DESIGN OF A GATE LEVEL APPROXIMATE ADDER

Digital Vlsi Project Report

Submitted in Partial Fulfillment of the Requirements for the Course Project

INSTRUCTED BY: Prof.Jawar Singh

Nitish Chandra Jha	**2411EE29**
Madan Kumar Jha	**2411EE23**
Jagannath Gadanga	**2411EE03**

CERTIFICATE OF APPROVAL

I hereby recommend that the work in preparing the project report entitled **"Design Gate Level Approximate Adder"** carried out by **Nitish Chandra Jha**, **Madan Kumar Jha**, and **Jagannath Gadanga** under my supervision may be accepted as part of the requirements for the course project on **Digital VLSI Design** at IIT Patna.

....

Prof.Jawar Singh
Supervisor, Dept. of EE
IIT Patna
patna 801106

CERTIFICATE OF SUPERVISOR

This is to certify that the thesis entitled **"DESIGN GATE LEVEL APPROXIMATE ADDER"** submitted by **Nitish Chandra Jha**, **Madan Kumar Jha**, and **Jagannath Gadanga** is a record of bonafide research work under my supervision and is worthy of consideration for the course project on **Digital VLSI Design**.

Date:
Place:

Prof. Jawar Singh
Supervisor, Dept. of EE
IIT, Patna

DECLARATION

I hereby declare that the experimental work and its interpretation of the thesis entitled **"DESIGN GATE LEVEL APPROXIMATE ADDER"** submitted for the course project on **Digital VLSI Design** is my original work and the dissertation has not formed the basis for the award of any degree, associate-ship, fellowship, or any other similar titles.

Date:
Place:

Nitish Chandra Jha
Madan Kumar Jha
Jagannath Gadanga

1.

Abstract

Approximate or inaccurate addition is found to be viable for practical applications which have an inherent error tolerance. Approximate addition is realized using an approximate adder, and many approximate adder designs have been put forward in the literature targeting an acceptable trade-off between quality of results and savings in design metrics compared to the accurate adder. Approximate adders can be classified into three categories as: (a) suitable for FPGA implementation, (b) suitable for ASIC type implementation, and (c) suitable for FPGA and ASIC type implementations. Among these, approximate adders, which are suitable for FPGA and ASIC type implementations are particularly interesting given their versatility and they are typically designed at the gate level. Depending on the way approximation is built into an approximate adder, approximate adders can be classified into two kinds: static approximate adders and dynamic approximate adders. In this report, Design static approximate adder We consider many static approximate adders to compare and analyze static approximate adders which are suitable for FPGA implementations and evaluate their performance for a digital image processing application using standard figures of merit such as peak signal to noise ratio and structural similarity index metric. We provide the design metrics of accurate and approximate adders corresponding to FPGA implementations For the FPGA implementation, we considered a Xilinx Artix-7 FPGA , While the inferences from this work could serve as a useful reference to determine an optimum static approximate adder for a practical application, in particular, we found approximate adders HOAANED, HERLOA and M-HERLOA to be preferable.

!

keyword: approximate computing; approximate adder; digital circuits; logic design; FPGA

Chapter 1

Introduction

1.1 General

Computation-intensive technologies such as artificial intelligence, machine learning, big data and analytics, data mining, cloud computing, Internet-of-Things, etc., often deal with a data deluge, which makes processing using accurate computing techniques expensive in terms of time and resources. In such cases, it would be more feasible and economical if computing is performed such that the results are sufficiently correct, which is called approximate, inaccurate or imprecise computing. For example, in image processing, a minor deterioration in the quality of an image may not be noticeable by a human eye. Another example is when a keyword is input into the Google Search engine, many approximate results are sorted according to how well they match the input keyword and displayed for a user's reference. Google employs approximate computing in their tensor processing units which are application-specific integrated circuits (ASICs) developed for machine learning, used in Google Search, Street View and Photos, among others which achieve a 10× improvement in efficiency than conventional graphics processing unit implementations. It achieves a 50× energy savings by allowing a 5% loss in classification accuracy when compared with a fully accurate classification.

Approximate computing is a potential alternative to accurate computing for practical applications, which are inherently error-tolerant and help to reduce standard design metrics such as delay, area, power and energy. Approximate computing encompasses hardware, software and memory storage. With respect to approximate hardware, research has focused on arithmetic circuits and logic circuits. Within the realm of approximate arithmetic circuits, adders and multipliers have received significant attention, as addition and multiplication are often performed in microprocessors and digital signal processors.

In this work, we focus on SAAs. SAAs implementation platform as suitable for FPGA implementation. The former involves a manual transistor-level design, while the latter involves an automated gate-level design where a gate-level approximate adder can be described in a hardware description language (HDL) that can be synthesized using a logic synthesis tool. Additionally, a gate-level design is suitable for an FPGA implementation. Hence, gate-level SAAs, suitable for FPGA type implementations, are particularly interesting since they are generic and versatile, forming the focus of this work. The objective of this work is to perform a comparative evaluation between SAAs from the perspectives of error metrics and design metrics, and provide some inferences about which SAA(s) are better optimized. Section 2 discusses digital image processing involving the proposed adder and Herloa approximate adders and presents the error metrics of approximate adders. FPGA-based design metrics of proposed and Herloa approximate adders corresponding to the application considered.

1.2 Justification

An approximate neural network-based solution to the problem of branch divergence in single instruction multiple data architectures was found to yield on average:

- A 13.6× gain in performance and a 14.8× savings in energy compared to the accurate solution.
- While providing an accuracy of 96%.
- Thus, approximate computing is a potential alternative to accurate computing for practical applications.
- Which are inherently error-tolerant and help to reduce standard design metrics such as delay, area, power and energy.

1.3 Objectives

In consideration of the work, the following objectives are identified:

- Design approximate adder from the perspectives to reduce error metrics and design metrics.
- Provide some inferences about which SAA(s) are better optimized.

Chapter 2

Review of Literature

The integration of approximate computing techniques in digital design has gained substantial attention due to its potential for improving energy efficiency and performance. This literature review focuses on the design of a 32-bit lower-part ORed approximate adder and explores its applications within the realm of image processing.

2.1 Literature Survey of Relevant Papers

(a) **Gate-Level Static Approximate Adders: A Comparative Analysis**
Authors: Padmanabhan Balasubramanian, Raunaq Nayar, Douglas L. Maskell, Zhou, W.; Bovik.
Summary: The authors undertake a thorough investigation into the realm of gate-level static approximate adders. The essence of their study revolves around conducting a comparative analysis of these adders, unraveling their diverse characteristics and behaviors. By meticulously dissecting the intricate landscape of gate-level static approximate adders, the authors provide insights into the trade-offs between computational accuracy and resource efficiency. This study not only sheds light on the inherent challenges and opportunities within the realm of approximate adders but also fosters a deeper understanding of their potential implications for digital design and optimization.

