16 Bit Low Power High Speed RCA Using Various Adder Configurations

Jasbir Kaur [#1], Dr.Neelam RupPrakash [*2]

Electronics & Comminucation Enfineering, P.E.C University of Technology
[1] jasbirkaur70@yahoo.co.in
[2] neelamrprakashpec @yahoo.com

Abstract— **The operation of subtraction, multiplication, division and address calculation rely on the operation of addition . The adder which lies in the critical delay path determines the speed of the systems overall speed. In recent years the option of reducing the power consumption been gaining prominence. Low power high speed parameters has been achieved with the 28 CMOS 1 bit full adder circuit in 4 bit , 8 bit and 16 bit RCA.The circuits of 1-bit full adder and RCA in this paper has been designed using cadence virtuoso tool.**

Keyword - Conventional Adder, Full Adder, Low Power, Ripple Carry Adder (RCA)

I. INTRODUCTION

To process the fundamental arithmetic operations addition is the mandatory operation. In many VLSI applications addition is the most frequently used operation in a general purpose system and in application specific processors .The operations such as subtraction ,multiplication and address calculation usually rely on the operation of addition dubbed as the heart of any microprocessor, input bits to be of equal or lower magnitude. Addition is a very crucial operation because it usually involves a carry ripple step which must propagate a carry signal from each bit to its higher bit position results in circuit substantial delay.Adder which lies in the critical delay path , effectively determines the systems overall speed . Half adder can only be used to add two single binary number , where as the full adder adds two binary numbers with a carry in.

II. VARIOUS 1 BIT FULL ADDER CIRCUITS

2.1 Conventional Adder :A full adder adds two binary no's with a carry in The conventional adder is constructed using two half adders and an OR gate.there are a total of three inputs for the full adder , two for the input no's A and B,and one for the carry in C_{in}.the outputs are the sum and carry out C_{out}

$$Sum = A \text{ XOR } B \text{ XOR } C_{in} \qquad (1)$$

$$C_{out} = (A \text{ XOR } B) C_{in} + AB \qquad (2)$$

The CMOS level implementation of a conventional CMOS full adder cell design using a total of 32 CMOS.
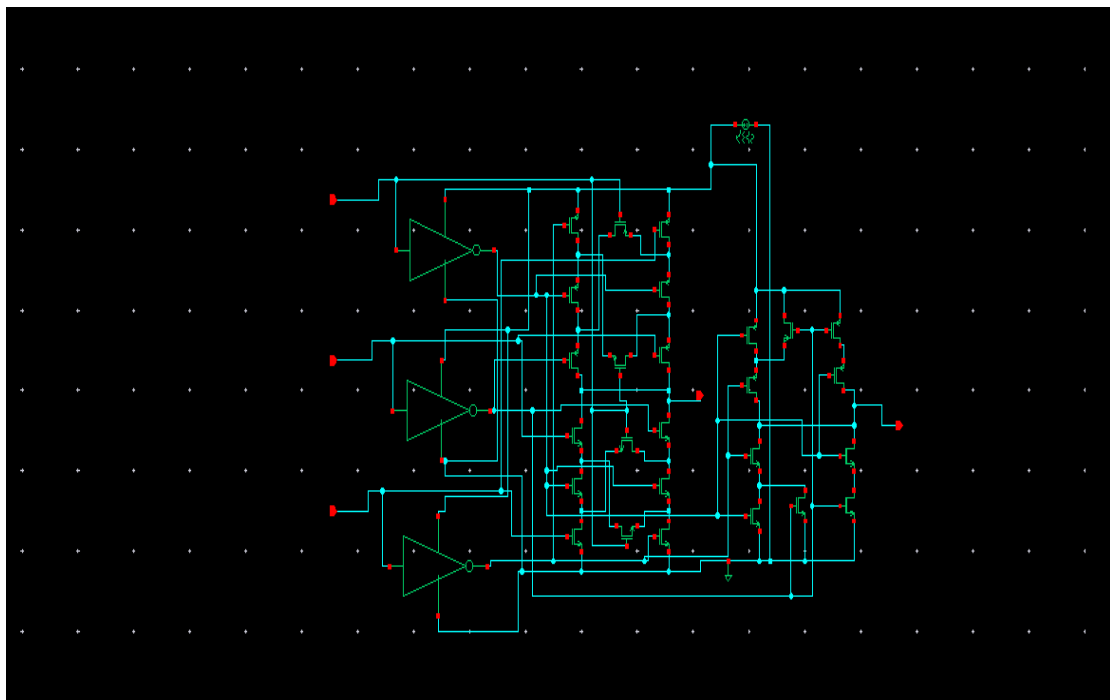

