

April 2015

A DESIGN OF AREA AND POWER EFFICIENT HIGH SPEED DATA PATH LOGIC SYSTEM

G. NARAYANA MURTHY

VLSI Design, Sir C.R.REDDY College of Engineering, Eluru, narayan.murthy@gmail.com

R. TRINATH

Dept. of E.C.E, Sir C.R.REDDY College of Engineering, Eluru, trinath@gmail.com

Follow this and additional works at: <https://www.interscience.in/ijeee>



Part of the [Power and Energy Commons](#)

Recommended Citation

MURTHY, G. NARAYANA and TRINATH, R. (2015) "A DESIGN OF AREA AND POWER EFFICIENT HIGH SPEED DATA PATH LOGIC SYSTEM," *International Journal of Electronics and Electrical Engineering*: Vol. 3 : Iss. 4 , Article 5.

Available at: <https://www.interscience.in/ijeee/vol3/iss4/5>

This Article is brought to you for free and open access by Interscience Research Network. It has been accepted for inclusion in International Journal of Electronics and Electrical Engineering by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

A DESIGN OF AREA AND POWER EFFICIENT HIGH SPEED DATA PATH LOGIC SYSTEM

G.NARAYANA MURTHY¹, R.TRINATH²,

¹M.Tech-Vlsi, Sir C.R.R Engineering College, Eluru

²M.Tech, Assitant Professor, Sir C.R.R Engineering College, Eluru.

Abstract- Carry Select Adder (CSLA) is one of the fastest adders use in many data-processing processors to perform fast arithmetic functions. From the structure of the CSLA, it is clear that there is scope for reducing the area and power consumption in the CSLA. This work uses a simple and efficient gate-level modification to significantly reduce the area and power of the CSLA. Based on this modification 8-, 16-, 32-, and 64-b square-root CSLA (SQRT CSLA) architecture have been developed and compared with the regular SQRT CSLA architecture. The proposed design has reduced area and power as compared with the regular SQRT CSLA with only a slight increase in the delay. This work evaluates the performance of the proposed designs in terms of delay, area, power, and their products by hand with logical effort and through custom design and layout in 0.18- m CMOS process technology. The results analysis shows that the proposed CSLA structure is better than the regular SQRT CSLA.

(Keywords- Application-specific integrated circuit (ASIC), area-efficient CSLA, low power...)

1. INTRODUCTION:

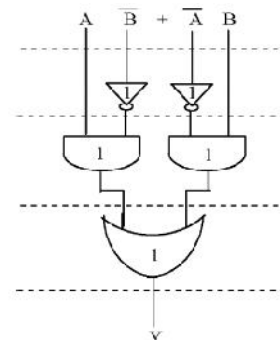
Design of area- and power-efficient high-speed data path logic systems are one of the most substantial areas of research in VLSI system design. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder. The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position.

The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum [1]. However, the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input $C_{in}=0$ and $C_{in}=1$, then the final sum and carry are selected by the multiplexers (mux).

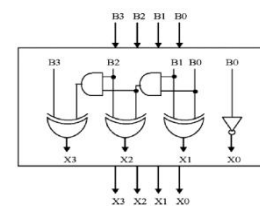
The basic idea of this work is to use Binary to Excess-1 Converter (BEC) instead of RCA with $C_{in}=0$ in the regular CSLA to achieve lower area and power consumption [2]–[4]. The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure. The details of the BEC logic are discussed in Section III.

This brief is structured as follows. Section II deals with the delay and area evaluation methodology of the basic adder blocks. Section III presents the detailed structure and the function of the BEC logic. The SQRT CSLA has been chosen for comparison with the proposed design as it has a more balanced delay, and requires lower power and area [5], [6]. The delay and area evaluation methodology of the regular

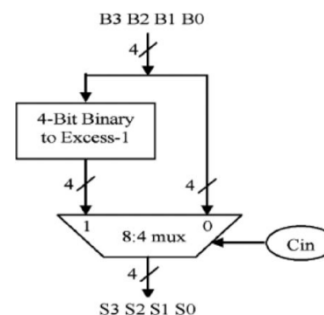
and modified SQRT CSLA are presented in Sections IV and V, respectively. The ASIC implementation details and results are analyzed in Section VI. Finally, the work is concluded in Section VII.



