ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Improved Carry Select Adder Design Using Dynamic Designing

Guntupalli Praveena
MTech Student
Department of ECE
AnuBose Institute Of Technology(ABIT)
Paloncha, Khammam, India

T. Suma Latha
Assistant Professor
Department of ECE
AnuBose Institute Of Technology(ABIT)
Paloncha, Khammam, India

ABSTRACT: ***In this brief, the logic operations involved in conventional carry select adder (CSLA) and binary to excess-1 converter (BEC)-based CSLA are analyzed to study the data dependence and to identify redundant logic operations. We have eliminated all the redundant logic operations present in the conventional CSLA and proposed a new logic formulation for CSLA. In the proposed scheme, the carry select (CS) operation is scheduled before the calculation of final-sum, which is different from the conventional approach. Bit patterns of two anticipating carry words (corresponding to cin = 0 and 1) and fixed cin bits are used for logic optimization of CS and generation units. An efficient CSLA design is obtained using optimized logic units. The proposed CSLA design involves significantly less area and delay than the recently proposed BEC-based CSLA. Due to the small carry-output delay, the proposed CSLA design is a good candidate for square-root (SQRT) CSLA. A theoretical estimate shows that the proposed SQRT-CSLA involves nearly 35% less area–delay–product (ADP) than the BEC-based SQRT-CSLA, which is best among the existing SQRT-CSLA designs, on average, for different bit-widths. The application-specified integrated circuit (ASIC) synthesis result shows that the BEC-based SQRT-CSLA design involves 48% more ADP and consumes 50% more energy than the proposed SQRT-CSLA, on average, for different bit-widths.***

Index Terms*—*Adder, arithmetic unit, low-power design

INTRODUCTION:

LOW-POWER, area-efficient, and high-performance VLSI systems are increasingly used in portable and mobile devices, multistandard wireless receivers, and biomedical instrumentation [1], [2]. An adder is the main component of an arithmetic unit. A complex digital signal processing (DSP) system involves several adders. An efficient adder design essentially improves the performance of a complex DSP system. A ripple carry adder (RCA) uses a simple design, but carry propagation delay (CPD) is the main concern in this adder. Carry look-ahead and carry select (CS) methods have been suggested to reduce the CPD of adders.

A conventional carry select adder (CSLA) is an RCA–RCA configuration that generates a pair of sum words and outputcarry bits corresponding the anticipated input-carry (cin = 0 and 1) and selects one out of each pair for final-sum and final-output-carry [3]. A conventional CSLA has less CPD than an RCA, but the design is not attractive since it uses a dual RCA. Few attempts have been made to avoid dual use of RCA in CSLA design. Kim and Kim [4] used one RCA and one add-one circuit instead of two RCAs, where the add-one circuit is implemented using a multiplexer (MUX). He et al. [5] proposed a square-root (SQRT)-CSLA to implement large bit-width adders with less delay. In a SQRT CSLA, CSLAs with increasing size are connected in a cascading structure. The main objective of SQRT-CSLA design is to provide a parallel path for carry propagation that helps to reduce the overall adder delay.

Existing System:

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.

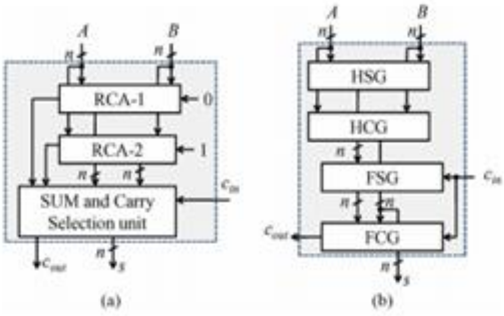

Fig. 1. (a) Conventional CSLA; n is the input operand bit-width. (b) The logic operations of the RCA is shown in split form, where HSG, HCG, FSG, and FCG represent half-sum generation, half-carry generation, full-sum generation, and full-carry generation, respectively.

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) fullcarry 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

$$s_0^0(i) = A(i) \oplus B(i) \quad c_0^0(i) = A(i) \cdot B(i) \qquad (1a)$$

$$s_1^0(i) = s_0^0(i) \oplus c_1^0(i-1) \qquad (1b)$$

$$c_1^0(i) = c_0^0(i) + s_0^0(i) \cdot c_1^0(i-1) \quad c_{\mathrm{out}}^0 = c_1^0(n-1) \qquad (1c)$$

$$s_0^1(i) = A(i) \oplus B(i) \quad c_0^1(i) = A(i) \cdot B(i) \qquad (2a)$$

$$s_1^1(i) = s_0^1(i) \oplus c_1^1(i-1) \qquad (2b)$$

$$c_1^1(i) = c_0^1(i) + s_0^1(i) \cdot c_1^1(i-1) \quad c_{\mathrm{out}}^1 = c_1^1(n-1) \qquad (2c)$$

vhere $c_1^0(-1) = 0$, $c_1^1(-1) = 1$, and $0 \leq i \leq n-1$.

