# 128-BIT AREA EFFICIENT RECONFIGURABLE CARRY SELECT ADDER

Gurunadha. Ravva

Department of ECE, JNTUK-UCEV, Vizianagaram, Andhra pradesh, India

## ABSTRACT

*Adders are one of the most critical arithmetic circuits in a system and their throughput affects the overall performance of the system. 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. In this paper, we proposed an area-efficient carry select adder by sharing the common Boolean logic term. After logic optimization and sharing partial circuit, we only need one XOR gate and one inverter gate for sum generation. Through the multiplexer, we can select the final-sum only and for carry selection we need only one AND gate and one OR gate. Based on this modification 16-, 32-, 64-, and 128-bit CSLA architecture have been developed and compared with the conventional CSLA architecture. The proposed design greatly reduces the area compared to other CSLAs. From this improvement, the gate count of a 128-bit carry select adder can be reduced from 3320 to 1664. The proposed structure is implemented in Artix-7 FPGA. Compared with the proposed design, the conventional CSLA has 65.80% less area.*

.

## KEYWORDS

,
*Carry Select Adder, Area-Efficient*

## 1. INTRODUCTION

Addition is the most common and often used arithmetic operation on digital signal processor microprocessor, especially digital computers. Also, it serves as a building block for synthesis all other arithmetic operations. In electronic applications adders are most widely used. Applications where these are used are multipliers, DSP to execute various algorithms like FFT, FIR and IIR. Therefore, regarding the efficient implementation of an arithmetic unit, the binary adder structures become a very critical hardware unit. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder. There are many binary adder architecture ideas to be implemented in such applications. However, it has been difficult to do well both in speed and in area. The easiest type of parallel adder to build is a ripple carry adder, which uses a chain of one-bit full adder to generate its output. The Ripple Carry Adder (RCA) gives the most compact design but takes longer computation time. The time critical applications use Carry Look-ahead scheme (CLA) to derive fast results but lead to increase in area. The Carry Select Adder (CSLA) provides a compromise between small areas but longer delay Ripple Carry Adder (RCA) and a larger area with shorter delay Carry Look-Ahead Adder (CLA) [1].

In mobile electronics, reducing area and power consumption are key factors in increasing portability and battery life. Even in servers and desktop computers, power consumption is an important design constraint. Design of area- and power-efficient high-speed data path logic

DOI: 10.5121/ijci.2016.5438

systems are one of the most substantial areas of research in VLSI system design. In digital adders, the speed of addition is limited by the time required to propagate 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 [3]. Among various adders, the CSLA is intermediate regarding speed and area [2].

To add multiple inputs various types of Carry Propagate Adders (CPA) are present. Among them RCA design occupies the small area but takes longer computing time. The delay of RCA is linearly proportional to number of input bits. For some input signals carry has to ripple all the way from least significant bit (LSB) to most significant bit (MSB). The propagation delay of such a circuit is defined as the worst case delay over all possible input patterns also called as critical path delay. The Carry Skip Adder (CSKA) uses a carry skip scheme to reduce the additional time taken to propagate the carry signal in RCA. Thus, CSKA is faster than RCA at the expense of a few simple modifications. The CLA offers a way to eliminate the ripple effect. For every bit, sum and carry is independent of the previous bits. CLA is faster than RCA but consumes large area. CLA is fast for a design having less input bits, for higher number bits it shows the worse delay [5]. In RCA every full adder has to wait for the incoming carry before an outgoing carry is generated. The CSLA provide a way to get around this linear dependency is to anticipate both possible values of the carry input i.e. 0 and 1 and evaluate the result in advance. Once the real value of the carry is known the result can be easily selected with the help of a multiplexer stage. The CSLA is intermediate between longer delay RCA and large area CLA.

