Design of Low Power-delay Consumption Kogge-Stone Parallel Prefix Adder for High Speed Computing

Mohanraj.M
Department of ECE,
SNS college of technology, Coimbatore,
Tamilnadu, India.

Nethaji.B
Department of ECE,
SNS college of technology, Coimbatore,
Tamilnadu, India.

Nithya.S
Department of ECE,
SNS college of technology, Coimbatore,
Tamilnadu, India.

Nivetha.N
Department of ECE,
SNS college of technology, Coimbatore,
Tamilnadu, India.

***Abstract*-Parallel Prefix Adder is one of the most fastest type of adder that had been created and developed. There are two types of parallel prefix adders are present. They Brent Kung and Kogge-Stone adders. This research involve an investigation of the performance of the Kogge-Stone adder in terms of computational power and delay.The investigation and comparison for Kogge-Stone adder was conducted for 8, 16, 32, 64, and 128 bits size. By using the Xilinx 10.1 software, the design for the Kogge-Stone adder was developed. The simulation result produced the vector waveform which then shows the computational power delay for the adder. However, this project the proposed design 128-bit Kogge-Stone adder has reduced more delay and power as compared with the regular other parallel prefix adder.**

***Index terms* —kogge stone, parallel prefix adder, carry generation, power delay product, xor gate.**

I. INTRODUCTION

Design of delay and power efficient high speed data path logic systems are one of the most substantial areas of research in VLSI system design. The power advantage is important with the growing popularity of mobile and portable electronics, which make extensive use of VLSI design.The structure of the prefix network, described by Haiku Zhu, Chung-Kuan Cheng and Ronald Graham , has the minimal depth for a given 'n' bit adder. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder. The sum for each bit position in a parallel adder is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position. The parallel prefix adder (PPA) is used in many computational systems to all deviate the problem of carry propagation delay by in dependently generating multiple carries and then select a carry to generate the sum. However, the basic idea of this work is to use the Kogge–Stone adder (KSA) to achieve lower carry propagation delay and power consumption. The main advantage of this structure low fan-out in each stage. However, because of the structure of the configurable logic and resources, that the parallel-prefix adder will have a different performance in VLSI implementations. In particular, PPA is a fast-carry chain which optimizes the carry path for the simple Kogge-Stone Adder.In general, addition is a process which involves two numbers which are added and carry and sum will be generated. In this paper, an attempt has been made to design and simulate the different types of adders using Verilog language and simulate in Xilinx ISE 10.1. Then the performance parameters of the various adders are calculated and compared. The remaining paper is organized as follows: Section I deals with the introduction about PPA. The details of Kogge Stone Adder are discussed in section II, Section III deals with methodology of the basic adder blocks.The methodology of 128 bit proposed system is described in section IV. And section V deals with the simulation and synthesis results of KSA.The ASIC implementation details and results are analysed in Section VI. Finally, the research paper is concluded in Section VII.

II.PARALLEL PREFIX ADDERS

The parallel prefix adders are more flexible and are used to speed up the binary additions.The Kogge-Stone adder is an example of a parallel prefix adder. The following section is discussed about KSA.

A.KOGGE–STONE ADDER

The Kogge–Stone Adder (KSA) is the common design for high-performance adders. It is a favored adder is because of its minimum logic depth. As a result of that, the KSA becomes a fast adder. The delay of KSA is equal to log2n which is the number of stages for the "*o*" operator.It generates the carry signals in log n time, and is widely considered the fastest adder design possible. It takes more area to implement than the Brent–Kung adder, but has a lower fan-out at every stage, which increases performance. The Kogge-Stone adder generates and the propagate signal in the pre-computed process. In the tree-based adder, carries are generated in tree and fast computation is obtained at the expense of increase 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 PPA are more favourable in terms of speed due to the O (log2*n*) delay through the carry path compared to O(*n*) for the Ripple carry adder. The Kogge-Stone adder is widely used in high performance 32-bit, 64-bit, and 128-bit adders as it reduces the critical path to a great extent compared to the ripple carry adder.The algorithm was invented by Peter M. Kogge and Harold S. Stone has both optimal depth and low fan-out but produces massively complex circuit realizations. In Kogge-Stone adder, carries are generated fast by computing them in parallel.

III.METHODOLOGY

The adders to be studied were designed with varied bit widths up to 128 bits and coded in VERILOG language. The function of the adder was verified by using Model-sim Simulator. The Xilinx ISE Design Suite 10.1 software was used to synthesize the designs onto the Spartan3 XC3S4000L device. Parallel prefix adders are fastest adders and used for high performance arithmetic circuits in electronics. Considering the structure of the Generate-Propagate blocks, we were able to considering the outcome of the operation depends on the initial inputs. It involves the execution of an operation in parallel. Any arbitrary primitive operator "o"that is associative is parallelized. The construction of parallel prefix adder involves three stages:

1. Pre- processing
2. Carry generation
3. Post processing

The total function of KSA can be easily analyzing in the following of three different processes:

1. Pre-processing:

This first step involves computation of generate and propagate signals. These signals are given by the following equations:

$$pi = Ai \; XOR \; Bi \quad (1)$$

$$gi = Ai \; AND \; Bi \quad (2)$$

2. Carry look ahead:

This second step involves computation of carries. It uses group propagate and generate as intermediate signals which are given by the logic equations below:

$$Pi{:}j = Pi{:}k+1 \; AND \; Pk{:}j \quad (3)$$

$$Gi{:}j = Gi{:}k+1 \; OR \; (Pi{:}k+1 \; AND \; Gk{:}j) \quad (4)$$

3. Post processing:

This is the final step and is common to all adders of this PPA family. It involves computing the sum bits. Sum bits are computed by the following logic equation:

$$Si = pi \; XOR \; Ci\text{-}1 \quad (5)$$

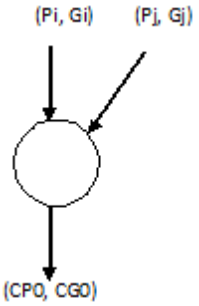

Figure 1: Carry operator

An example of a 4-bit Kogge–Stone adder is shown to the fig.2. Each vertical stage produces a "propagate" and a "generate" bit, as shown. The culminating generates bits (i.e.) the carries are produced in the last stage, and these bits are XOR with the initial propagate after the input in the square boxes to produce the sum bits. E.g., the propagate is calculated by XORing the A value as 1 and B values as 0. It producing the propagate output as 1. The generate value is calculated by AND the A value as 1, B value as 0. It produced the generate output value as 0. The first sum bit is calculated by XORing the propagate in the circle box, the value as "1" with the carry-in as "0", it producing a "1". Likewise all the sum bits are calculated. Inputs: A = 1011 and B = 1100 Outputs: sum =0111, carry Cout = 1.

The working of KSA can be understood by the following fig.2 which corresponds to 4-bit KSA

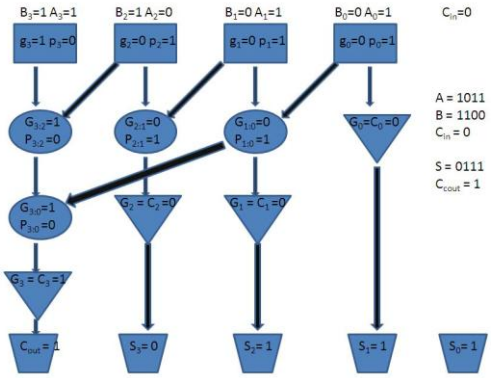

Fig 2: Illustration of 4 bit Kogge-Stone adder

In a 4 bit adder proceed the 5 outputs in the following expansion:

S0 = (A0 ^ B0) ^ CIN

S1 = (A1 ^ B1) ^ (A0 & B0)

S2 = (A2 ^B2) ^ (((A1 ^ B1) & (A0 & B0)) | (A1 & B1))

S3 = (A3 ^ B3) ^ ((((A2 ^ B2) & (A1 ^ B1)) & (A0 & B0)) | (((A2 ^ B2) & (A1 & B1)) | (A2 & B2)))

S4 = (A4 ^ B4) ^ ((((A3 ^ B3) & (A2 ^ B2)) & (A1 & B1)) | (((A3 ^ B3) & (A2 & B2)) | (A3 & B3)))

III. PROPOSED 128 BIT KOGGE-STONE ADDER

In this architecture a two 128 bit inputs are added using a Kogge-Stone adder. The total number of stages present in the 128 bit Kogge-Stone adder is 8 stages (log n).

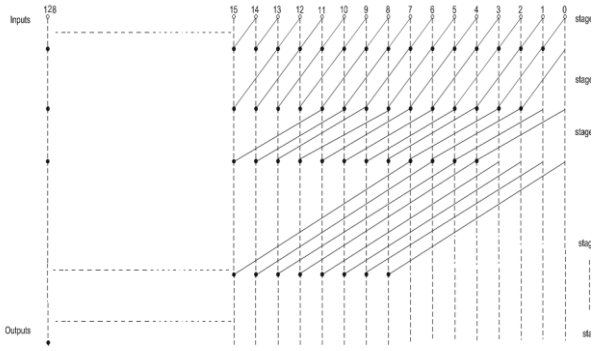

Fig 3: The Kogge-Stone parallel prefix graph

where n= 128 the total number of input bits.

In the figure 3, the Kogge-Stone adder operation is shown for 128 bit.From LSB to MSB we can observe that in each stage the black nodes are reducing or shifted to $2^{l}-1$, (where l = stage number) horizontally and the reduced black nodes are inserted with white nodes. The black nodes are reduced finally to $n2$ (i.e., in the final stage).

