Design of 8 Bit Hybrid Carry Save Adder

Akash Sankhe[1], Dhanush Shetty[2], Rushikesh Sawant[3], Sandhya Supalkar[4]

Dept. of Electronics & Telecommunication Engineering, Vidyavardhini's College of Engineering & Technology

[1]sankheakash100@gmail.com

[2]shettydhanush403@gmail.com

[3]rushi1013193543@gmail.com

[4]sandhya.supalkar@vcet.edu.in

***Abstract*— Adder is a digital circuit that performs the basic function of addition. It is an essential part of every digital circuit. It is used in computers to compute addresses and in ALUs. The most time critical application of adder is in Digital Signal Processing, where it is used to compute Fast Fourier Transforms, multiplier outputs, etc. This often involves three or more inputs to the adder. One of the adders capable of computing 3 or more inputs is the Carry Save Adder (CSA). The main flaw of conventional Carry Save Adder is that it uses Ripple Carry Adder (RCA) in its last stage to compute the output, thus making it slow. In this paper a new, hybrid design of Carry Save Adder is proposed, which employs Carry Look Ahead Adder (CLA) in the last stage thus making carry generation much faster and improving overall delay of the Carry Save Adder.**

***Keywords*— Adder, Carry Save Adder (CSA), Ripple Carry Adder (RCA), Carry Look Ahead Adder (CLA), Hybrid Adder.**

I. Introduction

In today's VLSI circuit design industry, high speed and low power consumption is of key importance. One of the important digital VLSI circuits is adder. An adder is a basic calculator used to find the sum of 2 binary numbers. Adders are used in processors, ALUs to calculate memory addresses and other similar operations. There are two types of basic adders:

A. Half Adder

Half Adder is an electronic circuit that performs addition of two numbers. In half adder, addition of two single bit binary digits takes place and sum and carry outputs are generated. Half adder is a combinational logic circuit which is implemented by using one XOR gate and one AND gate as shown in Fig. 1.1. The sum is obtained from XOR gate output and carry is obtained from AND gate output.

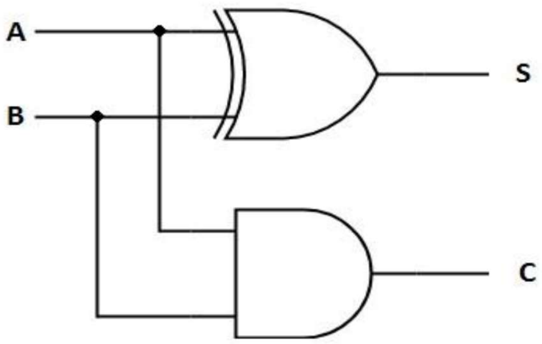


Fig. 1.1. Gate Level Design of Half Adder

B. Full Adder

Full Adder is a circuit that performs addition of two binary numbers and a carry input. If a carry is generated from the previous stage the carry input is logic 1 and if carry is not generated from the previous stage then carry is logic 0. Full adder, as shown in Fig. 1.2, is designed by using two XOR gates, two AND gates and one OR gate. Full adder circuit has three inputs and generates two outputs.

ISBN: 978-93-5659-047-2

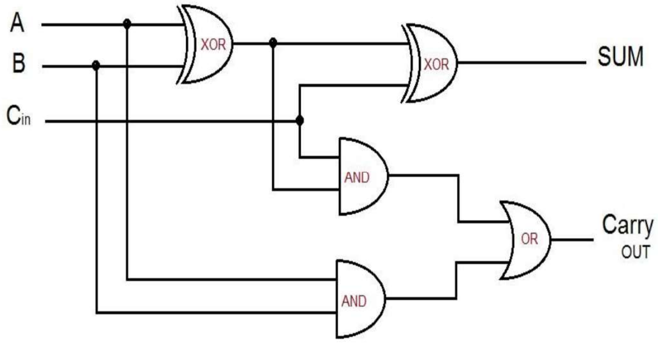


Fig. 1.2. Gate Level Design of Full Adder

Full adders and half adders can be used in digital circuits but only to some extent. In most ALUs, addition of multiple binary inputs takes place and we require parallel adders like Carry Save Adder (CSA), Carry Skip Adder (CSkA), Carry Select Adder (CSlA), Carry Look Ahead Adder (CLA), etc. to meet the timing requirements. An adder circuit implemented using the above mentioned adders will perform much faster than addition performed using half and full adders.

