ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

Performance of Delay, Power and Area for Parallel Prefix Adders with Xilinx

K. Mounika, R.Jawaharlal

PG Student, JITS, Karimnagar, India

Associate Professor, JITS, Karimnagar, India

ABSTRACT: In Very Large Scale Integration (VLSI) designs, Parallel prefix adders (PPA) have the better delay performance. A parallel prefix adder involves the execution of the operation in parallel which can be obtained by segmentation into smaller pieces. The binary addition is the basic arithmetic operation in digital circuits and it became essential in most of the digital systems including Arithmetic and Logic Unit (ALU), microprocessors and Digital Signal Processing (DSP). At present, the research continues on increasing the adder's delay performance.In this paper the investigation of four types of PPA's (Kogge Stone Adder (KSA), Spanning Tree Adder (STA), Brent Kung Adder (BKA) and Sparse Kogge Stone Adder (SKA)) is done. Additionally Ripple Carry Adder (RCA), Carry Lookahead Adder (CLA) and Carry Skip Adder (CSA) are also investigated. These adders are implemented in verilog Hardware Description Language (HDL) using Xilinx Integrated Software Environment (ISE) 13.4 Design Suite. The area, delay and power consumed by all types of PPA's are analyzed. The area of the adder design are given in terms of Look Up Tables (LUT's) and Input Output bounds (IOB's). The adder designs are implemented and delay, power and area of all the adders are investigated.

KEYWORDS: PPA, RCA, CLA, SKA, KSA, BKA, STAFPGA, Delay, Power, Area

I. INTRODUCTION

Binary adders are one of the most basic and widely used arithmetic operations in modern integrated circuits. They tend to play a critical role in determining the performance of the design. Arithmetic operations are the regular common operations in digital integrated circuits. The simplest circuit adds, subtracts, and multiplies or divides. The computation should be fast and the area consumed by the arithmetic units should be small. These are the two basic requirements for any adder [2]. Parallel computing isform of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can often be divided into smaller ones, which are then solved concurrently in parallel. There are several different forms of parallel computing: bit-level, instruction level, data, and task parallelism. Parallelism has been employed for many years, mainly in high-performance computing, but interest in it has grown lately due to the physical constraints preventing frequency scaling.

As power consumption and consequently heat generation by computers has become a major concern in recent years,parallel computing has become the dominant paradigm in computer architecture, mainly in the form of multi-core processors. Parallel computers can be roughly classified according to the level at which the hardware supports parallelism, with core and multi-processor computers having multiple processing elements within a single machine. Area and Time consumed by the circuit are the basic and important requirements. Numbers can be represented in digital circuits in various ways. Hence, developing efficient adder architecture is crucial to improving the efficiency of the design. Generally ripple carry adder is used for binary addition. After the design of ripple carry adder several techniques are used for the computation of parallel adders.

Carry look ahead adders are based on parallel prefix computation which gives better performance than ripple carry adder. After many years of research, focus is on improving the delay performance of the adder. As such,extensive research continues to be focused on improving the delay performance of the adder. Next, Brent and Kung (BK)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

designed parallel prefix-computation graph in anarea-optimal way and the kogge-stone (KS) architecture is optimized for timing.

The binary addition is the basic arithmetic operation in digital circuits and it became essential in most of the digital systems including Arithmetic and Logic Unit (ALU), microprocessors and Digital Signal Processing (DSP). At present, the research continues on increasing the adder's delay performance. In many practical applications like mobile and telecommunications, the Speed and power performance improvement in FPGAs is better than microprocessor and DSP's based solutions. Additionally, power is also an important aspect in growing trend of mobile electronics, which makes large-scale use of DSP functions. Because of the Programmability, structure of configurable logic blocks (CLB) and programming. interconnects in FPGAs, Parallel prefix adders have better performance [1].

II. LITERATURE REVIEW

Sudheer Kumar Yezerla et al.[1] did research on, "Design and Estimation of delay, power and area for Parallel Prefix Adders", where the design was implemented using Xilinx Virtex 5 FPGA and the adder delays were estimated using Agilent 1692A Logic Analyzer. The 16-bit SKA computes the carries with Black Cells (BC's) and Grey Cells (GC's) and terminates with a 4-bit RCA.

The 16-bit STA used also terminates with RCA and uses BC's and GC's and Full Adders but has difference in the interconnection between them. The 16-bit KSA designed used 16 BC's and 15 GC's with less delay compared to SKA and STA. The 16-bit BKA used 14 BC's and 11 GC's which is less compared to KSA and hence has less architecture and occupies less space than KSA. The delay measurements show that SKA and BKA have almost same delay where as STA has better delay results. The efficiency has increased for delay up to 5.77% for RCA and for KSA has improved by 19.28%.

