International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

An Efficient Architecture of Carry Select Adder Using Logic Formulation

Kaveri .N[1], Senthil Kumar .P[2]

[1]PG Student, ECE Department, Velalar College of Engineering and Technology, Erode

[2]Assistant Professor, ECE Department, Velalar College of Engineering and Technology, Erode

Abstract: ***A conventional Carry SeLect Adder (CSLA) consists of two Ripple Carry adders (RCA) and a multiplexer. It is used to generate two sum and carry words for both C_{in}=1 and C_{in}=0 and selects correct output sum and carry according to the input carry bit. The Carry Propagation Delay (CPD) is low but the area is high due to the use of 2-RCA unit. In the existing technique, the logic operation of CSLA can be performed by using Half Sum Generator (HSG), 2-Carry Generator (CG), Carry Selector (CS) and Full Sum Generator (FSG). It has less CPD but 2-CG unit increases the delay and area. In the proposed method, the CS unit is scheduled before the generation of carry words and hence the single CG unit is enough to calculate the final output carry. It results in reduction of area by 15% and delay by 10%.***

Keywords: Adder, Carry Generator, Carry Selector, Ripple carry adder, Carry select adder.

1. Introduction

An efficient VLSI system is widely used in portable, military and bio-medical applications. An adder is a main component in most of the DSP systems which are used in VLSI. Hence an efficient adder design can be used to increase the performance of VLSI systems. Generally, an n-bit addition is performed using RCA which has high CPD. An efficient adder must consist of less CPD, area and power.

A carry select adder is efficient among various types of adder. The main drawback of this adder is its high carry propagation. Hence, different architectures of carry select adder are proposed to increase its efficiency.

Ramkumar B. and Kittur H.M. [9] alleviate the problem of carry propagation delay. They select a carry bit to generate a final sum after all possible carry bits are generated independently. This operation can be done using Binary Excess Converter (BEC) and a multiplexer. This architecture has less number of logic gates than regular Full Adder (FA) structure but it consumes more power when number of bits increases.

The ripple carry adder design can be constructed by cascading each single bit full adder [12]. Here, the RCA is portioned into M-parts and a duplicated (N/M)-bit carry ripple adder pair is used in each part. It has less logic resources but the carry propagation delay and a delay time is high for large number of bits.

The Binary excess converter can be replaced by a common Boolean logic to increase the efficiency of CSLA [5]. The duplicated adder cell can be removed by sharing the common Boolean logic term. This architecture consumes less area but the delay of BEC is high and also it requires more logic resources.

Basant Kumar Mohanty and Sunil Kumar Patel [2] analyzed the data dependencies of various carry select adder. They construct CSLA using sum and carry generation unit and sum and carry selection unit. This architecture has the advantage of less power consumption but the use of 2-CG block increases the overall area and delay.

2. Carry Select Adder

An N-bit addition can be performed using ripple carry adder. The RCA consists of N-full adder which calculates the sum and carry for each bit. Here, each full adder has to wait for previous carry propagation. This delay is called carry propagation delay and it has to reduce for an efficient adder. The carry select adder is efficient to reduce the overall CPD.

The ripple carry adder and a multiplexer is a major component of a conventional CSLA. It consists of 2-RCA, one generates the set of sum words by assuming the carry is '0' and the other generates another set of sum words by assuming carry is '1'. Then the final sum words are selected once the input carry is known.

The selection process is done using the multiplexer. Two sets of sum words are given to sum and carry selection unit which acts as a multiplexer. Here, C_{in} acts as a control signal. The multiplexer selects the sum words which is calculated by assuming carry is '0' when C_{in}=0. When C_{in}=1, it selects another set of sum words. It needs not to wait for previous carry to propagate. Hence, the overall carry propagation delay is reduced.