IV. SIMULATION RESULTS

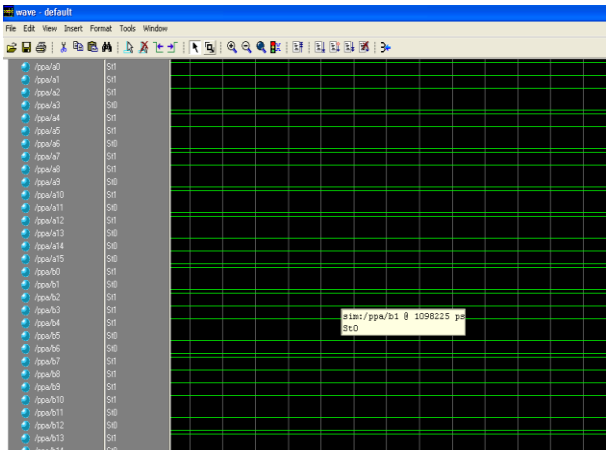

Fig 4: proposed 128 bit kogge-stone adder

V. IMPLEMENTATION

Table (1) shows the comparison of regular KSA and modified KSA in terms of power and delay.

Table (1): Comparison of power and delay:

Adder	Delay (ns)	Power (mW)	Power-delay product
Regular KSA	10.584	3062.17	32.410
Modified KSA	82.706	264	21.834

From above table (1) represents the evident that there is reduction in overall delay in the proposed 128 bit Kogge-Stone adder.

Name	Power (W)	Used	Total Available
Logic	0.000	414	55296
Signals	0.000	609	---
IOs	0.000	385	633
Total Quiescent Power	**0.264**		
Total Dynamic Power	**0.000**		
Total Power	**0.264**		

Fig 5: power analyser

The figure (5) represents the power analyser report. The logic delay and power of modified Kogge-Stone adder is 82.706 nanosecond and 264 mill watt.When compared to normal Kogge-Stone adder architecture there is an improvement in the overall delay. The proposed structures have the least power delay product amongst all other adders.Results obtained from modified KSA is better in delay and power consumption.

VI. CONCLUSION

The above experimental results are proved that Kogge-Stone adder is very high speed than normal other adders when it will increase the width of the adders. All adders will successfully synthesize using Verilog HDL in Xilinx 10.1 for Spartan3 XC3S4000L with speed grade-4.Synthesis tool and simulation will done using ISE simulator. The replacement of prefix adders in place of Kogge-Stone adder offers great advantage in the reduction of delay.It is concluded that, carry increment adder achieves better performance in terms of power and delay compared to that all other adder topologies. The proposed KSA using common Boolean logic has low power, and less delay than all other adder structures. It is also little bit faster than all the other adders which makes it simple and efficient for VLSI hardware implementations.

REFERENCES

[1] David Harris, "A Taxonomy of parallel prefix networks," Proceedings of the 37th Asilomar Conference on Signals, Systems and Computers Pacific Grove, California, pp.2213-2217, November 2003.

[2] David Jeff Jackson and Sidney Joel Hannah, "Modelling and Comparison of Adder Designs with Verilog HDL", 25th South-eastern Symposium on System Theory, pp.406-410, March 1993.

[3] Dimitrakopoulos, Nikolos, "High-Speed Parallel-Prefix VLSI Ling Adders", IEEE 2005

[4] Haikun Zhu, Chung-Kuan Cheng and Ronald Graham, "ConstructingZero Deficiency Parallel Prefix adder of Minimum Depth," Proceedingsof 2005 Asia South Pacific Design Automation Conference, pp.883–888, 2005.

[5] Madhu Thakur and Javed Ashraf (2012), "Design of Braun Multiplier with Kogge-Stone Adder & It's Implementation on FPGA", *International Journal of Scientific & Engineering Research*, Vol. 3, No. 10,pp. 03-06, ISSN 2229-5518.

[6] M. Snir, "Depth-size trade-offs for parallel prefix computation," in Journal of Algorithms 7, pp.185–201, 1986.

[7] P. Kogge and H. Stone, "A parallel algorithem for the efficient solution of a general class of recurrence relation," IEEE transactions on computers, vol. C-22, no.8, pp.786-793, Aug 1973.

[8] P.Ramanathan, P.T.Vanathi, "Novel Power Delay Optimized 32-bitParallel Prefix Adder for High Speed Computing", International Journal of Recent Trends in Engineering, Vol 2, No. 6, November 2009.

[9] Richard P. Brent and H. T. Kung, "A Regular Layout for Parallel Adders", IEEE transactions on Computers, vol.c-31, pp.260-264, March 1982.

[10] Y. Choi, "Parallel Prefix Adder Design," *Proc. 17th IEEE Symposium on Computer Arithmetic*, pp 90-98, 27th June 2005.

Authors Profile

M.Mohanraj pursuing the final year **B.E.** in electronics and communication engineering at SNS college of technology, Coimbatore, Tamilnadu, India, in 2011-15.His research interest includes digital signal processing, digital electronics.

B.Nethaji pursuing the final year **B.E.** in electronics and communication engineering at SNS college of technology, Coimbatore, Tamilnadu, India, in 2011-15.His research interest includes control system, imaging processing.

S.Nithya pursuing the final year **B.E.** in electronics and communication engineering at SNS college of technology, Coimbatore, Tamilnadu, India, in 2011-15.Her research interest includes devices and circuit, digital electronics.

N.Nivetha pursuing the final year **B.E.** in electronics and communication engineering at SNS college of technology, Coimbatore, Tamilnadu, India, in 2011-15.Her research interest includes designing of control system, digital signal processing.