RESEARCH ARTICLE OPEN ACCESS

An Efficient Higher Order And High Speed Kogge-Stone Based CSLA Using Common Boolean Logic

Kuppampati Prasad, Mrs.M.Bharathi
M. Tech (VLSI) Student, Sree Vidyanikethan Engineering College (Autonomous) A. Rangampeta, Tirupathi, India
Assistant Professor Sree Vidyanikethan Engineering College (Autonomous) A. Rangampeta, Tirupathi, India

Abstract
Adders are the basic building blocks of any processor or data path application. In adder design carry generation is the critical path. In this paper, we propose An Efficient and high speed carry select adder by replacing Ripple Carry Adders with parallel prefix adders (Kogge-Stone) for Cin=0 stage and common Boolean logic for Cin=1 stage. In this proposed method we can reduce delay and area by 3%, 26% and 5% for 16-bit , 5%, 34% and 14% for 32-bit, 8%, 41% and 19% for 64-bit and 9%,46% and 21% for 128-bit compare to the modified adders (Regular CSLA, Regular with BEC and Regular with CBL).
Keywords*- prefix adder (Kogge-Stone), CSLA, CBL, delay, area-efficient.*

I. INTRODUCTION

In Digital systems Design adder is an important component and it is used in multiple blocks of its architecture. In many Computers and in various classes of processor specialization, adders are not only used in Arithmetic Logic Units [4], but also used to calculate addresses and table indices. There exist multiple algorithms to carry on addition operation ranging from simple Ripple Carry Adders to complex CLA.

So, speed of operation is the most important constraint. 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. 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 [5]. 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 kogge-stone adder cell (pre-fix adders) instead of RCA with Cin = 0 or Cin = 1 anyone in the regular CSLA to achieve High speed, lower area and power consumption [2]–[5]. The main advantage of this pre-fix adder logic comes from High Speed than the n-bit Full Adder (FA) structure.

This paper is organized as follows; Section II explains the regular CSLA and detail structure of BEC and CBL CSLA respectively. A section III deals with proposed architecture of CSLA section IV explain about Comparisons of area and delay and Section V concludes.

II. CARRY SELECT ADDER (CSLA)

In digital electronics, carry select adder (CSLA) is an efficient adder. It is a logic element that computes the sum of two n-bit numbers. The carry-select adder generally composes of two ripple carry adders [10] and a multiplexer.

A. Ripple Carry Adder

The Ripple Carry Adder consists of group of full adders. It is used to compute addition of two N-bit numbers. It consists of N full adders to add N-bit numbers. From the second full adder, carry input of every full adder is the carry output of its previous full adder. This kind of adder is typically known as Ripple Carry Adder because carry ripples to next full adder. The layout of Ripple Carry Adder is simple, which allows fast design time. The Ripple Carry Adder [9] is slowest among all the adders because every full adder must wait till the previous full adder generates the carry bit for its input. The 3-bit RCA is shown in Figure 1. Theoretically the Ripple Carry Adder has delay of *o*(n) and area of *o*(n).

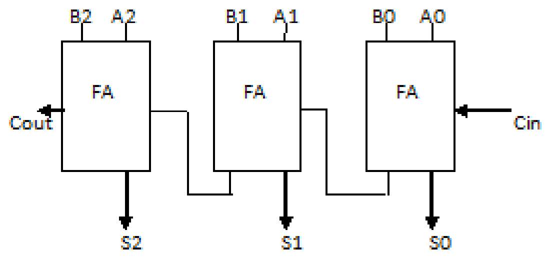


Figure1. 3-bit Ripple Carry Adder

B. Multiplexer

Multiplexer is also called as data selector or universal element. It is a combinational circuit which has many inputs and single output. Depending on the select input combination the content on one of the selected input line is transferred on to the output line. The 6:3 Multiplexer is shown in Figure 2.

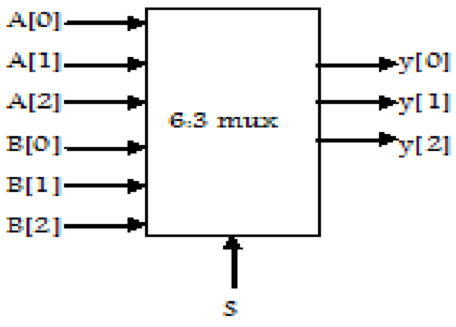


Figure 2.6:3 multiplexer

C. Carry Select Adder

The Carry Select Adder [5] consists of dual Ripple Carry Adders and a multiplexer. The modified CSLA [2] is shown in Figure 3.In this diagram the addition of two 16-bit numbers is done with two RCAs [3] of Cin=0 and Cin=1. After the calculation for two cases of carry, the correct sum as well as correct carry is selected by using multiplexer once the correct carry is known.

