

EFFICIENT AND ENHANCED CARRY SELECT ADDER FOR MULTIPURPOSE APPLICATIONS

A. RAMESH

Asst. Professor,
E.C.E Department,
PSCMR CET, Kothapet,
Vijayawada, A.P,
India.

rameshavula99@gmail.com

B. Siva Nageswara Rao

Asst. Professor
E.C.E Department, Eswar
College of Engineering,
Narasaraopet, Guntur, A.P,
India.

bsnr2002@gmail.com

D. SATYANARAYANA

Asst. Professor
E.C.E Department, Eswar
College of Engineering,
Narasaraopet, Guntur, A.P,
India.

satyaniet@gmail.com

ABSTRACT

Carry Select Adder (CSLA) is one of the fastest adders use 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. Based on this modification 8 and 16 bit square-root CSLA (SQRT CSLA) architecture have been developed and compared with the regular SQRT CSLA architecture. The proposed design has reduced area and power as compared with the regular SQRT CSLA with only a slight increase in the delay.

Key Words: Adder, Carry select adder, Ripple carry adder, BEC.

INTRODUCTION

The challenge of the verifying a large design is growing exponentially. There is a need to define new methods that makes

functional verification easy. Several strategies in the recent years have been proposed to achieve good functional verification with less effort. Recent advancement towards this goal is methodologies. The methodology defines a skeleton over which one can add flesh and skin to their requirements to achieve functional verification. 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. 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 report is organized as two major portions; first part is brief introduction and history of the functional verification of regular Carry select adder which tells about different advantages of Carry select adder and RCA architecture and in this Regular model, there is a drawback and in order to overcome that

complexity, the modified architecture of CSLA has been designed. The importance of a fast, low-cost binary adder in a digital system is difficult to overestimate. Not only are adders used in every arithmetic operation, they are also needed for computing the physical address in virtually every memory fetch operation in most modern CPUs. Adders are also used in many other digital systems including telecommunications systems in places where a full-fledged CPU would be superfluous. Many styles of adders exist. Ripple adders are the smallest but also the slowest. More recently, carry-skip adders [1, 2, 3] are gaining popularity due to their high speed and relatively small size. Normally, in an N-bit carry-skip adder divided into a proper number of M-bit blocks [1, 4], a long-range carry signal starts at a generic block B_i , rippling through some bits in that block, then skips some blocks, and ends in a block B_j . If the carry does not end at the LSB of B_j then rippling occurs in that block and an additional delay is needed to compute the valid sum bits. Carry-look-ahead and carry-select adders [1] are very fast but far larger and consume much more power than ripple or carry-skip adders. Two of the fastest known addition circuits are the Lynch-Swartzlander's [5] and Kantabutra's [6]. Hybrid carry-look-ahead adders they are based on the usage of a carry tree that produces carries into appropriate bit positions without back propagation. In order to obtain the valid sum bits as soon as possible, in both Lynch-Swartzlander's and Kantabutra's adders the sum bits are computed by means of carry-select blocks, which are able to perform their

operations in parallel with the carry-tree. Thus, the aim of this project is to design a simple and efficient gate level modification to significantly reduce the area and power of the CSLA. Based on this modification, 8-, 16-, 32-, and 64-b square-root CSLA (SQRT CSLA) architecture have been developed and compared with the regular SQRT CSLA architecture. 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. 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 $C_{in}=0$ and $C_{in}=1$, 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 $C_{in}=1$ in the regular CSLA to achieve lower area and power consumption. The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure.

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. 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.

This adder plays a vital role in many data processing processors to perform fast arithmetic functions. Hence to resolve this issue, this adder has been developed to reduce the propagation delay for the carry to propagate to the next position. Now another important point here is the evaluation of the carry select adder is compared with the proposed design as it has a more balanced delay and requires lower power and area. Adders are commonly found in the critical path of many building blocks of microprocessors and digital signal processing chips. Adders are essential not only for addition, but also for subtraction, multiplication, and division. Addition is one of the fundamental arithmetic operations. A fast and accurate operation of a digital system is greatly influenced by the performance of the resident adders. The most important for measuring the quality of adder designs in the past were propagation delay, and area. The three most widely accepted metrics for measuring the Performance of a circuit are power, delay and area. Minimizing Area and delay has always been considered important, but Reducing power consumption has been gaining prominence Recently with the increasing level of device integration and the Growth in complexity of micro-electronic circuits, reduction of Power dissipation has come to fore as a primary design goal. While power efficiency has always been desirable in electronic Circuits, only recently has it become a limiting factor for a broad Range of applications, thereby requiring consideration early on in the design process.

IMPLEMENTAION

REGULAR CARRY SELECT ADDER

The Regular Carry Select Adder is represented in Figure 1.1. Basically this project is mainly targeted for data processing processors to perform fast arithmetic functions.