C. Carry Save Adder

Carry Save Adder (CSA), as shown in Fig. 1.3, is one of the high speed, multi-input digital adder circuits. A conventional CSA uses 16 full adders (8 in the first stage and 8 in the second stage) to compute 9-bit sum and carry outputs. Each of the 8 full adders in the first stage computes a single sum and carry bit based solely on the corresponding bits for the three inputs; it produces partial sum and shift carry; these outputs act as inputs to the 8 full adders in the second stage (which make up the ripple carry adder) which then compute the final sum and carry outputs.

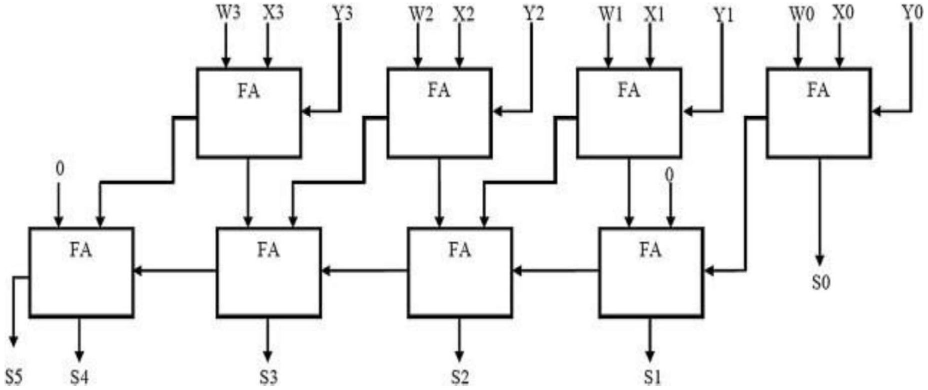


Fig. 1.3. Conventional Carry Save Adder Block Diagram

CSA was designed to avoid rippling of carry but its use of RCA defeats this purpose. The use of RCA reduces the overall performance of the CSA. Hence in this paper a new hybrid design is proposed which replaces the RCA with CLA. CLA uses the concept of generation and propagation of carries. The proposed

ISBN: 978-93-5659-047-2

hybrid adder will be designed and simulated in Xilinx ISE 14.7 for the Spartan-3 target board with a speed grade of -4.

II. Literature Review

Adders form a critical hardware unit of ALU and all its complex adder architecture are based on its basic building blocks such as half adder and full adder. In paper [1], a multiplexer is used to propagate the carry in full adder, and other adders are implemented using multiplexed full adder. The authors conclude that proposed adder topologies performed better than their conventional counterparts in area used, power distribution, delay and gate count. Even while using conventional full adders, CSA has the least area usage as compared to CBA, CSkA and CSlA.

As the demand for high performance in the VLSI industry is increasing, research has been done to improve the delay of basic adders. In paper [2] & [4], authors did a comparative study of various parallel adders. A hybrid adder circuit design consisting of CSA and CSkA to improve the propagation delay was proposed in paper [2]. The proposed hybrid design performs better than conventional CSA for higher bit operations but the implementation of CSkA takes more area than the CSA itself. So if area is the main design constraint then this type of adder design cannot be implemented.

A 16-bit area efficient RCA and CLA was implemented, based on the theory of reversible logic, by the authors in paper [3]. The design was optimized in terms of delay and hardware complexity and the authors establish that carry look ahead adders are the best among all the designs.

The authors in paper [5] design and compare the performance of various adders on the basis of LUT's, Slices occupied and delay. It was observed that CLA is having better performance in terms of area (slices occupied) and CSA is having better performance in terms of delay.

In paper [6] the authors designed and simulated RCA and CLA in HSPICE. They compared both the adders and concluded that if performance is the key concern then CLA is more suited architecture and if the adder is expecting a large number of computations then CLA is the way to go due to the lower dynamic power dissipation.

