





# Evaluation of Run-Time Energy Efficiency using Controlled Approximation in a RISC-V Core

A. Delavari  F. Ghoreishy  H. S. Shahhoseini  S. Mirzakuchaki 

*School of Electrical Engineering  
Iran University of Science and Technology  
Tehran, Iran*

{arvin\_delavari, faraz\_ghoreishy}@elec.iust.ac.ir, {shahhoseini, m\_kuchaki}@iust.ac.ir

**Abstract**—The limited energy available in most embedded systems poses a significant challenge in enhancing the performance of embedded processors and microcontrollers. One promising approach to address this challenge is the use of approximate computing, which can be implemented in both hardware and software layers to balance the trade-off between performance and power consumption. In this study, the impact of dynamic hardware approximation methods on the run-time energy efficiency of a RISC-V embedded processor with specialized features for approximate computing is investigated. The results indicate that the platform achieves an average energy efficiency of 13.3 pJ/instruction at a 500MHz clock frequency adhering approximation in 45nm CMOS technology. Compared to accurate circuits and computation, the approximate computing techniques in the processing core resulted in a significant improvement of 9.21% in overall energy efficiency, 60.83% in multiplication instructions, 14.64% in execution stage, and 9.23% in overall power consumption.

**Index Terms**—Embedded systems, RISC-V, approximate computing, low-power design, energy efficient embedded systems, very large scale integration

## I. INTRODUCTION

The increasing demand for energy-efficient and low-power embedded systems has driven research efforts towards enhancing the overall performance of embedded processors, especially microcontrollers. This is an important challenge, as the limited energy available in most embedded applications imposes constraints on the design and operation of these devices. A promising approach to address this challenge is the use of approximate computing, which can be implemented in both hardware and software layers to balance the trade-off between performance and power consumption. Approximation is allowed in error-resilient applications such as image processing [1], neural networks [2], Processing in Memory (PIM) [3] and etc.

The RISC-V instruction set architecture has emerged as a popular choice for embedded processor designers, offering a flexible platform for a wide range of applications [4] [5]. The integration of RISC-V and approximate computing techniques presents an opportunity to further improve the energy efficiency and performance of embedded processors.

This study investigates the impact of dynamic hardware approximation methods on the run-time energy efficiency of a RISC-V embedded processor with specialized features for

approximate computing. The results demonstrate that the platform achieves a significant improvement in energy efficiency, area, and power consumption compared to accurate circuits and computation, highlighting the potential of this approach for enhancing the performance of embedded systems. Main contributions of this paper are as follows:

- Utilizing a RISC-V embedded processor platform Hardware/Software interface method using standard RISC-V CSRs and special programming conventions for approximate computing.
- Design of error-configurable adder/subtractor and multiplier circuit for the experimental setup.
- Analysis and evaluation of approximation and fault-injection methods efficiency at core level, in an embedded processor.

In the remainder of this paper, we will present: a review of related previous works in this subject (Section II), a description of the experimental setup used in this research (Section III), the results of the experiments and a comparison of the findings (Section IV), and an overall summary of the project along with concluding remarks (Section V).

## II. RELATED WORKS

Approximate computing has enhanced energy efficiency in many applications in recent years. There has been exploration into combining RISC-V with approximate computing methods. Software adaptations have proposed ISA extensions [6] and multi-level accuracy control mechanisms to enable this. The AxPIKE ISA simulator [7] enables incorporating hardware approximation at the instruction level to assess quality-energy trade-offs. Other studies have characterized RISC-V architecture extensions that orchestrate diverse circuit-level approximation techniques, especially approximate multipliers [8]. Customized RISC-V cores have also integrated approximate multipliers to demonstrate energy-efficient execution of neural networks [9].

There have been investigations on the advantages of multi-core systems that combine exact and approximate compute cores [10]. This heterogeneous approach aims to leverage the benefits of approximation while maintaining the precision of exact computation. [11] proposes an approach to improve the

performance of multi-core systems by configurable approximate arithmetic logic units, with a machine learning based framework for online power regulation and quality monitoring to adjust the frequency and precision of the arithmetic units and maximize performance.