1.1.Delay and area evaluation of an XOR gate



1.2.4-b BEC



1.3.4-B BEC with 8:4 MUX

The carry-ripple adder is composed of many cascaded single-bit full-adders. The circuit architecture is simple and area-efficient. However, the computation speed is slow because each full-adder can only start operation till the previous carry-out signal is ready. In the carry select adder, N bits adder is divided into M parts. Each part of adder is composed two carry ripple adders with cin_0 and cin_1, respectively. Through the multiplexer, we can select the correct output according to the logic state of carry-in signal. The carry-select adder can compute faster because the current adder stage does not need to wait the previous stage's carry-out signal. The summation result is ready before the carry-in signal arrives; therefore, we can get the correct computation result by only waiting for one multiplexer delay in each single bit adder. In the carry select adder, the carry propagation delay can be reduced by M times as compared with the carry ripple adder. However, the duplicated adder in the carry select adder results in larger area and power consumption.

1.1 AREA-EFFICIENT CARRY SELECT ADDER

The carry ripple adder is constructed by cascading each single-bit full-adder [1]. In the carryripple adder, each full-adder starts its computation till previous carry-out signal is ready. Therefore, the critical path delay in a carry ripple adder is determined by its carry-out propagation path. For an N-bit full-adder as illustrated in Fig. 1, the critical path is N-bit carry propagation path in the full-adders. As the bit number N increases, the delay time of carry ripple adder will increase accordingly in a linear way. In order to improve the shortcoming of carry ripple adder to remove the linear dependency between computation delay time and input word length, carry select adder is presented [2]. The carry select adder divides the carry ripple adder into M parts, while each part consists of a duplicated (N/M)-bit carry ripple adder pair, as illustrated in Fig. 2 as M=16 and N=4. This duplicated carry ripple adder pair is to anticipate both possible carry input values, where one carry ripple adder is calculated as carry input value is logic "0" and another carry ripple adder is calculated as carry input value is logic "1". When the actual carry input is ready, either the result of carry "0" path or the result of carry "1" path is selected by the multiplexer according to its carry input value. An example of 5-bit carry select adder is illustrated in Fig. 3. To anticipate both possible carry input values in advance, the start of each M part carry ripple adder pair no longer need to wait for the coming of previous carry input. As a result, each M part carry ripple adder pair in the carry select adder can compute parallel

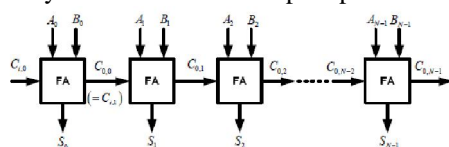


Fig. 1.4 The N-bit carry ripple adder constructed by N set single bit full-adder

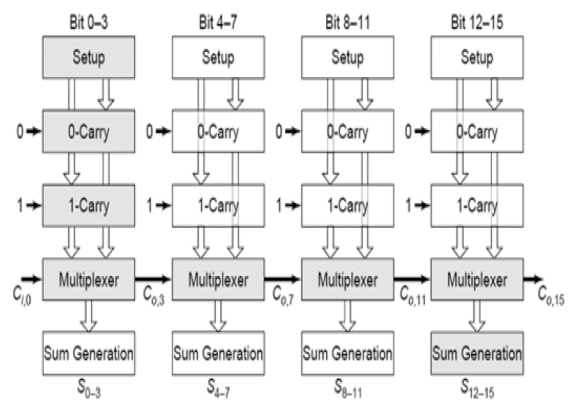


Fig1.5 The 16-bit carry select adder is divided the carry ripple adder into 4 parts, while each part consists of a duplicated 4-bit carry ripple adder pair.

In this way, the critical path of N bit adder can be greatly reduced. In the conventional N-bit carry ripple adder design, the critical path is N-bit carry

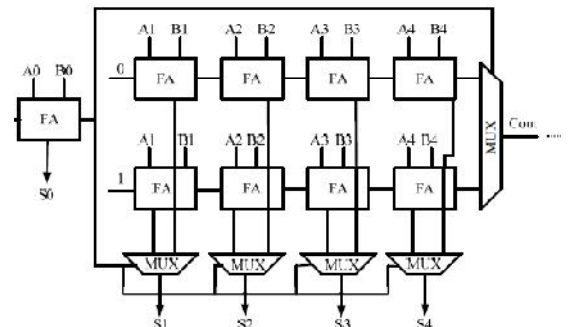


Fig:1.6 5-bit carry select adder [1], [2].

propagation path plus one summation generation stage. Alternatively, the critical path is (N/M)-bit carry propagation path plus M stage multiplexer with one summation generation stage in the N-bit carry select adder. Since M is much smaller than N and delay in the multiplexer is smaller than that in the full adder, the computation delay in the carry select adder is much shorter than that in the carry ripple adder. However, implementing the adder with duplicated carry generation circuit costs almost twice hardware and twice power consumption as compared with the carry ripple adder. Therefore, in this paper, we proposed an area-efficient carry select adder by sharing the common Boolean logic term to remove the duplicated adder cells in the conventional carry select adder. In this way, we can save many transistor counts and achieve a lower PDP.