III. Proposed Hybrid Adder Design

In the proposed hybrid adder design shown in Fig. 3.1, CSA and CLA are used. In the first stage the normal carry save addition concept is used and for the final stage the ripple carry adder is replaced with a carry look ahead adder. Using the CLA in the final stage of CSA causes the delay to decrease significantly. Each full adder used in the first stage of CSA has two outputs: partial sum and carry. This partial sum and carry is passed on to the CLA which is used to generate the final sum and carry. In this method there is no delay introduced when calculating the sum and carry because CLA does not wait for the carry to be generated by the previous stage; instead it uses carry generation and propagation logic to generate the outputs.

ISBN: 978-93-5659-047-2

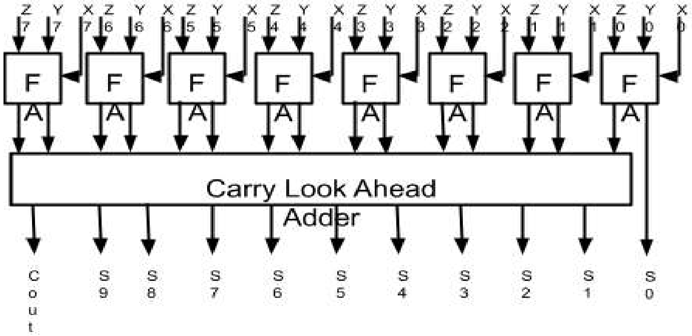


Fig. 3.1. Proposed Carry Save Adder Block Diagram

IV. Implementation

The first stage of the 8-bit hybrid CSA adder uses 8 full adders. The working algorithm for full adders is shown in (1) and (2). The second stage of the hybrid CSA adder uses CLA which uses a carry generation and propagation algorithm as shown in (3), (4), (5) and (6). This algorithm is used for generation of carry outputs (Ci+1) and sum outputs (Si) without needing to wait for the previous carry.

$$Sum = Ai \oplus Bi \oplus Ci \tag{1}$$

$$Carry = A*B + B*C + A*C \tag{2}$$

$$Pi = Ai \oplus Bi \tag{3}$$

$$Gi = Ai * Bi \tag{4}$$

$$Si = Pi \oplus Ci \tag{5}$$

$$Ci+1 = Gi + Pi * Ci \tag{6}$$

Fig. 4.3 and Fig. 4.4 shows the technology schematic, this displays the top block which shows inputs and outputs.

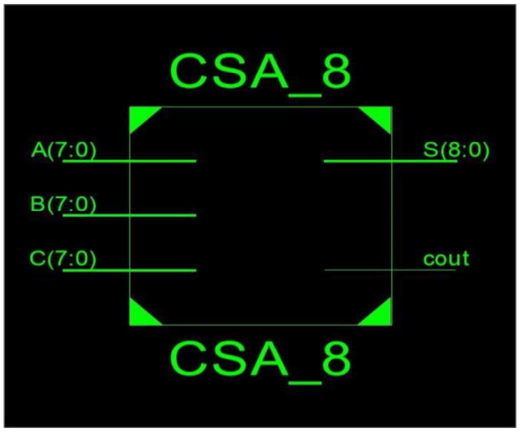


Fig. 4.3. Technology Schematic of 8-bit Conventional Carry Save Adder

ISBN: 978-93-5659-047-2

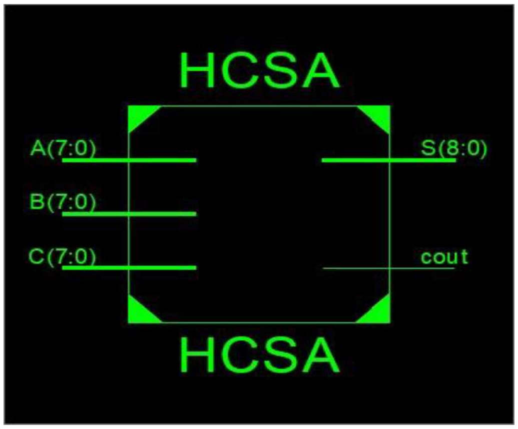


Fig. 4.4. Technology Schematic of 8-bit Hybrid Carry Save Adder

Fig. 4.5 and Fig. 4.6 shows the Design Implementation schematic which displays the hardware implementation. In Fig. 4.5, each rectangular block is a full adder module and there are 16 full adder blocks that make up the conventional carry save adder. In Fig. 4.6, each rectangular block on the leftmost and rightmost side is a full adder module and the center block contains carry look ahead logic. Thus the 8 full adder blocks on either side and a carry look ahead adder block in the center make up the hybrid carry save adder.

