I J C T A, 9(16), 2016, pp. 7843-7849

High Speed Han Carlson Adder Using Modified SQRT CSLA

D. Vamshi Krishna*, P. Radhika and T. Vigneswaran*****

ABSTRACT

Binary addition is the most important math functions in recent digital VLSI systems. Adders are majorly used as DSP filter where the sqrt adder is replaced in the place of ripple carry adder to minimize the calculation time. The man requirement of the adder is fast and efficient at power consumption and also chip area. To improve the speed of addition, parallel prefix adder is the best adder among the other existing techniques. Parallel prefix adders gives a good performance to make a wide range of design tradeoffs in terms of area, delay and power, this technique is more suitable for the adders with wider word lengths. In this paper, the technique which is designed is to reduce the complexity of the adder design and also to minimize the delay. The code is been synthesized using the Xilinx Integrated Software Environment (ISE) 13.2 Design Suite.

Index Terms: Parallel Prefix Adder, variable latency adders, speculative units, addition, RCA, sqrt adder.

1. INTRODUCTION

The binary addition is the basic math operation in digital circuits and it became extremely important in most of the digital systems including Math and Logic Unit (ALU), microprocessors and Digital Signal Processing (DSP).At present, the research continues by increasing the adder's delay performance. In many practical applications like mobile, telecommunications and others, the speed and power performance improved in FPGAs is better than microprocessor and DSP's based solutions. Additionally, power is also an important area in growing trend of mobile electronics, which makes large-scale use of DSP functions. Because of the programmability, the structure of configurable logic blocks (CLB) and programming interconnects in FPGAs, Parallel prefix adders have better performance. The delays of the adders are discussed. In electronics aspect, an adder is a digital circuit that performs addition of *n* numbers, in many computers and other kinds of processors, adders are not only used in the arithmetical logic units, but also in other parts of the processor where they are used to calculate addresses, table indices, and other kind of similar operations. Though adders can be constructed for many numerical representations, such as binary-coded decimal or excess-3, the most common adders operate on binary numbers only. In cases where 2's complement or 1's complement is being used to represent negative numbers, it is unnecessary to modify an adder into an adder or subractor. Other signed number representations require a more better and complicated adders. As to the VLSI, Parallel prefix adders (PPA) have the better delay performance. This paper investigates four types of PPA's (Kogge Stone Adder (KSA), Spanning Tree Adder (STA), Brent Kung Adder (BKA) and Sparse Kogge Stone Adder (SKA)). Additionally Ripple Carry Adder (RCA), Carry Look-ahead Adder (CLA) and Carry Skip Adder (CSA) are also investigated. These adders are implemented in the Verilog Hardware Description Language (VHDL) using Xilinx Integrated Software Environment (ISE) 13.2 Design Suite and are also synthesized. These designs are implemented in Xilinx Field Programmable Gate Arrays (FPGA) and delays are measured using logic analyzer and all these adder's delay, power and area are

* M.Tech VLSI Design, SRM University, *Email: dumpalakrishna@gmail.com*

** Assistant Professor, SRM University,

*** Professor, VIT University.

verified and compared finally. The adders in the papers are replaced with sqrt adders which plays important role in the reduction of the delay. The exact arithmetic operation of the adder is replaced with the speculative one and the adder is been replaced with sqrt adders to produce better results in the case of delay.

2. EXISTING METHOD

The existing adder uses parallel prefix adders and speculates the output, where the exact output is replaced with the speculating one. This adder speculates the output instead of giving exact output. This has been considered the fastest when compared with the previous adders. This adder comes with an error which has to be detected and then corrected. A network has been designed for assessment of the output with an error, at this point one more clock cycle is used to obtain the error less output. The addition time is considered one clock cycle when there is no error and the clock cycle is considered to be two when the speculation fails and error is present, the average time Tavg is been calculated as

$$T_{avg} = P_{err}.2.T_{clk} + (1-P_{err}).T_{clk} = T_{clk}(1 + P_{err}) \quad (1)$$

Where, Tclk is clock period and Perr is error probability of the adder. This adder is main based on the critical path. The output of the unit depends upon all the previous bits, so the output of the adder depends upon all the n input bits of the adder.

