International Journal of Emerging Engineering Research and Technology
Volume 2, Issue 6, September 2014, PP 187-194
ISSN 2349-4395 (Print) & ISSN 2349-4409 (Online)

A Novel 8-bit Carry Select Adder using 180nm CMOS Process Technology

Yogita Hiremath

PG Student, Dept. of ECE
Dr. Ambedkar Institute of Technology, Bangalore
Karnataka, India
yogita.1491@gmail.com

Abstract: *This paper presents an efficient high-sped 8-bit carry select adder. The adder is designed and implemented using 180nm CMOS process technology. The proposed adder provides a good compromise between cost and performance in carry propagation adder design. It decreases the computational time compared to ripple carry adder and thus increases the speed. The carry select adder consists of 4-bit ripple carry adders and an array of 2:1 multiplexers. The carry is selected through the multiplexer. The layout of the design is efficiently optimized in terms of area using CMOS 180nm technology micron rules. The performance of the adder and its blocks is analysed in terms of different performance parameters. According to the estimations done, the transistor count, propagation delay and power consumption of the adder was found to be 246, 340ps and 13.77µW.*

Keywords: *Adder, cadence, carry select adder (CSA), delay, power consumption, ripple carry adder (RCA)*

1. INTRODUCTION

Addition is the heart of computer arithmetic and the arithmetic unit is often the work house of a computational circuit. There are many ways to design adder. The ripple carry adder (RCA) provides the most compact design but takes longer computing time. If there is N-bit RCA, the delay is linearly proportional to N. Thus for large values of N the RCA gives highest delay of all the adders. The carry look ahead adder (CLA) gives fast results but consumes large area. If there is N-bit adder, CLA is fast for $N \leq 4$, but for large values of N its delay increases more than other adders.

The problem of the ripple-carry adder is that each adder has to wait for the arrival of its carry-input signal before the actual addition can start. The basic idea of the carry select adder [1] is to use blocks of two ripple carry adders, one of which is fed with a constant 0 carry-in while the other is fed with a constant 1 carry-in. Thus, both blocks can calculate in parallel. When the actual carry-in signal for the block arrives, multiplexers are used to select the correct one of both precalculated partial sums. Also, the resulting carry-out is selected and propagated to the next carry select block. Hence, the time for the implementation of carry select adder is expressed in equation 1, where

$t_{RCAcarry}$ is the delay for the carryout of a full adder

t_{RCAsum} is the delay for the sum of a full adder

$$\text{Propagation Delay } (t_{RCAprop}) = (N-1) \times (t_{RCAcarry} + t_{RCAsum}) \quad (1)$$

From equation 1, we can see that the delay is proportional to the length of the adder. An example of a worst case propagation delay input pattern for a 4 bit ripple-carry adder is where the input operands change from 1111 and 0000 to 1111 and 0001, resulting in a sum changing from 01111 to 10000.

The proposed carry select adder is implemented using Cadence EDA tool [2]. The tool provides sophisticated features such as Cadence Virtuoso Schematic Editor which provides sophisticated capabilities which speed and ease the design, Cadence Virtuoso Visualization and Analysis which

efficiently analyzes the performance of the design and Cadence Virtuoso Layout Suite that speeds up the physical layout of the design.

In this paper, we propose a design and implementation of 8-bit carry select adder. The paper is organized as follows: in section 2, basics of carry select adder is presented. Subsequently, in section 3, the implementation of the adder is presented. In section 4, the schematic and layout of the adder is presented. In section 5, the simulation results are given and discussed. The evaluation of performance parameters for the proposed design is carried out. Finally a conclusion will be made in the last section.

2. BASICS OF CSA

Carry select adders use multiple narrow adders to create fast wide adders used in many data processing processors to perform fast arithmetic operations. Consider the addition of two n-bit numbers with a=a(n-1)……..a(0) and b=b(n-1)……..b(0). At the bit level, the adder delay increases from the least significant 0^{th} position upward, with the $(n-1)^{th}$ requiring the most complex logic. A carry select adder breaks the addition problem into smaller groups.

