High Speed, Low Power And Area Efficient Carry-Select Adder

Nelanti Harish
M.tech .VLSI Design
Electronics and Communication Department
SRM university, Chennai
Tamil Nadu

Mrs.V.Sarada
Asst.prof (Sr.grade)
Electronics and Communication Department
SRM university, Chennai
Tamil Nadu

Abstract- **Low power, area efficient and high performance VLSI system are used in portable and mobile devices , wireless receivers, and biomedical instrumentation 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. general types of adder are RCA ,Conventional CSLA, BEC-Based CSLA. 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. the logic operation 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 this method we use different sub modules half sum generation ,carry generation-0,carry generation-1,carry selection and full sum generation and the carry select (CS) operation is scheduled before the calculation of final-sum, due to this circuit will generate small delay, less power and less area.**

Index Terms- Adders, Binary Excess Converter (BEC), Carry Select Adder (CSLA), Logic gates.

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. Concatenating the N full adders forms N bit Ripple carry adder. In this carry out of previous full adder becomes the input carry for the next full adder. It calculates sum and carry according to the following equations. As carry ripples from one full adder to the other, it traverses longest critical path and exhibits worst-case delay.

$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_{i+1} = A_i B_i + (A_i + B_i) C_i$$

where i = 0, 1, …, n-1

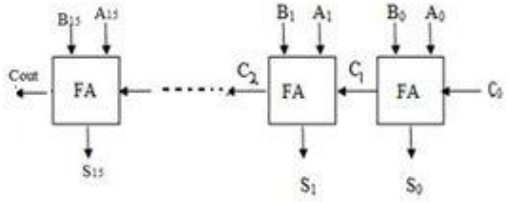

FIG:1 Ripple carry adder

RCA is the slowest in all adders (O (n) time) but it is very compact in size (O (n) area). If the ripple carry adder is implemented by concatenating N full adders, the delay of such an adder is 2N gate delays from Cin to Cout. The delay of adder increases linearly with increase in number of bits. Block diagram of RCA is shown in figure 1. In digital adders, the speed of addition is limited by the time required to propagate

ISSN: 2278 – 7798

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 Cin=0 and Cin=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 Cin=1 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. proposed a square-root (SQRT)-CSLA to implement large bit-width adders with less delay. In a SQRT CSLA, 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. Ramkumar and Kittur [6] suggested a binary to BEC-based CSLA. The BEC-based CSLA involves less logic resources than the conventional CSLA, but it has marginally 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 conventional 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 conventional and BEC-based CSLAs to study the data dependence and to identify redundant logic operations. Based on this analysis, we have proposed a logic formulation for the CSLA. The main contribution in this brief are logic formulation based on data dependence and optimized carry generator (CG) and CS design. Based on the proposed logic formulation, we have derived an efficient logic

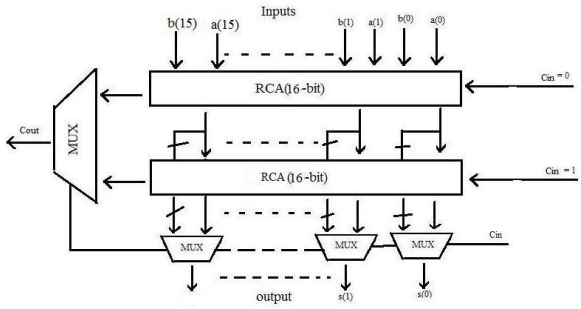

Fig:2 conventional CSLA

II. Existing System

To reduce the area and power consumption Binary Excess-1 converter instead of RCA with Cin = 1. To reduce delay compared to regular SQRT CSLA. To replace the n-bit RCA, an n+1 bit BEC is required. A structured and the function table of a 4-b BEC are shown in fig 3 , respectively. Fig 4 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 cin. The Boolean expressions of the 4-bit BEC is listed as[5]

$$X_0 = \sim B_0$$
$$X_1 = B_0 \oplus B_1$$
$$X_2 = B_2 \oplus (B_0 \,\&\, B_1)$$
$$X_3 = B_3 \oplus (B_0 \,\&\, B_1 \,\&\, B_2)$$

