International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 3, Issue-5 (September-October 2015), PP. 116-119

VHDL IMPLEMENTATION OF 32-BIT SPARSE KOGGE-STONE ADDER USING CARRY SELECT LOGIC

Shubham1 And Navdeep Prashar2
Bahra University Waknaghat
shubhamsaini555@gmail.com

Abstract— **Parallel prefix adder is used for speeding up the logical operation of the system. Implementation of parallel prefix adder's structure in VLSI have effective performance. The different types of parallel prefix adder structures are Kogge-Stone , Brent-Kung, Sparse Kogge-Stone adder etc. have been proposed previously. Among them Kogge- Stone adder is the fastest adder structure. Sparse Kogge-Stone adder is the sub-type of Kogge-Stone adder in which it uses less black cells and grey cells as compared with the Kogge-Stone adder and final sum is calculated through ripple carry adder. In this paper, firstly Basic Sparse Kogge Stone adder is implemented and secondly, implementation of the Sparse Kogge-Stone adder using carry select logic is performed that result in reduction in critical path delay and increase in speed. On verifying its synthesis report it is observed that it requires 72 of its total numbers of slices with minimum path delay of 18.830 ns and a maximum frequency of 53.10 MHz of modified Sparse Kogge-Stone adder. On comparing with Basic Sparse Kogge-Stone adder and previous reference paper, it is observed that the modified design shows the improvement in speed with considerable reduction in delay.**

Key Words: Parallel Prefix Adder, Kogge stone Adder, Carry Select Adder

I. INTRODUCTION

The binary adder is that the important component in most digital circuit styles together with digital signal processors and micro chip information path units. As such, intensive analysis continues to be targeted on raising the facility delay performance of the adder [1]. In Very Large Scale Integration implementations, parallel-prefix adders are glorious to own the most effective performance. Reconfigurable logic like Field Programmable Gate Arrays has been gaining in quality in recent years as a result of it offers improved performance in terms of speed and power over DSP-based and microprocessor-based solutions for several sensible styles involving mobile DSP and telecommunications applications and a big reduction in development time and price over Application Specific computer circuit styles. The facility advantage is particularly vital with the growing quality of mobile and moveable physical science, that create intensive use of DSP functions [2].

However, due to the structure of the configurable logic and routing resources in Field Programmable Gate Arrays, parallel-prefix adders can have a special performance than Very Large Scale Integration implementations. Specifically, most up-to-date Field Programmable Gate Arrays use a fast-carry chain that optimizes the carry path for the straightforward Ripple Carry Adder. During this paper, the sensible problems concerned in coming up with and implementing tree-based adders on Field Programmable Gate Arrays are delineate. Associate degree economical testing strategy for evaluating the performance of those adders is mentioned [3]. Many tree-based adder structures are enforced and characterised on a Field Programmable Gate Arrays and compared with the Ripple Carry Adder and also the Carry Skip Adder. Finally, some conclusions and suggestions for rising Field Programmable Gate Arrays styles to change higher tree-based adder performance are given [4].

A. 1.1Parallel prefix adders

The PPA is sort of a Carry Look Ahead Adder. The assembly of the carriers the prefix adders will be designed in many various ways that supported the various needs. We have a tendency to use tree structure type to extend the speed of operation. Parallel prefix adders are quicker adders and these ar quicker adders and used for prime performance arithmetic structures in industries [5]. The parallel prefix addition is completed in three steps.

1) Pre-processing stage
2) Carry generation network
3) Post process stage

B. 1)Pre-processing stage

In this stage we have a tendency to work out, generate and propagate signals are wont to generate carry input of every adder [6]. A and B are the inputs. These signals are given by the equation 1&2.

$$P_i = A_i \; xor \; B_i \qquad (1)$$

$$G_i = A_i . B_i$$

(2)

C. 2)Carry generation network

In this stage we have a tendency to work out carries such as every bit. Execution is completed in parallel type [5]. Once the computation of carries in parallel they're divided into smaller items. Carry operator contain 2 AND gates, one gate. It uses propagate and generate as intermediate signals that are given by the equations 3& 4.

$$P_{i:k} = P_{i:j} \cdot P_{j-1:k} \quad (3)$$

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

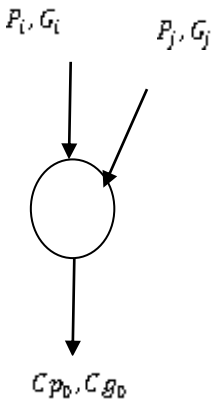

Fig.1: Carry Operator

D. 3)Post process stage

This is the ultimate stage to work out the summation of input bits. This is same for all adders and adds bit equation given

$$S_i = P_i \, xor \, C_i \quad (5)$$

$$C_{i+1} = (P_i \cdot C_0) + G_i \quad (6)$$

1) 1.2Parallel Prefix Adders [7] are classified into

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

E. 1)Kogge - Stone Adder

Kogge-Stone adder may be a parallel prefix type carry look ahead adder. The Kogge-Stone adder was developed by peter M. Kogge and Harold S. Stone that they printed in 1973. Kogge-Stone prefix adder may be a quick adder style [8]. American state adder has best performance in VLSI implementations. Kogge-Stone adder has massive space with minimum fan-out. The Kogge- Stone adder is wide referred to as a parallel prefix adder that performs quick logical addition. Kogge-Stone adder is employed for broad adders due to it show the less delay among different architectures. In fig2 every vertical stage produces Propagate and Generate bits. Generate bits are created within the last stage and these bits are XORed with the initial propagate once the input to supply the add bits [9]. The 2-bit and 32- bit Kogge- Stone adder figures shown below.

