IJARST

International Journal of Advanced Research in Science and Technology

https://doi.org/10.62226/ijarst20241357

ISSN 2319 – 1783 (Print)
ISSN 2320 – 1126 (Online)

4 BIT FLASH ADC DESIGNED BY CMOS AND PSEUDO NMOS LOGIC WITH 0.18 NM TECHNOLOGY

T Kamala Kumara[1], R Vinay Kumar[2]
[1]M.Tech Student, Dept of ECE International School of Technology & Sciences, (Approved by AICTE & Affiliated to JNTUK Kakinada), Rajahmundry, India
[2]Assistant.Professor, Dept of ECE International School of Technology & Sciences, (Approved by AICTE & Affiliated to JNTUK Kakinada), Rajahmundry, India

ARTICLEINFO

Article history:

Received	10 Apr 2024
Accepted	16 Apr 2024
Available online	19 Apr 2024

ABSTRACT

Approximate computing is an efficient approach for error-tolerant applications because it can trade off accuracy for power. Addition is a key fundamental function for these applications. We proposed a low-power yet high speed accuracy-configurable adder that also maintains a small design area. The proposed adder is based on the conventional carry look-ahead adder, and its configurability of accuracy is realized by masking the carry propagation at runtime. Compared with the conventional carry look-ahead adder, with only 14.5% area overhead, the proposed 16-bit adder reduced power consumption by 42.7% and critical path delay by 56.9% most according to the accuracy configuration settings, respectively. Furthermore, compared with other previously studied adders, the experimental results demonstrate that the proposed adder achieved the original purpose of optimizing both power and speed simultaneously without reducing the accuracy. Inexact (or approximate) computing is an attractive paradigm for digital processing at nanometric scales. Inexact computing is particularly interesting for computer arithmetic designs.

INTRODUCTION

Digital logic circuits are used in the majority of computer arithmetic applications, enabling very precise and reliable operations. Many applications, such as multimedia and image processing, can, nevertheless, withstand computational errors and imprecision and still yield meaningful and practical results. It is not always appropriate or efficient to utilize accurate and precise models and algorithms in these kinds of applications. To develop energy-efficient systems, for example, the paradigm of inexact computation relies on relaxing accurate and perfectly deterministic building components. This enables imprecise computation to take advantage of a potential boost in performance and power efficiency along with a drop in complexity and cost, rerouting the current design process of digital circuits and systems. Using this feature, approximate (or inexact) computing relies on designing simpler, yet approximate circuits that perform better and/or use less power than precise (exact) logic circuits. In computer arithmetic, addition and multiplication are frequently performed operations. In the case of addition, full-adder cells have been thoroughly examined for approximate computing, allowing for comparisons between these adders and the proposal of multiple new metrics for assessing approximate and probabilistic adders in terms of unified figures of merit for inexact computing application design evaluation.

Recent advancements in various applications, such as image recognition, synthesis, and computationally intensive digital signal processing, as well as the development of wearable devices reliant on battery power, pose significant challenges related to power consumption. Among the fundamental arithmetic functions crucial for these applications, addition stands out prominently. Many of these applications exhibit a natural tolerance for minor inaccuracies, further complicating the issue. Approximate computation can be used to balance power and accuracy by taking advantage of the built-in tolerance capability. This trade-off is currently important in several application fields. It is better to create high-quality adjustable systems that may trade off computing effort and quality in accordance with application needs, since the computation quality requirements of an application can change dramatically during runtime. Previous approaches to configurability have encountered drawbacks such as heightened power consumption or increased latency. To address the needs of applications in this context, there is a pressing demand for a low-power, high-speed adder capable of configurable approximation. Consequently, we introduce a configurable approximate adder designed to minimize power consumption while maintaining comparable levels of delay and area utilization. Furthermore, the suggested adder's delay is significantly less than that of adders with equivalent power usage. Our main contribution is that the suggested adder optimized power and delay simultaneously and without favoring one over the other in order to attain accuracy and configurability. Using a

45-nm library, we developed the proposed adder, the ripple carry adder (RCA), and the traditional carry look-ahead adder (CLA) in Verilog HDL. Next, we assessed each of these implementations' power consumption, critical path delay, and design area. With a mean relative error distance (MRED) of 1.95%, the suggested adder lowered critical path delay and power consumption by 56.9% and 42.7%, respectively, in comparison to the traditional CLA. We presented a cross-wise comparison to show that the suggested adder is superior. In addition, we put into practice two customizable adders that had been previously examined in order to assess accuracy, design area, power consumption, and critical path latency. We assessed these precision customizable adders' quality in a real-world image processing application as well.

