ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 7, September 2012

Design and analysis of 16-bit Full Adder using Spartan-3 FPGA

Rongali Aneel Kumar, B.N. Srinivasa Rao, R. Prasad Rao
Avanthi Institute of Engineering and Technology, Visakhapatnam.

***Abstract*— In ALU, adders plays an important role in arithmetic operations. Depending on the application, n-Bit adders to be designed like 8-bit, 16-bit, 32-bit and etc. Design of High-Speed adders is real challenge, especially in Semi-Custom designs, because the technique that is used in the design of various n-bit adders is different. In ALU, the use of ripple-carry adder takes most of the time in addition. The general strategy for designing fast adders is to reduce the time required to form carry signals. Adders that use this principle are called carry look-a-head adder. This paper presents the design and implementation of 16-bit adder by using 4-bit CLA. Here the most powerful ECAD tool VHDL is used for design and analyzed with SPARTAN-3 FPGA. The analysis is taken place in between VHDL primitive 4-bit adder and VHDL design of 4-bit CLA. Finally the result is concluded.**

***Keywords* — Semi-Custom full-adder, FPGA SPARTAN-3, ALU.**

I. Introduction

The aim of this paper is to implement a 4-bit CLA and compare with the VHDL primitive and to prove that VHDL primitive is implemented as optimized 4-bit CLA and can be used to develop n-bit adder. The overall delay is reduced and ultimately the speed is increased than the normal CLA design.

II. Overview of CLA Design

Various procedures and techniques are used to implement n-bit adders like Carry Save Adder, Carry Select Adder, Ripple Carry Adder, Carry Look a Head Adder etc.,. All these techniques have their own unique feature as an advantage. To design high speed adders CLA is used because the next level carry is depends on only primary inputs rather than the previous carry and primary inputs. The adders plays a key role in ALU because most of the arithmetic operations performed by the adders. For faster arithmetic operations, fast adders required. So that CLAs are widely used in ALU designs. Apart from the advantage of CLA, it has the disadvantage of circuit overhead. For every next level of carry, number of gates and number of inputs per a gate also increased. Due to the limitations of number inputs per a gate, the CLA is limited to 4-bit.

By using carry look-ahead mechanism, the propagation delay is reduced to four-gate level irrespective of the number of bits in the adder. The VDHL hardware description language is used to provide a gate level model and simulation of each design. A number of differing configurations of binary adders exist for inclusion into ALU design[1].

A. *Implementation Program*

VHDL is a general-purpose hardware description language, similar syntax to the C programming language. VHDL allows different levels of abstraction to be mixed in the same model. So a hardware model can be defined interms of switches, gates, RTL, or behavioral code. Now a days most digital design of processors and related hardware system is designed using a hardware description language. Such a language serves two purposes, first it provides on abstract description of the hardware to simulate and debug the design. Second, with use of logic synthesis and hardware compilation tools, this description can be compiled into the hardware implementation.

VHDL provides the concept of module. A module is the basic building block in VHDL. A module can be implemented in terms of the desired design algorithm without concern for the hardware implementation details [2]. These modules can be substituted in place of the 16-bit full adder modules described before without changing any other component of the simulation. The simulation results will be unchanged.

III. HIGH SPEED ADDER

The general strategy for designing fast adders is to reduce the time required to form carry signals. One approach is to compute the input carry needed by stage i directly from carry like signals obtained from all the preceding stages i-1,i-2,……0,rather than waiting for normal carries to ripple slowly from stages to stages. Adders that use this principle are called carry look-ahead adders. An n-bit carry look-ahead adder is formed from n stages , each of which is basically a full adder modified by replacing its carry output line ci by two auxiliary signals called gi and pi or generate and propagate, respectively ,which are defined by the following logic equation (1);

$$g_i = x_i y_i \qquad p_i = x_i + y_i \qquad (1)$$

The name generate comes from the fact that stage i generates a carry of 1(c_i=1) independent of the value of c_{i-1} if both xi and yi are 1; that is, if $x_i y_i$=1.Stage i propagates c_{i-1}; that is makes c_i=1 in response to c_{i-1}=1 if x_i or y_i is 1_ in order words, if x_i+y_i=1. Now the usual equation $c_i = x_i y_i + x_i c_{i-1} +$

y_ic_{i-1}, denoting the carry signal c_i to be sent to stage i+1, can be rewritten in terms of g_i and p_i.

