ISSN: 2321-2152

IJMECE

International Journal of modern electronics and communication engineering

E-Mail
editor.ijmece@gmail.com
editor@ijmece.com

www.ijmece.com

ISSN2321-2152

www.ijmece .com

Vol 12, Issue.2 April 2024

APPROXIMATE PARALLEL PREFIX ADDERS

SANDENABOINA SRINIVAS*, DR. E KRISHNAHARI**

PG SCHOLAR*, ASSOCIATE PROFESSOR**

DEPARTMENT OF ECE, HOLYMARY INSTITUTE OF TECHNOLOGY AND SCIENCE (APPROVED BY AICTE NEW DELHI, AFFILIATED TO JNTUHYDERABAD) BOGARAM (V), KEESARA (M), MEDCHAL DISTRICT -501 301

ABSTRACT: A variety of other applications, including those dealing with machine learning, signal processing, image processing, and video processing, rely heavily on addition units as part of their essential algorithms. In addition to their intrinsic utility, addenda serve as crucial parts in a variety of mathematical operations such as division, subtraction, comparison, multiplication, and squaring. In terms of speed, the parallel prefix adder (PPA) is one of the most effective adders. The parallel prefetch graph is a graphical representation used to display prefetch operators (POs), also known as transport operator nodes. Parallel Prefix Adders' (PPAs') fast performance can be due to their successful parallelization implementation in both the carry production (G) and propagation (P) stages. This paper introduces AxPPAs, a PPA approximation based on estimations taken from POs. Each of these uses less energy than the others. We conducted studies with two major kernels to examine signal processing applications: a video accelerator using sum-of-squares (SSD) and a filter kernel with finite impulse response (FIR). Examine the effectiveness of the AxPPA-LF in light of recent improvements in energy-saving devices. When applied to available data on energy quality and area quality, the AxAs framework reveals the existence of a fresh Pareto front.

INTRODUCTION VLSI (Very Large-Scale Integration) design refers to the process of producing integrated circuits (ICs) with a large number of transistors on a single device. As time has passed, significant advances in the realm of integrated circuit (IC) design have been made, resulting in the introduction of electronic devices with expanded functions and improved energy efficiency. The reduction of transistor size is a significant accomplishment in the realm of integrated circuit (IC) design. Moore's Law asserts that the number of transistors incorporated into a microprocessor doubles every two years due to its enduring validity spanning multiple decades. The constant shrinking of transistor size has enabled the development of miniature microchips that are faster, more energy-efficient, and environmentally

friendly. One notable advancement is the creation of novel materials and technologies. The effective integration of high-k metal gate (HKMG) technology has allowed for the production of transistors with less power leakage and a smaller overall size. Furthermore, due to their unique electrical properties, new materials such as graphene and carbon nanotubes show promise for future integrated circuit (IC) designs. Furthermore, developments in design automation technologies have significantly improved the efficacy and output of design procedures using Very Large Scale Integration (VLSI). By easing the simulation and design of complex circuits, computer-aided design (CAD) technologies have considerably decreased the time and mental effort necessary for design iterations. Using machine learning and artificial intelligence (AI) has also made it possible to add intelligent automation to many stages of integrated circuit (IC) design, which improves the accuracy and efficiency of the design process. Electronic devices that stand out for their smaller dimensions, quicker operation, and improved energy conservation are the result of advancements in integrated circuit (IC) design. The advancement of design automation tools, the use of groundbreaking materials and technologies, and the constant reduction in transistor size have all contributed to the accomplishments listed above. As technology advances, it is envisaged that the domain of integrated circuit (IC) design will undergo further alterations, allowing for the development of increasingly sophisticated and novel electronic gadgets. Approximation computing, also known as AxC, has been shown to be successful in energy conservation and error resilience enhancement in a variety of scenarios. This technique has a wide range of applications, including machine learning and the manipulation of digital signals, photos, and videos. According to Cross layer AxC, in order to achieve increased hardware (HW) performance across the whole VLSI design hierarchy, a trade-off in quality is unavoidable. The extensive examination of cross-layer AxC is meant to maximize cost savings in scaled design space research. Researchers are becoming more interested in the AxC domain because of its potential to reduce the energy consumption of custom-level arithmetic units such as squarer’s, multipliers, and adders. Furthermore, novel data-centric AxC approaches have shown promise. These solutions include relevant data to improve the accuracy of circuit predictions within the application. During program execution,

