IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 11, Issue 3, Ver. I (May-Jun .2016), PP 38-43
www.iosrjournals.org

Parallel Prefix Speculative Han Carlson Adder

Gayathri.G[1], Raju S.S[2], Suresh.S[3]
[1]*Master of engineering student,* [2]*Assitant Professor, 3 Associate Professor,*
[1,2,3] *Department of E.C.E, A.K.T Memorial college of Engineering and Technology, Kallakurichi, Tamilnadu, India*

Abstract: *Binary addition is one of the most important arithmetic function 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 fast and secondly efficient in terms of power consumption and chip area. Speculative variable latency adders have attracted strong interest thanks to their capability to reduce average delay compared to traditional architectures. This paper proposes a novel variable latency speculative adder based on Han-Carlson parallel-prefix topology that resulted more effective than variable latency Kogge-Stone topology. The paper describes the stages in which variable latency speculative prefix adders can be subdivided and presents a novel error detection network that reduces error probability compared to variable latency adder.*

Keywords: *Parallel Prefix Adder, Kogge Stone Adder, Han-Carlson Adder.*

I. 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 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 good results as compared to the conventional adder[1].

Parallel Prefix Adder includes Brent-Kung [2], Kogge-Stone [3]. The architecture operates at fixed latency. It proposes a novel variable latency speculative adder based on Han-Carlson parallel-prefix topology. The Han-Carlson topology uses one more stage than Kogge-Stone adder, while requiring a reduced number of cells and simplified wiring. Thus, it can achieve similar speed performance compared to Kogge-Stone adder, at lower power consumption and area. We show that a speculative carry tree can be obtained by pruning some intermediate levels of the classical Han-Carlson topology. It provides rigorous derivation of the error detection network and shows that the error detection network required in speculative Han-Carlson adders is significantly faster than the one used by speculative Kogge-Stone architecture.

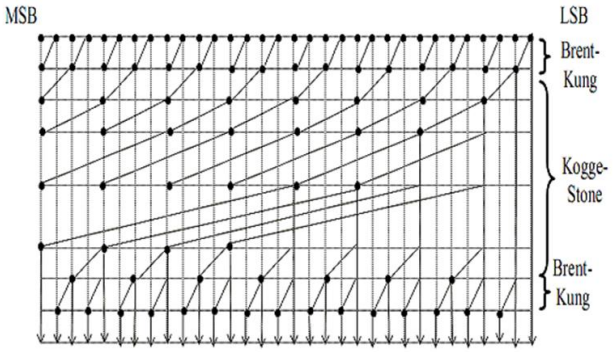
The main aim in ALU design is to reduce the adder critical path, which decides execution time in terms of delay(6.133ns) and power(35%). These two factors are most essential in adder design[4]. Parallel prefix adder design is most preferable for their higher speed of operation. There are different algorithms are used in process of addition. They main aim on improvising the performance of Parallel Prefix Adder (PPAs)by optimizing performance parameters such as Speed, Power, Area and number of gate counts. There are various topologies of prefix adders are there, they give the comparisons among the various parallel tree adders.

By using variable latency speculative adder, kogge stone speculative adder produces 45% of area is reduced and 35% of power is saved compared to kogge stone non- speculative adder. The kogge stone adder uses the minimum number of logic levels and has fan-out of 2[5,7].

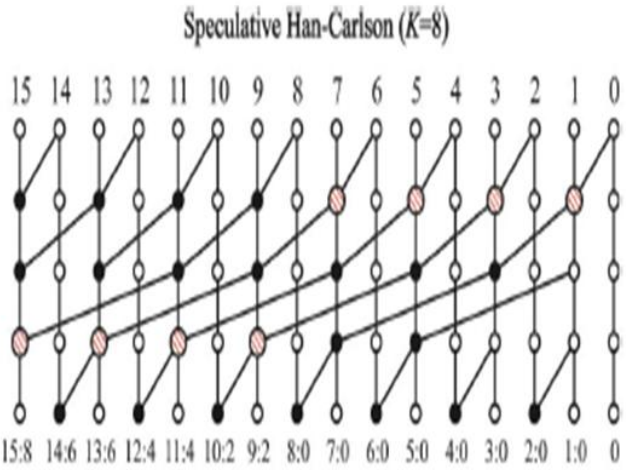
These tree structures validate the benefits of each one of them with the other by making use of the performance parameters[8]. The various tree structures are mentioned in the literature based on area(45%), fan-out-2 and complexity in circuit design. [9]In case of the Kogge-Stone, it uses recursive doubling property which leads to the fan-out limited to unity at the each stage of carry merge[10].

