

# Homogenous Adder Based Logic Realization for Ripple Carry Adder

<sup>1</sup>Komal, <sup>2</sup>Dr. A.K. Gautam

<sup>1</sup>M.Tech Scholar, <sup>2</sup>Principal

Department of Electronics & Communication Engineering,  
S.D. College of Engineering and Technology, Muzaffarnagar, India

**Abstract:** An adder is the fundamental building block of the digital circuits such as ALU, microprocessors and microcontrollers, DSP processors and several arithmetic operations. Full adder is the main part of the arithmetic circuits. Full adder circuit is the basic cell of arithmetic circuits. Many applications need logic circuits of small chip area, high throughput, and low power utilization. The paper is focused on the design of 16-bit ripple carry adder based on homogeneous adder-based concept. The chip design is done in Xilinx ISE 14.2 using VHDL and simulated in Modelsim 10.1 software. The design is synthesized on SPARTAN - 3E FPGA to verify the results.

**Index Terms:** Ripple Carry Adder, Digital Signal Processing, FPGA, VHDL programming

## I. INTRODUCTION

Adders are the basic building blocks of the digital applications and computing of microprocessors. The adders used for these systems are half adder and full adder. Half adder and full adder [1, 2] are the combinational circuit that provides the sum of two bit binary and three bits binary respectively and provides the sum and carry. The larger bit data is added by parallel adder or the ripple carries adder. The adder is used for different DSP applications such as filter design [3]. The full adder is the basic component of the ripple carry adder. Full adder [4, 5] is combinational logic which is accepting 3 binary bits to provide sum and carry as the output. The logic diagram is shown in fig. 1 and behavior of full adder is shown in table 1. The logical expressions are presented with the help of equ.(1) and equ.(2).

Table 1 Truth table of full adder

Input			Output	
a	B	c	sum	carry
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

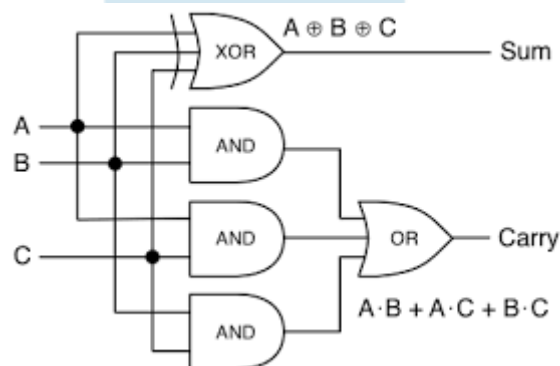


Fig. 1 Logic diagram of full adder

$$sum = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc = a \oplus b \oplus c \quad (1)$$

$$carry = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc = ab + bc + ca \quad (2)$$

**II. RIPPLE CARRY ADDER (RCA)**

The addition operation in DSP application is followed by Ripple Carry Adder (RCA) [6, 7] instead of decimal adder. The RCA will accept the 4-bit data as shown in Fig. 2. The 4-bit adder using full adder circuits can add two 4-bit numbers resulting a 4-bit sum and a carry. Hence the sum by each full adder is performed simultaneously. This circuit is called the parallel adder or ripple carry adder. In a 4-bit parallel binary adder circuit, [8] the input of each full adder will be  $A_i, B_i$  and  $C_i$  and the outputs will be  $S_i$  and  $C_{i+1}$ , where  $i$  varies from 0 to 3. The carry of the previous stage is associated with carry input of succeeding stage [9, 10]. In the first stage,  $A_0, B_0$  and  $C_0$  (or  $C_0 = C_{in}$ , which is 0) are added resulting sum  $S_0$  and carry  $C_1$ . This carry  $C_1$  becomes the carry input to the second stage.