The proposed design here involves a RISC-V core where each execution unit can dynamically switch between accurate and approximate circuits and also if applicable control the error level of integrated approximate execution units. This combines the efficiency gains of dynamic approximation with the precision of exact computation all within a single reconfigurable core.

### III. EXPERIMENTAL SETUP

The considered platform for these experiments is an embedded processor named phoenix [12] which supports RV32IEM standard instruction set of the open-source and license-free RISC-V ISA. The chosen core is a 3-stage pipelined scalar processor with specialized and novel features for approximate computing, in order to be a general platform for fault-tolerant applications and approximation-enhanced low-power design studies. In this study, “fault-tolerant” refers to applications that can accommodate errors (specifically approximations) resulting from their functions and data. Pipeline stages are considered as Instruction Fetch (IF) and Decode (ID), Execution (EXE) and Memory access with Write-back (MEM/WB). The high-level architecture of the processor’s execution stage within the pipeline is illustrated in Fig. 1.

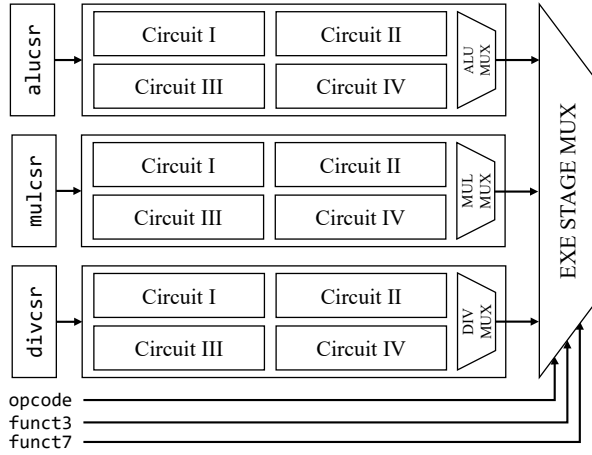


Fig. 1. Execution stage high-level block diagram of the processor including accurate and approximate arithmetic circuits

Execution stage consists of Arithmetic Logic Unit, Multiplier Unit and Divider Unit due to the support of RV32IEM instruction set. These three computational units share a unique method for adhering and controlling approximation in circuit-level. The approximation level is determined from a customized special-purpose control status registers addressed in  $0x800$ ,  $0x801$  and  $0x802$  in standard RISC-V CSR addressing space [5]. Each execution unit can host four arithmetic circuit

(accurate or approximate) supported by the standard signaling conventions of the processor’s execution engine. The execution unit’s active circuit and target result along with approximation control logic for each is determined by the value in the CSRs, which the decoding and framing is explained further in this paper.

These CSRs are named `alucsr`, `mulcsr` and `divcsr`, designed for enabling and controlling approximation level of the dynamically configurable circuits. Each field of CSRs is responsible for a feature in approximation control and advancement. Bit 0 is for enabling approximation, in which approximate arithmetic is used when signal value is 1, and will perform accurate arithmetic with value of 0. Bits 2 to 1 are for determination of selected circuit, for controlling result multiplexer and switching off unused circuits. Bits 3 to 7 are used for dynamic truncation control in arithmetic circuits which may support this technique. Bits 8 to 11 and bits 12 to 15 are also custom fields of control status registers for designer-defined control logic and features. In the end, bits 16 to 31 are responsible for dynamic error control in case of integration of error controllable arithmetic circuits within the core.

Each execution unit can embed up to 4 arithmetic circuit with platform’s special conventions for circuit integration. Circuits can be accurate, approximate with static configurations and error, or dynamic error-configurable such as selected circuits in this study. The platform can adhere and synchronize execution process with different circuit which may vary in timing and other hardware specifications. Block diagram of the execution units’ structure is shown in Fig. 2. In this experiment an error-configurable multiplier, alongside an accurate multiplier, and fully accurate division circuit is considered, because of the importance of multiplication and accumulation in most computational workloads, including fault-tolerant applications. The accurate multiplier is positioned in the default slot (Circuit I), while the approximate multiplier is placed in the secondary slot (Circuit II). The remaining slots are unutilized and remained empty in this configuration.

