FPGA Design of Low Power Efficient Carry-Select Adder in MIMO-OFDM Systems

S.SivaSubramanian
M.E (VlSI),ECE Department,
K.Ramakrishnan College of Technology,
Tiruchirappalli, India.
e-mail : siva.ece09@gmail.com.

S.Divya
Asst. Professor , ECE Department,
K.Ramakrishnan College of Technology,
Tiruchirappalli, India.
e-mail: eceap18@gmail.com.

S.SyedAkbar
Head of the Department, ECE,
K.Ramakrishnan College of Technology,
Tiruchirappalli, India.
e-mail :akbarkrct@gmail.com.

Abstract—Logic operations involved in usual carry select adder (CSLA) and CSLA based on BEC are analyzed to study the data dependence and also identified the redundant logic operations involved in it. We have removed all the redundant logic operations involved in the usual CSLA and described a new logic derivation for CSLA. In the proposed scheme, the carry select (CS) operation is scheduled before the calculation of *final-sum*, which is entirely different from the usual approach. Bit module of two anticipating carry words (corresponding to *c*in = 0 and 1) and fixed *c*in bits are used for logic optimization of CS and generation units. An efficient CSLA design is obtained by using optimization of logic units. Results obtained from proposed CSLA design involves significantly less area and delay than the recently proposed CSLA based on BEC. Small carry delay occurred at the output, the proposed CSLA design is a good candidate for square-root (SQRT) CSLA.

Keywords—CSLA, BEC, low-power.

I. INTRODUCTION

LOW-POWER, area-efficient, and high-performance VLSI systems are increasingly used in transportable devices, various standard 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. 60% of total power is consumed by adders in conventional circuits. 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 major distress in this adder. Carry look-ahead and carry select (CS) methods have been recommended to reduce the CPD of adders. A usual carry select adder (CSLA) is an RCA–RCA configuration that generates a pair of *sum* words and *output-carry* bits related to the anticipated input-carry (*c*in = 0 and 1) and selects one out of each pair for *final-sum* and *Final-output-carry* [3]. A usual CSLA has less CPD than an RCA, but the plan is not attractive because 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 offer a parallel path for carry propagation that helps to reduce the overall adder delay. Ramkumar and Kittur [6] suggested a CSLA based on BEC. The CSLA based on BEC requires less logic resources than the usual CSLA, but it has higher delay. A CSLA based on common Boolean logic (CBL) is also proposed in [7] and [8]. The CBL-based CSLA of [7] involves significantly less logic resource than the usual CSLA but it has longer CPD, which is almost equal to that of the RCA. To overcome this problem, a SQRT-CSLA based on CBL was proposed in [8]. However, the CBL-based SQRTCSLA design of [8] requires more logic resource and delay than the BEC-based SQRT-CSLA of [6]. We observe that logic optimization largely depends on availability of redundant operations in the formulation, whereas adder delay mainly depends on data dependence. In the existing designs, logic is optimized without giving any consideration to the data dependence. In this brief, we made an analysis on logic operations involved in usual CSLA and CSLA based on BEC to study the data dependence and to identify redundant logic operations. Based on this analysis, we have proposed logic formulation for the CSLA. The main contribution in this brief is logic formulation based on data dependence and optimized carry generator (CG) and CS design. Based on the proposed logic equation, we have derived an efficient logic design for CSLA. Due to optimized logic units, the proposed CSLA involves significantly less ADP than the existing CSLAs.

II. LOGIC FORMULATION

The CSLA consist of two units: 1) the *sum* and *carry* generator unit (SCG) and 2) the *sum* and *carry* selection unit. The SCG unit consumes most of the logic resources of CSLA and a lot contribute to the critical path. Different logic design has been suggested for efficient implementation of the SCG unit. We made a study of the logic designs suggested for the SCG unit of usual CSLA and CSLA based on BEC of [6] by suitable logic expressions. The main objective of this study is to identify the redundant logic operations and data dependence. Eliminated all redundant logic operations and sequence logic operations based on their data dependence.

SCG based CSLA Architecture

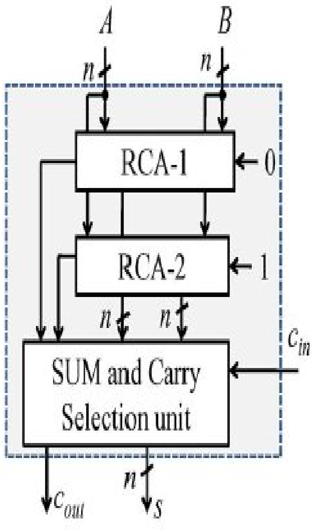

Fig.1.Usual CSLA; n is the input operand bit-width.