CH. Chimpiraiah et al.[2] work on, "An efficient architecture for Parallel Adders" presents an efficient structure for parallel adders with fast performance which are particularly attractive for VLSI implementations. The simulation and synthesis was done in Xilinx ISE Foundation 9.2i software. The proposed design was the combination of KSA and BKA and the result is compared with KSA. The area given in terms of slices was reduced from 81 to 63 and the delay from 12.27ns to 11.04ns. The number of logic levels was reduced from 8 to 7 compared to KSA which gives better power performance. V. Krishna Kumari et al. [3] researched on, "Designing and Characterization of Kogge Stone, Sparse Kogge stone, Spanning tree and BrentkungAdders",and the results were obtained using modelsim 6.4b,Xilinx ISE 10.1i for simulation and synthesis. The practical issues while testing the adders were analyzed and a circuit to test the adders is designed. The objectiveof the testing adder circuit was to find the worst path and the input causing it was combination which is stored in ROM.

A.N. Jayanthi et al. [4] work on, "Comparison of performance of high speed VLSI adders" ,proposed 16-bit and 64-bit adder design for all the adders and the comparison was made in terms of delay. The 16-bit Ling adder design proposed has delay reduced to half compared to the 16-bit RCA, but the circuit has limitations on fan-in. The KSA adder scheme reduces logical fan-out in each node but increases the wiring within the circuit which in turn will have high power consumption. The delay of the 64-bit KSA is 48.064ns and of the Ling adder is 51.841ns. They concluded that several architectures are often close to the optimum performance for the given design environment. The result obtained shows that the given methodologies lead to the best design at all technological nodes. Neil Burgess [5]in the paper, "Fast Ripple-Carry Adders in Standard-Cell CMOS VLSI", presented new high-radix ripple carry adder based on Ling's addition technique. The critical path introduced inverting CMOS cell per stage along carry in and carry out. The fastest of them matches speed of 16-bit prefix adder for only 63% of the area.

Mangesh B Kondalkar et al. [6] work on, "Improved Fault Tolerant Sparse KOGGE Stone ADDER", proposed a method for error correction due to inherent redundancy in the carry tree and also error detection was possible. The prior work had only error correction which used Kogge Stone configuration. Several enhancements are introduced in the design; the error recovery time is reduced by using a 16-bit register, error correction due to fault in multiple ripple carry adders is included which improves the reliability of the circuit.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

Triple Mode redundancy Ripple carry (TMR-RC) is simple and effective method of fault detection and correction but it increases the area overhead and almost results in tripling of the associated power dissipation. A Sparse Kogge-Stone adder which is fully fault tolerant in its lower half (i.e., in the ripple carry adders) was proposed.
The addition of register reduces the error recovery time and also makes it possible to detect and correct the error in multiple RCA.

III. 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.

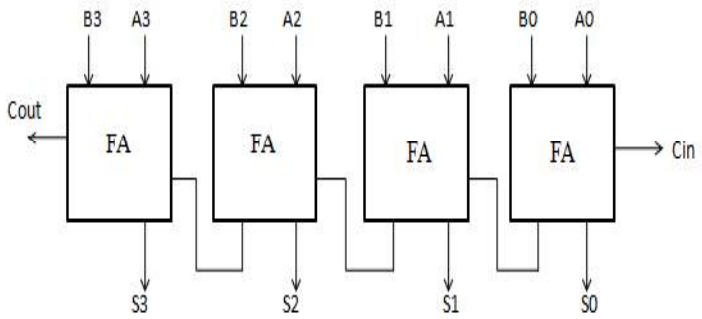

Figure 1: 4-bit ripple carry adder carry look ahead adder

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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

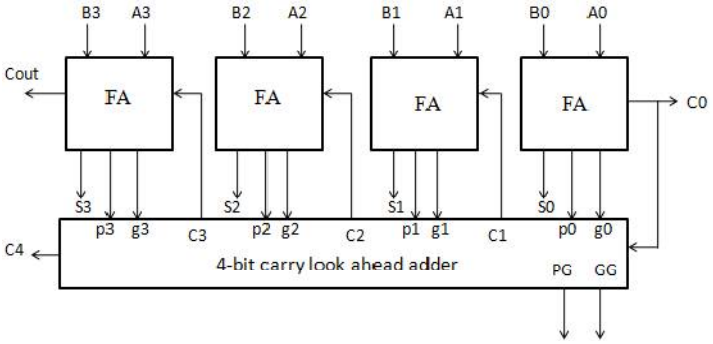

Figure 2: 4 bit Carry Look Ahead Adder

IV. 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

. Pi=Ai Bi.....(1)

GI=Ai .Bi.....(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.

