ISSN (Online) : 2454-4159
International Journal of Advanced Research in Science Management and Technology
Volume 3, Issue 1, January 2017

An Analysis of Delay and Area Efficient Carry-Select Adder

Vinod A. Malpure[1], Prof.Gurpreet Singh [2]
PG Scholar, Department of VLSI Engg., Trinity Institute Of Technology And Research, Bhopal M.P., India[1]
Professor, Department of VLSI Engg., Trinity Institute Of Technology And Research, Bhopal M.P., India[2]

ABSTRACT— A digital adder with best area & speed is one of the major areas of study in VLSI system design. With finest area & speed, reducing the power utilization is also main area of research in VLSI system design. Our approach uses a carry select adder design for the realization of fast adder. There are various options for implementation of carry select adder. Carry select adder (CSLA) is one of the fastest adders and in many data processing processors to achieve fast arithmetic function. From the construction of the CSLA, it is clear that there is an option for reducing the area and power utilization in the CSLA. This work uses an easy and well-organized gate-level modification to considerably reduce the area and power consumption of CSLA. We compare some of these techniques with existing conventional fast adder design to verify its efficiency.

KEYWORDS— VLSI, Carry select adder, fastest adders, power utilization.

I. INTRODUCTION

Design of high speed digital adders with efficient area and power is one of the major areas of study in VLSI system design. Adders are the central components in common purpose microprocessors and digital signal processors. The Ripple Carry Adders (RCAs) have the most compact design between all types of adders; but they are the slowest types of adders. On the other hand, Carry Look-ahead Adders (CLAs) are the fastest adders, except they are the worst from the area point of view. Carry Select Adders (CSAs) have been measured as a compromise result between RCAs and CLAs as they suggest a good exchange between the compact area of RCAs and the short delay of CLAs. In digital adders, the speed of addition is restricted by the time required to circulate a carry through the adder. The addition for each bit position in a basic adder is generated in order only after the previous bit position has been summed and a carry propagated into the next position.

The CSLA used in a lot of arithmetic systems to solve the trouble of carry propagation delay by separately generating multiple a carry and then choose a carry to create the final sum [1] [2]. Though, the CSA is not area efficient since it uses many pair of adders to generate a partial sum and carry by considering carry input Cin=0 and Cin=1, after that the final sum and carry are selected by the mux [1].The carry out bit of the previous block of the adders acts as the pick signal to the mux. A number of examples of such adders have been published and there are several efficient executions.

II. LITERATURE SURVEY

As we identify adders are of fundamental importance in a wide variety of digital systems, a lot of fast adders be present however adding fast using low area and power is still difficult. In digital adders, the speed of addition is restricted by the time required to spread a carry through adder. So the CSLA is used in many computational systems to improve the problem of carry propagation delay. Thus many papers were published on this through a number of examples of such adders and many capable implementations were also done.

In 1962, O.J.Bedrij [1] described the extremely fast digital adder with sum selection and multiple-radix carry. He compared the amount of hardware and the logical delay for a 100-bit ripple-carry adder and a carry-select adder. The difficulty of carry-propagation delay was overcome by separately generating multiple-radix carries and using these carries to select among all together generated sums. In this adder system, the addend and augend were divided into subaddend and subaugend sections that were added twice to produce two sub sums. One addition was done with a carry

ISSN (Online) : 2454-4159
International Journal of Advanced Research in Science Management and Technology
Volume 3, Issue 1, January 2017

digit forced into each section, and the other addition combined the operands without the forced carry digit. The selection of the correct, or true, sub sum from each of the adder sections depended upon whether or not there actually was a carry into that adder section.

There are many carry select adder approaches available but most of them use ripple carry adder. T.Y.Chang and M.J.Hsiao [3], suggested that instead of using dual carry-ripple adders, a carry select adder scheme using an add one circuit to replace one carry-ripple adder requires 29.2% fewer transistors with a speed penalty of 5.9% for bit length n=64. If speed was crucial for this 64 bit adder, then two of the original carry-select adder blocks could be substituted by the proposed scheme with a 6.3% area saving and the same speed.

