A Novel Approach For Error Detection And Correction Using Prefix-Adders

B. Naga Jyothi*[1], K.S.N.Murthy[2], K.Srinivasarao[3]

[*1]PG Student Department of ECE, K.L. University Green fields-522502, AP, India
bondalapati.nagajyothi@gmail.com

[2] Professor Department of ECE, K.L. University Green fields-522502, AP, India
ksnmurty@kluniversity.in

[3]Professor Department of ECE, K.L. University Green fields-522502, AP, India
drksrao@kluniversity.in

Abstract - The variable latency speculative Han-Carlson adder is a newly proposed adder to perform high speed arithmetic operations. Han-Carlson adder gives accurate results with error detection when compared to other adders like Kogge-Stone adder. In this paper, number of parallel prefix adders can be sub divided into number of stages and perform arithmetic operations. By using the Xilinx 14.2 software, the design of Kogge-Stone adder and Han-Carlson adder is developed. This paper focuses on the implementation and simulation of 8-bit, 16-bit Kogge-stone adder and Han- Carlson adder based on Verilog code and compared for their performance in Xilinx. When compared to other adders the delay performance for Han Carlson adder is less and it reduces the complexity. It is concluded that the proposed adder is better in terms of computational delay. By using Brent –Kung and Kogge-stone adder the parallel prefix Han-Carlson adder also be proposed.

Keywords – Addition, Variable latency adders, Han-Carlson adder, Kogge stone adder, Xilinx, delay.

I. Introduction

In general, adders are used for various types of number –crunching operations like addition, subtraction, division and floating point multiplication. Adders of fast depend on their parallel prefix designs [2]. Adders are used for improving performance. Kogge-Stone, Brent-Kung , Han-Carlson , Sklansky , Knowles ,Ladner-Fischer , are different types of high speed adders. The Han Carlson adder is the further implementation of Kogge-Stone adder which is used to detect the errors. A first hypothetical approach to manage extension was proposed by Nowick [12] in non-simultaneous test, which completes a variable latency adder cutting the most minimal levels of a Kogge-Stone adder where as a adder in perspective of theoretical procedure is proposed by Verma [13] working in the same model as of Nowick. Later a variable latency carry select adder is developed in which the adder is divided into number of parts such that each part is working based on the principle of Kogge-Stone adder.

The Kogge-stone viper uses least number of logic levels and uses most extreme number of propagate and generate cells. It uses more number of wires routed together to connect each one of the logic levels. When compared to other adders Han-Carlson adder uses more number of logic levels. Both Kogge-stone adder and Han-Carlson adder requires same speed. Less number of wires is required to connect the logic levels in Han Carlson adder. This adder is used to reduce the complexity and delay.

There are three stages in performing prefix computation:

1. The pre-processing stage is used to calculate, generate and propagate bit.
2. The carry computation stage is used to compute the carry bit.
3. To compute the sum the post-processing stage is used.

The figure 1 shows the Han- Carlson adder for 16-bit. There are 2 stages in the figure 1, they are error correction and error detection. Gates are used in the error detection stage to increase the fan out for each cell. However there exists some theoretical lower bounds on the area and delay of different n- bit adders in which the area is varied in linear to the size of the adder and the delay is of the order $O(\log_2(n))$ [1].There exists some critical path in traditional adder which is rarely used as an observation to construct the speculative adders [9]-[13].But the output of these type of speculative adders depending on the bits of previous stages. This may be k and is of the order $O(\log_2(n))$ [12]-[15].

