International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 7
July 2017

Academic Science

Differential Latency Hypothetical Han Carlson Adder

Suvarna Dilip Shekade
Department of Electronics and Telecommunication
S.V.C.E.T Pune, India
Prof. Manoj Kumar
Department of Electronics and Telecommunication
S.V.C.E.T Pune, India
Prof. Balramudu P
Department of Electronics and Telecommunication
S.V.C.E.T Pune, India

ABSTRACT— *In this paper we have proposed hypothetical Han-Carlson adder. The proposed adder employs hypothecation: the exact arithmetic function is replaced with an approximated one that is faster and gives the correct result most of the time, but not for all time. The approximated adder is augmented with an error detection network that asserts an error signal when speculation fails. The hypothetical adder to reduce delay and power consumption compared to non-hypothetical adders. The paper describe the stages in which differential latency hypothetical prefix adder can be subdivided and presents a novel error detection network that reduces error probability compared to previous approaches*

KEYWORDS—Addition, digital arithmetic, parallel-prefix adders, hypothetical adders, hypothetical functional units, variable latency adders.

I. INTRODUCTION

Adders are basic functional units in computer arithmetic. Binary adders are used in microprocessor for addition and subtraction operations as well as for floating point multiplication and division. Therefore adders are fundamental components and improving their performance is one of the major challenges in digital designs. Theoretical research [1] has established lower bounds on area and delay of n-bit adders: the former varies linearly with adder size, the latter has a $O(log_2(n))$ behavior.

In this paper we propose a novel differential latency hypothetical adder based on Han-Carlson [3] 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 [8]. We show that a hypothetical carry tree can be obtained by pruning some intermediate levels of the classical Han-Carlson topology. The paper presents a rigorous derivation of the error detection network and shows that the error detection network required in hypothetical Han-Carlson adders is significantly faster than the one used by speculative Kogge - Stone architecture. An extensive set of implementation results for 65 nm CMOS technology shows that proposed Han-Carlson differential latency adders outperform previously developed differential latency Kogge-Stone architectures.

II. DIFFERNTIAL LATENCY HYPOTHETICAL PREFIXADDERS

Variable latency speculative prefix adders can be subdivided in five stages: pre-processing, hypotheticalprefix-processing,post-processing, error detection and error correction. Theerrorcorrectionstageisoffthecriticalpath,asithastwoclockcycles to obtain the exact sum when speculationfails.

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 7
July 2017

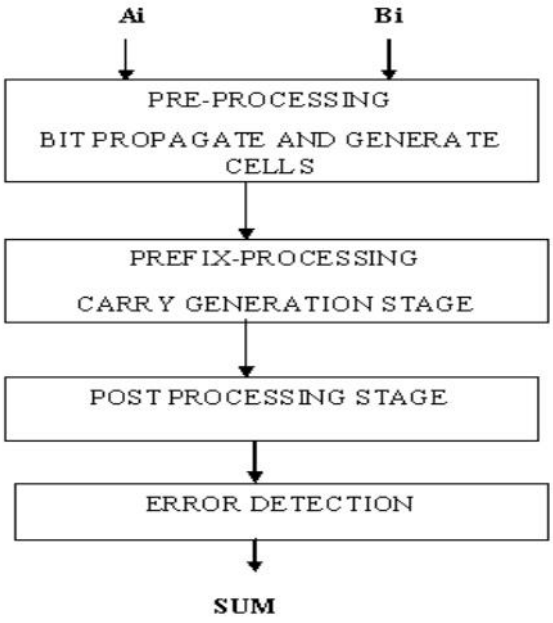

Fig. 1.Block Diagram

A. PRE-PROCESSING

In the pre-processing stage the generate g_i and propagate p_i signals are computed.

Gi= a and b &

Pi= a xor b

B. HYPOTHETICAL PREFIX-PROCESSING

The hypothetical prefix-processing stage is one of the main differences compared with the standard prefix adders recalled in previous section. Instead of computing all the g[i:0] and p[i:0] required in (8) to obtain the exact carry values, only a subset of block generate and propagate signals is calculated; in the post-processing stage approximate carry values are obtained from this subset. The output of the hypothetical prefix-processing stage will also be used in the error detection and in the error correction stages discussed in the following.

The basic assumption behind hypothetical prefix-processing stage is that carry signals propagate for no more than K bits, with K<n and K=0($\log_2$(n)).This assumption is corroborated by the analyses in [5],[6] that demonstrate that having a propagate chain longer that $\log_2$(n) is a very rare event.

