SPECULATIVE HAN-CARLSON ADDER FOR THE IMPROVEMENT OF SPEED OF CONVENTIONAL CARRY SKIP ADDER

Mr. M. Jaiganesh[1], Ms. S.Madhumadhi[2], Ms. V.Lavanya[3], Ms.M.Manju[4], Ms.R.Kiruthiga[5]

[1]*Asst. Professor, Dept. of Electronics and Communication Engineering, Panimalar Engineering College, Tamil Nadu, India*

[2] *Student, Dept. of Electronics and Communication Engineering, Panimalar Engineering College, Tamil Nadu, India*

[3] *Student, Dept. of Electronics and Communication Engineering, Panimalar Engineering College, Tamil Nadu, India*

[4] *Student, Dept. of Electronics and Communication Engineering, Panimalar Engineering College, Tamil Nadu, India*

[5] *Student, Dept. of Electronics and Communication Engineering, Panimalar Engineering College, Tamil Nadu, India*

ABSTRACT

This paper was designed to enhance the speed performance of the conventional carry skip adder. Many methods have already been designed for this purpose. One such method was the use of concatenation and incrementation scheme where the multiplexer logic was replaced by the And-Or-Invert (AOI) and Or-And-Invert (OAI) logic blocks. Also for further speed improvement the parallel prefix adder was introduced in the former system. This parallel prefix adder that was employed is the Brent Kung Adder. So, in this paper for further improving the speed, we make use of Speculative Han-Carlson Adder in place of Brent Kung Adder. Here, the adder can be implemented in both fixed stage size and variable stage size. The results were analyzed using Xilinx 13.2 software. The outcomes proved that Speculative Han-Carlson Adder showed a delay of 11.381ns which is less than that of Brent Kung Adder having a delay of 14.64ns.

Keyword—*Carry skip adder, speed improvement , Speculative Han-Carlson, parallel prefix adder.*

1.INTRODUCTION

As technology is advancing towards the core, the need for high speed is very demanding. The most basic units of any processor or any other electronic gadgets is an Arithmetic Logical Unit commonly referred to as ALU. For this ALU to have better performance, it is necessary to increase the speed of the adder circuits that has been inbuilt within them. There are many number of researches that are still going on for achieving high speeds. This also burdens many designers of general purpose processors.

There are many adders serving this purpose. Beginning from Half adder to the new generation of parallel prefix adders, many adders have been designed. Some of them includes Ripple Carry Adder (RCA), Carry Select Adder(CSA), Carry Lookahead Adder(CLA), and Parallel Prefix Adders (PPA). Each of them have their own advantages and disadvantages. For instance, the RCA has simple structure, small area and low power consumption. But it suffers from a serious disadvantage of high critical path delay. On the other hand, CLA has high speed compared to RCA but it also has high power consumption and requires high area .

The PPA adders are of many types such as Brent kung adder, Kogge-Stone adder, Ladner-Fischer adder, Knowles adder, Sklansky adder and Han-Carlson adder. Of all these, the Brent kung adder has the lowest area with high delay whereas the Kogge-Stone adder has the highest speed. Combining the advantages of these two adders is the Han-Carlson adder with low area requirement and with considerable speed.

The next is the Carry Skip Adder which is the best of all. It has low power consumption with less area requirement. The critical path delay is very less compared to RCA with the power-delay product smaller than CLA and PPA. It has short wiring length and provides simple layout for easy designing. But it has a serious disadvantage of low speed compared to other structures.

This feature of CSKA can be used for the initiation of this project. This paper focusses on improving the speed of conventional CSKA without comprimising the other factors that make it advantageous compared to other adders.

The objectives of this paper are summarized below:

i. Improving the speed of conventional Carry Skip adders by using concatenation and incrementation schemes for the skip logic. This makes the skip logic more simple since AOI and OAI logics are used rather than multiplexers.

ii. A hybrid variable latency structure is propose based on the extensions of the conventional CSKA, by replacing some of the middle structures in this paper by PPA. The PPA that has been used is the Speculative Han-Carlson Adder.

The rest of this paper is summarized as follows:

Section 1. gives a general information about the overall paper followed by Section 2, that describes the existing system, speed enhancement achieved by modifying the CSKAs, its structure, implementation of variable latency adders and its pros and cons. The Section 3 provides the details of the proposed system and its structure. The Section 4. Analyses and compares the simulation results of both the existing system and the proposed system. Finally Sections 5 and 6, that gives the overall conclusion of the paper and references respectively.

2. EXISTING SYSTEM

Since this paper is concerned with the modification of speed of the conventional CSKA, let us first analyse the prior work of the structures.