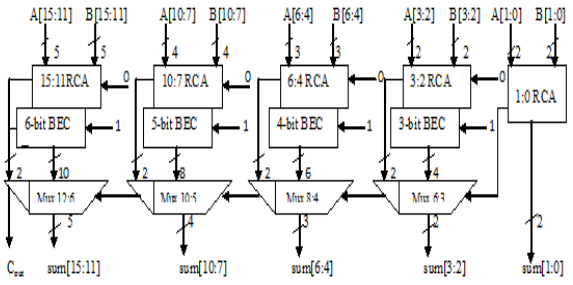


Figure3. Modified CSLA using BEC logic

There are two types of Carry Select Adders one is uniform and another one is variable carry select adder [6]. In uniform Carry Select Adder each block size is fixed in all stages, but in variable Carry Select Adder block size is variable. The delay at Cin input stage can be reduced using variable type of CSLA [2]. Theoretically delay and area of Carry Select Adder are O(√n) and O(2n) respectively.

This method replaces the BEC add one circuit Common Boolean Logic.[1] The output waveform of full adder for carry in signal is '1' is generate summation and carry signal by just using an TNV and OR gate. It is shown in figure 4

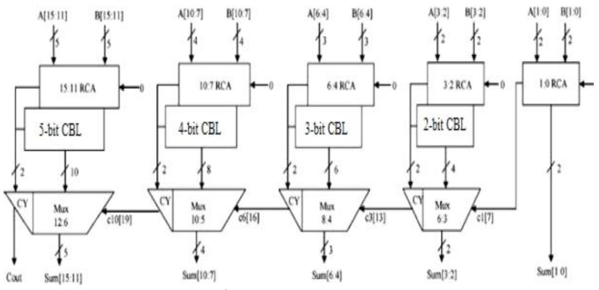


Figure 4.Modified CSLA using CBL logic

The Summation and carry signal for FA which has Cin=l, Generate by INV and OR gate. Through the multiplexer, we can select the correct output result according to the logic state of carry-in signal.

III. PROPOSED CSLA

D. Parallel prefix adders

The parallel prefix adders [7] are more flexible and are used to speed up the binary additions. Parallel prefix adders are obtained from Carry Look Ahead (CLA) structure. We use tree structure form to increase the speed [8] of arithmetic operation. Parallel prefix adders are fastest adders and these are used for high performance arithmetic circuits in industries. The construction of parallel prefix adder [9] involves three stages:

1. Pre- processing stage
2. Carry generation network
3. Post processing

1. Pre-possessing stage

In this stage we compute, generate and propagate signals to each pair of inputs A and B. These signals are given by the logic equations 1&2:

Pi=Ai xor Bi .... (1)
Gi=Ai and Bi .... (2)

2. Carry generation network

In this stage we compute carries corresponding to each bit. Execution of these operations is carried out in parallel [9]. After the computation of carries in parallel they are segmented into smaller pieces. It uses carry propagate and generate as intermediate signals which are given by the logic equations 3&4:

CPi:j=Pi:k+1 and Pk:j ....(3)
CGi:j=Gi:k+1 or (Pi:k+1 and Gk:j) ....(4)

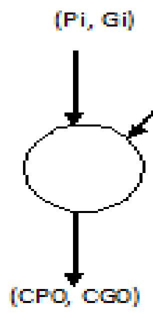


Figure5. Carry operator

3. *Post processing*

This is the final step to compute the summation of input bits. It is common for all adders and the sum bits are computed by logic equation 5&6:

Ci-1=(Pi and Cin) or Gi .... (4)

Si=Pi xor Ci-1 .... (5)

E. Kogge-Stone (KS) adder

Kogge-stone adder is a parallel prefix form of Carry Look-ahead Adder. Kogge-Stone adder can be represented as a parallel prefix graph consisting of carry operator nodes. The time required to generate carry signals in this prefix adder is *o*(log n). It is the fastest adder with focus on design time and is the common choice for high performance adders in industry. The Kogge-Stone adder concept was developed by Peter M. Kogge and Harold S. Stone [7], which was published in 1973. The better performance of Kogge-Stone adder is because of its minimum logic depth and bounded fan-out. On the other side it occupies large silicon area. The construction of 2, 3, 4, 5-bit Kogge-Stone adder are shown below.

