



ISSN: 2319-5967

ISO 9001:2008 Certified

International Journal of Engineering Science and Innovative Technology (IJESIT)

Volume 2, Issue 4, July 2013

Performance Analysis of Different Bit Carry Look Ahead Adder Using VHDL Environment

Rajender Kumar, Sandeep Dahiya

SES, BPSMV, Khanpur Kalan, Gohana, Sonipat, Haryana

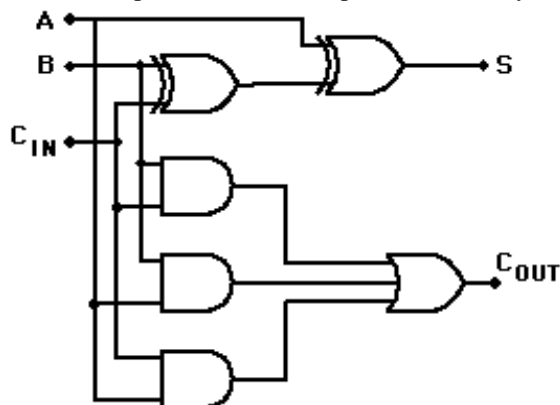
Abstract- Adders are some of the most critical data path circuits requiring considerable design effort in order to squeeze out as much performance gain as possible. Various adder structures can be used to execute addition such as serial and parallel structures and most of researches have done research on the design of high-speed, low-area, or low-power adders. Adders like ripple carry adder, carry select adder, carry look ahead adder, carry skip adder, carry save adder etc exist numerous adder implementations each with good attributes and some drawbacks. This paper focuses on the implementation and simulation of 4-bit, 8-bit and 16-bit carry look-ahead adder based on Very High Speed Integrated Circuit Hardware Description Language (VHDL) and compared for their performance. The simulation is done using ModelSim SE 6.3f and we have recorded the performance improvements in propagating the carry and generating the sum when compared with the traditional carry look ahead adder designed in the same technology.

Keywords- Addition, Carry-Lookahead Adder (CLA), High Performance, VHDL Simulation, ModelSim, Xilinx, FPGA.

I. INTRODUCTION

Arithmetic operations such as addition, subtraction, multiplication and division are widely used and play an important role in various digital systems such as digital signal processor (DSP) architecture, microprocessor and microcontroller and data process unit.

Adders are the logic circuits designed to perform high speed arithmetic operations and are important components in digital systems because of their extensive use in other basic operations such as subtraction, multiplication and division. In many computers and other kinds of processors, adders are used not only in the arithmetic logic unit, but also in other parts of the processor, where they are used to calculate addresses, table indices, and same kind of operations. The very basic arithmetic operation is the addition of two binary digits, i.e. bits. A combinational circuit that adds two bits according to the scheme outlined below is called a half adder. A full adder along with block diagram and truth table is shown in Fig. 1 is one that can add three bits, the third bit produced from a previous addition operation i.e. carry coming from lower order bits [1]-[6].



A	B	Cin	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Fig. 1 Block Diagram and truth table of Full Adder Circuit

Addition is a fundamental arithmetic operation that is used in many VLSI design systems like DSP architecture, microprocessor, microcontroller and data process unit. This VLSI system requires fast addition which impacts the overall performance of digital system. These addition operations are done by using adders. Various adder structures can be used to execute addition such as serial and parallel structures and most of researches are done on the design of high-speed, low-area, or low-power adders [7]-[10].



ISSN: 2319-5967

ISO 9001:2008 Certified

International Journal of Engineering Science and Innovative Technology (IJESIT)

Volume 2, Issue 4, July 2013

Moreover, there are various types of adders such as Ripple Carry Adder (RCA), Carry-Lookahead Adder (CLA), Carry Save Adder (CSA), Carry Select Adder, Carry-Bypass Adder or Carry Skip Adder (CSK) etc discussed in brief [11-22].

A. *Ripple Carry Adder (RCA)*

RCA is simple several full adders connected in series so that the carry must propagate through every full adder before the addition is complete. Each full adder inputs a C_{in} , which is the C_{out} of previous adder. This kind of adder is a ripple carry adder, since each carry bit ripples to the next full adder. RCA is always preferred in terms of power and area when it appears to be fast enough for its intended purpose. RCA require the least amount of hardware of all adders, but they are the slowest [11]-[12].

B. *Carry Lookahead Adder (CLA)*

A fast method of adding numbers is called carry-lookahead. This method does not require the carry signal to propagate stage by stage, causing a bottleneck. Instead it uses additional logic to expedite the propagation and generation of carry information, allowing fast addition at the expense of more hardware requirements. Further, CLA is discussed in length in section II [15]-[18].

