IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735. Volume 6, Issue 1 (May. - Jun. 2013), PP 01-06
www.iosrjournals.org

Design of 32 bit Parallel Prefix Adders

P.Chaitanya kumari[1,] R.Nagendra[2]
PG Student, VLSI, Assistant Professor, ECE Dept., SVEC, Tirupati, Chittoor, A.P, India ,

Abstract: *In this paper, we propose 32 bit Kogge-Stone, Brent-Kung, Ladner-Fischer parallel prefix adders. In general N-bit adders like Ripple Carry Adders (slow adders compare to other adders), and Carry Look Ahead adders (area consuming adders) are used in earlier days. But now the most Industries are using parallel prefix adders because of their advantages compare to other adders. Parallel prefix adders are faster and area efficient. Parallel prefix adder is a technique for increasing the speed in DSP processor while performing addition. We simulate and synthesis different types of 32-bit prefix adders using Xilinx ISE 10.1i tool. By using these synthesis results, we noted the performance parameters like number of LUTs and delay. We compare these three adders in terms of LUTs (represents area) and delay values.*
Keywords− *prefix adder, carry operator, Kogge-Stone, Brent-Kung, Ladner-Fischer.*

I. Introduction

Arithmetic circuits are the ones which perform arithmetic operations like addition, subtraction, multiplication, division, parity calculation. Most of the time, designing these circuits is the same as designing muxers, encoders and decoders. In electronics, an adder or summer is a digital circuits[7] that performs addition of numbers. In many computers and other kind of processors, adders are other parts of the processor, many computers and other kinds of processors, where they are used to calculate addresses, table and similar. The binary adder[7,10] is the one type of element in most digital circuit designs including digital signal processors(DSP) and microprocessor data path units. Therefore fast and accurate operation of digital system depends on the performance of adders [6]. Hence improving the performance of adder is the main area of research in VLSI[10] system design. The Conventional adders discussed in section II. The details of R Kogge-Stone adder, Brent-Kung adder and Ladner- Fischer adders are discussed, and the implementation of proposed system is described in section III. The performance and simulation results were presented and discussed in section IV.

II. Conventional Adders

Ripple Carry Adder

Ripple Carry Adder is constructed by cascading full adder blocks in series. A RCA is a logic circuit in which the carry out of one stage fed directly to the carry in of the next stage. It is called RCA because each carry bit gets rippled into the next stage. The main drawbacks of the ripple adder is every bit being added has to propagate through each digital logic gate in the circuit before an answer can be generated. This is known as a gate delay. The 4 bit RCA figure shown below.

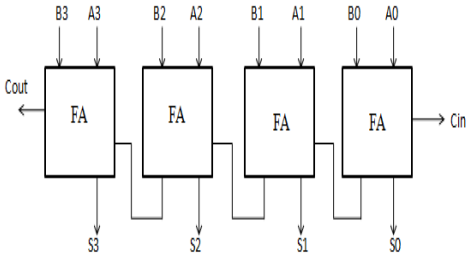

Figure1: 4-bit Ripple Carry Adder

Carry Look Ahead Adder

A Carry Look Ahead adder(CLA) is a type of adder used in digital circuits. 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[16] 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. The Kogge-Stone adder and Brent-Kung adder and Ladner-

Fischer[14] are examples of this type of adder. To reduce the computation time, engineers devised faster ways to add two binary numbers by using carry-look ahead adders. They work by creating two signals (*P* and *G*) for each bit position, based on if a carry is propagated through from a less significant bit position (at least one input is a '1'), a carry is generated in that bit position (both inputs are '1'), or if a carry is killed in that bit position (both inputs are '0'). In most cases, *P* is simply the sum output of a half-adder and *G* is the carry output of the same adder. After *P* and *G* are generated the carries for every bit position are created. Some advanced carry-look ahead architectures are the Brent-Kung adder, and the Kogge-Stone adder and Ladner-Fischer adder[5].The 4 bit CLA figure shown below.

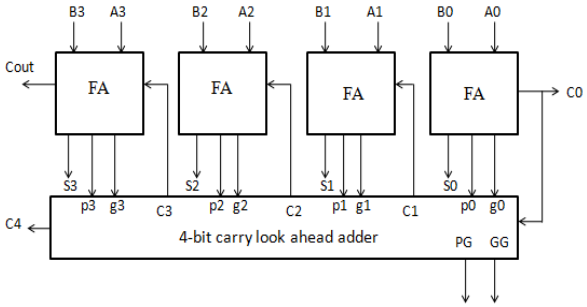

Figure2: 4 bit Carry Look Ahead Adder

III. Proposed Adders

Parallel prefix adders

The PPA is like a Carry Look Ahead Adder. The production of the carriers the prefix adders [1] can be designed in many different ways based on the different requirements. We use tree structure form to increase the speed [13] of arithmetic operation. Parallel prefix adders are faster adders [1] and these are faster adders [4] and used for high performance arithmetic structures in industries. The parallel prefix addition is done in 3 steps.