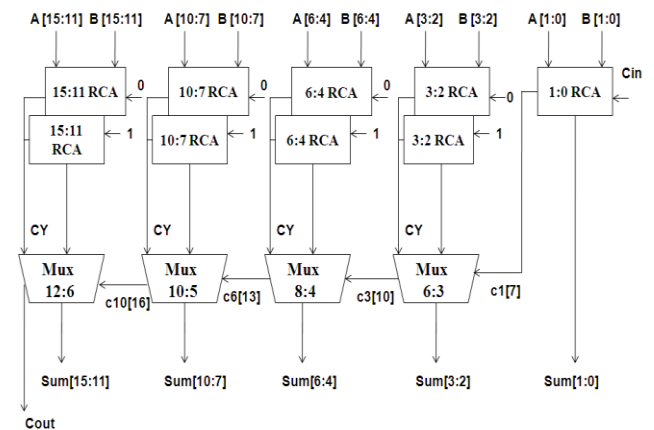


Fig 1.1: Regular Carry Select Adder

Addition is basic operation used in many data path logic systems such as Adders, Multipliers etc. Carry select adders are used for high speed operation by reducing the Carry propagation delay. The basic operation of Carry Select Adder (CSLA) is Parallel Computation. The carry-select adder (CSLA) provides a compromise between small area but longer delay ripple carry adder (RCA) and larger area with shorter delay carry look-ahead adder . CSLA uses multiple pairs of ripple carry adder (RCA) to generate partial sum and carry by considering carry input $C_{in}=0$ and $C_{in}=1$, then the final sum and carry are selected by multiplexers (MUX).

The modified carry select adder one RCA(Cin=1) is replaced by BEC. BEC design consists of AND, XOR and NOT gates as its structure. In this structure the XOR gate will be replaced by MUX with NOT gate. The least significant bit of the input is given to NOT gate and it is given as control signal to the MUX for the next input value. The LSB and the next immediate bit is provided as input to the AND gate where its corresponding output value is given as a control signal to the Multiplexer. Depending upon the control signal it will produce the sum value. If it is 0, then its output is same as the input otherwise it produces its complement value. It is continued till the end of MSB which is the proposed BEC design shown in below figure. The operation of XOR is same as that of the MUX with NOT gate. The proposed CSLA design reduces the area and power by replacing the gates in BEC design. The proposed CSLA consumes less power and area compared to the modified CSLA.

BEC:

The basic work is to use Binary to Excess-1 Converter (BEC) in the regular CSLA to achieve lower area and increased speed of operation. This logic is replaced in RCA with Cin=1. This logic can be implemented for different bits which are used in the modified design. The main advantage of this BEC logic comes from the fact that it uses lesser number of logic gates than the n-bit Full Adder (FA) structure. As stated above the main idea of this work is to use BEC instead of the RCA with Cin=1 in order to reduce the area and increase the speed of operation in the regular CSLA to obtain modified CSLA. To replace the n-bit RCA, n+ 1 bit BEC logic is required.