C. *Carry Save Adder (CSA)*

CSA is a kind of adder with low propagation delay, but instead of adding two input numbers to a single sum output, it adds three input numbers to an output pair of numbers. When its two outputs are then summed by a traditional carry-lookahead or ripple carry adder, we received the sum of all three inputs. In particular, the propagation delay of a CSA is not affected by the width of vectors being added. Each full adder's output S is connected to corresponding output bit of one output, and its output C_{out} is connected to the next higher output bit of the second output; the lowest bit of the second output is fed directly from the carry-save's C_{in} input [23].

D. *Carry Select Adder*

The n -bit adder is divided into ' k ' ripple-carry adders of n/k bits each and except the lowest order part; all these adder blocks are simulated. The simplest n -bit carry select adder is built using three $n/2$ bit ripple carry adders. The first adder is utilized to compute the lower half of the n -bit sum, while the other two compute the higher half: one based on the assumption that the input carry is zero, the other based on the assumption that it is one. This way the computation of the higher half can start immediately; there is no need to wait for the lower half to complete. When the lower half of the sum is computed and the carry input for the next stage is available, the correct half of the sum is selected by a multiplexer. Because of the simulation technique the required area and power consumption of this adder is particularly doubles with respect to RCA [23].

E. *Carry-Bypass Adder or Carry Skip Adder (CSK)*

CSK is very simple but creative adder with a minimum of additional logic. The n -bit adder is divided into ' k ' ripple-carry adder blocks. Each adder block has a group propagate signal meaning that when this signal is 1, an incoming carry can not be absorbed and will propagate through the adder block as an alternative by skipping the adder segment via the skip logic [14].

These adders have different performance in terms of delay, area and power for same length of binary numbers. The paper is organized as follows: section II provides the overview of CLA adder, different section of CLA adder and its functioning. Section III focuses on design methodology and VHDL code. Section IV presents the simulation results along with device utilization summary following with conclusion.

II OVERVIEW OF CARRY LOOK-AHEAD ADDER

A carry-lookahead adder improves speed by reducing the amount of time required to determine carry bits. It can be contrasted with the simpler, but usually slower, ripple carry adder for which the carry bit is calculated alongside the sum bit, and each bit must wait until the previous carry has been calculated to begin calculating its own result and carry bits. The carry-lookahead adder calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger value bits. The CLA solves the problem of delay it takes to propagate the carry, by calculating the carry signal in advance based on the input signal [11]-[13], [16]-[19]. The working of this adder can be understood by manipulating Boolean expressions dealing with full adder. The propagate ' P_i ' and generate ' G_i ' in a full adder is given by

$P_i = x_{in} \text{ xor } y_{in}$ Carry propagate



ISSN: 2319-5967

ISO 9001:2008 Certified

International Journal of Engineering Science and Innovative Technology (IJESIT)

Volume 2, Issue 4, July 2013

$G_i = x_{in} + y_{in}$ Carry generate

Both propagate and generate signals depend only on the input bits and thus will be valid after one gate delay.

The new expressions for the output sum and the carryout are given by:

$$\text{sum} = S_i = P_i \text{ xor } C_{i-1}$$

$$\text{carry_out} = C_{i+1} = G_i + P_i \text{ and } C_i$$

These equations show that a carry signal will be generated in two cases:

- 1) If both bits x_{in} and y_{in} are 1
- 2) If either x_{in} or y_{in} is 1 and the carry_{in} is 1.

Let's apply these equations for a 4-bit adder

$$C_1 = G_0 + P_0C_0$$

$$C_2 = G_1 + P_1C_1 = G_1 + P_1(G_0 + P_0C_0) = G_1 + P_1G_0 + P_1P_0C_0$$

$$C_3 = G_2 + P_2C_2 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$$

$$C_4 = G_3 + P_3C_3 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0$$

Similarly we can write the general expression as

$$C_{i+1} = G_i + P_iG_{i-1} + P_iP_{i-1}G_{i-2} + \dots + P_iP_{i-1}\dots P_2P_1G_0 + P_iP_{i-1}\dots P_1P_0C_0.$$

Carry look-ahead adder's structure can be divided into three parts: the propagate/generate generator, the sum generator, carry generator. The architecture of 4-bit Carry Look-Ahead adder is shown in Fig. 2 [source: Lecture notes on parallel adders by Jo Rajotte].