a technique known as "probabilistic pruning" is used. It falls under the category of data-driven approximate logic synthesis (ALS). This method efficiently eliminates netlist gates by monitoring the most likely state of the circuit. The evolution of the AxLS open-source framework is described in the scholarly article [1]. The framework provided above is used to evaluate pruning procedures. With the goal of increasing the efficacy of gate-level netlist modification, AxLS created a graphical structure-based intermediary representation. [2] proposes a data-driven AxC approach for energy conservation. This technique allows for the selective control of logic lines in circuits while they are running. A simulation of the gatelevel netlist is needed to put together the frameworks from [3] and [4] that need data-driven AxC circuits. The activity information must be received from the circuit's nodes. An alternate gate-level simulation (GLS) application must be used to generate the SAIF1 file for this specific data-driven AxC. It is critical to recognize and address this essential need. The degree of optimization obtained is dependent on the ability of the data-driven AxC simulation to properly recreate the operational circuit application. It is critical to recognize that standard gate simulation tools do not account for netlist consumption at the application level. A cross-layer design, as presented in reference [5], can be implemented in both video compression and circuit applications. However, the technology described in citation [6] is specifically designed for video compression and, due to its rigid coding structure, cannot be extended to other types of accelerators. At the moment, there is no modeling generator that is not specialized for any given application. A generator of this type would be impractical to utilize in conjunction with the methods outlined in reference [7], preventing it from being effectively adapted to novel circuits or applications. This investigation elucidates the MAxPy conceptual framework. The framework, which functions independently of any particular application, is capable of efficiently generating cycle-accurate models of a design that closely approach the hardware (HW) results. The implementation of these concepts for game development in Python is simple. Participants are displaying our framework in public as part of a bigger effort to merge a diverse set of tools into a single collection. Our goal is to gain academic institution permission so that they can incorporate open-source resources such as tools for approximate logic synthesis, AxC benchmarks, and insights on approximation

arithmetic units. It is expected that academic institutions will exercise caution and attempt to incorporate these tools. In this paper, we show how to use MAxPy. This paper presents a case study that combines our current collection of approximation arithmetic units with AxLS's probability pruning technique [8]. This makes it possible to create a full Design Space Exploration (DSE) at different levels of abstraction by combining techniques for Architecture Exploration and Customization (AxC) with basic and data-driven methods. Evaluating a Sobel circuit provides an illustration of an OpenCV framework application that makes use of a closed-loop connection. There is no information in the user's content that requires scholarly reworking

RELATED WORKS In paper [9], provides a comprehensive review of various approximate parallel prefix adder designs and their performance analysis. It discusses different design methodologies, trade-offs between accuracy and area/power consumption, and compares the performance of different adder architectures. In paper [10] provides an overview of different approximate parallel prefix adder designs proposed in recent years. It covers various design techniques, such as redundant number systems, approximate carry generation, and approximate logic gates, and evaluates their impact on accuracy, area, and power consumption. In paper [11] focuses on energy-efficient approximate parallel prefix adders for approximate computing applications. It discusses different design techniques, such as voltage scaling, gate sizing, and approximate carry generation, to reduce power consumption while maintaining acceptable accuracy levels. In paper [12], proposes low-power approximate parallel prefix adder designs specifically targeted for error-tolerant applications. It explores various power reduction techniques, such as approximate carry generation, gate sizing, and transistor sizing, and evaluates their impact on power consumption and accuracy. In paper [13], investigates the design trade-offs in approximate parallel prefix adders for error-tolerant applications. It **RELATED WORKS** In paper [9], provides a comprehensive review of various approximate parallel prefix adder designs and their performance analysis. It discusses different design methodologies, trade-offs between accuracy and area/power consumption, and compares the performance of different adder architectures. In paper [10] provides an overview of different approximate parallel

prefix adder designs proposed in recent years. It covers various design techniques, such as redundant number systems, approximate carry generation, and approximate logic gates, and evaluates their impact on accuracy, area, and power consumption. In paper [11] focuses on energy-efficient approximate parallel prefix adders for approximate computing applications. It discusses different design techniques, such as voltage scaling, gate sizing, and approximate carry generation, to reduce power consumption while maintaining acceptable accuracy levels. In paper [12], proposes low-power approximate parallel prefix adder designs specifically targeted for error-tolerant applications. It explores various power reduction techniques, such as approximate carry generation, gate sizing, and transistor sizing, and evaluates their impact on power consumption and accuracy. In paper [13], investigates the design trade-offs in approximate parallel prefix adders for error-tolerant applications. It analyzes the impact of different design parameters, such as approximate logic gates, approximate carry generation, and redundant number systems, on accuracy, area, and power consumption. It also discusses the challenges and future directions in designing efficient approximate parallel prefix adders. In paper [14], focuses on efficient approximate parallel prefix adder designs specifically tailored for machine learning applications. It explores techniques such as bit-level approximation, approximate carry generation, and approximate logic gates to optimize both accuracy and power consumption for neural network computations. In paper [15], investigates the error analysis and error resilience techniques in approximate parallel prefix adders. It discusses the impact of errors on the overall system performance and proposes techniques such as error detection, error correction, and error masking to enhance the robustness and reliability of approximate adder designs. In paper [16], focuses on approximate parallel prefix adder designs specifically optimized for FPGA-based systems. It explores techniques such as resource utilization, timing constraints, and routing congestion to achieve high-performance and low-power implementations of approximate adders on FPGA platforms. In paper [17], addresses the challenges of designing reliable approximate parallel prefix adders for safety-critical systems. It discusses techniques such as fault tolerance, error mitigation, and redundancy to ensure the robustness and dependability of