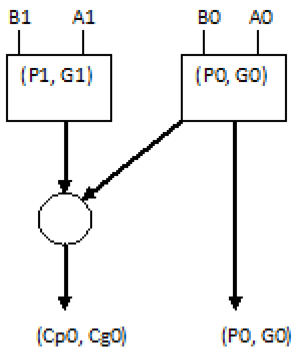


Figure 6. 2-bit KS Adder

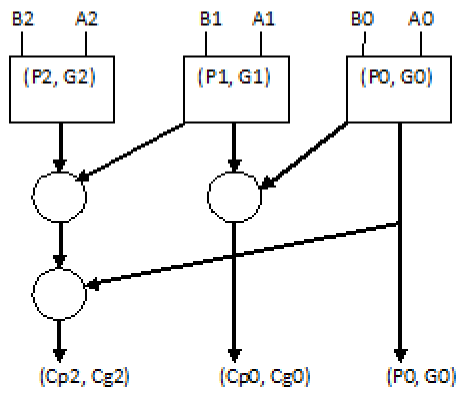


Figure7. 3-bit KS Adder

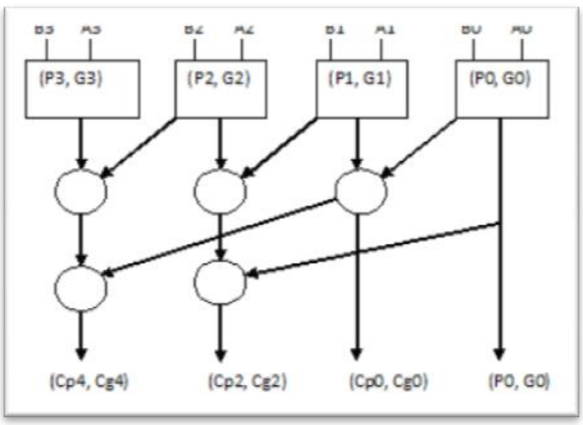


Figure8. 4-bit KS Adder

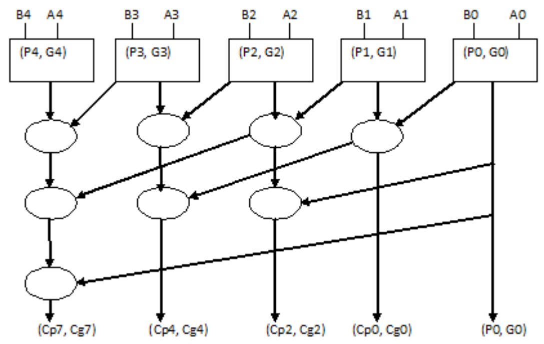


Figure 9. 5-bit KS Adder

In this proposed method modification is done by replacing the variable 2, 3, 4, 5-bit RCAs with 2, 3, 4, 5-bit Kogge-Stone adders. By using this logic we can reduce delay and area. The figure10 shows structure of modified CSLA using CBL logic.

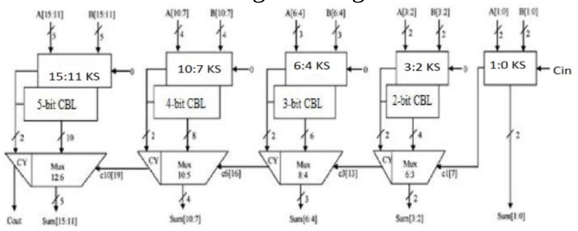
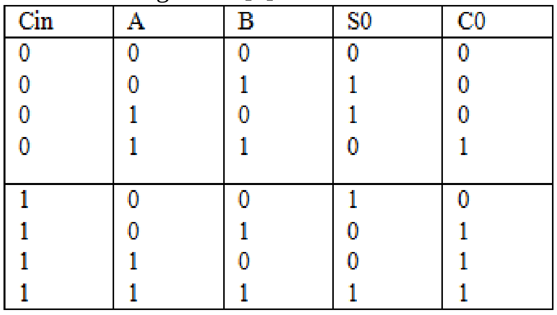


Figure10. Modified 16-b KS CSLA

F. Common Boolean logic

In proposed work, an area-efficient carry select adder by sharing the common Boolean logic term to remove the duplicated adder cells in the conventional carry select adder. In this way, it save many transistor counts and achieve a low Power. Through analyzing the truth table of a single-bit full adder, To find out that the output of summation signal as carry-in signal is logic "0" is the inverse signal of itself as carry-in signal is logic "1 ". As illustrated as S0 values in the truth table of Figure 11 [1].