For example, we can split the n-bit problem into two (n/2)-bit-selections, then give special attention to the higher order group that adds the word segments a(n-1)……a(n/2) and b(n-1)…….b(n/2). The carry delay will then center around the carry-out bit c(n/2) produced by the sum of lower order word segments a(n/2-1)…….a(0) and b(n/2-1)……b(0). We know that there are only two possibilities for the carry bit, c(n/2)=0 or c(n/2)=1.A carry-select adder provides two separate adders for the upper words, one for each possibility. A multiplexer is then used to select the valid result.

3. IMPLEMENTATION OF 8-BIT CSA

The 8-bit carry select adder [2] is implemented as shown in Figure 1.The adder is split into two 4-bit groups. The lower order bits a3 a2 a1 a0 and b3 b2 b1 b0 are fed into the 4-bit adder L to produce the sum bits s3 s2 s1 s0 and a carry-out bit c4. The higher order bits a7 a6 a5 a4 and b7 b6 b5 b4 are used as inputs to two 4-bit adders.

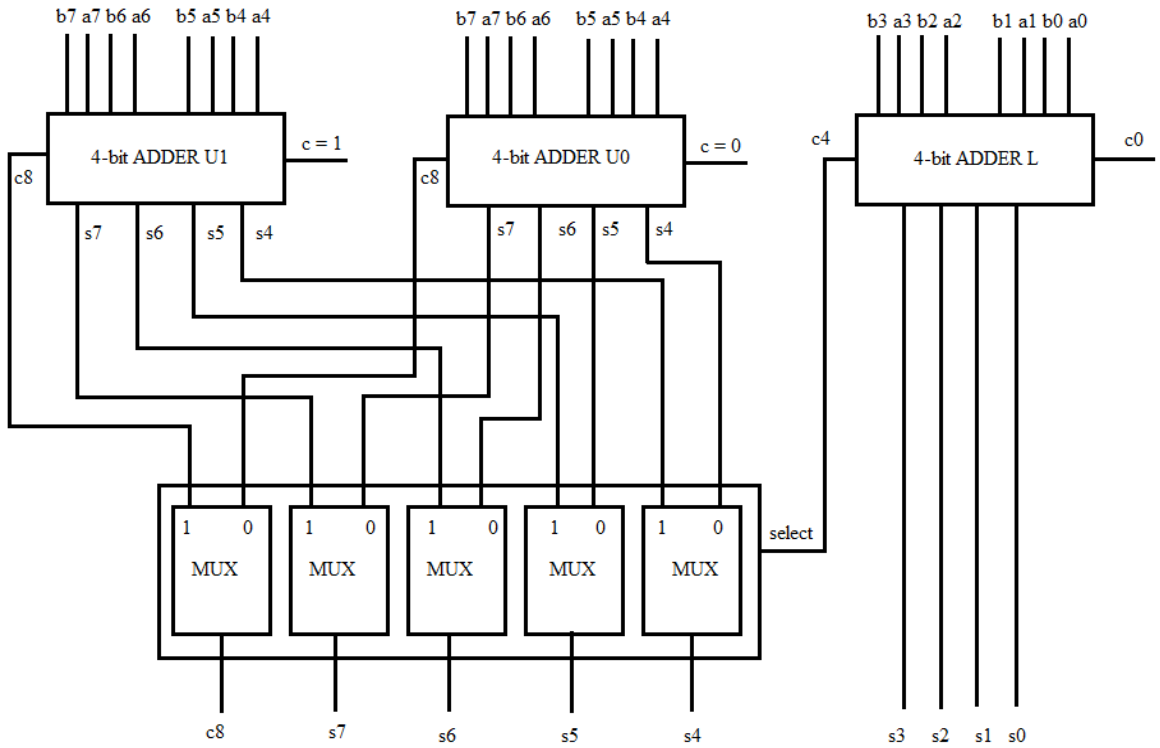

Fig1. *8-bit Carry Select Adder*

Adder U0 calculates the sum with a carry-in of c=0, while adder U1 calculates with a carry-in of c =1. Both sets of results are used as inputs to an array of 2:1 multiplexers. The carry bit from the c4 of adder L is used as the MUX select signal. If c4 = 0, then the results of U0 are sent to the output, while a value of c4 =1 selects the results of U1 for s7 s6 s5 s4. The carry-out bit c8 is also selected by the MUX array.

