



A Review on Implementation of Parallel Prefix Adders using FPGA'S

S. Lakshmipriya - PG (Scholar)

M.E, Embedded Systems, Department of Electronic and Instrumentation Engineering,
Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu

ABSTRACT

The paper mainly used in the implementation of parallel prefix adders using FPGA'S. The carry-tree adders constitute spanning tree adder, sparse Kogge-Stone & Kogge-Stone adder. It also presents a representation of the carry skip adder. The best resolution of the VLSI system is obtained by the Prefix adders. Thus the implementation of the block diagram is difficult so the FPGA technology is used instead. They produce the high efficiency of the layout using the FPGA analysis. Better delay and performance of the RCA design in the 128 bits in the use of the fast-chain array. Xilinx Spartan 3E is used in the implementation of the design. RCA is most commonly used in the implementation of the full adder and half adder. RCA is a serial which can perform many number of the addition but due to some situation delay occurs. Xilinx ISE is used in the simulation of the design.

I. INTRODUCTION

In the binary addition unit, the main component in the majority digital path classes together with digital signal processors as well as micro chip find out passageway units. Generally, severe examination carries on being beneath fire at rising the capability holdup time of the adder (Hoe, D.H.K., *et al.*, 2011). Reconfigurable logics such as Field Programmable Gate Arrays are most popular from the past few years and are latest in technology. As significance, it provides us with the benefits of improved rate and power for DSP and microprocessor based resolutions for abundant reasonable styles including mobile DSP as well as telecommunications applications as well as a deep degradation in progress time as well as price on to

Application Specific computer circuit designs. The skill benefit is mainly key with the increasing value of mobile as well as portable physical science, which causes austere use of DSP functions (Yezerla, S.K. and B. Rajendra Naik, 2014).

Though, because of the structure of the configurable logic as well as steering income in FPGA, parallel-prefix adders can have an improved performance than Very Large Scale Integration implementations. Field Programmable Gate Arrays which are up to date use a fast-carry chain that enhances the carry passageway for the Ripple Carry Adder. From this paper, problems appeared in coming with as well as designing tree-based adders on Field Programmable Gate Arrays are labeled. Assistant degree in luxurious testing criteria for contrasting the feat of those adders is described (Roy, S., *et al.*, 2014). Numerous tree-based adder designs are compulsory and featured on a Field Programmable Gate Arrays as well as estimated with the Ripple Carry Adder as well as also the Carry Skip Adder.

The binary adder is the critical element in most digital circuit designs including digital signal processors (DSP) and microprocessor data path units. As such, extensive research continues to be focused on improving the power delay performance of the adder. In VLSI implementations, parallel-prefix adders are known to have the best performance. Reconfigurable logic such as Field Programmable Gate Arrays (FPGAs) has been gaining in popularity in recent years because it offers improved performance in terms of speed and power over DSP-based and microprocessor-based solutions for many practical designs involving

mobile DSP and telecommunications applications and a significant reduction in development time and cost over Application Specific Integrated Circuit (ASIC) designs. The power advantage is especially important with the growing popularity of mobile and portable electronics, which make extensive use of DSP functions.

However, because of the structure of the configurable logic and routing resources in FPGAs, parallel-prefix adders will have a different performance than VLSI implementations. In particular, most modern FPGAs employ a fast-carry chain which optimizes the carry path for the simple Ripple Carry Adder (RCA). In this paper, the practical issues involved in designing and implementing tree-based adders on FPGAs are described. An efficient testing strategy for evaluating the performance of these adders is discussed. Several tree-based adder structures are implemented and characterized on a FPGA and compared with the Ripple Carry Adder (RCA) and the Carry Skip Adder (CSA). Finally, some conclusions and suggestions for improving FPGA designs to enable better tree based adder performance are given.

II. Parallel prefix adder:

It is a sort of a carry look ahead. It is designed in innumerable ways. These are fast adders and they are rapid adders. They are mainly used in the arithmetic calculations. The addition is mainly used here is a three step process.

Pre-processing adder:

It is worked out produce and broadcast signals. A and B are the inputs. It generates a carry output for the each input. These signals are given by the equations

$$P_i = A_i \text{ xor } B_i$$

$$G_i = A_i \cdot B_i$$

Carry generation network:

The carry is generated in the processing of the each bit. This is done by the parallel process. To process the data in parallel they are substituted in the small items. The process of generation of the carry bit and the addition by the process of AND gate and the OR gates. The following equations give the carry generation:

$$P_{i:k} = P_{i:j} \cdot P_{j-1:k}$$

$$G_{i:k} = G_{i:j} + (G_{j-1:k} \cdot P_{i:j})$$

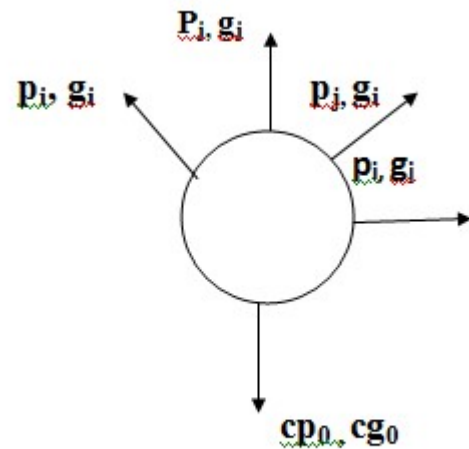


Fig:1 Carry Operator

Post process stage:

This is the final work out of the neglected set of instructions. This is similar for the adders.

$$S_i = P_i \text{ xor } C_i$$

$$C_i = (P_i \cdot C_0) + G_i$$

Design:

High performance adders, also known as carry-tree adders, pre-compute the propagate and generate signals. In tree adders, carries are generated in parallel and fast computation is obtained at the expense of increased area and power. The main advantage of the design is that the carry tree reduces the number of logic levels (N) by essentially generating the carries in parallel.