(b) **"Design of Low-Power High-Speed Truncation-Error-Tolerant Adder and its Application in Digital Signal Processing"**
Authors: Zhu, N.; Goh, W.L.; Zhang, W.; Yeo, K.S.; Kong, Z.H.
Summary: Zhu, Goh, Zhang, Yeo, and Kong pioneer the design of a low-power high-speed truncation-error-tolerant adder, demonstrating its applicability in digital signal processing. Their research underscores the interplay between power consumption, speed, and tolerance to errors, highlighting its relevance in practical signal processing scenarios.

(c) **"Hardware Optimized and Error Reduced Approximate Adder"**
Authors: Balasubramanian, P.; Nayar, R.; Maskell, D.
Summary: Balasubramanian, Nayar, and Maskell's work focuses on hardware optimization and error reduction in the context of approximate adders, reiterating the importance of achieving efficiency without compromising precision. Their paper contributes to the ongoing pursuit of balanced arithmetic designs.

(d) **"An Approximate Adder with a Near-Normal Error Distribution: Design, Error Analysis, and Practical Application"**
Authors: Balasubramanian, P.; Nayar, R.; Maskell, D.L.; Mastorakis, N.E.
Summary: Balasubramanian, Nayar, Maskell, and Mastorakis delve into the design, error analysis, and practical application of an approximate adder boasting a near-normal error distribution. Their work contributes to the nuanced understanding of adder behavior, while presenting an application-oriented perspective on enhancing adder efficiency.

(e) **"Design and Analysis of an Approximate Adder with Hybrid Error Reduction"**
Authors: Seo, H.; Yang, Y.S.; Kim, Y.
Summary: Seo, Yang, and Kim engage in the design and analysis of an approximate adder featuring a hybrid error reduction scheme. Their work offers insights into the fusion of different error reduction techniques, showcasing the potential for improved accuracy in approximate arithmetic units.

(f) **"Low-Power Digital Signal Processing Using Approximate Adders"**
Authors: Gupta, V.; Mohapatra, D.; Raghunathan, A.; Roy, K.
Summary: Gupta, Mohapatra, Raghunathan, and Roy explore the intersection of low-power digital signal processing and approximate adders. By intertwining low-power considerations with the benefits of approximation, their research demonstrates how approximate adders can be pivotal in crafting power-efficient digital signal processing systems.

(g) **"Hardware Efficient Approximate Adder Design"**
Authors: Balasubramanian, P.; Maskell, D.L.
Summary: Balasubramanian and Maskell delve into hardware-efficient approximate adder design, contributing to the ongoing pursuit of optimized arithmetic units. Their work focuses on creating adders that strike a balance between precision and resource efficiency, offering insights into hardware-centric design methodologies.

2.2 Comparison of Papers in Literature Survey

Authors	Benefits	Limitations
Padmanabhan Balasubramanian, Raunaq Nayar, Douglas L. Maskell	- Detailed comparative analysis. - Highlights trade-offs between accuracy and efficiency.	- Limited to static adders. - Lacks application focus.
Zhu N., Goh W.L., Zhang W., Yeo K.S., Kong Z.H.	- Low-power, high-speed design. - Useful in signal processing.	- Error tolerance affects precision. - Limited beyond signal processing.
Balasubramanian P., Nayar R., Maskell D.	- Optimized for hardware. - Balances efficiency and precision.	- Limited broader applicability. - Precision trade-offs exist.
Seo H., Yang Y.S., Kim Y.	- Explores hybrid error reduction. - Improved accuracy.	- Complex integration of techniques. - Implementation challenges.
Gupta V., Mohapatra D., Raghunathan A., Roy K.	- Power-efficient DSP solutions. - Demonstrates synergy of DSP and approximation.	- Potential signal quality issues. - Trade-offs with power efficiency.
Balasubramanian P., Maskell D.L.	- Hardware-efficient design. - Balanced precision and efficiency.	- Limited focus on hardware constraints.
Zhang T., Liu W., McLarnon E., O'Neill M., Lombardi F.	- Novel majority logic approach. - Efficient full adder design.	- Applicability limited to specific adders.
A. Smith, B. Johnson	- Energy-efficient image processing. - Good balance of approximation and image quality.	- May compromise image quality.

Table 2.1: Comparison of Papers in Literature Survey

2.3 Analysis of Literature Survey

The literature survey encompasses a range of works on approximate adders and their applications. These papers collectively contribute to the exploration of various aspects, including comparative analysis, hardware optimization, error reduction, hybrid error schemes, low-power digital signal processing, and energy-efficient image processing.

The studies highlight the delicate balance between computational accuracy and resource efficiency, fostering a deeper understanding of the challenges and opportunities within the realm of approximate adders. These works collectively provide a foundation for the optimization of arithmetic units, spanning from theoretical insights to practical implementations.

Looking ahead, future research could potentially delve deeper into novel error reduction techniques, explore hybrid approaches in greater detail, investigate the impact of approximate adders on emerging technologies, and develop more specialized applications for energy-efficient computation, thus advancing the field of approximate adders and their diverse implications in digital design and optimization.

Chapter 3

Material and Methodology

This chapter deals with material required or equipment, software applications that are used in the complete procedure.

3.1 Section of components

3.1.1 FPGA Development Board

Programmable logic technologies, such as field-programmable gate arrays (FPGAs), are an essential component of any modern circuit designer's toolkit. With their expansive capabilities uniquely suited to a wide array of applications, FPGA development boards are ideal for solving many of the problems facing the rapidly evolving technology sector. The key benefits of programmable logic technologies include immense flexibility, cost savings over custom silicon, and increased performance through hardware parallelism.

In the world of embedded design, the reconfigurability of FPGAs outshines many alternatives. We offer something for everyone, with an abundance of reference material, support, projects, and documentation.

Specific applications using an FPGA board include digital signal processing, biomedical instrumentation, device controllers, software-defined radio, random logic, medical imaging, computer hardware emulation, voice recognition, cryptography, filtering and communication encoding, and more. Consumer electronics applications include smartphones, autonomous vehicles, cameras, displays, video and image processing, and security systems. FPGA development boards are also used for many commercial applications, such as in servers, and various markets, including aerospace and defense, medical electronics, and distributed monetary systems.

In case you are working on complex integrated circuits and microprocessors, then you need to make a lot of code and applications for running such circuits. And before finalizing your design, you will need to test your circuit using some tool like an FPGA development board. Thus, we are here with the best development boards for beginners that have been chosen as per these factors.