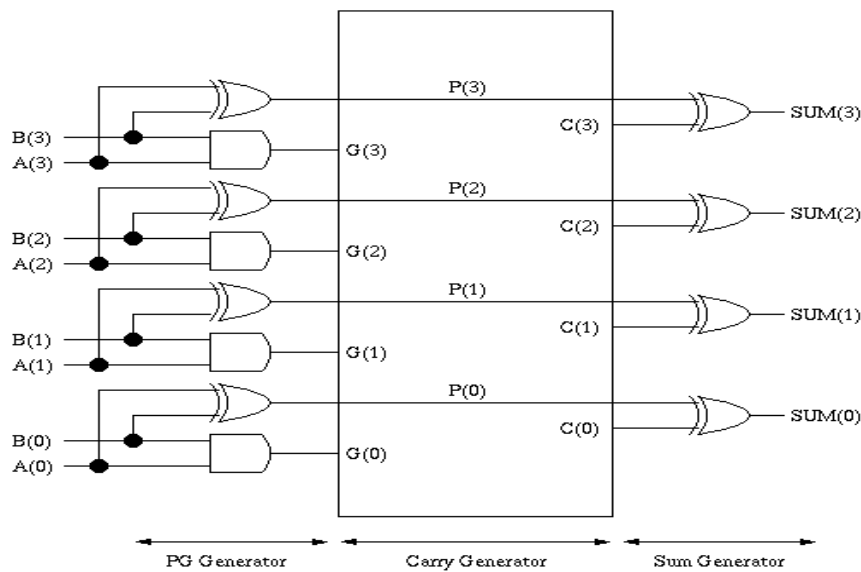


Fig. 2 4-bit Carry Look Ahead Adder

III DESIGN METHODOLOGY & IMPLEMENTATION

A. Introduction of Design Language

Design languages provide the means by which to describe the operation of both software programs and hardware. These descriptions, usually text based, are developed and stored as ASCII text files within the computer on which the descriptions are being developed. Design languages are of two types: software programming language & hardware programming language. The basic difference between two languages is shown in Fig. 3.



ISSN: 2319-5967

ISO 9001:2008 Certified

International Journal of Engineering Science and Innovative Technology (IJESIT)

Volume 2, Issue 4, July 2013

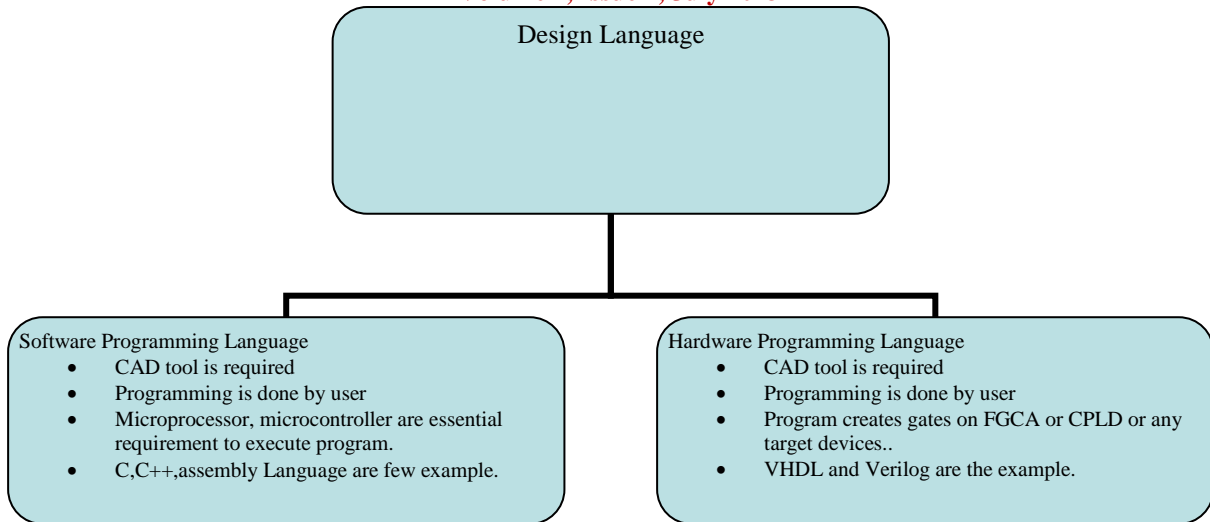


Fig. 3 Structure of Design Language

B. VHDL & its History

With rapid development in IC technology in 1980's which leads to develop standard design practices for digital circuits. VHDL was developed as a part of that effort. VHDL has become the industry standard language for describing digital circuits, largely because it is an official IEEE standard. The main purposes of VHDL are (1) Documentation (2) Simulation and (3) Hardware Synthesis. The development journey of VHDL is shown in Table 1.

