FPGA Implementation of Sparse Tree Adder

P. Suveetha Dhanaselvam [1], R. Iswarya Lakshmi [2] ,I. Shanmuga Priya [3] , S. Deepa [4]
[1 ,2,3,4] Velammal College of Engineering and Technology
[1] Associate Professor [2, 3, 4] B.E final year

***Abstract* - Sparse tree adder is a parallel prefix adder which is the advancement of kogge stone adder In this paper, we design the sparse tree adder using Xilinx Spartan 3E simulation tool.Weobtain the latency,number of LUT'S and Hardware Utilization using this tool. .Compared to the other parallel prefix adders, Sparse tree adder isfaster andoccupies less area. Because of this advantage over the other adders, now a days most of the processors have started using sparse tree adder. Further the output can be obtained for various bit rate. In this paper, a comparative study of sparse tree adder and the existing koggestone adder were analyzed.**

Keywords: Prefix adder,koggestoneadder,latency

INTRODUCTION

In Computers and Processors, adders are used in the Arithmetic Logic Unit(ALU).Hence the most commonly used Digital circuit is an adder. Till now, Parallel Prefix adders like Koggestone adder, Han Carlson adder, Brent kung adder are used to perform addition [1], [2]. However these adders provide carry to every individual bit of the adder. Due to high wiring congestion it occupies more area. Therefore it requires more time to perform the addition operation. The Koggestone adder is said to be fastest among other types of prefix adders [6-9]. Hence it is widely used in processors for high performance arithmetic circuits. In contrast Sparse tree adder provides group carry to every 4-bit summation block. It is the main advantage of the sparse tree adder over other parallel prefix adders.

In VLSI domain, the sparse tree was implemented using the ISTEC 10kA/cm2 ADP(Advanced Data process) fabrication process [3-5]. In the ADP Fabrication process, Superconducting Rapid Single Flux Quantum logic is used. In this process the power consumption is low, eventhough it is difficult to test the adders at high frequency.

To overcome this limitation,we are implementing the sparse tree adder using Xilinx ISE Design suite 13.2. The construction of Sparse tree adder includes three stages: Initialization, Prefix tree and summation. In the Initalization stage, it includes Generate and Propagate stage to provide generate and propagate signals. The prefix tree stage has carry merge block to generate group carry. In the final stage carry skip adder is used to calculate the output sum. Since carry skip adder reduces the delay with little effort when compared to other adders. By using the simulation results we calculated the speed and delay and also we analysed the power of the sparse tree adder.

Further the bit rate can be increased according to our requirements. By comparing the sparse tree adder with the existing Koggestone adder, we found that sparse tree adder is efficient in terms of area and latency.

PROPOSED METHOD

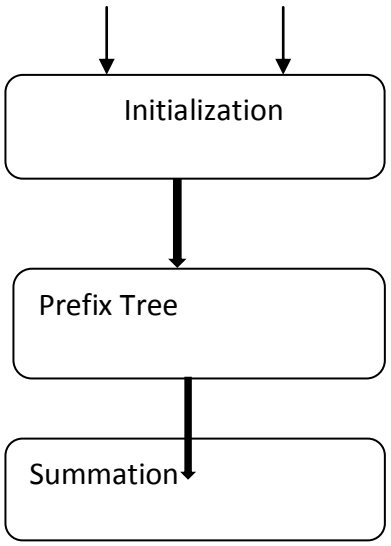

Fig.1. Block Diagram for Parallel Prefix adder

In this paper, we implement the 32 bit Sparse tree Adder using Xilinx ISE Design. The structural diagram of sparse tree adder includes three stages: Initialization, prefix tree, summation.

The *Initialization* stage receives two operands A and B. This stage consists of GPR block. The GPR block is used to create the prefix signals i.e Generate and Propagate signals. The bitwise Prefix functions are given as ,

$$G = A.B$$
$$P = A \wedge B$$

The *Prefix tree*stage consist of sparse Carry merge block [10]. The Carry merge block is used to generate the first predetermined number of carry signals based on the generate and propagate signals.

Similarly second predetermined number of carry signals will be generated using another carry merge circuit. Those predetermined number of carry signals will be again merged to calculate the group carry.