MOTIVATION

The incentive behind our project is the day-to-day increase in the demand for low power and high speed in INTEGRATING CIRCUITS for manyapplications such as Multimedia & Image, Data Processing, AI & Machine Learning, and many more where results do not need the highest accuracy which leads to the development of Approximate Computing.

PROBLEM DEFINITION

Nowadays, People depend on applications for almost everything and they want faster response out of it. However, every application needs some time for its operation. For example, Applications which are recently emerged in the computational demands of tasks like image recognition, synthesis, and Digital Signal Processing have posed significant challenges regarding power consumption. Many of these applications exhibit a natural capacity to accommodate minor inaccuracies. Utilizing this characteristic, Approximate Computing emerges as an efficient strategy for applications tolerant to errors, as it allows for a tradeoff between accuracy and power consumption.

LITERATURE SURVEY

Adders are the primary component of numerous circuits, such as microprocessor designs. Critical operations carried out by adders include comparison, incrementing and decrementing, and arithmetic (addition, subtraction, and multiplication). As a barrier to additional advancements in the functionality of microprocessors and other general- and specific-purpose circuits, adders are of interest.

Approximation is acknowledged as offering a pathway to enhance performance or diminish power consumption by employing simplified or imprecise circuits in scenarios where stringent demands are relaxed. The adoption of approximate design facilitates a balancing act between computation accuracy and performance and/or power efficiency. Many approximate arithmetic designs that offer a fixed approximation to accomplish a predetermined performance have been put forth. The error rate provided by the approximation is designed to be an acceptable penalty. The penalty rate is designed so that rectification still yields a net speed/computation gain if error detection and repair is applied.

Gupta et al. [6] explored methods for reducing the intricacy of a traditional mirror adder cell at the transistor level were discussed. Mahdiani et al. [7] suggested a lower-part-OR adder that adds bits by using precision adders for the upper bits and OR gates for the lower bits. Venkatesan et al. [8] proposed building a comparable untimed circuit to simulate the behavior of an approximative circuit. Miao et al. [9] presented a fixed internal-carry structure that was aligned, and after that, a dithering approximate adder that traded off error frequency and magnitude was suggested. Du et al. [10] developed a dependable variable latency speculative carry choose adder for error detection and result recovery. In actuality, an application's compute quality requirements could change dramatically during runtime. The aforementioned static approximate designs with fixed precision could not be able to meet the applications' quality criteria or waste energy when high quality is not needed. Kahng et al. [4] suggested a pipeline-structured accuracy-configurable adder (ACA). The ACA's correction plan moves from stage 1 to stage 4, and all four steps should be completed if the most important aspects of the results must be accurate. Motivated by the above, Ye et al. [5] suggested a gracefully-degrading adder (GDA) for accuracy. Every subadder block, except the one on the right, has a carry-in prediction block, adder unit, and multiplexer of its own, as shown in Fig. 1, which permits the selection of the precise and approximate sums of its subadders at any random moment. Control signals can choose the carry-out signals in any sequence from the adder units or the carry-in prediction blocks. Similar to [5], the adder presented in this paper does not take into account a pipeline structure.

PROPOSED METHOD

This project introduces an accuracy-configurable adder that avoids the drawbacks of increased power consumption or delay associated with configurability. Built upon the conventional Carry Lookahead Adder (CLA), the proposed adder achieves accuracy configurability by dynamically masking the carry propagation during runtime.

The experimental results show that compared to the standard CLA, the suggested adder offers significant power savings and speedup with a minimal area overhead.

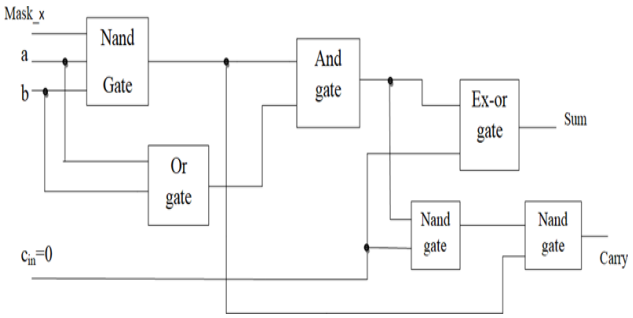

Fig.1 : CMFA Prototype

Mask_x	A	B	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Table.1 : Truth Table of CMFA

Ripple-Carry Adder

N-bit numbers can be added logically by building a circuit with numerous complete adders. Every complete adder inputs a Cin, which is the preceding adder's Cout.