II. Block diagram

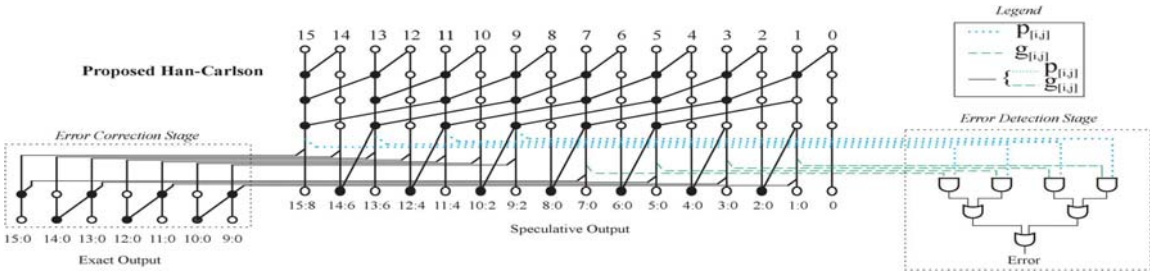

Fig1: Error detection and correction stages for proposed Han-carlson adder

III. Kogge-stone adder

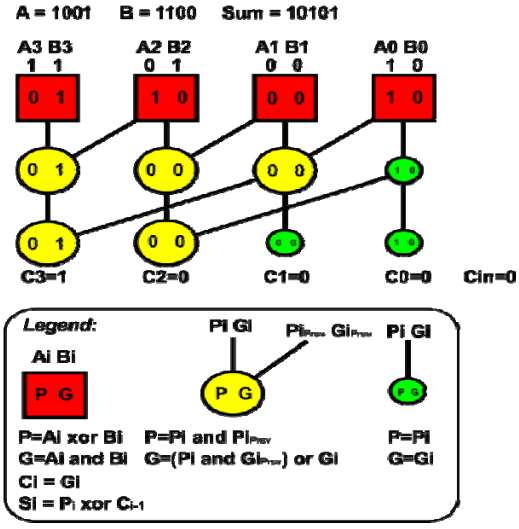

Fig 2: Example for 4 bit Kogge-stone adder

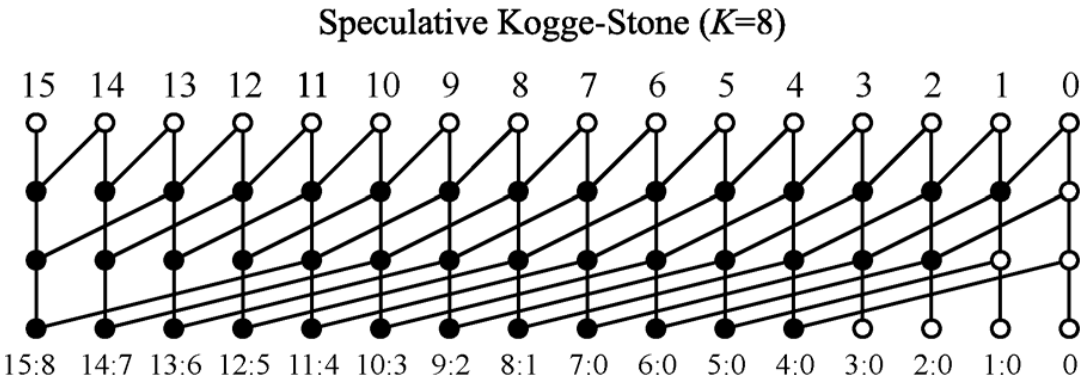

Fig 3: The 16-bit Kogge-stone adder with k=8 by pruning the last row of a n=16

The figure 3 shows the Kogge-stone adder for 16-bit. The Kogge-stone adder has high performance when compared to other adders. Using propagate and generate bits the Kogge-stone adder generates carries to each and every bit. Kogge- stone adder takes more area to implement when compared to other adders. For each level the fan-out is 2. Wiring blockage is frequently an issue for Kogge–Stone adders. The general issue of advancing parallel prefix adders is indistinguishable to the variable piece size, multi level, improvement issue. In this each vertical stage produces propagate and generate bits. The carry bit is generated at the last stage. In the first stage the propagate values are generated by XOR'ing the two input values and the generate bit is generated by AND'ing the two input values. After that the initial values are appeared same till the end of operation. The second sets of values are obtained by XOR'ing and AND'ing the initial value with the previous values.