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

Pre-processing stage

In this stage we compute, the generate and propagate signals are used to generate carry input of each adder. A and B are inputs. These signals are given by the equation 1&2.

$$P_i = A_i \oplus B_i \quad \text{.....(1)}$$
$$G_I = A_i \cdot B_i \quad \text{.....(2)}$$

Carry generation network]

In this stage we compute carries corresponding to each bit. Execution is done in parallel form [4].After the computation of carries in parallel they are divided into smaller pieces. carry operator contain two AND gates , one OR gate. It uses propagate and generate as intermediate signals which are given by the equations 3&4.

$$P_{(i:k)} = P_{(i:j)} \cdot P_{(j-1:k)} \quad \text{.....(3)}$$
$$G_{(i:k)} = G_{(i:j)} + (G_{(j-1:k)} \cdot P_{(i:j)}) \quad \text{.....(4)}$$

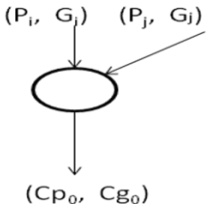

Figure3: Carry operator.

The operations involved in this figure are given as.

Post processing stage

This is the final stage to compute the summation of input bits. it is same for all adders and sum bit equation given

$$S_i = P_i \oplus C_i \quad \text{.....(5)}$$

$$C_{i+1} = (P_i \bullet C0) + G_i \quad \text{.....(6)}$$

Parallel Prefix Adders are classified into

1. Kogge- Stone Adder
2. Brent-Kung Adder
3. Ladner-Fischer Adder

Kogge - Stone Adder

Kogge-Stone adder is a parallel prefix form carry look ahead adder. The Kogge-Stone adder [3] was developed by peter M. Kogge and Harold S. Stone which they published in 1973. Kogge-Stone prefix adder is a fast adder design. KS adder has best performance in VLSI implementations. Kogge-Stone adder has large area with minimum fan-out. The Kogge- Stone adder is widely known as a parallel prefix adder that performs fast logical addition. Kogge-Stone adder[9] is used for wide adders because of it shows the less delay among other architectures. In fig2 each vertical stage produce Propagate and Generate bits. Generate bits are produced in the last stage and these bits are XORed with the initial propagate after the input to produce the sum bits. The 2-bit and 32- bit Kogge- Stone adder figures shown below.

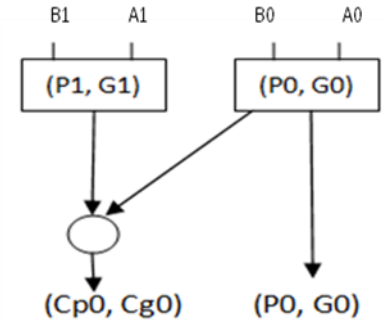

Figure4: 2-bit KS Adder

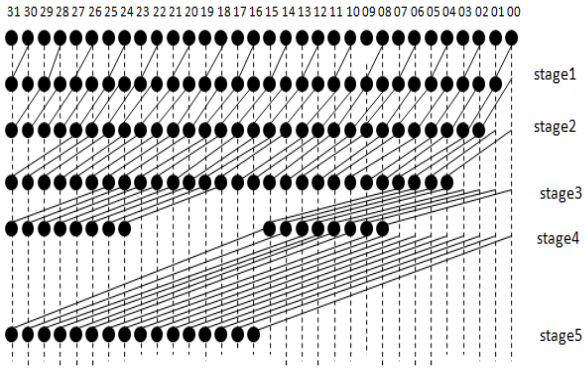

Figure 5: 32-bit Kogge-Stone Adder

Brent-Kung Adder

The Brent-Kung adder[3] is a parallel prefix adder. The Brent-Kung adder was developed by Brent and Kung which they published in 1982. Brent-Kung has maximum logic depth and minimum area. The number of cells are calculated by using $2(n-1) - Log2^n$.The 4-bit and 32 bit Brent- Kung adder figures shown below.