The Group carry should be calculated for every 4 bit summation block. To merge the generate and propagate signals, the following equations are used as follows which is given below,

$$G_j = (P_i \& G_i(prev)) + G_i$$
$$P_j = P_i \& P_i(prev)$$

The *Summation* Stage is the last and final step in the parallel prefix adder to perform the addition operation. It consists of Carry skip adder to calculate the output sum. It consists of a ripple carry adder block, an AND gate and a multiplexer. Carry skip adder is used to speed up the addition operation. Hence carry Skip adder is preferred [12].

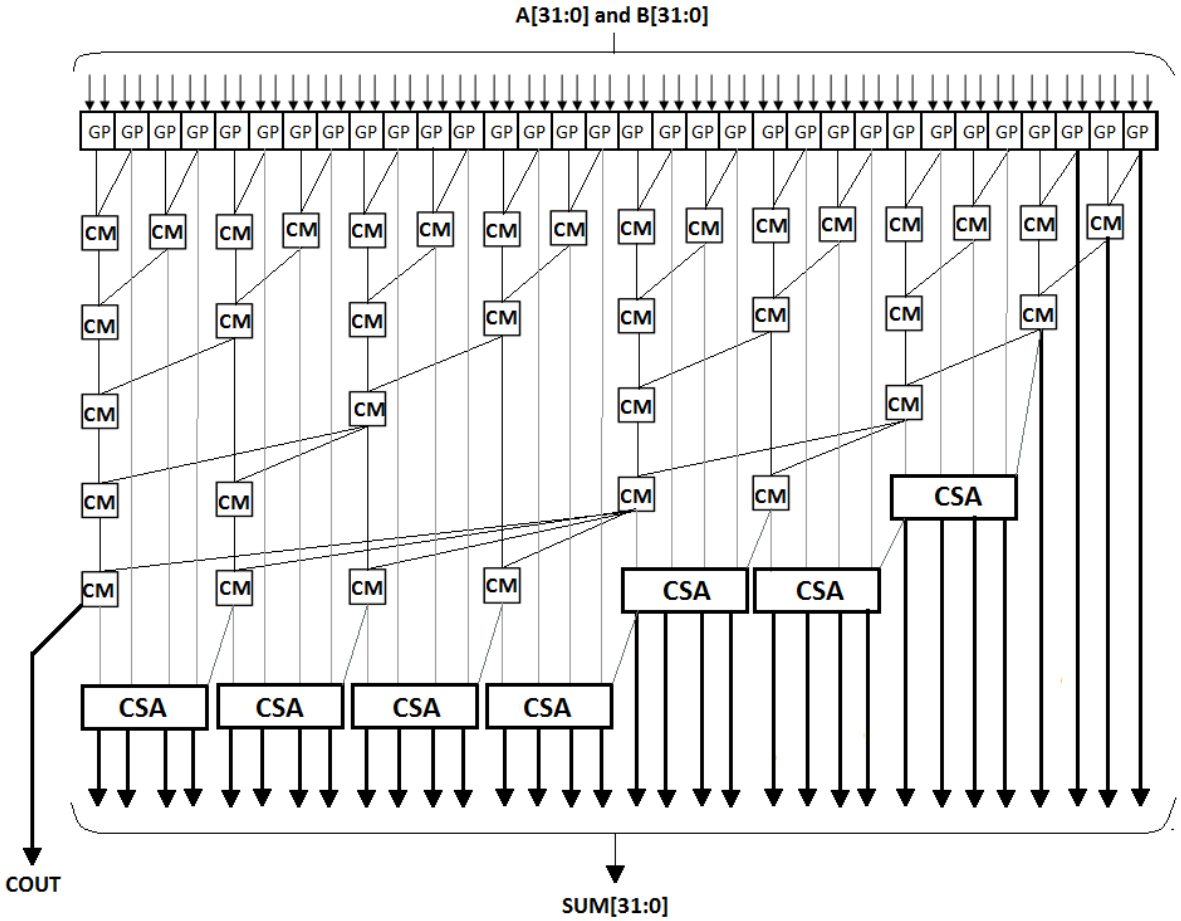

Fig.2. Structural diagram of the 16-Bit Sparse tree adder

RESULTS AND DISCUSSION

The 32 bit Sparse tree is designed in Xilinx ISE design suite 14.7 using verilog Hardware Description Language. In this proposed architecture, the sparse tree adder using carry skip adder was developed and the corresponding values for delay, speed and LUT's was also observed. Similarly, the Koggestone adder is also implemented using Xilinx ISE design [11].