Fig 1: 32 CMOS 1 bit full adder

2.2 Modified Conventional Adder: The modified conventional CMOS full adder cell uses only 20 CMOSs.There is an alternative implementation of the FA cell which does not use XOR gates but it uses 28 CMOSs. The implementation of this full adder cell is realized by reusing the C_{out} term in the Sum term as a common sub expression. To implement it there is no use of NAND and XOR gates .For implementation of full adder cell the logic functions are as follows:

$$C_{out} = AB + C_{in}\,(A+B) \tag{3}$$

$$Sum = AB\,C_{in} + (A+B+ C_{in}\,)\,C_{out}' \tag{4}$$

$$= AB\,C_{in} + (A+B+ C_{in}\,)\,[AB + C_{in}\,(A+B)]' \tag{5}$$

This is an alternative implementation of the full adder cell that does not use XOR gates but using only 28 CMOS and is realized by reusing the C_{out} term in the Sum term as in equation (5). Full adder cell can also be implemented that does not uses XOR gates and uses only 28 CMOS .
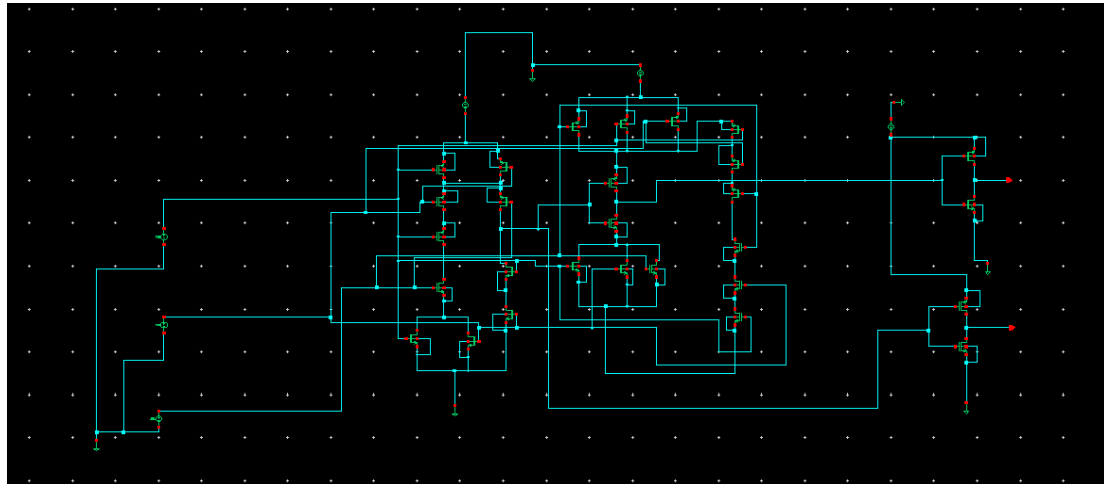

Fig 2: 28 CMOS 1 Bit Full Adder

2.3 Modified Conventional CMOS Full Adder: 20 CMOS full adder is the modified version of the conventional CMOS full adder that using 32 CMOS is logically equivalent full adder circuit based on transmission gates and inverters. The sum is obtained similar to the equation (1) where as the carry term is obtained as equation (7).

