International Journal for Modern Trends in Science and Technology
Volume: 02, Issue No: 11, November 2016
http://www.ijmtst.com
ISSN: 2455-3778

Implementation of Low Power and Area-Efficient Carry Select Adder

T. Bhavana[1] | V. Rajesh[2]

[1]PG Scholar, Department of ECE, St.Mary's Women's Engineering College, Budampadu, Guntur Dt, A.P, India.
[2]Assistant Professor, Department of ECE, St.Mary's Women's Engineering College, Budampadu, Guntur Dt, A.P, India.

To Cite this Article
T. Bhavana, V. Rajesh, Implementation of Low Power and Area-Efficient Carry Select Adder, *International Journal for Modern Trends in Science and Technology*, Vol. 02, Issue 11, 2016, pp. 20-24.

ABSTRACT

Carry Select Adder (CSLA) is one of the fastest adders used 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-,-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*

I. 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 CLSA is used in many computational systems to alleviate the problem of carry progpagation 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 and 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 converted (BEC) instead of RCA with 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 bit Full Adder (FA) structure. This brief is structured as follows. This paper deals with the delay and area evaluation methodology of the basic adder blocks. And also 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.

II. Logic Formulation

The CSLA has two units: 1) the sum and carry generator unit (SCG) and 2) the sum and carry selection unit [9]. The SCG unit consumes most of the logic resources of CSLA and significantly contributes to the critical path. Different logic designs have been suggested for efficient implementation of the SCG unit. We made a study of the logic designs suggested for the SCG unit of conventional and BEC-based CSLAs of [6] by suitable logic expressions. The main objective of this study is to identify redundant logic operations and data dependence. Accordingly, we remove all redundant logic operations and sequence logic operations based on their data dependence.

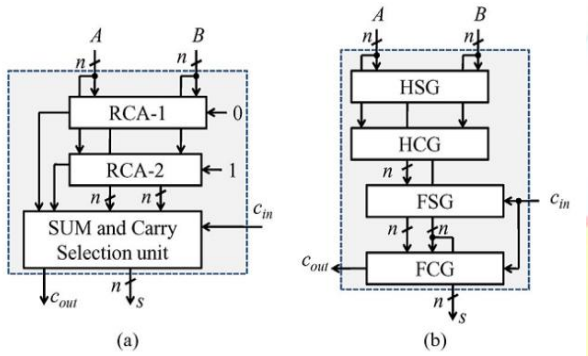

Fig. 1. (a) Conventional CSLA; n is the input operand bit-width. (b) The logic operations of the RCA

A. Logic Expressions of the SCG Unit of the

Conventional CSLA As shown in Fig.1(a), the SCG unit of the conventional CSLA [3] is composed of two n-bit RCAs, where n is the adder bit-width. The logic operation of the n-bit RCA is performed in four stages: 1) half-sum generation (HSG); 2) half-carry generation (HCG); 3) full-sum generation (FSG); and 4) full carry generation (FCG). Suppose two n-bit operands are added in the conventional CSLA, then RCA-1 and RCA-2 generate n-bit sum (s0 and s1) and output-carry (c0 out and c1 out) corresponding to input-carry (cin = 0 and cin = 1), respectively. Logic expressions of RCA-1 and RCA-2 of the SCG unit of the n-bit CSLA are given as

$$
\begin{aligned}
&s_0^0(i) = A(i) \oplus B(i) \quad c_0^0(i) = A(i) \cdot B(i)\\
&s_1^0(i) = s_0^0(i) \oplus c_1^0(i-1)\\
&c_1^0(i) = c_0^0(i) + s_0^0(i) \cdot c_1^0(i-1) \quad c_{\text{out}}^0 = c_1^0(n-1)\\
&s_0^1(i) = A(i) \oplus B(i) \quad c_0^1(i) = A(i) \cdot B(i)\\
&s_1^1(i) = s_0^1(i) \oplus c_1^1(i-1)\\
&c_1^1(i) = c_0^1(i) + s_0^1(i) \cdot c_1^1(i-1) \quad c_{\text{out}}^1 = c_1^1(n-1)
\end{aligned}
\tag{1}
$$

B. Logic Expression of the SCG Unit of the BEC-Based CSLA

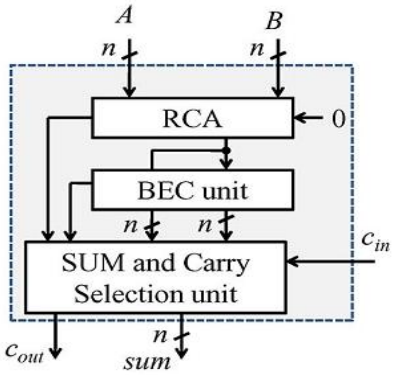

Fig.2. Structure of the BEC-based CSLA; n is the input operand bit-width.