approximate adder designs in applications where errors can have severe consequences. In paper [18], proposes hybrid approximate parallel prefix adder designs that combine accuracy and energy efficiency. It explores techniques such as dynamic voltage scaling, adaptive approximation, and error-aware scheduling to achieve a balance between accuracy and power consumption in approximate adder designs for energy-constrained systems. In paper [19], investigates the application of approximate parallel prefix adders in the context of quantum computing. It explores techniques such as error mitigation, noise tolerance, and fault-tolerant designs to optimize the performance and reliability of approximate adders in quantum computing architectures. In paper [20], focuses on the use of approximate parallel prefix adders in neuromorphic computing systems. It explores techniques such as spike-based computation, event-driven processing, and approximate synaptic operations to optimize the performance and energy efficiency of approximate adder designs for neuromorphic computing applications. In paper [21], addresses the challenges of designing approximate parallel prefix adders for edge computing systems. It explores techniques such as model compression, quantization, and low-bit precision arithmetic to achieve efficient and lightweight approximate adder designs for edge devices with limited computational resources. In paper [22], focuses on the application of approximate parallel prefix adders in high-performance computing systems. It explores techniques such as parallelism, pipelining, and dataflow optimization to achieve high-speed and scalable approximate adder designs for demanding computational workloads in HPC applications. In paper [23], addresses the security considerations in approximate parallel prefix adder designs for cryptographic applications. It explores techniques such as error masking, side-channel attack resistance, and secure key generation to ensure the confidentiality and integrity of cryptographic operations performed using approximate adders.

PROPOSED METHODOLOGY It is possible to conduct addition operations concurrently by using a parallel prefix adder, which is a type of digital circuit. When compared to traditional serial adders, they have higher computational efficacy and generate results faster. The development of numerous phases, each of which performs a distinct role in the addition method, is required for the construction of parallel prefix adders. The Kogge-Stone adder is a well-known

architecture that uses a tree-like structure to calculate the sum of two binary values. The technique can be divided into several consecutive stages, with the number of full adders in each stage being exactly half that of the preceding stage. As a result, the number of carry propagation stages can be reduced, resulting in a faster processing time. The Brent-Kung adder, which has a tree-like configuration, is a popular alternative design that uses fewer logic gates than the Kogge-Stone adder. This is accomplished by distributing carrying bits across multiple phases, which reduces the overall complexity of the circuit. Parallel prefix adders are frequently used in high-performance computer systems, such as microprocessors and digital signal processors, that demand fast arithmetic operations. Because of their streamlined architecture, these devices are ideal for jobs requiring significant computing power. Their efficient architecture allows for more throughput and lower latency. Parallel prefix adders are essential components of modern digital circuits. The BrentKung and Kogge-Stone adders, like the architecture under consideration, use a tree-like topology to limit the number of carry propagation stages. This design decision allows for faster addition operations. Adders are critical for increasing the speed and efficiency of computations across a wide range of computing platforms. Figure 1 shows the block diagram proposed methodology. In Modelsim, each design is structured and saved in a library. In Modelsim, the usual method for starting a new simulation is to create a functional library named "work." This is because the compiler sends the compiled design units to the library name's default location. After the units have been generated, they are integrated into the operational library. The Modelsim library format is compatible with all supported platforms. It is possible to reproduce your design on any platform; recompilation is not necessary. Begin the simulation by importing the design into the simulator. After compiling the design, it may be loaded into the simulator by executing it on a toplevel module in Verilog or an entity/architecture pair in VHDL. The simulation begins with the premise that the design restrictions are precisely implemented, and the simulation time is set to zero. Following that, the simulation is started by running the run command. Using Modelsim's comprehensive debugging environment, which enables methodical study and resolution of the identified issues, one can identify the root cause. A paper can be used to gather information about the testing or specification of a Hardware

ISSN2321-2152
www.ijmece .com
Vol 12, Issue.2 April 2024

Description Language (HDL) design. Although the completion of Modelsim projects is not needed, it may increase the program's usability. They can also be used to automate the process of file organization and define simulation parameters. The graphic below depicts the sequential procedure involved in simulating a design within a Modelsim project.