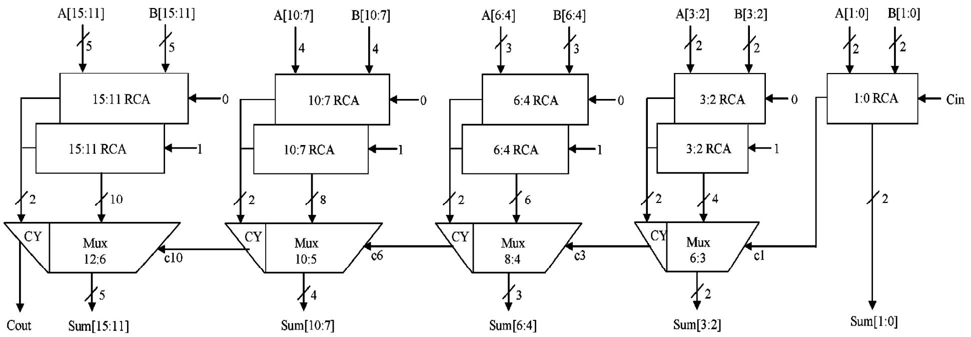

Fig. 1. 16-bit Conventional Carry Select Adder(CSLA

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).

In this paper, we proposed an area-efficient carry select adder by sharing the common Boolean logic term. After Boolean simplification, we can remove the duplicated adder cells in the conventional carry select adder. Alternatively, we generate duplicate carry-out and sum signal in each single bit adder cell. By utilizing the multiplexer to select the correct output according to its previous carry-out signal, we can still preserve the original characteristics of the parallel architecture in the conventional carry select adder. In this way, the circuit area and gate count can be greatly reduced and area delay product of the adder circuit can be also greatly lowered.

# 2. ARCHITECTURE OF CONVENTIONAL CSLA

A Carry Select Adder is a particular way to implement an adder, which is a logic element that computes the (n+1) bit sum of two n-bit numbers. The carry-select adder is simple but rather fast. The carry-select adder generally consists of two ripple carry adders and a multiplexer. Adding two n-bit numbers with a carry-select adder is done with two adders (therefore two ripple carry adders) in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one. After the two results are calculated, the correct sum, as well as the correct carry, is then selected with the multiplexer once the correct carry is known. The architecture of conventional 16-bit carry select adder is shown in Fig. 1.

## 2.1 PROPOSED CARRY SELECT ADDER

TRUTH TABLE OF FULL ADDER

| **Cin** | **A** | **B** | **Carry** | **Sum** |
|---|---|---|---|---|
| **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **0** | **1** |
| **0** | **1** | **0** | **0** | **1** |
| **0** | **1** | **1** | **1** | **0** |
| **1** | **0** | **0** | **0** | **1** |
| **1** | **0** | **1** | **1** | **0** |
| **1** | **1** | **0** | **1** | **0** |
| **1** | **1** | **1** | **1** | **1** |

The truth table of single bit full adder is shown in Table 1. From the observation of truth table, the output of full adder sum is as carry-in signal is logic ‘0’ is the inverse signal of itself as carry-in signal is logic ‘1’. The proposed logic formulation is given as

For $c_{in}$=0
Sum = A ^ B (1)
Carry = A&B (2)
For $c_{in}$=1
Sum = ~ (A ^ B) (3)
Carry = A|B (4)

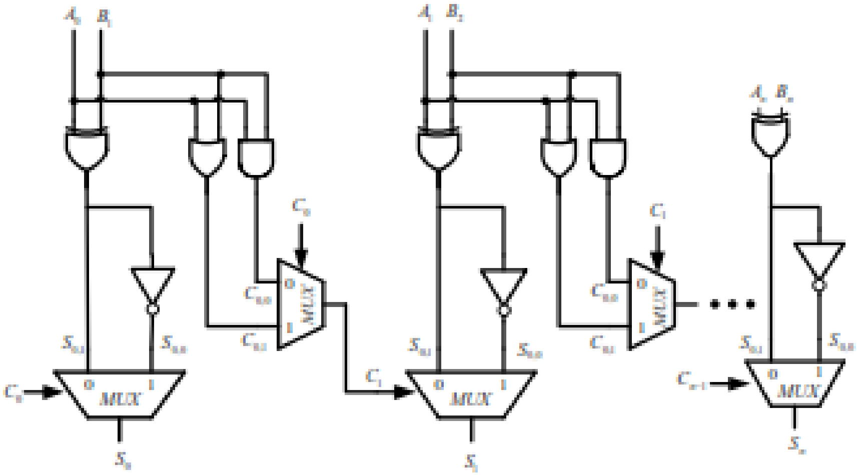

Fig. 2. The proposed adder.