$$c_i = g_i + p_ic_{i-1} \quad (2)$$

Similarly, c_{i-1} can be expressed in terms of g_{i-1}, p_{i-1} and c_{i-2},

$$c_{i-1} = g_{i-1} + p_{i-1}c_{i-2} \quad (3)$$

On substituting Equation (3) into Equation (2),

$$c_i = g_i + p_ig_{i-1} + p_ip_{i-1}c_{i-2} \quad (4)$$

Continuing in this way, c_i can be expressed as a sum-of-products function of the p and g outputs of all the preceding stages. For example, the carries in a four-stage carry look-ahead adder are defined as follows:

$$c_0 = g_0 + p_0.c_{in}$$
$$c_1 = g_1 + p_1.g_0 + p_1.p_0.c_{in}$$
$$c_2 = g_2 + p_2.g_1 + p_2.p_1.g_0 + p_2.p_1.p_0.c_{in}$$
$$c_3 = g_3 + p_3.g_2 + p_3.p_2.g_1 + p_3p_2.p_1.g_0 + p_3p_2.p_1.p_0.c_{in} \quad (5)$$

Fig. 1 shows the general form of a carry look-ahead adder circuit designed in this way [3].

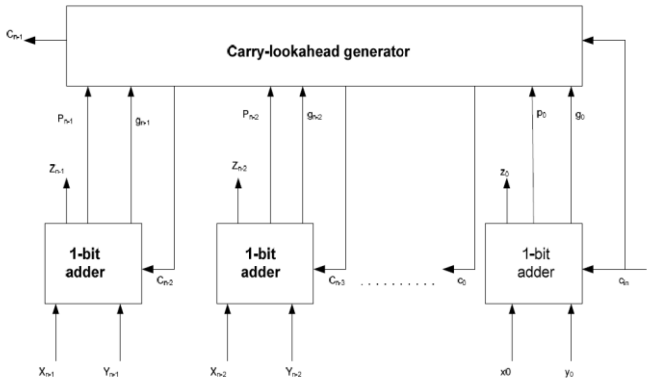

Fig. 1 Overall Structure of Carry Look-ahead Adder

A. Adder Expansion

The methods of handing carry signals in the two main combinational adder designs considered so far, namely, ripple carry propagation and carry look-ahead Fig. 1 can be extended to larger adders of the kind needed to execute add instructions in a 32-bit computer. If the n 1-bit (full) adder's stages are replaced in the n-bit ripple-carry design with n k-bit adders, an nk-bit adder can be obtained. Four 4-bit adders such as the 4-bit carry look-ahead circuit of Fig. 2 can be connected in this way to form the 16-bit adder appearing in Fig. 3.

This design represents a compromise between a 16-bit stage ripple-carry adder, which is cheap but slow, and a single-stage 16-bit carry look-ahead adder, which is fast, expensive, and impractical because of the complexity of its carry-generation logic. The circuit of Fig.3 effectively combines sets of four x_iy_i inputs into groups that are adder via carry look-ahead; the results computed by the various groups are then linked via ripple carries [3].

The components designed for 1-bit addition have been replaced with similar but larger components intended for 4-bit addition. The expanded design of Fig. 1 can be got. Again 1-bit adders with 4-bit adder are being replaced, but now each adder stage produces a propagate-generate signal pair pg instead of cout and a carry look-ahead generator converts the four sets of pg signals to the carry inputs required by the four stages. The "group" g and p signals produced by each 4-bit stage are defined by [3].

$$g = x_iy_i + x_{i-1}\ y_{i-1}(x_i+y_i) + x_{i-2}y_{i-2}(x_i + y_i)\ (x_{i-1} + y_{i-1}) + x_{i-3}y_{i-3}(x_i+y_i)\ (x_{i-1} + y_{i-1})\ (x_{i-2} + y_{i-2}) \quad (6)$$
$$p = (x_i + y_i)\ (x_{i-1} + y_{i-1})\ (x_{i-2} + y_{i-2})\ (x_{i-3} + y_{i-3}) \quad (7)$$

It is not hard to show that the logic to generate the group carry signals cout, c11, c7 and c3 in Fig. 4 is exactly the same as that of the carry look-ahead generator of Fig. 2.