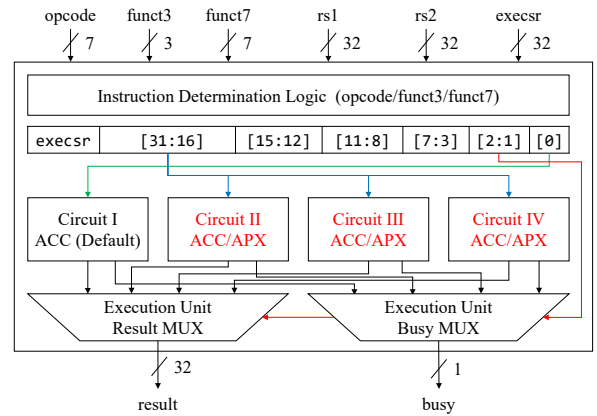


Fig. 2. Error control, circuit switching, and hardware approximation level determination based on custom CSR values in EXE stage of processor

After determination of the operating circuit through the respected CSRs of each execution unit, the other circuits will be switched off in order to exclude redundant processing of data and unnecessary static and dynamic power consumption. It is important to note that all execution units including adders in ALU Arithmetic Logic Unit, Multiplier Unit and Divider unit are capable of approximation features explained, except for addition required in address generation and other control flow instructions which uses an accurate adder because approximation is not allowed in such operations.

#### A. Dynamic Accuracy-Controllable Carry Select Adder

The adder/subtractor integrated within the processor, is a fast and low power carry select adder, with improved performance by decreasing the critical path in comparison with conventional design. Also, the proposed adder, depicted in Fig. 3, is an error-controllable design which can be used in different applications due to controllable accuracy of the results. The adder inside arithmetic logic unit is responsible for addition and subtraction of I/E extension of RISC-V instruction set architecture. In the experiments, the adder was set to default mode which enables accurate addition. In fact, the proposed adder can perform both accurate and approximate addition in a single circuit, without any need of additional hardware for circuit switching in the execution.

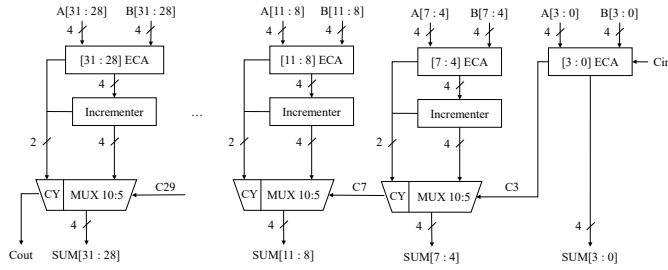


Fig. 3. Error controllable low-power and fast carry select adder

The ECA shown in Fig. 3 is a 4-bit error controllable ripple carry adder, in which the fundamental full-adders have an additional control signal in comparison with conventional full adder. This full adder as shown in Fig. 4, can perform accurately when the error control signal is high (logical value 1), but may produce inaccurate result by decreasing or increasing carry value when the signal is low. When using approximation, the error in the final result is applied unevenly, as in half the input scenarios, the approximate result is lower than the accurate result by one, and in the other half, the approximate result is higher than the accurate result by one.

#### B. Dynamic Accuracy-Controllable Array Multiplier

For this setup, an original low-power approximate 8x8 unsigned multiplier with reduced circuitual path and area in comparison with a conventional Wallace tree multiplier is designed. The design also incorporates the full-adder presented in Fig. 4. The unbiased error of the adder (having both positive

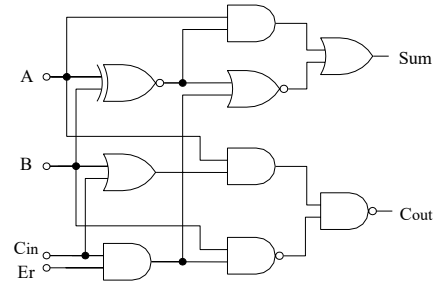


Fig. 4. Proposed full-adder with error signal used in 4-bit ripple carry adders of the CSA circuit

and negative error in results, in error enabled modes) improves the multiplier with Error Rate (ER) and Mean Relative Error Distance (MRED) [13] [14] in term of accuracy. The full-adder illustrated in Fig. 4, is integrated in the final addition stage, after partial product generation and compression stages of Wallace multiplier which is shown in Fig. 5.