2.1SPEED ENHANCEMENT ACHIEVED BY MODIFYING THE CSKAs

The previous structure of CSKA consisted of RCA blocks for each stage. The RCA was implemented as a chain of full adders. Also for the skip logic, multiplexers were employed. The RCA blocks of successive stages were connected through 2:1 multiplexers. This CSKA configuration had a great impact on the speed and performance of the overall system. This was due to the number of full adders that increased with the number of stages. Many methods have been suggested for optimizing the number of full adders. Also Variable Stage Size (VSS) were used for minimizing the delay of adders based o single level carry skip logic. Methods were also designed for increasing the speed of multilevel CSKAs. In addition carry look ahead logics were also used for reducing the propagation delay of the adders. But this resulted in large layout with a high power consumption and large area requirement. And this design approach was not general to all types of adders with different bit lengths and only supported 32 bit length adders.

Alioto and Palumbo proposed a simple strategy for designing a single level CSKA. In this method, VSS technique was used where the near optimal number of full adders is determined based on skip time and ripple time. The skip time is the delay of the multiplexer and ripple time is the time taken for the carry to ripple from one FA to another FA. The aim of this paper was to reduce the critical path delay by considering non integer ratio of skip time to ripple time rather than the integer ratio that was used in the prior studies.

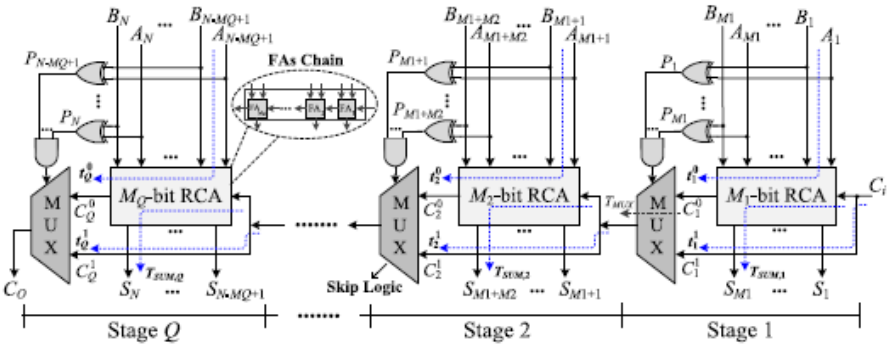

Fig-1: Conventional Carry Skip Adder

2.2 CONVENTIONAL CSKA

The conventional Carry Skip Adder consists of a chain RCA blocks each of which are made up of a sequence of FAs. The RCA blocks are connected through the multiplexer that serves as the skip logic along with some logic gates. If there are N cascaded FAs in an RCA block, then the worst propagation delay of the summation of two N-bit numbers A and B, belongs to the case where all FAs are in propagation mode. This worst case delay is given as

$$Pi = Ai \oplus Bi = 1 \text{ for } i = 1, \ldots, N$$

where P_i is the propagation signal. From this equation it is clear that the propagation delay is directly proportional to the number of cascaded FA stages. By concept, when a sequence of FAs are in propagate mode, then the carry out will be equal to the carry input. The skip logic will previously detect this and generate the carry for the next stage without waiting for the FA to complete the addition.

2.2.1FIXED STAGE SIZE CSKA

By assuming that each stage of the CSKA contains *M* FAs, there are $Q = N/M$ stages where for the sake of simplicity, we assume *Q* is an integer. The input signals of the *j*- th multiplexer are the carry output of the FAs chain in the *j* th stage denoted by *C*0*j* , the carry output of the previous stage (carry input of the *j* th stage) denoted by *C*1*J*. The critical path of the CSKA contains three parts:

- The path of the FA chain of the first stage whose delay is equal to $\boldsymbol{M \times T}$**CARRY**
- The path of the intermediate carry skip multiplexer whose delay is equal to $\boldsymbol{(Q-1)\times T}$**MUX**
- The path of the FA chain in the last stage that's its delay is equal to the $\boldsymbol{(M-1) \times T}$**CARRY** $+\boldsymbol{T}$**SUM.**

Note that *T*CARRY, *T*SUM, and *T*MUX are the propagation delays of the carry output of an FA, the sum output of an FA, and the output delay of a 2:1 multiplexer, respectively. Hence, the critical path delay of a FSS CSKA is formulated by

$$\boldsymbol{TD = [M \times T\text{CARRY}] + \_\_\ N\ M - 1\ \_ \times T\text{MUX}\ \_ + [(M-1) \times T\text{CARRY} + T\text{SUM}].}$$

Based on (1), the optimum value of *M* (*M*opt*)* that leads to optimum propagation delay may be calculated as *(*0.5*Nα)*1/2 where *α* is equal to *T*MUX/*T*CARRY.

Therefore, the optimum propagation delay (*TD,*opt*)* is obtained from

$$\boldsymbol{TD,\ opt = 2\ \_\ 2NT\text{CARRY}T\text{MUX} + (T\text{SUM} - T\text{CARRY} - T\text{MUX}) = T\text{SUM} + (2\sqrt{\ }2N\alpha - 1 - \alpha) \times T\text{CARRY}.}$$

Thus, the optimum delay of the FSS CSKA is almost proportional to the square root of the product of *N* and *α*.

