International Journal of Research
Available at
https://edupediapublications.org/journals

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 03
March 2017

An Efficient Error Correction and Detection for F.T 32 Bit Han-Carlson Adder

P. VIJAYA PARVATHI[1] P. BALA KRISHNA[2] S. KISHORE BABU [3]
(M.TECH), VIKAS GROUP OF INSTITUTIONS, NUNNA - JNTUK, India
Assistant Professor, VIKAS GROUP OF INSTITUTIONS, NUNNA -JNTUK, India
HOD, VIKAS GROUP OF INSTITUTIONS, NUNNA -JNTUK, India

ABSTRACT: *Binary addition is one of the most important arithmetic functions in modern digital VLSI systems. Adders are extensively used as DSP lattice filter where the ripple carry adders are replaced by the parallel prefix adder to decrease the delay. The requirement of the adder is that it is fast and secondly efficient in terms of power consumption and chip area.*

Parallel prefix adder is a technique for improving the speed of the addition. Parallel prefix adders provide a good theoretical basis to make a wide range of design tradeoffs in terms of area, delay and power. This technique is more suited for adders with wider word lengths. In this paper, a modified Parallel Prefix Han-Carlson Adder is introduced which uses different stages of Brent-Kung and Kogge-Stone adders which reduce the complexity of the adder design.

Key Words - Parallel Prefix Adders, Han-Carlson Adder, area, prefix computation, Power Consumption, delay

1. INTRODUCTION

VLSI binary adders are critically important elements in processor chips, they are used in floating-point arithmetic units, ALUs, and memory addresses program counter update and magnitude comparator. Adders are extensively used as a part of the filter such as DSP lattice filter. Ripple carry adder is the first and most fundamental adder that is capable of performing binary number addition. Since its latency is proportional to the length of its input operands, it is not very useful. To speed up the addition, carry look ahead adder is introduced. Parallel prefix adders provide a good result as compared to the conventional adders.

The adders with the large complex gates will be too slow for VLSI, so the design is modularized by breaking it into trees of smaller and faster adders which are more readily implemented. For large adders the delay of passing the carry through the look-ahead stages becomes dominated and therefore tree adders or parallel prefix adders are used.

High speed adders depend on the previous carry to generate the present sum. In integer addition any decrease in delay will directly relate to an increase in throughput. In nanometer range, it is very important to develop addition algorithm that provide high performance while reducing power.

Parallel prefix adders are suitable for VLSI implementation since they rely on the use of simple cells and maintain regular connection between them. We can define each prefix structures in terms of logic levels, fan out and wiring tracks. Zero or more inverters are added to each prefix cell output to minimize the delay based on this model, buffers are individually sized to minimize the delay, buffers are used to minimize the fan out and loading on gates since high fan out causes poor performance.

A modified Han-Carlson adder uses fewer number of prefix operations by adjusting the number of stages amongst Kogge-Stone and Brent-kung adder and thus reduces the area required by the adder circuitry.

There are three stages in performing prefix computation as shown in "Fig.1" below. First is the pre-processing stage to calculate generate and propagate bit, second stage is the carry computation stage to compute the carry bit and the third stage is the post-processing stage to compute the sum bit.

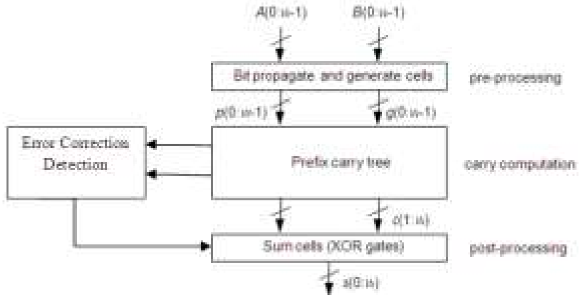

Fig.1 Parallel Prefix Adder Structure

The graph representation of Hybrid Han-Carlson Adder is shown in Fig.2 below

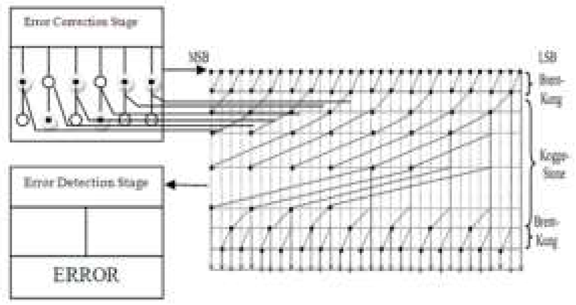

Fig. 2: Graph representation of 32-bit Hybrid Han-Carlson Adder

2. PREVIOUS WORK

The different types of parallel prefix adders available are Kogge-Stone adder, Brent-kung adder, Sklansky adder, Han-Carlson adder, Knowles adder and Ladner-Fischer adder. These adders offer tradeoffs among the number of stages of logic, the number of logic gates, fan out and amount of wiring between stages.

