© 2017 IJSRST | Volume 3 | Issue 7 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology

Area Efficient Speculative Han-Carlson Adder

A. Dhanunjaya Reddy
PG scholar, JNTUA College of Engineering, Anantapuramu, Andhra Pradesh, India

ABSTRACT

Parallel prefix adders are used to compute the result with fixed delay and high speed. Better results can be produced by using speculation in these adders. Speculation is nothing but an approximation which can increase microprocessor clock frequency by modifying complete logic function with simplified one that speculates and observe the estimated outputs. This paper introduces a new variable delay speculative han-carlson adder which is combination of Brent-kung and Kogge-stone topologies that gives better performances compared to variable delay kogge-stone adder. Speculative han-carlson introduces error detection network which reduces the error occurences when speculation fails compared to former approaches. Both kogge-stone and han-carlson speculative adders are synthesized in the Xilinx tool which tells that speculative han-carlson adders have less area and more speed compared to speculative kogge-stone and kogge-stone respectively. And non-speculative adders give the best outputs when delay constraint is relaxed.

Keywords : Addition, parallel prefix adders, error detection network speculative adders, variable delay adders, digital arithmetic.

I. INTRODUCTION

Adders are fundamental and basic functional units in computer arithmetics and improving adders performance is one of the big challenge in designing digital circuits. Binary adders are used in microprocessor for addition and subtraction operations as well as for floating point operations as multiplication and division. Theoritical research has implemented [1] on area and delay of n-bit adders: the previous ones varies linearly with adder size, the present ones are varies with O(log2(n)) behavior. Parallel prefix adders performs the n-bit addition with high speed [1]. The carry generating path in parallel prefix adder works on the basic principle of carry look ahead adder mechanism.

Parallel prefix standard architectures are Brent-Kung [3], Kogge-Stone [4], Sklansky [5], Han-Carlson [6], Ladner-Fischer [7] and Knowles [8].These standard architecures performs with fixed delay. By getting variable delay we can get the average performances. These variable delay adders [9] are proposed in recent literature. A variable delay adder performs [10] speculation i.e.,exact arithmetic function is modified by approximated function which gives high speed results correctly most of the time, but in some cases it not. The approximated adder [15] is placed with an error detection network that gives errored output if speculation fails. In such cases of misprediction, we need one more clock cycle to get the correct output in addition of a correction stage. So, the addition time is one clock cycle if no error occurs otherwise it needs one more clock cycle when speculation is wrong. The average delay time Tavg can be calculated as

$$Tavg = PErr \cdot 2 \cdot Tclk + (1 - PErr) \cdot Tclk = Tclk\,(1 + PErr) \quad (1)$$

Where Tclk is clock period and PErr is probability of error of speculative adder.This paper proposes a new variable delay [12] speculative adders of han-carlson topologies. The han-carlson has one stage more than kogge-stone but it has less number of black cells and reduces wiring. speculative han-carlson introduces error detection network which reduces error probability compare to [13] older approaches. Both kogge-stone and han-carlson speculative adders are synthesized in the Xilinx tool which tells that speculative han-carlson adders have less area and more speed compared to speculative kogge-stone and kogge-stone respectively. And non-speculative adders give the best outputs when delay constraint is relaxed.

The paper is organized as follows. In section II the basic structure of parallel prefix adders is [13] reviewed. In section III variable delay speculative prefix adders can be divided in to five stges after a sample review of kogge-stone and han-carlson speculative [11] topologies. Complete overview of error detection network is reviewed[14] in this section.In section IV detailed observation and synthesized results of the implementing adders, for n-bit size ranges of 16,32 and 64 bits. Section V concludes paper with improvements for further approach to present speculative adders with other architectures.

II. PRELIMINARIES

A. Prefix Addition

The binary addition as follows : Two n-bit inputs augend and addend $A=a_{n-1}a_{n-2}\ldots a_0$, $B=b_{n-1}\, b_{n-2}\ldots b_0$ and outputs sum $S=s_{n-1}s_{n-2}\ldots s_0$ and c_i is the carry output of i^{th} bit respectively. The outputs sum (s_i) and carry (c_i) can be calculated as follows:

$$S_i = a_i \oplus b_i \oplus c_{i-1} \qquad (2)$$

$$C_i = a_i b_i + b_i c_{i-1} + c_{i-1} a_i \qquad (3)$$

To do the prefix addition we need three steps to calculate the sum: pre, prefix and post processings.