Table 1 Development Journey of VHDL

S. No	Year	Development Path of VHDL
1	1981	Initiated by US DoD to address hardware life-cycle crisis
2	1983-85	Development of baseline language by Intermetrics, IBM and TI
3	1986	All rights transferred to IEEE
4	1987	Publication of IEEE Standard
5	1987	Mil Std 454 requires comprehensive VHDL descriptions to be delivered with ASICs
6	1994	Revised standard (named VHDL 1076-1993)
7	2000	Revised standard (named VHDL 1076 2000, Edition)
8	2002	Revised standard (named VHDL 1076-2002)
9	2007	VHDL Procedural Language Application Interface standard (VHDL 1076c-2007)
10	2009	Revised Standard (named VHDL 1076-2008)

The programming environment for implementing Carry look-ahead adder is based on VHDL. VHDL allows different levels of abstraction to be mixed in the same model. So a hardware model can be defined in terms of switches, gates, RTL, or behavioral code. The implementation steps of this design using VHDL and Spartan-3 FPGA board is shown Fig 4. Carry look-ahead adder is designed for N-bits using behavioral style and functionality of the design is verified via simulation with ModelSim 6.3f. The Xilinx ISE 12.2 software is used to synthesize the designs onto the Spartan 3E FPGA [24]-[25].



ISSN: 2319-5967

ISO 9001:2008 Certified

International Journal of Engineering Science and Innovative Technology (IJESIT)

Volume 2, Issue 4, July 2013

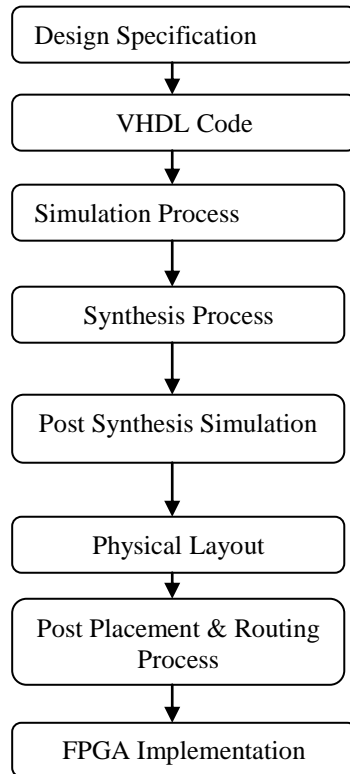


Fig. 4 VHDL Design Flow

C. VHDL Code of N-bit Carry look-ahead adder

The VHDL code of Carry look-ahead adder is written as:

```
LIBRARY ieee;
```

```
USE ieee.std_logic_1164.ALL;
```

```
ENTITY c_l_addr IS
```

```
PORT
```

```
(  
  x_in   : IN  STD_LOGIC_VECTOR (N DOWNTO 0);  
  y_in   : IN  STD_LOGIC_VECTOR (N DOWNTO 0);  
  carry_in : IN  STD_LOGIC;  
  sum    : OUT STD_LOGIC_VECTOR(N DOWNTO 0);  
  carry_out : OUT STD_LOGIC  
);
```

```
END c_l_addr;
```

```
ARCHITECTURE behavioral OF c_l_addr IS
```

```
SIGNAL h_sum      : STD_LOGIC_VECTOR (N DOWNTO 0);  
SIGNAL carry_generate : STD_LOGIC_VECTOR (N DOWNTO 0);  
SIGNAL carry_propagate : STD_LOGIC_VECTOR (N DOWNTO 0);  
SIGNAL carry_in_internal : STD_LOGIC_VECTOR(N DOWNTO 1);
```

```
BEGIN
```

```
  h_sum <= x_in XOR y_in;  
  carry_generate <= x_in AND y_in;  
  carry_propagate <= x_in OR y_in;
```



ISSN: 2319-5967

ISO 9001:2008 Certified

International Journal of Engineering Science and Innovative Technology (IJESIT)

Volume 2, Issue 4, July 2013

```
PROCESS (carry_generate,carry_propagate,carry_in_internal)
BEGIN
carry_in_internal(1) <= carry_generate(0) OR (carry_propagate(0) AND carry_in);
  inst: FOR i IN 1 TO (N-1) LOOP
    carry_in_internal(i+1) <= carry_generate(i) OR (carry_propagate(i) AND carry_in_internal(i));
  END LOOP;
carry_out <= carry_generate(N) OR (carry_propagate(N) AND carry_in_internal(N));
END PROCESS;
```

```
sum(0) <= h_sum(0) XOR carry_in;
sum(N DOWNT0 1) <= h_sum(N DOWNT0 1) XOR carry_in_internal(N DOWNT0 1);
END behavioral;
```

The entity part or the peripheral view and RTL Schematic of 8-bit Carry look-ahead adder is given in Fig. 5 and Fig. 6.