The Eq. (1) and (3) are generates the sum for cin=’0’ and ‘1’. The Eq. (2) and (4) are generates the carry for cin=’0’ and ‘1’. The final sum is selects the multiplexer. As for the carry propagation path, we construct one OR gate and one AND gate to anticipate possible carry input values in advance. Once the carry-in signal is ready, we can select the correct carry-out output according to the logic state of carry-in signal. In this way, we can keep both the summation generation circuit of XOR gate and INV gate and the carry-out generation circuit of OR gate and AND gate in parallel. Since we still retain part of parallel architecture of conventional carry select adder, we can still maintain some competitiveness in speed. On the other hand, we needn’t to prepare the duplicated adder cells in the conventional carry select adder, which can greatly reduce the gate count and area delay product. The proposed area efficient adder is shown in Fig. 2. In this architecture carry selection multiplexer is replaced with one AND gate and one OR gate. The advantage of this replacement is reduce gate count and simple circuit.

# 3. COMPARISON and RESULTS

*Theoretical Estimation of Area and Delay of 128-bit CSLA*

THEORETICAL ESTIMATION OF AREA AND DELAY OF 128-BIT CSLA

| Design | Delay (No. of Gate Delays) | Area (Gate count) |
|---|---|---|
| Conventional | 49 | 3320 |
| Proposed | 385 | 1664 |

The Table II. Shows the theoretical estimation of area and delay of gate count. From this table area in terms gate count is greatly reduced from 3320 to 1664. Compared with the proposed design, the conventional CSLA has 99.51% excess area. The conventional carry select adder performs better in terms of speed. The delay of our proposed design increases because of logic circuit sharing sacrifices the length of parallel path. The delay difference existing between these two designs is mainly come from the length difference in their parallel paths. In the conventional carry select adder, it divided N bits into M blocks; however, our proposed design divided every single bit as individual block. In other words, we still retain N blocks in the N bits adder. Such arrangement will lead to some speed sacrifice.

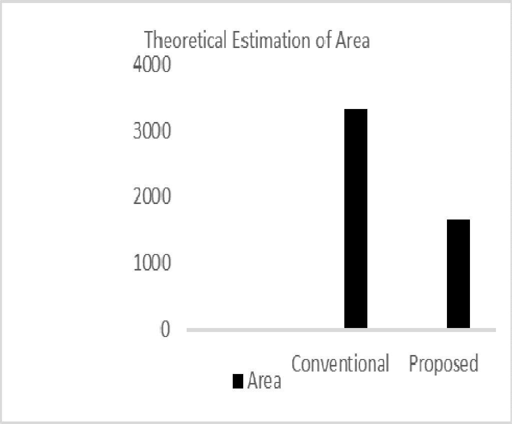

Fig. 3. The theoretical estimation of area for 128-bit CSLA.

### 3.1 FPGA Implementation

FPGA IMPLEMENTATION RESULTS

| Design | Width(n) | Delay(ns) | Slices | SDP |
|---|---|---|---|---|
| conventional | 16 | 5.902 | 26 | 153.45 |
| | 32 | 7.820 | 66 | 516.12 |
| | 64 | 11.922 | 138 | 1645.23 |
| | 128 | 28.17 | 320 | 9014.40 |
| Proposed | 16 | 6.467 | 24 | 155.20 |
| | 32 | 8.356 | 48 | 401.08 |
| | 64 | 16.134 | 96 | 1548.86 |
| | 128 | 45.009 | 193 | 8686.73 |

model to implement the proposed CSLA on the XILINX Artix 7 FPGA xc7a100t-3csg324 using the ISE Design Suite 14.5 platform. The proposed and conventional architectures are implemented 16-, 32-, 64-, and 128-bit wise. Compared with the proposed design, the conventional CSLA has 65.80% less area. The slice delay product (SDP) also less compared to conventional architecture. The implementation results in terms of logic slices and critical path latency are shown in Table III. It is obvious that our architecture has smaller hardware resource than the conventional architecture.

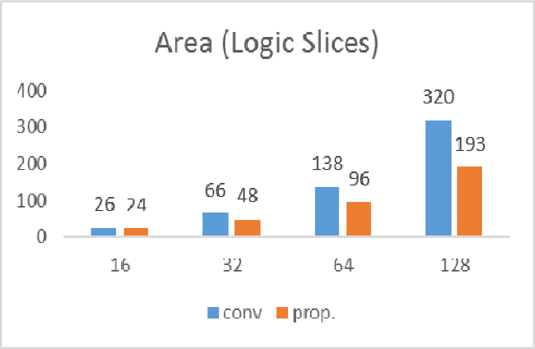

Fig. 4. Comparison of Area.