3.1. Implementation of 4-Bit Adder

The 4-bit adder block used in CSA is ripple carry adder. In ripple carry adder each carry bit from a full adder "ripples" to the next full adder [3]. The simple implementation of 4-bit ripple carry adder is shown below in Figure 2. C0 is the input carry, x3 x2 x1 x0 and y3 y2 y1 y0 represents two 4-bit input binary numbers.C4 is the output carry and s3 s2 s1 s0 is the sum output.

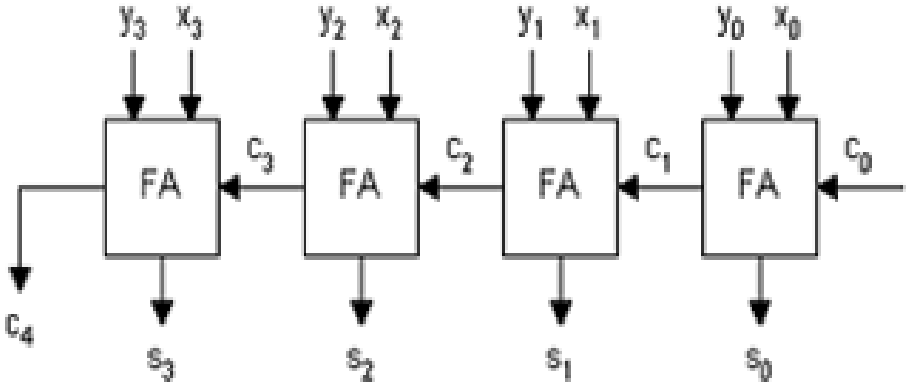

Fig2. *4-bit Ripple Carry Adder*

The ripple carry adder is designed using a full adder cell with 18-transisitors based on transmission gate logic [4]. The full adder is constructed using an XOR gate and two 2:1 multiplexers as shown in Figure 3.The SUM (A xor B xor Cin) is formed by a multiplexer controlled by A xor B (and complement). Examining the adder truth table reveals that when A xor B is true, COUT=C and SUM=complement of C. When A xor B is false, COUT=A (or B) and SUM=C.

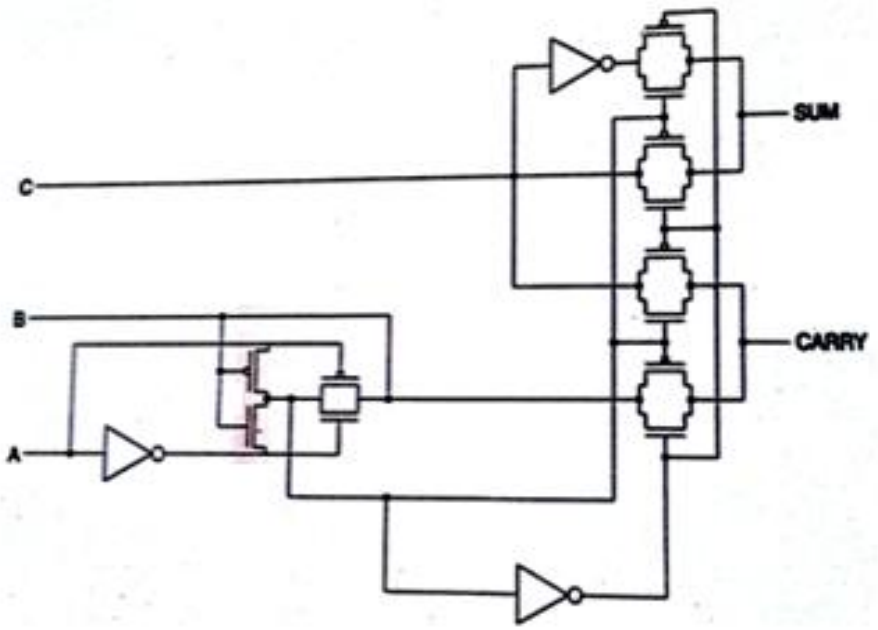

Fig3. *Full Adder Cell*

3.2. Implementation of 2:1 Multiplexer

In this design of multiplexer, two transmission gates are used as shown in Figure 4. The transmission gates select input A or B on the basis of the value of the control signal S. When S=0, Y=A and when S=1, Y=B.