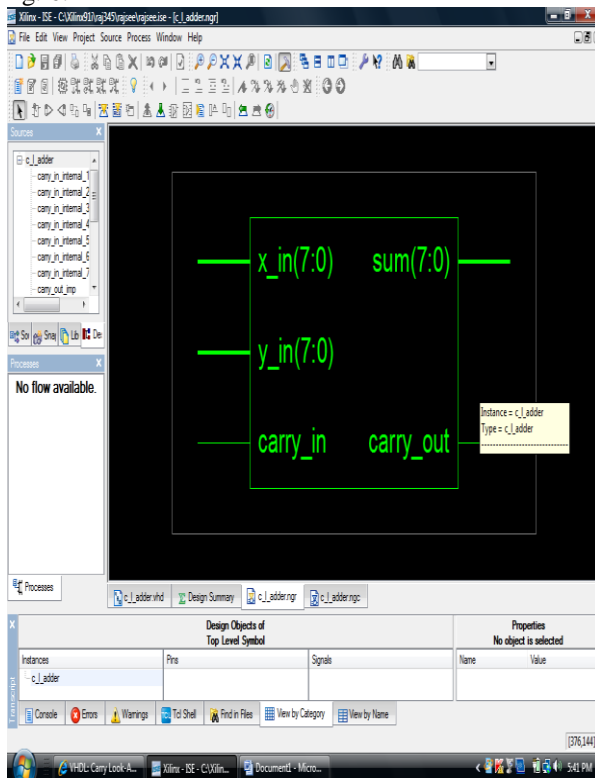


Fig. 5 Block Schematic of 8-Bit CLA

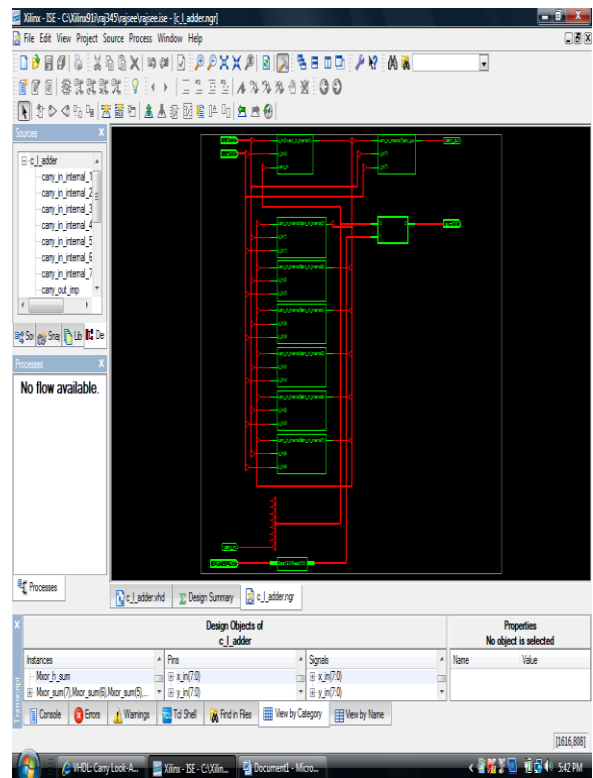


Fig. 6 RTL Schematic of 8-Bit CLA

IV. SIMULATION RESULT

It is now necessary to check whether the design works to meet the given specification to ensure that designed entity is correct. This is verified by the process of simulation and it is done by using ModelSim SE 6.3f and having the test bench and the behavioral design code for N-bit Carry look-Ahead adder in the same project folder. Simulation results show that the N-bit Carry look-Ahead adder implemented by the above described method and code, worked successfully for all the input combinations. For example,

```
x_in= 00000001=01H
y_in= 00000101=05H
carry_in=00000001=01H
sum= 00000111=07H
carry_out=00000000=00H
```

The modelSim result for above example is shown in Fig. 7.