The ripple carry adder's error in the final stage can be controlled using the error control signals of the full-adders. 7-bits of the 12-bit addition is controllable, which gives 128 configurations and error levels in one circuit, which means 128 configurations of dynamic power consumption. The mode with all error control signals equal to zero is the configuration with lowest power consumption, which also has the highest application-level quality in error (ER and MRED). .

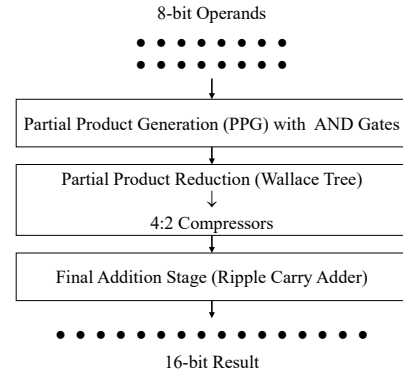


Fig. 5. 8-bit multiplier's high level block diagram

The proposed Multiplier has an area of  $269\mu\text{m}^2$  with a range of power between  $70.2\mu\text{W}$  and  $101.3\mu\text{W}$  in different configurations. Fig. 6 shows ER and MRED for all configurations of the proposed design.

Table I shows an analysis in terms of hardware (power, delay and area) and accuracy efficiency. The proposed design showcases the best result in terms of critical path delay and area occupation. Additionally, in one of the configurations the design has a power consumption of  $70.2\mu\text{W}$  which is the lowest in the comparison. The design in [10] produces the best results in accuracy efficiency criteria.

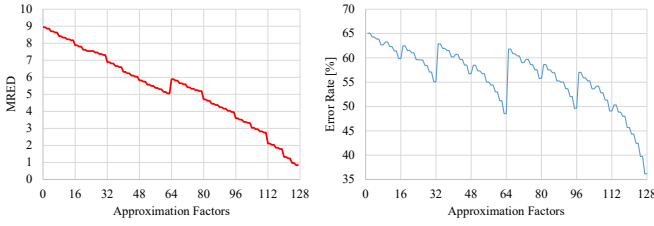


Fig. 6. Error Rate and Mean Relative Error Distance (MRED) analysis of the proposed multiplier in all dynamic configurations (128 Approximation Factors)

TABLE I  
HARDWARE - ACCURACY EFFICIENCY COMPARISON OF PROPOSED MULTIPLIER AND SIMILAR APPROXIMATE MULTIPLIERS

Multipliers	Power ( $\mu$ W)	Delay (ns)	Area ( $\mu$ m <sup>2</sup> )	ER (%)	MRED (%)
Proposed Design	70.2 - 101.3	0.64	269.6	36.16 - 65.06	0.85 - 8.94
[15]	81.2	0.64	27.8.8	32.42	0.18
[16]	156.7	0.80	323.9	48.35	1.26
[16]	143.5	0.77	301.6	69.73	2.83

### C. Hierarchical 32×32 Approximate Multiplier Design

For final integration of multiplier within the core, a 32-bit multiplier architecture consisting of several 8-bit modules is designed. The method used for creation of the 32-bit architecture is a hierarchical configuration of 8-bit multipliers. The original 8-bit multiplier is used over multiple cycles to perform 16-bit multiplication. This hardware is replicated four times to perform a 32-bit multiplication. There is also an accurate multiplier integrated within the core in order to have a comprehensive comparison in term of power and energy efficiency gains, using dynamic circuit switching feature which is presented in the paper. The accurate multiplier circuit is an original 32-bit multiplier implemented with DesignWare software, offered by Synopsys Design Compiler.

