

Pipelined VLSI Architecture for In-Exact Speculative Adder Using Carry Look Ahead Adder and Brent Kung Adder

Mehnaz Begum

Department of Electronics & Communication
Engineering,
HITAM, Hyderabad, Telangana-502401, India.

K. Anil Kumar

Department of Electronics & Communication
Engineering,
HITAM, Hyderabad, Telangana-502401, India.

Abstract

This paper presents the design of a novel architecture of a contemporary Inexact speculative adder with optimized hardware efficiency and advanced compensation technique with either error correction or error reduction which is fine grain pipelined to include few logic gates along its carry propagation chain which is considered 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 by splitting the critical path into two or more shorter paths, reducing spurious glitching power and error 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 general topology of speculative adders improves performance and enables 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.

INTRODUCTION

High-speed adders are highly desirable in the present day scenario, though power (or energy) and silicon area are equally vital. Spectrum sensors used in intelligent cognitive-radio environment as well as internet of everything (IoE) devices focused on physical interfaces are largely-explored research areas in the recent time. Hardware for the algorithms of such applications is basically focused on sensing and actuating where the response time is key component to be optimized for real-time interfaces. Thereby, the design of highly optimized adders in terms of speed play significant role in the present era and hence this paper focuses in the design of same. With tolerable degradation in accuracy and performance, it is feasible to conceive high-speed, low power and area efficient design using inexact and approximate circuit technique. Accuracy of such circuits can be traded off to improve the power and speed by speculation.

Thereby, such adders are referred as inexact speculative adder (ISA). Various optimized versions of such ISA have been reported in literature and these works concentrated mostly on enhancing the accuracy of their results. However, there is space to further improve the speed of such adders by retaining the accuracy with minimum error. Thereby, our contributions in this work are as follows: design and analysis of the carry lookahead adder (CLA) based ISA has been carried out. Thereafter, this adder is fine grain pipelined to reduce

Cite this article as: Mehnaz Begum & K. Anil Kumar, "Pipelined VLSI Architecture for In-Exact Speculative Adder Using Carry Look Ahead Adder and Brent Kung Adder", International Journal & Magazine of Engineering, Technology, Management and Research, Volume 6 Issue 1, 2018, Page 33-42.

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. Subsequently, clock signal fed to various stages of the deep pipelined ISA-architecture has been gated to reduce the power consumption. Eventually, ASIC synthesis and post-layout simulation of the proposed 32-bit ISA has been performed in 90 nm-CMOS technology node and is compared with the state-of-the-art ISA adder.

BRENT KUNG ADDER

The type of structure of any adder greatly affects the speed of the circuit. The logarithmic structure is considered to be one of the fastest structures. The logarithmic concept is used to combine its operands in a tree- like fashion. The logarithmic delay is obtained by restructuring the look-ahead adder. The restructuring is dependent on the associative property, and the delay is obtained to be equal to $(\log_2 N) t$, where ‘N’ is the number of input bits to the adder and t is the propagation delay time.

Hence, for a 16-bit structure, the logarithmic adder has a delay equal to ‘4t’, while for a simple ripple carry adder the delay is given by $(N-1)t$ and is equal to ‘15t’ for ‘N’ and ‘t’ being the number of input bits and the delay time, respectively. Hence it is seen that this structure greatly reduces the delay, and would be especially beneficial for a structure with large number of inputs. This advantage is, however, obtained at the expense of large area and a complex structure.

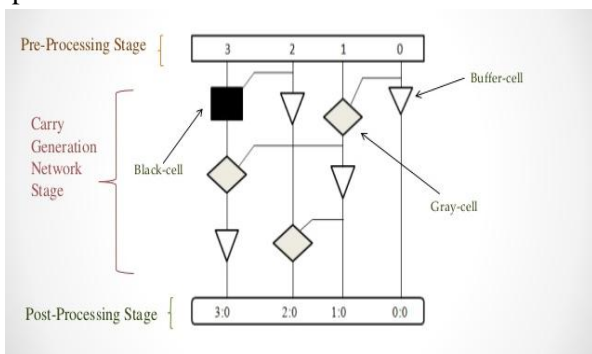


Figure 1: 4-Bit Brent Kung Adder

$$C_{0,0} = G_0 + P_0 C_{i,0} = a(G_0, P_0) \quad C_{0,1} = G_1 + G_0 P_1 = a((G_1, P_1), (G_0, P_0)) \dots \dots \dots C_{0,k} = a(((G_k, P_k) \cdot (G_{k-1}, P_{k-1}) \cdot \dots \cdot (G_0, P_0)) (3.4))$$

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 $C_{0,0}, C_{0,1}, C_{0,3}$ and $C_{0,7}$. The remaining carry signals are generated by the reverse binary tree.