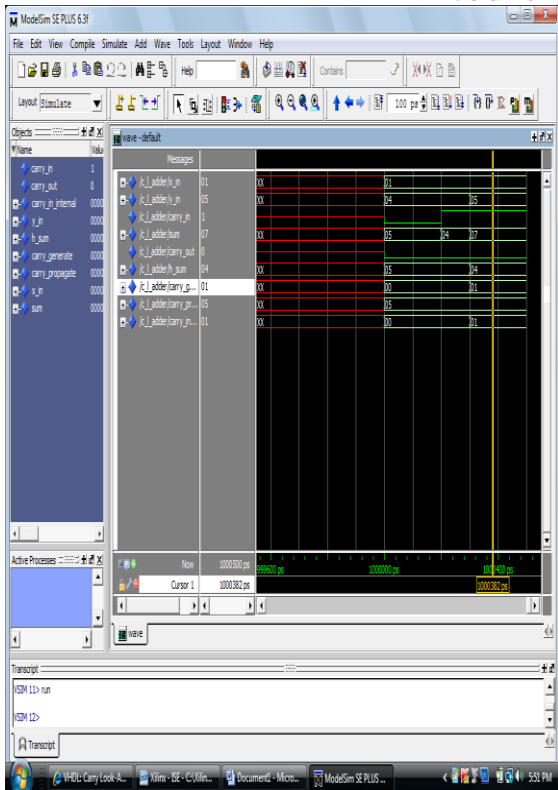


Fig. 7 Simulation Result of 8-bit CLA

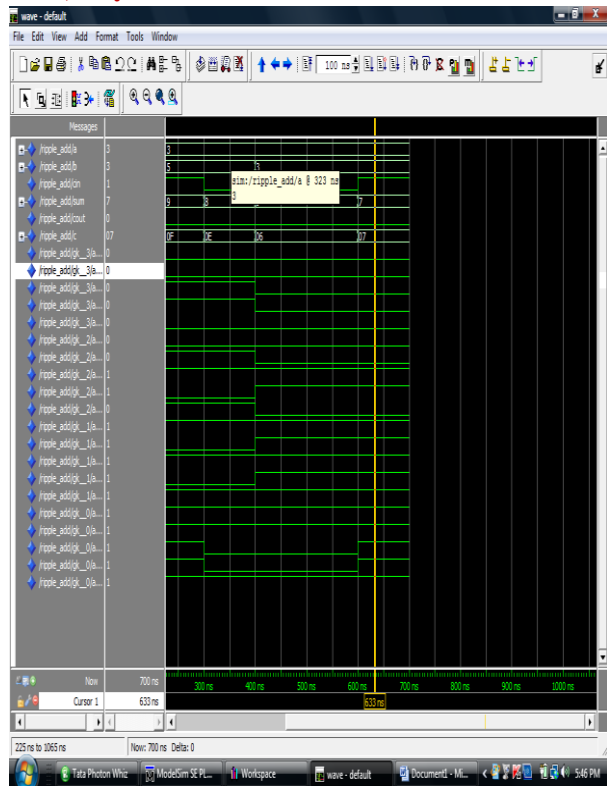


Fig. 8 Simulation Result of 4 bit CLA

Similarly simulation results for 4 bit and 16 bit carry-lookahead adder are shown in Fig. 8 and Fig. 9.

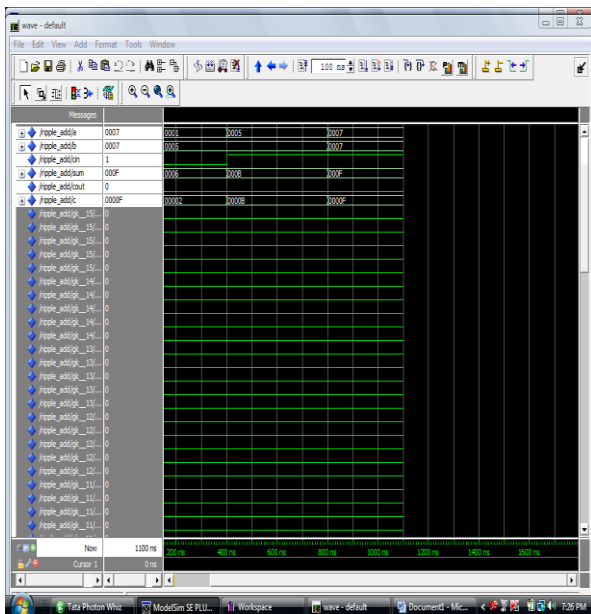


Fig. 9 Simulation result of 16 bit CLA

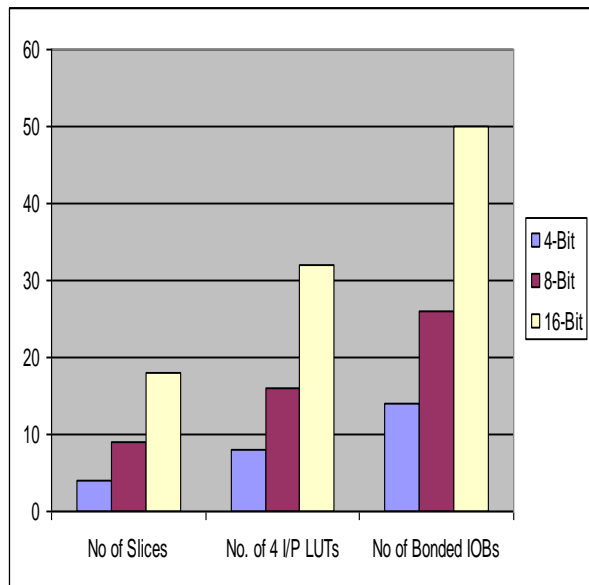


Fig. 10 Device Utilization Summary for Different Bit CLA

Device utilization summary obtained from the synthesis report of behavioral modeling of 4-bit, 8-bit and 16-bit carry-lookahead adder is shown Fig. 10 [26].

