International Journal of Electrical and Electronic Engineering & Telecommunications

ISSN 2319 – 2518 www.ijeetc.com
Vol. 4, No. 1, January 2015

Research Paper

DESIGN AND IMPLEMENTATION OF A HIGH SPEED 64 BIT KOGGE-STONE ADDER USING VERILOG HDL

Raghumanohar Adusumilli[1]* and Vinod Kumar K[1]

*Corresponding Author: Raghumanohar Adusumilli, ✉a.r.manohar99@gmail.com

Adder is one among the basic arithmetic operates. Currently implementing a high speed VLSI style could be an important topic and as adders are utilized in various fields of applications, coming up with a high speed adder is one among the necessary facets. In this paper we designed and enforced a high speed KoggeStone parallel prefix adder of 8, 16, 32 and 64 bit to be meted out and compared with Carry Look Ahead adder (CLA) and Carry Skip Adder (CSA) and also pointed out the potency of KoggeStone adder with relevance delay victimization using Xilinx ISE 14.7.

Keywords: 64 bit high speed adder, KoggeStone adder, CLA, CSA

INTRODUCTION

Adders are the basic build blocks in almost all electronic applications. There are different types of adders available today such as Ripple carry adder, carry look ahead adder, carry skip adder and many more. There are lots of researches going on in this field, mainly concentrating on parameters like less delay, low power consumption and reducing chip area. In this paper, mainly we are concentrating to increase the speed of operation to achieve fast calculation operation as in the current generation microprocessors many number of instructions are executed per second. But it is very difficult to achieve all the three parameters (i.e., area, power and speed) in a single design, one or the other constraint must be sacrificed to achieve the other. In this design we concentrated on the speed parameter and used the KoggeStone parallel prefix adder (KSA) developed by Peter Kogge and Harold Stone in the year 1973. KSA is a parallel prefix form carry look ahead adder. It generates carry in O (log n) time and is widely considered as the fastest adder. It is widely used in the industry for high performance arithmetic circuits. In KSA, carry is computed fast by computing it in parallel at the cost of increased area.

[1] PG Center, Visvesvaraya Technological University, Belgaum, Karnataka 590014, India.

KOGGE-STONE ADDER

The KoggeStone has low logic depth, high node count, and minimal fan out. While a high node count implies a larger area, the low logic depth and minimal fanout allow faster performance.

There are mainly three computational stages in KoggeStone Adder. They are:

1. Preprocessing
2. Carry generation network
3. Post processing

Preprocessing Stage

Preprocessing is the first stage where the generate and propagate signals of all the input pairs of signals A and B are generated separately for each bit. The logical equations of the propagate and generate signals are given by the following equations:

$$Pi = Ai \text{ x or } Bi \quad ...(1)$$

$$Gi = Ai \text{ and } Bi \quad ...(2)$$

Carry Generation Stage

Carry generation is the second stage of the KSA. At this stage the carries of all the bits are generated separately for each bit. They are divided into smaller pieces and this overall process is carried out in parallel for all the bits. Carry generate and Carry propagate bits are used as intermediate signals and their logical equations are given as follows:

$$CPi{:}j = Pi{:}k + 1 \text{ and } Pk{:}j \quad ...(3)$$

$$CGi{:}j = Gi{:}k + 1 \text{ or } (Pi{:}k + 1 \text{ and } Gk{:}j) \quad ...(4)$$

Post Processing Stage

This is the final step or stage of the KSA which is common for all types of adders, i.e., calculation of summation of the bits given by the logical Equations (5) and (6):

$$Ci{-}1 = (Pi \text{ and } Cin) \text{ or } Gi \quad ...(5)$$

$$Si = Pi \text{ x or } Ci{-}1 \quad ...(6)$$

The KoggeStone adder is the fastest possible layout, because it scales logarithmically. Every time we add a combining step, it doubles the number of bits that can be added. The only Problem in KSA is that the number of wires gets increased as the number of bits increases.

The schematic of the 8bit KSA is as shown in Figure 2. As shown in Figures 1 and 2 the wiring increases with respect to the number of bits thus increasing on chip area. A large part of delay is because of the route delay as the wirings are more in KSA. Thus in KSA design the area parameter is to be compromised to get the high speed.