The name ripple-carry adder (RCA) refers to this type of adder because each carry bit "ripples" to the next complete adder. Keep in mind that, provided Cin = 0, the first (and only the first) complete adder may be substituted with a half adder.

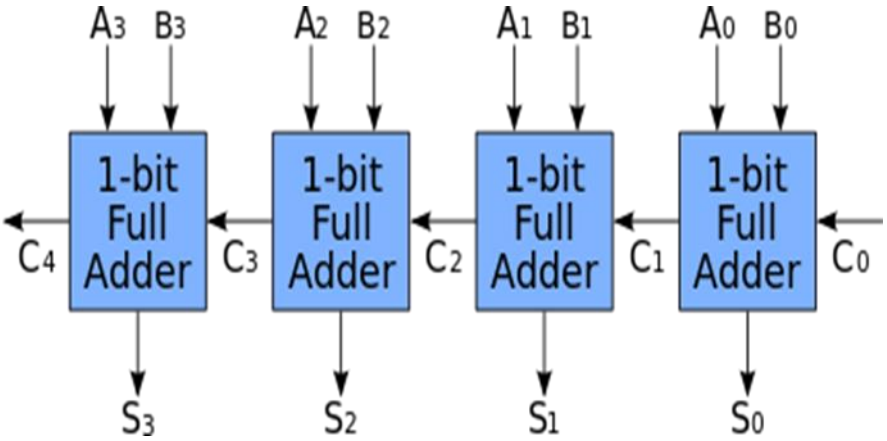

Fig.2 : 4-Bit Adder

Carry-Lookahead Adder

Engineers came up with quicker techniques to add two binary values utilizing carry-lookahead adders (CLA) to cut down on computation time. For every bit position, they generate two signals (P and G) according to whether a carry is generated in that bit position (both inputs are 1), propagated through from a less significant bit position (at least one input is a 1), or killed in that bit position (both inputs are 0). For the most part, P is just the half adder's sum output, and G is the adder's carry output. The carriers for each bit location are created after P and G are generated. The Brent–Kung adder (BKA), the Kogge–Stone adder (KSA), and the Manchester carry chain are a few examples of advanced carry–lookahead designs.

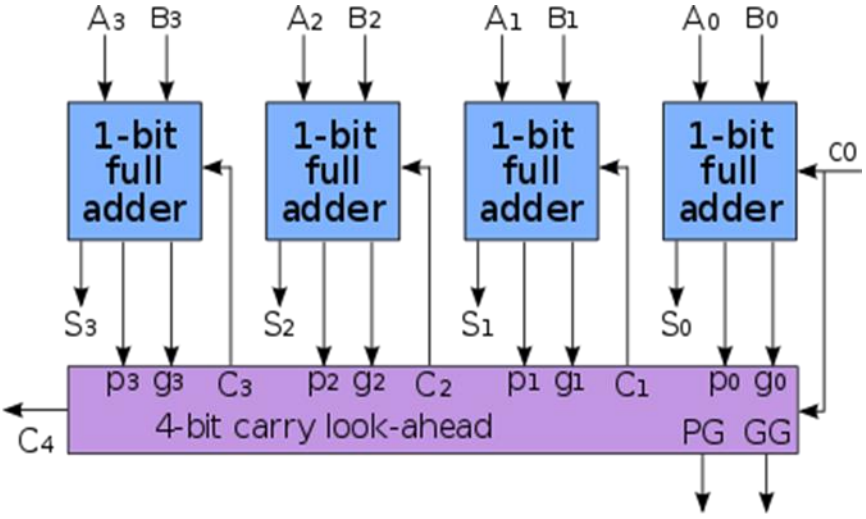

Fig.3: 4-bit Carry Look-Ahead Adder

Carry-Save Adders

It can be useful not to propagate the carry result in an adding circuit when computing the sum of three or more values. Rather, two outputs are produced: a total and a carry, using three-input adders. Without waiting for the carry signal to propagate, the total and the carry can be sent into two inputs of the ensuing 3-number adder. However, once all addition stages are complete, the final sum and carry values must be combined using a traditional adder (such the ripple-carry or the lookahead).