The total memory used and total path delay in designing of 4-bit, 8-bit and 16-bit carry-lookahead adder is as shown in Fig. 11.



ISSN: 2319-5967

ISO 9001:2008 Certified

International Journal of Engineering Science and Innovative Technology (IJESIT)

Volume 2, Issue 4, July 2013

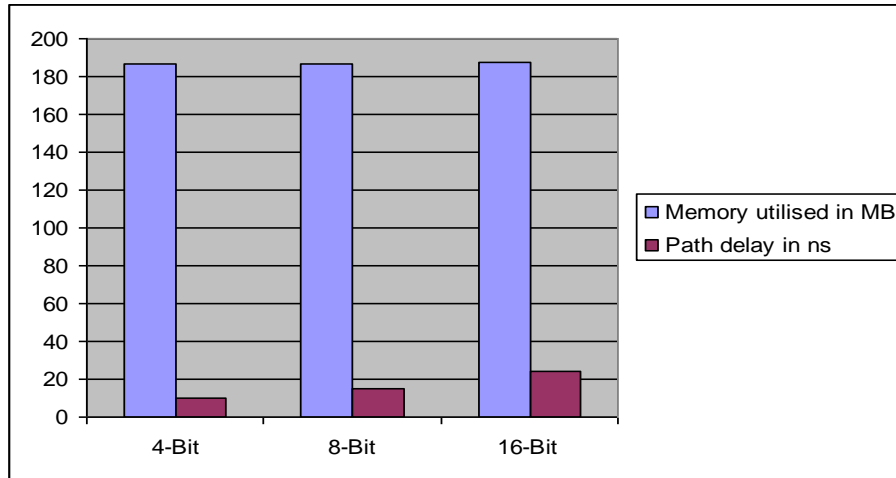


Fig. 11 Delay & Memory used in different Bit CLA

V CONCLUSION

Fast adders have the advantage to scale better with increasing word widths, since the delay of the n-bit fast adders is $O(\log(n))$ (carry-look ahead adder). It has been found that CLA occupies less area which considered from the utilization of LUTs, slices and over all gate count and total path delay shown in Fig.10 and Fig. 11. Different bit Carry-lookahead adder design using VHDL is successfully designed, simulated, tested and implemented onto the Spartan 3E FPGA. It is not possible to use CLA to realize constant delay for the wider-bit adders since there will be a substantial loading capacitance, and hence larger delay and larger power consumption. The CLA has the fastest growing area and power requirements with respect to the bit size. Similarly the other types of adders can also be designed and simulated using software and hardware used in this paper.

REFERENCES

- [1] I. Koren, Computer Arithmetic Algorithms, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 07632, 1993.
- [2] J. C. Lo, "A fast binary adder with conditional carry generation," IEEE Trans. Computers, vol. 46, no. 2, pp. 248-253, Feb. 1997.
- [3] S. Knowles, "A family of adders," Proceeding of 15th Symp. Computer Arithmetic, pp. 277-281, Jun. 2001.
- [4] V. G. Oklobdzija, B. R. Zeydel, H. Q. Dao, S. Mathew, and R. Krishnamurthy, "Comparison of high-performance VLSI adders in energy-delay space," IEEE Trans. VLSI Syst., vol. 13, no. 6, pp. 754-758, Jun. 2005.
- [5] J. Park, H. C. Ngo, J. A. Silberman, and S. H. Dong, "470 ps 64 bit parallel binary adder," Symp. VLSI Circuits, pp. 192-193, Jun. 2000.
- [6] T. Han and D. A. Carlson, "Fast area efficient VLSI adders," Proceeding of 8th Symp. Computer Arithmetic, pp. 49-56, May 1987.
- [7] J. M. Rabey, Digital Integrated Circuits, A Design Perspective, Prentice-Hall, 1996.
- [8] N. H. E. Weste and K. Eshraghian, Principles of CMOS VLSI Design, Addison Wesley, New York, NY, 1985.
- [9] Sung-Mo Kang and Yusuf Leblebici, CMOS Digital Integrated Circuits Analysis and Design, McGraw-Hill, 2002.
- [10] N. West and K. Eshraghian, Principles of CMOS VLSI Design: A System Perspective, 2nd edition, Addison Wesley, NY, pp. 513-536, 1992.
- [11] R. W. Doran, "Variants of an improved carry-lookahead adder," IEEE Trans. Computers, vol. 37, pp. 1110-1113, Sep. 1988.
- [12] G. A. Ruiz, Compact four bit carry look-ahead CMOS adder in multi-output DCVS logic, Electronic Letters, vol. 32, no. 17, pp. 1556-1557, Aug. 1996.
- [13] K. Ueda, H. Suzuki, K. Suda, H. Shinohara, K. Mashiko, et.al., "A 64-bit carry- lookahead adder using pass," IEEE Journal of Solid-State Circuits, vol. 31, no. 6, pp. 810-818, 1996.