The Youngjoon kim and Lee-Sup Kim [4] suggested that a carry-select adder could be implemented by using single ripple carry adder and an add-one circuit instead of using dual ripple-carry adders. They proposed a new add-one circuit using the first zero finding circuit and multiplexers to reduce the area and power with no speed penalty. For bit length n=64, this new carry-select adder requires 38 percent fewer transistors than the dual ripple-carry carry-select adder and 29 percent fewer transistors than Chang's carry-select adder using single ripple carry adder. This new 64b adder had 3.45 ns delay time at 2.5 V power supply using a 0.25 um CMOS technology. Behnam Amelifard et.al [6], suggested a new adder called carry select adder with sharing (CSAS) which was area efficient but the delay was more. M.Alioto et.al [5], suggested using variable size block sizing depending on the multiplexers delay.

The B.Ramkumar, H.M.Kittur, and P. M. Kannan [7] suggested a very simple approach to improve the speed of addition. Based on this approach a 16, 32 and 64-bit adder architecture was developed and compared with conventional fast adder architectures. In many parallel multipliers to speed up the final addition, CLA was arranged in the form of Carry Select adder (CSLA) & was used. But due to the structure of the CSLA it occupied more chip area, because it uses multiple pairs of RCA's to generate the partial sum and carry by considering Cin=0 and Cin=1.Thus the complexity of the final adder structure was high. So they replaced the RCA (CLA) with Cin=1 with BEC logic, which reduced the maximum area and delay in the final adder structure.

III. LOGIC FORMULATION

The CSLA has two parts:

1) The sum and carry generator unit (SCG)
2) The sum and carry selection unit

The SCG unit consumes for the most part of the logic resources of CSLA and considerably contributes to the critical path. Different logic designs have been recommended for efficient execution of the SCG unit. We prepared a study of the logic designs recommended for the SCG unit of conventional and BEC-based CSLAs of by appropriate logic terms. The most important aim of this study is to identify unneeded logic operations and data dependence.

Logic terms of the SCG Unit of the Conventional CSLA:

As shown in Fig.1 (a), the SCG unit of the conventional CSLA [1] is composed of two n-bit RCAs, where n is the adder bit-width. The logic function of the n-bit RCA is performing in four stages: 1) half-sum generation (HSG); 2) half-carry generation (HCG); 3) full-sum generation (FSG); and 4) full carry generation (FCG). Assume 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 (c0out and c1out) equivalent to input-carry (cin = 0 and cin = 1), respectively.

ISSN (Online) : 2454-4159
International Journal of Advanced Research in Science Management and Technology
Volume 3, Issue 1, January 2017