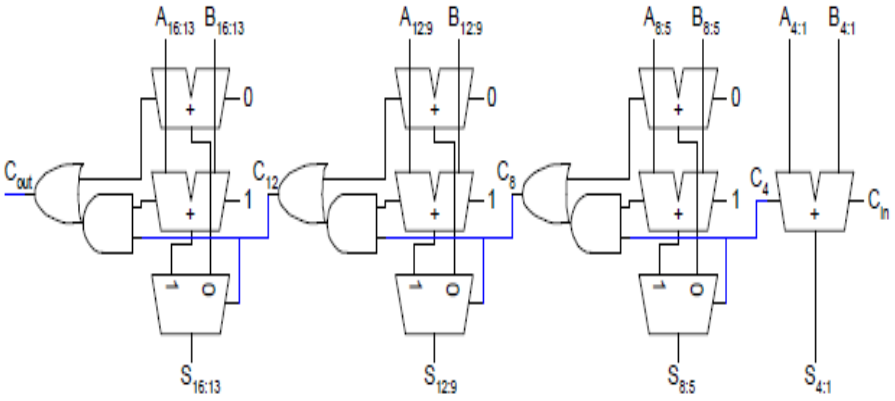

Figure 1: Conventional Carry select adder

3. BEC based CSLA

The logic operation of the RCA consists of Half Sum Generator (HSG), Half Carry Generator (HCG), Full Sum Generator (FSG) and Full Carry Generator (FCG). The HSG and HCG unit generates half sum and carry words and according to the C_{in} value, the FSG and FCG delivers the final output sum and carry.

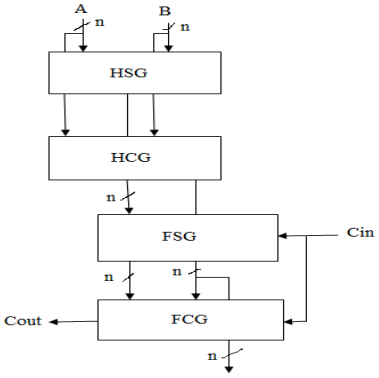

Figure 2: Logic operation of CSLA

The conventional carry select adder is not more attractive because of the use of two ripple carry adders. Hence, one of the ripple carry adder is replaced by the Binary excess Converter. The RCA generates the sum words by assuming carry is '0'. The sum words thus generated is given to the BEC units which increment the value by '1'. Then, according to the C_{in} value, the sum and carry selection unit gives the final sum and carry bits.

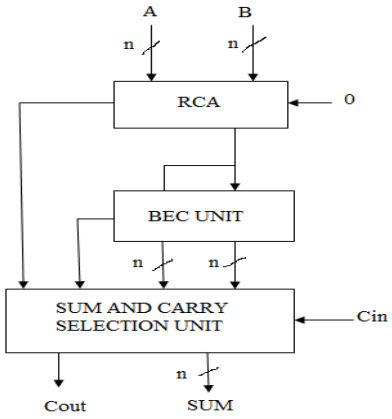

Figure 3: BEC based CSLA

4. Existing Adder Design

In the existing design, the logic operation of CSLA can be performed using four major units.

1) Half Sum Generator (HSG),
2) Carry Generator (CG),
3) Carry Selector (CS) and
4) Full Sum Generator(FSG)

This architecture consists of two carry generation units each generates carry words by assuming carry is '0' and '1'. Then the correct carry words can be selected using the carry selection unit.

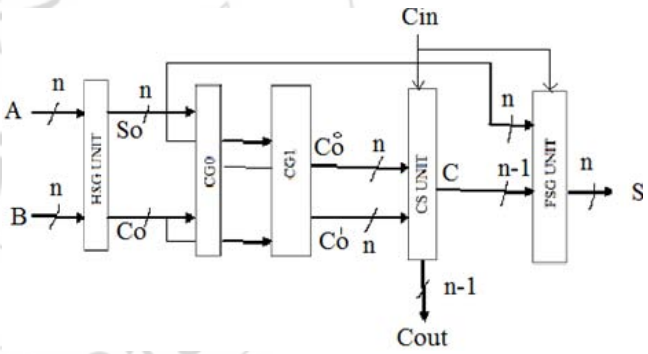

Figure 4: Existing CSLA design

The n-bit inputs are given to HSG unit. The HSG unit consists of EX-OR and AND gates. The output of EX-OR gate gives half sum S_0 and the output of AND gate is half carry word C_0. Then these half sum and half carry words are given as input to both CG_0 and CG_1. The CG_0 will generate the carry words for input carry is '0'. This unit consist of AND and OR gates. The AND gates has previous half carry and present half sum as input and generate the output. This output is ORed with present half carry bit to generate the carry bit for '0' input. This output is then given to the carry selection unit.