2.2.2VARIABLE STAGE SIZE CSKA

As mentioned before, by assigning variable sizes to the stages, the speed of the CSKA may be improved. The speed improvement in this type is achieved by lowering the delays of the first and third terms. These delays are minimized by lowering sizes of first and last RCA blocks. For instance, the first RCA block size may be set to one, whereas sizes of the following blocks may increase. To determine the rate of increase, let us express the propagation delay of the *C*1*j* (*t*1*j*).

$$t_j^1 = \max\left(t_{j-1}^0, t_{j-1}^1\right) + T_{\text{MUX}}$$

where t^0_{j-1} (t^1_{j-1}) shows the calculating delay of C^0_{j-1}(C^1_{j-1}) signal in the ($j-1$)th stage.

2.3 CI-CSKA DESIGN USING BRENT KUNG ADDER

The proposed CSKA structure uses the combination of concatenation and incrementation schemes for the skip logic. This logic uses the AOI and OAI logic blocks along with some logic gates. This skip logic was introduced because the number of transistors used for AOI and OAI was only six compared to the multiplexers that used twelve transistors.

Another point to note here is that as the carry propagates through the successive stages, it becomes complemented alternatively. Thus the carry generated at the even stages will be complemented.

Hence this structure has a comparatively less propagation delay compared to that of the conventional CSKA with a slight increase in area. Though the power consumed by the AOI and OAI blocks is lesser than that of multiplexer, the overall structure imposes a slight increase in power consumption due to increase in number of gates which results in high wiring capacitance.

2.3.1 BLOCK DIAGRAM DESCRIPTION

There are two inputs to the adder namely A and B, with Q stages in total. Each stage contains a RCA block of M_j size where j=1,2,….Q. Based on concatenation mechanism, the input carry of all the blocks except for the first block is zero. Hence it is possible for all the blocks to compute the result independently and simultaneously.

In this structure, the first block consists of only one RCA whereas the remaining stages from 2 to Q contain an additional incrementation block.

The incrementation block is made up of a chain of half adders. The input to this block is the output of the RCA along with the carry generated at the previous stage. The carry output of the incrementation block is not considered to reduce the delay considerably.

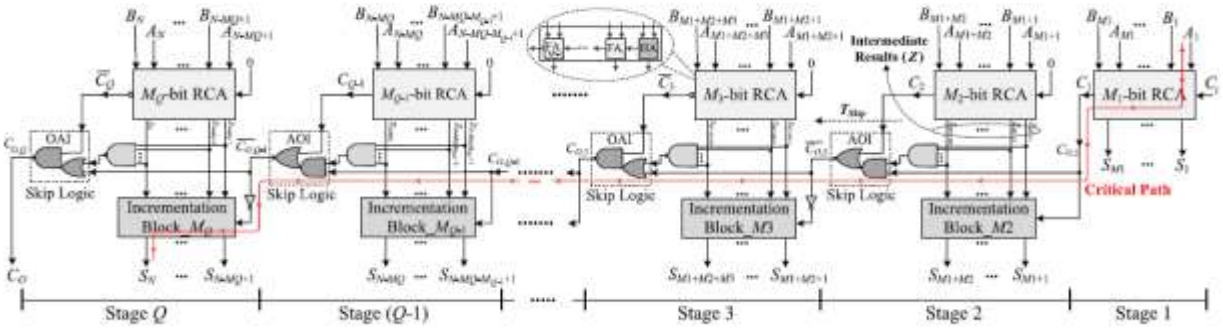

Fig-2: Proposed CI-CSKA structure

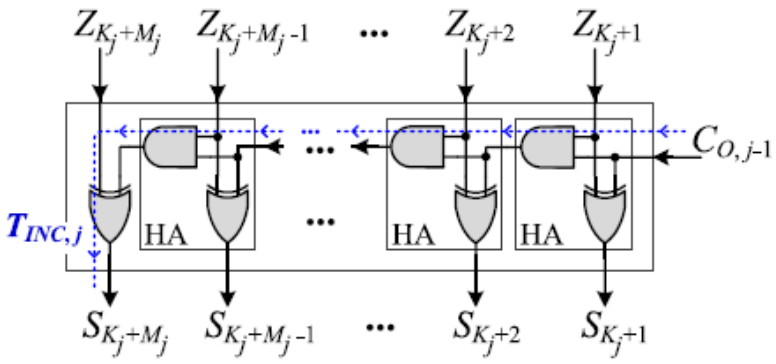

Fig-3: Incrementation Block

The AOI and OAI blocks are used as the skip logics because of their inverting functions in the standard cell library. This eliminates the use of the inverter as it consumes more power. But this skip logic increases the delay in the critical path. So, in order to reduce the critical path delay, the input carry of all the blocks is set to zero other than the first stage. This is called concatenation method by which the output carry of all the blocks can be computed in parallel.

