Parallel Prefix Adders using FPGAs

Avneet Kaur[1], Chanpreet Kaur Toor[2]
[1]*Student-M.Tech VLSI (ECE), Chandigarh Engineering College, Landran*
[2]*Assistant Professor, ECE, Chandigarh Engineering College, Landran*

Abstract - In terms of execution the parallel prefix adding units are the most performance giving in VLSI designs. But, this concert feature cannot not be translated to FPGA designs in direct manner since of force on logic chunk composition as well as routing in the clouds. This term document represents various varieties of carry-tree adders like the Kogge-Stone, sparse Kogge-Stone, and spanning tree adder and comparison with other carry adders is done. Brief study of various adder units with respect to speed, area consumtion, power consumption is made. Suitable to the attendance of a fast carry-chain, the RCA designs show better delay showing.

I. INTRODUCTION

The main component in almost every digital circuit styles jointly with digital signal processing units and micro chip data pathway units is binary adder. While, deep study continues to be pointed on improving the competence delay execution of the adder [1]. In Very Large Scale Integration designs, parallel-prefix adders are adapted to have for the majority part efficient execution. Reconfigurable logic similar to Field Programmable Gate Arrays has been profiting in features in last year's, therefore offering high execution in conditions of speed and power expenditure in contrast to DSP-based and microprocessor-based answers for numerous levelheaded manners which involve mobile DSP and telecommunications fields and minimization in rise time and price above Application Specific computer circuit designs. The ability benefit is chiefly vital with the increasing feature of mobile and moveable physical science, that make intensive use of DSP functions [2].

Yet, due to the make of the configurable logic and routing income in Field Programmable Gate Arrays, parallel-prefix adders can have a unique recital than Very Large Scale Integration utilization. Distinctively, mainly up-to-date Field Programmable Gate Arrays use a fast-carry chain that optimizes the carry alleyway for the clear-cut Ripple Carry Adder. Throughout this article, the sane harms worried in upcoming with and implementing tree-based adders on Field Programmable Gate Arrays are explain. Correlate degree inexpensive testing line of attack for evaluating the act of those adders is mentioned [3]. Many tree-based adder structures are forced and characterised on a Field Programmable Gate Arrays and compared with the Ripple Carry Adder and also the Carry Skip Adder. At last, a few conclusions and propounding for rising Field Programmable Gate Arrays styles to change higher tree-based adder show are given [4].

II. LITERATURE REVIEW

In (1) the comparison of three types of carry tree adders i.e. Kogge stone adder, spanning tree adder and sparse kogge stone adder is done with simple ripple carry adder and carry skip adder. The performance of these adders is discussing an well-organized testing setup for quality. The adders which are to be deliberate designed with 128 bit widths and coded in VHDL. To manufacture the design onto the Spartan 3E FPGA, Xilinx ISE 12.2 software is used. The lesser show of the carry skip adders is owing to routing over head and LUT delay. At low to moderate bit width, parallel prefix adders are not of use as simple ripple carry adders. However, the feat of beeline adder designs at soaring bit widths better than carry tree adders surpass.

In (2) four type of parallel prefix adders (PPA) i.e. Kogge stone adder (KSA), Sparse Kogge stone adder (SKA), spanning tree adder (STA) and Brent Kung adder (BKA) are discussed. Also, Ripple carry adder (RCA), Carry look ahead adder (CLA) and Carry skip adder (CSA) are discussed. The delay performance, power and area of these adders are discussed. Out of all adders CSA has more delay and RCA has less delay. STA has much better delay of above four parallel prefix adder. Out of all seven adders RCA has taken fewer area in terms of LUT's and IOB's. The power of all adders is approximately 1.1 nanoseconds. The designs are performed in Xilinx virtex 5 FPGA's.

In (3) an efficient algorithm is proposed to synthesize prefix chart assemble that turnout adders with best work area trade-off. This algorithm is effective in minimizing the size of prefix chart for given bit width n and random logic level than all other existing algorithms. Because algorithm can handle fan-out wire length check on nodes in prefix table level which also improves performance of adder. To make size optimum prefix graphs for 2^m bit adders with stage m is demonstrated in this. To evaluate the placement and wire congestion, a lot of candidate prefix graph frame are generated for a known set of constraints. Timing and wire distance end to end for high performance adders is yielded for better quality of results. The proposed approach has improved elastic up to 3% better delay and 9% better area.

In (4) Carry save and kogge stone parallel prefix adders are obtainable to solve the bad situation of long carry chains using FPGA based modulo multiplication. Enforced on Xilinx virtex-6 family device and the future planning are synthesized.

163 bit ripple carry adder, carry save adder and kogge stone adder design are used for planned execute. The delay is reduced by 74.20% and 28.82% in proposed architectures and is of high speed. Montgomery modulo multiplication is used to implement the popular RSA algorithm.

In (5) to build parallel prefix adders, the implementation of necessary basic building blocks is presented. Through the building of 4 bit PPA, these blocks has been evaluated and validated. The main focus is on PPA architectures. PPA algorithm is presented in this paper is design of complex adder. For the conversion from Boolean expressions to common equations the basic steps which were the creation of discrete gate. In this literature, the projected adder usage is collated to other QCA adders.