$$C_{out} = \{(AB)' \,.(BC)'.(CA)' \,\}' \tag{6}$$
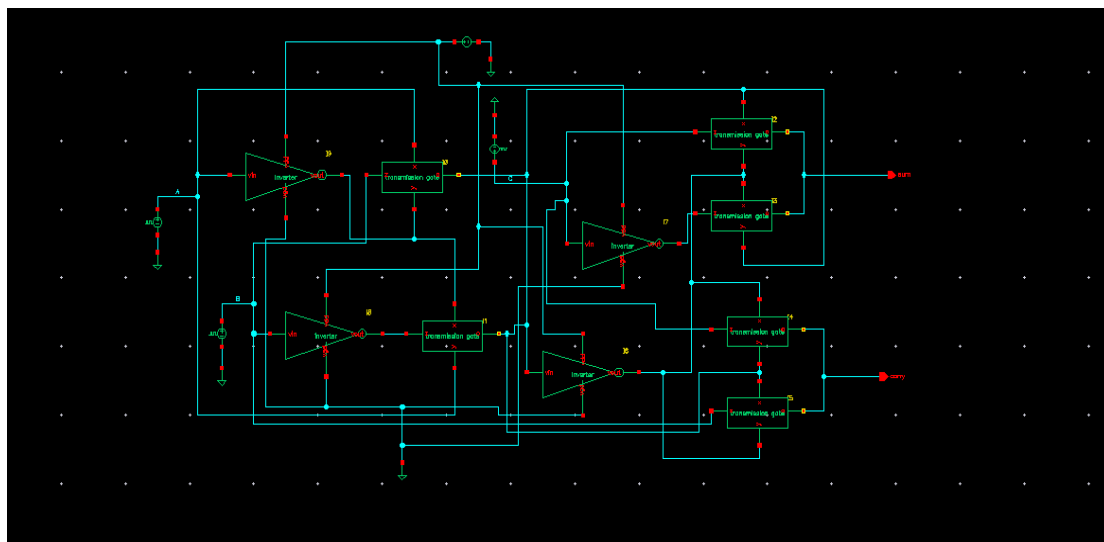
$$= AB+ BC+CA \tag{7}$$

Fig 3 :20 CMOS 1 Bit Full Adder

2.4 10 CMOS Full Adder: This type of full adder uses pass transistor, an XOR gate and an inverter . 10 CMOS full adder uses an XOR gate , an inverter and a pass CMOS.
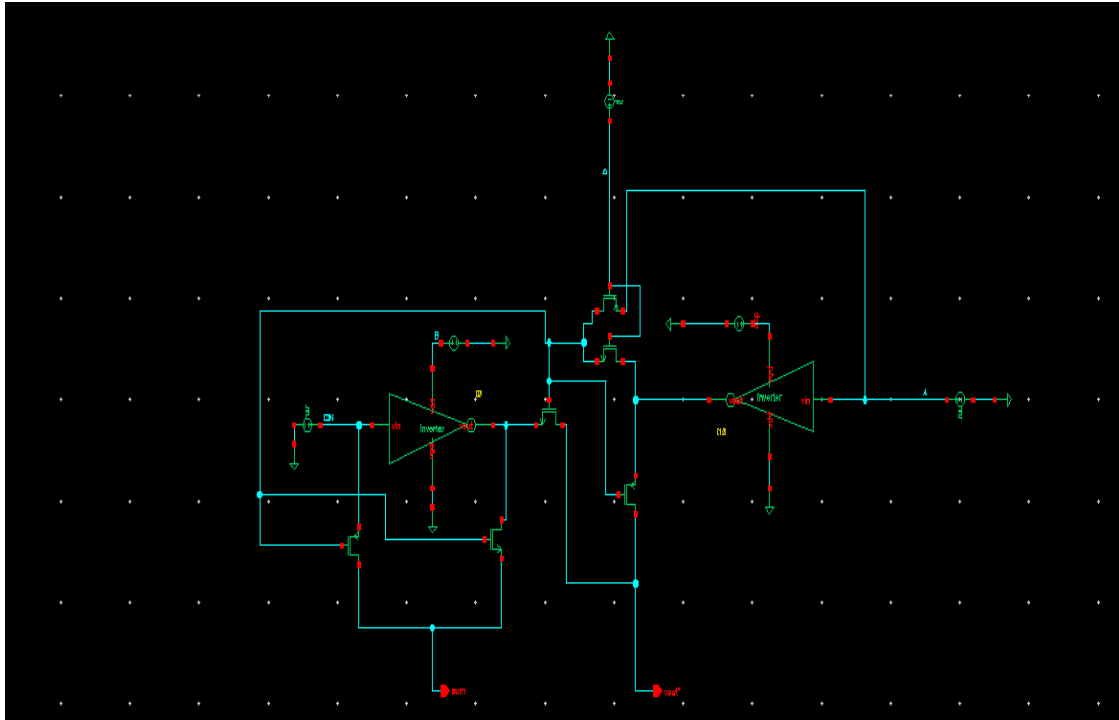
Fig 4 :10 CMOS 1 Bit Full Adder