A first based on guessing approach to addition was suggested by Nowick in (not happening at the same time) contest, which puts into use a changeable delay something that adds cutting the lowest levels of a Kogge Stone adder. In (two or more things happening at the same time) contest, Verma propose a changeable delay based on some guessing adders; here the based on guessing addition is done in the same way as, cutting the lower levels of a Kogge-Stone, something that adds. In a changeable delay carry-select something that adds is introduced, where the adder is broken-up in different windows, each one contains a Kogge-

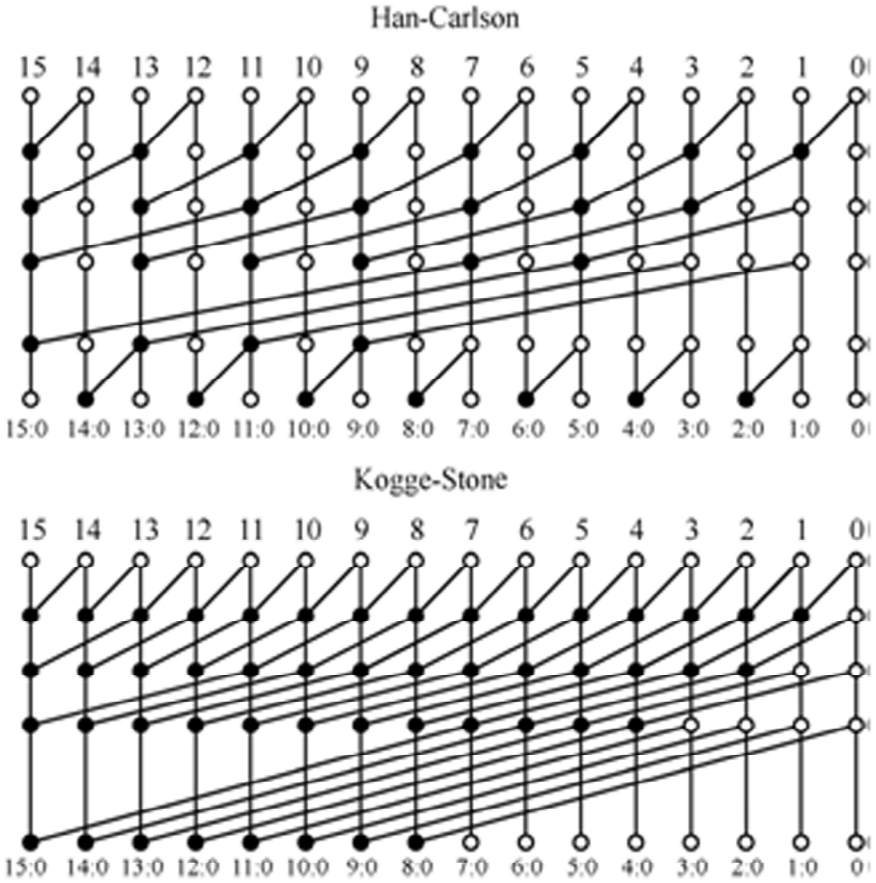

Figure 1: 16 bit Han Carlson and Kogge Stone Topologies

Stone adder. The Kogge-Stone adder is generally used when speed is the first (or most important) concerns the most, as it uses the lowest possible number of logic levels and each cell in the adder tree has fan-out of two. This comes at the cost of using many spread-create cells and many wires that required to be routed between stages. As it is shown in the fig 1, it requires 6 stages to achieve output.

2.1. Variable Latency Speculative Adder

The general prefix addition can be done as follows: given *n*-bit augends *A* and *B* which generates the *n-bit* sum, c_i is the carry out of *i*-th bit. The sum s_i and carry c_i can be calculated as following

$$s_i = (a_i)xor(b_i)xor(c_{i-1}) \quad (2)$$

$$c_i = a_i b_i + a_i c_{i-1} + b_i c_{i-1} \quad (3)$$

Three stages are used to compute the sum; pre processing, prefix processing and post processing. In the pre processing stage the generate g_{ig} as propagate p_i signal are computed as follows

$$gi = ai.bi \quad (4)$$

$$p_i = (a_i)xor(b_i) \quad (5)$$

The pre processing and post processing stages are having only simple operations on signals local to each bit position, hence the adder's performance mostly depends on the prefix processing stage. Fig 1., shows the topologies of Kogge Stone and Han Carlson adders. There the black balls represent the prefix operator and white balls represent the simple placeholders.