Firstly in pre processing step calculate the generation g_i and propagation terms p_i can be calculated.

$$g_i = a_i b_i \qquad (4)$$

$$p_i = a_i \oplus b_i \qquad (5)$$

That is $g_i=1$ means carry is generated at bit i, and $p_i=1$ means carry is propagated through bit i. the method of generate and propagate signals are collected to group of adjacent bits from bits k to i with $k \le i$, as follows:

$$g_{[i:k]} = \begin{cases} g_i & \text{if } i = k \\ g_{[i:j]} + p_{[i:j]}\, g_{[i:k]} & \text{else} \end{cases} \qquad (6)$$

$$p_{[i:k]} = \begin{cases} p_i & \text{if } i=k \\ p_{[i:j]}p_{[l:k]} & \text{else} \end{cases} \qquad (7)$$

where: $i \ge l \ge j \ge k$

That is $g_{[i:k]} = 1$ means carry is generated at bit k-i ,and $p_{[i:k]} = 1$ means carry is propagated through k-i .and carry calculated as :

$$C_i = g_{[i:0]} + p_{[i:0]}c_{-1} \qquad (8)$$

Where c_{-1} is the initial carry, generally we take $c_{-1} = 0$, then equation (8) will be as :

$$C_i = g_{[i:0]} \qquad (9)$$

The group generate (GG) and group propagate (PG) terms are calculated in the second step i.e., at the prefix processing of the adder to calculate the carry of n-bit. In this $(g_{[i:k]},p_{[i:k]})$ terms will be denoted with the need of prefix operator ● as follows:

$$(g_{[i:k]},p_{[i:k]}) = (g_{[i:j]},p_{[i:j]}) \bullet (g_{[l:k]},p_{[l:k]}) =$$

$$(g_{[i:j]} + p_{[i:j]}g_{[l:k]}, p_{[i:j]}p_{[l:k]}) \qquad (10)$$

Where: $i \ge l \ge j \ge k$

The prefix operator ● is useful in the prefix stage which is obeying associative and idempotent properties are useful to speed up the calculation.

Finally in the last step of post processing output of n-bit sum can be calculated using (8) as follows:

$$S_i = p_i \oplus c_{i-1} \qquad (11)$$

B.Han-Carlson and Kogge-Stone Parallel Prefix Adder Topologies

The 1st and 3rd steps of prefix adder of pre and post processing stages had simple operations on each bit positions. Prefix processing step decides the performance of adder.

Fig.1 shows Kogge-Stone prefix adder topology which has $\log_2(n)$ stages.it has a fan-out of two at each stage.it has more number of black dots which means more propagation and generation cells leads to more wired tracks.

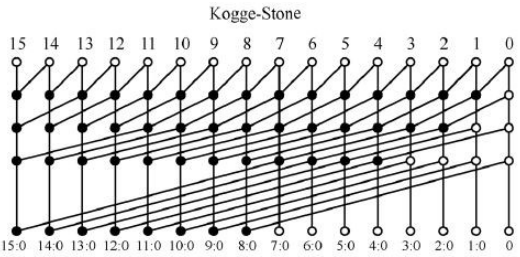

Figure 1. Kogge-Stone parallel prefix topology for n=16.

Han-Carlson adder is combination of Brent- Kung and Kogge-Stone having advantages of low area and high speed respectively.it is also called as hybrid adder, because it is mixing of two parallel prefix adders. The outer stages of han-carlson are brent-kung [3] stages at the begining and at the end of the prefix graph .

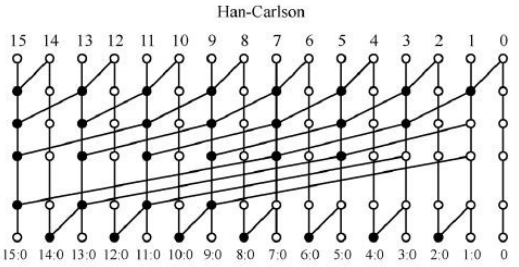

Figure 2. Han-Carlson parallel prefix topology for n=16

A good balance between number of black dots, logic levels, and fan-out is produced in the han-carlson adder. The number of stages in han-carlson is $1 + \log_2 (n)$.