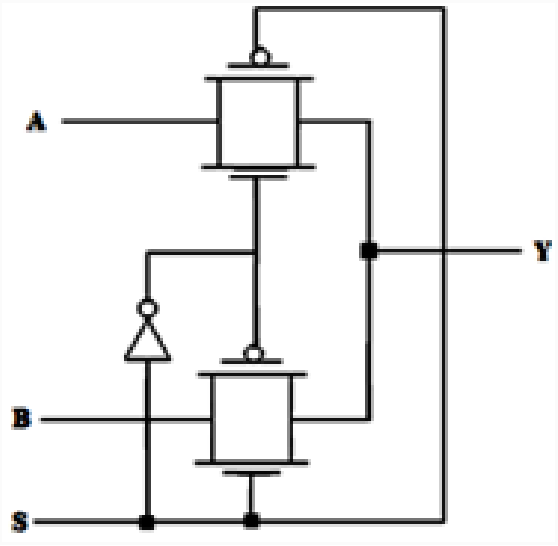

Fig4. *2:1 Multiplexer*

4. SCHEMATIC AND LAYOUT

The transistor-level diagram of each block of CSA is implemented in Cadence Virtuoso schematic editor [5]. Subsequently, the instances of all the individual blocks are integrated to form the schematic of the complete CSA circuit. The optimized layout is designed following the micron rules of 180nm CMOS process technology in Cadence Virtuoso Layout Suite. The sea of gate array concept is used to design the layouts.

First the schematic is designed and is simulated creating its test schematic. If the schematic satisfies the requirements, the layout is designed following the micron design rules. Then the layout versus schematic match is carried out. The layout is efficiently designed in terms of area by keeping minimum distance between different layers according to the micron rules specified for 180nm CMOS process technology.