Kogge Stone adder is composed by $\log_2(n)$ levels and also fanout of two is present at the each level using a large count of black balls and wire tracks. A good tradeoff between fanout and number of logic

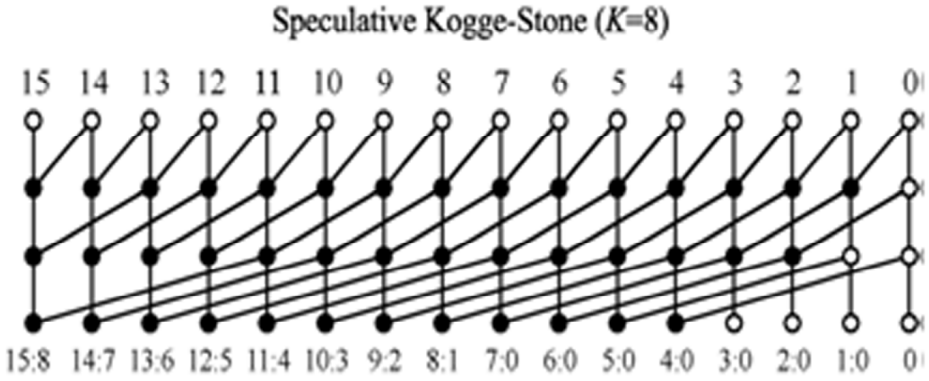

Figure 2: Kogge Stone speculative prefix processing stage where the last row of 16 bit Kogge Stone adder is been pruned which results in speculative adder with K = 8.

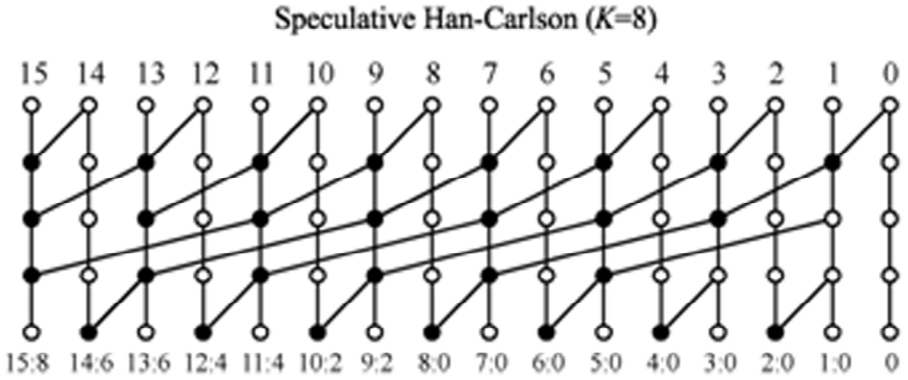

Figure 3: Han Carlson speculative prefix preprocessing stage where the last row of Han Carlson adder is been pruned to achieve speculative adder with K = 8

levels is given by Han Carlson as well as the tradeoff between the number of logic and the black balls is given by Han Carlson.

The Han Carlson adder is basically made of two adders, the outside rows of the Han Carlson adder are Brent Kung and similarly inside rows are Kogge Stone graphs. The Han Carlson in fig 1 is made of Brent Kung where single level at the beginning and single level at the bottom are Brent Kung graph, the number of levels are $1+\log_2(n)$. The normal stage has been divided in 3 stages where as the variable latency speculative prefix adder can be divided in to 5 stages which are as follows; pre processing, speculative prefix processing, post processing, error detection and error correction stages.

In the speculative stage the last rows of the adders are been pruned thus replacing the exact output with the speculative one as shown in the fig 2 and fig 3. To overcome this issue two stages have been designed they are error detection and error correction stages as shown in the fig 4. The error correction stage is have 2 clock cycle in the case of misprediction and when speculation fails. The pre processing stage g_i is been generated and as well as p_i is been propagated.

In the error detection stage the conditions where at least one of the propagation carries is wrong are signaled by the error detection block. It has 2 clock cycles in case of any misprediction the error signal is backed up by error detection stage and output of post processing stage is discarded. The error correction stage will give the right output in the next stage. The error signal of either Kogge Stone or Han Carlson is calculated by or-ing the p_i and g_i. The error rate analysis i.e, the value of the error probability is fundamental to understand the degradation of average time by misprediction for that Han Carlson and Kogge Stone adders have been simulated by using the Monte Carlo approach wih 1% relative error and 99% confidence level. The black balls are employes with AO gates. After the error signal has been detected by the error detection the error signal has to be corrected which is done by help of error correction stage.

3. PROPOSED METHOD

In the proposed method the adders are been replaced with the sqrt adder, it is also known as sqrt csla which uses the binary to excess converter instead of rca with cin = 1 to achieve lower delay and less area. The common way of implementation of csa is that it computes the (n + 1) bit sum of any two n-bit numbers.