In the proposed approach a high speed parallel prefix adder is proposed. In the proposed methodology, a high speed kogge stone adder is realized using high speed adder logic. The overall performance of the system is aimed to be improved in conditions of both area and timing complexity. In the proposed method the carry select adder is divided into further four blocks. In the first unit simple half adder logic is implemented.

The parallel-prefix tree adders are more favorable in terms of speed due to the complexity $O(\log_2 N)$ delay through the carry path compared to that of other adders. The prominent parallel prefix tree adders are Kogge-Stone, Brent-Kung, Han-Carlson, and Skanska. It was found from the literature that Kogge-stone adder is the fastest adder when compared to other adders. The adder priority in terms of worst-case delay is found to be Ripple-Carry, Carry-Look-Ahead, Carry- Select and Kogge-Stone. This is due to the number of "Reduced stages". Kogge-Stone adder implementation is the most straightforward, and also it has one of the shortest

critical paths of all tree adders. The drawback with the Kogge- Stone adder implementation is the large area consumed and the more complex routing (Fan-Out) of interconnects. A Parallel Prefix Adder (PPA) is equivalent to the CLA adder. The two differ in the way their carry generation block is implemented.

The parallel prefix carry look- ahead adder was first proposed some twenty years ago as a means of accelerating n-bit addition in VLSI technology. It is widely considered as the fastest adder and used for high performance arithmetic circuits in the industries. A three step process is generally involved in the construction of a Parallel Prefix Adder. The first step involves the creation of generate and propagate signals for the input operand bits. The second step involves the generation of carry signals. In the final step, the sum bits of the adder following stages of the operand bits and the preceding stage carry bit using a xor gate.

RESULT AND DISCUSSION:

The proposed technique is executed using the Xilinx Spartan 3 FPGA and Verilog language is used for the implementation of the design. The technique is associated with the basic RCA approach and the result shows an enhancement in both the area consumed by the device and the maximum combinational path delay. In Table 1 the comparison of both the approaches is shown on the basis of various performance parameters like number of slices, number of look up tables, delay and maximum operating frequency. Number of slices, number of 4 input LUTs are a amount of the number of resources used by the FPGA which comes out to be less in case of the proposed approach.

CONCLUSION:

This paper presents an innovative way of modifying the already existing Kogge-Stone adder by re-routing (wiring) and Black-cell reduction for increasing the speed of execution. The design originates from the principle of removing the redundant black cells and compensating this removed cells by rerouting. Arithmetic unit is the most widely researched and used designs in the digital electronics circuits. Many researchers in recent past have proposed different approaches for the design of adders and other logical circuits and compared their merits and demerits with the existing approaches. Parallel prefix adder is designed in this approach by using optimized CSLA in terms of area and speed.

REFERENCE:

1. Ahmet Akkas and Michael J Schult (2011), "A Decimal Floating-Point Fused Multiply-Add Unit with a Novel Decimal Leading-Zero Anticipator", **AMD IEEE**.
2. Anitha R and Bagyaveereswaran V (2012), "High Performance Parallel Prefix Adders with Fast Carry Chain Logic", *International Journal of VLSI Design & Communication Systems*, Vol. 3, No. 2, pp. 01-10, ISSN 0976-6499.
3. David Jeff Jackson and Sidney Joel Hannah (1993), "Modeling and Comparison of Adder Designs with Verilog HDL", 25th South-Eastern Symposium on System Theory, March, pp. 406-410.
4. Kogge P and Stone H (1973), "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Relations", *IEEE Transactions on Computers*, Vol. C-22, No. 8, pp. 786-793.
5. Krishna Kumari V, Sri Chakrapani Y and Kamaraju M (2013), "Design and Characterization of Koggestone, Sparse Koggestone, Spanning Tree and Brentkung Adders", *International Journal of Scientific & Engineering Research*, Vol. 4, No. 10, pp. 1502-1506, ISSN 2229-5518.
6. Madhu Thakur and Javed Ashraf (2012), "Design of Braun Multiplier with Kogge-Stone Adder & Its Implementation on FPGA", *International Journal of Scientific & Engineering Research*, Vol. 3, No. 10, pp. 03-06, ISSN 2229-5518.
7. Nurdiani Zamhari, Peter Voon, Kuryati Kipli, Kho Lee Chin and Maimun Hujahusin (2012), "Comparison of Parallel Prefix Adder (PPA)", *Proceedings of the World Congress on Engineering, WCE*, Vol. 2, July 4-6, London, UK.