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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

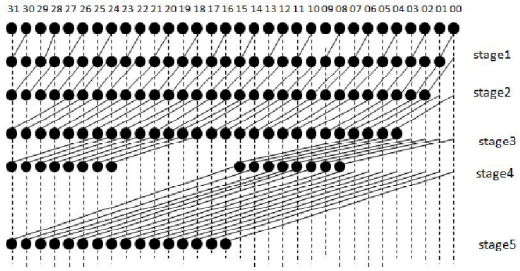

Figure 3: 32-bit Kogge-Stone Adder

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) -Log2n .The 4-bit and 32 bit Brent- Kung adder figures shown below

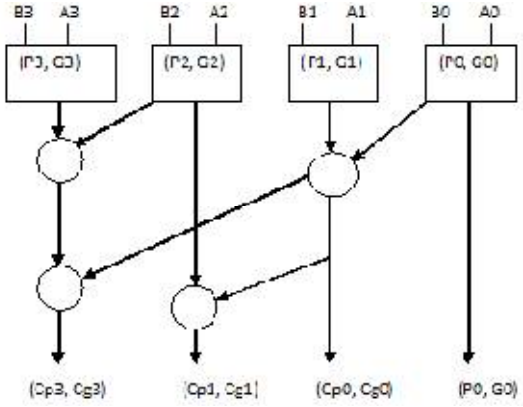

Figure 4: 4-bit BK Adder

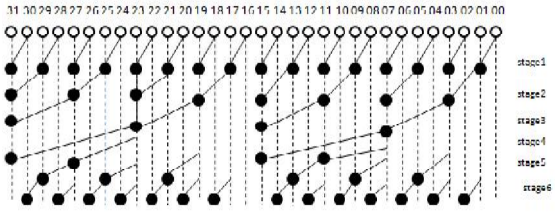

Figure 5: 32-bit Brent-kung adder Ladner-Fischer

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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

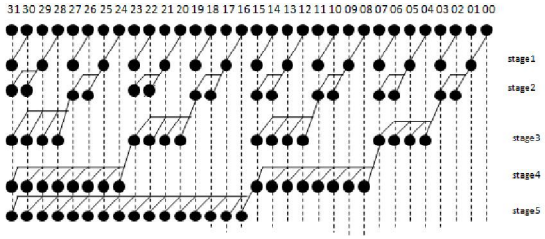

Figure 6 : 32- bit Ladner- Fischer adder

V. EXPERIMENTAL RESULTS

The Xilinx 14.5 version of VHDL is used with the targeted device of XC3S500E for the implementation of adders. Basic adders, ripple carry adder, carry look-ahead adder, carry bypass adder, carry select adder and hybrid parallel prefix adder is simulated and synthesize using Xilinx tool for 16-bit and 32-bit addition. The results of these adders for 16- bit addition and 32-bit addition are tabulated in table 1and table 2 respectively.

The comparison is done for three factors: speed, area and power consumption. The speed performance is evaluated with respect to and delay.The area requirement can be estimated from the utilization of number of slices and Look-up tables. The power consumption is analyzed by taking switching power (dynamic power) in account which mainly depends on the input test vectors that can be applied through the test bench. Static power is not considered because lack of ASIC tools available The design summary of synthesis and power analysis results of 32-bit hybrid parallel prefix adder are given in figures 12 and 13 respectively.

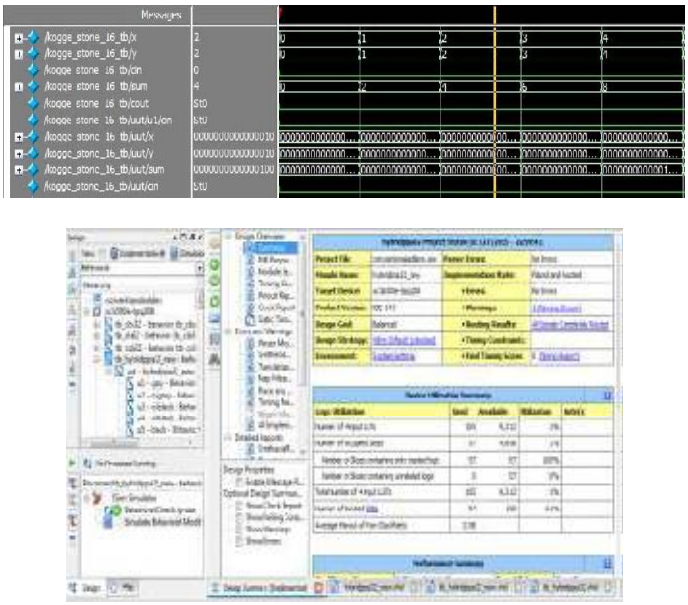

Figure 7 :Simulation result of adder

The device utilization summary report from Xilinx 13.2 gives the area occupied by each of the adders in terms of slices and LUT's when used in SPARTAN 3 Field Programmable Gate Array.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

VI.CONCLUSION AND FUTURE WORK

It is shown that the results obtained for Parallel Prefix Adders are better than the serial adders in terms of delay and at the same time there is a trade-off with the area occupied. The results obtained for carry chain adders at higher bit widths (128 to 256 bits) has higher performance when compared to serial adders. Because the adder is often the critical element which determines to a large part the cycle time and power dissipation for many digital signal processing and cryptographically implementations,it would be worthwhile for future FPGA designs to include an optimized carry path to enable tree based adder designs to be optimized for place and routing. Inclusion of a ROM is being done in few adder designs to find the worst case delay combination of the input. Architectures that include fast carry chains and the possible tradeoffs are investigated. In the future, designs for Spanning Tree Adder(STA) and Sparse Kogge Stone Adder (SKA) are done and delay and area occupied are calculated to check performance enhancement keeping serial adders as reference.

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.