

# Energy-Efficient Inexact Speculative Adder Using Carry Look Ahead Adder and Brent Kung Adder

Swetha Bai<sup>1</sup>, Vijay Murugan<sup>2</sup>

P.G Student, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India<sup>1</sup>

Assistant Professor, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India<sup>2</sup>

**ABSTRACT:** This paper presents the design of a novel architecture of a contemporary Inexact speculative adder with optimized hardware efficiency and advanced compensation Method with both mistakes correction or blunders reduction that's quality grain pipelined to encompass few common sense gates along its bring propagation chain which is taken into consideration as the critical path of the adder and thereby, enhancing the frequency of operation using carry look ahead adder and Brent Kung adder. The ISA enhances the performance of the adder through splitting the essential course into two or greater shorter paths, decreasing spurious glitching energy and errors management through an optimized speculative path and with a versatile dual direction error compensation technique. Pipelining is the process of shortening the critical path at the cost of area. The preferred topology of speculative adders improves performance and allows precise accuracy control This work exploits CLA in the architecture and Brent Kung adder in the proposed architecture, as it has smaller propagation delay compared to the conventional adders. Inexact and approximate circuit design is a promising approach to improve performance and energy efficiency in technology- scaled and low-power digital systems. Such strategy is suitable for error-tolerant applications involving perceptive or statistical outputs. Thereafter, the HDL(Verilog) coded architecture of 16 -bit proposed ISA synthesis and simulation using Xilinx software ISE 14.2 for power, area analysis and improving the energy-delay-area product(EDAP) beyond the theoretical bounds of exact adders.

**KEYWORDS:** Error Detection, Speculation, Critical Path delay, Carry look ahead adder.

## I. INTRODUCTION

High-pace adders are enormously appropriate inside the cutting-edge state of affairs, even though power (or strength) and silicon area are equally important. Spectrum sensors used in clever cognitive-radio surroundings as well as net of the whole lot (IoE) gadgets focused on bodily interfaces are largely-explored studies regions inside the latest time. Hardware for the algorithms of such programs is essentially focused on sensing and actuating in which the response time is key element to be optimized for real- time interfaces. Thereby, the design of noticeably optimized adders in phrases of speed play great function in the gift generation and as a result this paper focuses inside the design of equal. With tolerable degradation in accuracy and overall performance, it's miles feasible to conceive excessive-pace, low power and place efficient layout using inexact and approximate circuit technique. Accuracy of such circuits may be traded off to enhance the energy and pace by way of speculation. Thereby, such adders are referred as inexact speculative adder (ISA). Various optimized variations of such ISA had been pronounced in literature and these works concentrated mostly on improving the accuracy in their effects. However, there is space to further improve the velocity of such adders via preserving the accuracy with minimal blunders. Thereby, our contributions in this paintings are as follows: design and

analysis of the deliver look ahead adder (CLA) based totally ISA has been done. Thereafter, this adder is quality grain pipelined to reduce the critical path delay that further enhances the operating speed. FPGA implementation of 8, 16 and 32 bit versions of the proposed ISA has been carried. Obtained post place- &-route results of these adders are compared with reported non-pipelined ISAs.