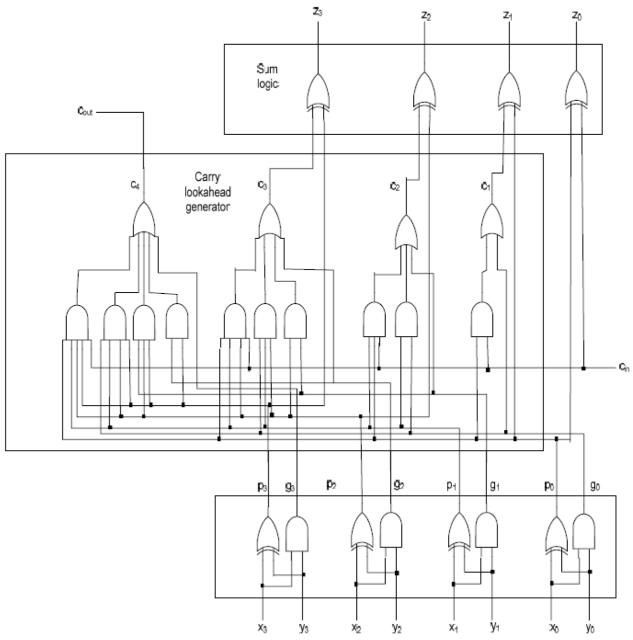

Fig. 2 A 4-Bit Carry Look-ahead Adder

B. 4-bit CLA Structural model Design

To achieve this, individual gates are designed in behavioral model and all the components are interconnected in structural model in VHDL as shown in the fig.2. By using post layout simulation we can find out the actual delays of this model. The technology schematic and simulation results are shown in fig 3 and 4 respectively.

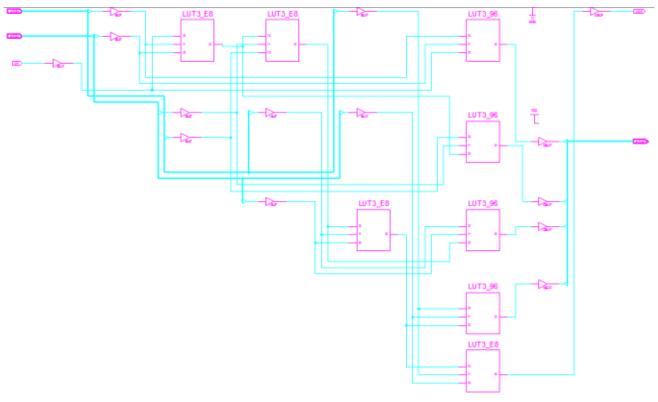
Fig. 4 4-bit CLA Technology Schematic

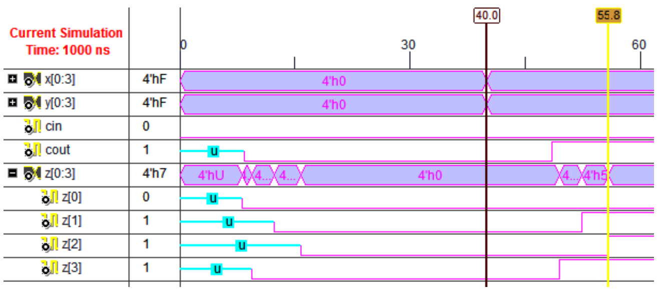

Fig. 5 4-bit CLA Post layout simulation

The following figures shows the technology schematic and post layout simulation results of VHDL primitive 4-bit CLA.