RESULTS AND SIMULATION

We implement and test the proposed adder, the standard CLA and RCA, and the previously researched configurable adder GDA [5] and [12] in order to elucidate the contributions to the power and latency of the proposed adder. These are all 16-bit adders. In GDA, the subadder units have a bit-length of four bits. In GDA, the carry-in prediction bit counts are 4, 8, and 12 bits. Since 16 bits are divided into four 4-bit-length groups by the proposed adder and adder [12], their setup parameters are identical, as is the case with GDA. Referred to as A_1, A_2, A_3, and A_4 (for the proposed adder), B_1, B_2, B_3, and B_4 (for adder [12]), and GDA1, GDA2, GDA3, and GDA4 (for GDA), accordingly, are the various configuration settings of these adders. For instance, A_1 indicates that CMHA3-0, CMHA7-4, and CMHA11-8 carry propagations are masked (M_X0 = M_X1 = M_X2 = 0). The same is true for adder [12] (B_1, B_2, B_3, and B_4).

Verilog HDL was employed to code all adders. Evaluation of the numerical outputs of the adders was conducted using Synopsys VCS, employing one million randomly generated input patterns, and generating Value Change Dump (VCD) files to precisely assess power consumption. Synopsys Design Compiler was utilized to synthesize the adders, employing the Nand Gate 45nm Open Cell Library [9]. An evaluation of the power usage was conducted at 0.5 GHz. Using the standard library conditions, all designs were synthesized and optimized using the default compile parameters. The power consumption from the switching activity interchange format files created from the VCD files was estimated using the Synopsys Power Compiler.