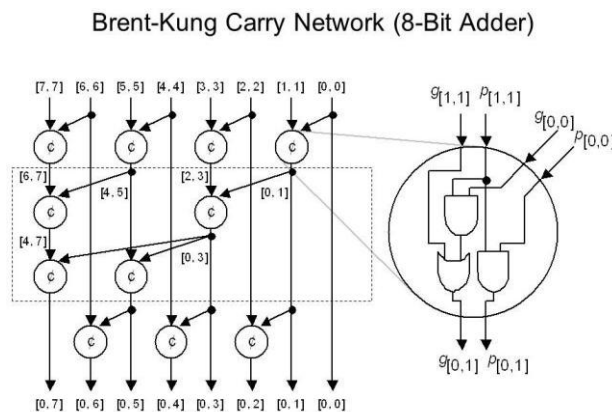


Fig.1 Brent kung adder

**II. RELATED WORKS**

**A. BRENT KUNG ADDER**

The kind of structure of any adder substantially affects the velocity of the circuit. The logarithmic structure is taken into consideration to be one of the fastest structures. The logarithmic idea is used to combine its operands in a tree-like fashion. The logarithmic delay is obtained by using restructuring the appearance-beforehand adder. The restructuring is dependent on the associative property, and the put off is acquired to be same to  $(\log_2 N) t$  wherein  $N$  is the range of input bits to the adder and  $t$  is the propagation postpone time. Hence, for a 16-bit structure, the logarithmic adder has a postpone equal to  $4t$  while for a easy ripple carry adder the delay is given by using  $(N-1)t$  and is identical to  $15t$  for  $N$  and  $t$  being the range of enter bits and the delay time, respectively. Hence it's far seen that this structure substantially reduces the delay, and might be especially useful for a structure with big wide variety of inputs. This advantage is, however, acquired at the price of massive vicinity and a complex structure.

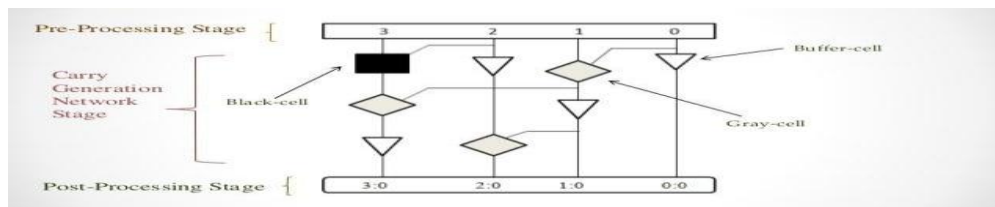


Fig.2 Preprocessing Stage

where  $a$  is a function defined in order to access all the tuples. The 4-bit Brent Kung Structure is shown in figure This figure shows all the carry signals generated at different stages in the structure. In the structure, two binary tree structure are represented -- the forward and the reverse trees. The forward binary tree alone is not sufficient for the generation of all the carry signals. It can only generate the signals shown as  $Co,0, Co,1, Co,3$  and  $Co,7$ . The remaining carry signals are generated by the reverse binary tree.

**Organized by**

**Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India**

The Brent-Kung tree computes prefixes for 2-bit groups. These are used to locate prefixes for four-bit agencies, which in turn are used to discover prefixes for 8-bit corporations, and so on. The prefixes then fan go into reverse to compute the incorporates-in to every bit. The tree requires  $2\log_2 N-1$  stages. The fan-out is constrained to two at every level.

The diagram indicates buffers used to reduce the fan-out and loading on the gates, but, in workout, the buffers are usually disregarded. The easy blocks utilized in this case are gray and black cells that are described. This adder is implemented for 8 bit using CMOS commonplace sense and transmission gate logic. The Brent-Kung adder is a parallel prefix adder. Parallel prefix adders are special class of adders that are primarily based on using generate and propagate indicators. Simpler Brent-Kung adders were proposed to remedy the risks of Kogge-Stone adders. The price and wiring complexity is greatly reduced. But the good judgment depth of Brent-Kung adders will increase to  $2\log(2n-1)$ , so the rate is decrease. We propose a method to lessen postpone and strength ate up by means of the Brent Kung adder by using reading and dividing the inputted records into blocks a good way to best be introduced if it holds any fee at all. Hence the inputted values are first of all in comparison before finding out up to how a few of the adder blocks must be activated. This method can without difficulty be incorporated to the prevailing design of the Brent Kung adder, for this reason making it greater green.

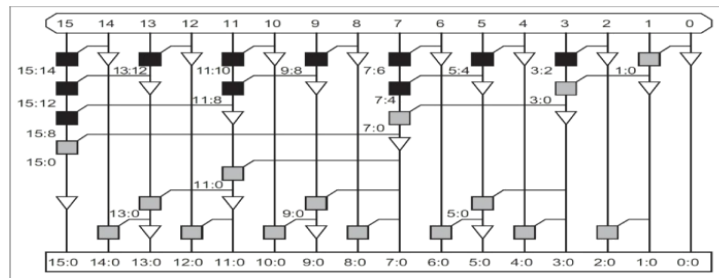


Fig.3 Brent kung tree

#### IV. PROPOSED SYSTEM

Block diagram and data flow of conventional ISA for  $n$ - bit addition is shown in Figure. In the proposed structure, we've got segregated the  $n$ -bit enter into four-bit blocks (i.E., the price of  $x =$  four in Figure and every of these blocks is fed as operands to the  $x$ -bit adder. Unlike the conventional ISA architecture, the adder unit has been replaced with 4-bit CLA to further enhance the speed of operation. Comprehensive explanation with circuit details of various sub blocks of this adder are presented as follows