In the CG_1 unit, OR-AND-OR gate are present. The OR gate has present half sum and present half carry to generate the present carry for '1' input. This output is given to AND gate and carry for all n-bits are generated as CG_0 unit. This final carry is also sent to the carry selection unit.

The carry selection unit is used to select to carry output according to the input carry bit. This block can be

implemented using the 2-1 multiplexer. But it has specific bit pattern which is analyzed using the truth table of carry selection unit. If $C_1^0(i)$='1', then $C_1^1(i)$='1' independent of $S_0(i)$ and $C_0(i)$. Hence, the carry selection unit consists of AND-OR gate. The AND gate has C_{in} and carry output of CG_0 as its input and gives its output to OR gate. It is OR-ed with carry output of CG_1 and generates the final carry output.

The sum output can be generated using the FSG unit. The FSG unit consists of N-number of EX-OR gates. It has previous carry bit and half sum word as its input and generates the sum output.

5. Proposed Adder Design

The existing adder design has the advantage of less carry propagation delay. Due to the use of two carry generation unit and carry selection unit, it consumes more area. In proposed adder design, in order to reduce the area, the carry selection and generation can be done in the same unit.

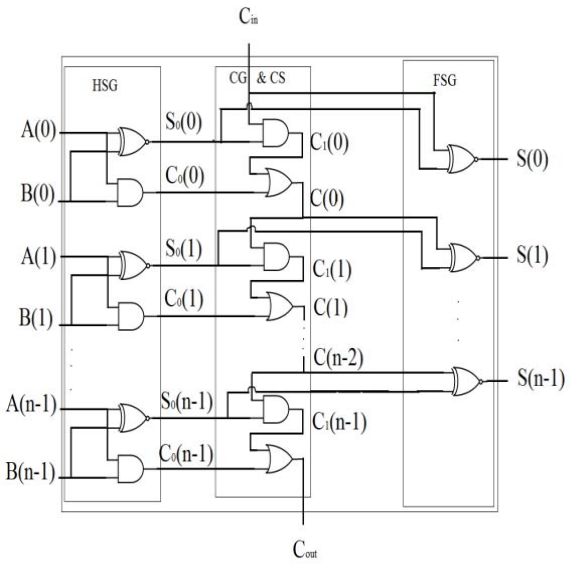

Figure 5: Proposed CSLA design

The HSG unit and FSG unit is same as existing adder design. The modification is done in the carry generation and carry selection unit. Here, the carry selection is scheduled before the carry generation. To generate the carry output, AND-OR gates are used. The HSG unit generates the half sum and half carry word.

The carry input C_{in} and half sum words are given as input to AND gate. It gives the output of half carry word for input '1'. The half carry word from HSG unit can be directly taken as half carry word for input '0'. Both these half carry words are given to the OR gate to generate the final carry bit. The carry output of (n-1)-bit is taken as C_{out}. The sum words can be generated by the FSG unit.

6. Result Analysis

Various types of carry select adders are analyzed and its corresponding performance in terms of area, delay and power was calculated. The design is coded in VHDL language and simulated using Modelsim 10.1b.

The BEC-based CSLA consumes less area and power compared to the conventional CSLA. It is due to the replacement of 1-RCA with binary excess converter. The BEC based CSLA is simulated for 16-bits. The existing adder design replaces both RCA with HSG, CG, CS and FSG units. The design is simulated for16-bit input. It consumes less power and area than BEC-based CSLA.