2.3.2 HYBRID VARIABLE LATENCY CI-CSKA

The reason for using variable latency adders is that their critical paths are activated very rarely. This scales down the supply voltage thereby maintaining the clock frequency. If critical path is not activated, then one clock cycle is enough for the overall operation. On the other hand, two clock cycles are required if the critical path is activated. Hence for determining the critical path activation, a predictor block is used that works depending on the inputs.

Using the predictor block for the overall structure will increase the area and power consumption. Hence only the middle portions of the structure use this predictor block. The predictor block basically consists of AND and EXOR gates for determining the product of propagate signals.
There are some cases when the predictor block mispredicts the critical path activation. By increasing the number of bits in the predictor block, the number of mispredictions decreases thereby increasing the longest off-critical path. For balancing this trade off, the size of the predictor block must be appropriately selected.

2.3.3 HYBRID VARIALBE LATENCY CI-CSKA STRUCTURE

To provide variable latency feature for the VSS CSKA, the middle stages of the CI-CSKA are replaced by the PPA. An M_p –bit modified PPA is used for the pth stage. This pth stage is the nucleus stage having the largest size. Hence by replacing the nucleus stage by the PPA, the delay of the longest off-critical path will be reduced.

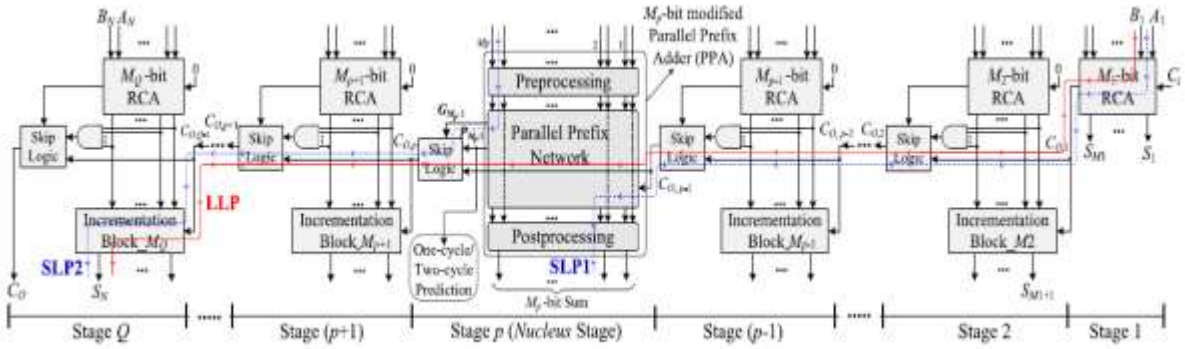

Fig-4: Hybrid Variable Latency CI-CSKA structure

In this structure, the PPA that has been used is the Brent Kung Adder. The advantage of using this adder is that it occupies very less area with fan-out of 2 and only one wiring track. But it possesses a serious disadvantage of high propagation delay. Then the idea of using Kogge-Stone adder was discussed. But then it too possessed some disadvantages of low fan-out with a complex circuitry in spite of having the highest speed among all other PPAs.

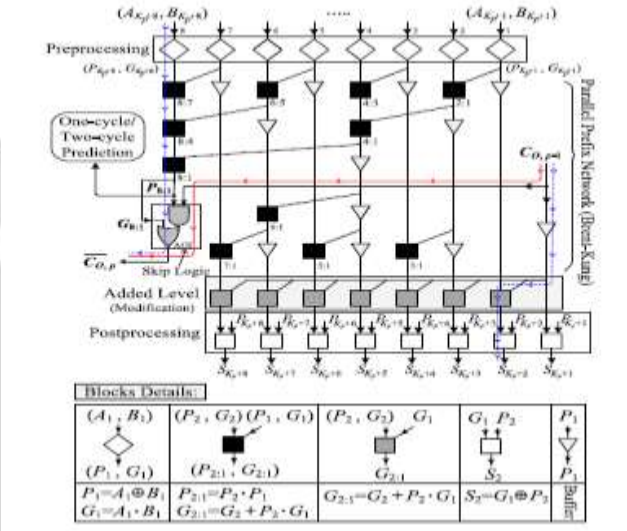

Fig-5: Tree diagram of Brent Kung adder

3. PROPOSED SYSTEM

The proposed system is similar to that of the existing system. But here the Brent Kung adder that was used as PPA is now replaced by Speculative Han-Carlson adder.

The concept of Han-Carlson is introduced. The idea of choosing this adder was that, it combined the advantages of both Brent Kung adder and Kogge-Stone adder. Thus, providing a considerable speed as that of Kogge- Stone adder and area as low as that of Brent Kung adder.
The operation of the PPAs can be divided into three structural stages. They are as follows:

Stage 1: Computation of generate and propagate signals.
Stage 2: Compute the carry. This stage makes use of two properties namely associative property and idem potency property. This is used for parallelized computation of the carry of each stage.
Stage 3: Computation of final sum.