Kogge-Stone adder, Brent-kung adder and Sklansky adder are the fundamental adders. Brent-Kung uses minimal number of computation nodes which yields in reduced area but structure has maximum depth which yields slight increase in latency.

Slansky reduces the delay at the expense of increased fan out. Kogge-Stone achieves high speed and low fan out but produces complex circuitry with more numbers of wiring tracks. The Knowles trees are family of network between between Kogge-Stone and Sklansky with increased fanout. Ladner Fischer introduced a network between Sklansky and Brent-Kung which provides tradeoffs between logic levels and fanout.

T. Han and D.A. Carlson presented a hybrid construction of a parallel prefix adder using two designs the Kogge-Stone construction having the best feature of higher speed and the Brent-kung construction with best feature of low area requirement. A modified Han-Carlson adder uses fewer number of prefix operations by adjusting the number of stages amongst Kogge-Stone and Brent-kung adder and thus reduces the area required by the adder circuitry.

Fig 3.below shows 3-dimentional taxonomy of tree adders. There are three axis representing the fan out, wiring tracks and logic levels and each tree is indicated by three integers (l, f, t) in the range [0, L-1]. The tree adders lie on the plane l + f + t = L-1, where L= log2N and indicates the number of bits. Brent-Kung, Kogge-Stone and Sklansky represent the vertices of the cube (3, 0, 0),(0, 0, 3) and (0, 3, 0) respectively. Han-Carlson, Ladner-Fischer and Knowles lie along the diagonals. Where N indicates the number of bits the variables l, f, and t are integers in the range [0, L−1] indicating:

- Logic Levels: L+ 1
- Fan-out: 2f+1
- Wiring Tracks: 2t

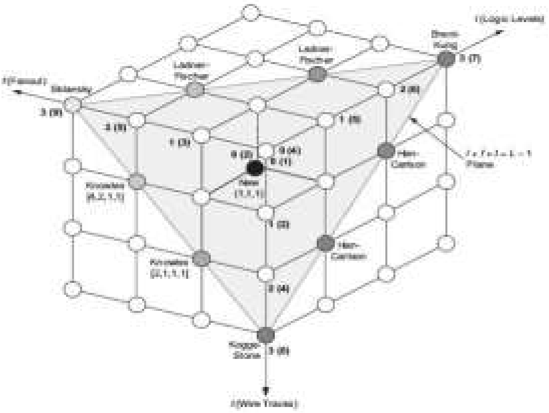

Fig 3. Taxonomy of prefix networks

2.1 PROPOSED WORK

To design Parallel Prefix Hybrid Han-Carlson Adder. It differs from other adder in that it can be used for large word sizes. The proposed design reduces the number of prefix operation by using more number of Brent-Kung stages and lesser number of Kogge-Stone stages. This also reduces the Complexity, silicon area and power consumption significantly.

3. IMPLEMENTATION

The designing of proposed adder architecture is done using Xilinx ISE 13.1 Tool and the complete source code for 32 bit Implementation of proposed adder is done. The design is implemented using Spartan 6 device. The basic elements of the design are modeled as components which are independently functional. These are then wired together by means of signals to construct the structure of the adder. The design is implemented using the Spartan 6 device.