Figure 1: Schematic of the 4 Bit Kogge-Stone Adder

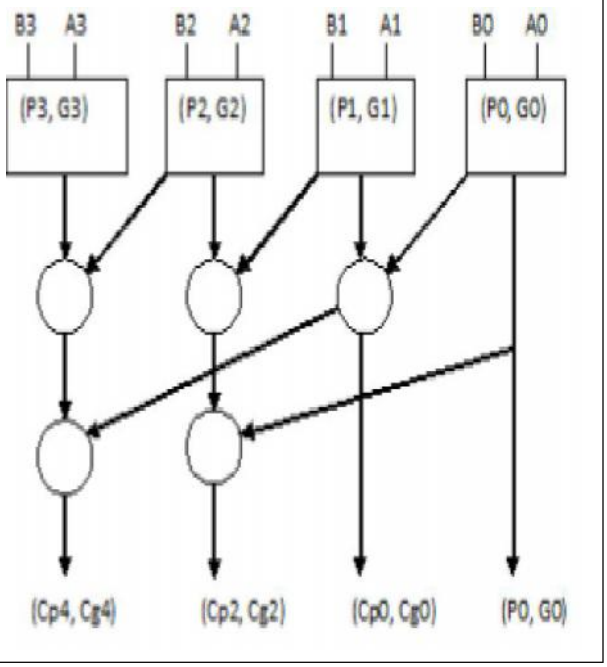
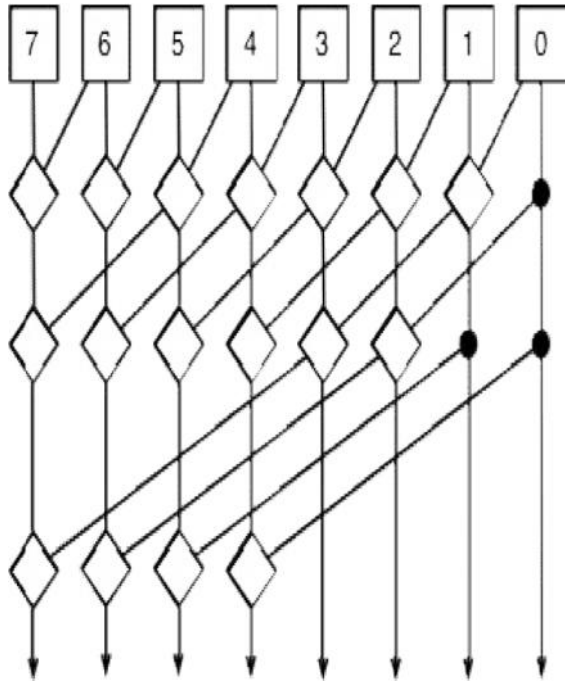


Figure 2: Schematic of the 8 Bit Kogge-Stone Adder



METHODOLOGY

To Design the adders based on FPGA, procedure is:

- To study the adder with their benefits and limitations.
- Making comparison write verilog code. Implementation and Synthesis on the Xilinx ISE Design Suit 14.7, ISim simulator.
- After synthesisation observe the simulation results.

Software used:

1. Xilinx ISE Design Suit 14.7, ISim simulator
2. FPGA Development board: Xilinx Artix-7_{KIT}
3. Altera Quartus II ModelSim 10.1d

IMPLEMENTATION

KSA is implemented for 4, 8, 16, 32 and 64 bits using verilog HDL for Xilinx ARTIX-7 Device using Xilinx 14.7 and the designs are simulated using ISim Simulator and Altera Modelsim 10.1d. A comparison is made about the efficiency of KSA with Carry Look Ahead adder (CLA) and Carry Skip Adder (CSA) with respect to speed and number of slice LUT's used. The simulation results are shown in the below figures.

Figure 3 shows the RTL schemat of 64 bit KSA, Figure 4 shows the magnified view of RTL schematic 64 bit KSA, Figure 5 shows the Technology schematic of 64 bit KSA, Figure 6 shows the magnified view of Technology schematic of 64 bit KSA.