The Fig. 4.7 shows the final delay and power optimized 8-bit hybrid CSA design. The final hybrid design is optimized to reduce logic delay, route delay and also to reduce the power consumed by the hybrid design.

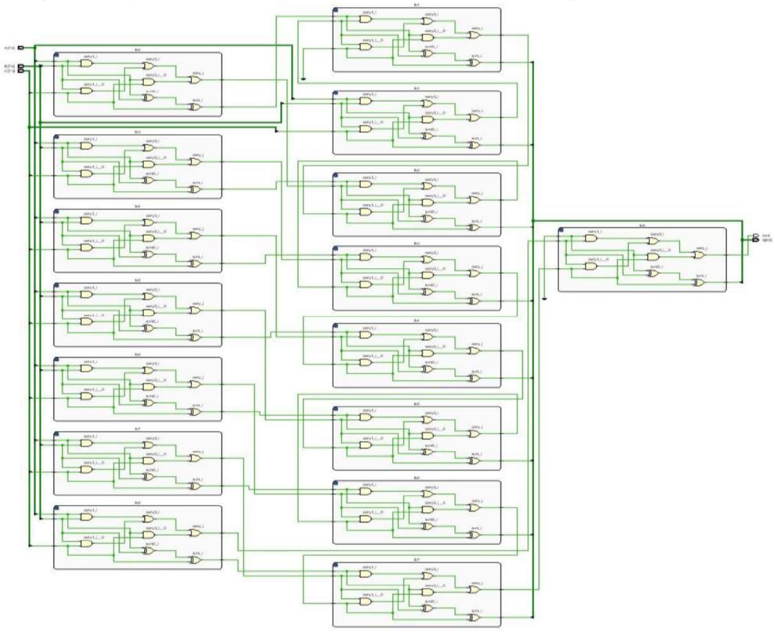


Fig. 4.5. Design Implementation of 8-bit Conventional Carry Save Adder

ISBN: 978-93-5659-047-2

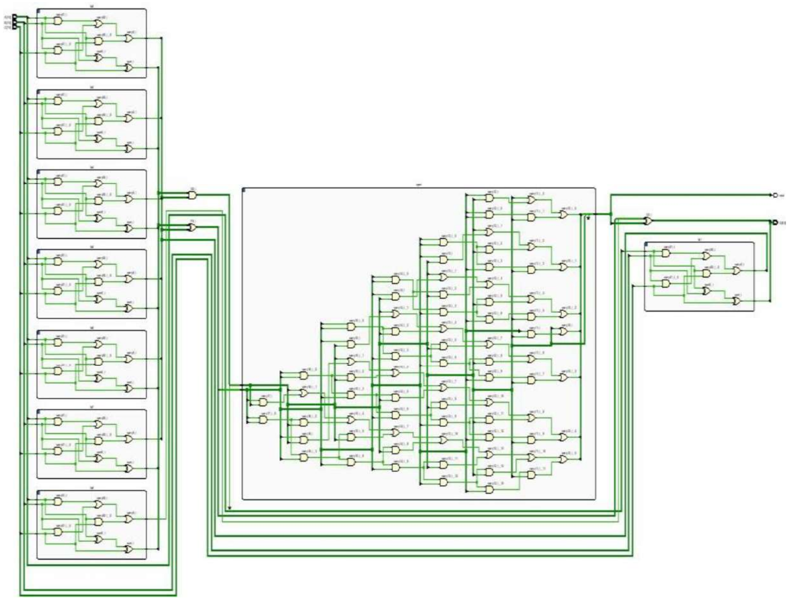


Fig. 4.6. Design Implementation of 8-bit Hybrid Carry Save Adder

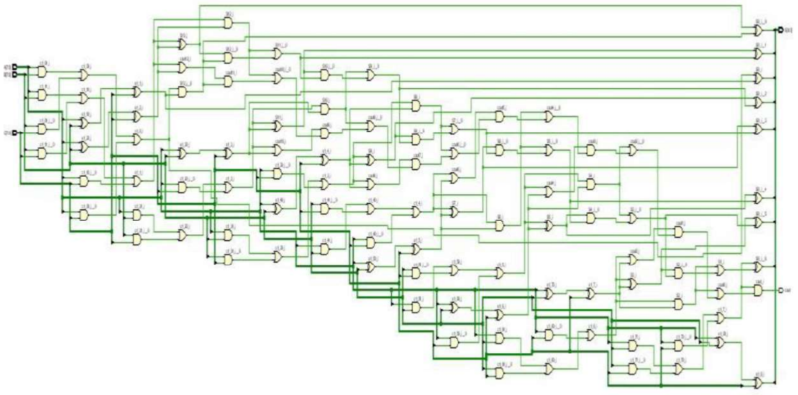
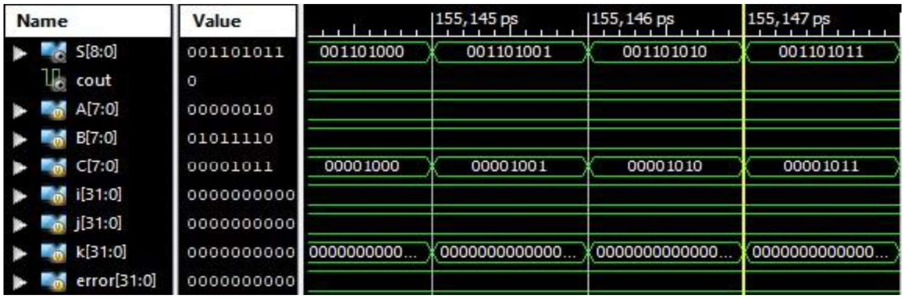


Fig. 4.7. Design Implementation of Final, Optimized, 8-bit Hybrid Carry Save Adder

V. Results