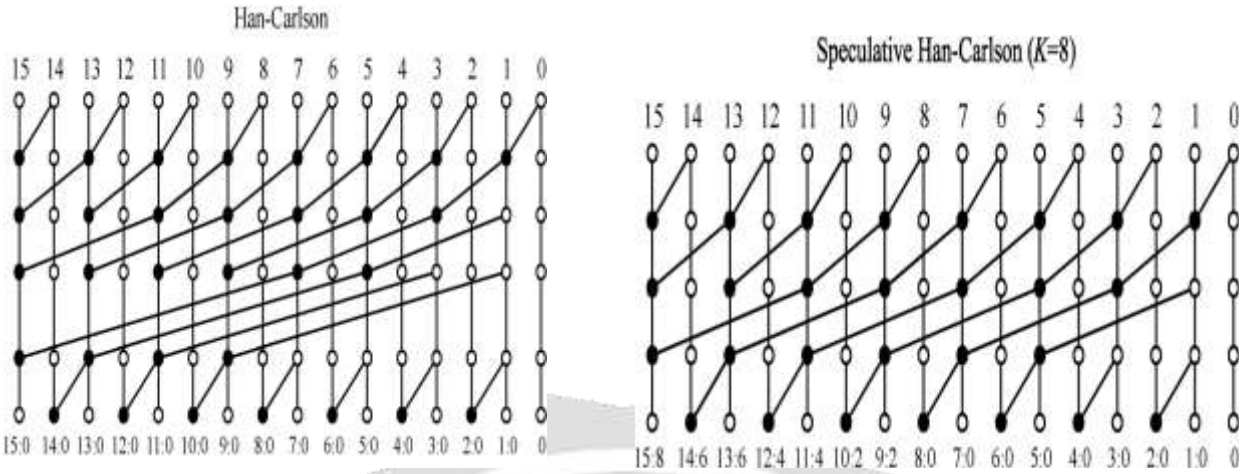

Fig-6: Tree diagram of Han-Carlson and Speculative Han-Carlson adders

For further speed improvement, the Han-Carlson adder can be replaced by the Speculative Han-Carlson adder. In this adder, the exact arithmetic function is replaced with an approximated one that is faster and gives the correct result most of the time, but not always. The approximated adder is augmented with an error detection network that asserts an error signal when speculation fails.

Speculative variable latency adders have attracted strong interest thanks to their capability to reduce average delay compared to traditional architectures. But, non-speculative adders remain the best choice when the speed constraint is relaxed.

T. Han and D.A. Carlson presented a hybrid construction of a parallel prefix adder using two designs the Kogge-Stone construction having the best feature of higher speed and the Brent-kung construction with best feature of low area requirement. A modified Han-Carlson adder uses fewer number of prefix operations by adjusting the number of stages amongst Kogge-Stone and Brent-kung adder and thus reduces the area required by the adder circuitry.

The idea of Han-Carlson prefix tree is similar to Kogge-Stone's structure since it has a maximum fan-out of 2 or f = 0. The difference is that Han-Carlson prefix tree uses much less cells and wire tracks than Kogge-Stone. The cost is one extra logic level. Han-Carlson prefix tree can be viewed as a sparse version of Kogge-Stone prefix tree. In fact, the fan-out at all logic levels is the same (i.e. 2). The pseudo-code for Kogge-Stone's structure can be easily modified to build a Han-Carlson prefix tree. The major difference is that in each logic level, Han-Carlson prefix tree places cells every other bit and the last logic level accounts for the missing carries.

This type of Han-Carlson prefix tree has log2n + 1 logic levels. It happens to have the same number cells as Sklansky prefix tree since the cells in the extra logic level can be move up to make the each of the previous logic levels all have n=2 cells. The area is estimated as (n/2)log2n. 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.

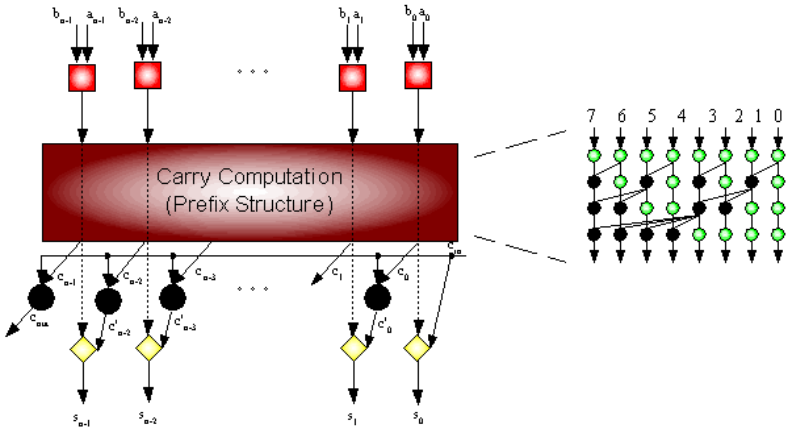

Fig-7: Prefix structure of Speculative Han-Carlson adder

4. SIMULATION RESULTS

Both the structures namely CI-CSKA using Brent Kung adder and CI-CSKA using Speculative Han-Carlson adder were analyzed. The results were obtained using Xilinx 13.2. The outcomes showed that the Brent Kung adder had a delay of 14.640ns which is considerably high compared to Speculative Han- Carlson adder that produced a delay of 11.381ns. The adders that were implemented are both 32-bit adders.

