THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

Performance, Analysis and Comparison of 8 Bit Adders

Maruti L. Doddamani
IV Semester, M.Tech (Digital Electronics), Department of Electronics & Communication
S.D.M. College of Engineering and Technology, Dharwad, Karnataka, India
Vinayak Shanwad
Assistant Professor, Department of Electronics & Communication
B.V.B College of Engineering and Technology, Hubli, Karnataka, India

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, Kogge Stone Adder etc exist numerous adder implementations each with good attributes and some drawbacks. This paper focuses on the implementation and simulation of 8-bit ripple carry adder, carry look-ahead adder and Kogge stone adder based on Very High Speed Integrated Circuit Hardware Description Language (VHDL) and compared for their performance. Adders play a very important role in every application that involves digital processing or an embedded application, in this project the a set of different adders will be designed including Kogge stone adder (KSA), KSA is designed in such a way that it consumes less area and delay than the existing techniques, projects simulation will be done on Xilinx12.2 Design Suite and Modelsim Simulator 6.3f.

Key words*: Carry look Ahead Adder, Kogge Stone adder, Ripple Carry Adder, VHDL Simulation, ModelSim, Xilinx 12.2*

1. Introduction

In electronics, an adder or summer is a digital circuit that performs addition of numbers. 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 similar. Although adders can be constructed for many numerical representations, such as binary-coded decimal or excess-3, the most common adders operate on binary numbers. In cases where two's complement or ones' complement is being used to represent negative numbers, it is trivial to modify an adder into an adder–subtractor. Other signed number representations require a more complex adder.

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.

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]. Moreover, there are various types of adders such as Ripple Carry Adder (RCA), Carry-Look ahead Adder (CLA), Kogge Stone Adder discussed in brief.

2. Literature Survey

Much of the research efforts of the past years in the area of digital electronics have been directed towards increasing the speed of digital system. Recently the requirement of portability and the moderate improvement in battery performance indicates that the power dissipation is one of the most critical design parameter.

The three most widely accepted metrics to measure the quality of a circuit or to compare various circuit styles are area, delay and power dissipation. Portability imposes a strict limitation on power dissipation while still demands high computational speeds. Hence, in recent VLSI Systems the power-delay product becomes the most essential metric of performance. The reduction of the power dissipation and the improvement of the speed require optimizations at all levels of the design procedure. Since, most digital

circuitry is composed of simple and/or complex gates; we study the best way to implement adders in order to achieve low power dissipation and high speed.
In this paper, the practical issues involved in designing and simulation tree-based adders on FPGAs are described. An efficient testing strategy for evaluating the performance of these adders is discussed. Several tree-based adder structures are implemented and characterized on a FPGA and compared with the Ripple Carry Adder (RCA) and the Carry look Ahead Adder (CSA) and Kogge Stone Adder (KSA). Finally, some conclusions and suggestions for improving FPGA designs to enable better tree-based adder performance are given.

3. 8 Bit Ripple Carry Adder

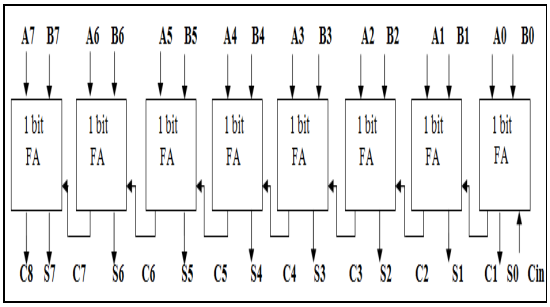

Figure 1: 8 bit Ripple carry Adder

It is possible to create a logical circuit using multiple full adders to add N bit numbers. Each full adder inputs a Cin, which is the Cout of the previous adder. This kind of adder is called a ripple-carry adder, since each carry bit "ripples" to the next full adder. The layout of a ripple-carry adder is simple. The disadvantage is each full adder must wait for the carry bit to be calculated from the previous full adder as shown in figure1.