Here instead of using a couple of rca blocks we are using the modified sqrt csla which developed using just single carry adder with bec converter which also replaces rca block for cin = 1 to improve the performance of the adder

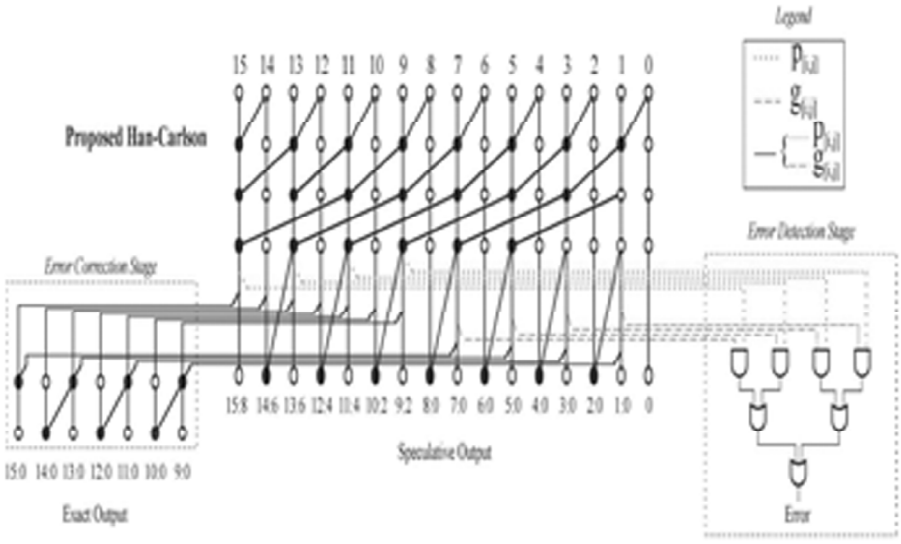

Figure 4: Speculative Han Carlson adder with the Error Detection and Error Correction stage

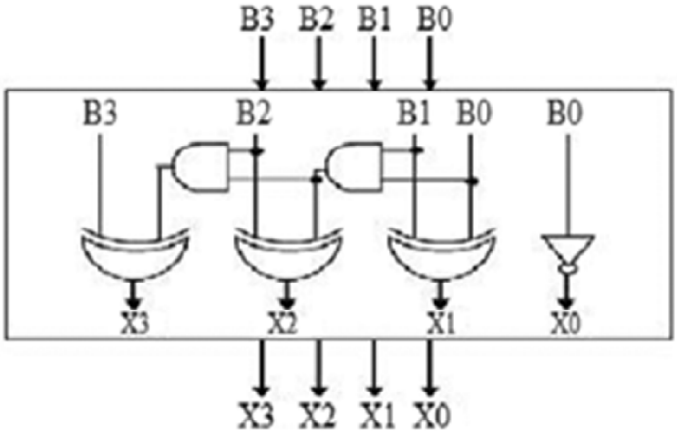

Figure 5: Binary to Excess one converter(BEC) circuit

The bec replaces the rca it shows better performance. The synthesized results are given in the below section as well as the simulated output. When the output of the Kogge Stone, Han Carlson and speculative Han Carlson is compared the speculative Han Carlson performs better and when the sqrt adders are includes it almost show 5.77%improveent in the performance.

4. SIMULATION RESULTS

The simulation is been performed using Xinx 13.1 software where also the timing, power and analysis has been done there is a noticeable change in the delay on the other hand the power and area are almost same, the change in the delay as been shown in the below table

Table 1: Delay comparision Table

Technique used	Delay (in ns)
Kogge Stone	6.7
Han Carlson	5.2
Modified sqrt csla	4.901

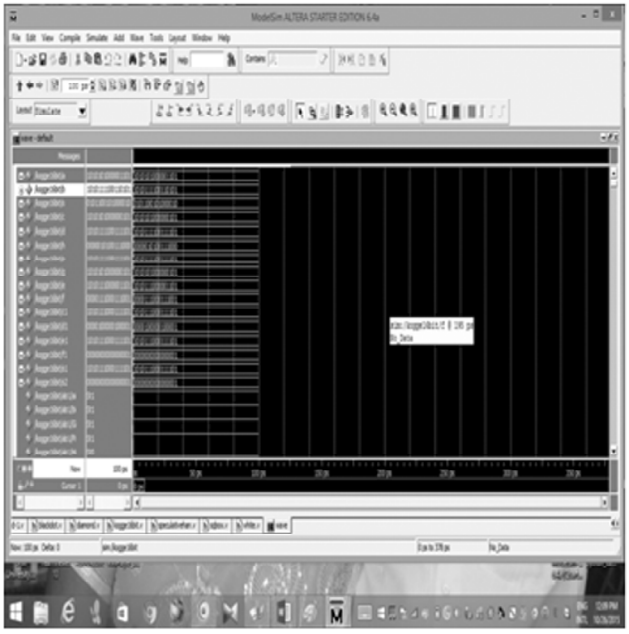

Figure 6: The Output of the Kogge Stone adder.