In (6) Data processing application specific integrated circuits and digital signal processors is calculation of two binary numbers because the simple mean process are on microprocessors. For speeding up binary addition parallel prefix adders are used. Four unlike category prefix cell operators, even-dot cells, odd-dot cells, even-semi-dot cells and odd-semi-dot cells is used to design the wished-for 64-bit adder; these four cells used for low power and high feat because it offers robust adder solutions. Four different prefix cells were used to design the anticipated low power 64-bit parallel prefix adder and observe with existing parallel prefix adders such as 8-bit PPA, 16-bit PPA, and 32-bit PPA.

In (7) A error broadminded parallel prefix adder can be completed by using a kogge stone arrangement due to inbuilt redundancy in the carry tree. This paper proposes the use of an adder that is able of both error correction and fault detection which is called sparse kogge stone adder. A number of smaller ripple carry adders are used to complement the sparse carry tree. Fault tolerance is achieved by two additional ripple carry adders. A triple-mode outmoded ripple carry adder (TMR-RCA) is used as a point of suggestion. Synthesis and simulation is carried out for FPGA stage. There are mention that the Kogge-Stone adder gives best recital over an RCA when equipment on an FPGA for very large bit widths. Simulation and synthesis has acceptance the achievement of the lower-half and upper-half SKS for fault detection and improvement by using the FPGA design tools.

In (8) a high presentation and below power 8 bit parallel prefix adder format is designed. The meaning of low power is in aerospace appliance and is linked to point of packaging and cost. For better speed and to decrease the power, static and dynamic power is reduced. Si tempo tools are used to analyze the CMOS circuits. Structural style of modelling style and a parallel prefix adder design is passed out at frequency of 200.00 MHz is used to optimize the proposed design. Only propagation signals and use of multiplexer's leads to large reduction in power and delays are used to implement the design. X-Power Tool Spartan3_XPE_11_1.xls at voltage levels of 1.14V and 2.375V giving extra access outcomes are used to estimate the power. The static power is twenty two mega watt as dynamic is two mega watt.

In (9) Parallel-prefix computation graph based carry look-ahead adder architecture is presented. In proposed methodology the computation of generated and propagated signal for a merged block is defined by a triple-carry-operator and the merged block is combined by three next blocks. Timing driven character of proposed design is used to reduce the depth of the adder. The future adder exhibits very conducive timing area trade-off individuality. The trial and error aftereffect specify that our aim adder with some area overhead is considerably active than the approved BK adder. The plan adder display faster act than fast KS adder with expressive area savings.

In (10) prefix adders with area and timing constraints are proposed to build minimal power in integer programming method. This paper contains consumption of static power and dynamic power. It switches gate sizing buffer addition to get better the act and counts both wiring and gate capacitances. To construct finest prefix adder with minimum power consumption, the prefix adder synthesis trouble is formulated to integer linear programming problem. Output times and non uniform input coming times necessary for each purpose are handled.

III. CONCLUSION

A high speed adder must be required along with the lesser delay methodology of the carry generation logic. Sparse Kogge Stone adder is implemented using the proposed methodology. The number of black cells and grey cells are greater in kogge stone adder than Sparse Kogge Stone adder. Carry is generated by using sparse technique. Among all of the above adders, Carry select adder is the best adder which is used for the final calculation of the sum.

IV. REFERENCES

[1] David H. K. Hoe, Chris Martinez and Sri Jyothsna Vundavalli. "Design and Characterization of Parallel Prefix Adders using FPGAs." 978-1-4244-9593-1/11/$26.00 ©2011 IEEE.

[2] Sudheer Kumar Yezerla, B Rajendra Naik. "Design and Estimation of delay, power and area for Parallel prefix adders." Proceedings of 2014 RAECS UIET Panjab University Chandigarh, 06 - 08 March, 2014

[3] Subhendu Roy, Student Member, IEEE, Mihir Choudhury, Member, IEEE, Ruchir Puri, Fellow, IEEE, and David Z.' "Towards Optimal Performance-Area Trade-Off in Adders by Synthesis of Parallel Prefix Structures." IEEE transactions on computer-aided design of integrated circuits and systems, vol. 33, no. 10, October 2014

[4] Pravin P. Zode, Raghavendra B. Deshmukh. "Fast Modular Multiplication using Parallel Prefix Adder." 978-1-4799-6986-9/14/$31.00©2014 IEEE

[5] Kim A. Escobar, Renato P. Ribas. "Parallel Prefix Adder Design Using Quantum-dot Cellular Automata." 978-1-4799-1132-5/13/$31.00 ©2013 IEEE

[6] K.Nehru, A.Shanmugam, S.Vadivel. "Design of 64-BitLow Power Parallel Prefix VLSI Adder for High Speed Arithmetic Circuits."

[7] Chris D. Martinez, L. P. Deepthi Bollepalli, and David H. K. Hoe. "A Fault Tolerant Parallel-Prefix Adder for VLSI and FPGA Design." 44th IEEE South eastern Symposium on System Theory University of North Florida, Jacksonville, FL March 11-13, 2012

[8] Shrinivas K Saptalakar, Manjunath Lakkannavar. "Design and Analysis of 8-bit Low Power Parallel Prefix VLSI Adder." International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-2, Issue-1, March 2013

[9] Sabyasachi Das and Sunil P. Khatri. "A Novel Hybrid Parallel-Prefix Adder Architecture with Efficient Timing-Area Characteristic." IEEE transactions on very large scale integration (VLSI) systems, vol. 16, no. 3, march 2008

[10] Jianhua Liu, Yi Zhu, Haikun Zhu, Chung-Kuan Cheng, John LilliS. "Optimum Prefix Adders in a Comprehensive Area, Timing and Power Design Space." 1-4244-0630-7/07/$20.00 ©2007 IEEE.