Figure 3: RTL Schematic of 64 Bit Kogge-Stone Adder

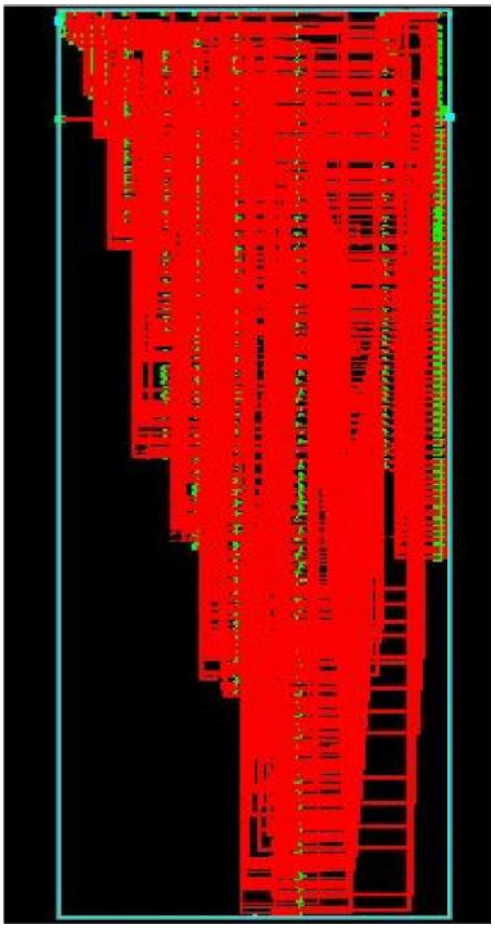


Figure 4: Magnified RTL Schematic of 64 Bit Kogge-Stone Adder

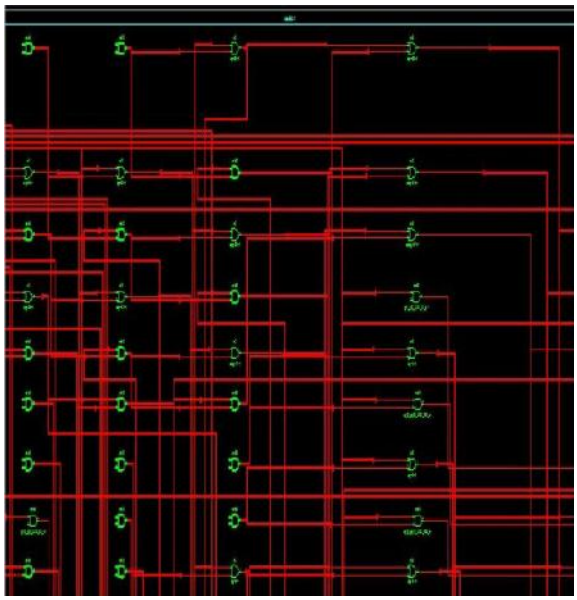


Figure 5: Technology Schematic of 64 Bit Kogge-Stone Adder

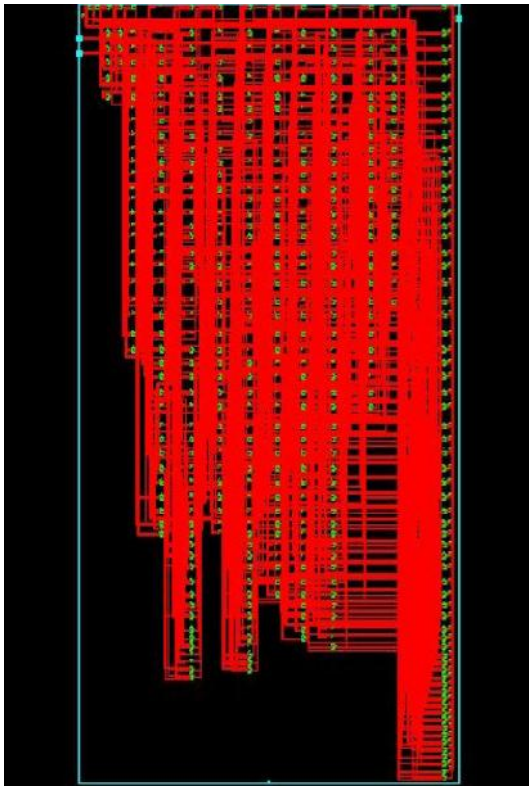
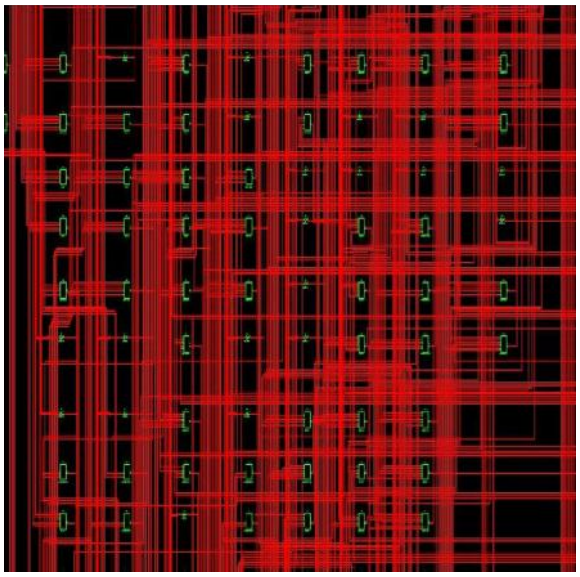


Figure 6: Magnified Technology Schematic of 64 Bit Kogge-Stone Adder



DEVICE UTILIZATION AND DELAY SUMMARY

Device used to implement the KSA adder for different bit sizes is ARTIX7-xc7a100t and the utilization summary is as given in the Table 1. The delay details are classified into two categories and they are logical delay and route delay. The delay values of KSA are compared with the CLA and CSA delay values. Table 2 shows the Delay summary of KSA, Table 3 shows the comparison of delays of KSA, CSA and CLA.