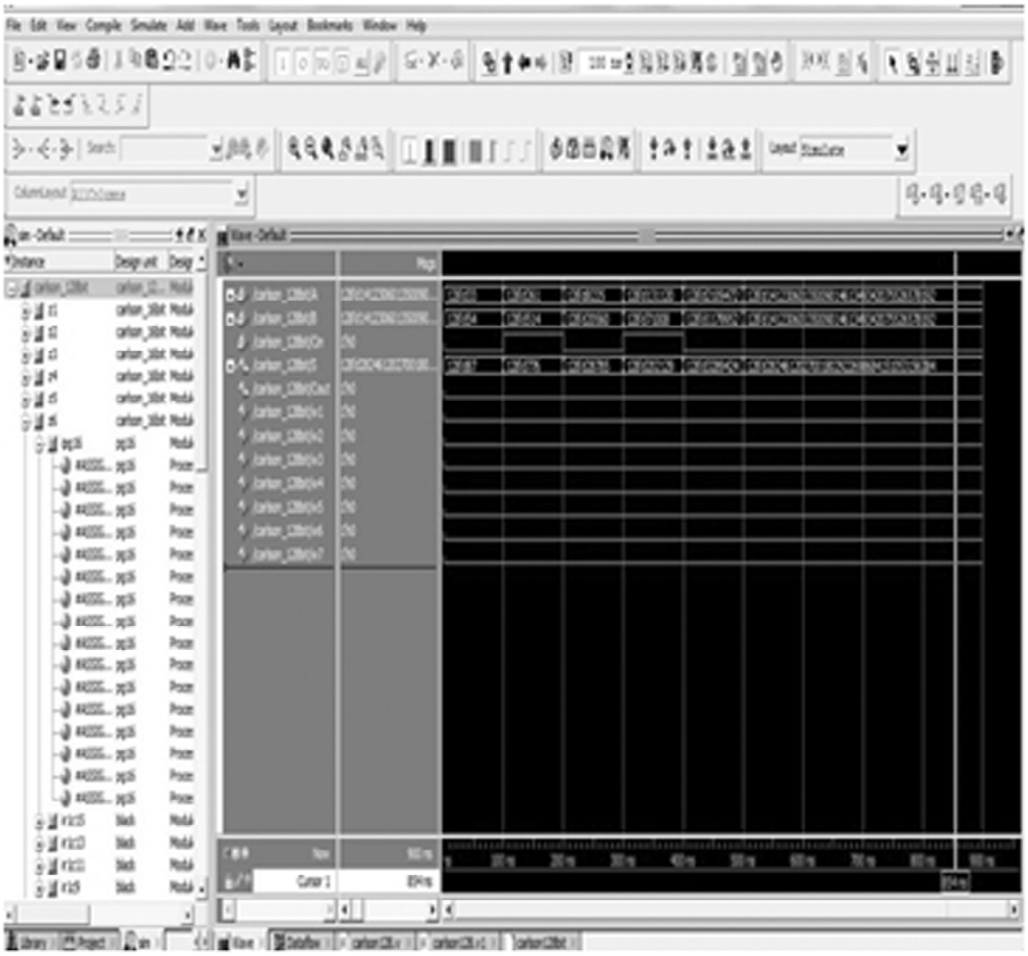

Figure 7: The Output of the Han Carlson adder using speculative adders.