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

As shown in Fig. 2, the RCA calculates n-bit sum s0 1 and c0 out corresponding to cin = 0. The BEC unit receives s0 1 and c0 out 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 s1 1. The logic expressions

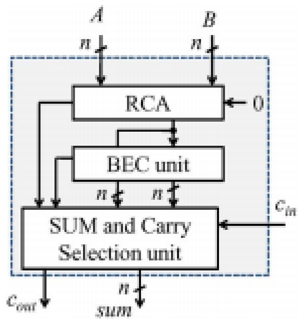

Fig. 2. Structure of the BEC-based CSLA;

n is the input operand bit-width. of the RCA are the same as those given in (1a)–(1c). The logic expressions of the BEC unit of the n-bit BEC-based CSLA are

$$s_1^1(0) = \overline{s_1^0}(0) \quad c_1^1(0) = s_1^0(0) \qquad (3a)$$

$$s_1^1(i) = s_1^0(i) \oplus c_1^1(i-1) \qquad (3b)$$

$$c_1^1(i) = s_1^0(i) \cdot c_1^1(i-1) \qquad (3c)$$

$$c_{\mathrm{out}}^1 = c_1^0(n-1) \oplus c_1^1(n-1) \qquad (3d)$$

Proposed System:

The proposed CSLA is based on the logic formulation given in (4a)–(4g), 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 c1 1 corresponding to input-carry ‘0’ and ‘1’, respectively. The logic diagram of the HSG unit is shown in Fig. 3(b). The logic circuits of CG0 and CG1 are optimized to take advantage of the fixed input-carry bits. The optimized

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

designs of CG0 and CG1 are shown in Fig. 3(c) and (d), respectively.

PERFORMANCE COMPARISON

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:

	AND-gate	OR-gate	NOT-gate
Area (um^2)	7.37	7.37	6.45
Delay (ps)	180	170	100

$$A = a \cdot N_a + r \cdot N_o + i \cdot N_i \qquad (5a)$$

$$T = n_a \cdot T_a + n_o \cdot T_o + n_i \cdot T_i \qquad (5b)$$

Single-Stage CSLA

The general expression to calculate the AOI gate counts of the n-bit proposed CSLA and the BEC-based CSLA of [6] and CBL-based CSLA of [7] and [8] are given in Table II of singlestage design. We have calculated the AOI gate counts on the critical path of the proposed n-bit CSLA and CSLAs of [6]–[8] and used those AOI gate counts in (5b) to find an expression for delay of final-sum and output-carry in the unit of Ti (NOTgate delay). The delay of the n-bit single-stage CSLA is shown in Table II for comparison. For further analysis of the critical 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.

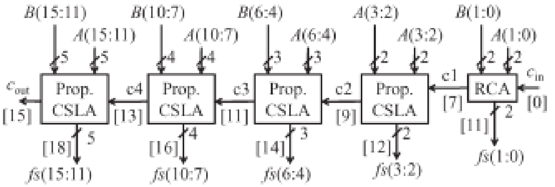

Fig. 4. Proposed SQRT-CSLA for $n = 16$. All intermediate and output signals are labeled with delay (shown in square brackets).

Multistage CSLA (SQRT-CSLA)

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.

ASIC Synthesis Results

We have coded the SQRT-CSLA in VHDL using the proposed CSLA design and the existing CSLA designs of [6] and [7] for bit-widths 16, 32, and 64. All the designs are synthesized in the Synopsys Design Compiler (DC) using the SAED 90-nm CMOS library. The netlist file obtained from the DC are processed in the IC Compiler (ICC). After placement and route, the area, data-arrival time (DAT), and power reported by the ICC are listed in Table V for comparison. The synthesis result of Table V confirms the theoretical estimates given in Table IV. As shown in Table V, the proposed SQRT-CSLA involves significantly less area and less delay and consumes less power than the existing designs.

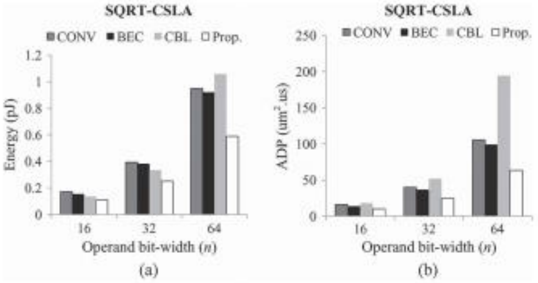

Fig. 5. (a) Comparison of energy consumption. (b) Comparison of ADP.

CONCLUSION

We have analyzed the logic operations involved in the conventional and BEC-based CSLAs to study the data dependence and to identify redundant logic operations. We have eliminated all the redundant logic operations of the conventional CSLA and proposed a new logic formulation for the CSLA. In the proposed scheme, the CS operation is scheduled before the calculation of final-sum, which is different from the conventional approach. Carry words corresponding to input-carry '0' and '1' generated by the CSLA based on the proposed scheme follow a specific bit pattern, which is used for logic optimization of the CS unit. Fixed input bits of the CG unit are also used for logic optimization. Based on this, an optimized design for CS and CG units are obtained. Using these optimized logic units, an efficient design is obtained for the CSLA. The proposed CSLA design involves significantly less area and delay than the recently proposed BEC-based CSLA. Due to the small carryoutput delay, the proposed CSLA design is a good candidate for the SQRT adder. The ASIC synthesis result shows that the existing BEC-based SQRT-CSLA design involves 48% more ADP and consumes 50% more energy than the proposed SQRTCSLA, on average, for different bit-widths.

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.

[8] S. Manju and V. Sornagopal, "An efficient SQRT architecture of carry select adder design by common Boolean logic," in Proc. VLSI ICEVENT, 2013, pp. 1–5.

[9] B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs, 2nd ed. New York, NY, USA: Oxford Univ. Press, 2010.