As shown in Table 3 the delay of the KSA compared with other adders is very less as

Table 1: Slice LUT's

Number of Bits	Used	Available	Utilization-on
8	11	63400	1%
16	48	63400	1%
32	119	63400	1%
64	332	63400	1%

Table 2: Delay Summary of KSA

Number of Bits	Logic Delay in ns	Route Delay in ns	Total Delay in ns
8	0.497	2.813	3.328
16	0.745	4.717	5.462
32	1.365	7.230	8.959
64	1.681	8.640	10.3

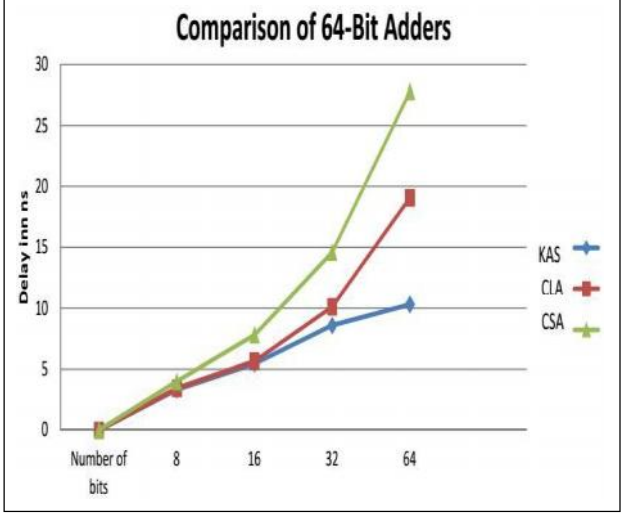
Table 3: Delay Summary of CSA

Number of Bits	Total Delay of KSA in ns	Total Delay of CSA in ns	Total Delay of CLA in ns
8	3.328	3.997	3.441
16	5.462	7.811	5.655
32	8.595	14.611	10.123
64	10.32	27.797	19.085

the overall operation of generating carry proceeds in parallel form.