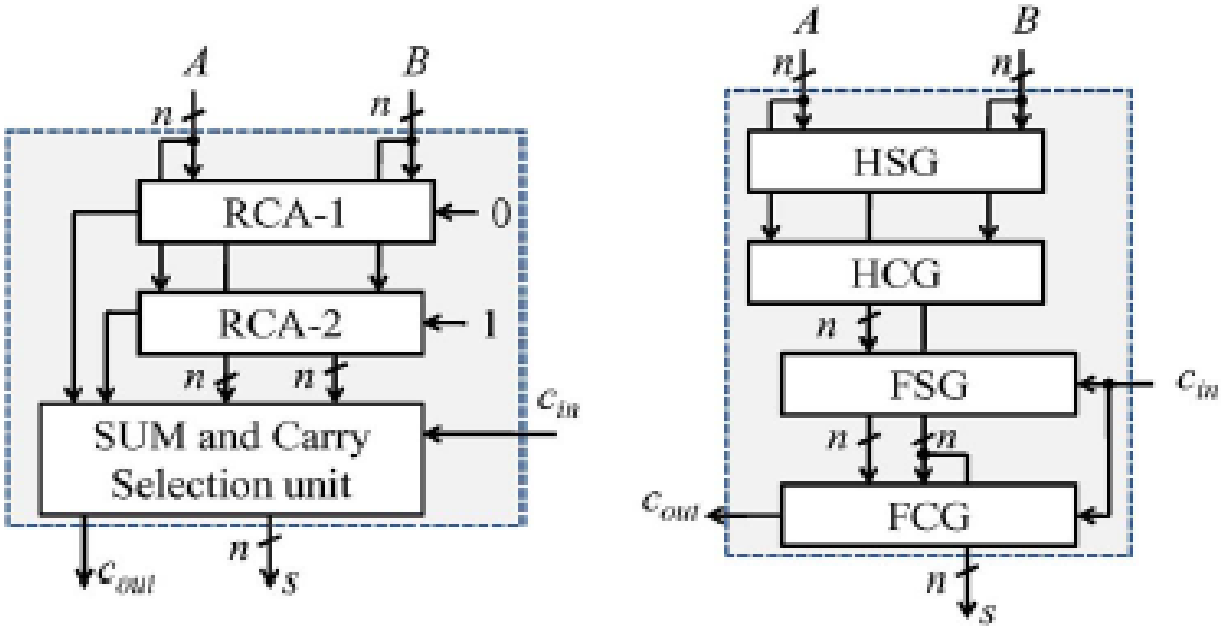

Fig.1 (a) Conventional CSLA; *n* is the input operand bit-width. (b) The logic operations of the RCA is shown in split form, where HSG, HCG, FSG, and FCG represent half-sum generation, half-carry generation, full-sum generation, and full-carry generation, respectively

Logic terms of RCA-1 and RCA-2 of the SCG unit of then-bit CSLA are given as

$s00(i) = A(i) \oplus B(i)$ $c00(i) = A(i) \cdot B(i)$ ---------- (1a)

$s01(i) = s00(i) \oplus c01(i-1)$ ---------- (1b)

$c01(i) = c00(i) + s00(i) \cdot c01(i-1)$ $c0out = c01(n-1)$ ---------- (1c)

$s10(i) = A(i) \oplus B(i)$ $c10(i) = A(i) \cdot B(i)$ ---------- (2a)

$s11(i) = s10(i) \oplus c11(i-1)$ ---------- (2b)

$c11(i) = c10(i) + s10(i) \cdot c11(i-1)$ $c1out = c11(n-1)$ ---------- (2c)

Where $c01(-1) = 0$, $c11(-1) = 1$, and $0 \le i \le n-1$.

As shown in (1a) – (1c) and (2a) – (2c), the logic term of {s00(i), c00(i)} is the similar to that of {s10(i), c10(i)}. These unneeded logic operations can be removed to have an improved design for RCA-2, in which the HSG and HCG of RCA-1 is shared to construct RCA-2. Based on this, [4] have used an add-one circuit instead of RCA-2 in the CSLA, in which a BEC circuit is used in [8] for the same purpose. Since the BEC-based CSLA offers the best area–delay–power efficiency among the existing CSLAs, we discuss here the logic terms of the SCG unit of the BEC-based CSLA as well.

Logic terms of the SCG Unit of the BEC-Based CSLA:

As shown in Fig.2, the RCA calculates n-bit sum s01 and c0out related to cin = 0. The BEC unit receives s01 and c0out from the RCA and generates (n + 1)-bit excess-1 code. The most significant bit (MSB) of BEC represents c1out, in which n least significant bits (LSBs) represent s11. The logic terms of the RCA are the same as those given in (1a)–(1c).

ISSN (Online) : 2454-4159
International Journal of Advanced Research in Science Management and Technology
Volume 3, Issue 1, January 2017