Brent-Kung Implementation:

The Brent-Kung tree computes prefixes for 2-bit groups. These are used to find prefixes for 4-bit groups, which in turn are used to find prefixes for 8-bit groups, and so forth. The prefixes then fan back down to compute the carries-in to each bit. The tree requires $2\log_2 N - 1$ stages. The fan-out is limited to 2 at each stage.

The diagram shows buffers used to minimize the fan-out and loading on the gates, but, in practice, the buffers are generally omitted. The basic blocks used in this case are gray and black cells which are explained. This adder is implemented for 8 bit using CMOS logic and transmission gate logic.

Brent-Kung Carry Network (8-Bit Adder)

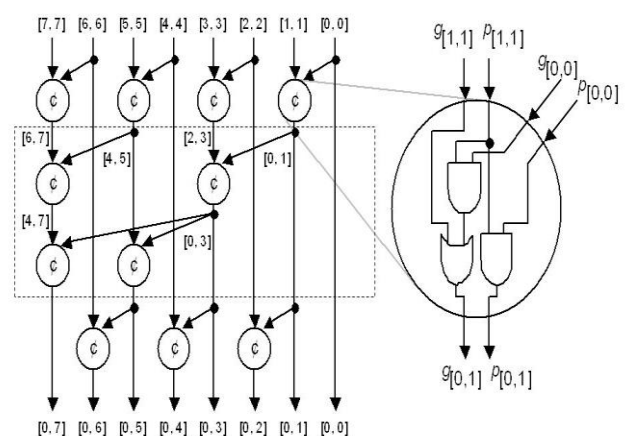


Figure 2: 8-Bit Brent Kung Network

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. Simpler Brent-Kung adders have been proposed to solve the disadvantages of Kogge-Stone adders.

The cost and wiring complexity is greatly reduced. But the logic depth of Brent-Kung adders increases to $2\log(2n-1)$, so the speed is lower.

We propose a method to reduce delay and power consumed by the Brent Kung adder by analyzing and dividing the inputted data into blocks that will only be added if it holds any value at all. Hence the inputted values are initially compared before deciding up to how many of the adder blocks should be activated.

This approach can easily be integrated to the existing design of the Brent Kung adder, thus making it more efficient. The block diagram of 16-bit Brent-Kung adder is show

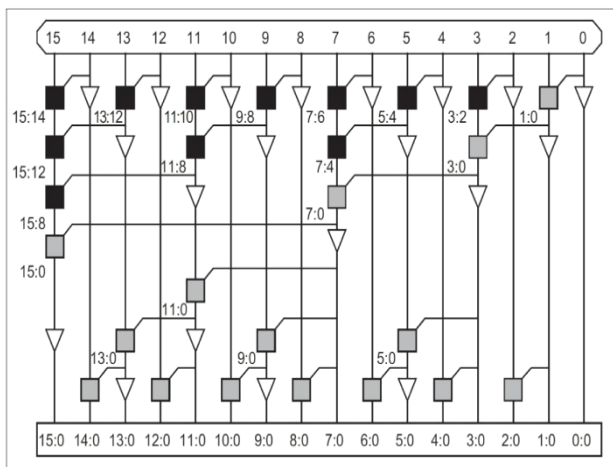


Figure 3: Block Diagram of Brent Kung Adder

PROPOSED SYSTEM

Block diagram and data flow of conventional ISA for n -bit addition is shown in Figure. In the proposed architecture, we have segregated the n -bit input into 4-bit blocks (i.e., the value of $x = 4$ in Figure and each 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:

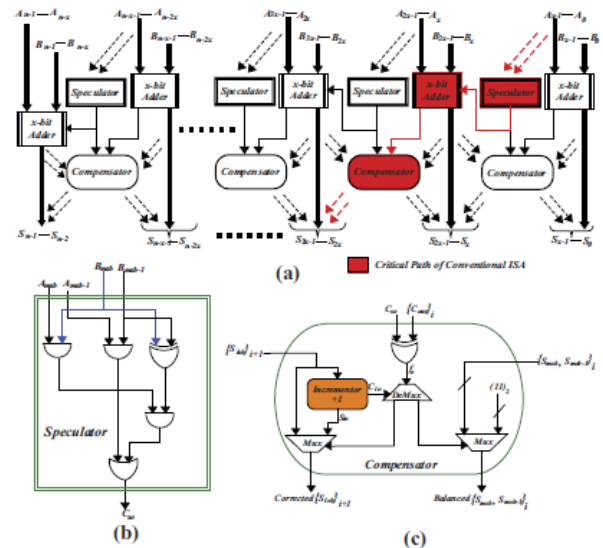


Figure 4: (a) Basic block diagram of n -bit conventional In-exact speculation adder (ISA). (b) Gate-level circuit representation of speculator block. (c) Digital architecture of compensator block

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 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 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.