The proposed adder design has single carry generation and selection block. Hence the delay associated gets reduced. In an existing design, for the generation of single carry output, it requires about 3 AND gates and 4 OR gates. In our proposed adder design, carry output of each bit can be generated using 1 AND and 1 OR gate. Hence the overall gate count is reduced.

The design is simulated for 16-bit input and its corresponding area, delay and power were calculated. The analysis report shows that the power consumption, area and delay are reduced in this design than existing adder design, BEC-based adder design and conventional adder design. Hence it is efficient adder in terms of area, delay and power consumption.

The waveform of proposed adder design can be given as,

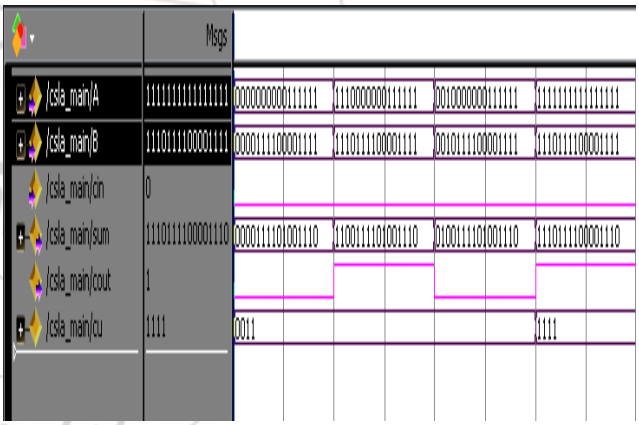

Figure 6: Waveform of Proposed CSLA

The comparison table of BEC-based CSLA, Existing CSLA and proposed CSLA are tabulated.

Table 1: Analysis of various CSLA design

Parameters	*BEC-based CSLA*	*Existing CSLA*	*Proposed CSLA*
DELAY	30.153ns	27.573ns	24.686ns
NO. OF SLICES	31	22	18
NO. OF LUTs	55	37	32
NO. OF IOs	50	50	50
NO.OF BONDED IOs	50	50	50
POWER	331mW	320mW	315mW

7. Conclusion

In this project, various Carry Select Adders have been studied and also the impacts of power consumption, delay and area have been analyzed. The designs are coded in

VHDL and simulated in ModelSim and its area, delay and power are analyzed using EDA tool Xilinx_ISE 9.2i.

References

[1] M.Alioto and G.Palumbo , 'Analysis and comparison on full adder block in submicron technology', IEEE Trans. VLSI Systems, Vol. 10, pp. 806–823, 2002.
[2] Basant Kumar Mohanty and Sujit Kumar Patel, 'Area–Delay–Power Efficient Carry-Select Adder', IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, Vol. 61, No. 6, 2014.
[3] Y.He , C.H.Chang and J.Gu, 'An area-efficient 64-bit square root carry select adder for low power application', in Proc. IEEE Int. Symp. Circuits Syst., Vol. 4, pp.4082–4085, 2005.
[4] Y.Kim and L.S.Kim, '64-bit carry-select adder with reduced area', Electron. Lett. Vol. 37, no. 10, pp. 614-615, 2001.
[5] S.Manju and V.Sornagopal, 'An efficient SQRT architecture of carry select adder design by common Boolean logic', in Proc. VLSI ICEVENT, pp. 1–5, 2013.
[6] B.Parhami, 'Computer Arithmetic: Algorithms and Hardware Designs', 2nd ed. New York, NY, USA: Oxford Univ. Press, 2010.
[7] K.K.Parhi, 'VLSI Digital Signal Processing', New York, NY, USA:Wiley, 1998.
[8] J.M.Rabaey, 'Digtal Integrated Circuits - A Design Perspective', Prentice Hall Press, 2001.
[9] 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, 2012.
[10] 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. S3-S8, 2010.
[11] A.Shams., T.Darwish and M.Byoumi, 'Performance analysis of low-power 1-bit CMOS full adder cells', IEEE Trans. VLSI Systems, Vol. 10, pp. 20-29, 2002.
[12] I.C.Wey, CC.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, pp.1–4, 2012.