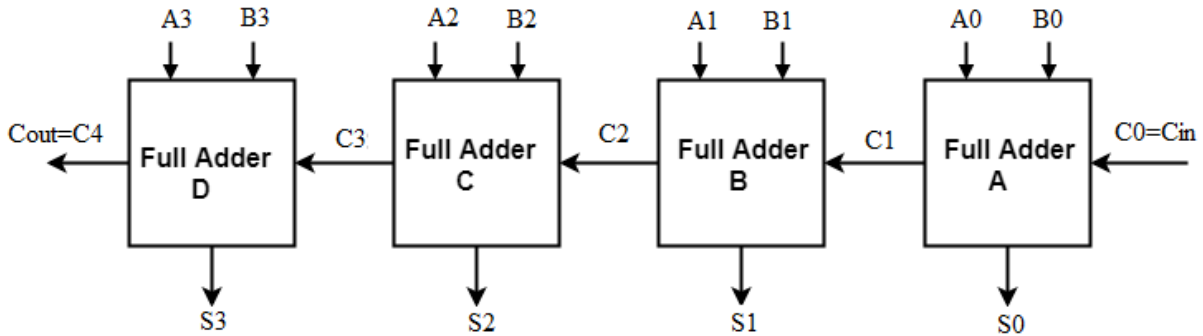


Fig. 2 Four-bit binary adder or Parallel binary adder

In the same way, at the second stage  $A_1, B_1$  and  $C_1$  are added resulting  $S_1$  and  $C_2$ . Carry  $C_2$  is the input for third stage.  $A_2, B_2$  and  $C_2$  are added resulting  $S_2$  and  $C_3$ . Carry  $C_3$  is carried in the fourth stage.  $A_3, B_3$  and  $C_3$  are added and resulting  $S_3$  and  $C_4$  as out carry or  $C_{out}$ . Thus, this circuit results sum  $(S_3S_2S_1S_0)$  and carry  $C_{out}$ .

**III. REALIZATION OF 16-BIT HOMOGENOUS ADDER**

In the same way, 16-bit ripple adder can be designed using either direct 16 number of full adders based on the concept of 4-bit RCA or it is also possible to use the cascade structure shown in the fig. 3. In which 4 RCA are used in which RCA1 performs the addition of  $x(3:0)$  and  $y(3:0)$ , RCA2 adds  $x(7:4)$  and  $y(7:4)$ , RCA3 adds  $x(11:8)$  and  $y(11:8)$  and RCA4 performs  $x(15:12)$  and  $y(15:12)$ . The output of the RCA1, RCA2, RCA3 and RCA4 gives the out of sum  $S_{out}(3:0), S_{out}(7:4), S_{out}(11:8)$ , and  $S_{out}(15:12)$ . The output carry of one RCA is given as the input to the next RCA. In the RCA as the length increases there are the chances that the delay will increase. The delay calculation for the 'N' bit RCA is given by the equ. (3).