1) KOGGE-STONE TOPOLOGY:

The Kogge-Stone speculative prefix-processing stage has been proposed in [4],[5] and can be obtained by pruning the last levels of a traditional Kogge-Stone adder. In the example shown in Fig. 2, the last level of a n=16 bit Kogge-Stone adder is pruned. As it can be observed, for i≥8 the length of propagate chains extends for 8 bits, resulting in a speculative prefix-processing stage with K=8. .

In general, one has $K=n/2^P$, where P is the number of pruned levels; the number of levels of the speculative stage is correspondingly reduced from $\log_2$(n) to $\log_2$(K) (assuming that K is a power of two). In general, the computed propagate and generate signals for the speculative Kogge-Stone architecture are:

$(g,p)_{[i:0]}$ for: i≤ K-1

$(g,p)_{[i:i-k+1]}$ otherwise (1)

2) HAN-CARLSON TOPOLOGY:

Han-Carlson adder constitutes a good trade-off between fanout, number of logic levels and number of black

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 7
July 2017

cells. Because of this, Han-Carlson adder can achieve equal speed performance respect to Kogge-Stone adder, at lower power consumption and area [8]. Therefore it is interesting to implement a hypothetical Han-Carlson adder. Moved by these reasons, we have generated a Han-Carlson hypothetical prefix-processing stage by deleting the last rows of the Kogge-Stone part of the adder. As an example, the Fig. 3 shows the Han-Carlson adder of Fig. 1 in which the two Brent-Kung rows at the beginning and at the end of the graph are unchanged, while the last Kogge-Stone row is pruned. This yields a speculative stage with K=8=n/2. In general, one has $K=n/2^P$, where P is the number of pruned levels: the number of levels of hypothetical Han-Carlson stage reduces from $1+\log_2(n) + \log_2(K)$ (assuming that K is a power of two).

As it can be observed in Fig. 2, the length of the propagate chains is K=8 only for i = 9,11,13,15, while for i = 10, 12, 14 the propagate chain length is K+1=9 .

In general, the computed propagate and generate signals for the hypothetical Han-Carlson architecture are:

$(g,p)_{[i:0]}$for: $i \le k$

$(g,p)_{[i:i-K+1]}$for: $i > k$, i odd

$(g, p)_{[i: i-K]}$for: $i > k$, i even (2)

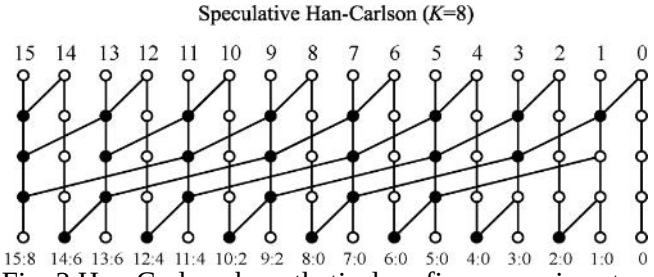

Fig. 2.Han-Carlson hypothetical prefix-processing stage.

C. POST-PROCESSING

In the post-processing stage we firstly compute the approximate carries, c_i, and then use them to obtain the approximate sum bits as follows:

$S_i = p_i + c_{i-1}$

$$\tilde{c}_i = \begin{cases} g_{[i:0]} & \text{for: } i \le K \\ g_{[i:i-K+1]} & \text{for: } i > K,\ i \text{ odd} \quad (\text{Han} - \text{Carlson}) \\ g_{[i:i-K]} & \text{for: } i > K,\ i \text{ even} \end{cases} \quad (3)$$

D. ERROR DETECTION

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. The error correction stage will give the correct sum in the next clock period.

1) Kogge-Stone: The error condition for carry can be obtained from (2),(3) and using the properties of propagate generate signals as:

$e_i = 0$ for: $i \le K-1$

$p[i:i-K+1]g[i-K:0]$ otherwise (4)

Thus, the error signal can be expressed as:

outputs are needed to compute the error signal, are named “checking nodes” and are highlighted as big hatched dots, for the topologies in Fig. 3.

$$E_{ks} = \sum_{n-1} p[i:i-k+1]g[i-k:0] \quad (5)$$

i=K

Where the ∑ represents the logical OR.

