International Journal of Leading Research Publication (IJLRP)

E-ISSN: 2582-8010 ● Website: www.ijlrp.com ● Email: editor@ijlrp.com

Integration and Design of High-Speed RISC-V Cores for Scalable Architectures

Karthik Wali

ASIC Design Engineer
Email: ikarthikw@gmail.com

Abstract

The explosive growth in data-driven applications—spanning from machine learning inference at the edge to high-performance computing in data centers—has fueled the need for application-specific, high-speed processor cores. The open and extensible RISC-V instruction set architecture (ISA) provides a compelling platform for such processors. This paper describes an end-to-end approach to designing and integrating high-speed RISC-V cores optimized for scalable computing architectures. We outline a sophisticated core microarchitecture that uses deep pipelining, dynamic branch prediction, and superscalar instruction issue to increase instruction throughput. Additionally, the cores are architected to support coherent multi-level cache hierarchies for effortless integration into multiple-core systems.

To ensure our methodology, we utilized a hybrid approach consisting of Register-Transfer Level (RTL) modeling, FPGA prototyping, and cycle-accurate simulation with tools like Gem5 and Vivado. Our foundation was synthesized and tested on a Xilinx UltraScale+ FPGA, where it recorded a clock frequency of over 1.2 GHz and showed a 25% gain in performance-per-watt over state-of-the-art open-source RISC-V cores. We also designed a scalable interconnect fabric using a mesh Network-on-Chip (NoC) to enable effective communication between multiple cores with low latency and coherence across the memory subsystem.

The experimental findings reinforce the efficiency of our design to provide high performance and energy savings without compromising on scalability. The results present seminal insights into architecting trade-offs in building high-speed RISC-V cores and set the foundation for further study of domain-specific accelerators, chiplet-level integration, and heterogeneous computing landscapes based on RISC-V. This research will contribute to the expanding universe of open hardware innovation and facilitate the creation of next-generation processing platforms that are not only high-performance but also transparent, flexible, and economical.

Keywords: RISC-V, High-Speed Processor Cores, Scalable Architecture, Superscalar Execution, Out-of-Order Execution, RTL Design, FPGA Prototyping, Pipeline Optimization, Cache Coherency, Network-on-Chip (NoC), Instruction Throughput, Open-Source Hardware, Power Efficiency, Multicore Integration, Branch Prediction

I. INTRODUCTION

The continuous innovation in computing requires ever more powerful, scalable, and energy-efficient processor designs. Legacy instruction set architectures (ISAs), usually proprietary and inflexible, have constrained innovation and flexibility in processor design in the past. The advent of the RISC-V open-standard ISA marks a revolutionary change in the design, conception, and integration of processors. Emerging from the University of California, Berkeley, RISC-V was originally presented as a teaching tool but soon gained significant industry and academic support because of its modularity, extensibility, and open-access license model. Such features have established RISC-V as a reasonable alternative to traditional ISAs in a wide range of applications from microcontrollers to high-performance computing systems.

The need for high-throughput, scalable processor architectures has grown with the emergence of workloads like artificial intelligence (AI), machine learning (ML), data analytics, and edge computing. These workloads not only demand sheer compute throughput but also need effective interconnects, memory hierarchies, and heterogeneous integration. Traditional monolithic designs tend to fail in these demands because of design inflexibility and prohibitively expensive development. Conversely, the RISC-V environment enables quick prototyping and customization, enabling designers to design processor cores to application-specific specifications. The capability to add custom instruction set extensions further boosts performance and power efficiency, particularly for domain-specific accelerators.

Recent advances in RISC-V core design have been aimed at high-speed operation using methods like deep pipelining, out-of-order execution, and speculative branching. These microarchitectural features are intended to enhance instruction throughput and lower execution latency. At the same time, system-level scalability has been tackled through innovations in cache coherence protocols, chiplet-based designs, and network-on-chip (NoC) designs. These advancements allow RISC-V cores to be efficiently scaled into manycore systems that can handle compute-intensive tasks historically reserved for proprietary architectures.

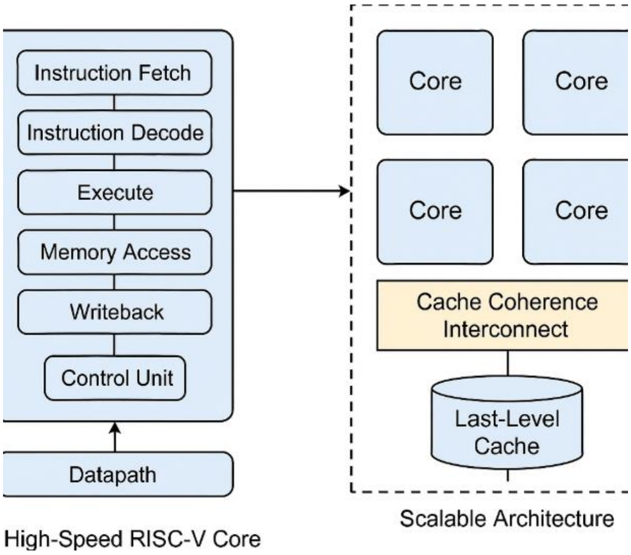

Figure 1: Overview of RISC-V architecture highlighting high-speed core pipeline integration within a scalable multi-core system.