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 '0' 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.

Fine-Grain Pipelined Architecture:

In the conventional ISA architecture, let us assume that the combinational delay of 4-bit adder, speculator and compensator blocks to be $\partial 4b\text{-adder}$, $\partial spec$ and $\partial comp$ respectively. In this architecture, carryin is speculated for each 4-bit adder block and based on this; adder block calculates the local sum. Thereafter, 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 architectures remain unchanged.

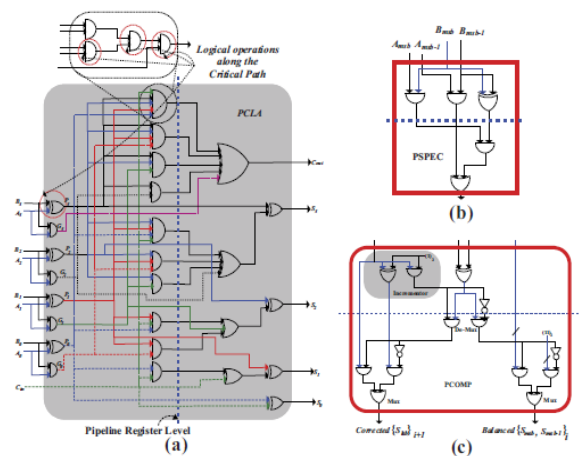


Figure 5: Gate-level circuit of (a) Four-bit pipelined-carry look-ahead adder (PCLA) (b) Pipelined speculator (PSPEC) (c) Pipelined compensator (PCOMP) used in the proposed ISA VLSI architecture

The proposed 16-bit ISA VLSI-architecture is shown in Figure, where the conventional blocks has been replaced by the pipelined speculator (PSPEC), pipelined compensator (PCOMP) and pipelined 4-bit CLA (PCLA)

units. 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).

Proposed Pipelined Architecture for In-exact speculative adder using Brent Kung Adder:

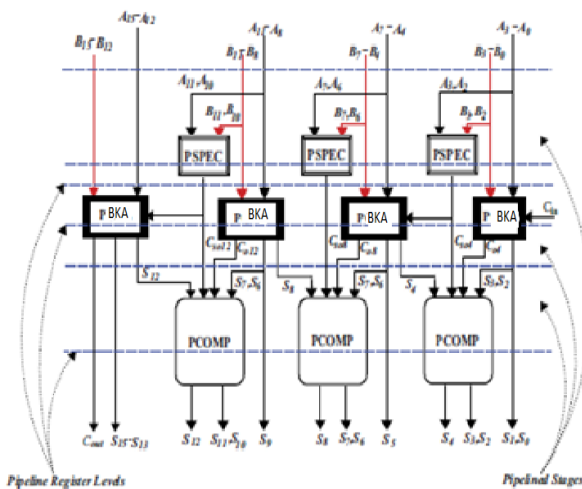


Figure 6: Deep-pipelined VLSI architecture of the proposed ISA for $n = 16$ bits and $x = 4$ bits, with five pipeline stages, for high speed applications

In the proposed architecture the adder 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.

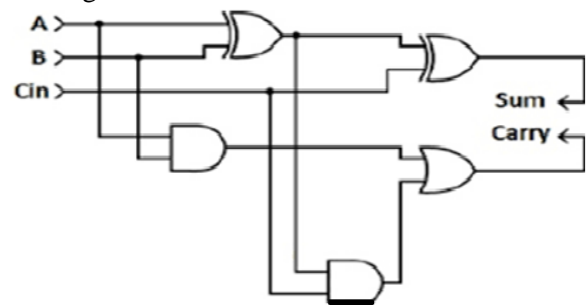


Figure 7: Gate level block of Brent Kung Adder

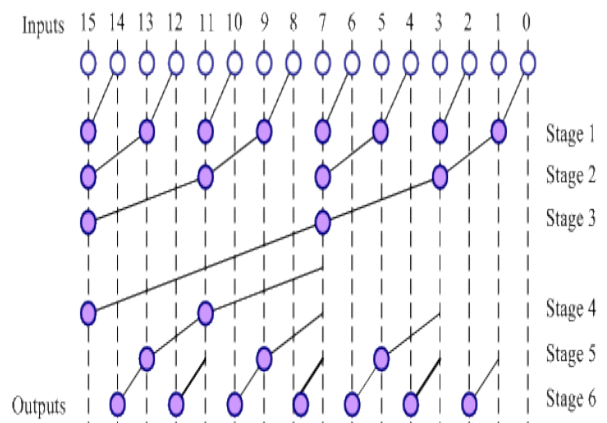


Figure 8: Working of 16-bit Brent Kung Adder

IMPLEMENTATION

Flow chart of implemented adders:

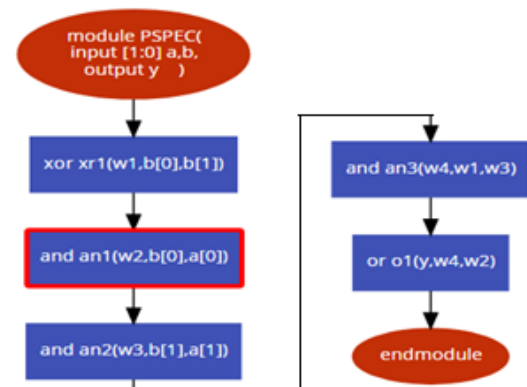


Figure 8: Flow chart of speculator block

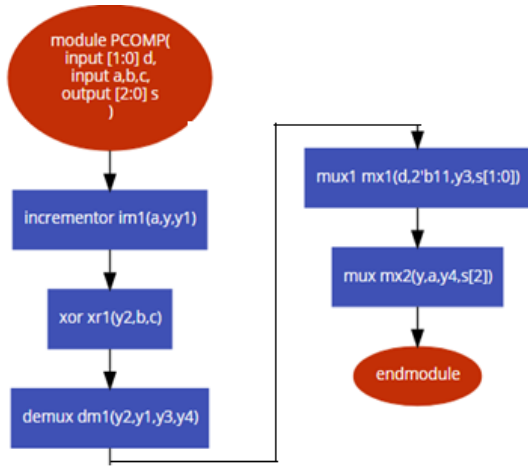


Figure 9: Flow chart of compensator block

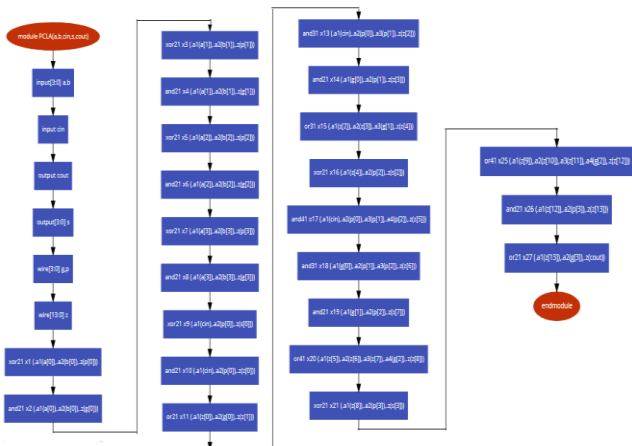


Figure 10: Flow chart of 4-bit carry look ahead adder block

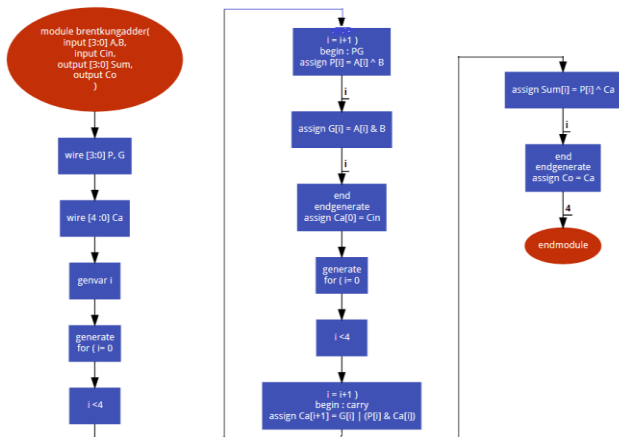


Figure 11: Flow chart of 4-bit Brent kung adder block

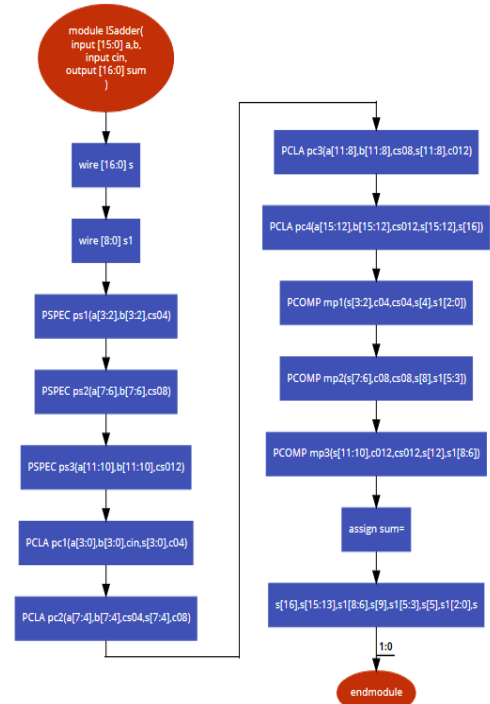


Figure 12: Flow chart of 16-bit pipelined architecture of in-exact speculative adder using CLA



Figure 13: Flow chart of 16-bit pipelined architecture of in-exact speculative adder using Brent kung adder

RESULT ANALYSIS

Input/output waveform simulation of implemented 16-bit pipelined in-exact speculative adder using Carry look ahead adder:

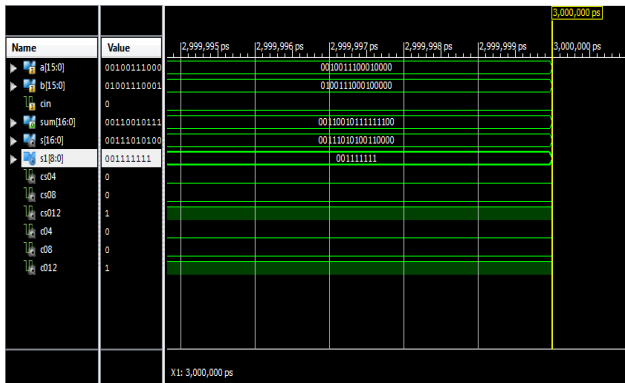


Figure 14: Simulation of implemented adder based binary values

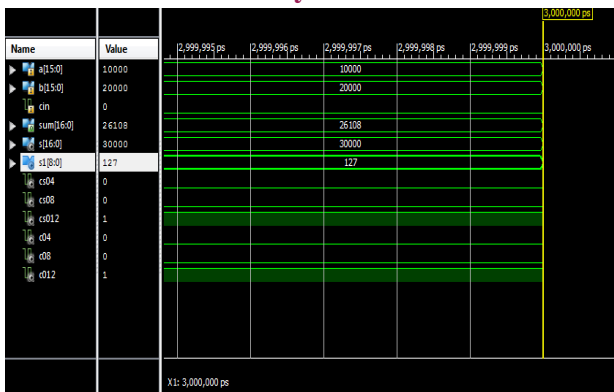