4.1 4-bit Adder

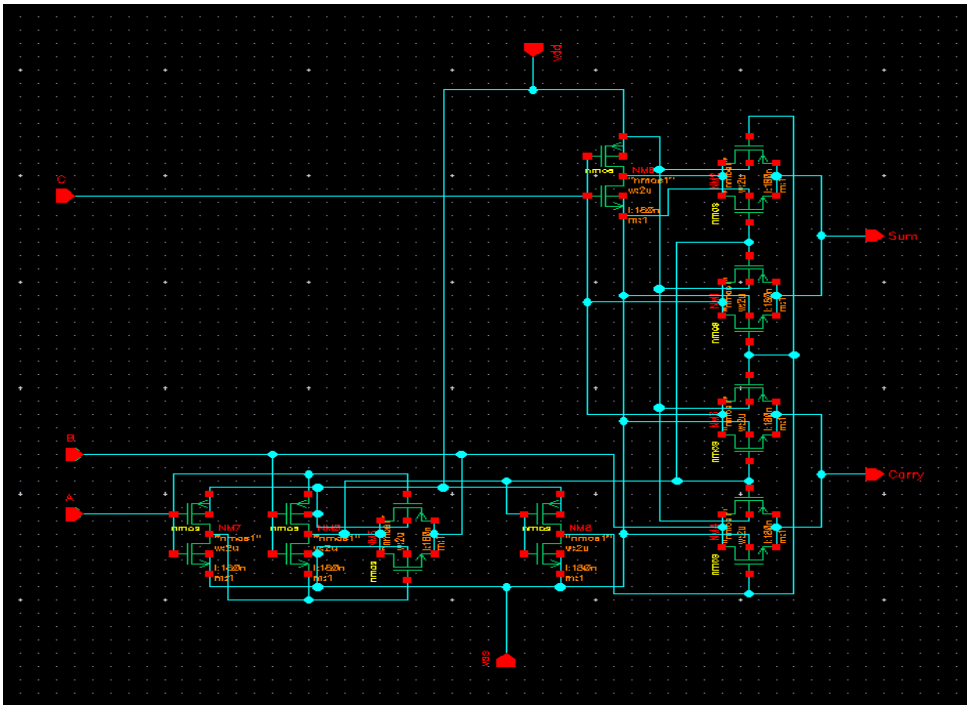

Fig5. *Schematic of Full Adder*

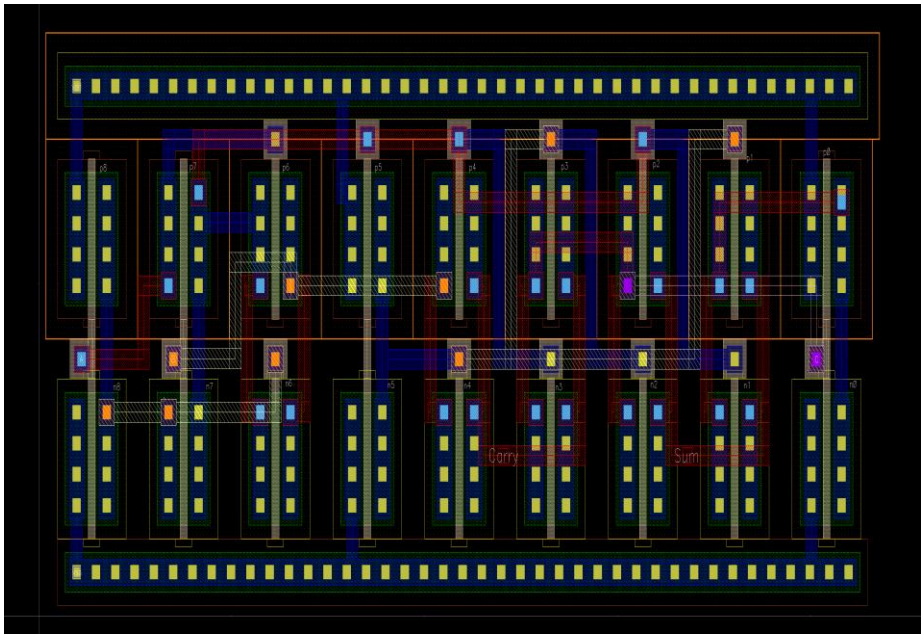

Fig6. *Layout of Full Adder*

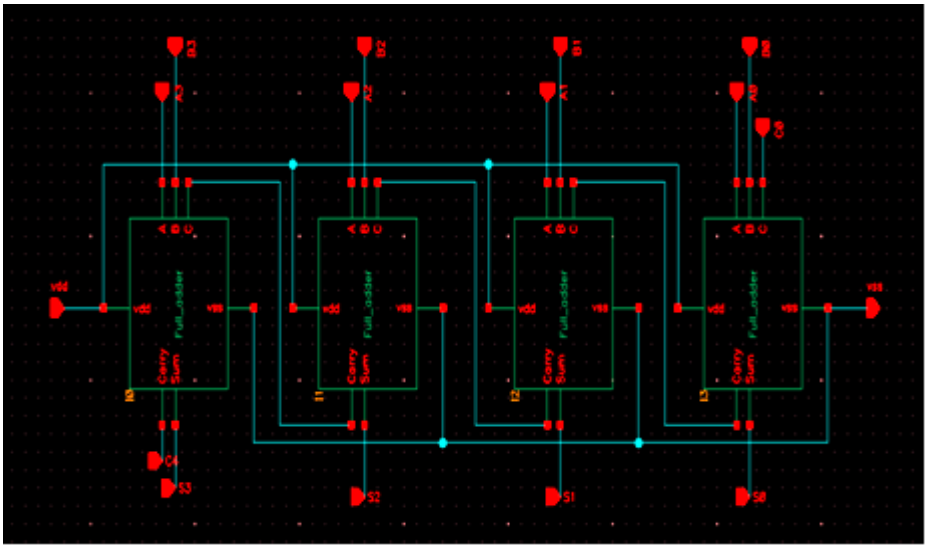

Fig7. *Schematic of 4-bit Adder*

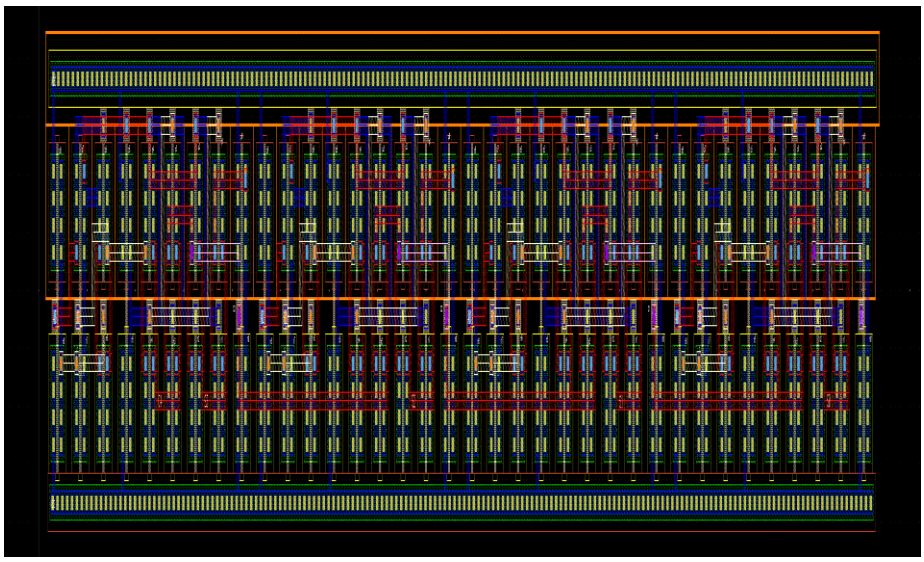