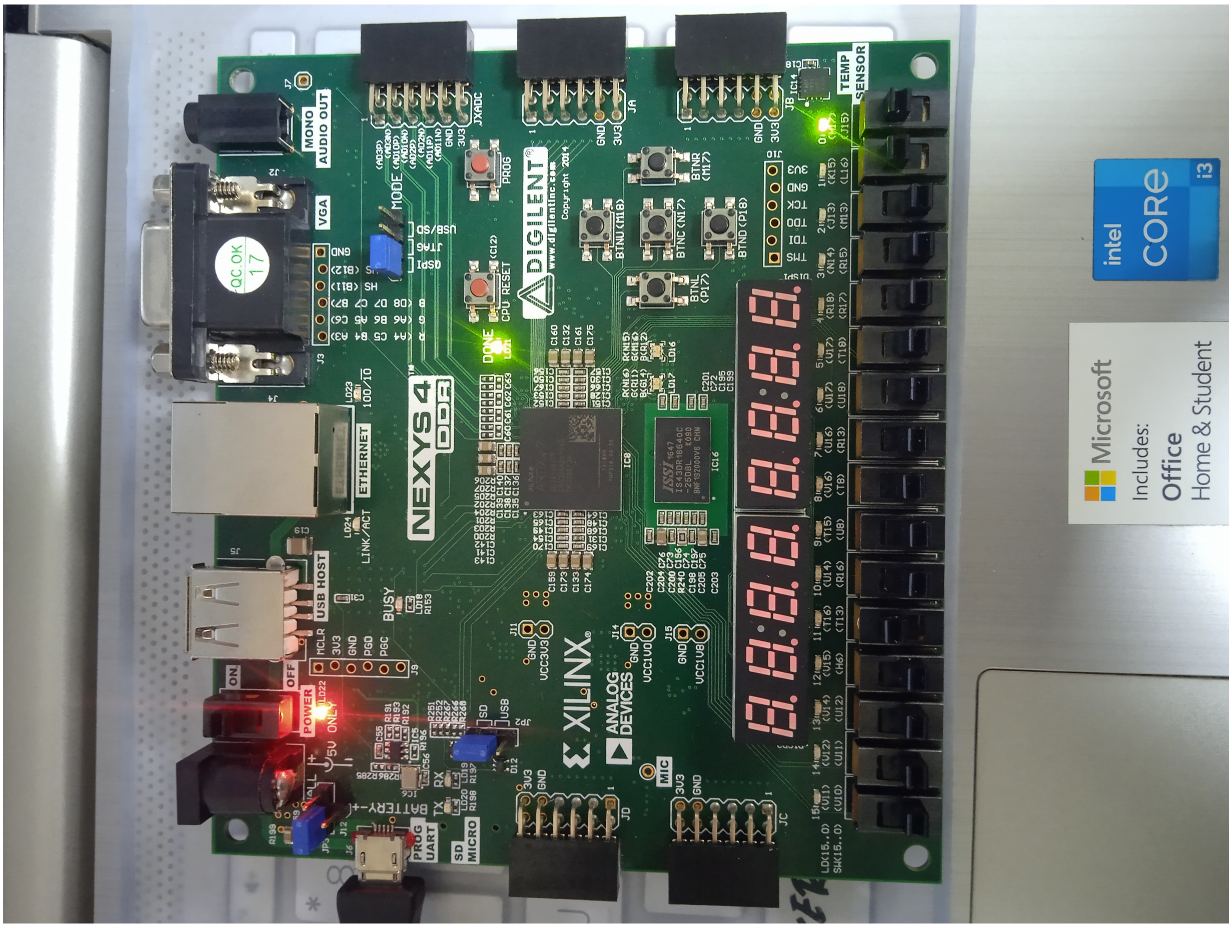
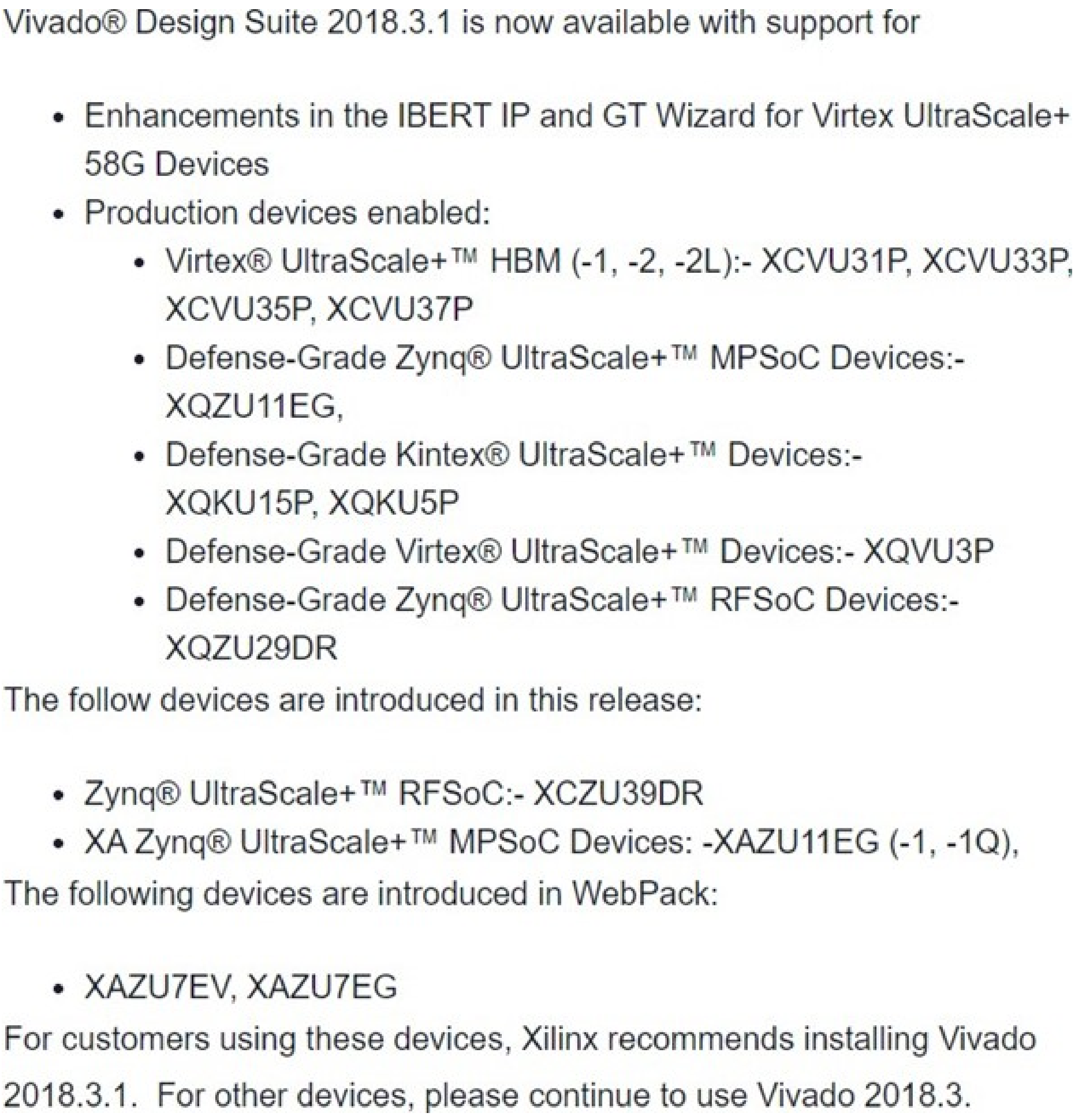