## IV. RESULTS AND COMPARISON

The processor and its approximate and accurate execution units are implemented in 45nm CMOS technology using Design Compiler, enabling a clock frequency of 500MHz which is considered for this set of experiments. All power and energy consumption data reported in this paper are extracted with post-synthesis simulation based on switching activity files.

Before delving into the main experiment and comparisons, Fig. 7 is shown for an estimation of the power distribution in the processor, excluding memories and register files.

It can be concluded from Fig. 7 that the execution stage has the highest power consumption in the inner core microarchitecture. In the target processor, the execution stage, including the arithmetic logic unit, multiplier unit (with 2 multipliers), and divider unit, have the largest share of the power distribution

chart, accounting for 80%. The results provide motivation for using approximation techniques not just in software or memory approaches, but also at the circuit level. The execution stage, excluding memories and register files, consumes the most power within the inner modules. This is the reason for the focus of this study, the impact of user-controlled circuit level approximation on the overall performance of a processor.

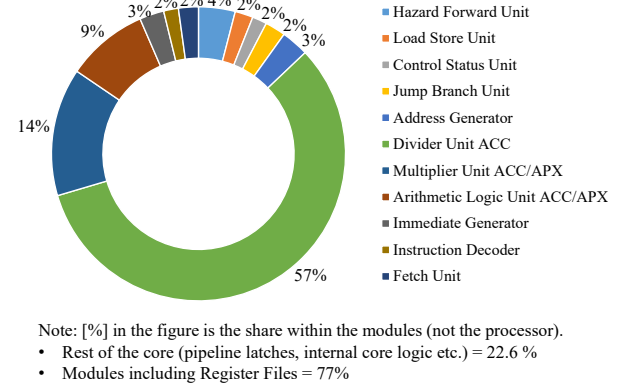


Fig. 7. Modules power distribution in the processor's inner modules excluding memories and register files

Fig. 8 presents a detailed breakdown of instruction types following compilation. The average code size is comprised of 221992 instructions and utilization of multiplication is 242 instructions per code in average. Multiplication instructions are 6.47% of arithmetic instruction (addition/multiplication/division) in average. Approximation is enabled exclusively for multiplication instructions, specifically the mul and mulh instructions. Both of these instructions are classified as R-type, meaning they operate with registers without involving any immediate values. They utilize the same circuitry, which is determined by the mulcsr setting in the program. In contrast, addition instructions (add and addi) operate in accurate mode.

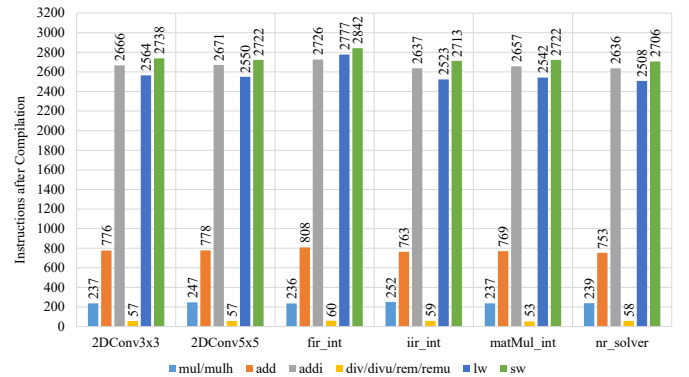


Fig. 8. Number of arithmetic and memory accessing instructions after code compilation using GCC compiler [17] with constant options

There is a selected set of standard applications using a notable load of multiplication within the calculation process.

Some of these applications are regarded as fault-tolerant applications due to their inherent resilience against error in accumulative workload. As an example, 2D convolutions are used in image processing applications and filters, which are perfect examples of applications allowed to adhere approximation. All of the programs are compiled using standard GCC-GNU [17] compiler for RISC-V with similar compilation options. Duration for execution of each code and the instruction count is measured using RISC-V standard CSRs in order to calculate energy efficiency of the processor in each application. Energy efficiency in this study is regarded as the consumed power in a limited time period per instructions executed in the program.