III. RIPPLE CARRRY ADDER

3.1 16- Bit Ripple Carry Adder Circuit (RCA) Using Various 1- Bit Full Adder Cell: Full adder is the basic unit of RCA . In this simply cascading to any number by connecting the carry out of the previous 1 bit full adder to the carry in of the next 1 bit full adder . So n- bit RCA is cascading the n, 1- bit full adder. The worst case delay increases linearly with the length of the carry propagation path ,which depends on the number of bits processed by the operands n. The 4- bit Ripple carry adder circuits have been designed using the above discussed 1-bit full adder circuits by simply cascading the carry output of the previous adder is the carry input of the next full adder .The worst case delay increases linearly with the length of the carry propagation path which depends on the number of bits processed by the operands n. After designing the 4 bit RCA and using 4 bit RCA, 8 bit RCA has been designed by simply cascading the carry out of the first 4 bit RCA to the carry in of the second RCA. Again using the 8 bit RCA and following the same procedure the 16 bit RCA has been designed.
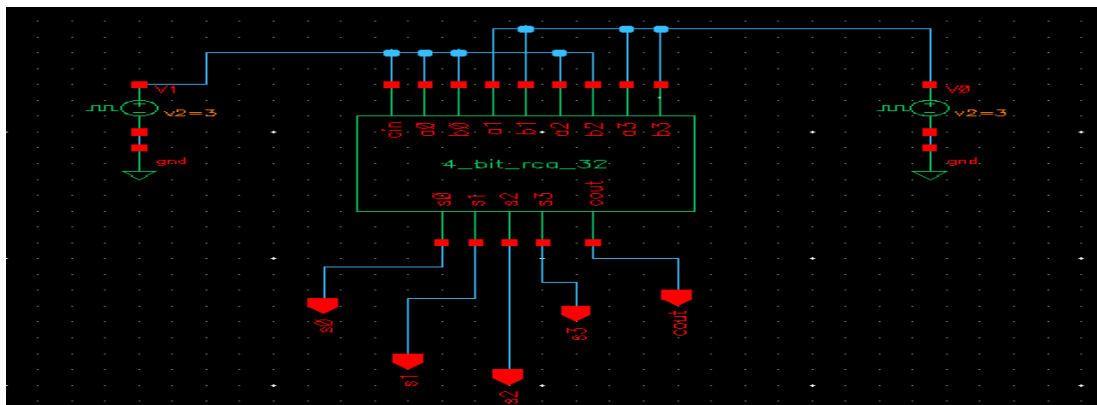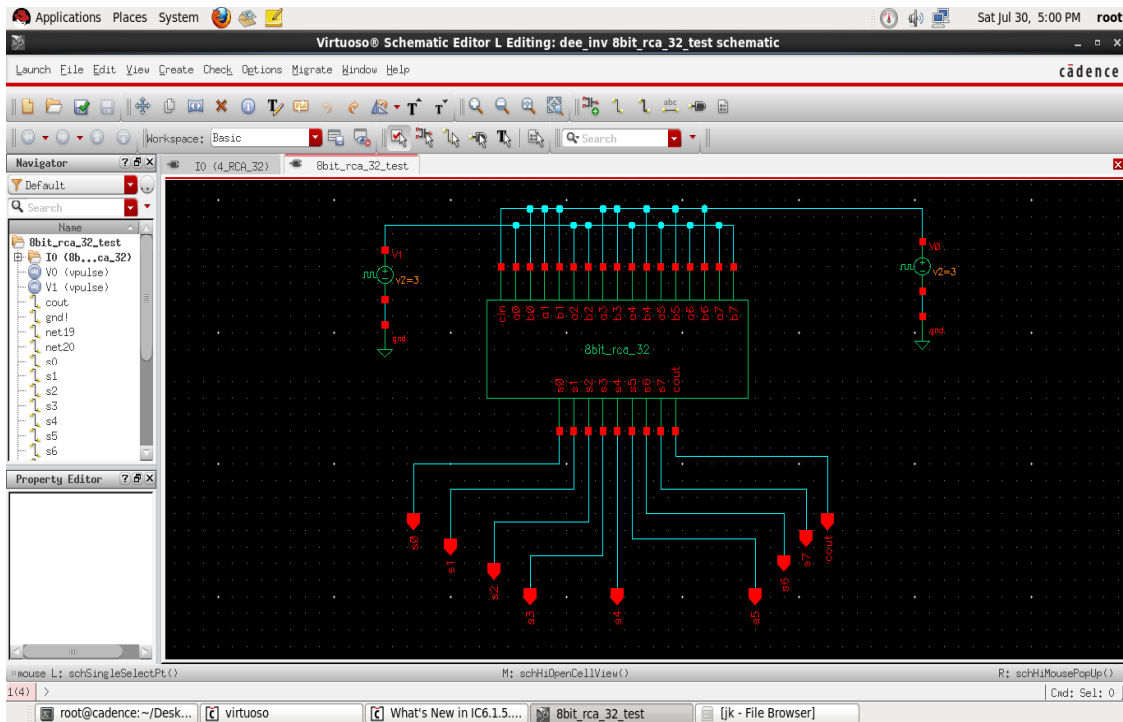