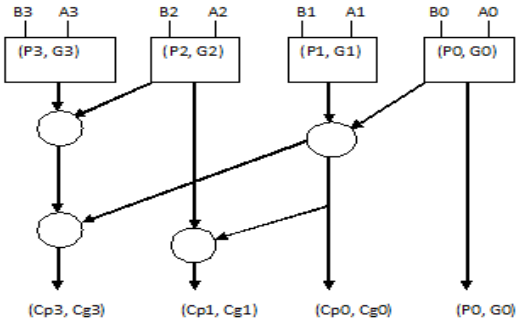

Figure 6: 4-bit BK Adder

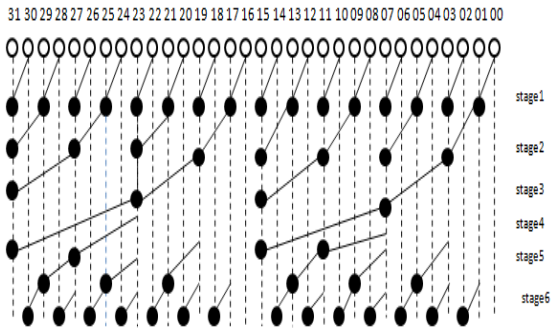

Figure7: 32-bit Brent-kung adder

Ladner-Fischer Adder

Ladner- Fischer adder is a parallel prefix adder. This was developed by R. Ladner and M. Fischer in 1980.Ladner- Fischer adder[6] has minimum logic depth but it has large fan-out . Ladner- Fischer adder has carry operator nodes. The 3-bit and 32 bit Ladner- Fischer adder figures shown below.

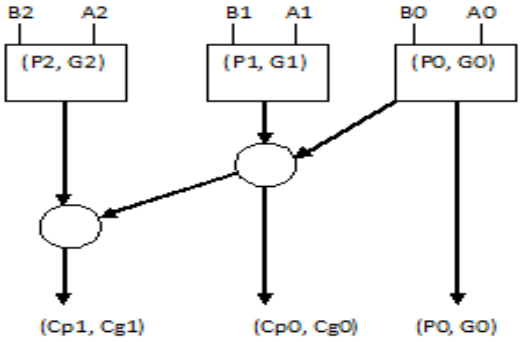

Figure8: 3-bit LF Adder

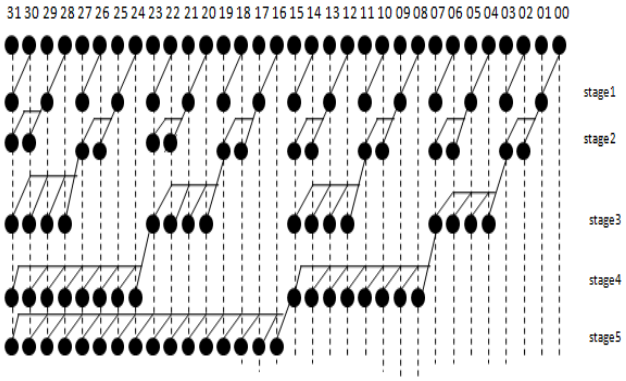

Figure9: 32- bit Ladner- Fischer adder

IV. Simulation Results And Comparisions

Various adders were designed using Verilog language in Xilinx ISE Navigator 10.1 and all the simulations are performed using Modelsim 6.5e simulator. The performance[12] of proposed adders are analyzed and compared. In this proposed architecture , the implementation code for 32-bit Kogge-Stone, Brent-Kung adder, Ladner-Fischer adders were developed and corresponding values of delay and area were observed. Table1 shows the trade-off between different topologies and table2 shows the comparision of adders. The simulated outputs of 32-bit proposed adders are shown in Figure8,9&10.

Table1: Trade-off between different topologies

Topology	Logic Level	Fan-out	Wiring Track
Kogge-Stone	LOW	LOW	HIGH
Ladner-Fischer	HIGH	LOW	LOW
Brent-Kung	LOW	HIGH	LOW

Table2: Comparision of adders

Topology	Delay	No of LUTs
Kogge-Stone	20.262ns	204
Ladner-Fischer	23.218ns	87
Brent-Kung	22.961ns	85