III. VARIABLE DELAY SPECULATIVE PREFIX ADDERS

Speculation is the new concept introduced in standard parallel prefix adders to produce the minimum achievable average delay. speculation is nothing but an exact arithmetic function is replaced with an approximated function by cutting the stages of a standard prefix adder. Variable delay adder is achieved by using speculation.

Variable delay speculative adders are divided in to five steps: the first three steps are pre-processing, speculative prefix-processing and post-processing as per the same standard parallel prefix adder calculation but change is speculative prefix processing. And the last two steps are error detection and error correction. suppose When error occurs one more clock cycle is needed to get the correct sum output if speculation fails.

A. Pre-Processing

In this step the generate (g_i) and propagate (P_i) terms are calculate same as from equations (4) and (5).

B. Speculative Prefix-Processing

This step is one which is the major difference when compared to standard prefix adders in earlier section. In this stage only a subset of group generate and group propagate signals is computed instead of calculating all the generation $g_{[i:0]}$ and $p_{[i:0]}$ terms required in equation (8) to get the exact carry values. In the 3rd step that is at post-processing stage approximated output carry's are obtained from this stage. The results of this stage are used in next 4th and 5th steps of error detection and in the error correction stages will be discussed as follows.

The basic hypothesis behind this stage is carry signals does n't propagate for more than K bits, where $K < n$ and $K = O(\log_2(n))$. This assumption will be proved by analysis in [13] ,[17] that shows clearly having a propagating length is more than $\log_2(n)$ is very less chances.

Kogge-Stone topology:

The Speculative Kogge-Stone prefix processing stage has been introduced in [12],[13] and we can get it by removing the last stages of basic Kogge-Stone adder. if we observe in Fig.3, the last stage is of n=16 bit kogge-stone adder is removed. For $i \geq 8$, the propagate chain length extends up to 8 bits only, results in to speculative prefix processing stage with $K = 8$, where $K = n/2^P$. p denotes number of cutting stages or levels. For speculative Kogge-Stone the number of stages will be reduced from $\log_2(n)$ to $\log_2(K)$ that is stages are reduced from 4 to 3.(assuming K in terms of power of 2).the calculated propagate and generate signals for speculative Kogge-Stone are as follows:

$$\begin{array}{ll} (g,p)_{[i:0]} & i \leq K\text{-}1 \\ (g,p)_{[i:i-K+1]} & \text{else} \end{array} \qquad (12)$$

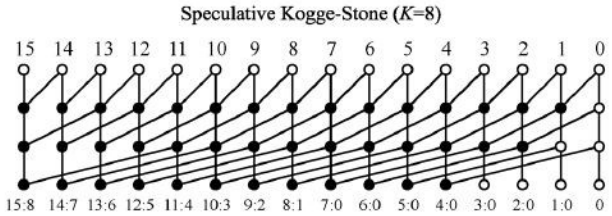

Figure 3. Kogge-Stone speculative prefix-processing stage.The last row of n=16 bit Kogge-Stone adder is pruned, results in a speculative prefix-processing stage with K=8.

Han-Carlson Topology:

Han-Carlson adder has a good balance among fan-out, number of black dots and stages. So, it can achieve equal speed performance to Kogge-Stone adder at low power consumption and area [16]. The first and last stages are Brent-Kung and remaining 3 stages are Kogge-Stone with shorter wire span. we have generated a speculative prefix processing stage of Han-Carlson by deleting the last stage of Kogge-Stone present in Han-Carlson adder shown in Fig.4 where as the first and last stages of Brent-Kung remains same. This changes to a speculative stage with K = 8 = (16/2) for 16 bit . one has $K = n/2^P$, wher p denoted number of removing stages; the stages of speculative Han-Carlson reduces from $1+\log_2(n)$ to $1+\log_2(k)$ i.e., from 5 to 4.

If we observe in Fig.4 the propagation chain length is K = 8 only for i = 9,11,13,15, where as for i = 10,12,14 the propagation chain length is K+1 = 9.