IV. Han-Carlson adder

Han Carlson adder equals Kogge stone adder in terms of speed even at low power levels. In the middle of number of logic levels, fan out and number of dark cells the trade off existing. Han Carlson adder involves less area contrasted with Kogge stone adder. Han Carlson adder involves less area contrasted with Kogge stone adder. Hence it is preferred compared to Kogge stone adder. In this paper it is proposed that a Han Carlson

adder added as a prefix stage by removing the Kogge stone adder in the last stage. The proposed system uses Brent-Kung adder at two stages without Kogge stone adder. The levels eliminated are of the number k=n/2^p, where p is the propagation bit. For the 16-bit Han Carlson adder, in the first stage Brent-kung operation is used. In the second and third stages Kogge-stone adder operation and in the last stage Brent-kung adder operation is used. As a result the sum bits are obtained.

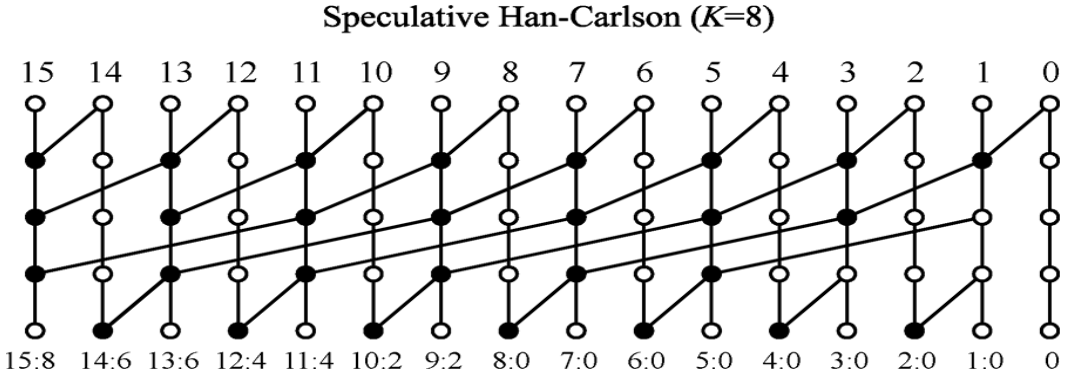

Fig 3: The 16-bit Han-carlson adder with k=8 by removing the last row of Kogge-stone adder

V. Results

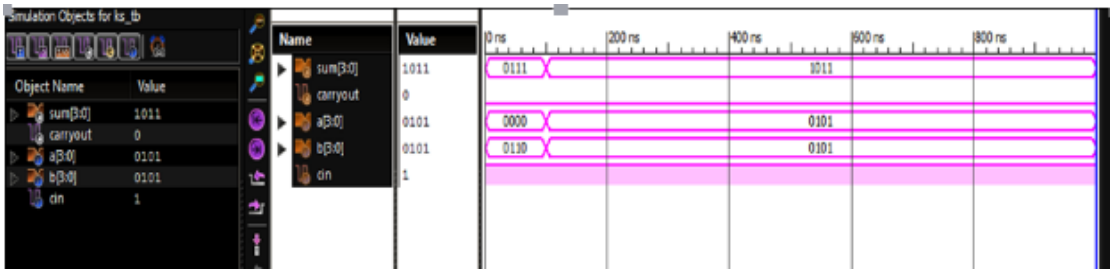

Fig 4: 8 bit Kogge Stone adder

The 8-bit Kogge-stone adder which is used in the last stages of the proposed adder.

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	4	5888	0%
Number of 4 input LUTs	7	11776	0%
Number of bonded IOBs	14	372	3%

Fig 5: Device Utilization Summary for 8-bit Kogge Stone Adder

This adder performs same as the proposed system. It has maximum combinational path delay of 9.297ns.The device utilization summary indicates the available resources and the resources utilized by the 8 bit Kogge Stone adder.