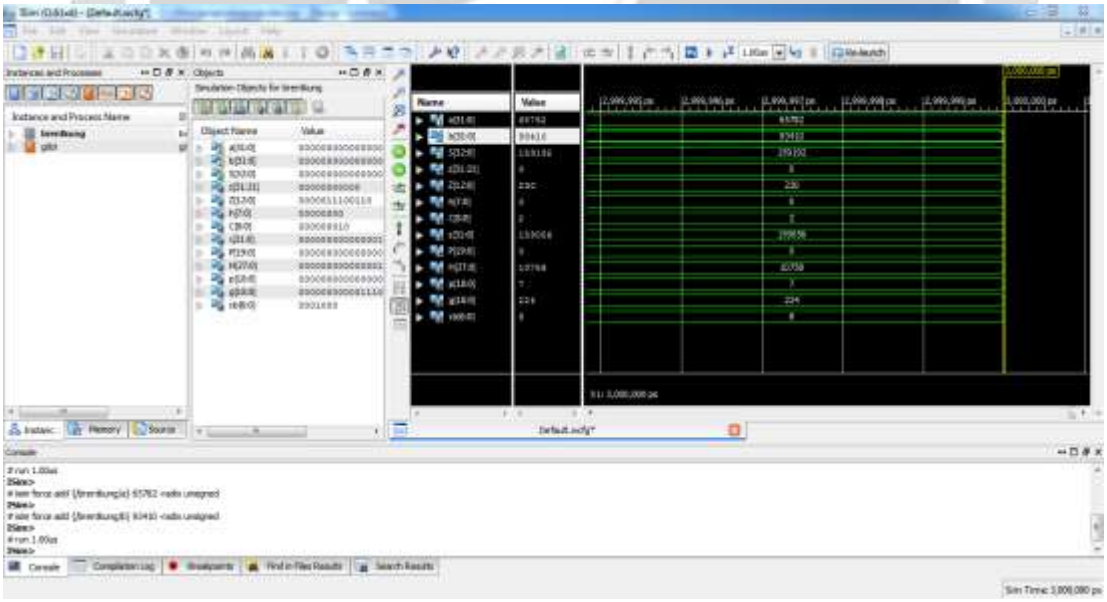

Fig-8: Snap shot of simulated result of CI-CSKA using Brent Kung adder

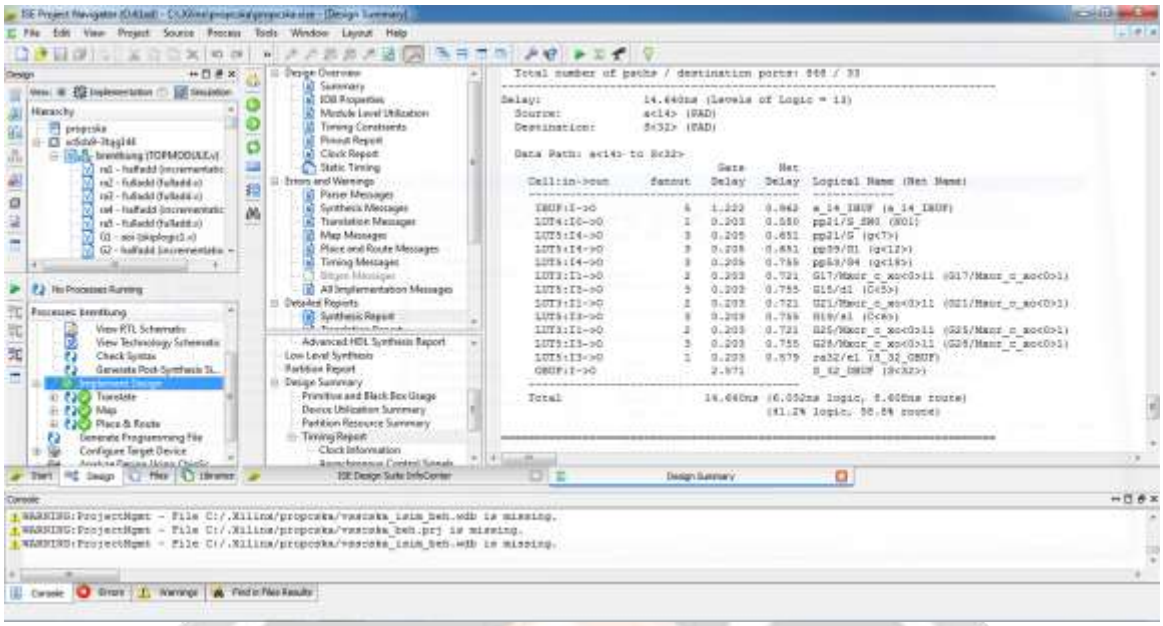