2. DELAY AND AREA EVALUATION METHODOLOGY OF THE BASIC ADDER BLOCKS

The AND, OR, and Inverter (AOI) implementation of an XOR gate is shown in Fig. 1. The gates between the dotted lines are performing the operations in parallel and the numeric representation of each gate indicates the delay

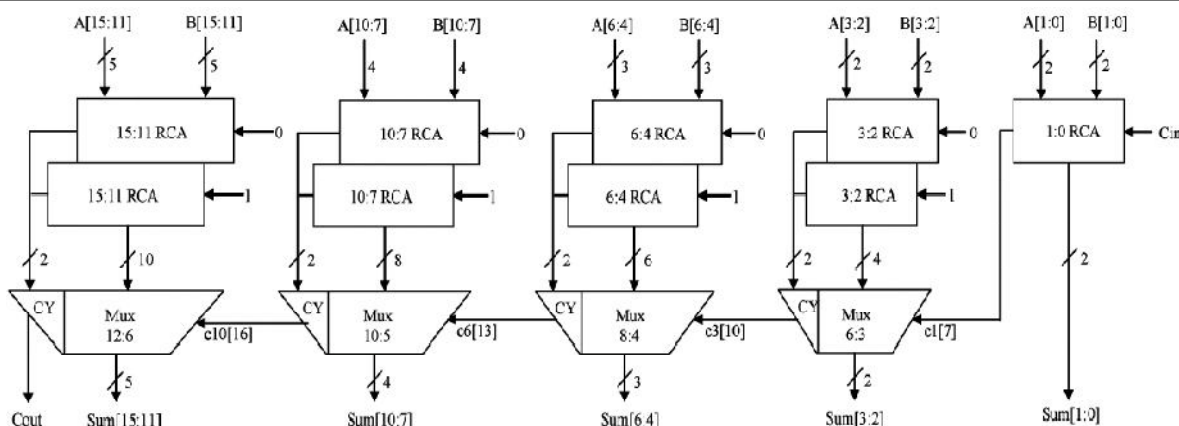


Fig 2.1 Regular 16-b Sqrt CSLA

B[3:0]	X[3:0]
0000	0001
0001	0010
:	:
:	:
1110	1111
1111	0000

Table 2.1 Delay and area Count of the Basic Blocks of csLA. contributed by that gate. The delay and area evaluation methodology considers all gates to be made up of AND, OR, and Inverter, each having delay equal to 1 unit and area equal to 1 unit. We then add up the number of gates in the longest path of a logic block that contributes to the maximum delay. The area evaluation is done by counting the total number of AOI gates required for each logic block. Based on this approach, the CSLA adder blocks of 2:1 mux, Half Adder (HA), and FA are evaluated and listed in Table I

Table 2.2 Function Table of the BEC

Adder blocks	Delay	Area
XOR	3	5
2:1 Mux	3	4
Half adder	3	6
Full adder	6	13

3. BEC

As stated above the main idea of this work is to use BEC instead of the RCA with $C_{in}=1$ in order to reduce the area and power consumption of the regular CSLA. To replace the n-bit RCA, an n+1bit BEC is required. A structure and the function table of a 4-b BEC are shown in Fig. 2 and Table II, respectively. Fig. 3 illustrates how the basic function of the CSLA is obtained by using the 4-bit BEC together with the mux. One input of the 8:4 mux gets as it input (B3, B2, B1, and B0) and another input of the mux is the

BEC output. This produces the two possible partial results in parallel and the mux is used to select either the BEC output or the direct inputs according to the control signal C_{in} . The importance of the BEC logic stems from the large silicon area reduction when the CSLA with large number of bits are designed. The Boolean expressions of the 4-bit BEC is listed as (note the functional symbols NOT, AND, XOR).

$$\begin{aligned}
 X0 &= \overline{B}0; \\
 X1 &= B0 \wedge B1; \\
 X2 &= B2 \wedge (B0 \& B1); \\
 X3 &= B3 \wedge (B0 \& B1 \& B2)
 \end{aligned}$$