SCG unit of the usual CSLA 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).Logic expressions of the RCA-1 and RCA-2 of the SCG unit of the n-bitCSLAaregivenas

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

$$s_1^0(i) = s_0^0(i) \oplus c_1^0(i-1) \tag{1b}$$

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

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

$$s_1^1(i) = s_0^1(i) \oplus c_1^1(i-1) \tag{2b}$$

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

As shown in above (1a)-(1c) and (2a)-(2c), these redundant logic operations can be removed to have optimized design for RCA-2, in which the HSG and HCG of RCA-1 is shared to construct RCA-2.

CSLA based on BEC Architecture

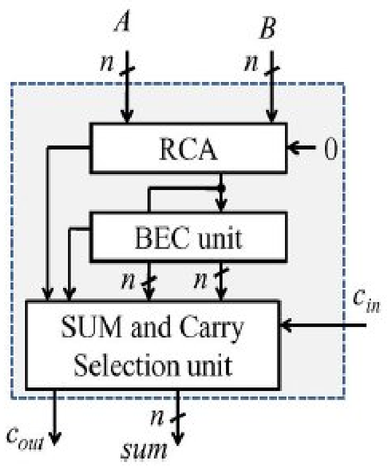

Fig.2.structure of CSLA based on BEC; n is the input operand bit-width.

BEC circuit will be added instead of RCA-2 in the CSLA. BEC unit receives input from RCA-1 and generate (n+1) bit excess code, MSB of BEC represent c_{out} and LSB represents s_1^1.CSLA based on BEC offers the best delay-area-power efficiency among the existing CSLAs, The logic expressions of BEC based CSLA are given as

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

$$s_1^1(i) = s_1^0(i) \oplus c_1^1(i-1) \tag{3b}$$

$$c_1^1(i) = s_1^0(i) \cdot c_1^1(i-1) \tag{3c}$$

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

We can find from (1a)-(1c) and (3a)-(3d), in the case of the BEC-based CSLA, c_1^1 depends on s_1^0, otherwise has no dependence on s_1^0 in the case of conventional CSLA. The BEC method increases the data dependences in the CSLA.

Proposed Adder Design

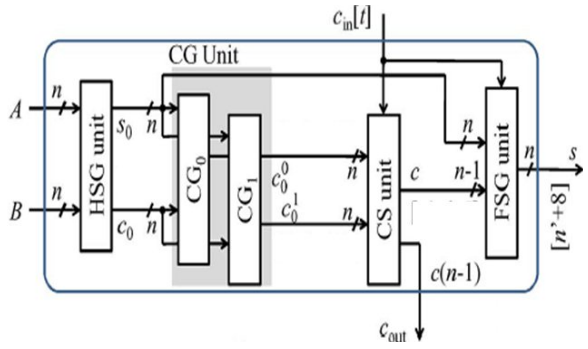

Fig.3. Proposed CS adder design.

We have removed all the redundant operations of (1a)-(1c) and (2a)-(2c) and rearranged the logic expressions of (1a)-(1c) and (2a)-(2c) based on their dependence. Proposed CSLA design is based on their logic formulation are given as

$$s_0(i) = A(i) \oplus B(i) \quad c_0(i) = A(i) \cdot B(i) \tag{4a}$$
$$c_1^0(i) = c_1^0(i-1) \cdot s_0(i) + c_0(i) \quad \text{for } \left(c_1^0(0) = 0\right) \tag{4b}$$
$$c_1^1(i) = c_1^1(i-1) \cdot s_0(i) + c_0(i) \quad \text{for } \left(c_1^1(0) = 1\right) \tag{4c}$$
$$c(i) = c_1^0(i) \quad \text{if } (c_{in} = 0) \tag{4d}$$
$$c(i) = c_1^1(i) \quad \text{if } (c_{in} = 1) \tag{4e}$$
$$c_{out} = c(n-1) \tag{4f}$$
$$s(0) = s_0(0) \oplus c_{in} \quad s(i) = s_0(i) \oplus c(i-1). \tag{4g}$$