ISSN: 2319-5967

ISO 9001:2008 Certified

International Journal of Engineering Science and Innovative Technology (IJESIT)

Volume 2, Issue 4, July 2013

- [14] P. K. Chan, M. D. F. Schlag, C. D. Thom Orson, and V.G. Oklobdzija, "Delay optimization of carry-skip adders and block carry-lookahead adders," 10th IEEE Proceedings on Computer Arithmetic, pp.154–164, June 26-28, 1991.
- [15] C-H. Huang, J-S. Wang, C. Yeh and C-J. Fang, "The CMOS carry-forward adders," IEEE Journal of Solid-State Circuits, vol. 39, no. 2, pp. 327-336, February 2004.
- [16] M.C.B. Osorio, C.A. Sampaio, A. I. Reis, R.P. Ribas, et.al., "Enhanced 32-bit carry- lookahead adder using multiple output enable-disable CMOS differential logic," SBCCI, pp. 181-185, 2004.
- [17] Y. T. Pai and Y. K. Chen, "The fastest carry-lookahead adder," Proceedings of the second IEEE International Workshop on Electronic Design Test and Applications (DELTA), 2004.
- [18] Jagannath Samanta, Mousam Halder and Bishnu Prasad De, "Performance analysis of high speed low power carry-lookahead adder using different logic styles," International Journal of Soft Computing and Engineering (IJSCE), vol. 2, issue 6, pp. 330-336, Jan- 2013.
- [19] K-H. Cheng, W-S. Lee and Y-C. Huang, "A 1.2V 500 MHz 32-bit carry-lookahead adder," 8th IEEE International Conference on Electronics Circuits and Systems, vol. 2, pp. 765–768, September 2-5, 2001.
- [20] P. Corsonello, S. Perri and G. Cocorullo, "Hybrid carry select statistical carry- lookahead adder," Electronics Letters, vol. 35, issue 7, pp. 549–551, April 1, 1999.
- [21] Santanu Maity, Bishnu Prasad De and Aditya Kr. Singh, "Design and implementation of low-power high-performance carry skip adder," International Journal of Engineering and Advanced Technology, vol. 1, issue 4, pp. 212-218, April 2012.
- [22] R. Barnes Earl and G. Oklobdzija Vojin, "New multilevel scheme for fast carry skip addition," in IBM Technical Disclosure Bulletin, vol.27, pp133-158 27, April, 2009.
- [23] Nagaraj Y, Shrinivas K, Veeresh K, Veeresh A, Madhu Patil and Dr. Chirag Sharma, "FPGA implementation of different adder architectures," International Journal of Emerging Technology and Advanced Engineering, vol. 2, issue 8, pp. 362-364, August 2012.
- [24] Douglas L. Perry, VHDL Programming by Example, fourth edition, McGraw-Hill, 2002.
- [25] Goldschlag, D. M., "A formal model of several fundamental VHDL concepts," Proceedings of the Ninth Annual Conference on Computer Assurance (COMPASS '94), pp. 177-181, Gaithersburg, Maryland, June 1994.
- [26] G. T. Zardalidis, and I Karafyllidis, "Design and simulation of a single-electron full-adder," IEEE Proceedings on Circuits, Devices and Systems, vol. 150, issue 3, pp. 173-177, June 6, 2003.

AUTHOR BIOGRAPHY

Rajender Kumar received the B.E. in Electronics and Communication Engineering from Career institute of technology and management and M.Tech. in VLSI Design and Embedded System from GJUS&T, Hisar, in 2001 and 2008, respectively. Currently he is working as Assistant Professor in BPS Women University, Sonipat. His research interests are mainly in Linear Integrated Circuit, VLSI Design and active filter design.

Sandeep Dahiya received his B.E. in Electronics Engineering (First class with distinction) in 2000 and M.Tech in Electronics and Communication Engineering in 2006.

Presently, he is working as an Assistant Professor in School of Engineering and Sciences, Bhagat Phool Singh Mahila Vishwavidyalaya Khanpur-Kalan, Sonapat since July 2009 having teaching experience of 12 years at UG and PG level. He has published 12 research papers in reputed International/National journals and conferences. He is also the recipient life member of Scientific and Professional societies such as Indian Society for Technical Education (ISTE), Computer Society of India (CSI) & The Institution of Electronics and Telecommunication Engineers (IETE). His current area of interest includes application of Information & Communication Technology in diverse fields, Engineering Education Development and simulation & modeling.