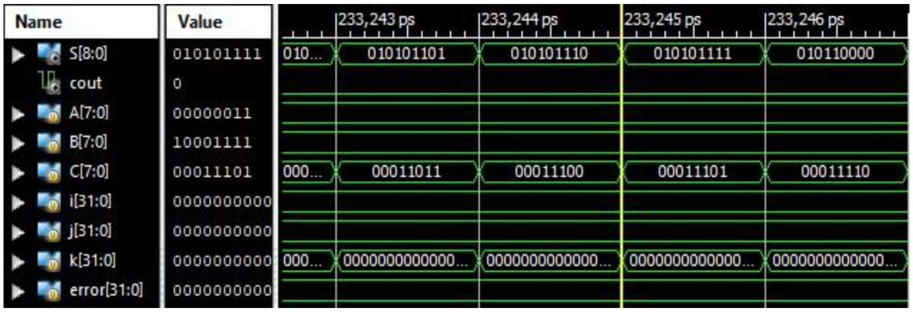
The hybrid design was simulated and the results are generated by writing a test bench program which tests all the possible input combinations and verifies the obtained result. If there are any errors then they are recorded in the signal named error, and if the adder is performing as it should then the error signal remains at 0.

Fig. 5.1 and Fig 5.2 show the simulation results of 8-bit conventional CSA and 8-bit hybrid CSA respectively. Here A,B and C are input signals, S and cout are output signals. The variables i, j, k and error are used for verification of the output signals.