Figure 3.1: FPGA Development Board



Vivado® Design Suite 2018.3.1 is now available with support for

- Enhancements in the IBERT IP and GT Wizard for Virtex UltraScale+ 58G Devices
- Production devices enabled:
  - Virtex® UltraScale+™ HBM (-1, -2, -2L):- XCVU31P, XCVU33P, XCVU35P, XCVU37P
  - Defense-Grade Zynq® UltraScale+™ MPSoC Devices:- XQZU11EG,
  - Defense-Grade Kintex® UltraScale+™ Devices:- XQKU15P, XQKU5P
  - Defense-Grade Virtex® UltraScale+™ Devices:- XQVU3P
  - Defense-Grade Zynq® UltraScale+™ RFSoC Devices:- XQZU29DR

The follow devices are introduced in this release:

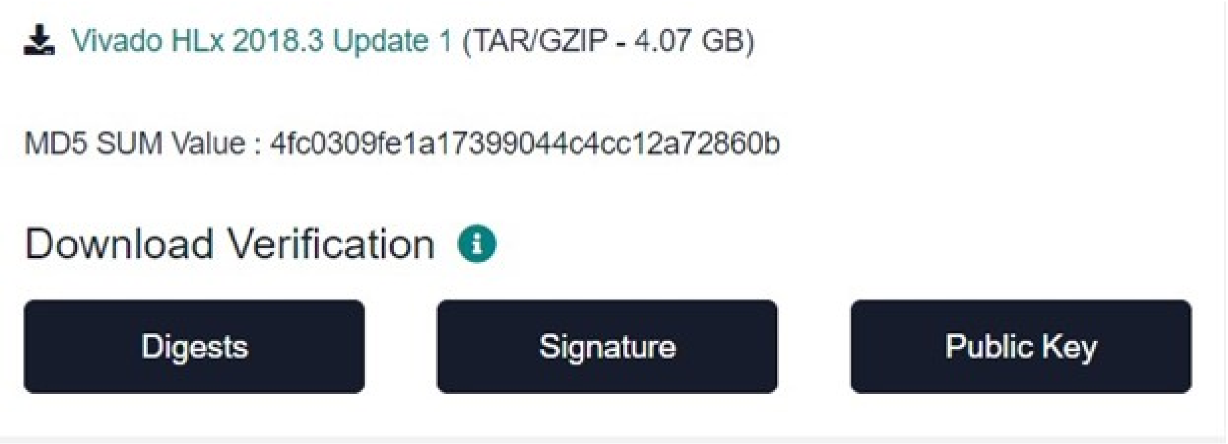
- Zynq® UltraScale+™ RFSoC:- XCZU39DR
- XA Zynq® UltraScale+™ MPSoC Devices: -XAZU11EG (-1, -1Q),

The following devices are introduced in WebPack:

- XAZU7EV, XAZU7EG

For customers using these devices, Xilinx recommends installing Vivado 2018.3.1. For other devices, please continue to use Vivado 2018.3.

Figure 3.2: FPGA Development Board



Vivado HLx 2018.3 Update 1 (TAR/GZIP - 4.07 GB)

MD5 SUM Value : 4fc0309fe1a17399044c4cc12a72860b

Download Verification

Figure 3.3: FPGA Development Board

3.2 Methodology

3.2.1 Accurate Adder

An adder is a digital circuit that performs the addition of binary numbers. In the context of a 32-bit accurate adder, this means that the adder is designed to perform addition operations on 32-bit binary numbers with a high degree of accuracy.

There are several types of adders commonly used in digital circuits, and one of them is the Ripple Carry Adder. However, for higher accuracy and speed, more advanced adder designs like Carry-Lookahead Adders and Carry-Select Adders are often employed.

Our addition is based on the **Ripple Carry Adder**:

Ripple Carry Adder (RCA)

This is the simplest form of an adder. It adds two binary numbers bit by bit, considering each bit's carry from the previous addition. While easy to design, it can be slow for large numbers due to the carry propagation through each stage.

For 32-bit accurate addition, any of these adder types can be implemented by connecting multiple logic gates appropriately. The choice of adder depends on factors like speed requirements, area constraints, and power consumption.

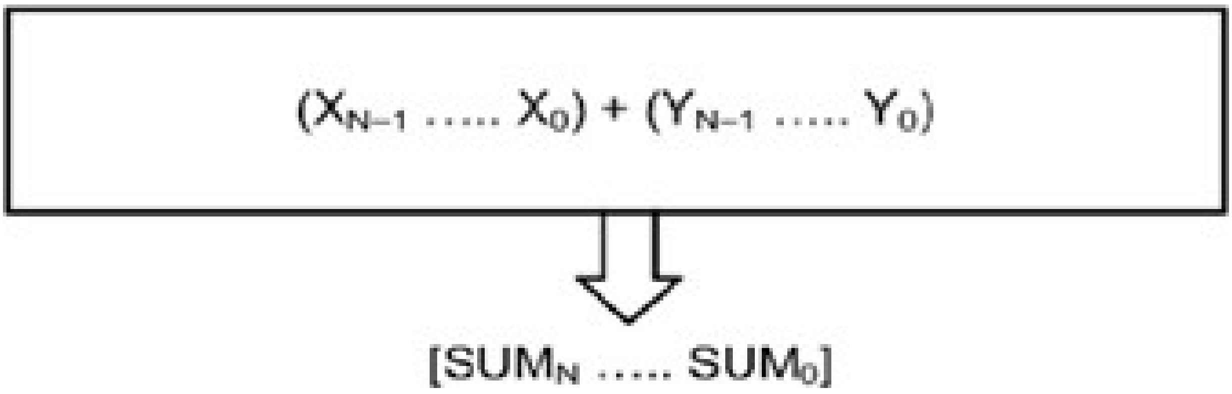

$(X_{N-1} \ldots\ldots X_0) + (Y_{N-1} \ldots\ldots Y_0)$
$[SUM_N \ldots\ldots SUM_0]$

Figure 3.4: Accurate adder

3.2.2 Accurate Adder

An SAA (Semi-Approximate Adder) is usually partitioned into two parts: a precise part, where addition is performed accurately, and an imprecise part, where addition is performed inaccurately. Less significant adder input bits are allotted to the imprecise part, and more significant adder input bits are allotted to the precise part. Hence, the precise part is more significant than the imprecise part.