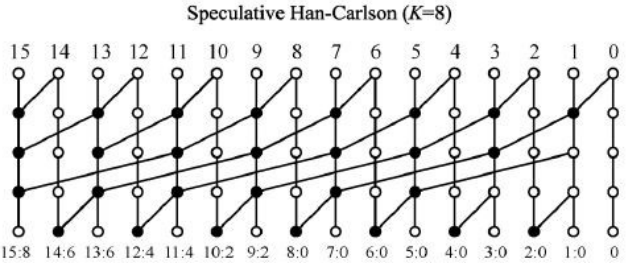

Figure 4. Han-Carlson speculative prefix-processing stage.The last Kogge-Stone row of n=16 bit Han-Carlson adder is pruned, results in a speculative prefix-processing stage with K = 8.

The calculated propagate and generate signals for Speculative Han-Carlson adder 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 (13)

It will be clearly understood from above that the propagating chain length equals to K + 1 for half of the outputs are easier to do the error detection.

C. Post Processing

Here firstly we need to calculate the approximated carries c_i, and use that carries to get the approximated sum ouput S_i as follows:

$$S_i = p_i \oplus c_{i-1} \quad (14)$$

Like equation (9) ,the approximate carries will be calculated as generate signals available in the last stage of speculative prefix processing stage.
We have:

$$\tilde{c_i} = \begin{matrix} g_{[i:0]} & \text{for : } i \le K\text{-}1 \\ g_{[i:i-K+1]} & \text{else (Kogge-Stone)} \end{matrix} \quad (15)$$

$$\tilde{c_i} = \begin{matrix} g_{[i:0]} & \text{for : } i \le K \\ g_{[i:i-K+1]} & \text{for : } i > K, \text{ i odd} \\ g_{[i:i-K]} & \text{for : } i > K \text{ ,i even} \\ & \text{(han-carlson)} \end{matrix} \quad (16)$$

C. Error Detection

In case the computations in which at least one of the approximate carry is incorrect will be signaled in this stage. In chance of misprediction error signal is accepted by this error detection stage and the output of the post-processing step is removed. After that one more clock cycle is needed in the form of error correction stage to get the exact sum.

i) Kogge-Stone: To calculate carry error condition will be derived from (9) ,(15)and also use the properties of propagation and generation bits as:

$$e_i = \begin{matrix} 0 & \text{for : } i \le K\text{-}1 \\ p_{[i:i-K+1]} g_{[i-K:0]} & \text{else} \end{matrix} \quad (17)$$

The mathematical form of error condition will be expressed a

$$_{EKS} = \sum_{i=K}^{n-1} p_{[i:i-K+1]}\, g_{[i-K:0]} \quad (18)$$

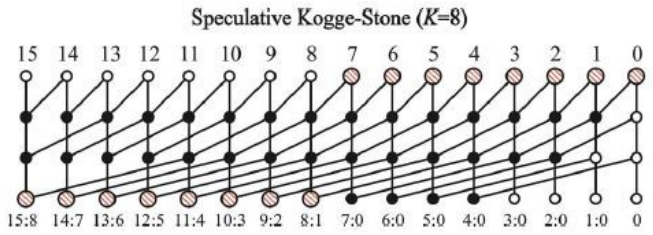

Figure 5. The nodes of the prefix-processing stage, whose outputs are needed to Calculate the error signal, are named "checking nodes" and are highlighted as big hatched dots, for the topologies in Fig.3

Where the above summation symbol denotes logical OR. It is necessary to mention that equation (18) is needed and sufficient condition for error calculation requires $g_{[i-k:0]}$. These terms are not calculated by speculative prefix processing step. (This is the main idea of speculative adders to neglect the computation of these terms.) Thus in previous papers equation (18) is modified by:

$$E_{KS} = \sum_{i=K}^{n-1} p_{[i:i-K+1]} \qquad (19)$$

The above equation is error condition.by using equation (19), the error signal can be found also even in absence of actual misprediction. while this does not effect the correct operation which is speculation having high chances of false positive errors which reduces average addition time (1).we consider the last two terms in equation (18) of OR combination with index n-1 and n-2:

$$P_{[n-1:n-K]}g_{[n-K-1:0]} + p_{[n-2:n-K-1]}g_{[n-K-2:0]} \qquad (20)$$

From basic formula

$$g_{[n-K-1:0]} = g_{n-K—1} + p_{n-K-1}g_{[n-K-2:0]} \qquad (21)$$

on substituting equation (21) in (20) then equation (18) can be simplified as follows:

$$p_{[n-1:n-K]}g_{n-K-1} + p_{[n-2:n-K-1]}g_{[n-K-2:0]} \qquad (22)$$

same simplification can be realized by using (18) the terms n-2 and n-3 etc., finally we can get

$$E_{KS} = \sum_{i=K}^{n-1} p_{[i:i-K+1]}\, g_{i-K} \qquad (23)$$

The error signal (23) is re-written as

$$E_{KS} = p_{[8:1]}g_0 + p_{[9:2]}g_1 + p_{[10:3]}g_2 + \ldots\ldots + p_{[15:8]}g_7 \qquad (24)$$

ii)Han-Carlson: To calculate the carry c_i the error condition will be derived from (9) ,(16) as:

$$e_i = \begin{cases} 0 & \text{for : } i \le K \\ p_{[i:i-K+1]}g_{[i-K:0]} & \text{for : } i > K, \text{ i odd} \\ p_{[i:i-K]}g_{[i-K-1:0]} & \text{for : } i > K, \text{ i even} \end{cases} \qquad (25)$$

the error signal can be denoted as

$$E_{HC} = \sum_{\substack{i=K+1 \\ i\ odd}}^{n-1} p_{[i:i-K+1]}\, g_{[i-K:0]} + \sum_{\substack{i=K+1 \\ i\ even}}^{n-1} p_{[i:i-K]} \qquad (26)$$

It can be clearly seen that in (26) the terms in second OR are implied by the terms in first OR. Let us assume K is even. Then the first two terms are

$$P_{[K+1:2]}g_{[1:0]} + p_{[K+2:2]}g_{[1:0]} = p_{[K+1:2]}g_{[1:0]} \qquad (27)$$

Then we can mention as:

$$E_{HC} = \sum_{\substack{i=K+1 \\ i\ odd}}^{n-1} p_{[i:i-K+1]}\, g_{[i-K:0]} \qquad (28)$$

The above equation can be simplified by an approach which is similar to previous section. let us see the last two terms of OR in (28) with index n-1 and n-3 of (28), assume n is even:

$$P_{[n-1:n-K]}g_{[n-1-K:0]} + p_{[n-3:n-2-K]}g_{[n-3-K:0]} \qquad (29)$$

One has:

$$g_{[n-1-K:0]} = g_{[n-1-K:n-2-K]} + p_{[n-1-K:n-2-K]}g_{[n-3-K:0]} \qquad (30)$$

Substitute the equation (30) in (29) ,the equation can be return with terms n-1 and n-3 then equation (28) can be simplified as:

$$p_{[n-1:n-K]}g_{[n-K-1:n-K-2]} + p_{[n-3:n-K-2]}g_{[n-K-3:0]} \qquad (31)$$

similar calculations can be done by taking equation (28),the terms n-3 and n-5 etc., finally error obtained will be:

$$E_{HC} = \sum_{\substack{i=K+1 \\ i\ odd}}^{n-1} p_{[i:i-K+1]}\, g_{[i-K:i-K-1]} \qquad (32)$$

The error signal (32) can be re-written as

$$E_{HC} = p_{[9:2]}g_{[1:0]} + p_{[11:4]}g_{[3:2]} + \ldots + p_{[15:8]}g_{[7:6]} \quad (33)$$

By comparing equations (23) and (32),it can be seen that number of terms to be OR-ed to to get the error signal is half of the Han-Carlson when compared with Kogge-Stone.