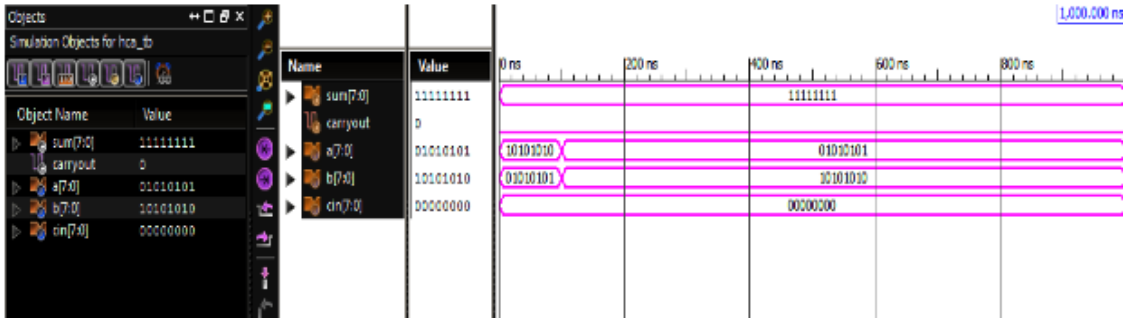

Fig 6: 8 bit Han Carlson adder

The Han Carlson adder is verified for 8 bit addition and the simulation results are shown in Fig 8.It also shows the sum and carry bits that are obtained in the result. It has a combinational delay of 11.914ns.The device utilization summary for 8 bit Han Carlson adder is as shown in fig 7

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	9	5888	0%
Number of 4 input LUTs	17	11776	0%
Number of bonded IOBs	26	372	6%

Fig 7: Device utilization summary for 8- bit Han Carlson adder

The Han Carlson adder for 16bit addition is done using Kogge stone adder and Brent-Kung adder .The simulation results for the proposed system using two adders is done and it is shown in Fig 8.

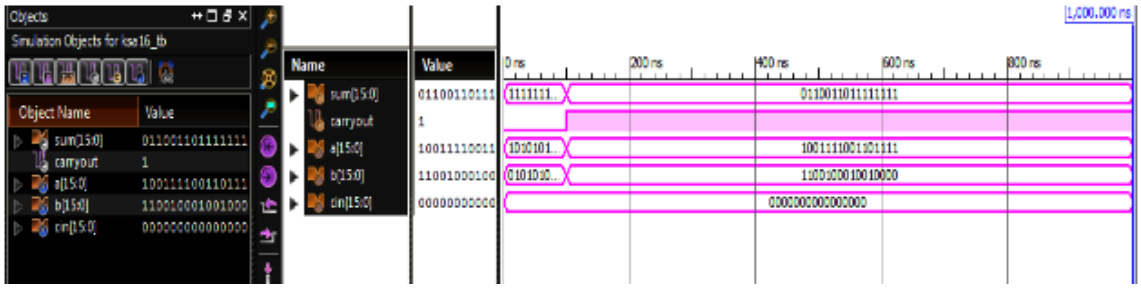

Fig 8: Simulation result for16- bit Han Carlson adder

The proposed adder with error detection and correction is shown in fig 8. It has a combinational delay of 13.088ns.

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	43	5888	0%
Number of 4 input LUTs	74	11776	0%
Number of bonded IOBs	50	372	13%

Fig 9: Device utilization summary for 16 bit Han Carlson adder

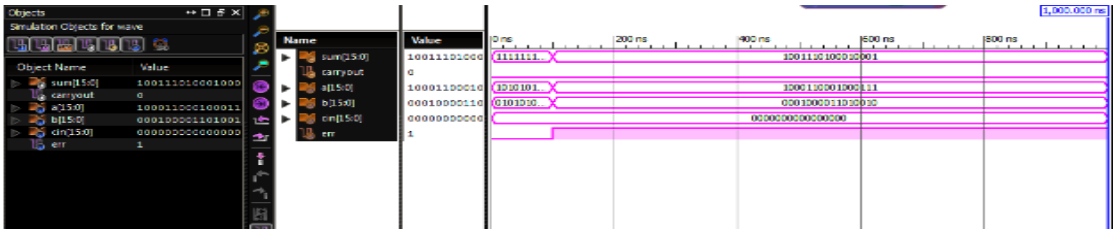

Fig 9: Simulation result for 16 bit Han Carlson adder with error detection and correction

The proposed adder has a device utilization summary as shown in fig 10 with a maximum combinational path delay of 15.044ns.