Fig-9: Snap shot of timing report of CI-CSKA using Brent Kung adder

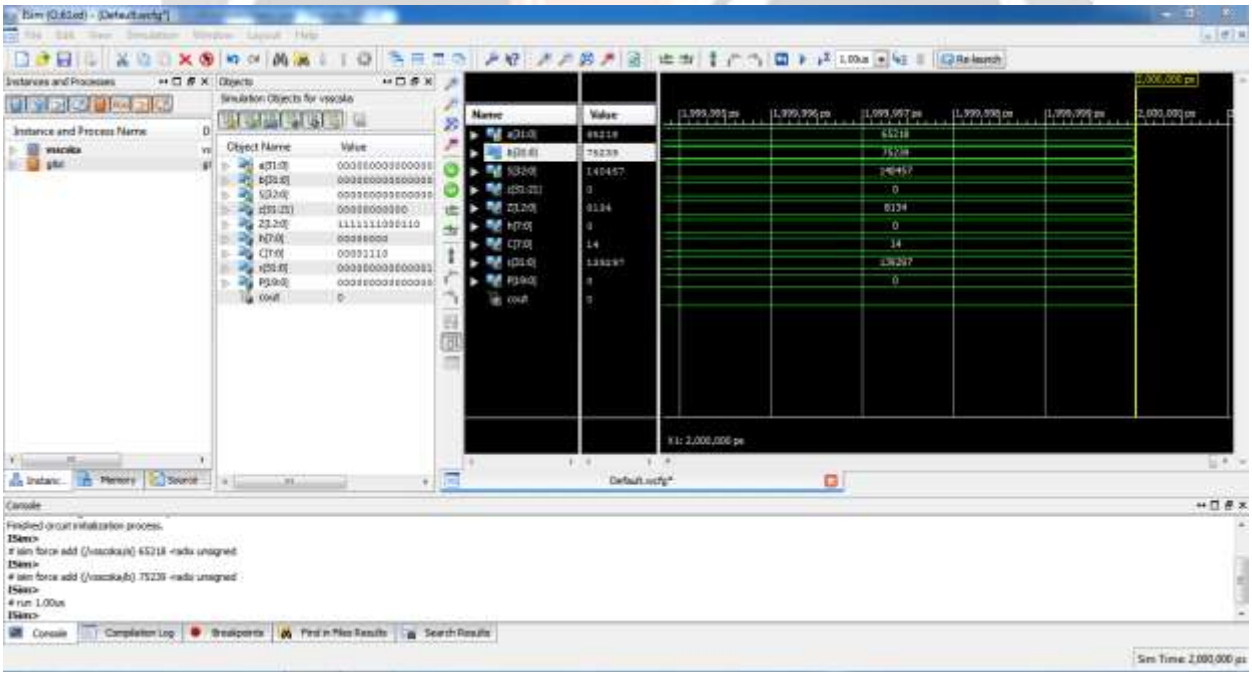

Fig-10: Snap shot of the simulated result of CI-CSKA using speculative Han-Carlson adder