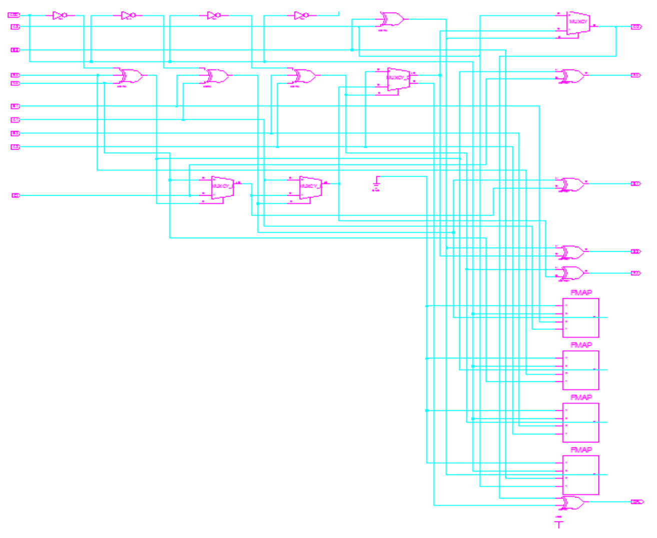
Fig. 6 4-bit CLA Technology Schematic (VHDL Primitive)

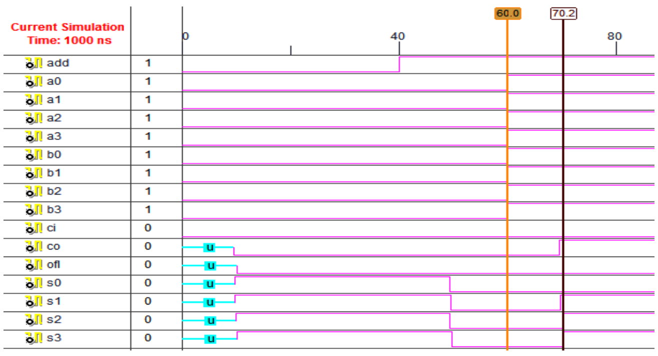

Fig. 7 4-bit CLA (VHDL Primitive)Post layout simulation

V. 16-BIT ADDER DESIGN

The design goal is to minimize the number of gates used; operation speed is not of concern. The circuit is required in several versions that handle different data word sizes, including 4, 8 and 16.

Assume that we have standard gate-level and 4-bit register-level components available as building blocks. The lowest cost adders employ ripple-carry propagation.

Four 4-bit CLAs are interconnected as ripple adder form as shown in the fig.8. here the carry is propagated to the next stage 4-bit CLA. The internal carry propagation delay of 4-bit adder is reduced by using the technique CLA. So that the overall speed of the 16-bit Adder is increased.

Moreover the VHDL Primitive is designed by using faster design elements. So that the sum delay as well as the carry delay is very much improved. For faster Full Adders designs VHDL Primitive can be used to reduce design time and to achieve the speed. The comparison is shown in the table 1.

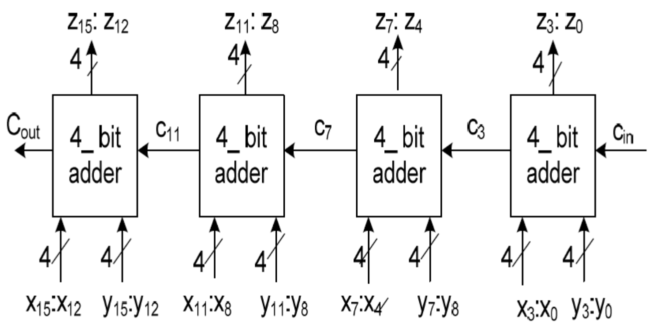

Fig. 8 16-Bit Composed of 4-Bit Adders Linked by Ripple Carry Propagation

The hierarchical design is shown in fig 9. The same concept is used in VHDL Primitive also but the number of 4-bit CLA internal stages are reduced.