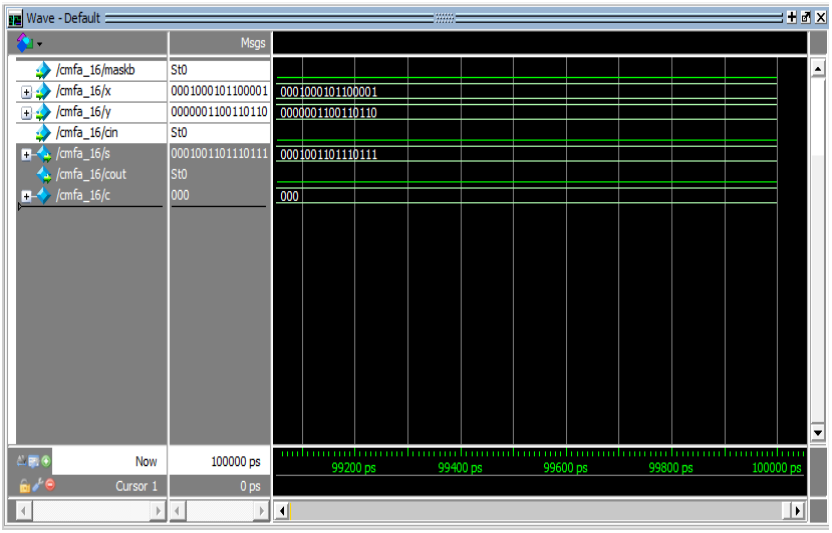

Fig: 4 When the mask input is 0

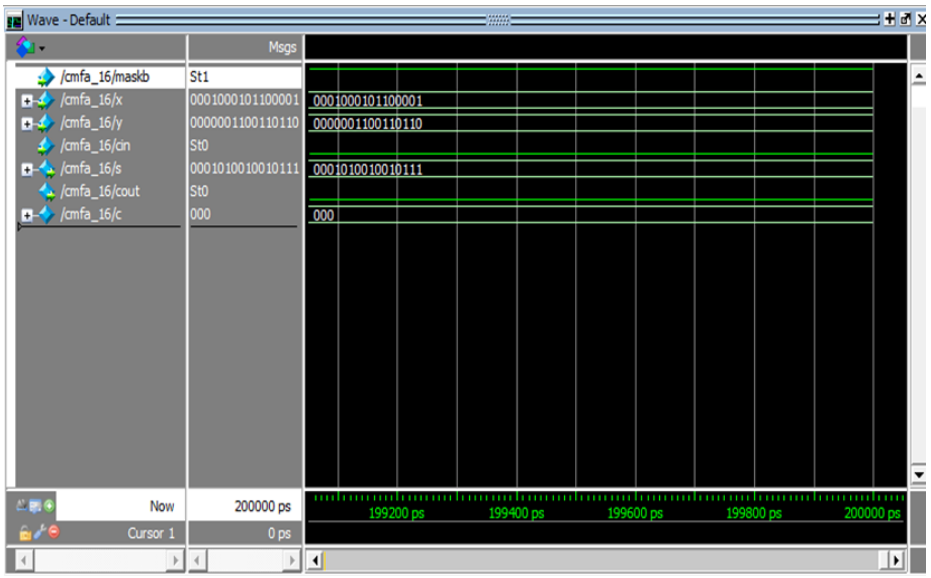

Fig: 5 When the mask input is 1

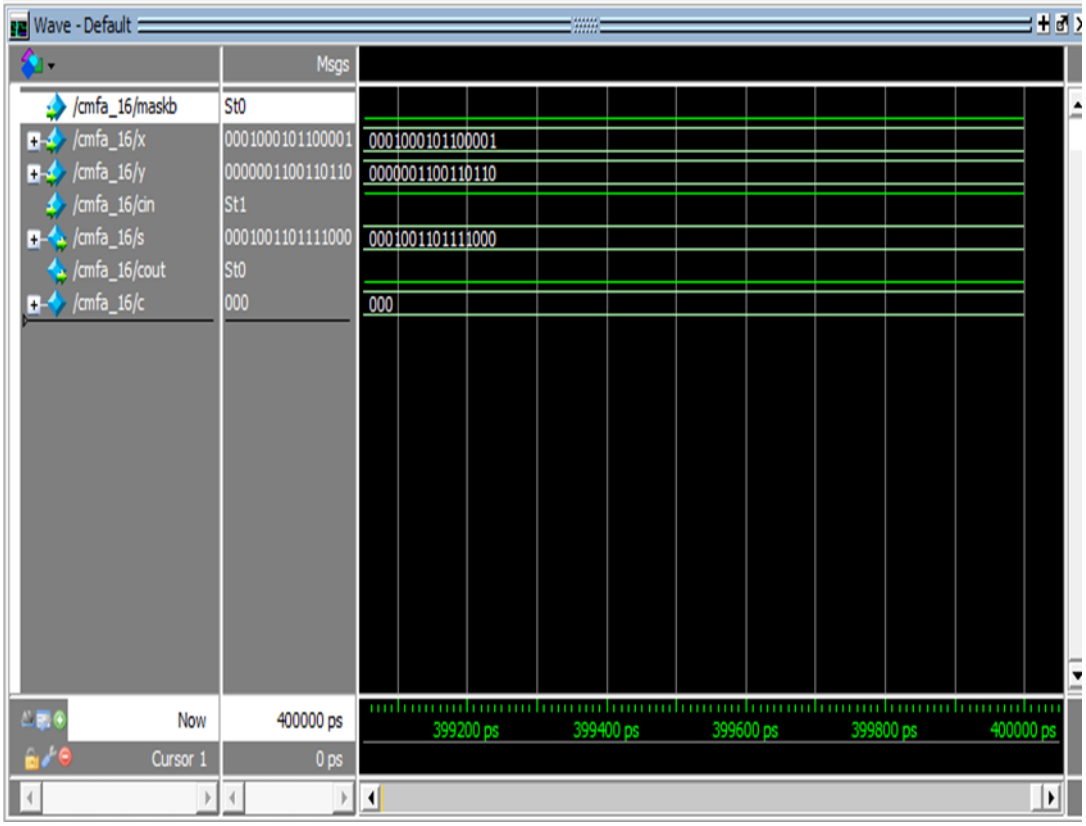

Fig: 6 Carry input=1 when mask input is 1

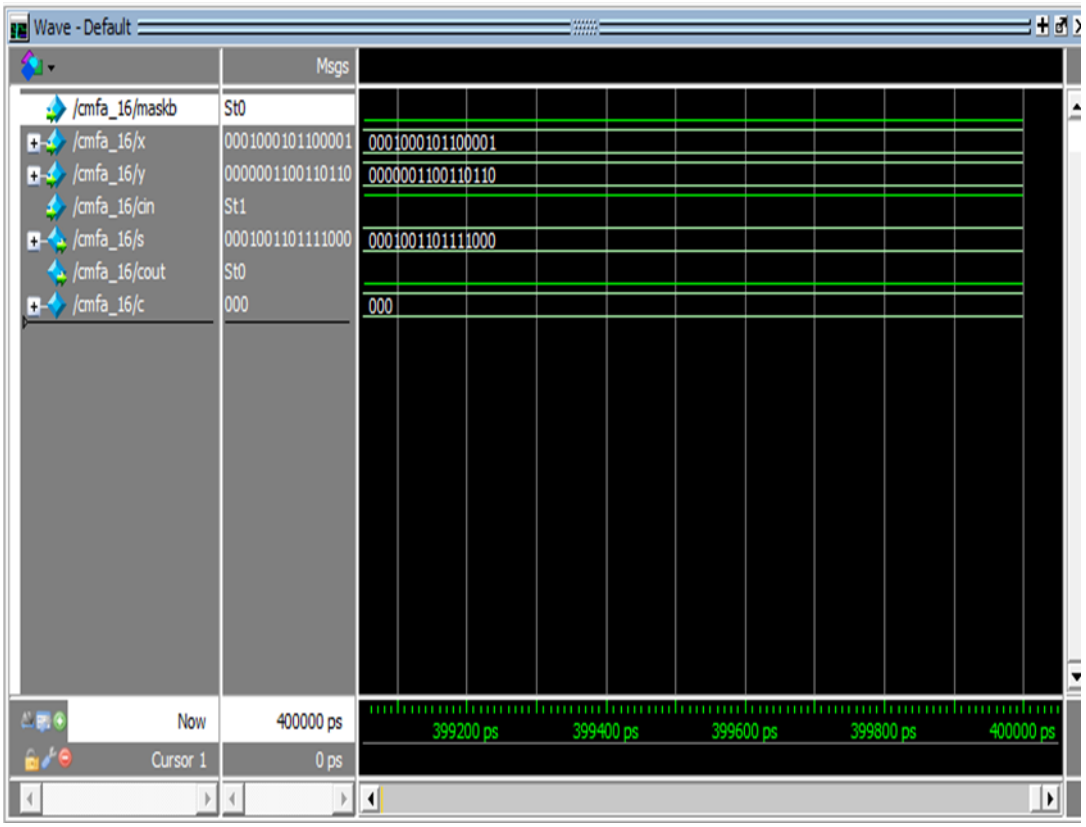

Fig: 7 Carry input =1 when mask input is 0

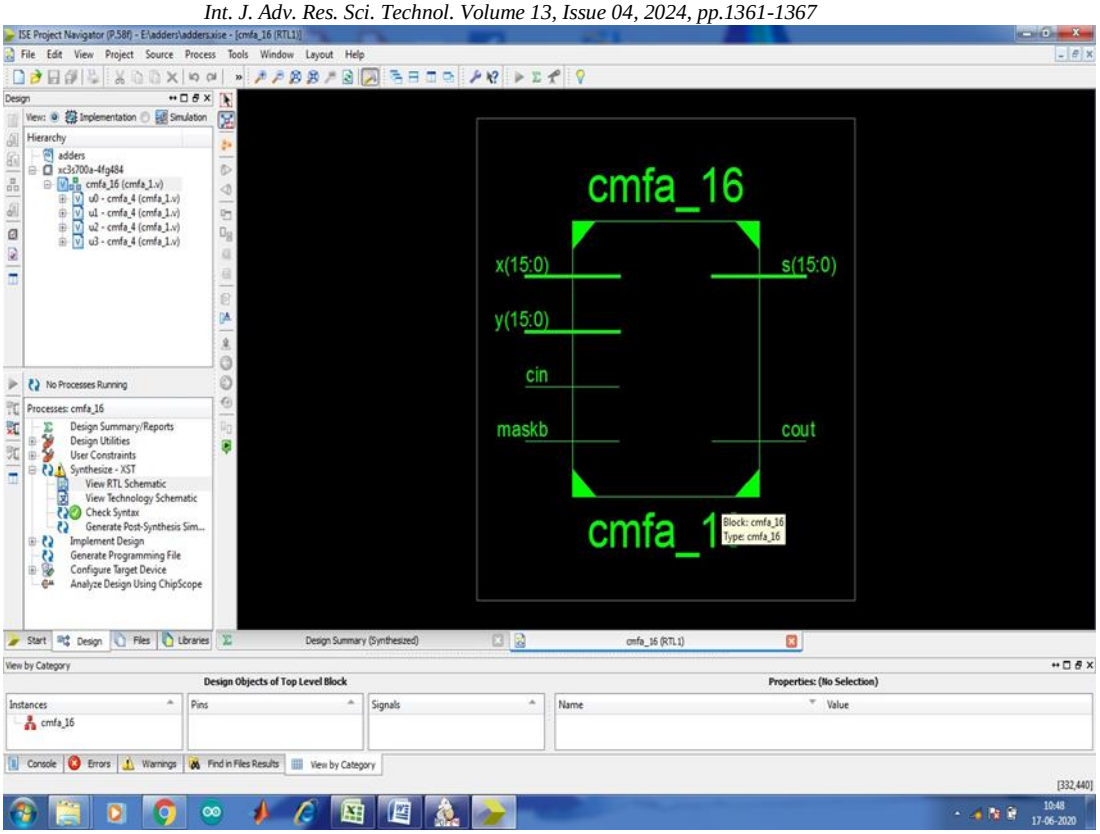

Fig: 8 RTL Schematic view of CMFA_16

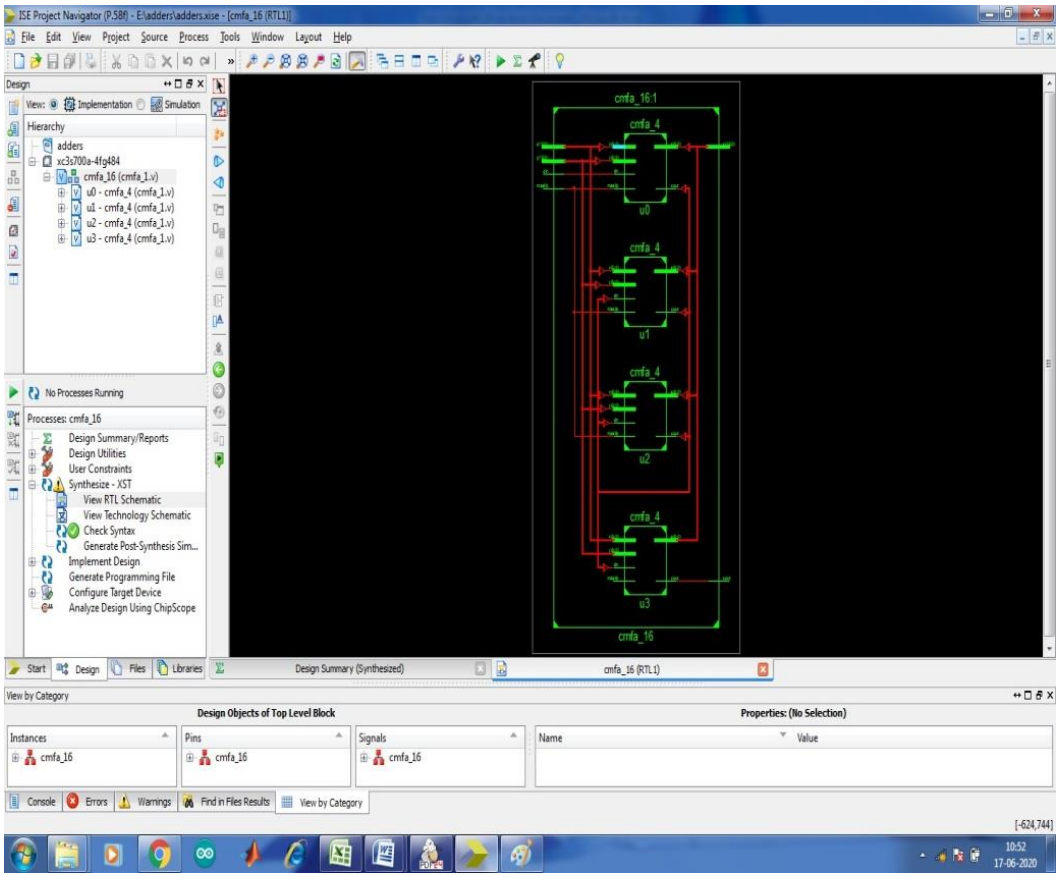

Fig: 9 RTL Schematic view of CMFA_16 using CMFA_4

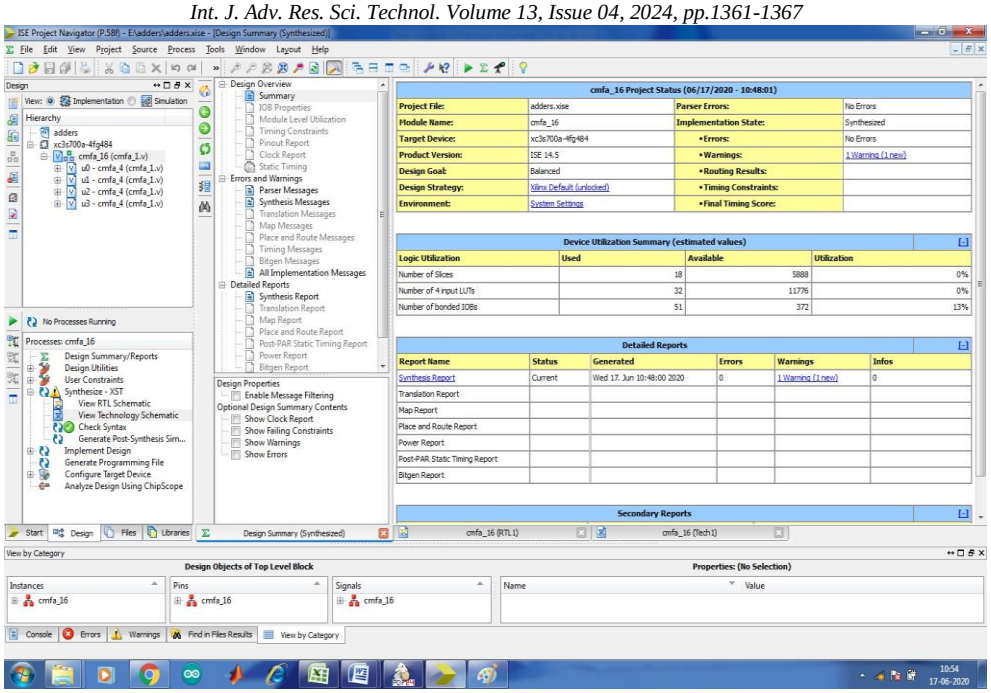

Fig: 10 Design Summary

CONCLUSION

This work proposed an accuracy-configurable adder that can be configured without incurring additional power or delay costs. The suggested adder is based on the traditional CLA, and by hiding the carry propagation during runtime, its accuracy can be modified. The experimental results show that compared to the standard CLA, the suggested adder offers significant power savings and speedup with a minimal area overhead.