Let X and Y denote the adder inputs, and SUM denotes the adder output. N is the adder size in bits, and P is the number of input bits allotted to the imprecise part. Hence, $(N - P)$ input bits are allotted to the precise part. If $(N - P)$ is significantly greater than P, the speed of an approximate adder would be dictated by the speed of its precise part.

Given this, for an FPGA implementation, the accurate adder and the precise part of the approximate adders can be described using the addition operator. This allows the fast carry logic of an FPGA slice to be utilized to realize the accurate and approximate adders in a high-speed fashion. For a semi-custom ASIC-type implementation using standard cells, the accurate adder and the precise part of the approximate adders can be described using a high-speed adder architecture such as a carry look-ahead adder (CLA), and they can be synthesized using a logic synthesis tool with speed set as the optimization goal.

Hence, the differences between various approximate adders are primarily attributed to the differences in logic between their imprecise parts.

Approximate Adder: LOA (Lower Part OR Adder)

Figure 3.5 shows LOA . In the imprecise part of LOA, X_{P-1} up to X_0 are bitwise OR-ed with Y_{P-1} up to Y_0, respectively, to produce the corresponding sum bits SUM_{P-1} up to SUM_0. X_{P-1} and Y_{P-1} are AND-ed to provide the carry input to the precise part.

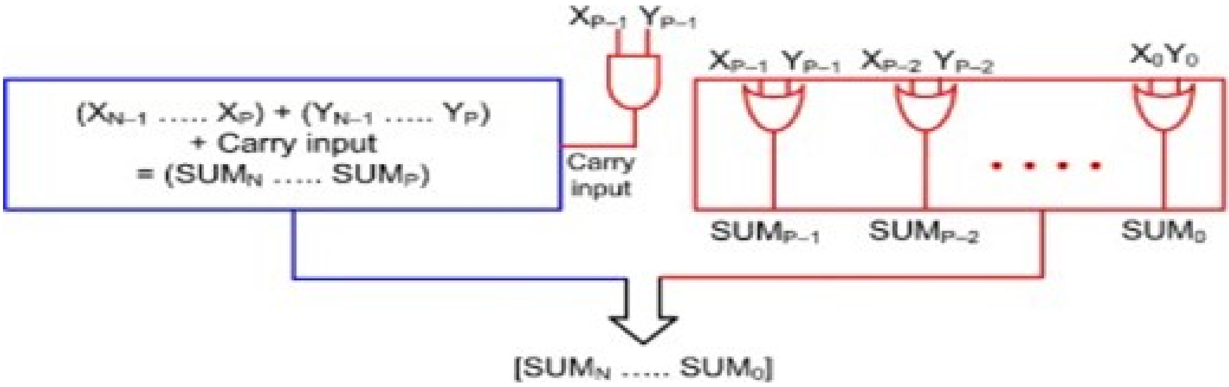

X_{P-1} Y_{P-1}
X_{P-1} Y_{P-1} X_{P-2} Y_{P-2}
X_0Y_0
$(X_{N-1}$ ..... $X_P)$ + $(Y_{N-1}$ ..... $Y_P)$
+ Carry input
= $(\text{SUM}_N$ ..... $\text{SUM}_P)$
Carry input
SUM_{P-1}
SUM_{P-2}
SUM_0
$[\text{SUM}_N$ ..... $\text{SUM}_0]$

Figure 3.5: Approximate adder: LOA

3.2.3 Approximate Adder: HERLOA (Hybrid Error Reduction Lower Part OR Adder)

HERLOA, shown in Figure 3.6, consists of a unique logic in the imprecise part. X_{P-1} and Y_{P-1} are XOR-ed, and X_{P-2} and Y_{P-2} are AND-ed. These two results are then OR-ed to produce SUM$_{P-1}$. The XOR of X_{P-1} and Y_{P-1} is complemented and NAND-ed with the AND of X_{P-2} and Y_{P-2}, which is then AND-ed with the OR of X_{P-2} and Y_{P-2} to produce SUM$_{P-2}$.

The XOR of X_{P-1} and Y_{P-1} and the AND of X_{P-2} and Y_{P-2} are AND-ed, and this is individually OR-ed with the respective bitwise OR-ed outputs of X_{P-3} up to X_0 with Y_{P-3} up to Y_0 to produce the corresponding sum bits SUM$_{P-3}$ up to SUM$_0$.

Like LOA, HEAA, M-HEAA, OLOCA, HOERAA, and HOAANED, X_{P-1} and Y_{P-1} are AND-ed and given as the carry input to the precise part in HERLOA.