Figure 15: Simulation of implemented adder based unsigned values

Input/output waveform simulation of implemented 16-bit pipelined in-exact speculative adder using Brent Kung adder:



Figure 16: Simulation of implemented adder based binary values

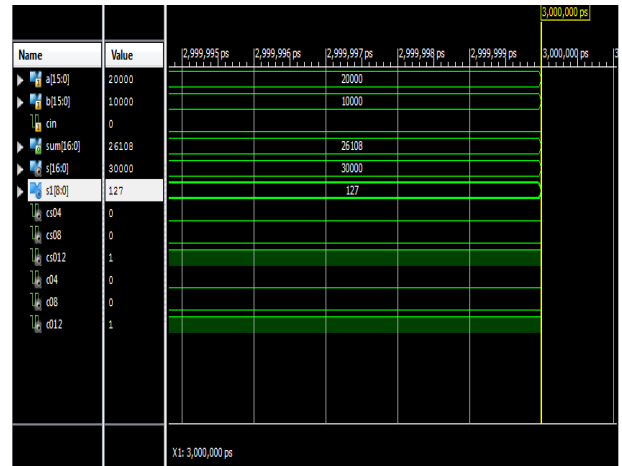


Figure 17: Simulation of implemented adder based unsigned values

Synthesis result:

The proposed project is simulated and verified their functionality. Once the functional verification is done, the RTL model is taken to the synthesis process using the Xilinx ISE tool. In synthesis process, the RTL model will be converted to the gate level net list mapped to a specific technology library. Here in this ISE project navigator of Spartan 3E family, many different devices were available in the Xilinx ISE tool. In order to synthesis this design the device named as “XC3S500E” has been chosen and the package as “FG320”.