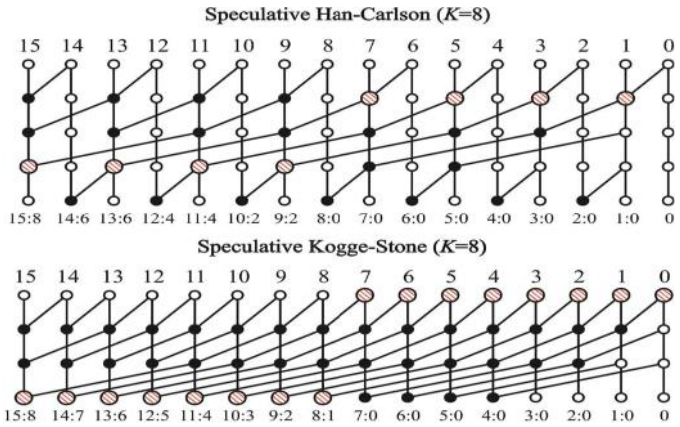

Fig. 3. The nodes of the prefix-processing stage

it is important to note that above equation is a necessary and sufficient error condition that requires the calculation of g[i-k:0].unfortunately, these terms are actually *not* computed by the speculative prefix-processing stage (avoiding the computation of these terms is the key idea of speculative adders). thus, in previous papers,(5) is replaced by the following looser relation:

n-1

E_{ks}=∑p[i:i-k+1] (6)

i=K

The last equation is a necessary-only error condition. By using above, the error signal can be triggered even in absence of actual misprediction. While this does not harm the correct operation of the hypothetical adder, having an high rate of such "false positive" errors degrades the average addition time (1).In this paper, instead, we rewrite the necessary and sufficient condition 1 in a form that does not require the g[i-K:0] terms. To that purpose, let us consider the last two terms of the OR in (1), with index n-1 and n-2:

finally one obtained

$$E_{KS} = \sum_{i=K}^{n-1} p_{[i:i-K+1]} g_{i-K} \quad (7)$$

2) Han-Carlson: The error condition for carry c_i canbe obtained from c_i and eq(3).

$$e_i = \begin{cases} 0 & \text{for: } i \le K \\ p_{[i:i-K+1]} g_{[i-K:0]} & i > K,\ i \text{ odd} \\ p_{[i:i-K]} g_{[i-K-1:0]} & i > K,\ i \text{ even} \end{cases} \quad (8)$$

The error signal can be written

$$E_{HC} = \sum_{\substack{i=K+1 \\ i \text{ odd}}}^{n-1} p_{[i:i-K+1]} g_{[i-K:0]} + \sum_{\substack{i=K+1 \\ i \text{ even}}}^{n-1} p_{[i:i-K]} g_{[i-K-1:0]} \quad (9)$$

It can easily be seen that in above the terms in theSecond OR are implied by the terms in the first OR.Let us

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 7
July 2017

Academic Science

consider, for instances, the first two terms of the OR assume that K is evenWe have:

$$p_{[K+1:2]}g_{[1:0]} + p_{[K+2:2]}g_{[1:0]} = p_{[K+1:2]}g_{[1:0]} \quad (10)$$

Thus we can write

$$E_{HC} = \sum_{\substack{i=K+1 \\ i \text{ odd}}}^{n-1} p_{[i:i-K+1]}g_{[i-K:0]} \quad (11)$$

Similar simplifications can be realized by considering in above eq the terms n-3 and n-5 and so on. Finally one obtains

$$E_{HC} = \sum_{\substack{i=K+1 \\ i \text{ odd}}}^{n-1} p_{[i:i-K+1]}g_{[i-K:i-K-1]} \quad (12)$$

From the above observations, it can be concluded that error detection is sensibly simplified and potentially faster in Han-Carlson. Compared to Kogge stone.

E. ERROR CORRECTION

The error correction stage computes the exact carry signals ci = g[i,0], to be used in case of misprediction.The error correction stage is composed by the levels of the prefix-processing stage pruned to obtain the hypothetical adder.

F. POST-PROCESSING

The approximate carries are already available at the output of the prefix-processing stage. The post-processing, according to (12), is equal to the one of a non-hypothetical adder and consists of xor gates.

III. SIMULATION RESULTS

To investigate the advantages of using our technique in terms of area overhead against “Fully ECC” and against the partially protection, we implemented and synthesized for a Xilinx XC3S500E different versions of a32-bit, 32-entry, dual read ports, single write port register file.