Cin	A	B	S0	C0
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 11. The truth table of single-bit full-adder with common Boolean logic.

IV. COMPARISON AND SIMULATION RESULTS

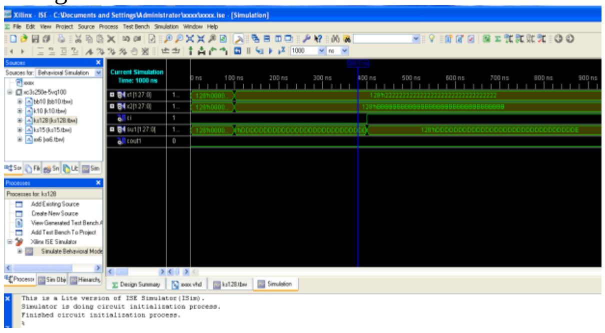
Table1. Logic and Route delay values of regular, Modified and proposed CSLA.

Word size	Adder	Max. Path delay (ns)	Logic delay (ns)	Route delay (ns)
16-bit CSLA	Regular (dual RCA)	13.88	9.39	4.48
	Modified (with BEC)	17.56	11.69	5.94
	Modified (with CBL)	14.47	9.78	4.69
	Proposed (KS with CBL)	13.83	9.17	4.66
32-bit CSLA	Regular (dual RCA)	23.63	14.51	9.12
	Modified (with BEC)	30.12	18.96	11.15
	Modified (with CBL)	25.64	16.40	9.24
	Proposed (KS with CBL)	22.43	13.45	8.98
64-bit CSLA	Regular (dual RCA)	42.56	24.74	17.81
	Modified (with BEC)	55.52	33.65	21.87
	Modified (with CBL)	46.58	28.53	18.04
	Proposed (KS with CBL)	39.28	22.02	17.26
128-bit CSLA	Regular (dual RCA)	80.50	45.21	35.18
	Modified (with BEC)	106.33	63.02	43.31
	Modified (with CBL)	88.44	52.78	35.65
	Proposed (KS with CBL)	72.97	39.15	33.81

The 8-bit CSLA is done by the same structure of 16- bit CSLA except group 4 and group 5. The 8th bit inputs are directly given to the full adder to complete the 8-bit sum and carry. The 32-bit CSLA is done by cascading two 16-bit CSLA and 64-bit CSLA is done by cascading two 32-bit CSLA [10] respectively.

When compared to the regular CSLA (RCA with BEC logic) and modified CSLA (RCA with CBL) the delay and area of proposed method (kogge-stone adder with CBL) decreased by 36% of regular CSLA and 18% of modified CSLA. Table 1 shows the comparison of regular CSLA, modified CSLA and proposed CSLA adders in terms of logic and route delay.

Various adders were designed using VHDL language in Xilinx ISE Navigator 10.1 and all the simulations are performed using Xilinx ISE simulator. The performance of proposed CSLA is analyzed and compared against the conventional CSLA designs. In this proposed CSLA architecture, the implementation code for 2, 3, 4, 5-bit Kogge-Stone adders were developed and corresponding values of logic and route delay were tracked. The simulated output of 128-bit proposed CSLA is shown in Figure 12.


Xilinx - ISE - C:\Documents and Settings\Administrator\xxxx\xxxx.ise - [Simulation]
File Edit View Project Source Process Test Bench Simulation Window Help
Sources
Sources for: Behavioral Simulation
Current Simulation Time: 1000 ns
x1[127:0]
x2[127:0]
ci
su1[127:0]
cout1
Processes
Add Existing Source
Create New Source
Add Test Bench To Project
Xilinx ISE Simulator
Design Summary
Simulation

```
This is a Lite version of ISE Simulator(ISim).
Simulator is doing circuit initialization process.
Finished circuit initialization process.
```

Figure12. Simulated Output of 128-b KS CSLA

The comparison between 16, 32, 64, 128-bit proposed, modified and regular CSLA are shown in Figure 13 in terms maximum path delay values.