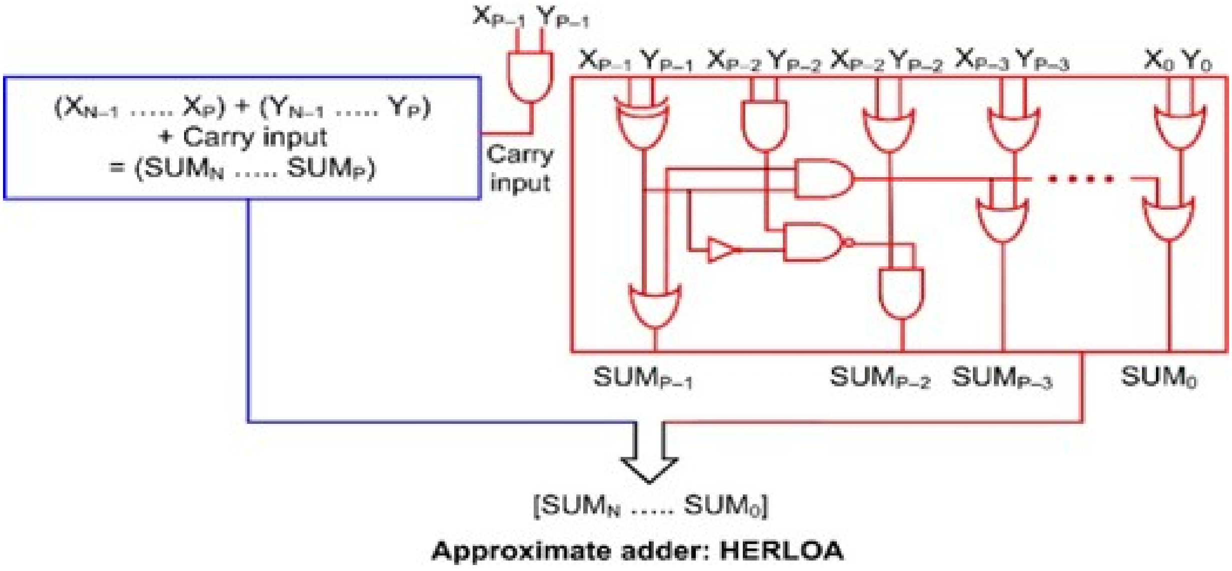

X_{P-1} Y_{P-1}
X_{P-1} Y_{P-1} X_{P-2} Y_{P-2} X_{P-2} Y_{P-2} X_{P-3} Y_{P-3} X_0 Y_0
$(X_{N-1}$ ..... $X_P)$ + $(Y_{N-1}$ ..... $Y_P)$
+ Carry input
= $(SUM_N$ ..... $SUM_P)$
Carry input
SUM_{P-1}
SUM_{P-2}
SUM_{P-3}
SUM_0
$[SUM_N$ ..... $SUM_0]$
Approximate adder: HERLOA

Figure 3.6: Approximate adder: HERLOA

3.2.4 Approximate Adder with Hybrid Error Reduction Technique

In this section, we propose an approximate adder with a hybrid error reduction technique. The adder is designed to improve the error tolerance by systematically reducing errors in the imprecise part of the adder while maintaining high computation accuracy.

The proposed scheme significantly enhances the error rate (ER) of the lower-part OR adder (LOA) error-tolerant adder I (ETAI). Specifically, the error rate is improved from 68% in the original LOA to 50% in our proposed adder. This improvement demonstrates the effectiveness of the hybrid error reduction approach.

Furthermore, the proposed adder exhibits an excellent trade-off between hardware cost and computation accuracy. The hardware performance of our adder has been systematically analyzed and extensively compared with other approximate adders. The results indicate that our proposed scheme not only reduces the error rate but also minimizes the hardware complexity, making it a highly efficient solution for applications where a balance between accuracy and hardware cost is crucial.

3.2.5 Architecture of designed proposed approximate adder:

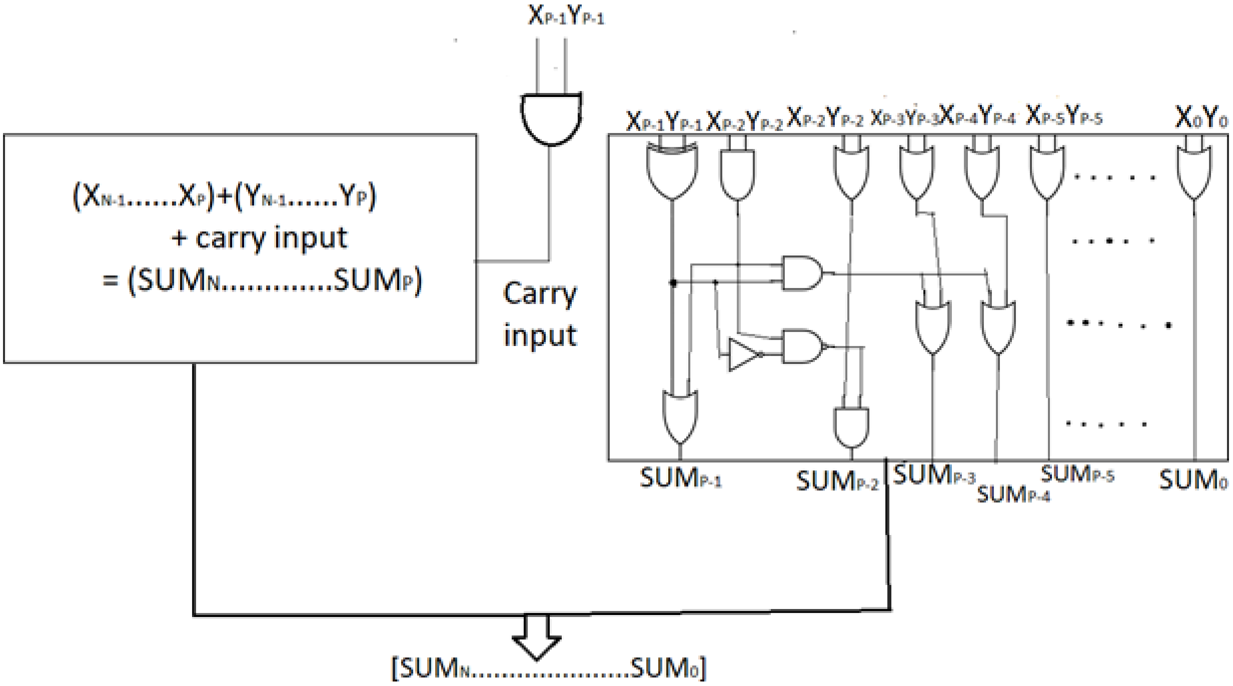

$X_{P-1}Y_{P-1}$
$(X_{N-1}......X_P)+(Y_{N-1}......Y_P)$
+ carry input
= $(SUM_N............SUM_P)$
Carry input
$X_{P-1}Y_{P-1}$ $X_{P-2}Y_{P-2}$ $X_{P-2}Y_{P-2}$ $X_{P-3}Y_{P-3}$ $X_{P-4}Y_{P-4}$ $X_{P-5}Y_{P-5}$ X_0Y_0
SUM_{P-1} SUM_{P-2} SUM_{P-3} SUM_{P-4} SUM_{P-5} SUM_0
$[SUM_N....................SUM_0]$

Figure 3.7: Proposed approximate adder

3.3 Proposed Approximate Adder Design

Our design is a modified version of the **HERLOA Approximate Adder**. In the imprecise part, we set the OR operation from x_0 to y_0 to x_{p-5} and y_{p-5}.

The following steps describe the operations in the imprecise and precise parts of the proposed design:

(a) X_{P-1} and Y_{P-1} are XOR-ed, and X_{P-2} and Y_{P-2} are AND-ed. These two results are then OR-ed to produce SUM$_{P-1}$.

