INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 4.295

(Volume 4, Issue 3)

Available online at: www.ijariit.com

Design of high-speed multiplier by using carry select adder

Sai Bhavya
saibhavya073@gmail.com
Geethanjali Institute of Science and Technology, Nellore, Andhra Pradesh

Mahesh Kumar
maheshkumarmulakala@gmail.com
Geethanjali Institute of Science and Technology, Nellore, Andhra Pradesh

ABSTRACT

An adder is a digital circuit that performs addition of numbers. In many computers and other kinds of processors in the arithmetic logic units, adders are used. In other parts of the processor, they are also utilized. Where they are used to calculate addresses, table indices, increment and decrement operators, and similar operations. Although adders can be constructed for many number representations, such as BINARY-DECODED DECIMAL or EXCESS-3, the most common adders operate on binary numbers. We designed an adder which is of high speed and applied this to a new multiplier for better performance multiplier by using CARRY SELECT ADDER in this project. In arithmetic operations, addition and multiplication are having a major role. When the number of bit increases, the complexity of the adder circuits increases and the speed performance decreases. The delay will be very much reduced proposed carry select adder based multiplier on comparing with carrying look ahead adder based multiplier, and the carry save adder based multiplier. The code is written in VHDL and Verilog and synthesized the design in Xilinx ISE 14.1.

Keywords: *Shift register, Single bit flip-flop, Multi-bit flip-flop, Carry look ahead adder, Carry select adder, Carry save adder, VLSI, VHDL, Spartan*

1. INTRODUCTION

A digital designer has to concentrate on many criteria like Circuit speed, power consumption, area, and cost. The fundamental arithmetic operations like addition and multiplication are the main optimizations while designing the digital circuits. For larger applications, our design concentrates on multiplication of binary numbers.

For the multiplication operation, adding as well as shifting of bits is necessary. Considering these two operations, we have designed a high-speed multiplier by using carry select adder for addition and multi-bit flip-flop based shift register for shifting of bits.

In this paper, we have shown the comparison among the adders for multiplication based on the time. On comparison with carrying look ahead adder (CLAA) and carry select adder (CSLA), carry save adder (CSA) based multiplier is less complex and results have shown that CSA based multiplier is very faster than the other two multipliers.

A shift register is a very important digital building block. It has a large number of applications. Registers are often used to temporarily store binary information appearing at the output. Shift registers are the logic types which are used basically for the storage and transfer of digital data. The basic storage elements are the flip-flops. The most of the registers use D flip-flops due to its simplicity [1].

Basically a flip-flop stores a single bit data. In our proposed design the flip-flop stores a multi-bit data. By using the multi-bit flip flops, we have shown a shift register which is serial in serial out (SISO). Finally, we have designed a high-speed multiplier by using carry select adder and multi-bit flip-flop shift register.

This paper is scheduled as first it discusses and compares the about the different adders in sections II, III and IV. After that in section. V It illustrates the simulation results and Section VI. It concludes the paper.

2. CARRY LOOK AHEAD ADDER

Carry look ahead adder generates carries parallel at a time. We should have the extra circuitry for this. Carry Look Ahead adder needs more area and the power using by the whole circuit is high, because of these extra circuits and storage elements. The below equations describes the pattern how a carry look ahead generates the carries [2]. Figure 1 shows the block diagram for carrying look ahead adder.

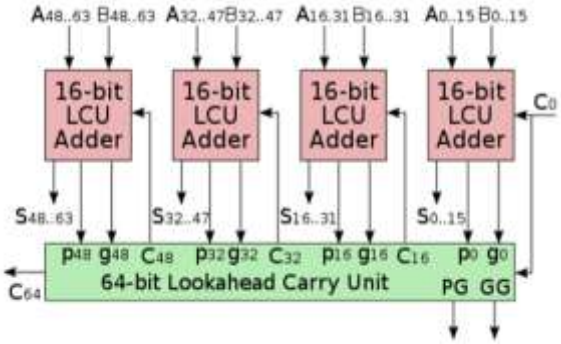

Fig. 1: Carry look ahead adder

Let G_i be the generating the function and P_i be the propagating function then we can write the carries as:

$$C_{i+1}=A.B+ (A \text{ XOR } B).C_i, \text{ Where} \quad (1)$$

$$G_i = A.B \quad (2)$$

$$P_i= (A \text{ XOR } B).C_i \quad (3)$$

From the equations (1) (2) (3), we can understand that the sum of adder circuit does not depend on the carry generation so we can generate all the carries at a time as

$C_1= A_0.B_0 + (A_0 \text{ XOR } B_0). C_0$

$C_2=A_1.B_1 + (A_1 \text{ XOR } B_1). C_1$

$C_3=A_2.B_2 + (A_2 \text{ XOR } B_2). C_2$ ……..