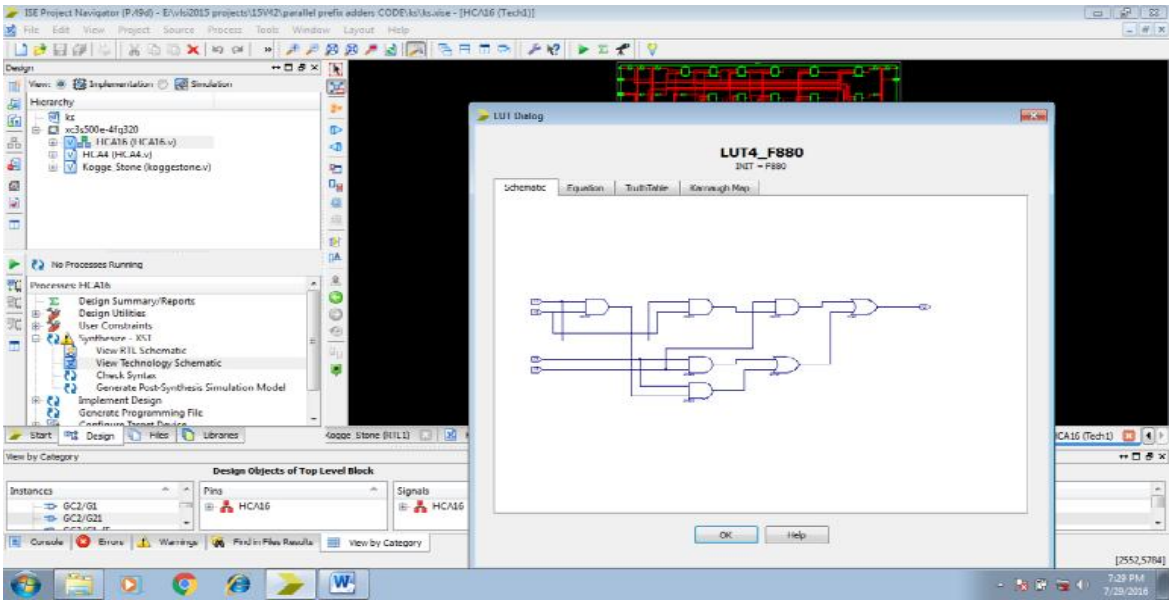

Fig 5: Internal block Han-Carlson Adder

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 7
July 2017

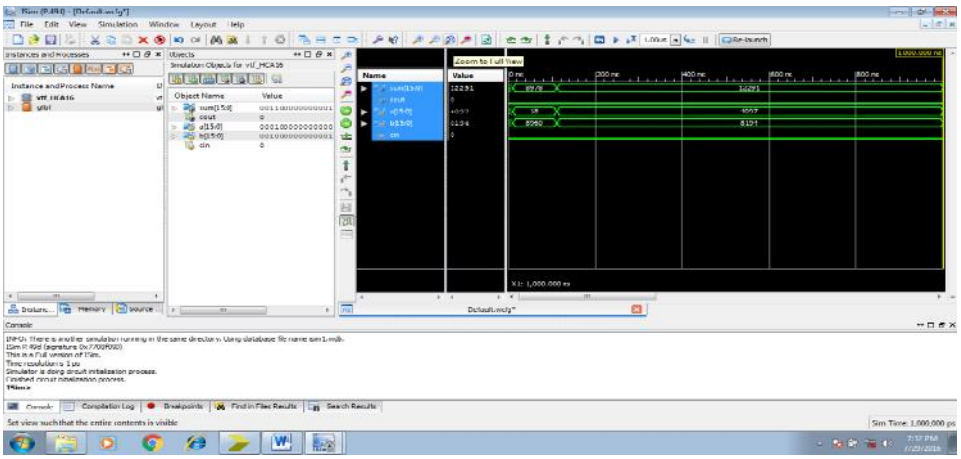

Fig 6: Simulated output for Han-Carlson Adder

IV. CONCLUSION

From the above work, we conclude that the Han-Carlson adder presented a reduction in the complexity and hence provides 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.

REFERENCES

[1] I.Koren, *Computer Arithmetic Algorithms*. Natick, MA, USA: A K Peters, 2002 [

2] R. Zimmermann, "*Binary adder architectures for cell-based VLSI andtheir synthesis,*" Ph.D. thesis Swiss Federal Institute of Technology, (ETH) Zurich, Zurich, Switzerland.

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

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

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

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

[7]K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition."

[8] S. K. Mathew, R. K. Krishnamurthy, M. A. Anders, R. Rios, K. R.Mistry, and K. Soumyanath, "Sub-500- ps 64-b ALUs in 0.18- m SOI/ bulk CMOS: Design and scaling trends," *IEEE J. Solid-State Circuits*,vo 36.