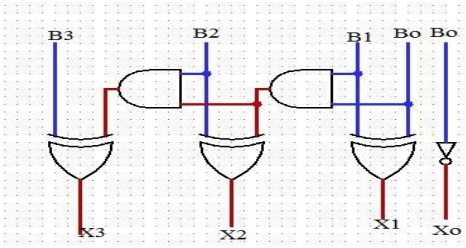

FIG:3 BEC-1(Binary to exces-one convertor) circuit

ISSN: 2278 – 7798

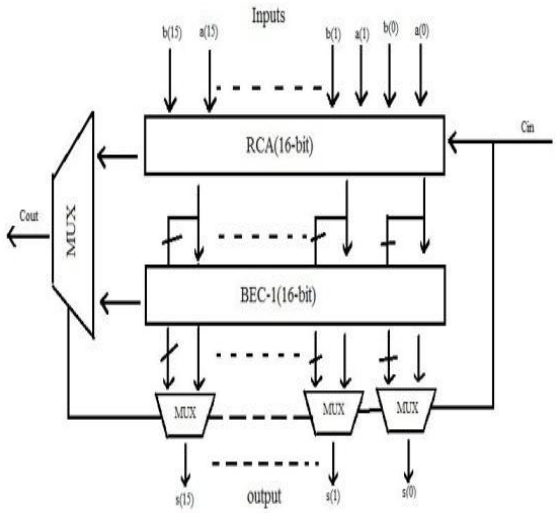

FIG:4 Block Diagram of BEC-1 Based CSLA

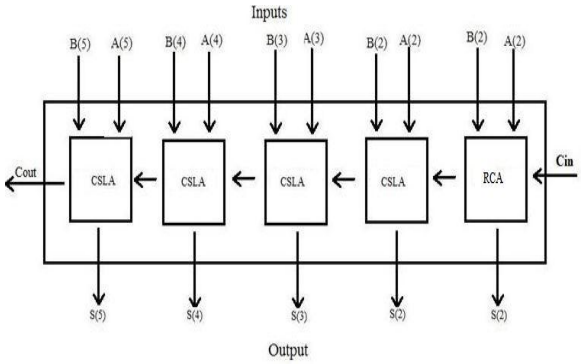

FIG: 5 Block diagram of SQRT CSLA

III. PROPOSED ADDER DESIGN

Design consists of one HSG unit, one FSG unit, one CG unit, and one CS unit. The CG unit is composed of two CGs (CG0and CG1) corresponding to input-carry '0' and '1'. The HSG receives two n-bit operands (A and B) and generate half-sum and half-carryword $c_0$of width n-bits each. Both CG0 and CG1 receive s_0 and c_0 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. 5. The logic circuits of CG0and CG1are optimized to take advantage of the fixed input-carry bits. The optimized designs of CG0and CG1are shown in Fig. 6 and 7, respectively.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.8, 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 XOR with(n−1)MSBs of half-sum(s0)in the FSG [shown in Fig.9] to obtain(n−1)MSBs of final-sum(s).TheLSBofs0isXORed with Cin to obtain the LSB of s.[1]