Fig. 9 illustrates energy efficiency in each program in pJ/instructions using both accurate and approximate multipliers. Circuit switching was done utilizing the `mulcsr` circuit selection field. The approximate multiplier is set to a default accuracy with error control field of CSR equal to `0x7E`. Fig. 10 shows the impact of approximate multiplication on total power consumption of the embedded processor. In every application, the approximate multiplication has decreased overall power consumption. The conserved power in each application highlights the efficiency of approximation techniques in scenarios where a reasonable margin of calculation error is permissible. The selected error-level (`0x7E`) of the multiplier has an MRED of 0.85% and ER of 36.2%.

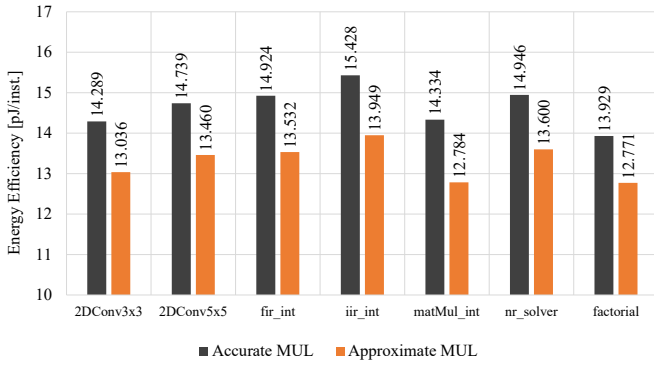


Fig. 9. Energy efficiency comparison of applications executed with accurate and approximate multiplication

Fig. 10 shows the impact of approximate multiplication on total power consumption of the embedded processor. In every application, the approximate multiplication has decreased overall power consumption. The saved power in each application showcases the effectiveness of approximation in applications where error in calculations are allowed by a reasonable margin. The selected error-level of the multiplier has an MRED of 0.85% and ER of 36.2%.

Overall improvements in term of power consumption are illustrated in Fig. 11. This analysis was carried out from 3 separate perspectives: Total improvement (processor's power consumption), Execution stage of the processor, and the Multiplier Unit (multiplication operation solely). The percentage of improvement in each considered hardware is written next to

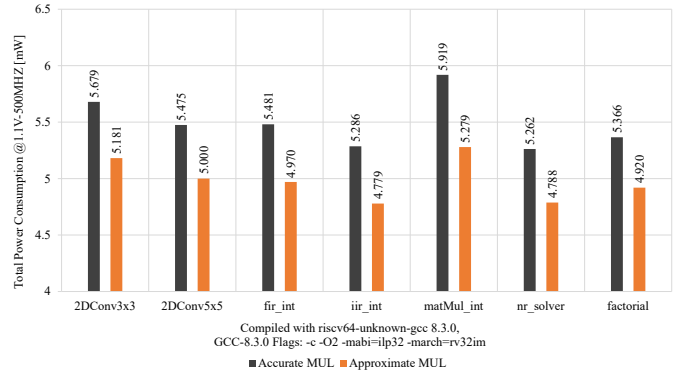


Fig. 10. Overall power consumption comparison of the processor, executing applications using accurate and approximate multiplication

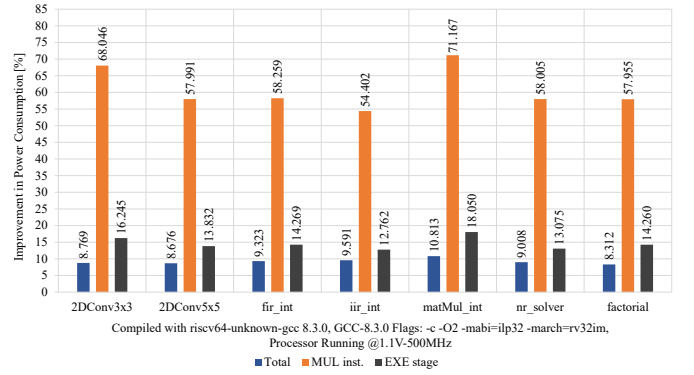


Fig. 11. Power consumption improvements by approximation, in the processor's overall power, execution stage and multiplication instructions

each column. Power consumption in each execution unit and the execution stage of the core are presented in Table II and Table III in both accurate and approximate modes.