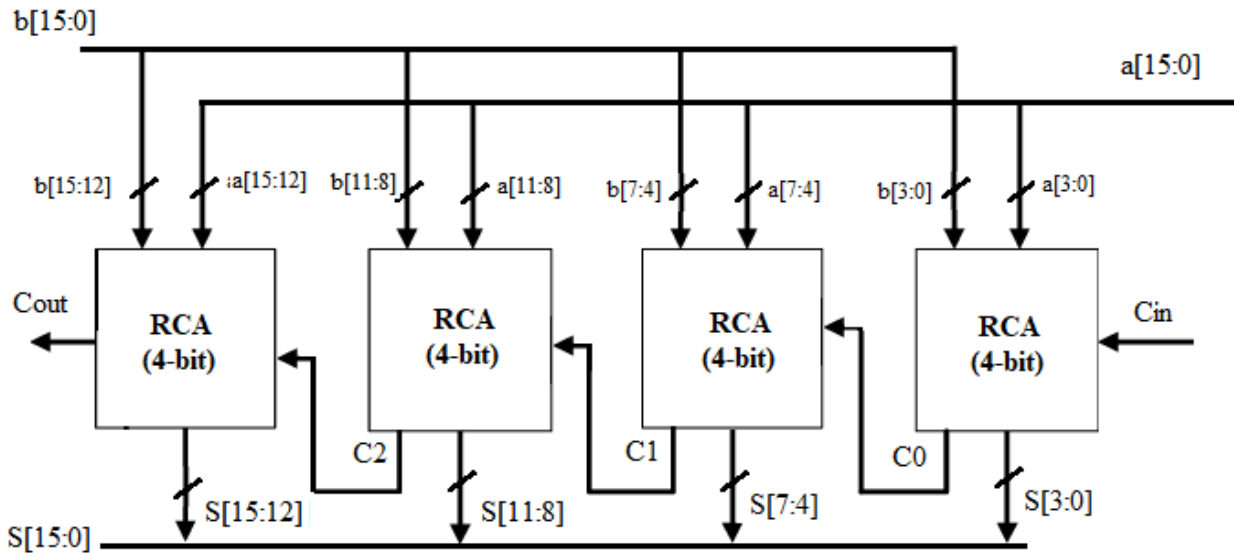


Fig. 3 Homogenous adder-based structure (16 bit)

$$T_{logic} = T_{Fulladder}((x_0, y_0) \text{ to } c_0) + (n - 2) \times T_{Fulladder}(C_{in} \text{ to } C_{out}) + T_{Fulladder}(C_{in} \text{ to } S_{out}(n - 1)) \quad (3)$$

In This equation,  $T_{Fulladder}$  presents the delay of the full adder for the specified input and output with respect to the path of I/O.

**IV. RESULTS & DISCUSSIONS**

The Xilinx software provides the RTL of the developed chip of adder which is depicted in the fig. 4 and internals schematic is also presented in the fig.5. The RTL provides the complete information of the input and output pins used in the design. The description of all the pins is given in table 2. The waveform simulation of the same is given in fig.6