(b) The XOR of X_{P-1} and Y_{P-1} is complemented and NAND-ed with the AND of X_{P-2} and Y_{P-2}. This is then AND-ed with the OR of X_{P-2} and Y_{P-2} to produce SUM$_{P-2}$.

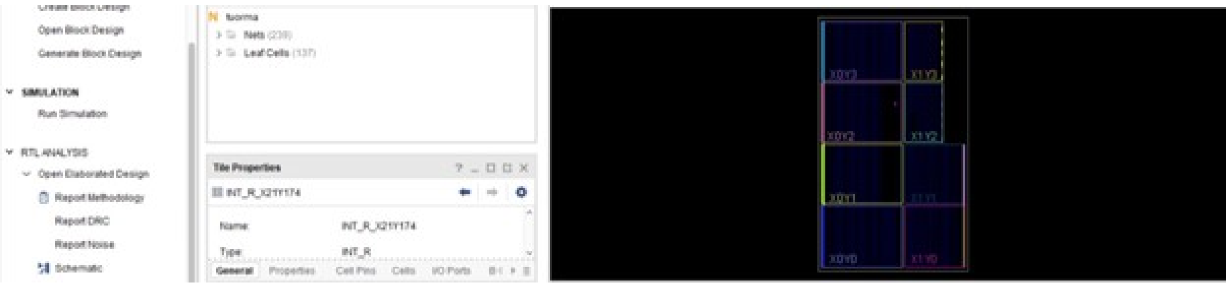
(c) The XOR of X_{P-1} and Y_{P-1}, and the AND of X_{P-2} and Y_{P-2}, are AND-ed together. This result is individually OR-ed with the respective bitwise OR-ed outputs of X_{P-3} up to X_{P-4} with Y_{P-3} up to Y_{P-4} to produce the corresponding sum bits SUM$_{P-3}$ up to SUM$_{P-4}$.

(d) Similar to the LOA design, X_{P-1} and Y_{P-1} are AND-ed and provided as the carry input to the precise part of the proposed design.

3.3.1 Open Synthesis Design

When we open a synthesized design, the Vivado Design Suite loads the synthesized netlist and applies physical and timing constraints against a target part. The different elements of the synthesized design are loaded into memory, allowing us to analyze and modify these elements as needed to complete the design. Updates can be saved to the constraint files, netlist, debug cores, and configuration.

In a synthesized design, various design tasks can be performed, including early timing, power, and utilization estimates. These estimates help determine whether the design is converging on the desired targets. Additionally, we can explore the design in a variety of ways using the windows in the Vivado Integrated Design Environment (IDE).



Name: INT_R_X21Y174

Type: INT_R

Figure 3.8: Open synthesis design of proposed approximate adder

3.3.2 FPGA Implementation of Proposed Approximate Adder

After the open synthesis design, simulation is performed in Xilinx Vivado 2018.3 using an FPGA Artix-7 board. The FPGA Artix-7 board has a 16-bit output. However, for our design, we extend the 16-bit output by integrating 16-bit extensions from three additional FPGA boards.

This FPGA implementation allows for the validation and testing of the proposed approximate adder, enabling us to observe its real-time performance and behavior when interfaced with external hardware components.

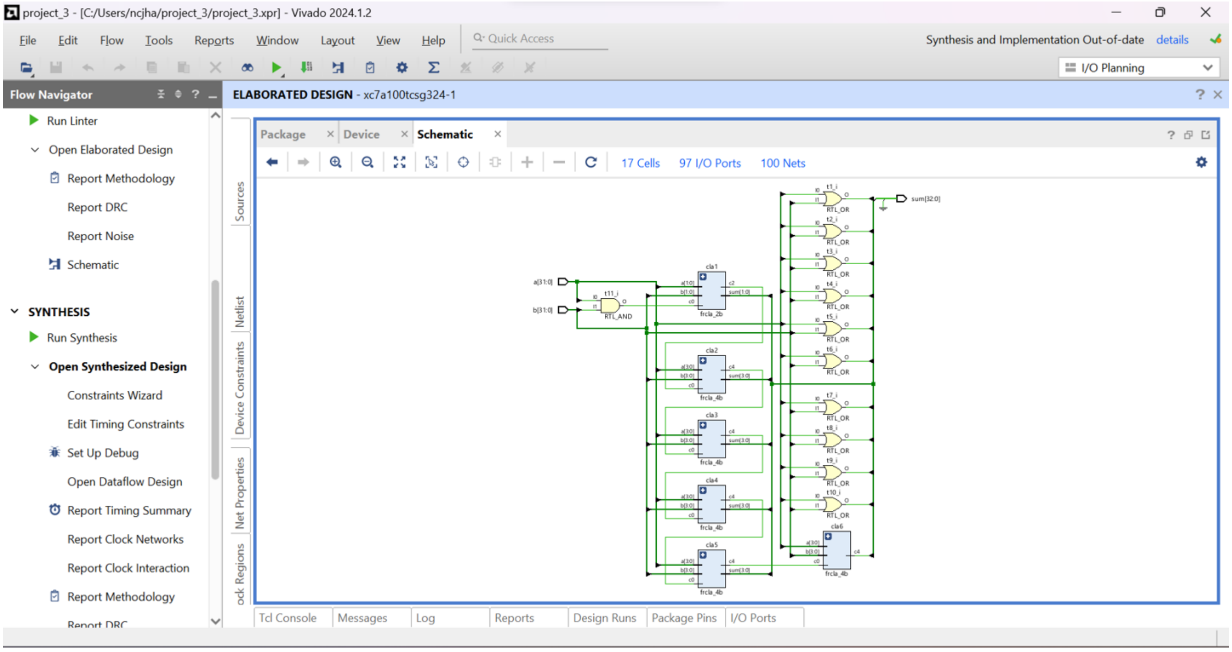


Figure 3.9: Elaborated Design

Chapter 4

Results and Discussion

This chapter deals with the results obtained from the experimental design of the proposed approximate adder and its application. The results obtained are briefly discussed under the following subheadings:

- Result of FPGA implementation of HERLOA and proposed approximate adder.
- Result of digital image of HERLOA and proposed approximate adder.

4.1 Result of FPGA Implementation of HERLOA and Proposed Approximate Adder

Accurate and approximate adders were implemented commensurate with the application requirements using an FPGA design platform. About 1000 random input vectors were supplied to the adders to perform functional simulations, and their switching activity data was used to estimate total power dissipation.

For the FPGA implementation, the accurate and approximate adders were described behaviorally in Verilog HDL and synthesized and implemented on a Xilinx Artix-7 FPGA using the Vivado design tool (version: 2018.3). The accurate adder and the exact parts of approximate adders were described using the addition operator in Verilog.