The results of the sparse tree adder is compared with the koggestone adder inorderto prove that the Proposed adder is efficient in terms of speed, delay and hardware utilization. The following table shows the comparison of existing koggestone adder and sparse tree adder.

Fig.3. Simulated output for 32 bit sparse tree adder

Fig.4. Simulated output for 32 bit koggestone adder

Table.1. Comparison between 32-bit sparse tree adder and 32-bit koggestone adder

parameters	32-BIT SPARSE TREE ADDER	32-BIT KOGGESTONE ADDER
No of slices	70	90
No of 4 input LUTs	117	170
No of bonded IOBs	105	98
Delay	10.018ns	18.53ns

CONCLUSION

We have implemented the 32 bit sparse tree adder using XilinxISE 13.2tool. The performance parameters - number of LUT's and delay areanalyzed. The results have been compared with the existing koggestoneadder. The analysis shows that the sparse tree adder is faster and area efficient to the other parallel prefix adders.

REFERENCES

[1] D. Harris, "A taxonomy of parallel prefix networks"in Signals, Systems and Computers, Conference Record of Thirty Seventh Asilomar Conference on, vol. 2, Nov. 2003.

[2] A.Beaumont-Smith, C.Lim, "Parallel Prefix adder design" IEEE symp.Comp. Arith., pp.218-225, June.2001.

[3] Mikhail Dorojevets, Christopher L.Ayala, Nobuyuki Yoshikawa,AkiraFujimaki, "16-Bit Wave Pipelined SparseTree RSFQ Adder" IEEE Trans. on Applied super Conductivity, Vol.23, No.3, Jun. 2013.

[4] Mikhail Dorojevets, Christopher L.Ayala, Nobuyuki Yoshikawa,AkiraFujimaki, "8-Bit Asynchronous Sparse tree Superconductor RSFQ Arithmetic logic unit with a rich set of Operations" IEEE Trans. on Applied Super Conductivity, Vol.23, Jun. 2012.

[5] M. Tanaka, H. Akaike, A. Fujimaki, Y. Yamanashi, N. Yoshikawa,S. Nagasawa, K. Takagi, and N. Takagi, "100-GHz single-flux-quantumbit-serial adder based on 10-kA/cm2 niobium process," IEEE Trans. Appl.Supercond., vol. 21, no. 3, pp. 792–796, Jun. 2011.

[6] B.Tapasvi, K.Balasinduri, B.Lakshmi, N.Udayakumar, "Implementation of 64-Bit Koggestone carry select adder with ZFC for efficient area" IEEE Trans. on Electrical,computer and communication Technologies, Aug. 2015.

[7] SudheerkumarYezerla, B.RajendraNaik, "Design and estimation of delay,power and area for Parallel Prefix Adders" IEEE Trans. on Engineering and computational sciences, April. 2014.

[8] David H.K.Hoe, Chris Martinez, Sri JyothsnaVundavalli, "Design and characterization of parallel Prefix adders using FPGAs" IEEE Trans. on system theory, April.2011.

[9] KonstantinosVitoroulis, Asim J, Al-Khalili, "Performance of parallel prefix adders implemented with fpga technology" IEEE Trans. on circuits and systems, April 2007.

[10] P.SaiPhaneendra, SreehariVeera, N.MoorthyMuthukrishnan, M.B.Srinivas, "Conditional sum block for High Sparse adders" IEEE Trans. on microelectronics and electronics, Nov.2011.

[11] Sunil M, Ankith R D "Design and implementation offaster parallel Prefix kogge stone adder" journal ofIJEETC on vol 3 no. 1 2014.

[12] Yu Shen Lin, DamuRadhaKrishnan, "Delay efficient 32-bit carry skip adder" IEEE Trans. on Electronics, Circuits and Systems, July 2007.