From the results, it is indicated that there is an average improvement of 9.213% in energy-efficiency in comparison with accurate calculation. By selecting the desired circuit in each execution unit though CSRs, the other circuits will be turned off, decreasing the dynamic and static power consumption of unused units near to zero. The highest improvement is `formatMul_int` with 10.813% percent with a power consumption of 5.279mW with approximate multiplier.

The overall power consumption during the execution stage of the pipelined processor is reduced by 18.05% through the use of approximate multiplication, with the multiplication process alone achieving a significant improvement of 71.16% within this stage. The most significant improvement in these metrics is resulted from `thematMul_int` and `2Dconv3x3`.

## V. CONCLUSION

This study examines the impact of dynamic hardware approximation on the energy efficiency of a RISC-V embedded processor with specialized capabilities for approximate computing. The findings demonstrate an average 9.21% improvement in the processor's run-time energy efficiency, along



TABLE II  
POWER CONSUMPTION OF EXECUTION UNITS IN ACCURATE  
MULTIPLICATION MODE

Applications	Power Consumption [mW]			
	MUL	ALU	DIV	EXE Stage
2DConv3x3	0.435	0.626	0.601	1.662
2DConv5x5	0.438	0.627	0.605	1.670
fir_int	0.448	0.598	0.622	1.668
iir_int	0.443	0.592	0.634	1.669
matMul_int	0.437	0.646	0.507	1.590
nr_solver	0.431	0.589	0.632	1.652
factorial	0.440	0.593	0.615	1.648

TABLE III  
POWER CONSUMPTION OF EXECUTION UNITS IN APPROXIMATE  
MULTIPLICATION MODE

Applications	Power Consumption [mW]			
	MUL	ALU	DIV	EXE Stage
2DConv3x3	0.139	0.638	0.615	1.392
2DConv5x5	0.184	0.636	0.619	1.439
fir_int	0.187	0.607	0.636	1.43
iir_int	0.202	0.605	0.649	1.456
matMul_int	0.126	0.655	0.522	1.303
nr_solver	0.181	0.6	0.655	1.436
factorial	0.185	0.602	0.626	1.413

with a 14.64% reduction in overall power consumption during the execute stage of the scalar core. These results confirm that approximate computing, particularly through hardware-approximated arithmetic circuits, is a promising approach for dynamic energy management in embedded processors. The research highlights the potential of such techniques to optimize power usage without compromising computational performance, making them particularly valuable in resource-constrained environments. This study further reinforces the growing relevance of hardware approximation in addressing energy efficiency challenges in modern embedded systems.

The results highlight the potential of integrating RISC-V processors and approximate computing for enhancing the performance of energy-constrained embedded systems in execution process. The ability to dynamically adjust approximation offers a flexible approach to manage accuracy and efficiency. Future work could explore extending these techniques across the processor architecture and to a wider range of embedded applications. Further hardware optimization and advanced error management could lead to even greater efficiency gains. Automating the process of accuracy and energy control using real-time monitoring systems within the processor can be a promising solution to address the accuracy/efficiency trade-off in embedded cores. Overall, this study demonstrates the sig-

nificant outlook for RISC-V embedded processors leveraging approximate computing.

#### ACKNOWLEDGMENT

The authors would like to express their appreciation to A. M. H. Monazzah for their valuable insights, which have significantly enhanced the quality of this paper.