RTL schematic:

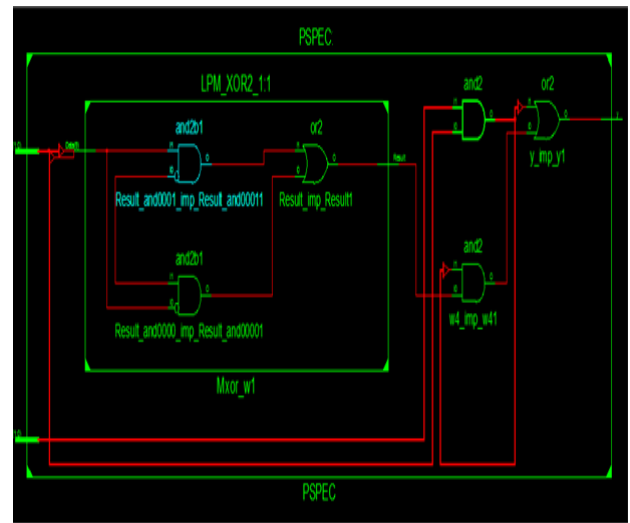


Figure 18: Schematic diagram of 2-bit Speculator block

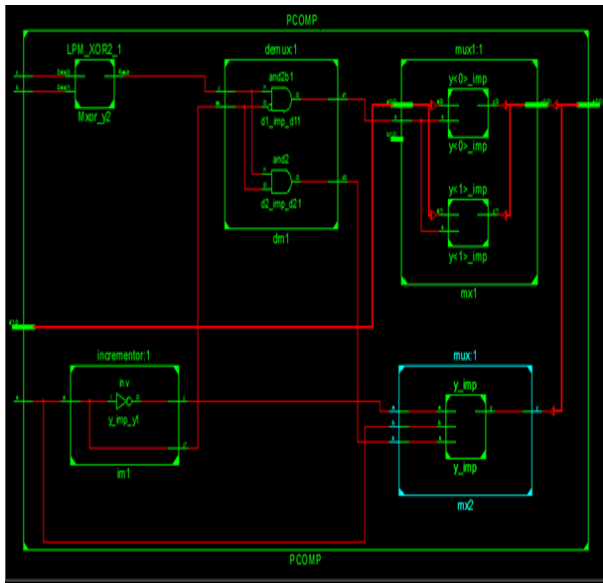


Figure 19: Schematic diagram of 2-bit Compensator block

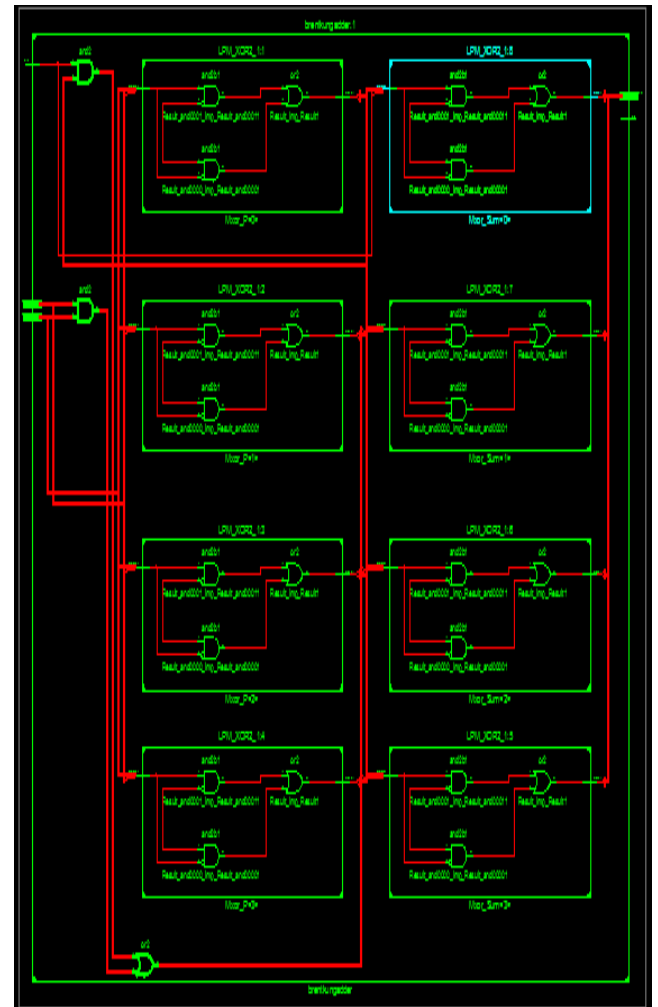


Figure 21: Schematic diagram of 4-bit Brent Kung adder