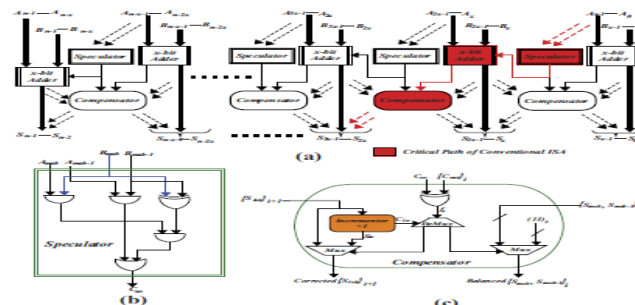


Fig 4: (a) basic block diagram of  $n$ -bit conventional in-exact speculation adder (isa). (b) gate-level circuit representation of speculator block.(c) Digital architecture of compensator block

##### A. Speculator and Adder Blocks

Prior delving into the circuit details, it is necessary to understand the notations used in this paper. Two  $n$ -bit operands for addition are represented as  $A = \{A_0, A_1, \dots, A_{n-1}\}$  and  $B = \{B_0, B_1, \dots, B_{n-1}\}$ ; whereas, the sum, carry input and carry output are expressed as  $S = \{S_0, S_1, \dots, S_{n-1}\}$ ,  $C_{in}$  and  $C_{out}$  respectively. Gate-level circuit

**Organized by**

**Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India**

diagram of the speculator used in our adder design is presented. This block is based on CLA logic to speculate the output carry for each 4-bit adder block. Speculation is carried out for  $r$  msb bits of each block where  $r$  is less than the size of block, (i.e.,  $r < x = 4$ ). Subsequently, the input carry for each speculator block is 0 (or 1) which introduces positive (or negative) errors respectively. The output carry, which is denoted as  $\mathcal{C}so$ , from each speculator block is fed as an input carry for the adder block succeeding it. Now, each 4-bit adder block need not wait for the input carry from the preceding 4-bit adder block. Instead, all such adder blocks perform simultaneous additions on receiving input carries from the concerned speculator blocks.

### B. Compensator Block

The digital architecture for compensator block used in the proposed ISA adder. This block compares the output carry from each 4-bit adder block with the corresponding speculated carry using a XOR gate. Thereafter, the output from XOR gate generates an error flag ( $f_e$ ) that triggers the activation of one of the two compensation techniques: error correction and reduction. If the XOR-gate output is 0 then the local sum is directly passed to the final output. Similarly, if the XOR gate gives, 1 then this indicates that an error has occurred which can be either positive or negative. A positive error indicates a speculation of instead of 1 and hence, induces too low sum. Albeit a negative error indicates a speculation of, 1 instead of, 0 which induces too high sum.

The compensation block performs an unsigned increment or decrement to the group of LSBs in the direction of this potential error (too high error is solved by a -1 and too low by a +1). This correction is possible only if it does not result overflow. In case of overflow, the compensation block balances a group of MSBs of the preceding sub-adder in the opposite direction of the error.

In general, if the number of bits used for correction is  $p$  then the first computed  $p$  LSBs from the 4-bit adder block are passed on to the compensation block where it is checked whether any overflow occurs or not. It does then the right compensation technique which is balancing will be selected. All this is carried out prior the 4-bit adder block finishes computing the sum of all other bits. Preferably the value of  $p$  is 1 for the optimum results. Thus, a significant feature of this adder is that neither the pre-computing of error correction nor the compensation choice lies in the critical path of the ISA adder. The components of compensation block which are involved in the overall critical path of ISA are the XOR gate, de-multiplexer and multiplexer.