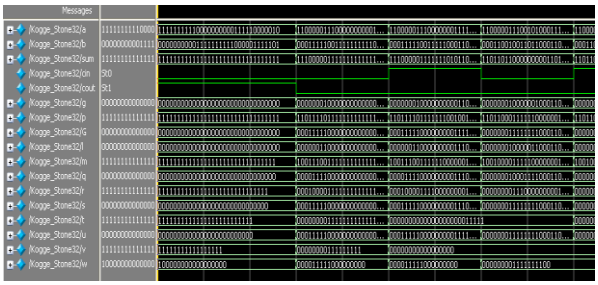

Figure8: Simulated Output of 32-bit KS Adder

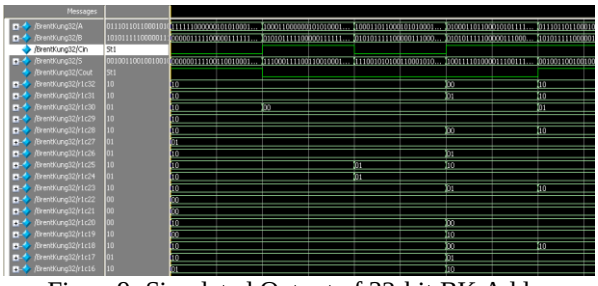

Figure9: Simulated Output of 32-bit BK Adder

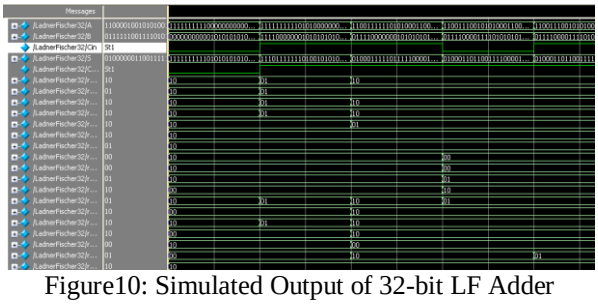

Figure10: Simulated Output of 32-bit LF Adder

V. Conclusion

The proposed adders are faster because of less delay and area efficient compared to other basic adders. Among these three prefix adders Ladner-Fischer adder has better performance compared to remaining adders. The performance comparisons between these adders are measured in terms of area and delay. It would be

interesting to investigate the design of the 128 and 256 bit adders. These adders are popularly used in VLSI implementations.

Acknowledgement

P. Chaitanya Kumari would like to thank Mr.R .Nagendra, Assistant Professor(senior grade) ECE Department who had been guiding throughout the project and supporting me in giving technical ideas about the paper and motivating me to complete the work efficiently and successfully.

References

[1] Y. Choi, "Parallel Prefix Adder Design", Proc. 17th IEEE Symposium on Computer Arithmetic, pp 90-98, 27th June 2005.
[2] R. P. Brent and H. T. Kung, "A regular layout for parallel adders", IEEE trans, computers, Vol.C-31,pp. 260-264,.March 1982.
[3] Kogge P, Stone H, "A parallel algorithm for the efficient solution of a general class Recurrence relations," IEEE Trans. Computers, Vol.C-22, pp 786-793,Aug. 1973.
[4] R. Zimmermann, "Non-heuristic operation and synthesis of parallel-prefix adders," in International workshop on logic and architecture synthesis, December 1996,pp. 123-132.
[5] C.Nagendra, M. J. Irwin, and R. M. Owens, "Area -Time-Power tradeoffs in parallel adders", Trans. Circuits Syst. II, vol.43, pp. 689– 702 Oct.1996.
[6] R. Ladner and M. Fischer, "Parallel prefix computation, Journal of ACM.La.Jolla CA,Vol.27,pp.831-838,October 1980.
[7] Reto Zimmermann. Binary Adder Architectures for Cell-Based VLSI an their Synthesis. Hartung-Gorre, 1998.
[8] Y. Choi, "Parallel Prefix Adder Design," Proc. 17th IEEE Symposium on Computer Arithmetic, pp 90-98, 27th June 2005.
[9] D. Harris, "A taxonomy of parallel prefix networks," in Signals, Systems and Computers,2003. Conference Record of Thirty Seventh Asilomar Conference on, vol. 2, the Nov. 2003,pp.2217.
[10] N. H. E. Weste and D. Harris, CMOS VLSI Design, 4th edition, Pearson Addison-Wesley, 2011.
[11] H. Ling, High-speed binary adder," IBM Journal of Research and Development, vol. 25,no. 3, pp. 156 March 1981.
[12] K.Vitoroulis and A. J. Al-Khalili "Performance of Parallel Prefix Adders Implemented with FPGA technology," IEEE Northeast Workshop on Circuits and Systems, pp. 498-501, Aug. 2007.
[13] D. H. K. Hoe, C. Martinez, and J. Vundavalli, "Design and Characterization of Parallel Prefix Adders using FPGAs, "IEEE 43rd Southeastern Symposium on System Theory, pp. 170-174, March 2011.
[14] T. Matsunaga, S. Kimura, and Y. Matsunaga."Power-conscious syntheses of parallel prefix adders under bitwise timing constraints," Proc. the Workshop on Synthesis And System Integration of Mixed Information technologies(SASIMI), Sapporo, Japan, October 2007,pp. 7–14.
[15] F. E. Fich, "New bounds for parallel prefix circuits," in Proc. of the 15thAnnu. ACM Sympos. Theory of Comput., 1983, pp.100–109.
[16] D. Gizopoulos, M. Psarakis, A. Paschalis, and Y.Zorian, "Easily Testable Cellular Carry Look ahead Adders," Journal of Electronic Testing: Theory and Applications 19, 285-298, 2003.