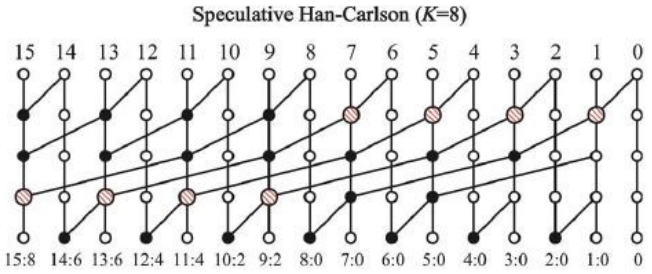

Figure 6. The nodes of the prefix-processing stage, whose outputs are needed to Calculate the error signal, are named "checking nodes" and are highlighted as big hatched dots, for the topologies in Fig.4.

The nodes of the prefix stage which are highlighted as big hatched dots are checking nodes for both kogge-stone and han-carlson of fig.3 and 4 are shown in fig.5 and 6 respectively. whose outputs are needed to calculate error signal.

As it can be seen in kogge-stone some of the checking nodes are present at last stage of graph. But in han-carlson the checking nodes are present in before last level i.e. second last level of graph. And both kogge-stone and han-carlson checking nodes will be after three black cells delay.

From above all, we observe that in han-carlson error detection is easily simplifird and more faster when compared to kogge-stone. The need of error detection stage leads to increase in fan-out of cheching nodes which slow downs the operation in speculative prefix stage.

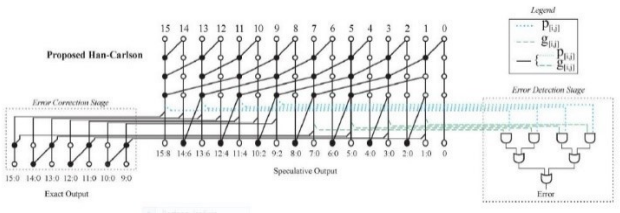

Figure 7. Error correction and detection stages for the proposed speculative Han-Carlson adder of Fig.4.

E .Error Correction

The error correction stage calculates the exact carry signals in equation (9) which are used in case of misprediction.

The error correction stage is developed by adding the stages which are removed in prefix stage of speculative adder.

Fig.7 shows the error correction stage of proposed speculative han-carlson adder , it can be mentioned that the addition of error correction stage increases the fanout of some of cells of speculative prefix stage, with adverse effect on adder speed.

F. Post-Processing

The approximate carries which are already calculated are available at the output of prefix processing stage. from equation (14) it is equal to one of the non-speculative adder and it consists of n xor gates.

IV. SYNTHESIS RESULTS

We have developed the vhdl code for kogge-stone and han-carlson for proposed speculative han-carlson adder and non-speculative adder for 16-bit. Then we simulate the code in Modelsim 6.3f for generating outputs sum and carry, and the propagation generation signals will be shown for each stage in the simulation window after giving the inputs A and B of 16- bits and give initial carry as zero (0). For speculative adders if speculation fails we are using error correction and detection is used in the proposed speculative han-carlson method. If speculation fails then by using error detection and correction method we get the exact outputs for speculative han-carlson. For this one in error detection stage speculation works error value is zero(0) otherwise it shows the error value one(1) if speculation fails, then it can correct the output by adding the stages which we remove in speculation.

The biggest advantage of han-carlson is the error probability will be less when compared to kogge-stone, this can be clearly understood by comparing the error equations of (23) speculative kogge-stone with (32) speculative han-carlson. The or terms which are used to

calculate error signal is halved in han-carlson when compared to kogge-stone.

We are also comparing area delay and power for all adders in Xilinx ISE 8.1i for 16- bit design. By using two 16 bit adders, four 16 bit adders the parameters can be compared for 32 and 64 bit adders respectively. Here we are comparing the non-speculative kogge-stone with speculative kogge-stone and non-speculative han-carlson with speculative han-carlson, It provides better results for han-carlson to kogge-stone for both speculative and non-speculative cases.

G. The optimal K choice

The variable delay prefix adders depends on the choice of parameter K where $K = n/2^P$,where P is number of removing stages or rows of parallel prefix stage. the optimum K value will goes down from following trade-off: on increasing K value we reduce error probability (with positive effects on average delay(1))and error detection will be slower .

Comparison between variable delay adder and non-speculative han-carlson shows that variable delay adders allow to reduce the minimum achievable delay.