As shown in Fig.2, the RCA calculates n-bit sum S_1^0 and C_{out}^0 corresponding to cin = 0. The BEC unit receives S_1^0 and C_{out}^0 from the RCA and generates (n + 1)-bit excess-1 code. The most significant bit (MSB) of BEC represents c1 out, in which n least significant bits (LSBs) represent S_1^1. The logic expressions

$$
\begin{aligned}
&s_1^1(0) = \overline{s_1^0(0)} \quad c_1^1(0) = s_1^0(0)\\
&s_1^1(i) = s_1^0(i) \oplus c_1^1(i-1)\\
&c_1^1(i) = s_1^0(i) \cdot c_1^1(i-1)\\
&c_{\text{out}}^1 = c_1^0(n-1) \oplus c_1^1(n-1)
\end{aligned}
\tag{2}
$$

The selected carry word is added with the half-sum (s0) to generate the final-sum (s). Using this method, one can have three design advantages:

1) Calculation of s0 1 is avoided in the SCG unit;
2) The n-bit select unit is required instead of the (n + 1) bit; and
3) Small output-carry delay. All these features result in an area–delay and energy-efficient design for the CSLA.

We have removed all the redundant logic operations of 2 and rearranged logic expressions of 2 based on their dependence. The proposed logic formulation for the CSLA is given as

$$
\begin{aligned}
&s_0(i) = A(i) \oplus B(i) \quad c_0(i) = A(i) \cdot B(i)\\
&c_1^0(i) = c_1^0(i-1) \cdot s_0(i) + c_0(i) \quad \text{for } (c_1^0(0) = 0)\\
&c_1^1(i) = c_1^1(i-1) \cdot s_0(i) + c_0(i) \quad \text{for } (c_1^1(0) = 1)\\
&c(i) = c_1^0(i) \quad \text{if } (c_{\text{in}} = 0)\\
&c(i) = c_1^1(i) \quad \text{if } (c_{\text{in}} = 1)\\
&c_{\text{out}} = c(n-1)\\
&s(0) = s_0(0) \oplus c_{\text{in}} \quad s(i) = s_0(i) \oplus c(i-1).
\end{aligned}
\tag{3}
$$

III. Proposed Adder Design

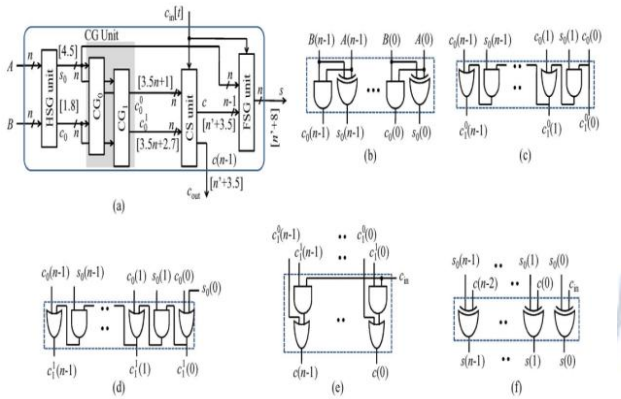

Fig. 3. (a) Proposed CS adder design, where n is the input operand bit-width, and [*] represents delay (in the unit of inverter delay), n = max (t, 3.5n + 2.7). (b) Gate-level design of the HSG. (c) Gate-level optimized design of (CG0) for input-carry = 0. (d) Gate-level optimized design of (CG1) for input-carry = 1. (e) Gate-level design of the CS unit. (f) Gate-level design of the final-sum generation (FSG) unit.