Fig8. *Layout of 4-bit Adder*

4.2 2:1 multiplexer

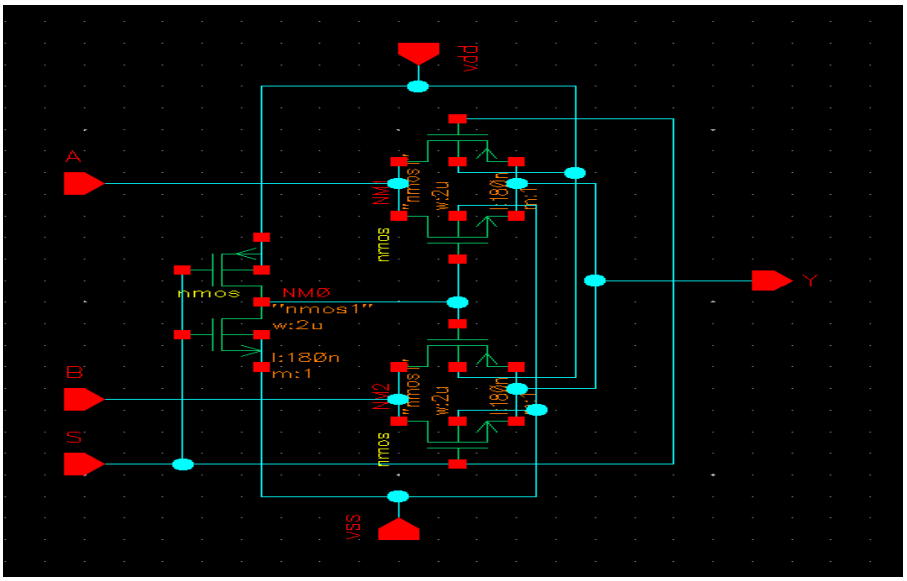

Fig9. *Schematic of 2:1 Multiplexer*

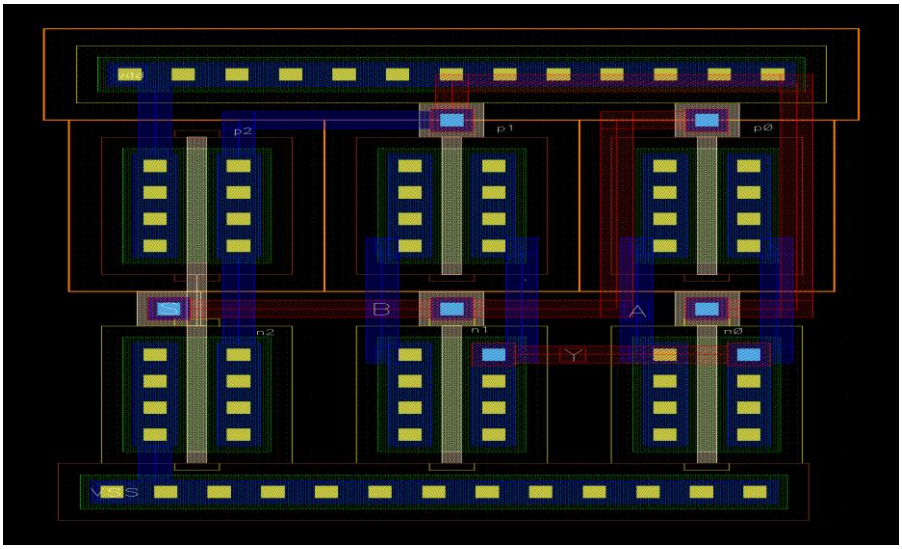

Fig10. *Layout of 2:1 Multiplexer*

4.3 8-bit CSA

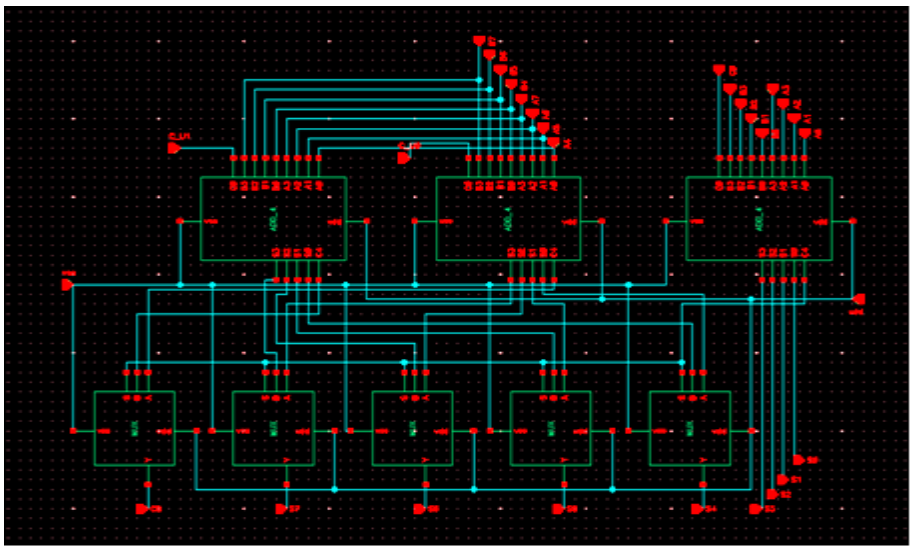