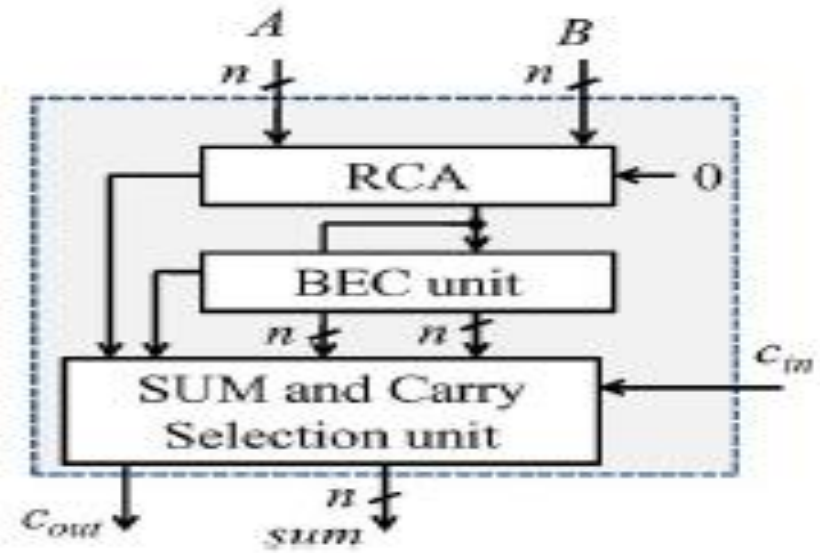

Fig.2 Structure of the BEC-based CSLA; *n* is the input operand bit-width.

The logic terms of the BEC unit of the n-bit BEC-based CSLA are given as

$s11(0) = s01(0)\ c11(0) = s01(0)$ ---------- (3a)

$s11(i) = s01(i) \oplus c11(i-1)$ ---------- (3b)

$c11(i) = s01(i) \cdot c11(i-1)$ ---------- (3c)

$c1out = c01(n-1) \oplus c11(n-1)$ ---------- (3d)

IV. PROPOSE SYSTEM

We can get from (1a)–(1c) and (3a)–(3d) that, in the case of the BEC-based CSLA, c11 depends on s01, which otherwise has no dependence on s01 in the case of the conventional CSLA. The BEC method hence increases data dependence in the CSLA. We have considered logic terms of the conventional CSLA and made a more study on the data dependence to find an optimized logic term for the CSLA. It is interesting to note from (1a)–(1c) and (2a)–(2c) that logic terms of s01 and s11 are the same but the terms c01 and c11 since (s00 = s10 = s0). Also, we find that c01 and c11 depend on {s0 , c0 , cin}, where c0 = c00 = c10 . Since c01 and c11 have no dependence on s01 and s11, the logic operation of c01 and c11 can be planned before s01 and s11, and the select unit can select one from the set (s01, s11) for the final-sum of the CSLA. We find that a major amount of logic resource is used up for calculating {s01, s11}, and it is not an efficient approach to reject one sum-word after the calculation. As an alternative, one can select the required carry term from the expected carry terms {c0 and c1} to calculate the final-sum. The selected carry term is added with the half-sum (s0) to generate the final-sum (s).

Using this method, one can have three design advantages:

- Calculation of s01 can be avoided in the SCG unit.
- The n-bit select unit can be required instead of the (n + 1) bit.
- Small output-carry delay.

V. CONCLUSION

In this paper, various types of Carry select adder design have been reviewed from the most current and previous published research work. A variety of different logics have been used till now, shown in this paper, to build the carry select adder to reduce the power, delay, area and power-delay product and transistors count. Based on this survey it is

conclude that the logic operations occupied in conventional carry select adder (CSLA) and BEC based carry select adder (CSLA) i.e., binary to excess-1 converter based carry select adder have the data dependence and unneeded logic operations. We can remove all the unneeded logic operations included in the conventional CSLA and propose a new logic formation for CSLA**.** In the propose idea, the carry select (CS) function can planned before the calculation of final-sum which will be different from the conventional approach. A capable CSLA design can be obtain by using better logic units. The propose carry select adder (CSLA) design will help to reduce more area and delay than the recently proposed BEC-based carry select adder.

REFERENCES

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

[2] **J.Sklansky, "Conditional-Sum Addition Logic," IRE Trans. Electron. Comput., vol. EC-9, pp.226-231, 1960.**

[3] **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.**

[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] **M.Alioto et.al, "A Gate Level Strategy To Design Carry Select Adders," ISCAS 2004.**

[6] **Behnam Amelifard et.al "Closing the Gap between Carry Select Adder and Ripple Carry Adder," Proceedings of the Sixth International Symposium on Quality Electronic Design (ISQED'05), 2005.**

[7] **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.**

[8] **B. Ramkumar and H.M. Kittur, "Low Power and Area Efficient Carry select Adder" Very Large Scale Integration (VLSI) Systems, IEEE Transactions on Volume: 20, Issue: 2, 2012.**