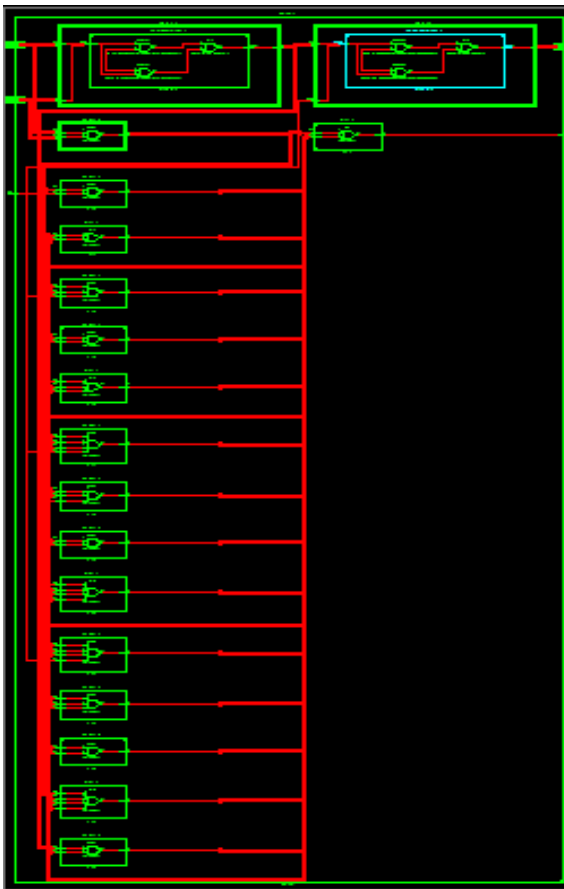


Figure 20: Schematic diagram of 4-bit Carry look ahead adder

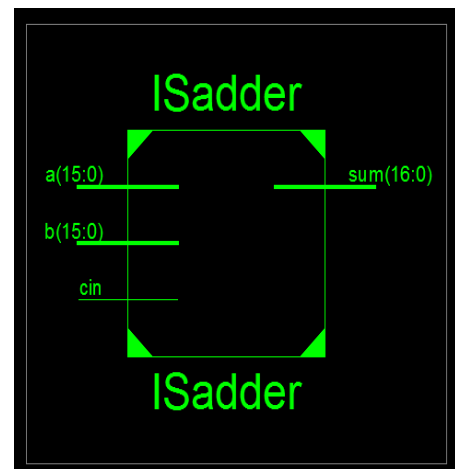


Figure 22: RTL block of implemented adder using CLA

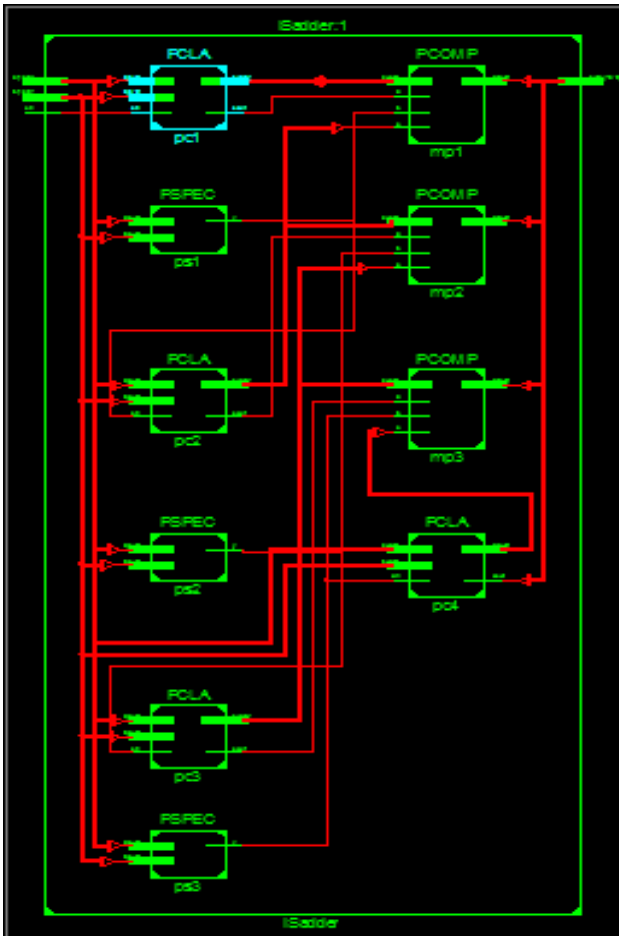


Figure 23: Schematic diagram of 16-bit pipelined Inexact Speculative adder using carry look ahead adder