$C_{64}=A_{63}.B_{63} + (A_{63} \text{ XOR } B_{63}).C_{63}$

The sum of the adder is written as

SUM= A XOR B XOR C

3. CARRY SELECT ADDER

Carry select adder is a different from the carry look ahead adder, in which we select the carry as 0 once and again select the carry as 1. After that, we perform the addition operation for the both cases and give these outputs to the 2:1 multiplexer. Finally we receive the single output. So, we do not need to wait for previous addition result to the next step [3]. The figure 2 represents the carry select adder. The multiplexer output equations for sum and carries are:

Sum function is as follows

If carry =1 then,

$SUM_0= A_0$ xor B_0 xor Carry

If carry =0 then,

$SUM_0=A_0$ xor B_0

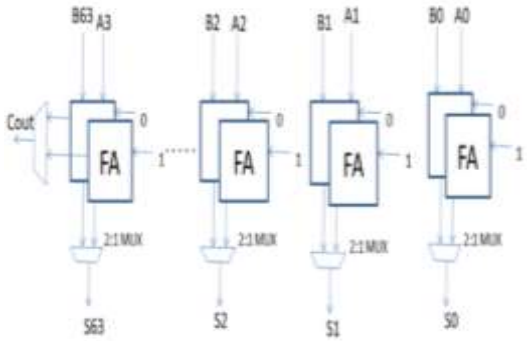

Fig. 2: Carry select adder

These two SUMs will be given to the multiplexer. Here, to perform the operation need not to wait for the previous stage carry. So, the time taken to perform the addition operation decreases. We can see a significant improvement in time.

4.CARRY SAVE ADDER

It further can decrease the timing performance of the addition by using carry save addition. In this carry save addition, the carry is neglected first and the only sum will be considered [4]. Let us see one example for the carry save addition with two numbers A and B:

A: 3 4 5 6
B: 4 3 6 5
Sum: 7 7 1 1

The carries generated by the addition are neglected here are 0 0 1 1. Now these carries can be shifted to the left and add to the sum, as:

Shifted Carries: 0 1 1
Sum: 7 7 1 1
Total sum: 7 8 2 1

Carry save adder does not have more complexity as carrying select adder and carry look ahead adder. So, we have implemented high-speed multiplier by using the carry save adder. Even though it needs a large number of registers to store the carries separately, this can be advantageous in its speed performance. The main advantage of carrying save addition is it can add three bits at a time. But in our design, we are not considering multi operand terminated addition. The third bit is considering as zero. Figure 3 represents the carry save adder block diagram.

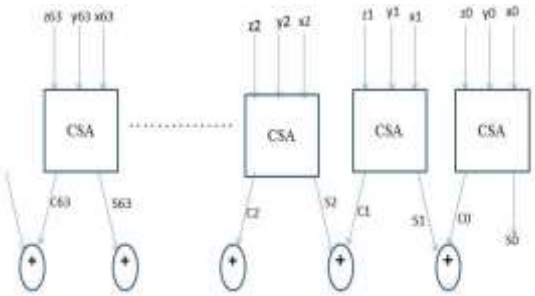

Fig. 3: Carry save adder

5. SIMULATION RESULTS

Fig. 4: Simulation results of ripple carry adder

Fig. 5: Simulation results of carrying look ahead adder

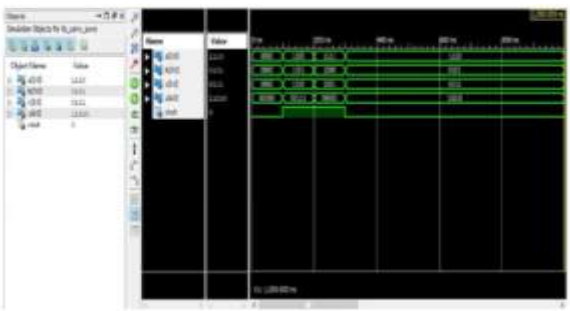

Fig. 6: Simulation results of carrying select adder

Fig. 7: Simulation results of multiplier

6. CONCLUSION

A design of high-speed multiplier with different adders have shown in this project. VHDL and Verilog language are used to simulate and synthesize the multiplier. By using the carry select adder improved the overall speed of the design. We have shown that proposed multiplier increased its speed by 93.5 % when compared with the CLAA and 88% when compared with CSA. A fast adder can significantly improve the performance of the multiplier. Overall the speed performance of the multiplier based on carrying, the select adder is high.

7. REFERENCES

[1] Anand Kumar "Fundamentals of digital circuits" Prentice hall India publications, 2004.

[2] Jagannath Samanta, Mousam Halder, Bishnu Prasad De "Performance Analysis of High-Speed Low Power Carry Look-Ahead Adder Using Different Logic Styles International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-6, Jan- 2013

[3] Deepthi Obul Reddy, P.Ramesh Yadav "Carry Select Adder with Low Power and Area Efficiency". International Journal of Engineering Research and Development, Volume 3, Issue 3 (August 2012), PP. 29-35.

[4] D.R Lutz, D.N. Jaya Simha, "The power of carrying save addition", Department of Computer and Information Science, The Ohio State University, 1991.

[5] Hasan Krad* and Aws Yousif "Design and Implementation of a Fast Unsigned 32-bit Multiplier Using VHDL" Qatar University/Department of Computer Science & Engineering, Doha, Qatar.

[6] W. Stallings, Computer Organization and Architecture Designing for Performance, 71h ed., Prentice Hall, Pearson Education International, USA, 2006, ISBN: 0-13-185644-8.

[7] V.Vijayalakshmil, R.Seshadd, Dr.S.Ramakrishnan3 "Design and Implementation of 32 Bit Unsigned Multiplier Using CLAA and CSLA" IEEE 2013.

[8] G. Prakash, K.Sathishkumar, B. Sakthibharathi S. Saravanan, R. Vijaysai "Achieveing reduced area by multi-bit flip flop design" International Conference on Computer Communication and Informatics (ICCCI -2013), Jan. 04 – 06, 2013, Coimbatore, INDIA, 2013