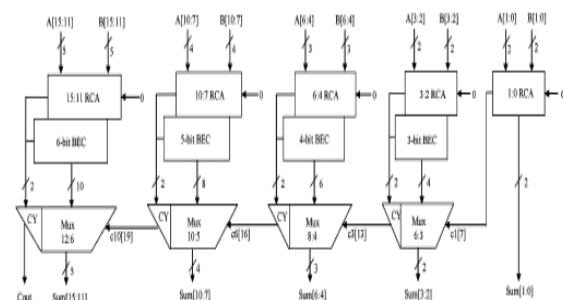


Fig. 4.1. Modified 16-b Sqrt CSLA. The parallel RCA with $c_{in}=1$ is replaced with BEC.

4. DELAY AND AREA EVALUATION METHODOLOGY OF REGULAR 16-B Sqrt CSLA

The structure of the 16-b regular Sqrt CSLA is shown in Fig. 4. It has five groups of different size RCA. The delay and area evaluation of each group are shown in Fig. 5, in which the numerals within [] specify the delay values, e.g., sum2 requires 10 gate delays. The steps leading to the evaluation are as follows.

1) The group2 [see Fig. 5(a)] has two sets of 2-b RCA. Based on the consideration of delay values of Table I, the arrival time of selection input $c1[t=7]$ of 6:3 mux is earlier than $s3[t=8]$ and later than $s2[t=6]$. Thus, $sum[t=11]$ is summation of $S3$ and $mux[t=3]$ and $sum[t=10]$ is summation of $c1$ and mux.

2) Except for group2, the arrival time of mux selection input is always greater than the arrival time of data outputs from the RCA's. Thus, the delay of group3 to group5 is determined, respectively as follows:

$$\{c6, \text{sum}[6:4]\} = c3[t=10] + \text{mux}.$$

$$\{c10, \text{sum}[10:7]\} = c6[t=13] + \text{mux}.$$

$$\{\text{cout}, \text{sum}[15:11]\} = c10[t=16] + \text{mux}.$$

TABLE 4.1

Delay and Area Count of Regular Sqrt CSLA groups

Group	Delay	Area
Group2	11	57
Group3	13	87
Group4	16	117
Group5	19	147

3) The one set of 2-b RCA in group2 has 2 FA for cin=0 and the other set has 1 FA and 1 HA for cin=0. Based on the area count of Table I, the total number of gate counts in group2 is determined as follows:

$$\text{Gate count} = 57(\text{FA} + \text{HA} + \text{Mux})$$

$$\text{FA} = 39(3 * 13)$$

$$\text{HA} = 6(1 * 6)$$

$$\text{Mux} = 12(3 * 4).$$

4) Similarly, the estimated maximum delay and area of the other groups in the regular Sqrt CSLA are evaluated and listed in Table III.

5. DELAY AND AREA EVALUATION METHODOLOGY OF MODIFIED 16-B Sqrt CSLA

The structure of the proposed 16-b Sqrt CSLA using BEC for RCA with Cin=1 to optimize the area and power is shown in Fig. 6. We again split the structure into five groups. The delay and area estimation of each group are shown in Fig. 7. The steps leading to the evaluation are given here.

1) The group2 [see Fig. 7(a)] has one 2-b RCA which has 1 FA and 1 HA for Cin=0. Instead of another 2-b RCA with Cin=1 a 3-b BEC is used which adds one to the output from 2-b RCA. Based on the consideration of delay values of Table I, the arrival time of selection input c1 [time(t)=7] of 6:3 mux is earlier than the s3 [t=9] and c3 [t=10] and later than the s2 [t=4]. Thus, the sum3 and final c3 (output from mux) are depending on s3 and mux and partial c3 (input to mux) and mux, respectively. The sum2 depends on c1 and mux.

2) For the remaining group's the arrival time of mux selection input is always greater than the arrival time of data inputs from the BEC's. Thus, the delay of the

remaining groups depends on the arrival time of mux selection input and the mux delay.

3) The area count of group2 is determined as follows:

$$\text{Gate count} = 43(\text{FA} + \text{HA} + \text{Mux} + \text{BEC})$$

$$\text{FA} = 13(1 * 13)$$

$$\text{HA} = 6(6 * 1)$$

$$\text{AND} = 1$$

$$\text{NOT} = 1$$

$$\text{XOR} = 10(2 * 5)$$

$$\text{Mux} = 12(3 * 4).$$

4) Similarly, the estimated maximum delay and area of the other groups of the modified Sqrt CSLA are evaluated and listed in Table IV.