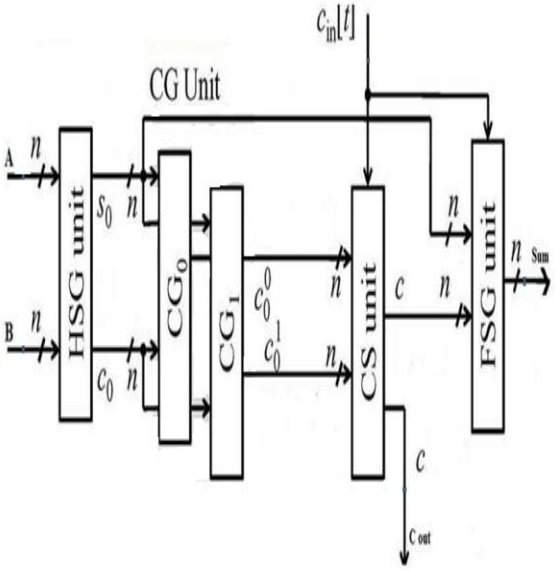

Fig:6 Sub Modules of Proposed CSLA

ISSN: 2278 – 7798

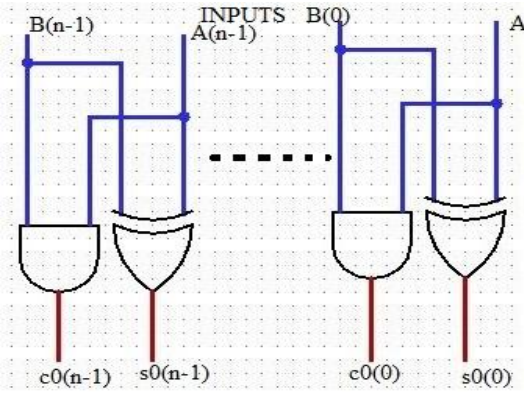

Fig:7 half sum generation

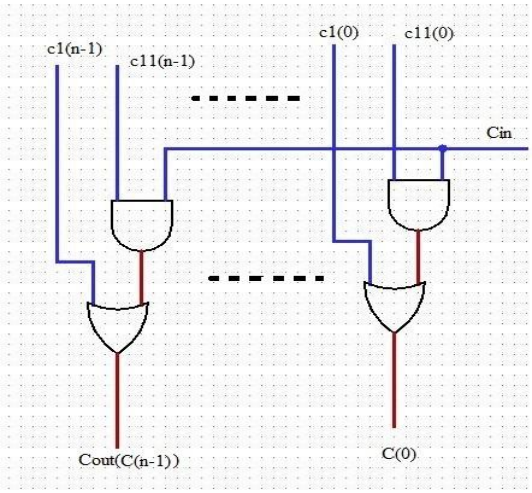

Fig:10 carry selection Unit

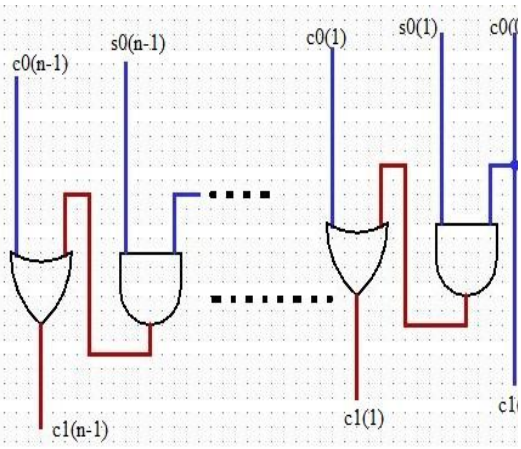

Fig:8 carry generation-0

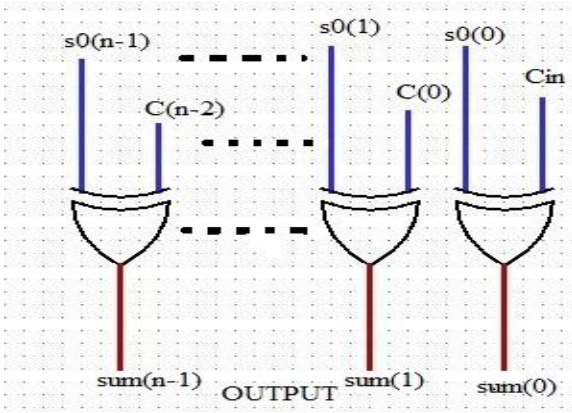

Fig:11 full sum generation

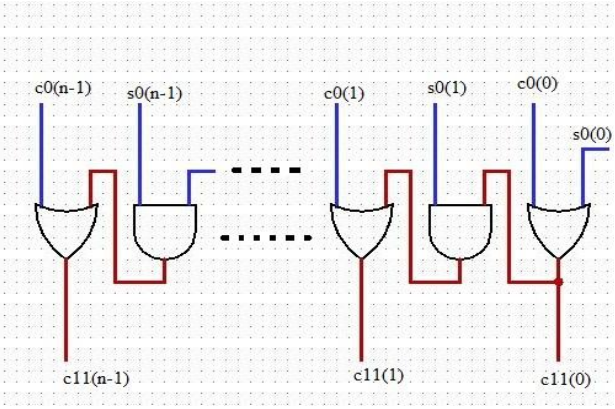

Fig:9 carry generation -1

Fig:12 Simulated result of SQRT CSLA

ISSN: 2278 – 7798

modification Project Status (03/08/2016 - 21:30:48)			
Project File:	modification.xise	**Parser Errors:**	No Errors
Module Name:	ptop16	**Implementation State:**	Synthesized
Target Device:	xc7a100tfi-2Lcsg324	**• Errors:**	No Errors
Product Version:	ISE 14.6	**• Warnings:**	5 Warnings (5 new)
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	25	63400	0%
Number of fully used LUT-FF pairs	0	25	0%
Number of bonded IOBs	50	210	23%

Fig:13 synthesize result of SQRT CSLA

IV. Performance Comparison

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. All the designs are synthesized in the xilinx tool.

TABLE 1 Comparison of BEC-based CSLA and SQRT-CSLA

Parameter	BEC-based CSLA(16-bit)	SQRT-CSLA (16-bit)
Delay(ns)	5.99(ns)	5.73(ns)
No of Slice LUT's used	42	25
Total slice LUT'S present	63400	63400
Number of bonded I/O's used	50	50
Total Number of bonded I/O's present	210	210
Power(mW)	82.16(mW)	77.23(mW)

Conclusion

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. A simple approach is to reduce the area and power of CSLA architecture. The reduced number of gates of this work offers the great advantage in the reduction of area and also the power. The modified CSLA architecture is , low area, low power, simple and efficient for VLSI hardware implementation.

References

[1] Basant Kumar Mohanty, Senior Member, IEEE, and Sujit Kumar Patel "Area–Delay–Power Efficient Carry-Select Adder" IEEE Transactions On Circuits And Systems—Ii: Express Briefs, Vol. 61, No. 6, June 2014

[2] K.K.Parhi,VLSI Digital Signal Processing. New York, NY, USA: Wiley,1998.

[3] 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.

[4] O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Comput.,vol. EC-11, no. 3, pp. 340–344, Jun. 1962.

[5] 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.

[6] Y. He, C. H. Chang, and J. Gu, "An area-efficient 64-bit square root carry select adder for low power application," inProc. IEEE Int. Symp. Circuits Syst., 2005, vol. 4, pp. 4082–4085.

[7] 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.

[8] 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," inProc. IMECS, 2012, pp. 1–4.

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

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

ISSN: 2278 – 7798