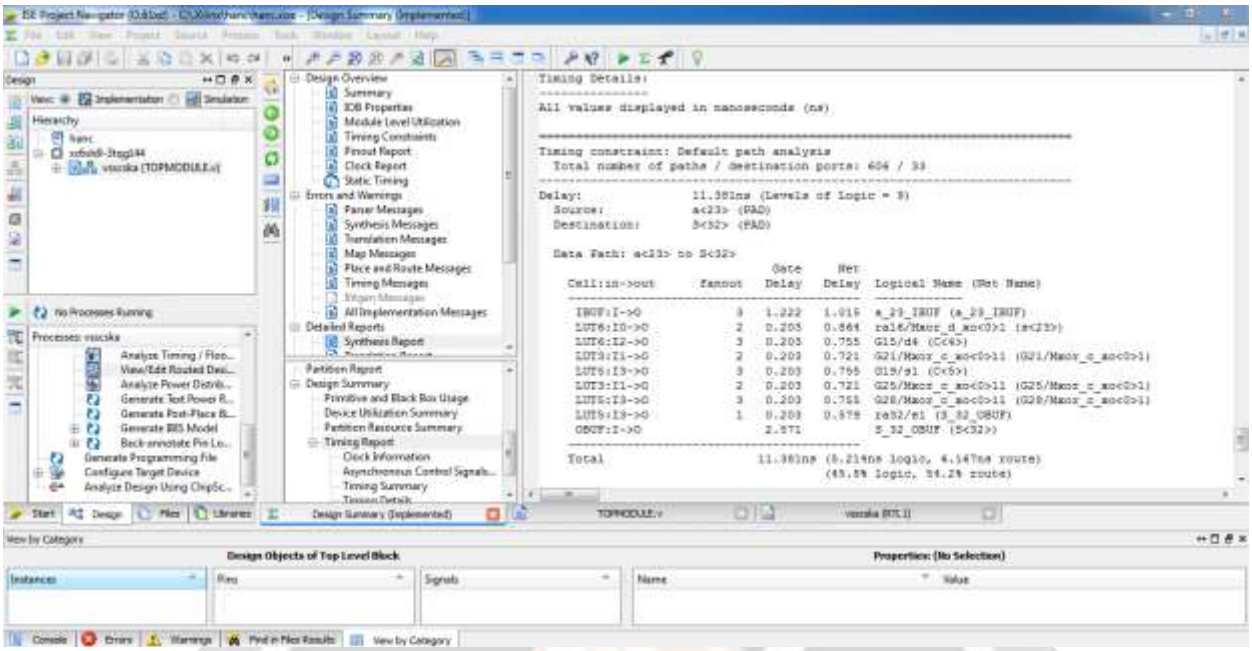

Fig-11: Snap shot of timing report of CI-CSKA using speculative Han-Carlson adder

5. CONCLUSIONS AND FUTURE ENHANCEMENTS

In this paper, a CI-CSKA using Speculative Han- Carlson adder was proposed. The speed enhancement was achieved by applying concatenation and incrementation schemes to the conventional CSKA. Also for further improvement of speed, the critical path delay was reduced by using a variable latency adder in the middle bit positions. The variable latency adder was built using a PPA. The PPA that was used is the Speculative Han- Carlson adder. The results showed that speculative Han-Carlson adder had low delay compared to the existing Brent Kung adder.

The proposed system has been implemented in 32-bit length. For further modification, it can be implemented as 64, 128 or 256-bit lengths to make it compatible with the current technology and devices. Also for further improving the performance, the PPA can make use of Lynch-Swartzlanger adder in the nucleus stage.

6. REFERENCES

[1] D. Harris, "A taxonomy of parallel prefix networks," in *Proc. IEEE Conf. Rec. 37th Asilomar Conf. Signals, Syst., Comput.*, vol. 2. Nov. 2003, pp. 2213–2217.

[2] M. Lehman and N. Burla, "Skip techniques for high-speed carry propagation in binary arithmetic units," *IRE Trans. Electron. Comput.*, vol. EC-10, no. 4, pp. 691–698, Dec. 1961.

[3] K. Chirca *et al.*, "A static low-power, high-performance 32-bit carry skip adder," in *Proc. Euromicro Symp. Digit. Syst. Design (DSD)*, Aug./Sep. 2004, pp. 615–619.

[4] A. Guyot, B. Hochet, and J.-M. Muller, "A way to build efficient carryskip adders," *IEEE Trans. Comput.*, vol. C-36, no. 10, pp. 1144–1152, Oct. 1987.

[5] P. K. Chan, M. D. F. Schlag, C. D. Thomborson, and V. G. Oklobdzija,\"Delay optimization of carry-skip adders and block carry-lookahead adders using multidimensional dynamic programming," *IEEE Trans. Comput.*, vol. 41, no. 8, pp. 920–930, Aug. 1992.

[6] V. Kantabutra, "Designing optimum one-level carry-skip adders," *IEEE Trans. Comput.*, vol. 42, no. 6, pp. 759–764, Jun. 1993.

[7] Y. Chen, H. Li, J. Li, and C.-K. Koh, "Variable-latency adder (VL-adder): New arithmetic circuit design practice to overcome
NBTI," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design (ISLPED)*, Aug. 2007, pp. 195–200.

[8] Y. Liu, Y. Sun, Y. Zhu, and H. Yang, "Design methodology of variable latency adders with multistage function speculation," in *Proc. IEEE 11th Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2010, pp. 824–830.

[9] Y.-S. Su, D.-C. Wang, S.-C. Chang, and M. Marek-Sadowska, "Performance optimization using variable-latency design style," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 10, pp. 1874–1883, Oct. 2011.

[10] 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.

[11] Darjn Esposito; Davide De Caro; Michele De Martino; Antonio G. M. Strollo, "Variable latency speculative Han-Carlson adders topologies" **IEEE Conference Publications** 2015 11th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)

[12] K. Golda Hepzibha; C P. Subha, "A novel implementation of high speed modified brent kung carry selectadder"**IEEE Conference Publications** 2016 10th International Conference on Intelligent Systems and Control (ISCO)

[13] Milad Bahadori, Mehdi Kamal, Ali Afzali-Kusha, *Senior Member, IEEE*, and Massoud Pedram, *Fellow, IEEE, "*High-Speed and Energy-Efficient Carry Skip Adder Operating Under a Wide Range of Supply Voltage Levels",