II. Han-Carlson Adder

The Han-Carlson adder is a blend of the Brent-Kung and Kogge-Stone adders. It uses one Brent-Kung stage at the beginning followed by Kogge-Stone stages, terminating with another Brent-Kung stage to compute the odd numbered prefixes. It provides better performance compared to Kogge-Stone for smaller adders.



GRAPICAL REPRESENTATION OF HAN CARLSON ADDER



HAN CARLSON ADDER CARRY LENGTH (K=16)

The Han-Carlson is the family of networks between Kogge-Stone and Brent-Kung. Han-Carlson adder can be viewed of Kogge-Stone adder. This adder is different from Kogge-Stone adder in the sense that these performs carry-merge operations on even bits and generate/propagate operation on odd bits. At the end, these odd bits recombine with even bits carry signals to produce the true carry bits.

This adder has five stages in which the middle three stages resembles with the Kogge-Stone structure. The advantage of the adder is that it uses much less cells and its shorter. Thus there is a reduction in complexity at the cost of an additional stage for carry-merge path.

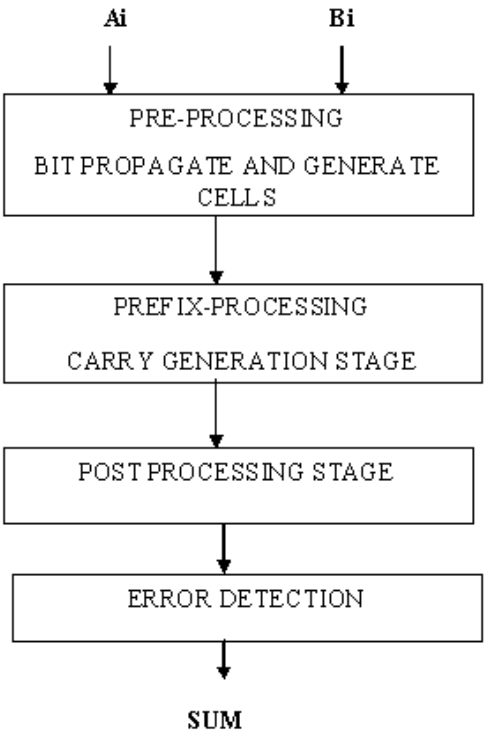
We have generated a Han- Carlson Speculative Prefix Processing stage by deleting the last rows of the kogge stone adder. This yields a speculative stage with k=8=n/2^p, where p is the number of pruned levels.

III. Parallel Prefix Adder

Parallel prefix adders are suitable for VLSI implementation since it differs from other adders, 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. Parallel Prefix Adder can be subdivided in the following stages: Pre-Processing, Post-Processing, Error Detection and Error Correction. The Error Correction Stage is Off the critical path, as it has two clock cycles to obtain the exact sum when speculation fails.

The Pre-Processing and Post-Processing Stages of a Prefix adder involve only simple operations on signals to each bit location. Hence, adder performs mainly on Prefix operation. Therefore black dots represent the prefix operator, while white dots represent simple place holders.

IV. Block Diagram

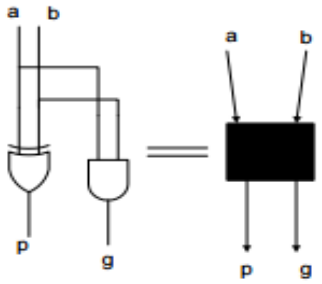


PRE PROCESSING

In the pre processing stage generate (Gi) and propagate (Pi) signals are calculated.

Gi= a and b &

Pi= a xor b



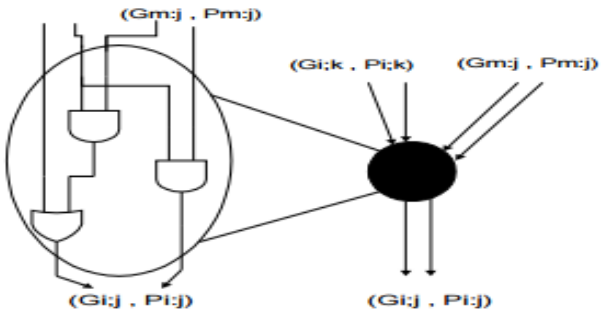
Square cell structure

SPECULATIVE PREFIX-PROCESSING

Instead of computing all the g(i:0)and p(i:0) required to obtain the exact carry values, only a subset of block generate and propagate signals is calculated; block generate and propagate signals is calculated;

Han-Carlson adder constitutes a good trade-off between fan out, number of logic levels and number of black cells. Because of this, Han-Carlson adder can

Achieve equal speed performance respect to Kogge-Stone adder, at lower power consumption and area. Therefore it is interesting to implement a speculative Han-Carlson adder.