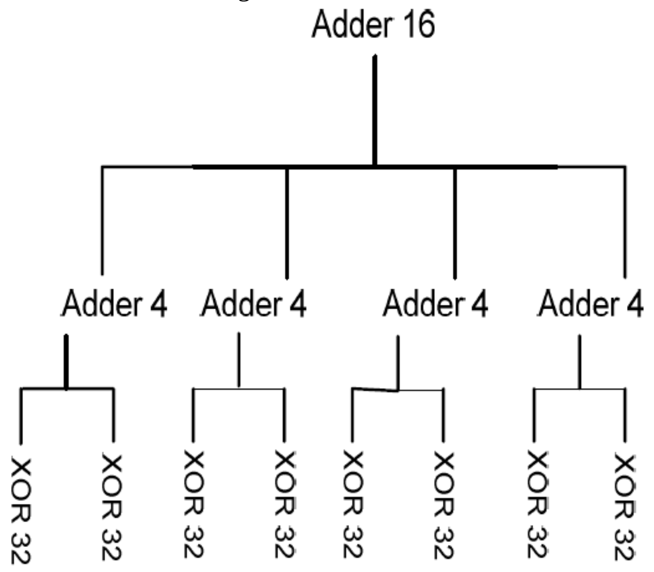

Fig. 9 Hierarchical Level for 16-Bit Carry Look-head Adder

The following table describes the comparison between general CLA design and VHDL Primitive. From this easily we can understand that VHDL Primitive Adder is faster than any other adder, because the VHDL Primitive is optimized design.

Design name/Delay	Sum Delay(ns)	Carry Delay(ns)	Final Delay(ns)
General CLA(4-bit)	15.8	8.4	15.8
VHDL Primitive	10.2	9.5	10.2
16-bit Adder(Using General CLA)	41	33.6	41
16-bit Adder(Using VHDL Primitive)	38.7	38	38.7

Table 1. Comparison between design elements

VI. CONCLUSION

In this paper 16-bit Adder is designed and compared with another 16-bit Adder designed by using VHDL Primitive and found 2.3ns difference in between to designs. So that is concluded that 16-bit Adder design using VHDL Primitive is faster than any other design in Semi-custom, because the VDHL Primitive is optimized 4-bit CLA.

REFERENCES

[1] Andrew S. Tanenbaum, Structure Computer Organization, Fourth edition, Prentice_Hall, Inc., 1999.

[2] Samir Palnitkar, Verilog HDL: A Guide to Digital Design and Synpaper, Sun Microsystems, Inc., 1996.

[3] William Stallings, Computer Organization & Architecture, Sixth edition.

[4] David A. Patterson, John L. Hennessy, Computer Organization and Design, (The Hardware /Software Interface), Third edition, Elsevier Inc., 2005.

[5] May Phyo Thwal, Khin Htay Kyi, and Kyaw S war Soe “Implementation of Adder-Subtractor Design with VerilogHDL, World Academy of Science, Engineering and Technlogy 39 2008.

R. Aneel Kumar was born in Vizianagaram, Andhra Pradesh, India. He has received his B.Tech degree in Electronics and Communication Engineering from Jawaharlal Nehru Technological University, Kakinada, India and pursuing M. Tech in VLSI System Design from Jawaharlal NehruTechnological University, Kakinada, India. Currently he is a student in Avanthi Institute of Engineering and Technology Makavarapalem, Visakhapatnam, Andhra Pradesh, India. His area of interests VLSI Semi-Custom design.

B.N.Srinivasa rao received his B.Tech degree in Electronics and Comunication Engineering from JNT University, Hyderabad, India and M.Tech in VLSI System Design from JNT University,Hyderabad, India. He is currently working as an Assistant Professor in Avanthi Institute of Engineering and Technology, Visakhapatnam, Andhra Pradesh, India. He has 5 years teaching and 9 years industrial experience. He has 7 publications in various International conferences. His area of interest VLSI Semi and full custom design. He guided many projects for B.Tech and M.Tech students.

R. Prasada rao received his B.Tech degree in Electronics and Comunication Engineering from Andhra University, Visakhapatnam, India and M.Tech in VLSI System Design from JNT University,Hyderabad, India. He is currently working as an Assistant Professor in Avanthi Institute of Engineering and Technology, Visakhapatnam, Andhra Pradesh, India. He has 8 years teaching experience. He has 2 publications in various International conferences. His area of interest is Digital Electronics, Signals and Systems and Signal Processing. He guided many projects for B.Tech and M.Tech students.