Fig11. *Schematic of 8-bit CSA*

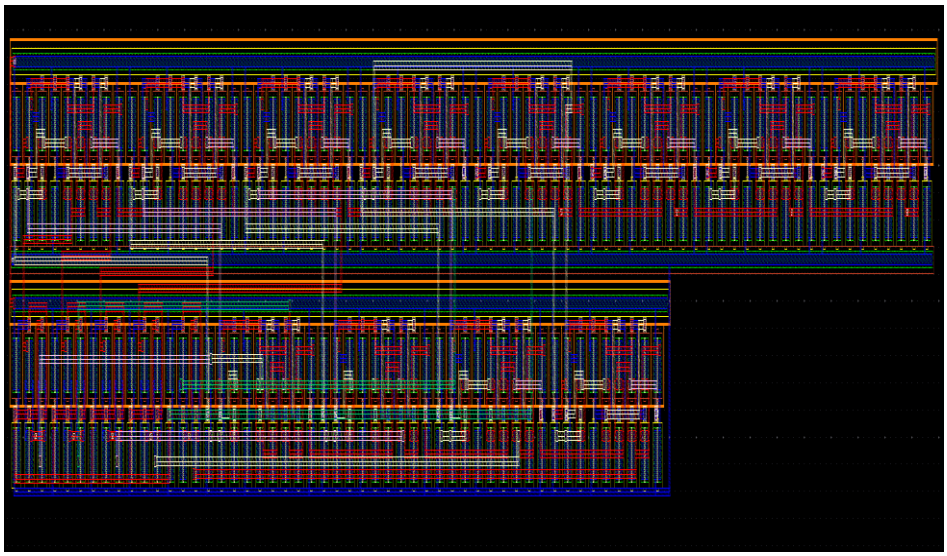

Fig12. *Layout of 8-bit CSA*

5. SIMULATION RESULTS

The 8-bit carry select adder is simulated with the supply voltage of 1.8V. Figure 10 shows the transient response of the designed adder circuit for the following inputs:

a7 a6 a5 a4 a3 a2 a1 a0 - 11010101

b7 b6 b5 b4 b3 b2 b1 b0- 00110111

c8 s7 s6 s5 s4 s3 s2 s1 s0 -100001100

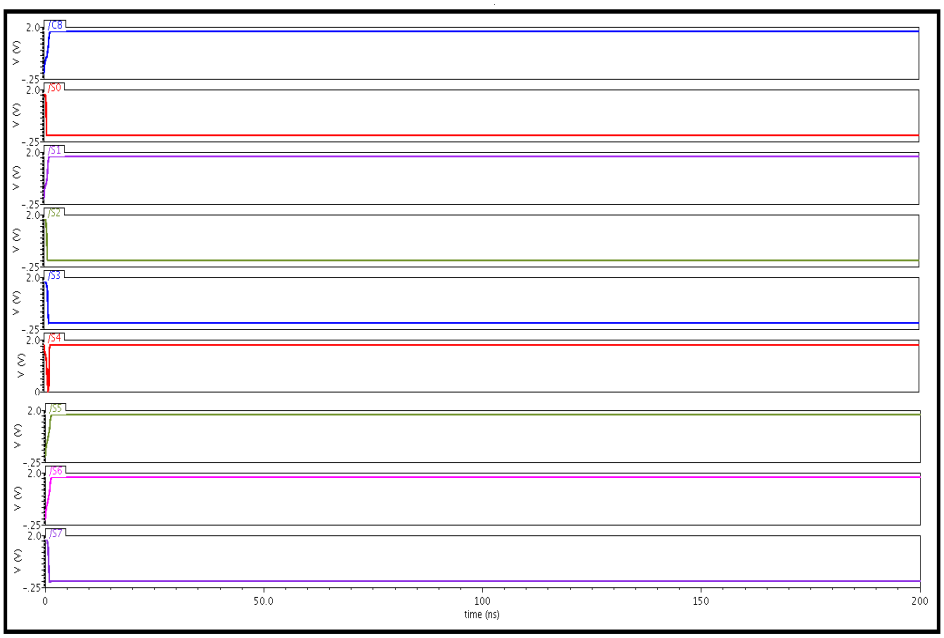

Fig13. *Simulation Results of 8-bit CSA*

The performance analysis of the designed 8-bit carry select adder and its blocks is carried out in terms of transistor count, propagation delay and power consumption.

Table1. *Performance Analysis of 8-bit RCA*

Circuit	Transistor Count	Propagation Delay in ps	Power Consumption in µW
Full Adder	18	98.7	4.7
4-bit Adder	72	105.6	22
2:1 Multiplexer	6	136.7	0.67
8-bit CSA	246	340	13.77

6. CONCLUSION

In this paper a novel 8-bit carry select adder is presented which overcomes certain drawbacks of ripple carry adder. The adder is designed using 180nm CMOS process technology. In ripple carry adder each block has to wait for the carry output of the previous block and thus propagation delay increases. The carry select adder efficiently overcomes this drawback by reducing the computational time and thus increases the speed. The performance of the design is analyzed in terms of transistor count, propagation delay and power consumption. The layout of the design is efficiently optimized based on 180nm micron rules.

REFERENCES

[1] John P. Uyemura, "Introduction to VLSI circuits and systems".

[2] Sarabdeep Singh and Dilip Kumar, "Design of area and power efficient modified carry select adder", International Journal of Computer Applications (0975 – 8887) Volume 33– No.3, November 2011.

[3] Dan Wang, Maofeng Yang, Wu Cheng XUguang Guan, Zhangming Zhu, Yintang Yang " Novel low power full adder cells in 180nm CMOS technology", 4th IEEE conference on Industrial Electronics and Applications, pp. 430-433,2009.

[4] Neil H. E. Weste, Kamran Eshranghian, " Principles of CMOS VLSI design", A systems perspective, 2nd edition.

[5] Cadence Analog and Mixed signal labs, revision 1.0, IC613, Assura 32, incisive unified simulator 82, Cadence design systems, Bangalore.

[6] Kuldeep Rawat, Tarek Darwish and Magdy Bayoumi, "A low power and reduced area carry select adder", 45th Midwest Symposium on Circuits and Systems, vol.1, pp. 467-470, March 2002.

[7] O. J. Bedrij, "Carry-select adder", IRE transactions on Electronics Computers, vol.EC-11, pp. 340-346, June1962.

[8] I-Chyn Wey, Cheng-Chen Ho, Yi-Sheng Lin and Chien-Chang Peng, "An area-efficient carry select adder design by sharing the common boolean logic term" Proceedings of the International Multiconference of Engineers and Computer Scientists 2012 Vol. II, IMECS 2012, March 14-16 2012, Hong Kong.

[9] Behnam Amelifard, Farzan Fallah, Massoud Pedram, "Closing the gap between carry select Adder and ripple carry adder: A new class of low-power high-performance adders".