The proposed CSLA is based on the logic formulation given in 4.3, and its structure is shown in Fig.3(a). It consists of one HSG unit, one FSG unit, one CG unit, and one CS unit. The CG unit is composed of two CGs (CG0 and CG1) corresponding to input-carry '0' and '1'. The HSG receives two n-bit operands (A and B) and generate half-sum word s0 and half-carry word c0 of width n bits each. Both CG0 and CG1 receive s0 and c0 from the HSG unit and generate two n-bit full-carry words c0 1 and c11 corresponding to input-carry '0' and '1', respectively.

The logic diagram of the HSG unit is shown in Fig3(b). The logic circuits of CG0 and CG1 are optimized to take advantage of the fixed input-carry bits. The optimized designs of CG0 and CG1 are shown in Fig. 3(c) and (d), respectively.

The CS unit selects one final carry word from the two carry words available at its input line using the control signal cin. It selects C_1^0 when cin = 0; otherwise, it selects C_1^1. The CS unit can be implemented using an n-bit 2-to-1 MUX. However, we find from the truth table of the CS unit that carry words c0 1 and c11 follow a specific bit pattern. If C_1^0 (i) = '1', then C_1^1 (i) = 1, irrespective of s0(i) and c0(i), for $0 \le i \le n - 1$. This feature is used for logic optimization of the CS unit. The optimized design of the CS unit is shown in Fig. 3(e), which is composed of n AND–OR gates. The final carry word c is obtained from the CS unit. The MSB of c is sent to output as cout, and (n − 1) LSBs are XORed with (n − 1) MSBs of half-sum (s0) in the FSG [shown in Fig. 3(f)] to obtain (n − 1) MSBs of final-sum (s). The LSB of s0 is XORed with cin to obtain the LSB of s.

IV. Performance Comparison

A. Area–Delay Estimation Method

TABLE I
AREA AND DELAY OF AND, OR, AND NOT GATES GIVEN IN THE SAED 90-nm STANDARD CELL LIBRARY DATASHEET

ADDER TYPE	NO.OF.SLICES	NO.OF. LUTS	CRTICAL PATH DELAY	QUSENT POWER(MW)	DYNAMIC POWER (MW)	POWER DELAY PRODUCT TERMS(MW)
32BIT RIPPLE CARRY ADDER	46	63	61.588	203	241	14.842
PROPOSED RIPPLE CARRY ADDER	71	140	47.668	203	254	12.107

We have considered all the gates to be made of 2-input AND, 2-input OR, and inverter (AOI). A 2-input XOR is composed of 2 AND, 1 OR, and 2 NOT gates. The area and delay of the 2-input AND, 2-input OR, and NOT gates (shown in Table I) are taken from the Synopsys Armenia Educational Department (SAED) 90-nm standard cell library datasheet for theoretical estimation. The area and delay of a design are calculated using the following relations:

$$A = a \cdot N_a + r \cdot N_o + i \cdot N_i$$

$$T = n_a \cdot T_a + n_o \cdot T_o + n_i \cdot T_i \quad (4)$$

Where (Na, No, Ni) and (na, no, ni), respectively, represent the (AND, OR, NOT) gate counts of the total design and its critical path. (a, r, i) and (Ta, To, Ti), respectively, represent the area and delay of one (AND, OR, NOT) gate. We have calculated the (AOI) gate counts of each design for area and delay estimation the area and delay of each design are calculated from the AOI gate counts (Na, No, Ni), (na, no, ni), and the cell details of Table I. The path of the proposed CSLA, the delay of each intermediate and output signals of the proposed n-bit CSLA design of Fig. 3 is shown in the square bracket against each signal. We can find from Table II that the proposed n-bit single-stage CSLA adder involves 6n less number of AOI gates than the CSLA of [6] and takes 2.7 and 6.6 units less delay to calculate final-sum and output-carry. Compared with the CBL-based CSLA of [7], the proposed CSLA design involves n more AOI gates, and it takes (n − 4.7) unit less delay to calculate the output-carry.