Name	Value
S[8:0]	001101011
cout	0
A[7:0]	00000010
B[7:0]	01011110
C[7:0]	00001011
i[31:0]	0000000000
j[31:0]	0000000000
k[31:0]	0000000000
error[31:0]	0000000000

Fig. 5.1. Simulation Result of 8-bit Conventional Carry Save Adder



Name	Value
S[8:0]	010101111
cout	0
A[7:0]	00000011
B[7:0]	10001111
C[7:0]	00011101
i[31:0]	0000000000
j[31:0]	0000000000
k[31:0]	0000000000
error[31:0]	0000000000

Fig. 5.2. Simulation Result of 8-bit Hybrid Carry Save Adder

The results for timing (maximum combinational path delay), area occupied and power consumed by Conventional CSA and Hybrid CSA are shown in both tabular form, in TABLE 1, TABLE 2 and TABLE 3 and in bar graph form, in Fig. 5.3, Fig. 5.4 and Fig 5.5 respectively. The power figures are obtained by using the Xilinx Xpower Analyzer.

TABLE I

TIMING COMPARISON TABLE OF CONVENTIONAL CSA AND HYBRID CSA

Adder	Total Delay	Logic Delay	Route Delay
Conventional CSA	18.3621 nsec	9.456 nsec	8.905 nsec
Hybrid CSA	15.923 nsec	8.977 nsec	6.946 nsec

TABLE II

AREA COMPARISON TABLE OF CONVENTIONAL CSA AND HYBRID CSA

Adder	Slices Occupied	LUTs Used
Conventional CSA	17	30
Hybrid CSA	19	33

TABLE III

POWER COMPARISON TABLE OF CONVENTIONAL CSA AND HYBRID CSA

Adder	Total Power (Logic Power + Signal Power)	Logic Power	Signal Power
Conventional CSA	970 μW	110 μW	860 μW
Hybrid CSA	900 μW	220 μW	680 μW

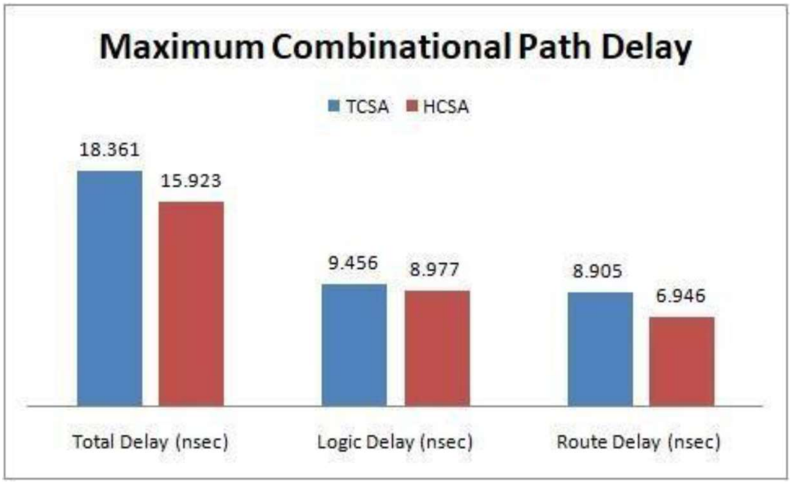


Fig. 5.3. Bar Graph Showing Total Delay, Logic Delay and Route Delay for Conventional CSA (TCSA) & Hybrid CSA (HCSA)

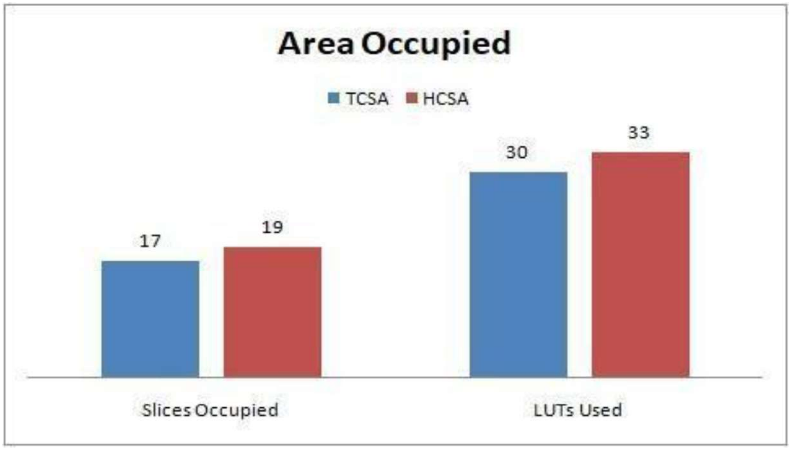


Fig. 5.4. Bar Graph Showing Total Slices Occupied and LUTs Used for Conventional CSA (TCSA) & Hybrid CSA (HCSA)