SPECULATIVE PREFIX PROCESSING
Circular cells: for computation of prefix operation
Calculations of all carry signals:
Gi:j = Gi:k +Pi:k . Gk-1:j
Pi:j = Pi:k . Pk-1:j
1. Bit propagate and generate
This block implements the following logic:
Gi = Ai AND Bi
Pi = Ai XOR Bi
2. Group propagate and generate
This block implements the following logic:
G2 = G1 OR (G0 AND P1)
P2 = P1 AND P0

POST PROCESSING

In the post processing stage approximate carry values are obtained from this subset and then use them to obtain the approximate sum bits Si as follows: The approximate carries are already available at the output of the prefix-processing stage. The post-processing is equal to the one of a non-speculative adder and consists of xor gates.
Calculation of Final Sum:

$$S_i = P_i \oplus G_{i-1:0}$$

ERROR DETECTION

The error detection circuit that flags an error if the sum computed by the Almost Carry Adder is incorrect. This only occurs when there is a chain of more than *k* propagates in the addenda. To check for the presence of an error, we must consider all chains of length *k* + 1, and check if any of them contain solely propagates. The expression for error signal is stated as follows:

$$ER = \sum_{i=0}^{n-k-1} pipi+1....pi+k$$

The conditions in which at least one of the approximate carries is wrong (misprediction) are signaled by the error detection stage. In case of misprediction, an error signal is asserted by error detection stage and the output of the post-processing stage is discarded.

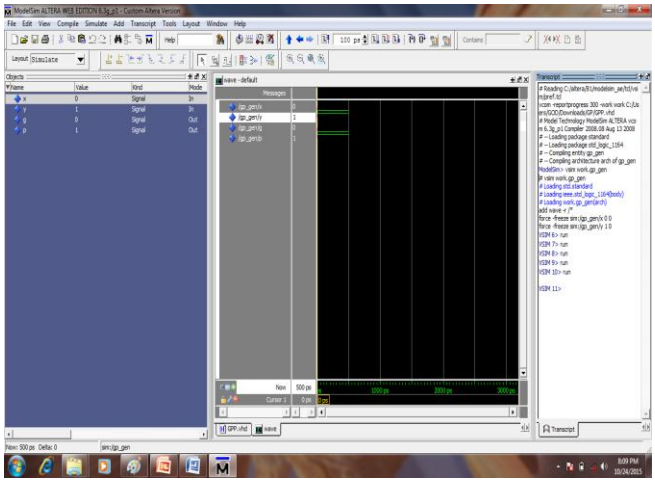
ERROR CORRECTION

This stage computes the exact carry signals to be used, incase of misprediction. It is composed by the levels of the prefix, processing stage pruned to obtain the speculative adder.

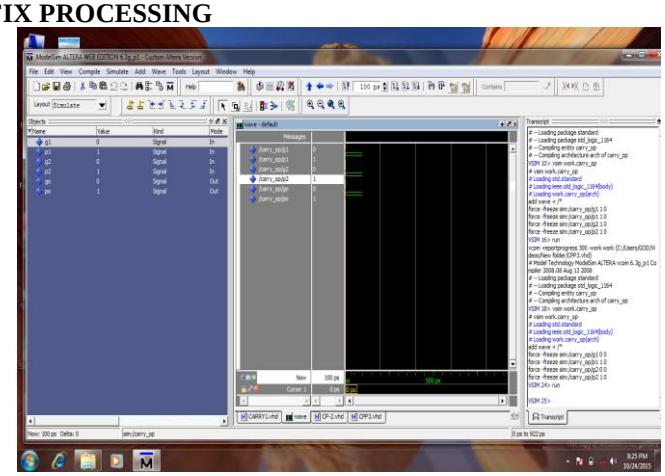
V. Results and Discussion

The proposed Parallel Prefix Adder can be analyzed using Xilinx. It reduces the minimum achievable delay. The analysis of Area and Power shows that speculative adders are not effective for large average delay. At timing constraint imposed during synthesis is made tighter speculative adders become advantageous.