4. 8 Bit Carry Look Ahead Adder

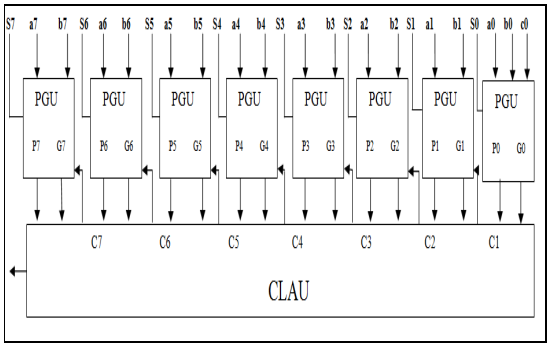

Figure 2: 8 bit Carry Look Ahead Adder

A Carry-Look Ahead Adder (CLA) is a type of adder used in digital logic as shown in figure 2. A carry-look Ahead 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-look Ahead adder calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger value bits.
Example: Kogge-Stone adder and Brent-Kung adder.
To reduce the computation time, faster ways to add two binary numbers by using carry-look ahead adders. Its implementation is based on the generate and propagate concept, which gives the circuit higher speed than its carry ripple adder. A diagram of a 8 bit carry look ahead adder is shown in fig. Its implementation is based on the generate and propagate concept, which gives the circuit higher speed than its carry ripple adder.
Consider two input bits a and b, then generate (g) and propagate (p) signals are defined as
g= a AND b
p= a XOR b
Such signals is computed in advance, because neither depends on the carry bit
In general, g(i)= a(i) AND b(i)
p(i)= a(i) XOR b(i)
Calculation for sum & carry bit:
Let us consider the carry vector c=c (i-1)…c (1) c (0). The carry bits can be computed from g and p
C(i+1)=g(i)+ p(i) c(i)
S(i)= p(i) XOR c(i)

5. 8 Bit Kogge Stone Adder

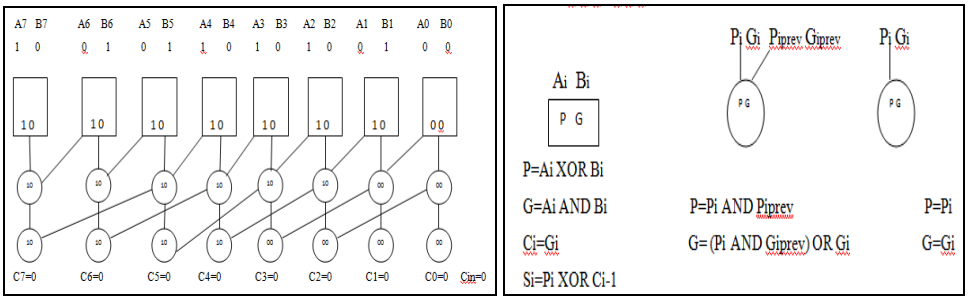

Figure 3: 8 bit kogge stone adder

The Kogge–Stone adder is a parallel prefix form carry look-ahead adder as shown in figure 3. It generates the carry signals in O(log n) time, and is widely considered the fastest adder design possible. It is the common design for high-performance adders .It takes more area to implement than the Brent–Kung adder, but has a lower fan-out at each stage, which increases performance. Wiring congestion is often a problem for Kogge–Stone adders as well.

The Kogge–Stone adder concept was developed by 'Peter M. Kogge' and 'Harold S. Stone' in 1974.A 8-bit Kogge–Stone adder is shown to the right. Each vertical stage produces a "propagate" and a "generate" bit, as shown. The culminating generate bits (the carries) are produced in the last stage (vertically), and these bits are XOR'd with the initial propagate after the input (the red boxes) to produce the sum bits. The advantage of Kogge stone adder is widely considered the fastest adder design possible, common design for high-performance adders in industry and has a lower fan-out at each stage, which increases performance.

The Kogge-Stone adder is classified as a

Parallel prefix adder since the generate and the propagate signals are pre-computed. In a tree-based adder, carries are generated in tree and fast computation is obtained at the expense of increased area and power. The main advantage of this design is that the carry tree reduces the logic depth of the adder by essentially generating the carries in parallel.

The table 2 shows the comparison of various adders with respect to the delay i.e., Ripple carry adder has 20.394 ns, carry look ahead adder 18.644ns and Kogge stone Adder has 16.509ns.