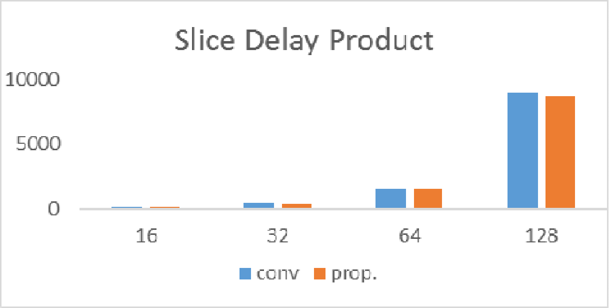

Fig. 5. Comparison of Slice Delay Product.

# 4. CONCLUSION

A simple approach is proposed in this paper to reduce the area of CSLA architecture. The proposed structure is implemented in Artix-7 FPGA. The proposed and conventional architectures are implemented 16-, 32-, 64-, and 128-bit wise. The reduced number of gates of this work offers the great advantage in the reduction of area, total power. From this improvement, the gate count of a 128-bit carry select adder can be reduced from 3320 to 1664. Compared with the proposed design, the conventional CSLA has 65.80% less area. The modified CSLA architecture is therefore, low area, simple and efficient for VLSI hardware implementation.

# REFERENCES

[1] R. E. Kalman, “A new approach to linear filtering and prediction problems,”J. Basic Eng., vol. 82, pp. 35–45, Mar. 1960.

[2] G. Enzner and P. Vary, “Frequency-domain adaptive Kalman filter foracoustic echo control in hands-free telephones,” Signal Process., vol.86, pp. 1140–1156, 2006.

[3] Jae Lim; A. Oppenheim, “All-pole modeling of degraded speech ,” IEEE Transactions on Acoustics, Speech, and Signal ProcessingYear: 1978, Volume: 26, Issue: 3

[4] C. Paleologu, J. Benesty, and S. Ciochina, “Study of the general Kalman ˇfilter for echo cancellation,” IEEE Trans. Audio, Speech, LanguageProcessing, vol. 21, pp. 1539–1549, Aug. 2013.

[5] S. E. Bou-Ghazale and K. Assaleh, “A robust endpoint detection of speech for noisy environments with application to automatic speech recognition,” in Proc. ICASSP2002, vol. 4, pp. 3808–3811, 2002.

[6] S. Gannot, “Speech processing utilizing the Kalman filter,” IEEE Instrum. Meas. Mag., vol. 15, no. 3, pp. 10–14, Jun. 2012.

[7] T. Kishore Kumar, Siva Prasad Nandyala, Ravi Bolimera," Speech Enhancement using Spectral Subtraction,Affine Projection Algorithms and Classical AdaptiveFilters ," International Conference On Advances In Electronics, Electrical And Computer Science Engineering - EEC 2012

[8] Sadaoki Furui, "50 Years of Progress in Speech and Speaker Recognition Research," ECTI TRANSACTIONS ON COMPUTER AND INFORMATION TECHNOLOGY, Vol.1, No.2, November 2005

[9] Omar Daoud, Abdel-Rahman Al-Qawasmi and Khaled daqrouq, "Modified PCA Speaker Identification Based System Using Wavelet Transform and Neural Networks," International Journal of Recent Trends in Engineering, Vol 2, No. 5, November 2009.

[10] Reynolds, D.A., "Experimental evaluation of features for robust speaker identification," IEEE Transactions on SAP, Vol. 2, pp. 639-643, 2008.

[11] Shing-Tai Pan and Chih-Chin Lai (2007)," Using Genetic Algorithm to Improve the Performance of Speech Recognition Based on Artificial Neural Network," Robust Speech Recognition and Understanding, Michael Grimm and Kristian Kroschel (Ed.)

[12] K. Koteswara rao, G. Appalanaidu, " Comparative Analysis of Eigen features based Speaker identification using Genetic Algorithm," The fifth national conference on Computer vision, pattern recognition, Image processing and Graphics (NCVPRIPG2015), December 16-19(2015)

.