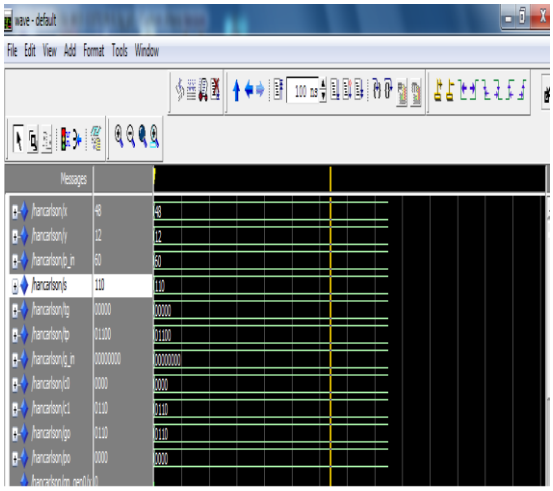
PRE-PROCESSING



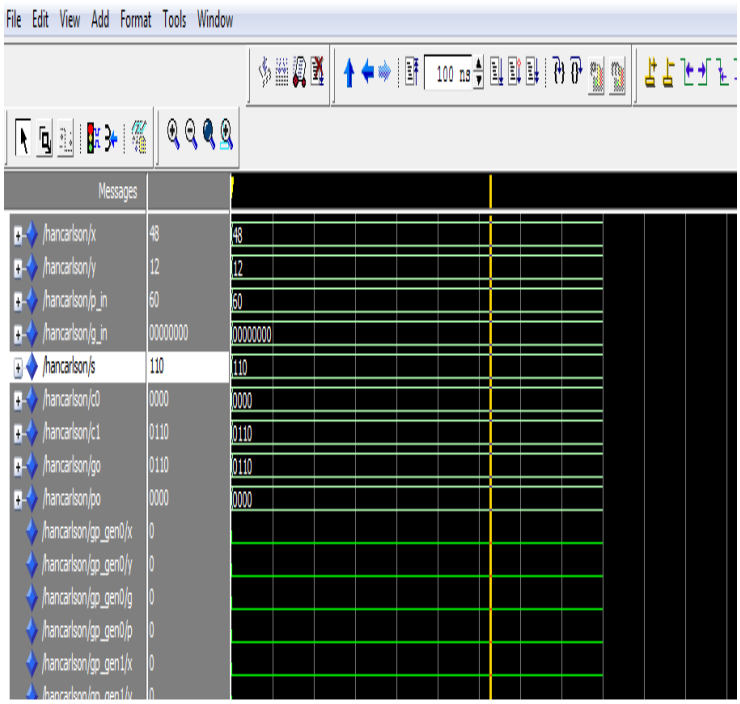
SPECULATIVE PREFIX PROCESSING



POST-PROCESSING



ERROR CORRECTION



VI. Conclusion

In this paper Han-Carlson adder presented 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. It is used to reduce error by using error correction and detection techniques. It is used to reduce number of prefix operation and reduces complexity.

References

[1]. Darjn Esposito, Davide De Caro, Senior Member,IEEE, Ettore Napoli, Nicola Petra, Member, IEEE and Antonio Giuseppe Maria Strollo, Senior Member, IEEE, " Variable Latency Speculative Han-Carlson Adder", May 2015.
[2]. R.P.Brent and H.T.Kung, ''A regular layout of parallel adders",IEEE Trans.Comput., Vol C-31, Mar 1982
[3]. 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, Aug 1973.
[4]. Cong Liu, Jie Han and Fabrizio Lombardi, "A Low Power, High Performance Approximate Multiplier with Configurable Partial Error Recovery,(DATE'14) Apr 2014.
[5]. 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.
[6]. 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.
[7]. A.K.Verma, P.Brisk, and P.Ienne, "Variable Latency Speculative Addition, A New Paradigm for Arithmetic circuit Design, Autom, Test Eur. (DATE'08), Mar 2008.
[8]. S-L.Lu,"Speeding up processing with Approximation Ciruits", Computer, Mar 2004.
[9]. I.Koren, "Computer Arithmetic Algorithms", Natick, MA, USA AK Peters, June 2002.
[10]. S.Knowles, "A Family of Adders", in Proc, 14th IEEE symp. Comput.Arith, Vail, CO, USA, June 2001.
[11]. S.K.Mathew, R.K.Krishnamoothy, M.A.Anders, R.Rios, K.R.Mistry, and K.Soumyanath, "'Sub-500-ps 64-b ALUs, Design and scaling trends, "IEEE J.Solid-State Circuits, Nov 2001.
[12]. Tong Liu and Shih-Lien Lu, Intel Corporation," Performance Improvement with Circuit-Level Speculation", May 2000.