Comparing Tables III and IV, it is clear that the proposed modified Sqrt CSLA saves 113 gate areas than the regular Sqrt CSLA, with only 11 increases in gate delays. To further evaluate the performance, we have resorted to ASIC implementation and simulation.

6. ASIC IMPLEMENTATION RESULTS

The design proposed in this paper has been developed using VHDL and synthesized in Cadence RTL compiler using typical libraries of TSMC 0.18 um technology. The synthesized Verilog netlist and their respective design constraints file (SDC) are imported to Cadence SoC Encounter and are used to generate automated layout from standard cells and placement and routing [7]. Parasitic extraction is performed using Encounter's Native RC extraction tool and the extracted

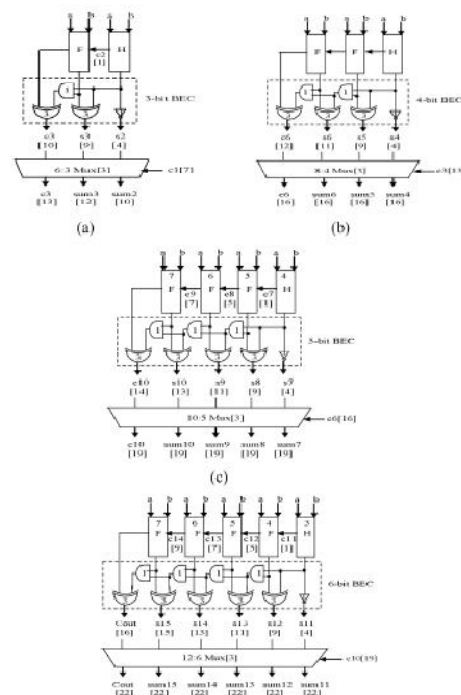


Fig. 5.1 Delay and area evaluation of modified Sqrt CSLA: (a) group2, (b) group3, (c) group4, and (d) group5. H is a Half Adder.

Table 5.1
Delay and Area Count of Modified SQRT CSLA

Group	Delay	Area
Group2	13	43
Group3	16	61
Group4	19	84
Group5	22	107

parasitic RC (SPEF format) is back annotated to Common Timing Engine in Encounter platform for static timing analysis. For each word size of the adder, the same value changed dump (VCD) file is generated for all possible input conditions and imported the same to Cadence Encounter Power Analysis to perform the power simulations. The similar design flow is followed for both the regular and modified SQRT CSLA.

Table V exhibits the simulation results of both the CSLA structures in terms of delay, area and power. The area indicates the total cell area of the design and the total power is sum of the leakage power, internal power and switching power. The percentage reduction in the cell area, total power, power-delay product and the area-delay product as function of the bit size are shown in Fig. 8(a). Also plotted is the percentage

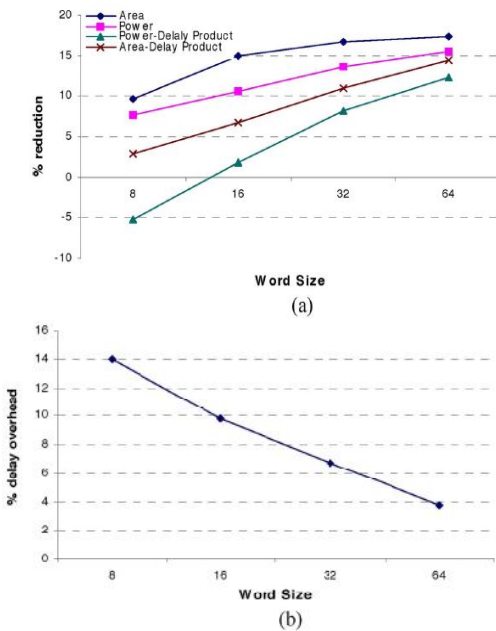


Fig. 6.1 (a) Percentage reduction in the cell area, total power, power-delay product, and area-delay product. (b) Percentage of delay overhead.

