.P), India . She is having 25 years of teaching experience. She published several papers in National & International Journals & Conferences. She is the life member of MISTE, FIETE, MIE (I), MBMESI, and MSEMCEI.