6. Results

The Simulation result for 8 bit Ripple Carry Adder is shown in fig 4.

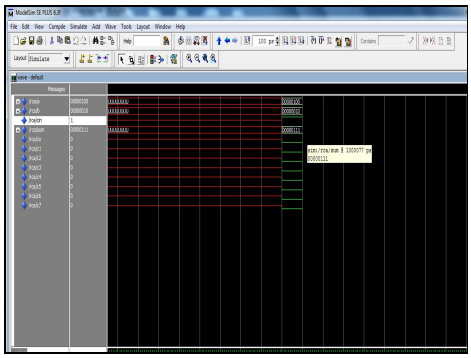

Figure 4: 8 bit Ripple carry Adder

- The Simulation result for 8 bit Carry Look Ahead Adder is shown in fig 5.

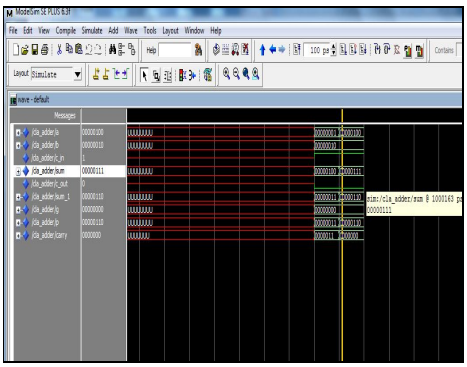

Figure 5: 8 bit Carry Look Ahead Adder

- The Simulation result for 8 bit Kogge Stone Adder is shown in fig 6.

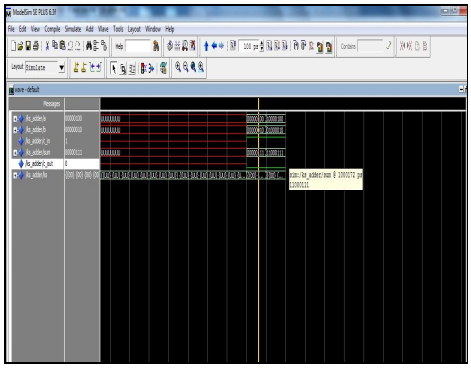

Figure 6: 8 bit Kogge Stone Adder

ADDERS	Results obtained Delay (ns)
Ripple carry adder	20.394
Carry look ahead Adder	18.644
Kogge Stone Adder	16.509

Table 1: Comparison of various Adders with delay

7. Conclusion

As shown in table 1 describes that Kogge Stone Adder has least delay with 16.509ns compared to ripple carry adder and carry look Ahead adder. So I conclude that Kogge stone Adder is efficient in the present market and consume very less area and produce less delay. By this, an efficient 8 bit kogge stone Adder is designed, utilized and implemented on FPGA.

8. References

1. J. C. Lo, "A fast binary adder with conditional carry generation," IEEE Trans. Computers, vol. 46, no. 2, pp. 248-253, Feb. 1997.
2. S. Knowles, "A family of adders," Proceeding of 15th Symp. Computer Arithmetic, pp. 277–281, Jun. 2001.
3. 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.
4. Sung-Mo Kang and Yusuf Leblebici, CMOS Digital Integrated Circuits Analysis and Design, McGraw- Hill, 2002.
5. May Phyo Thwal, Khin Htay Kyi, and Kyaw S war Soe "Implementation of Adder-Subtractor Design with Verilog HDL, World Academy of Science, Engineering and Technology 39 2008.
6. T. Han and D. A. Carlson, "Fast area efficient VLSI adders," Proceeding of 8th Symp. Computer Arithmetic, pp. 49–56, May 1987.
7. R. W. Doran, "Variants of an improved carry- lookahead adder," IEEE Trans. Computers, vol. 37, pp. 1110–1113, Sep. 1988.
8. J. M. Rabey, Digital Integrated Circuits, A Design Perspective, Prentice-Hall, 1996.
9. C. Nagendra, M. Irwin, and R. Owens, "Area-time- power tradeoffs in parallel adders," Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on, vol. 43, no. 10, pp. 689–702, Oct 1996
10. N. H. E. Weste and K. Eshraghian, Principles of CMOS VLSI Design, Addison Wesley, New York, NY, 1985

.