Fig 5: 4 Bit RCA Circuit

Jasbir Kaur et al. / International Journal of Engineering and Technology (IJET)

Fig 6: 8 Bit RCA Using 32 CMOS 1 Bit Full Adder Circuit
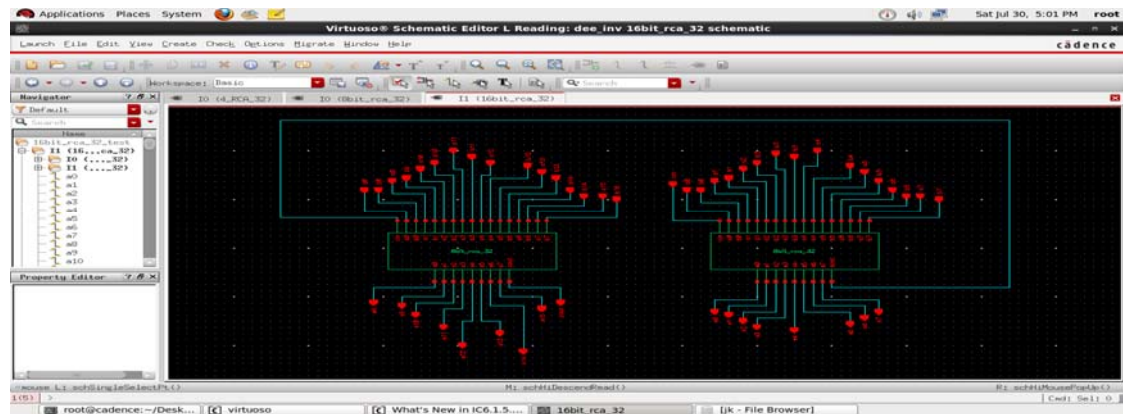