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	57	63400	0%
Number of fully used LUT-FF pairs	0	57	0%
Number of bonded IOBs	51	210	24%

Fig 10: Device utilization summary for 16 bit Han-Carlson adder with error detection and correction

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	8	9,312	1%
Number of occupied slices	6	4,656	1%
Number of slices containing only related logic	6	6	100%
Number of slices containing unrelated logic	0	6	0%
Number of bonded IOBs	14	232	6%
Average fanout of non-clock nets	1.71		

Fig 11: Device Utilization Summary Of Ripple Carry Adder

Logic Utilization	Used	Available	Utilization
Number of slices	6	4,656	0%
Number of 4 input LUTs	10	9,312	0%
Number of bonded IOBs	14	232	6%

Fig 12: Device Utilization Summary Of carry bypass Adder

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	9	9,312	1%
Number of occupied slices	6	4,656	1%
Number of slices containing only related logic	6	6	100%
Number of slices containing unrelated logic	0	6	0%
Number of bonded IOBs	14	232	6%
Average fanout of non-clock nets	2.38		

Fig 13: Device utilization Summary Of Carry Select Adder

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	8	9,312	1%
Number of occupied slices	6	4,656	1%
Number of slices containing only related logic	6	6	100%
Number of slices containing unrelated logic	0	6	0%
Number of bonded IOBs	14	232	6%
Average fanout of non-clock nets	1.71		

Fig 14: Device Utilization Summary Of carry save Adder

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	10	9,312	1%
Number of occupied slices	6	4,656	1%
Number of slices containing only related logic	6	6	100%
Number of slices containing unrelated logic	0	6	0%
Number of bonded IOBs	14	232	6%
Average fanout of non-clock nets	2.29		

Fig 15: Device Utilization Summary Of carry look ahead Adder

Name of the adder	Delay
Ripple carry adder	9.926ns
Carry bypass adder	9.882ns
Carry select adder	9.069ns
Carry save adder	9.882ns
Carry look ahead adder	8.736ns

Fig 16: comparison of adders in terms of delay

Multiplier name	Delay
Conventional multiplier	15.339ns
Vedic multiplier	15.246ns
Modified booth multiplier	7.204ns

Fig 17: comparison of multipliers in terms of delay

Conclusion

This paper mainly proposes a 16- bit Han Carlson adder using Brent-Kung adder and Kogge-Stone adder. In addition, the adder is designed with error detection and correction. The proposed adder has an improved performance compared to the traditional adders. The proposed adder is designed in Xilinx using Verilog HDL. The same procedure may be implemented to 32- bit and 64- bit adders also.

References

[1] I. Koren, Computer Arithmetic Algorithms. Natick, MA, USA: A K Peters, 2002.
[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), , pp. 49–56, May 18–21, 1987.
[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, pp. 277–281,Jun. 2001.
[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), pp. 348–355,2000.
[11] 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), pp. 69–72,Dec. 14–16, 2009.
[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), pp. 664–669, Apr. 2009.

Author Profile

Bondalapati Nagajyothi is pursuing M. Tech VLSI Design in K L University. Her research interests include FPGA Implementation, Low Power VLSI and Testing for VLSI Circuits.

K.S.N.Murthy is working as professor in K L University. His areas of research interest are in MEMS, Electronic Instrumentation.

K.Srinivasarao is working as professor in K L University. His areas of research interest are in MEMS.