### C. Fine-Grain Pipelined Architecture

In the traditional ISA structure, allow us to expect that the combinational postpone of 4-bit adder, speculator and compensator blocks to be  $\partial 4b\text{-adder}$ ,  $\partial spec$  and  $\partial comp$  respectively. In this structure, carry in is speculated for each 4-bit adder block and based totally on this; adder block calculates the local sum. There after, the faulty speculation is detected by comparing speculated carry-in and prior carry-out from 4-bit adder. Subsequently, compensator block performs the correction and balancing operation. Thus, the critical path of the conventional ISA architecture includes delays of speculator of the  $i$ th instant and the 4-bit adder plus compensator delays of  $(i+1)$ th instant, as shown, (with coloured lines and blocks) For the ease of understanding, pipelining process of this work has been explained using  $n = 16$  bit ISA architecture. Even though the value of  $n$  increases, critical path delay is unaffected because the value of  $x$  is always 4 bit (as discussed earlier) and the adder, speculator as well as compensator.

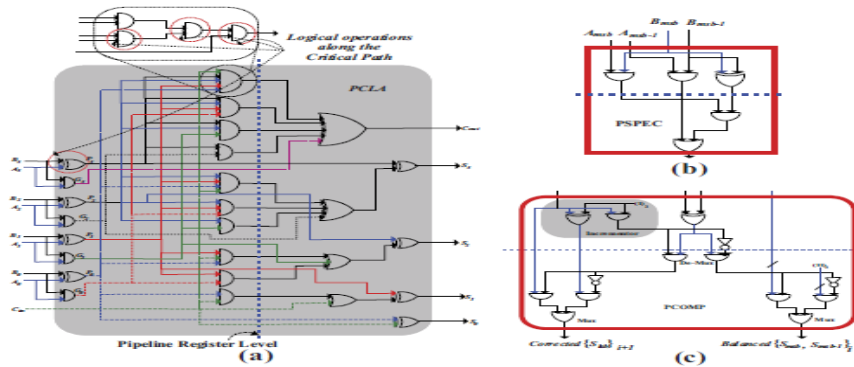


Fig 5: Gate-level circuit of (a) Four-bit pipelined-carry look-ahead adder (PCLA) (b) Pipelined speculator (PSPEC) (c) Pipelined compensator (PCOMP) used in the proposed ISA VLSI architecture

Sub blocks PSPEC, PCLA and PCOMP contain two pipelined stages. Overall architecture of the suggested ISA adder has been designed with five pipelined stages and there are six levels of registers included in this design, as shown in Figure. This is a scalable architecture because the number of pipeline stages remains constant on increasing the bit widths of the operands, retaining the same critical path delay. The deep pipelined architectures of sub block have been illustrated in Figure.. It shows the gate-level designs of PSPEC, PCOMP, PCLA and their respective pipelined stages. On observing the proposed VLSI architectures, it can be seen that the critical path of suggested architecture lies in PCLA and it includes only four two- input gate delays (one XOR and three AND gate delays).

**D. Proposed Pipelined Architecture for In-exact speculative adder using Brent Kung Adder**

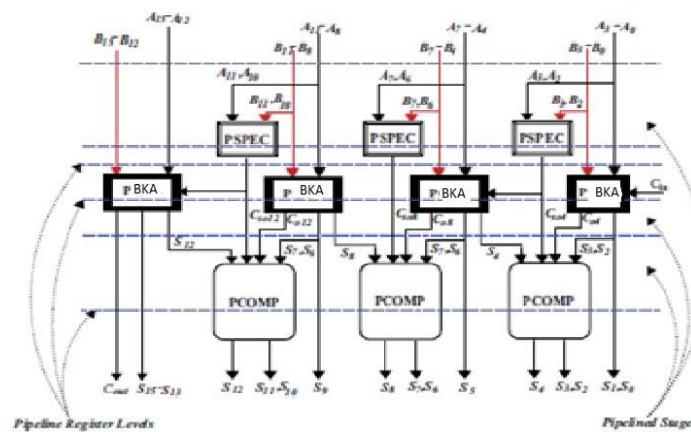


Fig.6 Deep-pipelined VLSI architecture of the proposed ISA for  $n = 16$  bits and  $x = 4$  bits, with five pipeline stages, for high speed applications