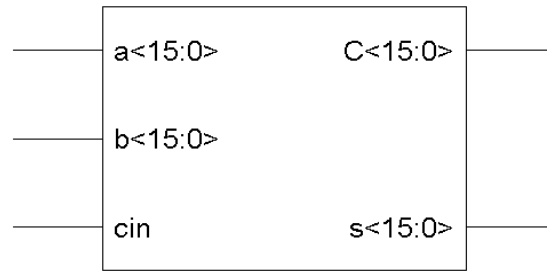


Fig.4 RTL view of adder

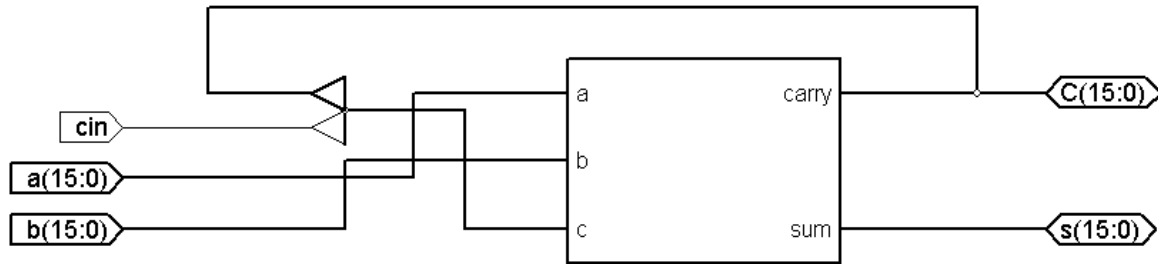


Fig.5 Internal schematic of 16-bit adder

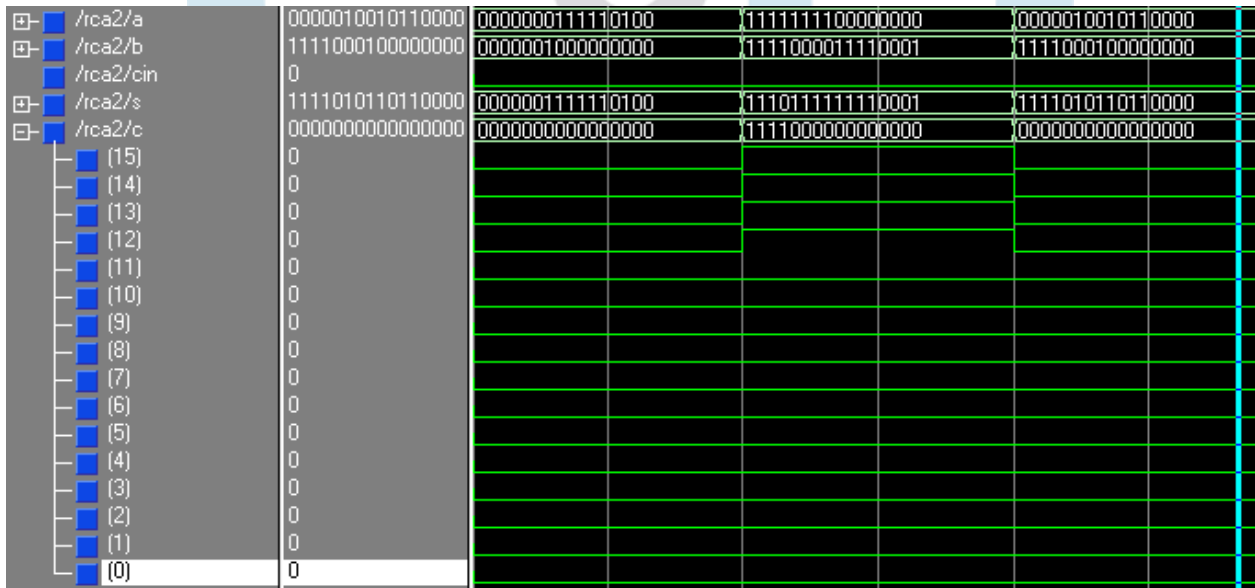


Fig.6 Modelsim waveform simulation of 16-bit adder

**Test input-1:** a = 0000000111110100 and b = 0000001000000000 then output s = 000000111110100  
**Test input-2:** a = 1111111100000000 and b = 1111000011110001 then output s = 1111000000000000  
**Test input-3:** a = 0000010010110000 and b = 1111000100000000 then output s = 1111010110110000

Table 2 Pin details of the chip

Pin	Description
A[15:0]	It is the first 16-bit input of the adder. It is of std_logic_vector data type.
B[15:0]	It is the second 16-bit input of the adder. It is of std_logic_vector data type.
Cin	It is the input carry, considered as Cin = 0. It is of std_logic.
S[15:0]	It is the adder output of 16-bit. It is of std_logic_vector data type.
C[15:0]	It is 16-bit outputs considered as intermediate carry of the adder including output carry as c[15]. It is of std_logic_vector data type.

The Xilinx software gives the details of the hardware utilization for the homogenous adder based 16-bit ripple carry adder. The table 3 provides the details of hardware summary and timing values. The hardware parameter slices, LUTs, memory and bounded input/output (IoBs), and 1-bit XOR2. Fig.7 presents the graph for the hardware resources. The combinational time presents the logic delay and routing delay from input to output. The information is directly extracted from the software with the target device on SPARTAN 3E FPGA.

Table 3 Xilinx software parameters for 16- bit adder

Parameter	16-bit Homogenous Adder
Slices Usage	23
LUTs Usage	40
IoBs	65
1 bit XOR2	16
Memory Usage	111572 kB
Combinational time	36.323 ns