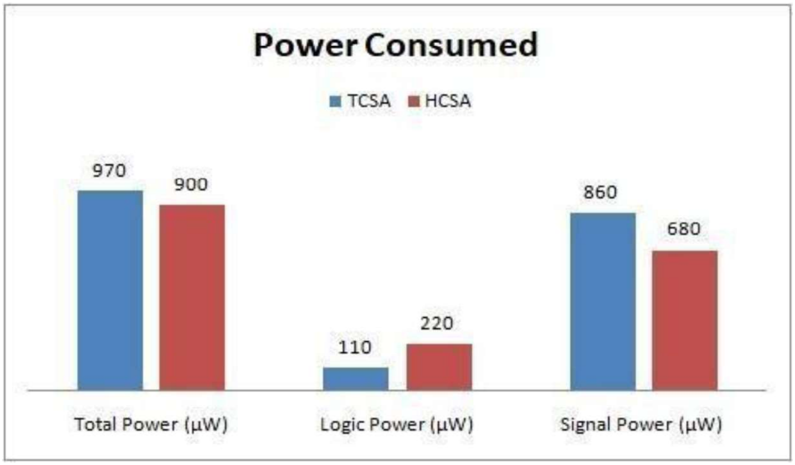


Fig. 5.5. Bar Graph Showing Total Power, Logic Power and Signal Power for Conventional CSA (TCSA) & Hybrid CSA (HCSA)

VI.Conclusion

A hybrid adder design is proposed in this paper. The proposed design combines the advantages of carry save adder and carry look ahead adder. The proposed design is 13.28% faster than its conventional

ISBN: 978-93-5659-047-2

counterpart while using 7.22 % less power. This is possible by replacing the ripple carry adder in conventional carry save adder with carry look ahead adder and delay optimizing the final design. Thus this hybrid design can be used in time critical applications like in DSPs to calculate FFTs and other time crucial applications. This, ofcourse, comes at the cost of some added area.

Acknowledgment

It is our privilege to express our sincere gratitude to our project guide, Prof. Sandhya Supalkar and all the teaching as well as non-teaching staff for their inputs, guidance, cooperation and constructive criticism throughout the duration of our project. We would like to extend our gratitude to Principal Dr. H. V. Vankudre and HOD Dr. Vikas Gupta for their moral support and encouragement. We would also like to thank Vidyavardhini's College of Engineering & Technology and Department of Electronics & Telecommunication for providing us with an ideal learning environment and facilities to undertake this project.

References

[1] D. Mohanapriya, Dr. N.Saravanakumar "A Comparative Analysis of Different 32-bit Adder Topologies with Multiplexer Based Full Adder", International Journal of Engineering Science and Computing, Vol.6 Issue 5, 2016.

[2] Sujan Sarkar, Jishan Mehedi "Design of Hybrid (CSA-CSkA) Adder for Improvement of Propagation Delay", Third International Conference on Research in Computational Intelligence and Communication Networks, 2017.

[3] Samrat Kumar Singh, Deepak Kumar "Design and Implementation of Low Delay Look Ahead and Ripple Adder", International Journal for Scientific Research & Development, Vol.5 Issue 8, 2017.

[4] Jasbir Kaur, Lalit Sood "Comparison Between Various Types of Adder Topologies", International Journal of Computer Science and Telecommunications, Vol.6 Issue 1, 2015.

[5] Tsung-Han Sher, Shermin Arab "Comparison Between Ripple-Carry Adder and Carry-Look Ahead Adder", University of Southern California, 2015.

[6] Maroju Saikumar, Dr. P. Samundiswary "Performance Analysis of Various Adders Using Verilog", International Journal of Computer Science and Mobile Computing, Vol.2 Issue 9, 2013.

ISBN: 978-93-5659-047-2