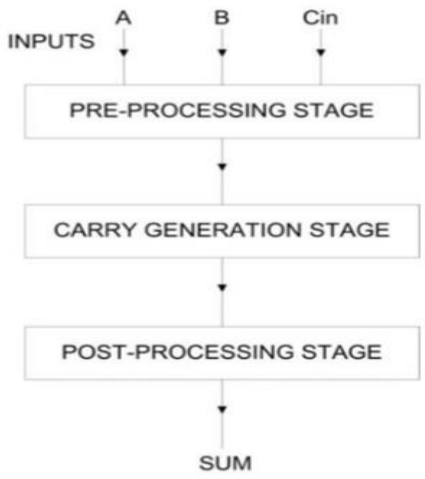

Fig: Block Diagram Proposed Methodology

CONCLUSION The newly developed approximation of the AxPPA architecture is utilized for the computation of the prefix computation phase's partial orders (POs). We conducted an evaluation of our AxPPA ideas through individual case studies, disregarding their practical applicability. After considering three widely recognized measures, namely MAE and MRED, our proposition demonstrates superior performance compared to the energy-efficient AxAs approach documented in existing literature, specifically in application-independent case studies. In the context of evaluating the trade-off between circuit space and power quality, we have conducted application-specific case studies to showcase the superior performance of our technique when integrated into an SSD video accelerator and a FIR-filter signal processing accelerator. In both instances examined, our AxPPA approach demonstrates the most cost savings in synthesis when compared to the combined AxAs solutions documented in existing literature. The AxPPA framework is designed to facilitate higher levels of approximation and adhere to rigorous quality criteria. The AxPPA-LF algorithm demonstrates a favorable equilibrium between precision and estimation, leading to notable reductions in energy consumption across many case studies, including both application-specific scenarios such as FIR and SSD, as well as application-independent instances like MAE and MRED.

REFERENCES

[1] D. Geer, “Is it time for clockless chips? [Asynchronous processor chips],” *IEEE Comput.*, vol. 38, no. 3, pp. 18–19, Mar. 2005.

[2] J. Sparsø and S. Furber, *Principles of Asynchronous Circuit Design*. Boston, MA, USA: Kluwer Academic, 2001.

[3] P. Choudhury, S. Sahoo, and M. Chakraborty, "Implementation of basic arithmetic operations using cellular automaton," in *Proc. ICIT*, 2008, pp. 79–80.

[4] M. Z. Rahman and L. Kleeman, "A delay matched approach for the design of asynchronous sequential circuits," Dept. Comput. Syst. Technol., Univ. Malaya, Kuala Lumpur, Malaysia, Tech. Rep. 05042013, 2013.

[5] M. D. Riedel, "Cyclic combinational circuits," Ph.D. dissertation, Dept. Comput. Sci., California Inst. Technol., Pasadena, CA, USA, May 2004.

[6] R. F. Tinder, *Asynchronous Sequential Machine Design and Analysis: A Comprehensive Development of the Design and Analysis of Clock-Independent State Machines and Systems*. San Mateo, CA, USA: Morgan, 2009.

[7] W. Liu, C. T. Gray, D. Fan, and W. J. Farlow, "A 250-MHz wave pipelined adder in 2-μm CMOS," *IEEE J. Solid-State Circuits*, vol. 29, no. 9, pp. 1117–1128, Sep. 1994.

[8] F.-C. Cheng, S. H. Unger, and M. Theobald, "Self-timed carrylookahead adders," *IEEE Trans. Comput.*, vol. 49, no. 7, pp. 659–672, Jul. 2000.

[9] S. Nowick, "Design of a low-latency asynchronous adder using speculative completion," *IEE Proc. Comput. Digital Tech.*, vol. 143, no. 5, pp. 301–307, Sep. 1996.

[10] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*. Reading, MA, USA: Addison-Wesley, 2005.

[11] C. Cornelius, S. Koppe, and D. Timmermann, "Dynamic circuit techniques in deep submicron technologies: Domino logic reconsidered," in *Proc. IEEE ICICDT*, Feb. 2006, pp. 1–4.

[12] M. Anis, S. Member, M. Allam, and M. Elmasry, "Impact of technology scaling on CMOS logic styles," *IEEE Trans. Circuits Syst., Analog Digital Signal Process.*, vol. 49, no. 8, pp. 577–588, Aug. 2002.