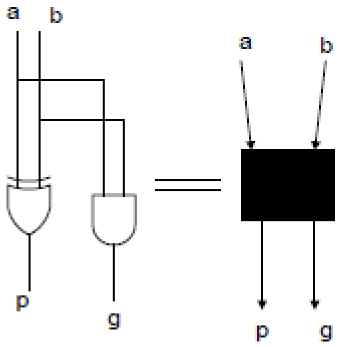

Fig.4 Square cell structure

Circular cells: for computation of prefix operation (gi. pi) o (gj. pj) = (gi+ pi . gj, pi . pj)

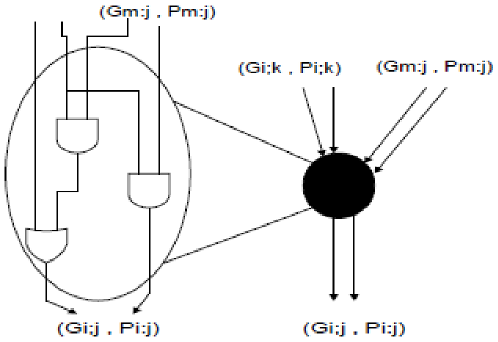

Fig. 5 Black cell

Studied the structure of Hybrid Han-Carlson Adder and various design parameters. Also studied different Prefix cells that are being used in the design and their equations are shown below Square Cells for pre-processing parallel prefix stage to calculate generates and propagates. g= a and b & p= a xor b

4. RESULT

FIG.6 Simulation result for 32-bit han-carlson adder

Han-Carlson adder presented a reduction in the complexity and hence provides a tradeoffs for the construction of large adders. These wide adders are useful in applications like cryptography for security purpose, global unique identifiers used as a identifier in computer software and this wide adder also provides good speed.

6.REFERENCES

[1] Darjn Esposito, Davide De Caro, Ettore Napoli, Nicola Petra, Antonio Giuseppe Maria Strollo "Variable Latency Speculative Han-Carlson Adder" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 62, NO. 5, MAY 2015

[2] R. Zimmermann, "Binary adder architectures for cell-based VLSI and their synthesis," Ph.D. thesis, Swiss Federal Institute of Technology, (ETH) Zurich, Zurich, Switzerland, 1998, Hartung-Gorre Verlag.

[3] R. P. Brent and H. T. Kung, "A regular layout for parallel adders,"IEEE Trans. Comput., vol. C-31, no. 3, pp. 260–264, Mar. 1982.

[4] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," IEEE Trans. Comput., vol. C-22, no. 8, pp. 786–793, Aug. 1973.

[5] J. Sklansky, "Conditional-sum addition logic," IRE Trans. Electron. Comput., vol. EC-9, pp. 226–231, Jun. 1960.

[6] T. Han and D. A. Carlson, "Fast area-efficient VLSI adders," in Proc. IEEE 8th Symp. Comput. Arith. (ARITH), May 18–21, 1987, pp. 49–56.

[7] R. E. Ladner and M. J. Fischer, "Parallel prefix computation," J. ACM, vol. 27, no. 4, pp. 831–838, Oct. 1980.

[8] S. Knowles, "A Family of Adders," in Proc. 14th IEEE Symp. Comput. Arith., Vail, CO, USA, Jun. 2001, pp. 277–281.

[9] S.-L. Lu, "Speeding up processing with approximation circuits," Computer, vol. 37, no. 3, pp. 67–73, Mar. 2004.

[10] T. Liu and S.-L. Lu, "Performance improvement with circuit-level speculation," in Proc. 33rd Annu. IEEE/ACM Int. Symp. Microarchit. (MICRO-33), 2000, pp. 348–355.

[11] N. Zhu, W.-L. Goh, and K.-S. Yeo, "An enhanced low-power highspeed Adder For Error-Tolerant application," in Proc. 2009 12th Int. Symp. Integr. Circuits (ISIC '09), Dec. 14–16, 2009, pp. 69–72.

[12] S. M. Nowick, "Design of a low-latency asynchronous adder using speculative completion," IEE Proc. Comput. Digit. Tech., vol. 143, no. 5, pp. 301–307, Sep. 1996.

[13] A. K. Verma, P. Brisk, and P. Ienne, "Variable Latency Speculative Addition: A New Paradigm for Arithmetic Circuit Design," in Proc. Design, Autom., Test Eur. (DATE '08), Mar. 2008, pp. 1250–1255.

[14] A. Cilardo, "A new speculative addition architecture suitable for two's complement operations," in Proc. Design, Autom., Test Eur. Conf. Exhib. (DATE '09), Apr. 2009, pp. 664–669.

[15] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," in Proc. Design, Autom., Test Eur. Conf. Exhib. (DATE '12), Mar. 2012, pp. 1257–1262.