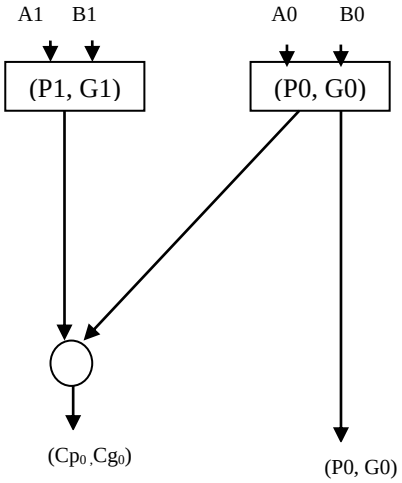

Figure 2: 2-bit KS adder

F. 2.PRESENT WORK

In present work, firstly basic Sparse Kogge-Stone adder is implemented and secondly modified Sparse Kogge-Stone is implemented using carry select logic for increasing the speed of basic Sparse Kogge-Stone adder. In the basic Sparse Kogge -Stone adder the numbers of black and grey cells used are less in number as compared to the Kogge-Stone adder and final sum calculated through ripple carry adder. In the Modified Sparse Kogge-Stone adder, modification in the architecture of Sparse Kogge-Stone adder. Along with the Sparse technique used in the generation of carry, the adder used for the final calculation of sum is carry select adder which is the fastest known adder available in the literature. Fig.3 below shows the block diagram of the modified Sparse Kogge-Stone Adder.

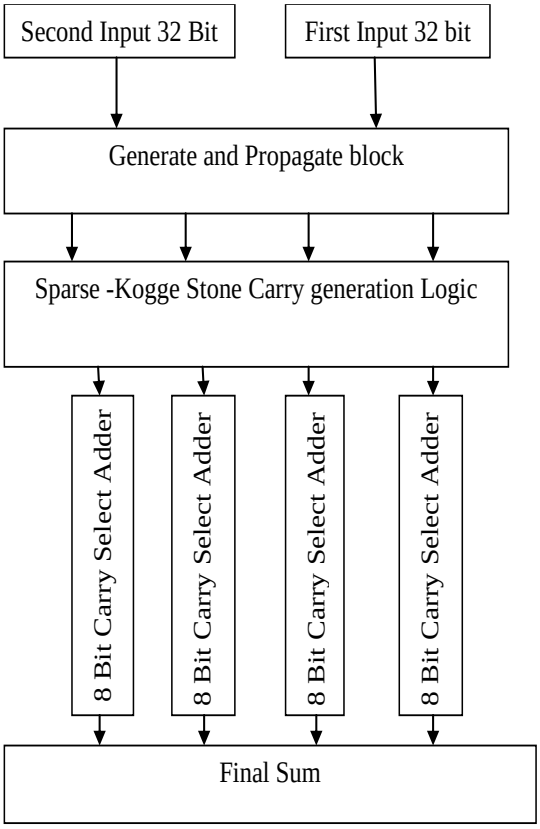

Fig.3: Block Diagram of Modified Sparse Kogge-Stone Adder

RTL View of the Modified Sparse Kogge-stone adder using carry select logic as shown in Fig4

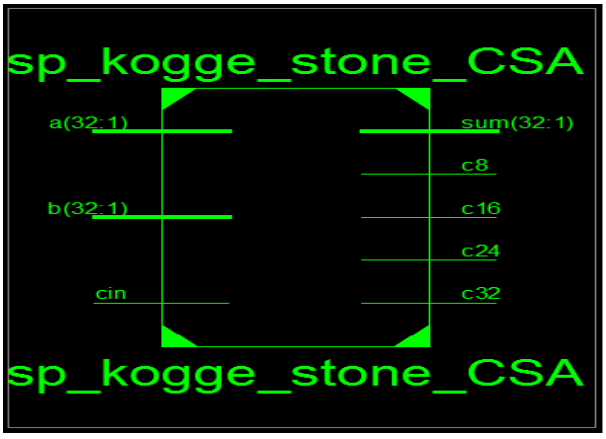

Fig.4: RTL View Modified Sparse Kogge-Stone Adder

G. 3.RESULTS AND SIMULATION

The basic and modified Sparse Kogge-Stone adder is implemented in Xilinx ISE12.3 and the simulations are shown in Xilinx ISim in VHDL language. The device used for the simulation of the basic and modified Sparse Kogge-Stone adder is Spartan 3E. Fig.4 shows the RTL View of the Modified Sparse Kogge-Stone adder. From the results shows that the modified parallel prefix Sparse Kogge-stone adder has faster as compared with the Basic Sparse Kogge-stone adder. The delay of the Modified Sparse Kogge-Stone Adder comes out to be 18.830 ns and the basic Sparse Kogge-Stone is 19.895 ns. The Fig.5 shows the simulation of Modified Sparse Kogge-Stone Adder:

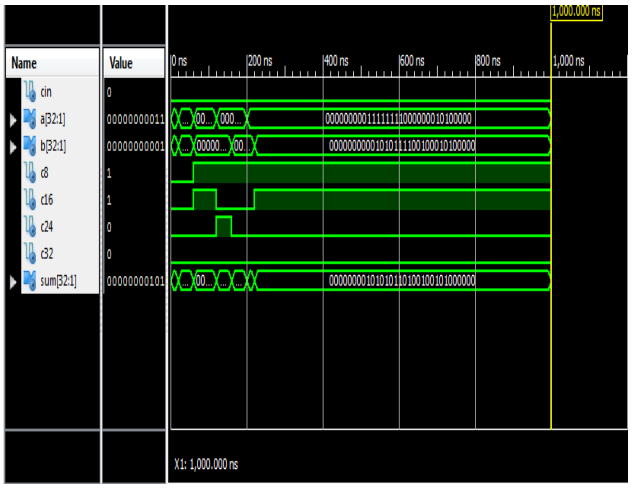

Fig.5: Simulation of Modified Sparse Kogge-Stone Adder

H. 4. Comparison of Modified and Basic Sparse Kogge-Stone Adder and reference paper

Modified and Basic Sparse Kogge-Stone Adder is implemented in Xilinx ISE12.3 and the simulations are shown in ISim in VHDL language. Thus, Finally comparison of Modified and Basic Sparse Kogge-Stone Adder and reference paper

Table 6.3 Comparison between modified and basic Sparse Kogge-Stone adder and reference paper [10]

Parameters	**Modified Sparse Kogge-Stone adder**	**Basic Sparse Kogge-Stone adder**	**Reference [10]**
No. of Slices	72	41	134
No. 4 input LUTs	128	72	235
No. of bonded I/Os	101	93	264
Delay (ns)	18.830	19.895	26.543
Frequency (MHz)	53.10	50.26	37.67

II. .CONCLUSION

Parallel Prefix Adder is a type of process that increase the speed of arithmetic operations of the system. The simulation and synthesis of the Sparse Kogge-Stone adder have been performed on ISim (VHDL/Verilog) simulator. The area required has been measured in terms of slices, flip-flops, LUTs and IOBs. The synthesis results shows that utilization of the number of slices has been found to be 72 out of 960, number of IOBs are 101 out of 66. Speed has been calculated to be equal to 53.10MHz with delay of 18.830ns of modified Sparse Kogge-Stone adder. On comparing with Basic Sparse Kogge-Stone adder and previous reference paper, it is observed that the modified design shows the improvement in speed with considerable reduction in delay. So, the Modified Sparse Kogge-Stone adder can be used for various high speed applications. In future scope ,the parallel prefix adders must be tested for other adders also to optimize the area and timing both.

REFERENCES

[1] Padmajarani, S. V., and M. Muralidhar. "A New Approach to implement Parallel Prefix Adders in an FPGA." (2012): 1524-1528.

[2] Jianhua LiuZhu, Haikun, Chung-Kuan Cheng, John Lillis, “Optimum prefix Adders in a Comprehensive Area,Timing and power Design Space”., Proceeding of the 2007 Asia and South pacific Design Automation conference.Washington, pp.609-615,jan 2007.

[3] Taeko Matsunaga and Yusuka Matsunaga., “Timing-Constrained Area minimization Algorithm for parallel prefix adders”, IEICE TRANS, Fundamentals, vol.E90-A, No.12 Dec, 2007.

[4] Hoe, David HK, Chris Martinez, and Sri Jyothsna Vundavalli. "Design and characterization of parallel prefix adders using FPGAs." System Theory (SSST), 2011 IEEE 43rd Southeastern Symposium on. IEEE, 2011.

[5] Giorgos Dimitrakopoulos and Dimitric Nikolos, “High Speed Parallel –Prefix VLSI Ling Adders”, IEEE Trans on computers, Vol.54, No.2, Feb 2005

[6] V.Choi and E.E.Swartz lander, Ir, “Parallel Prefix adder design with matrix representation”,, in Proc.17th IEEE symp, comput.Arithmatic (ARITH), PP 90-98,2005

[7] John F.Wakerly, Digital Design Principles and Practices, 4th Edition, Pearson Education, 2009.

[8] Saptalakar, Shrinivas K., and Manjunath Lakkannavar. "Design and Analysis of 8-bit Low Power Parallel Prefix VLSI Adder."

[9] Rajaram, Srinath, and K. Vanithamani. "Improvement of Wallace multipliers using parallel prefix adders." Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011 International Conference on. IEEE, 2011.

[10] Geeta Rani1 and Sachin Kumar, " Implementation of 128-bit Sparse Kogge-Stone Adder using verilog" Vol. 2, Issue 4, PP. 90-92, July-Aug 2014.