#### REFERENCES

- [1] G. Anusha and P. Deepa, "Design of approximate adders and multipliers for error tolerant image processing," *Microprocessors and Microsystems*, vol. 72, p. 102940, 2020.
- [2] S. Shakibhamedan, N. Amirafshar, A. S. Baroughi, H. S. Shahhoseini and N. Taherinejad, "ACE-CNN: Approximate carry disregard multipliers for energy-efficient CNN-Based image classification," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2024.
- [3] S. Bavikadi, P. R. Sutradhar, M. Indovina, A. Ganguly and S. M. P. Dinakarrao, "ReApprox-PIM: Reconfigurable approximate Look-Up-Table (LUT)-Based Processing-in-Memory (PIM) machine learning accelerator," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2024.
- [4] A. Waterman and K. Asanovic. (2021). "The RISC-V Instruction Set Manual—Volume II: Privileged Architecture," [Online]. Available: <https://riscv.org/specifications/privileged-isa/>
- [5] E. Cui, T. Li and Q. Wei, "RISC-V Instruction Set Architecture Extensions: A Survey," in *IEEE Access*, vol. 11, pp. 24696-24711, 2023.
- [6] T. Trevisan J et al., "Approxrisc: An approximate computing infrastructure for RISC-V," *RISC-V Workshop in Barcelona*, May 2018, poster.
- [7] I. Felzmann, J. F. Filho and L. Wanner, "AxPIKE: Instruction-level Injection and Evaluation of Approximate Computing," *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Grenoble, France, 2021, pp. 491-494.
- [8] A. Verma, P. Sharma and B. P. Das, "RISC-V Core with Approximate Multiplier for Error-Tolerant Applications," *2022 25th Euromicro Conference on Digital System Design (DSD)*, Maspalomas, Spain, 2022, pp. 239-246.
- [9] F. Guella, E. Valpreda, M. Caon, G. Masera and M. Martina, "MARLIN: A Co-Design Methodology for Approximate Reconfigurable Inference of Neural Networks at the Edge," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, doi: 10.1109/TCSI.2024.3365952.
- [10] A. S. Baroughi, S. Huemer, H. S. Shahhoseini and N. TaheriNejad, "AxE: An approximate-exact multi-processor system-on-chip platform," *2022 25th Euromicro Conference on Digital System Design (DSD)*, pp. 60-66, 2022.
- [11] S. A. K. Gharavi and S. Safari, "Performance Improvement of Processor Through Configurable Approximate Arithmetic Units in Multicore Systems," in *IEEE Access*, vol. 12, pp. 43907-43917, 2024.
- [12] A. Delavari, F. Ghoreishy, H. S. Shahhoseini and S. Mirzakuchaki, "A Reconfigurable Approximate Computing RISC-V Platform for Fault-Tolerant Applications," *2024 27th Euromicro Conference on Digital System Design (DSD)*, Paris, France, 2024, pp. 81-89.
- [13] A. G. M. Strollo, E. Napoli, D. D. Caro, N. Petra, and G. D. Meo, "Comparison and extension of approximate 4-2 compressors for lowpower approximate multipliers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 9, pp. 3021-3034, Sep. 2020.
- [14] P. Yin, C. Wang, H. Waris, W. Liu, Y. Han, and F. Lombardi, "Design and analysis of energy-efficient dynamic range approximate logarithmic multipliers for machine learning," *IEEE Trans. Sustain. Comput.*, vol. 6, no. 4, pp. 612-625, Oct. 2021.
- [15] N. Amirafshar, A. S. Baroughi, H. S. Shahhoseini and N. TaheriNejad, "An Approximate Carry Disregard Multiplier with Improved Mean Relative Error Distance and Probability of Correctness," *2022 25th Euromicro Conference on Digital System Design (DSD)*, Maspalomas, Spain, 2022, pp. 46-52.
- [16] H. Waris, C. Wang, W. Liu, J. Han and F. Lombardi, "Hybrid Partial Product-Based High-Performance Approximate Recursive Multipliers," in *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 1, pp. 507-513, 1 Jan.-March 2022.
- [17] RISC-V Collaboration, "GNU toolchain for RISC-V, including GCC." [Online]. Available: <https://github.com/riscv-collab/riscv-gnu-toolchain>