delay overhead in Fig. 8(b). It is clear that the area of the 8-, 16-, 32-, and 64-b proposed SQRT CSLA is reduced by 9.7%, 15%, 16.7%, and 17.4%, respectively. The total power consumed shows a similar trend of increasing reduction in power consumption 7.6%, 10.56%, 13.63%, and 15.46 % with the bit size. Interestingly, the delay overhead also exhibits a similarly decreasing trend with bit

size. The delay overhead for the 8, 16, and 32-b is 14%, 9.8%, and 6.7% respectively, whereas for the 64-b it reduces to only 3.76%. The power-delay product of the proposed 8-b is higher than that of the regular SQRT CSLA by 5.2% and the area-delay product is lower by 2.9%. However, the power-delay product of the proposed 16-b SQRT CSLA reduces by 1.76% and for the 32-b and 64-b by as much as 8.18%, and 12.28% respectively. Similarly the area-delay product of the proposed design for 16-, 32-, and 64-b is also reduced by 6.7%, 11%, and 14.4% respectively.

Table 6.2
Comparison of the Regular and Modified SQRT CSLA

Word size	Adder	Delay (ns)	Area (um ²)	Leakage power	Switching Power	Total Power	Power-Delay product (10 ⁻¹⁵)	Area-Delay product (10 ⁻²¹)
8-bit	Regular CSLA	1.719	991	0.007	101.9	23.0	35.0	17.035
	Modified CSLA	1.958	895	0.006	94.2	18.8	36.8	17.524
16-bit	Regular CSLA	2.775	2272	0.017	263.7	52.7	14.63	63.048
	Modified CSLA	3.048	1929	0.013	235.9	47.0	14.38	58.796
32-bit	Regular CSLA	5.137	4783	0.036	563.6	112.9	57.90	24.570
	Modified CSLA	5.482	3985	0.049	483.3	107.5	53.53	22.2

	A M o d i f i e d C S L A			02 7		9 6 9 9	16 9	21 84 5. 7
64- bit	Re g u l a r C S L A M o d i f i e d C S L A	9.1 74	991 6	0. 07 5	12 12. 4	2 4 2 5. 0	22 24 6. 9 19 77	90 96 9. 3
		9.5 19	818 3	0. 05 7	10 25. 0	2 0 0 5 0. 1	19 51 4. 9 9	89 3. 9

Total power=leakage power+Internal power+Switching power

7. CONCLUSION

A simple approach is proposed in this paper to reduce the area and power of SQRT CSLA architecture. The reduced number of gates of this work offers the great advantage in the reduction of area and also the total power. The compared results show that the modified SQRT CSLA has a slightly larger delay (only 3.76%), but the area and power of the 64-b modified SQRT CSLA are significantly reduced by 17.4% and 15.4% respectively. The power-delay product and also the area-delay product of the proposed design show a decrease for 16-, 32-, and 64-b sizes which indicates the success of the method and not a mere tradeoff of delay for power and area. The modified CSLA architecture is therefore, low area, low power, simple and efficient for VLSI hardware implementation. It would be interesting to test the design of the modified 128-b SQRT CSLA.

REFERENCES

[1]. B. Ramkumar, H.M. Kittur, and P. M. Kannan, "ASIC implementation of modified faster carry save adder," *Eur. J. Sci. Res.*, vol. 42, no. 1, pp. 53–58, 2010

[2]. Cadence, "Encounter user guide," Version 6.2.4, March 2008.

[3]. Y. He, C. H. Chang, and J. Gu, "An area efficient 64-bit square root carry-select adder for lowpower applications," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2005, vol. 4, pp. 4082–4085.

[4]. Y. Kim and L.-S. Kim, "64-bit carry-select adder with reduced area," *Electron. Lett.*, vol. 37, no. 10, pp. 614–615, May 2001.

[5]. T. Y. Ceiang and M. J. Hsiao, "Carry-select adder using single ripple carry adder," *Electron. Lett.*, vol. 34, no. 22, pp. 2101–2103, Oct. 1998.

[6]. O. J. Bedrij, "Carry-select adder," *IRE Trans. Electron. Comput.*, pp. 340–344, 1962.



R. Trinadh received the M.Tech. Degree in embedded System from the gudlavalleru engineering college affiliated to JNTU Kakinada in 2011. He is currently working as an Assistant Professor with the Department of Electronics & communication Engineering, SIR.C.R.Reddy college of Engineering, Eluru. His research interests in embedded real time operating system in relation with power consume and defilation



G. N. Murthy received the B.Tech Degree in Electronics and Communication Engineering from JNTU KKD in 2010. Currently pursuing M.Tech in Sir C R Reddy College of Engineering Eluru. His areas of interest are Low Power VLSI Design.