The analysis of area in terms of gate count and power dissipation shows that speculative adders are not effective for large average delay. As timing constraint imposed during synthesis is made tighter speculative adders become advantageous. When compared han-carlson uses gate count 450 and speculative han-carlson uses 264 gate count which is (41% reduction) for 16 bit adders. and delay for han-carlson is 16.572 ns where as for speculative han-carlson delay is 12.32 ns which is (25% reduction) and power is 116 mw for han-carlson and 96 mw for speculative han-carlson (17 % reduction).

Table-1

	PERFORMANCE PARAMETERS (16-bit)		
	AREA (gate count)	DELAY (ns)	POWER (mw)
1.KOGGE-STONE	576	21.79	129
2.HAN-CARLSON	450	16.572	116

Table-2

	PERFORMANCE PARAMETERS (32-bit)		
	AREA (gate count)	DELAY (ns)	POWER (mw)
1.KOGGE-STONE	660	21.844	164
2.HAN-CARLSON	588	16.779	151

Table-3

	PERFORMANCE PARAMETERS (64-bit)		
	AREA (gate count)	DELAY (ns)	POWER (mw)
1.KOGGE-STONE	1320	22.937	288
2.HAN-CARLSON	1200	17.818	270

Table-4

	PERFORMANCE PARAMETERS (16-bit)		
	AREA (gate count)	DELAY (ns)	POWER (mw)
1.SPECULATIVE KOGGE-STONE	426	14.903	112
2.SPECULATIVE HAN-CARLSON	264	12.321	91

Table-5

	PERFORMANCE PARAMETERS (32-bit)		
	AREA (gate count)	DELAY (ns)	POWER (mw)
1.SPECULATIVE KOGGE-STONE	612	14.993	148
2.SPECULATIVE HAN-CARLSON	540	13.387	147

Table-6

	PERFORMANCE PARAMETERS (64-bit)		
	AREA (gate count)	DELAY (ns)	POWER (mw)
1.SPECULATIVE KOGGE-STONE	1104	15.083	257
2.SPECULATIVE HAN-CARLSON	960	13.540	253

Tables 1,2 and 3 shows the kogge-stone and han-carlson results of area, delay and power respectively, and **Tabels** 4,5 and 6 shows the speculative results of kogge-stone and han-carlson results of area, delay and power respectively.

Table 7 shows the results of han-carlson error detection and correction performances of area,delay and power for 16,32 and 64 bits respectively

Table-7

		AREA(gate count)			DELAY(nano seconds)			POWER(milli watts)	
	16-BIT	32-BIT	64-BIT	16-BIT	32-BIT	64-BIT	16-BIT	32-BIT	64-BIT
HAN-CARLSON ERROR DETECTION AND CORRECTION	318	648	1296	13.927	16.032	16.032	96	161	290

CONCLUSION

In this paper new approach of variable delay han-carlson parallel prefix adder is introduced by using speculation technique and it is used in high speed applications. Error detection network is introduced for more accuracy which assures and allows reduces the probability of error occurences compared to previous techniques.

The han-carlson variable delay adders performance is better than the kogge-stone variable delay adders. Compared with former non-speculative adders, our analysis demonstrates that variable delay han-carlson adder gives high speed when it is required.

Additional work is needed to extend speculative approach to other parallel-prefix architectures, such as Brent-Kung, Ladner-Fischer, and Knowles.

V. 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, 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 efficientsolution 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]. N. Zhu, W.-L. Goh, and K.S. Yeo, "An enhanced low-power high speed Adder For Error-Tolerant application," in Proc. 2009 12th Int.Symp. Integr. Circuits (ISIC '09), Dec. 14-16, 2009, pp. 69-72.
[11]. 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.
[12]. 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.
[13]. B. Parhami, Computer Arithmetic: Algorithms and Hardware Design.New York: Oxford Univ. Press, 2000
[14]. A. Tyagi, "A reduced-area scheme for carry-select adders,"IEEETrans.Comput., vol. 42, no. 10, pp. 1163-1170, Oct. 1993.
[15]. 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.

About Author:

Mr. A. Dhanunjaya Reddy completed B.Tech in ECE from Sri Sai Institute of Science and Technology Engineering College, Rayachoty in 2013. Now He is pursuing M.Tech in JNTUA college of engineering Anantapuramu. His areas of interests are Vlsi Design and Digital Electronics.