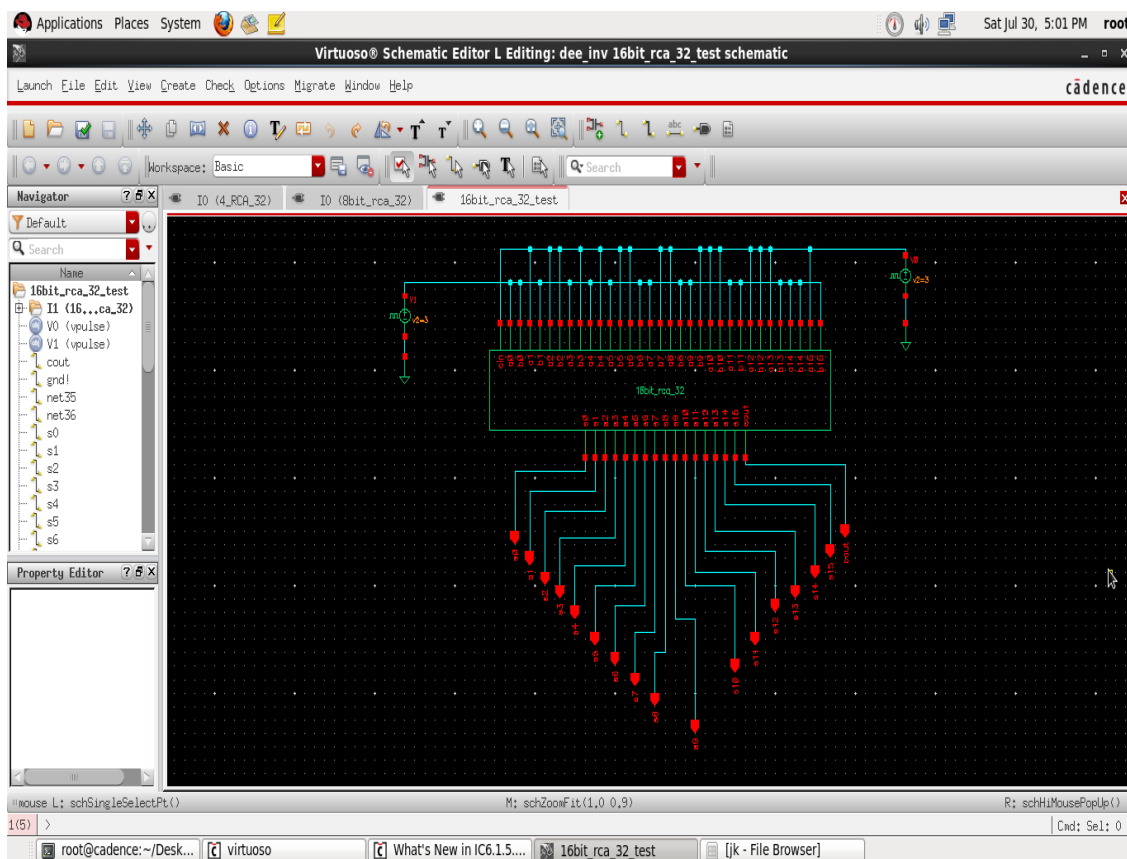
Fig 7: Internal Circuit of 8 Bit RCA

Fig 8 : 16 Bit RCA

IV. SIMULATION AND RESULTS

The simulation of 1 bit adder using 32 CMOS,28 CMOS,20 CMOS and 10 CMOS , and 4 bit RCA,8 bit RCA and 16 bit RCA circuits has been implemented using the Cadence Virtuoso tool using 180 nm technology. The results shows that the 28 CMOS 1 bit full adder having the delay parameter of 58.28 ps where as the 32 CMOS of 278.1ps,20 CMOS of 30460 ps and 10 CMOS of 5085 ps and power consumption of 10.6 µw for 28 CMOS full adder is the minimum power consumption. Further the 4 bit RCA using 28 CMOS full adder also shows the minimum power consumption of 119.8 µw and minimum delay of 549.3 ps.8 bit and 16 bit RCA using 28 CMOS full adder also outperforms in power consumption and delay as compare to the 32 CMOS, 20 CMOS and 10 CMOS full adder. The output waveforms shows the correctness of the functionality of the circuits.
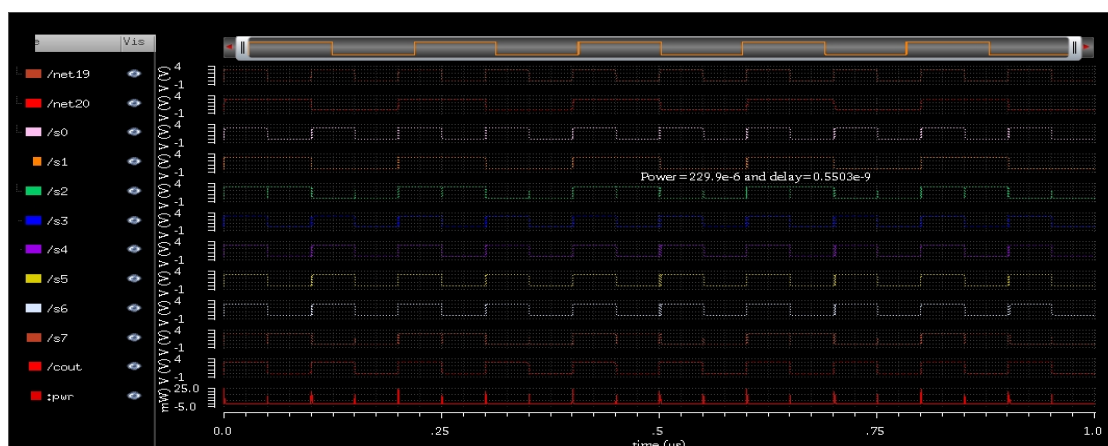

Fig 9: Output Waveform of 8 Bit RCA

The table 1 shows the results of the designed circuits using the cadence tool. The power and delay parameters are obtained using the Cadence tool for all the circuits .

Table 1 Comparison of Delay and Power of Various 1- Bit Adder and RCA

Type of the circuit	Circuit using no. of CMOS	Power consumption (μw)	Delay(ps)	PDP femto watt s
1 bit full adder	32 CMOS	25.48	278.1	7.085988
	28 CMOS	10.6	58.28	.617768
	20 CMOS	56.44	30460	1719.1624
	10 CMOS	183100	5085	931063.500
4 bit RCA	32 CMOS	132.6	50840	6741.384
	28 CMOS	119.8	549.3	65.806140