According to the results the delay of the KSA for lower order bits is nearer to CLA, this is because of the wiring or route delays which are more in KSA, but when higher order KSA are implemented the delays are very less compared with others, because just by increasing single stage of computation we can design the next higher order adder. The plot shown in Figure 7 shows the characteristic of

Figure 7: Plot of Delay vs Bit Numbers

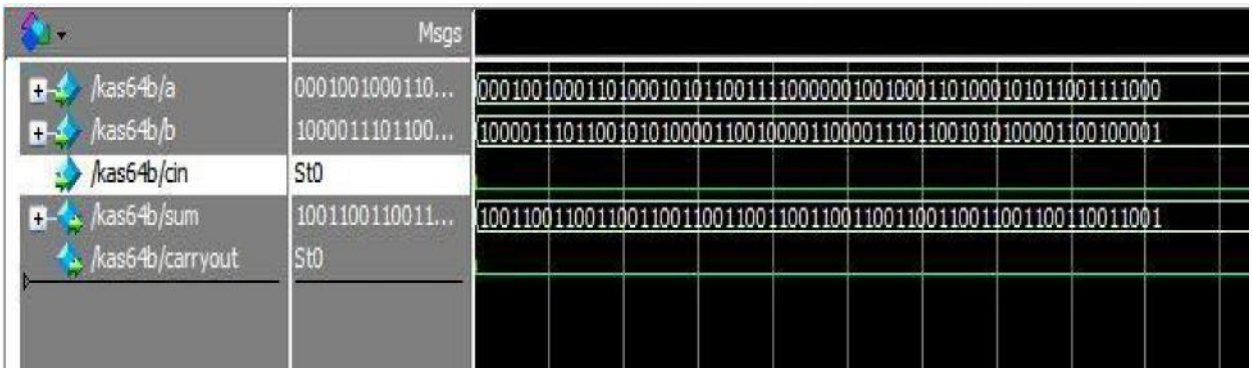


the KSA. Simulated outputs are shown in the Figure 8.

CONCLUSION

In this paper the Kogge-Stone adder for 8, 16, 32 and 64 bit are implemented and the results are compared to justify that the Kogge-Stone method is good for higher order Adders. Also the speed of Kogge-Stone adder is compared with other methods of addition and proved that the Kogge-Stone adder is best among all other adders with respect to speed. Further if the chip area consumption by the KSA is reduced this would create a revolution in the electronics

Figure 8: Simulated Output of 64 Bit KSA



providing high speed and compact electronic devices.

REFERENCES

1. Asadi P and Navi K (2007), "A Novel High-Speed 54-54-Bit Multiplier", *Am. J. Applied Sci.*, Vol. 4, No. 9, pp. 666-672.
2. Bedrij O J (1962), "Carry-Select Adder", *IRE Transactions on Electronics Computers*, Vol. EC11, June, pp. 340-346.
3. Chang T Y and Hsiao M J (1998), "Carry-Select Adder Using Single Ripple-Carry Adder", *Electronics Letters*, Vol. 34, October, pp. 2101-2103.
4. Choi Y (2005), "Parallel Prefix Adder Design", Proc. 17th IEEE Symposium on Computer Arithmetic, June 27, pp. 90-98.
5. Hasan Krad and AwsYousif Al-Taie (2008), "Performance Analysis of a 32-Bit Multiplier with a Carry-Look-Ahead Adder and a 32-Bit Multiplier with a Ripple Adder Using VHDL", *Journal of Computer Science*, Vol. 4, No. 4, pp. 305-308.
6. Kuldeep Rawat, Tarek Darwish and Magdy Bayoumi (2002), "A Low Power and Reduced Area Carry Select Adder", 45th Midwest Symposium on Circuits and Systems", Vol. 1, March, pp. 467-470.
7. Lai H C (1976), "A Study of Current Logic Design Problems", January, Ph.D. Dissertation, Dep. Computer Sci., Univ. of Illinois.
8. Ling H (1981), "High Speed Binary Adder", *IBMJ. Res. Develop.*, Vol. 25, No. 3, pp. 156-166.
9. Rabaey J M (2001), "Digital Integrated Circuits—A Design Perspective", Prentice-Hall, New Jersey.
10. Youngjoon Kim and Lee-Sup Kim (2001), "64-Bit Carry-Select Adder with Reduced Area", *Electronics Letters*, Vol. 37, No. 10, pp. 614-615.