V. Extension Concept of Multistage CSLA (SQRT-CSLA)

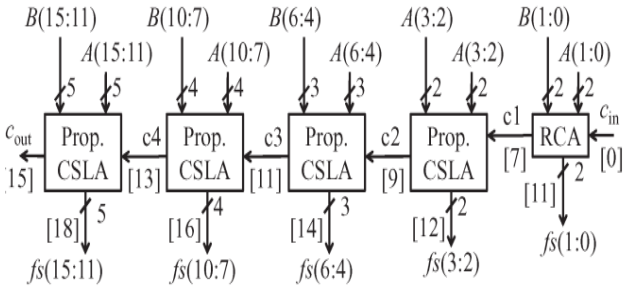

Fig. 4. Proposed SQRT-CSLA for n = 16. All intermediate and output signals are labeled with delay

The multipath carry propagation feature of the CSLA is fully exploited in the SQRT-CSLA [5], which is composed of a chain of CSLAs. CSLAs of increasing size are used in the SQRT-CSLA to extract the maximum concurrence in the carry propagation path. Using the SQRT-CSLA design, large-size adders are implemented with significantly less delay than a single-stage CSLA of same size. However, carry propagation delay between the CSLA stages of SQRT-CSLA is critical for the overall adder delay.

Due to early generation of output-carry with multipath carry propagation feature, the proposed CSLA design is more favourable than the existing CSLA designs for area–delay efficient implementation of SQRT-CSLA. A 16-bit SQRT-CSLA design using the proposed CSLA is shown in Fig. 4, where the 2-bit RCA, 2-bit CSLA, 3-bit CSLA, 4-bit CSLA, and 5-bit CSLA are used. We have considered the cascaded configuration of (2-bit RCA and 2-, 3-, 4-, 6-, 7-, and 8-bit CSLAs) and (2-bit RCA and 2-, 3-, 4-, 6-, 7-, 8-, 9-, 11-, and 12-bit CSLAs), respectively, for the 32-bit SQRTCSLA and the 64-bit SQRT-CSLA to optimize adder delay. To demonstrate the advantage of the proposed CSLA design in SQRT-CSLA, we have estimated the area and delay of SQRTCSLA using the proposed CSLA design and the BEC-based CSLA of [6] and the CBL-based CSLA of [7] for bit-widths 16, 32.