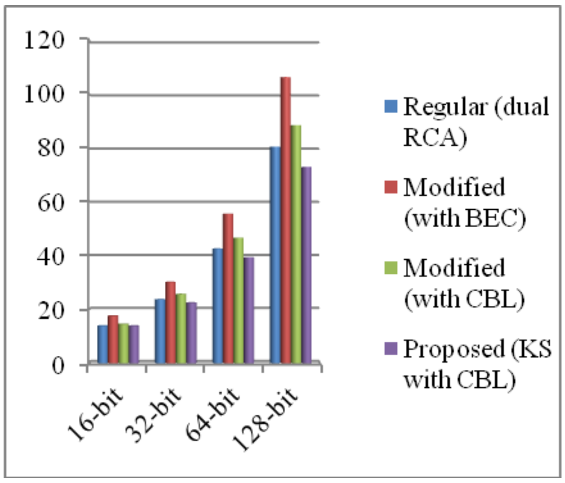


Figure13. Max. Path delay values comparison Between different adders

V. CONCLUSION

A new approach is proposed in this paper to reduce the delay of SQRT CSLA. The replacement of prefix adders and CBL logic in place of Ripple Carry Adders offers great advantage in the reduction of delay. The compared result shows that the proposed CSLA greatly reduces delay.

VI. ACKNOWLEDGEMENT

K. Prasad would like to thank Mrs. M. Bharathi, Assistant professor ECE Department who had been guiding throughout the project and supporting me in giving technical ideas about the paper and motivating me to complete the work efficiently and successfully.

REFERENCES

[1] Ms. S.Manju, Mr. V.Sornagopal” *An Efficient SQRT architecture of carry select adder designed Common Boolean logic*” IEEE transaction on very Large scale integration (VLSI) systems, Feb 2013

[2] B. Ramkumar, Harish M Kittur, “*Low – Power and Area-Efficient Carry Select Adder*”, IEEE transaction on very large scale integration (VLSI) systems, vol.20, no.2, pp.371-375, Feb 2012

[3] I-Chyn Wey, Cheng-Chen Ho, Yi-Sheng Lin, and ChienChanPeng”*An Area-Efficient Carry Select Adder Design by Sharing the Common Boolean Logic Term*” Proceeding on the international Multi Conference of eng. and computer scientist 2012, IMECS 2012

[4] J. M. Rabaey, “*Digital Integrated Circuits- A Design Perspective*”, New Jersey, Prentice-Hall, 2001

[5] O.J Bedrij “ *Carry select adder*,” IER transactions Electron. Computer, pp.340-344,1962

[6] Youngjoon Kim and Lee-Sup Kim, “*64-bit carry-select adder with reduced area*”, Electronics Letters, vol.37, issue 10, pp.614-615, May 2001

[7] M. Snir, “*Depth-size trade-offs for parallel prefix Computation,”* in Journal of Algorithms 7, pp.185– 201, 1986

[8] Belle W.Y.Wei and Clark D.Thompson, “*Area-Time Optimal Adder Design*”, IEEE transactions on Computers, vol.39, pp. 666-675, May1990

[9] Y. Choi, “*Parallel Prefix Adder Design*,” *Proc. 17th IEEE Symposium on Computer Arithmetic*, pp 90-98, 27th June 2005.

[10] Akhilesh Tyagi, “*A Reduced Area Scheme for Carry-Select Adders*”, IEEE International Conference on Computer design, pp.255-258, Sept 1990

K. Prasad completed his B.Tech in Electronics and Communication Engineering from Gokula Krishna College of Engg., Nellore, Andhra Pradesh, India in 2011. He is now pursuing his Master of Technology (M.Tech) in VLSI at Sree Vidyanikethan Engineering College, Tirupati, Andhra Pradesh, India. His interest includes Digital Design, ASIC Design, and VLSI Testing.

Mrs. M. Bharathi , M.Tech., is currently working as an Assistant Professor in ECE department of Sree Vidyanikethan Engineering College, Tirupati. Her research areas are VLSI System Design and Digital Design.