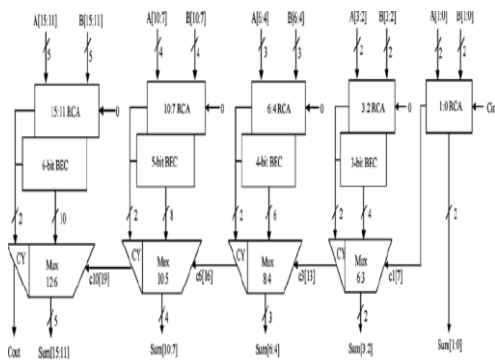


Figure: Proposed diagram

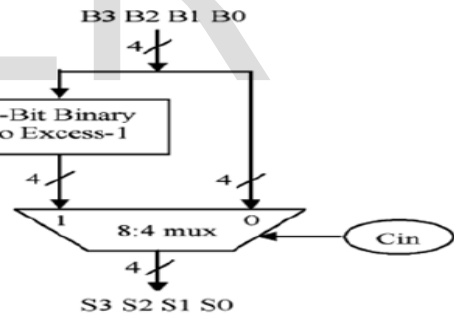


Fig: BEC with MUX

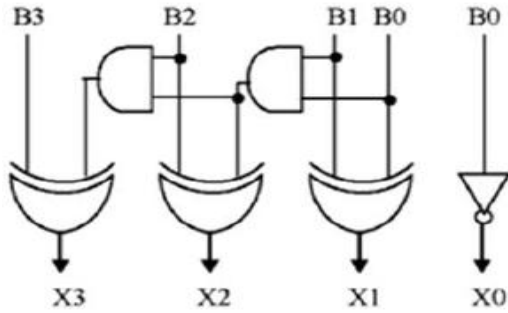


Fig: BEC internal architecture

RESULTS

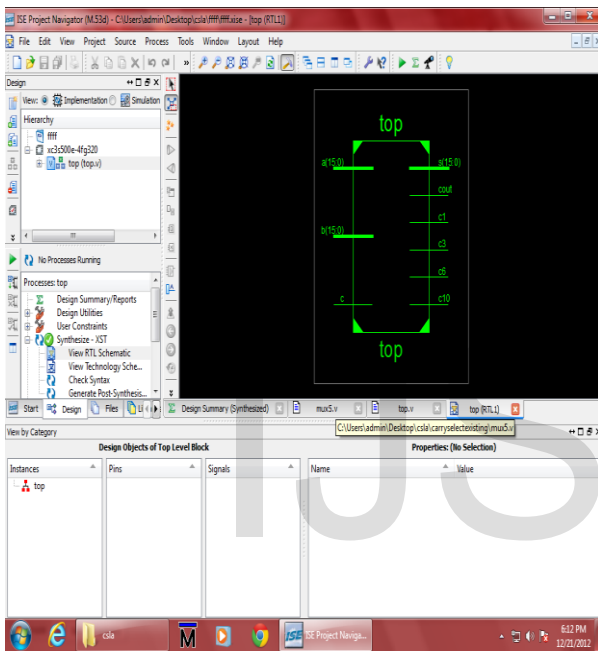
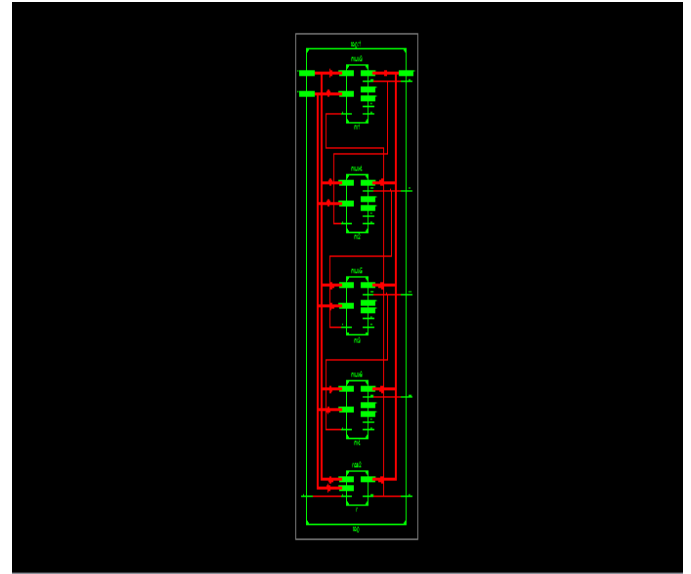


Fig 5.4: Top level schematic of Regular CSLA

The figure 5.4 shows the top level schematic of regular CSLA where the inputs are shown on the left side as a, b and c and the outputs sum s and carry C_{out} are shown on the right side of the schematic. Also, the intermediate outputs like c1, c3, c6 and c10 are shown on the right side of the figure.



The RTL schematic of regular CSLA is shown in the figure 5.5. The figure shows all the blocks like multiplexers, ripple carry adders and their internal connections.

POWER ANALYSIS

Finally, it can be seen from the figure 5.6, the total power consumed by the regular CSLA is 0.081 Watts.

Simulation results of Modified CSLA:

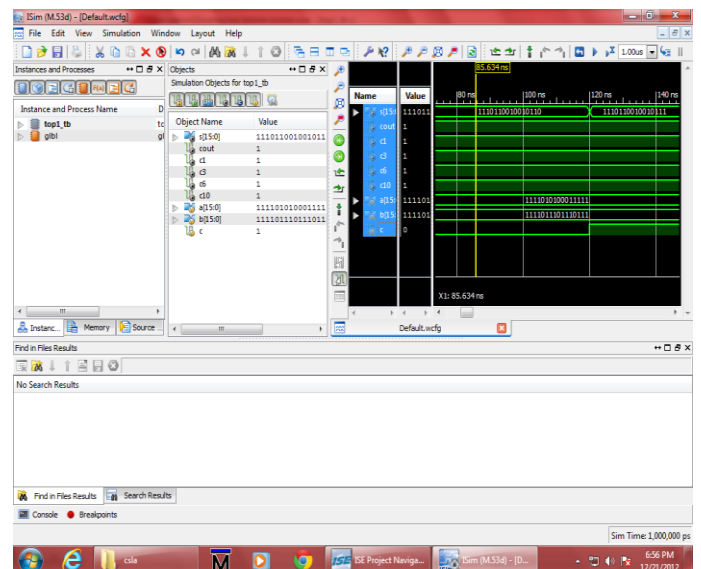


Fig 5.7: Simulation results of Modified CSLA

Here, the inputs given are
a=1111010100011111

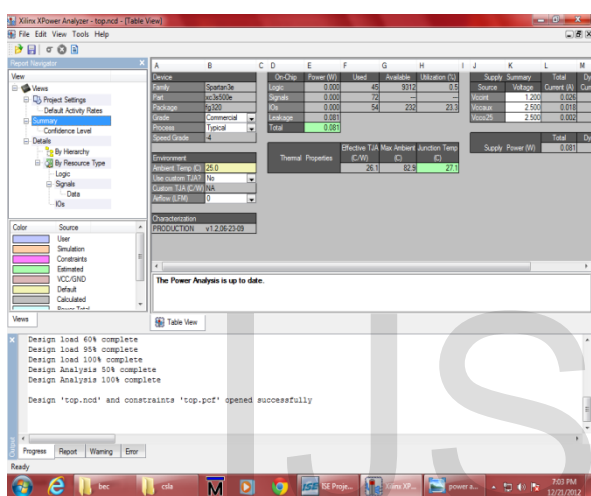
b=1111011101110111

carry c= 0

thus , the sum s= 1110110010010110

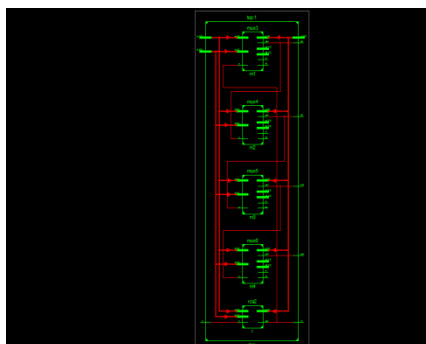
and carry outout=1

POWER ANALYSIS



Finally, it can be seen from the figure 5.10, the total power consumed by the modified CSLA is 0.081 Watts.

Fig 5.11: RTL schematic of Modified CSLA



Applications:

Image processing

In image processing with interpolation, an output of the gamma circuit and the input data are input to an adder circuit so as to obtain the added and averaged values at a predetermined ratio.

Signal processing

Addition is by far the most fundamental arithmetic operation. It has been ranked the most extensively used operation among a set of real-time digital signal processing benchmarks from application specific DSP to general purpose processors.

Arithmetic logic units

Carry select adder is used in arithmetic logic units to perform addition and multiplication in a less amount of time.

Advanced microprocessor design

In microprocessor design, the adder is used for the conversion mechanism in calculating the physical address using the offset address and segment address.

High speed multiplications

In multiplier, each bit of the Product P is obtained by a summation of bits A_iB_j using an array of single bit adders. The bits A_iB_j are formed using AND gates.

Advantages:

Low area:The modified Carry Select adder consumes less logic gates (low area) as it eliminates the pairs of Ripple carry adders.

Low power consumption:

In this design, it can be seen that the total power consumed by both the design is almost the same.

Disadvantages:

Increased delay: Even though there is reduction in area, a slight increase in delay can be seen in the modified CSLA.

CONCLUSION

A simple approach is proposed in this paper to reduce the area and power of SQR CSLA architecture. The reduced number of gates of this work offers the great advantage in the reduction of area and also the total power.

The compared results show that the modified SQR CSLA has a slightly larger delay (only 3.76%), but the area and power of the 64-bit modified SQR CSLA are significantly reduced by 17.4% and 15.4% respectively. The power delay product and also the area delay product of the proposed design show a decrease for 16-, 32-, and 64-bit sizes which indicates the success of the method and not a mere tradeoff of delay for power and area. The modified CSLA architecture is therefore low area, low power, simple and efficient for VLSI hardware implementation.

FUTURE SCOPE

This project uses System VHDL i.e., the technology used is direct test cases, randomized test cases, OVM for verification. Even though the coverage is 100%, there may be some errors which cannot be shown. So in order to overcome this, the new technology of System VHDL is OVM and UVM. In the coming future, the Router can be done by using OVM and UVM.

REFERENCES

Websites:

1. O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Comput., pp.340–344, 1962.
2. 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.
3. 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.
4. Y. He, C. H. Chang, and J. Gu, "An area efficient 64-bit square root carry-select adder for low power applications," in Proc. IEEE Int. Symp. Circuits Syst., 2005, vol. 4, pp. 4082–4085.
5. Cadence, "Encounter user guide," Version 6.2.4, March 2008.
6. www.uvmworld.org
7. www.testbench.in
8. www.ovmworld.org
9. www.systemVHDL.org

Books

1. VHDL HDL- Digital Design and Synthesis, by Samir Palnitkar
2. Writing test benches using system VHDL by Janick Bergeron
3. OVM Reference manual
4. System VHDL for verification by Chris Spear
5. UVM User guide
6. UVM Reference manual
7. Xilinx 13.1 Quick Start Tutorial

IJSER