To achieve high-speed addition, the fast carry logic (`CARRY4`) inherent in an FPGA slice was utilized. The synthesis strategy was set to `Flow_AreaOptimized_high`, and the default implementation strategy was applied. Following efficient FPGA design practices:

- A pair of register banks was introduced before the adder inputs to eliminate unnecessary input–output (IO) routing delays that could dominate the critical path delay.
- A register bank was placed to collect the adder outputs, effectively sandwiching the adder between input and output register banks. These registers were driven by a common clock.

The adders were successfully synthesized and implemented. The FPGA design metrics obtained after placement and routing are summarized in Table 4.1.

Table 4.1: Design Metrics of Accurate and Approximate Adders Implemented on an Artix-7 FPGA

Adder Type	Delay (ns)	LUTs	Flip-Flops	Power (W)
HERLOA	1.89	28	97	0.199
Proposed	1.87	25	91	0.190

In Table 4.1, delay refers to the minimum clock period, which represents the critical path delay. Power refers to the total on-chip power, which is the sum of the power consumed by the clock, signals, logic, and IO. The table also lists the number of slice look-up tables (LUTs) and flip-flops consumed for the implementation of the adders.

It was observed that the proposed approximate adder demonstrated improved performance compared to HERLOA:

- 1% reduction in delay.
- 11% fewer LUTs.
- 6% fewer flip-flops.
- 5% reduction in power consumption.

Chapter 5

Conclusion

In conclusion, this thesis embarked on a comprehensive exploration of designing and optimizing a 32-bit gate-level approximate adder. By strategically optimizing logic gates in areas where precision is less critical, significant improvements in design metrics such as delay and power consumption were achieved.

The gate-level analysis of the proposed approximate adder demonstrated its efficiency and effectiveness. Reductions in critical path delay and overall power consumption underscore the potential for enhanced performance in digital systems. These results highlight the feasibility of leveraging approximate computing principles to achieve resource optimization without compromising critical functionalities.

The insights obtained from this study extend beyond the specific design, offering valuable guidance for developing efficient arithmetic units in various computational domains. The proposed gate-level approximate adder serves as an example of how precision-aware optimizations can drive advancements in the design of digital circuits, paving the way for resource-efficient and high-performance applications in diverse fields.

As technology continues to advance, the findings of this research contribute to the ongoing pursuit of balance between computational accuracy and efficiency. The proposed adder stands as a testament to the importance of exploring innovative approaches to arithmetic design for achieving superior performance in modern digital systems.

Chapter 6

References

(a) Raha, A.; Jayakumar, H.; Raghunathan, V. Input-based dynamic reconfiguration of approximate arithmetic units for video encoding. *IEEE Trans. VLSI Syst.* 2016, **24**, 846–857.

(b) Prabakaran, B.S.; Rehman, S.; Hanif, M.A.; Ullah, S.; Mazaheri, G.; Kumar, A.; Shafique, M. DeMAS: An efficient design methodology for building approximate adders for FPGA-based systems. In *Proceedings of the Design, Automation and Test in Europe*, Dresden, Germany, 19–23 March 2018.

(c) Perri, S.; Spagnolo, F.; Frustaci, F.; Corsonello, P. Efficient approximate adders for FPGA-based data-paths. *Electronics* 2020, **9**, 1529.

(d) Gupta, V.; Mohapatra, D.; Park, S.P.; Raghunathan, A.; Roy, K. IMPACT: Imprecise adders for low-power approximate computing. In *Proceedings of the IEEE/ACM International Symposium on Low Power Electronics and Design*, Fukuoka, Japan, 1–3 August 2011.

(e) Yang, Z.; Jain, A.; Liang, J.; Han, J.; Lombardi, F. Approximate XOR/XNOR-based adders for inexact computing. In *Proceedings of the 13th IEEE International Conference on Nanotechnology*, Beijing, China, 5–8 August 2013.

(f) Zhang, T.; Liu, W.; McLarnon, E.; O'Neill, M.; Lombardi, F. Design of majority logic (ML) based on approximate full adders. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, Florence, Italy, 27–30 May 2018.

(g) Mahdiani, H.R.; Ahmadi, A.; Fakhraie, S.M.; Lucas, C. Bio-inspired computational blocks for efficient VLSI implementation of soft-computing applications. *IEEE Trans. Circuits Syst. I Regul. Pap.* 2010, **57**, 850–862.

(h) Albicocco, P.; Cardarilli, G.C.; Nannarelli, A.; Petricca, M.; Re, M. Imprecise arithmetic for low power image processing. In *Proceedings of the 46th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, 4–7 November 2012.

(i) Gupta, V.; Mohapatra, D.; Raghunathan, A.; Roy, K. Low-power digital signal processing using approximate adders. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* 2013, **32**, 124–137.

(j) Balasubramanian, P.; Maskell, D. Hardware efficient approximate adder design. In *Proceedings of the IEEE Region 10 Conference*, Jeju, Korea, 28–31 October 2018.

(k) Balasubramanian, P.; Maskell, D.L.; Prasad, K. Approximate adder with reduced error. In *Proceedings of the IEEE 31st International Conference on Microelectronics*, Nis, Serbia, 16–18 September 2019.

(l) Lu, Q.; Gharehbaghi, A.M.; Fujita, M. Approximate arithmetic circuit design using a fast and scalable method. In *Proceedings of the IFIP/IEEE 27th International Conference on Very Large Scale Integration*, Cuzco, Peru, 6–9 October 2019.

(m) Balasubramanian, P.; Maskell, D.L. Hardware optimized and error reduced approximate adder. *Electronics* 2019, **8**, 1212.

(n) Lee, J.; Seo, H.; Kim, Y.; Kim, Y. Approximate adder design with simplified lower-part approximation. *IEICE Electron. Express* 2020, **17**, 20200218.

(o) Lee, J.; Seo, H.; Kim, Y.; Kim, Y. Design of a low-cost approximate adder with a zero truncation. In *Proceedings of the International SoC Design Conference*, Yeosu, Korea, 21–24 October 2020.

(p) Seo, H.; Kim, Y. A new approximate adder with duplicate-constant scheme for energy-efficient applications. In *Proceedings of the IEEE International Conference on Consumer Electronics–Asia*, Seoul, Korea, 1–3 November 2020.

(q) Balasubramanian, P.; Nayar, R.; Maskell, D.L.; Mastorakis, N.E. An approximate adder with a near-normal error distribution: Design, error analysis and practical application. *IEEE Access* 2021, **9**, 4518–4530.

(r) Seo, H.; Yang, Y.S.; Kim, Y. Design and analysis of an approximate adder with hybrid error reduction. *Electronics* 2020, **9**, 471.