VI. Results

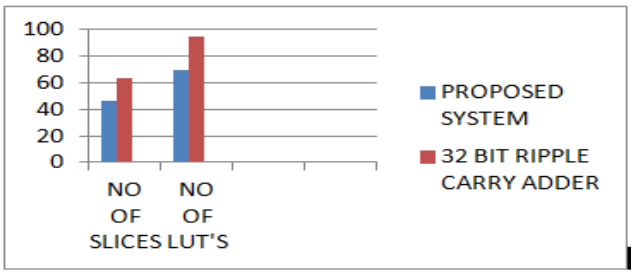

Fig 5 Area Comparison

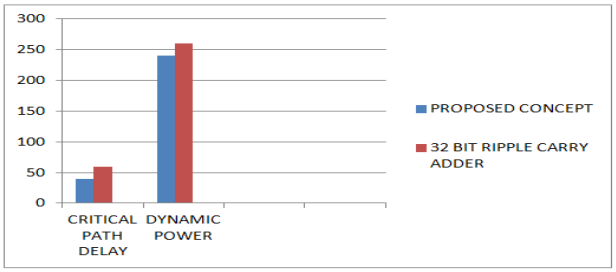

Figure delay comparison

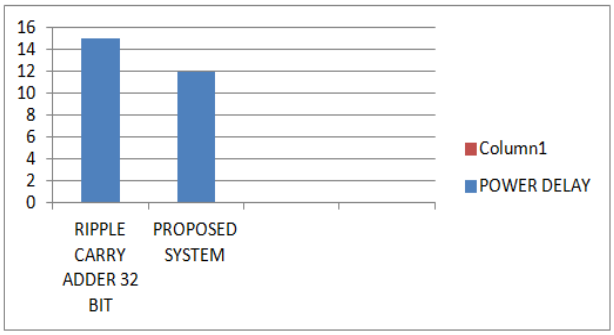

Fig 7 comparison methods

PROPOSED RESULTS

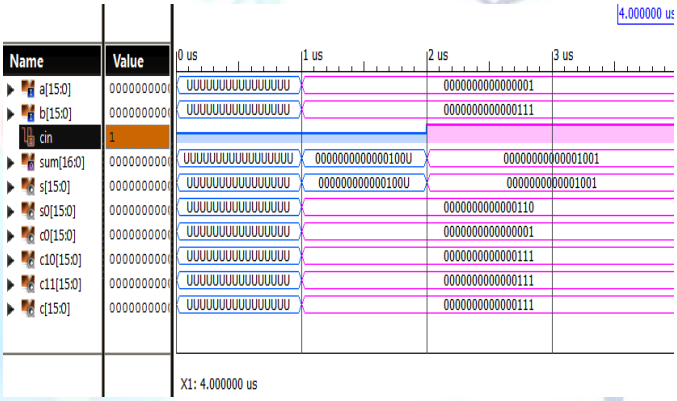

Figure 8 16 bit of proposed concept

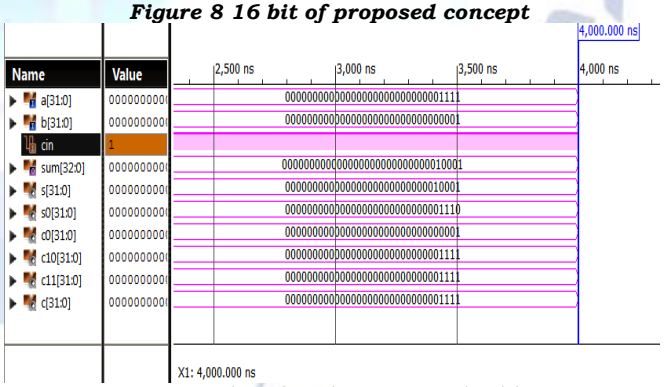

Fig 9 Results of 32 bit proposed adders

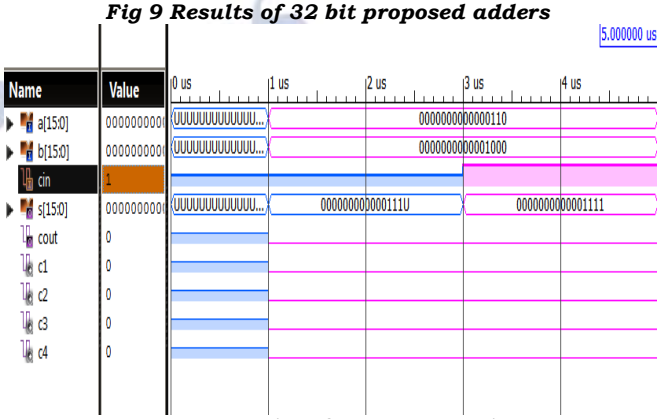

Fig 10 Results of extension 16 bit

VII. Conclusion

A simple approach is 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, but the area and power of the 32-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-b sizes which indicates the success of the method and not a mere trade off 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 SQRT CSLA.

References

[1] K. K. Parhi, VLSI Digital Signal Processing. New York, NY, USA: Wiley, 1998.
[2] A. P. Chandrakasan, N. Verma, and D. C. Daly, "Ultralow-power electronics for biomedical applications," Annu. Rev. Biomed. Eng., vol. 10, pp. 247– 274, Aug. 2008.
[3] O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Comput., vol. EC-11, no. 3, pp. 340–344, Jun. 1962.
[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] Y. He, C. H. Chang, and J. Gu, "An area-efficient 64-bit square root carryselect adder for low power application," in Proc. IEEE Int. Symp. Circuits Syst., 2005, vol. 4, pp. 4082–4085.
[6] B. Ramkumar and H. M. Kittur, "Low-power and area-efficient carry-select adder," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 2, pp. 371–375, Feb. 2012.
[7] I.-C. Wey, C.-C. Ho, Y.-S. Lin, and C. C. Peng, "An area-efficient carry select adder design by sharing the common Boolean logic term," in Proc. IMECS, 2012, pp. 1–4.

Thalupuri Bhavana Rao
Regd No: 14ND1D5715
PG Student Scholar
Father Name: Venkata Narayana Rao
Department of ECE (VLSI)
St.Mary's Women's Engineering College, Budampadu, Guntur district, A.P, India.
t.bhanavana07@gmail.com
+91 8142012134

V.Rajesh
Asst.professor,
Department of Electronics & Communication Engineering,
St.Mary's Women's Engineering College, Budampadu, Guntur district, A.P, India.
rajesh.v435@gmail.com
+91 9666959466