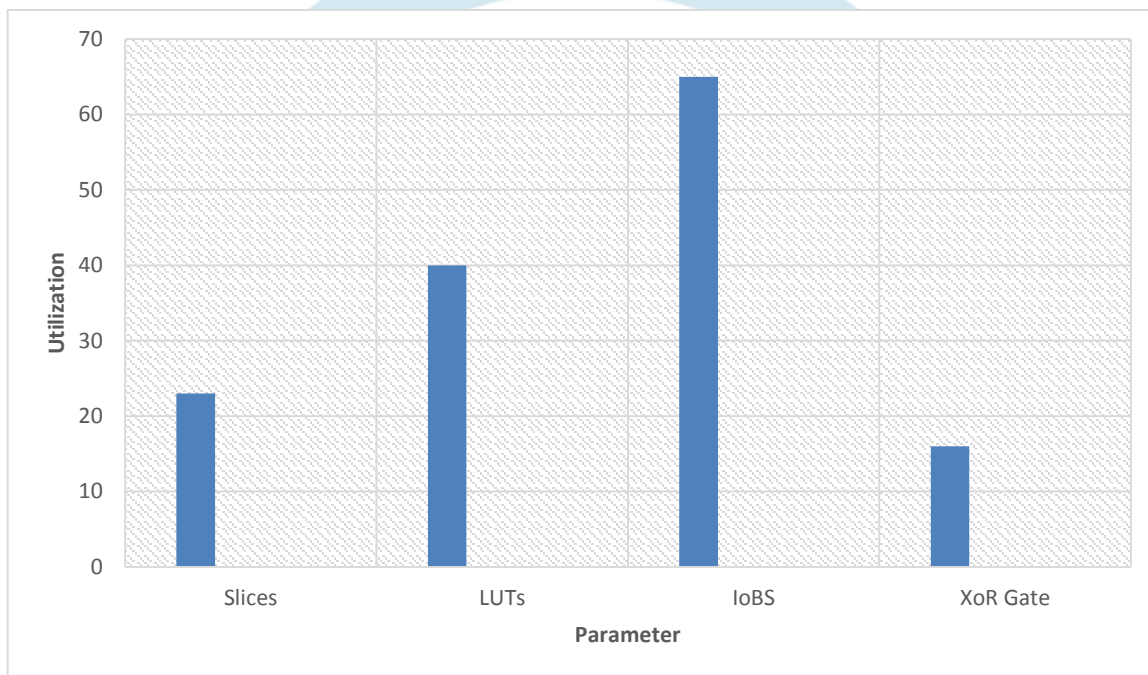


Fig.7 Hardware usage curve with different parameters

## V. CONCLUSIONS

The design of 16-bit homogenous ripple carry adder is done in Xilinx ISE 14.2 successfully. The output of the adder is simulated for many cases in modelsim10.1 software. The design is based on cascading approach of same type of full adders circuits to form the 4 bit ripple carry adder and further use to implement the 16 bit adder. The such design is applicable in several digital circuit and DSP based communication system. The designed chip is synthesized successfully on SPARTAN-3E FPGA and device parameters such slices, device LUT, IoBs, XOR gate are 23, 40, 65 and 16 respectively. The design is using the memory 111572 kB and combinational path time is 36.323 ns. The developed chip will be beneficial for the signal processing applications in which addition is the building blocks and computations are needed such as FIR filter design, DFT algorithm, FFT computations in which large scale adders are multipliers are used.

## References

- [1] Archana, S., & Durga, G. (2014, April). Design of low power and high-speed ripple carry adder. In *2014 International Conference on Communication and Signal Processing* (pp. 939-943). IEEE.
- [2] Devi, P., Girdher, A., & Singh, B. (2010). Improved carry select adder with reduced area and low power consumption. *International journal of computer applications*, 3(4), 14-18.
- [3] Fang, C. J., Huang, C. H., Wang, J. S., & Yeh, C. W. (2002). Fast and compact dynamic ripple carry adder design. In *Proceedings. IEEE Asia-Pacific Conference on ASIC*, (pp. 25-28). IEEE.
- [4] Ganesan, S. (2015). *Area, delay and power comparison of adder topologies* (Doctoral dissertation).
- [5] Hoe, D. H., Martinez, C., & Vundavalli, S. J. (2011, March). Design and characterization of parallel prefix adders using FPGAs. In *2011 IEEE 43rd Southeastern Symposium on System Theory* (pp. 168-172). IEEE.
- [6] Kim, Y., & Kim, L. S. (2001). 64-bit carry-select adder with reduced area. *Electronics Letters*, 37(10), 614-615.
- [7] Mahendran, N., & Vishwaja, S. (2016). Performance Analysis of High-Speed Adder for DSP Applications. *World Academy of Science, Engineering and Technology, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, 10(3), 464-469.

- [8] Mano, M. M. (2017). *Digital logic and computer design*. Pearson Education India.
- [9] Parmar, S., & Singh, K. P. (2013, February). Design of high-speed hybrid carry select adder. In *2013 3rd IEEE International Advance Computing Conference (IACC)* (pp. 1656-1663). IEEE.
- [10] Uma, R., Vijayan, V., Mohanapriya, M., & Paul, S. (2012). Area, delay and power comparison of adder topologies. *International Journal of VLSI Design & Communication Systems*, 3(1), 153.