In the proposed architecture the CLA unit has been replaced with 4-bit Brent Kung adder to further enhance the speed of operation. The Brent-Kung adder is a parallel prefix adder. Parallel prefix adders are special class of adders that are based on the use of generate and propagate signals. Compared to CLA the propagation delay is the reduced using Brent Kung Adder. Thereby, the design of highly optimized adder in terms of speed play significant role in the present era and hence this project focuses in the design of same as CLA. The methodology implemented using Brent Kung adder is similar as CLA and the only difference is that the parallel prefix CLA block is replaced with parallel prefix Brent kung adder.

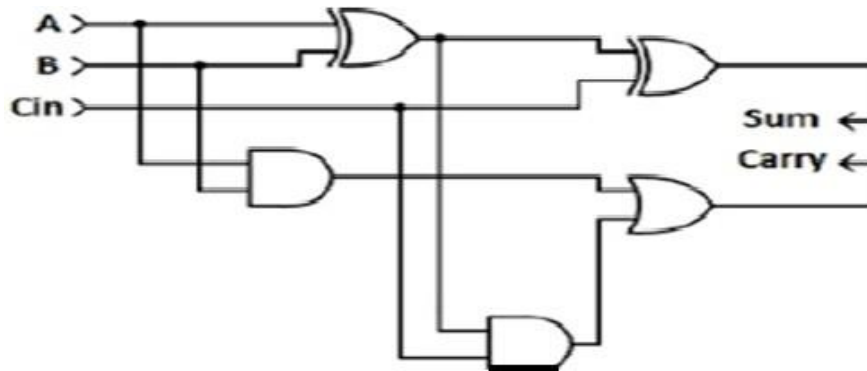


Fig.7 Gate level block of Brent Kung Adder

#### IV. CONCLUSION

In this paper, we presented high-speed and low-power version of the contemporary ISA design. Experimental results showed that the suggested ISA could operate in Xilinx version. Thereby, such design would definitely play significant role in the design of contemporary as well as future electronic devices for IoE and many other contemporary applications. However, the area issue can be resolved to some extent by using lower technology nodes in the design process.

#### REFERENCES

- [1]B. Moons et al., "An energy-efficient precision-scalable convnet processor in 40nm CMOS," in Journal of Solid-State Circuits (JSSC), IEEE, 2018.
- [2]E. Noguez et al., "A DVFS based HEVC decoder for energy-efficient software implementation on embedded processors," in International Conference on Multimedia and Expo (ICME), 2015 IEEE, 2017.
- [3]V. Camus et al., "Energy-efficient inexact speculative adder with high performance and accuracy control," in Circuits and Systems (ISCAS), 2015 IEEE International Symposium, 2017, pp. 45–48.
- [4]X. Jiao et al., "Supervised learning based model for predicting variability-induced timing errors," in New Circuits and Systems Conference (NEWCAS), 2015 IEEE 13th International, 2016, pp. 1–4.
- [5]R. Ragavan et al., "Adaptive overclocking and error correction based on dynamic speculation window," in Symposium on VLSI (ISVLSI), 2015 IEEE Computer Society Annual, 2016, pp. 325–330.
- [6]X. Jiao et al., "Wild: A workload-based learning model to predict dynamic delay of functional units," in Computer Design (ICCD), 2016 IEEE 34th International Conference on, 2016
- [7]G. Dimitrakopoulos and D. Nikolos, "High-speed parallel-prefix VLSI ling adders", IEEE Trans. Comput., ol. 54, no. 2, Feb.2005, Pages. 225–231.
- [8]V. Dave, E. Oruklu, and J. Saniie, "Performance evaluation of flagged prefix adders for constant addition," in Proc. IEEE Int. Conf. Electro/ inf. Technol., 2006, Pages 415–420.
- [9]Geeta Rani , Sachin Kumar, "Delay Analysis of Parallel-Prefix Adders", International Journal of Science and Research, Volume 3 Issue 6, June 2014, Pages 2339-2342.