Proposed CSLA consist of one FSG unit, one CG unit and one CS unit. The CG unit composed of two CGs (CG_0 and CG_1) corresponding to input carry '0' and '1'. The HSG receives two n-bit operands (A and B) and generate half-sum word s_0 and half-carry word c0 of width n bits each. Both CG_0 and CG_1 receive s_0 and c_0 from the HSG unit and generate two n-bit full-carry words $c_0^1$and c_1^1corresponding to input-carry '0' and '1', respectively.CG0 and CG1 are optimized to take advantage of the fixed input-carry bits. The CS unit selects one final carry word from the two carry words available at its input line using the control signal *c*in. It selects $c_0^1$when *c*in = 0; otherwise, it selects c_1^1.The CS unit can be implemented using an *n*-bit 2-to-l MUX. If $c_0^1(i)$ = '1', then $c_1^1(i)$ = 1,irrespective of $s_0(i)$ and $c(i)$, for $0 \le i \le n - 1$. This feature is used for logic optimization of the CS unit. Using this method ,one can have three design advantages; Calculation of s_1^0 is avoided in SCG unit ,n-bit select unit is required instead of (n+1)bit; small output carry delay, these feature results in efficient design of CSLA. The optimized design of the CS unit 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 *c*out, and $(n - 1)$ LSBs are XORed with $(n - 1)$ MSBs of *half-sum* (s_0) in the FSG. To obtain $(n - 1)$ MSBs of *final-sum*(*s*).The LSB of s_0 is XORed with *c*in to obtain the LSB of *s*.

III. SIMULATION RESULTS

Below figure shows the simulation waveform of proposed CS adder design

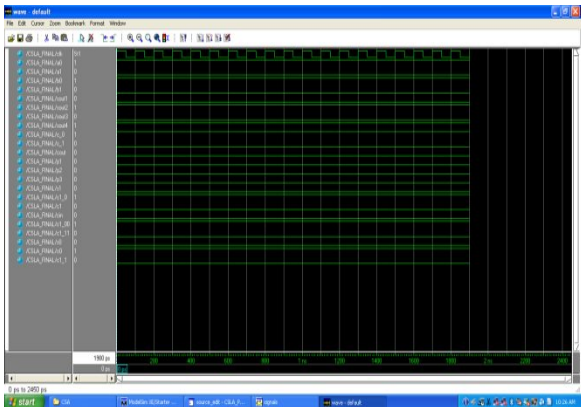

Fig.4.Simulation waveform of Proposed CS adder design.

The schematic view of proposed CS adder design is shown below

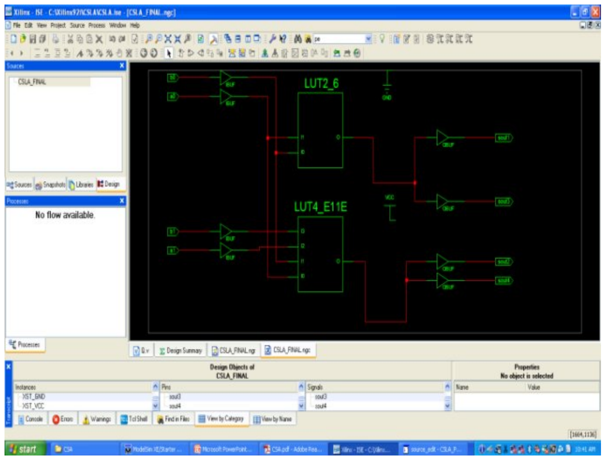

Fig.5.Schematic view of CS adder design.

Below table.1. shows the performance analysis of delay using various methodologies.

Performance Analysis:

Methodology	Delay
Proposed	5.799ns
Ramkumar et al(2012)	6.88ns
Wey et al(2012)	7.38ns

Area can be calculated as follows:

Area=Number of IOS *Number of bonded IOBs*IBUF*OBUF.

Area Analysis

Methodology	Area
Proposed	1152um
Conventional CSLA	1722um

Proposed CSLA design gives significantly less delay and area then compared to usual CSLA and CSLA based on BEC. Power consumption can be greatly reduced due to reduction of delay and area in the proposed CSLA design.

IV. CONCLUSION

Analyze the logic operations involved in usual carry select adder (CSLA) and CSLA based on BEC to study the data dependence, and also known the redundant logic operations in it. We have removed all the redundant logic operations involved in the usual CSLA and described a new logic formulation for the CSLA. In the proposed method, the carry select (CS) operation is scheduled before the calculation of *final-sum*, which is entirely different from the usual approach. Carry words corresponding to *c*in = 0 and cin=1 depend on the proposed scheme. It is used for logic optimization of CS unit. Based on this, an optimized design for Carry Select and Carry Generation units are obtained. An efficient CSLA design is obtained by using optimization of logic units. Results obtained from proposed CSLA design involves significantly less area and delay than the recently proposed CSLA based on BEC. Due to the small carry-output delay, the proposed design for CSLA is a good candidate for square-root (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.
[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.