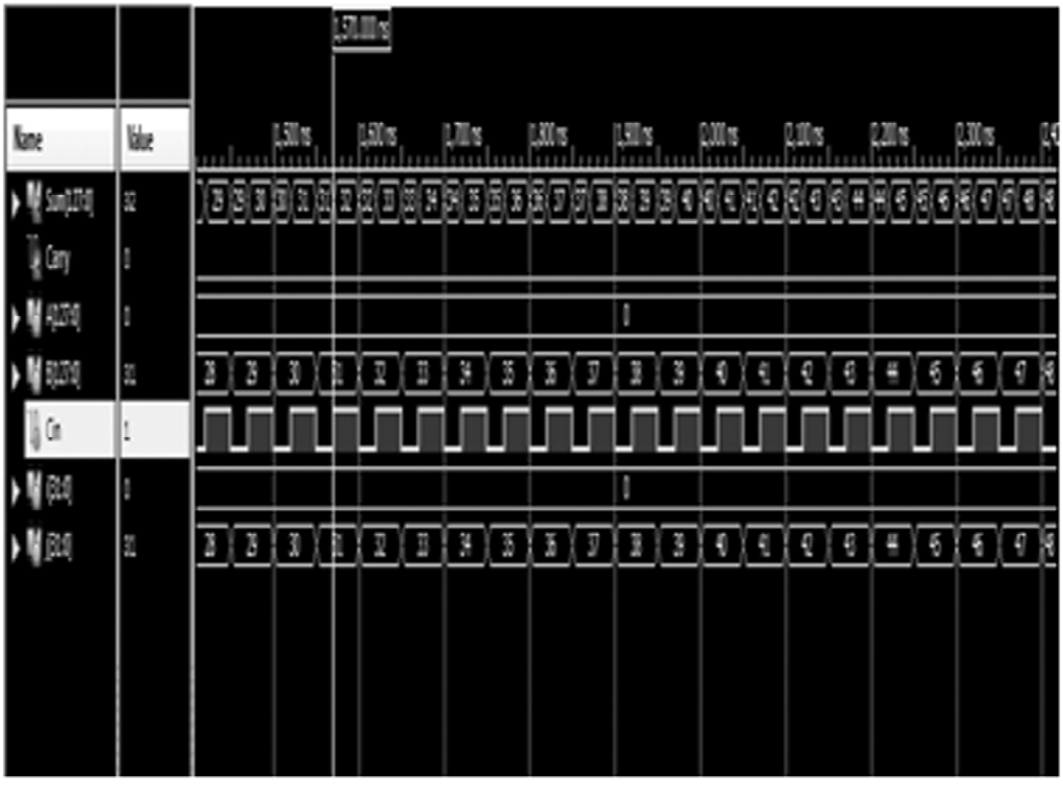

Figure 8: The Output of the Han Carlson adder using sqrt adder with bitexcess

5. CONCLUSION

From the study and analysis done on the Han Carlson Adder where the general adders are replaced by the sqrt adders the performance has show noticeable improvement where there is almost 5.5% improvement in the delay of the adder in Han Carlson adder when compared to the BEC based and RCA based adder structures.

REFERENCES

[1] David H.K. Hoe, Chris Martinez and Sri Jyothsna Vundavalli, "Design and Characterization of Parallel Prefix Adders using FPGAs", 2011 IEEE 43rd Southeastern Symposium in pp. 168-172, 2011.

[2] N. H. E. Weste and D. Harris, CMOS VLSI Design, 4th edition, Pearson–Addison-Wesley, 2011.

[3] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," IEEE Trans. Comput., vol. C-31, pp. 260-264, 1982.

[4] D. Harris, "A Taxonomy of Parallel Prefix Networks," in Proc. 37th Asilomar Conf. Signals Systems and Computers, pp. 2213–7, 2003.

[5] P. M. Kogge and H. S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations," IEEE Trans. On Computers, Vol. C-22, No 8, August 1973.

[6] D. Gizopoulos, M. Psarakis, A. Paschalis, and Y. Zorian, "Easily Testable Cellular Carry Lookahead Adders," Journal of Electronic Testing: Theory and Applications 19, 285-298, 2003.

[7] T. Lynch and E. E. Swartzlander, "A Spanning Tree Carry Lookahead Adder," IEEE Trans. on Computers, vol. 41, no. 8, pp. 931-939, Aug. 1992.

[8] Beaumont-Smith, A, Cheng-Chew Lim , "Parallel prefix adder design", Computer Arithmetic, 2001. Proceedings. 15th IEEE Symposium, pp. 218 – 225, 2001. M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[9] K. Vitoroulis and A. J. Al-Khalili, "Performance of Parallel Prefix Adders Implemented with FPGA technology," IEEE Northeast Workshop on Circuits and Systems, pp. 498-501, Aug. 2007. 172.

[10] S. Xing and W. W. H. Yu, "FPGA Adders: Performance Evaluation and Optimal Design," IEEE Design & Test of Computers, vol. 15, no. 1, pp. 24-29, Jan. 1998.