	20 CMOS	53.01	50810	2693.438
	10 CMOS	15.32	50800	778.256
8 bit RCA	32 CMOS	252.6	50840	12842.184
	28 CMOS	229.9	550.3	126.51397
	20 CMOS	169.4	50810	8607.214
	10 CMOS	869.6	50810	44184.376
16 bit RCA	32 CMOS	513.7	50900	26147.330
	28 CMOS	440.2	553.2	243.518640
	20 CMOS	365.1	50830	18558.033
	10 CMOS	2289	150700	34495.230

V. CONCLUSION

In this paper various RCA circuits using various adder architectures have been designed and the 28 CMOS adder architecture outperforms in all the cases . This shows that the 28 CMOS adder architecture is suitable for low power applications. The power delay product parameter of 28 CMOS 4 bit. 8 bit and 16 bit RCA is better.

REFERENCES

[1] A.M . Shams and M.A Bayoumi " A new full adder cell for low power specifications." in Proc IEEE 8th Great Lakes Symp . VLSI . Feb 1998 pp 45-49
[2] C. Nagendra , M.J Irwin and R. M Owens, " Area –time-power trade offs in parallel adders." IEEE Trans Circuits and Systems –II Analog and Digital Signal Processing. Vol. 43 ,No. 10 pp 689-702 , Oct 1996
[3] H.A. Mahmoud and M.A .Bayoumi, " A 10 CMOS low power high speed full adder cell," IEEE Int. Symp. Circuits and Systems , May June 1999, pp 43-46
[4] A. Bellaouar and M.I. Elmasry, Low power digital VLSI design :Circuits and systems, The Netherlands : Kluwer Academic Publishers,1995
[5] I.Koren Computer Arithmetic Algorithms, Englewood Cliffs, New Jersy , Prentice Hall, 1993 [6] M.W Allam and M.I. Elmsary, Low power implementation of fast addition algorithms," IEEE Canadian Conf. Electrical and Computer Engineering, May 1998, pp 645-647