Moreover, the experimental results show that the suggested adder accomplishes the initial goal of providing an objectively optimum outcome between power and latency without compromising accuracy when compared to previously researched configurable adders. Additionally, it was discovered that the examined application's quality standards were unaffected.

FUTURE SCOPE

To provide dynamic levels of approximation, accuracy variable adder designs have been created; however, because of carry prediction and redundant computing, these designs require bigger area overhead. So, Future work will include the reduction of power dissipation and area by reducing number of transistors used for implementing and we can also do masking for various adders.

REFERENCES

[1] S. Cotofana, C. Lageweg, and S. Vassiliadis, "Addition related arithmetic operations via controlled transport of charge", IEEE Transactions on Computers, vol. 54, no. 3, pp. 243-256, Mar. 2005.

[2] V. Beiu, S. Aunet, J. Nyathi, R. R. Rydberg, and W. Ibrahim, "Serial Addition: Locally Connected

[3] Architectures", IEEE Transactions on Circuits and Systems-I: Regular papers, vol. 54, no. 11, pp. 2564-2579, Nov. 2007.

[4] S. Venkataramani, V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Quality programmable vector processors for approximate computing", 46th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pp. 1-12, Dec. 2013.

[5] A. B. Kahng, and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs", IEEE/ACM Design Automation Conference (DAC), pp. 820-825, Jun. 2010.

[6] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, "On Reconfiguration- Oriented Approximate Adder Design and Its Application", IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 48-54, Nov. 2013.

[7] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-Power digital signal processing using approximate adders", IEEE Transactions on Comptuer Aided Design of Integrated Circuits and Systems, vol. 32, no. 1, pp. 124-137, Jan. 2013.

[8] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-Inspired imprecise computational blocks for efficient VLSI implementation of Soft- Computing applications", IEEE Transactions on Circuits and Systems I: Regular papers, vol. 57, no. 4, pp. 850-862, Apr. 2010.

[9] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "MACACO: modeling and analysis of circuits for approximate computing", IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 667-673, Nov. 2011.

[10] NanGate, Inc. NanGate FreePDK45 Open Cell Library, http://www.nangate.com/?page_id=2325, 2008

[11] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders", IEEE Transactions on Computers, vol. 62, no. 9, pp. 1760-1771, Sep. 2013.

[12] M. S. Lau, K. V. Ling, and Y. C. Chu, "Energy-Aware probabilistic multiplier: Design and Analysis", International Conference on Compliers, architecture, and synthesis for embedded systems, pp. 281-290, Oct. 2009.

[13] T. Yang, T. Ukezono, and T. Sato, "A Low-Power Configurable Adder for Approximate Applications", 19th International Symposium on Quality Electronic Design, Mar. 2018