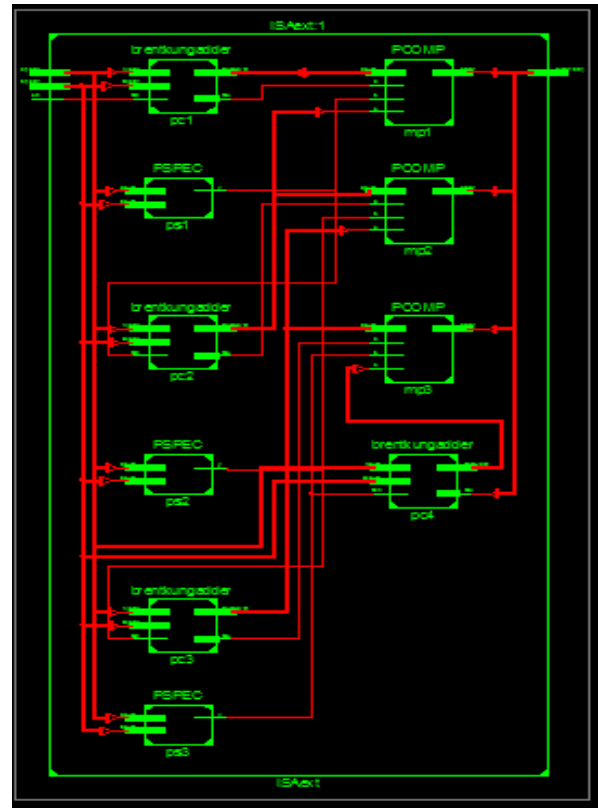


Figure 25: Schematic diagram of 16-bit pipelined Inexact Speculative adder using Brent Kung adder

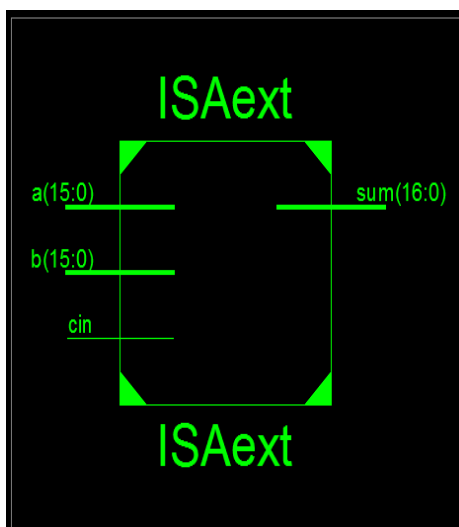


Figure 24: RTL block of implemented adder using Brent kung adder

CONCLUSION

In this paper, we presented high-speed and low-power version of the contemporary ISA design. This architecture has been fine grain pipelined and clock gated to escalate speed and alleviate power consumption respectively. 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]. High-Speed and Low-Power VLSI-Architecture for Inexact Speculative Adder, Rahul Shrestha, Member, IEEE. School of Computing & Electrical Engineering, Indian Institute of Technology (IIT) Mandi.

[2]. Energy-Efficient Inexact Speculative Adder with High Performance and Accuracy Control, Vincent Camus, Jeremy Schlachter, Christian Enz Integrated Circuits Laboratory (ICLAB)Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland.

[3]. Energy-Efficient Digital Design through Inexact and Approximate Arithmetic Circuits, Vincent Camus, Jeremy Schlachter, Christian Enz Integrated Circuits Laboratory (ICLAB)Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland.

[4]. An Enhanced Low-power High-speed Adder For Error-tolerant Application ,N. Zhu, W.-L. Goh, and K.-S. Yeo, in Integrated Circuits (ISIC), Proc. of the 2009 12th International Symposium on, Dec 2009, pp. 69–72.

[5]. Balancing Adder for Error Tolerant Applications, M. Weber, M. Putic, H. Zhang, J. Lach, and J. Huang, in Circuits and Systems (ISCAS), 2013 IEEE International Symposium on, May 2013, pp. 3038–3041.

[6]. Performance Improvement with Circuit-level Speculation, T. Liu and S. L. Lu, “33rd Annual IEEE ACM International Symposium on Micro-architecture(MICRO-33), pp. 348-355, 2000.

[7]. A New Approximate Adder with Low Relative Error and Correct Sign Calculation, J. Hu and W. Qian, in Design, Automation and Test in Europe(DATE), 2015 IEEE Conference and Exhibition on, March 2015.