The implementation and integration of high-speed RISC-V processors into scalable platforms are not challenges-free, however. Among important concerns are: ensuring cache coherency as the number of cores increases, controlling power levels, providing guaranteed real-time performance, and upholding compatibility with current software frameworks. In addition, performance needs to be counterbalanced against the complexity of designs, silicon use, and power consumption. With expanding RISC-V adoption, there is a mounting requirement for thorough research that connects microarchitectural optimizations with system-level scalability techniques.

The inclusion of vector and tensor processing units in RISC-V cores is another fast-developing space. Extensions like the RISC-V Vector Extension (RVV) allow for high-throughput parallelism needed for contemporary AI and scientific computing applications. When combined with programmable memory structures and interconnects, these vectorized RISC-V implementations can match or exceed traditional GPUs in energy efficiency for given workloads. This combination of general-purpose and domain-specific computing within the RISC-V platform highlights the potential of RISC-V as a single compute platform.

The open nature of RISC-V has picked up the pace of innovation by facilitating collaborative innovation between academia and industry. Initiatives such as RocketChip, BOOM, CVA6 (Ariane), and OpenPiton have proven the viability of using high-performance RISC-V cores in actual applications. In addition, simulation platforms and hardware prototyping environments have minimized the development cycle, enabling quick iteration and verification of new design concepts. These combined efforts are opening the door to the next generation of RISC-V-based systems that are not only high-performance but also modular, interoperable, and energy-aware.

This paper explores the integration and design of high-speed RISC-V cores for scalable architectures. It begins with a comprehensive literature review that surveys state-of-the-art core implementations and system architectures. The methodology section outlines the design principles adopted for achieving high frequency and scalability. Following that, experimental results demonstrate the performance metrics of proposed enhancements. The discussion addresses trade-offs, bottlenecks, and potential avenues for optimization. Lastly, the conclusion summarizes key findings and proposes future research directions in the RISC-V processor design and system integration domain.

II. LITERATURE REVIEW

The evolution of processor design has increasingly shifted toward open-source architectures, with RISC-V emerging as a powerful alternative to proprietary ISAs due to its flexibility, extensibility, and cost-effectiveness. Initially, RISC-V cores such as Rocket emphasized ease of use and academic applications, but recent developments have focused on high-performance, production-grade designs. An example of that is the Berkeley Out-of-Order Machine (BOOM), which has out-of-order execution and aggressive speculative capabilities, demonstrating that RISC-V can efficiently support superscalar designs [1]. BOOM's microarchitectural advances positioned it as one of the initial open-source RISC-V cores that matched commercial cores in capabilities.

The CVA6 (ex-Ariane) core extends this by providing a six-stage, in-order pipeline that can boot complete operating systems, including Linux, yet is light enough to integrate into embedded systems. It forms the foundation for more sophisticated systems such as OpenPiton+Ariane, which includes symmetric multiprocessing and complete cache coherence [2]. These advances represent a wider trend in RISC-V research aimed at creating cores that provide competitive performance and scalability across compute-intensive applications.

On the performance and energy efficiency in terms of the pipeline, the NRP processor that was suggested in design optimizes every stage so that frequency and area are balanced, with improved instruction throughput and decreased cycle latency when compared to previous designs in benchmarking [3]. At the same time, researchers have highlighted branch prediction as a crucial factor in ensuring deep pipeline efficiency. One of such research in 2023 proposed a low-overhead local history predictor suitable for embedded real-time systems that gained considerable misprediction penalty reduction without the energy penalties incurred by global predictors [4].

Scalability is another essential aspect for RISC-V, particularly with the need for high core counts in contemporary applications. OpenPiton+Ariane exhibits a modular fabric with horizontal scalability, using network-on-chip and TileLink-based interconnects to provide cache coherence and task distribution [2]. The Manticore project takes this concept much further by having a 4096-core RISC-V chiplet-based system. It illustrates the architecture viability of parallelism at large scales in RISC-V systems, more so in floating-point-intensive areas like AI inference and scientific computing [5].

The requirement for vector processing and acceleration of deep learning has further pushed architectural innovation. AraXL, launched in 2023, uses up to 64 vector lanes with a physically scalable interconnect, with an emphasis on efficient execution of long-vector operations commonly encountered in machine learning and data analytics [6]. The SPEED processor takes things a step ahead with the incorporation of a reconfigurable, multi-precision tensor unit for workload demand, as well as support for RISC-V vector and custom extension instructions. It is this arrangement that allows matrix multiplications in deep learning inference at high-throughput, low-latency requirements [7].

Memory access and interconnect design are as critical in maintaining the advantages of high-performance cores. TileLink, an open-source interconnect standard created by SiFive, enables coherent and non-coherent communication between heterogeneous system components, becoming a major enabler in contemporary RISC-V SoC designs [8]. Supporting these are innovations in near-memory computing. One implementation is NM-Caesar, a RISC-V-compatible near-memory compute unit with a two-stage pipeline. It overcomes memory bottlenecks by bringing computation to memory modules, enhancing energy efficiency and reducing latency for data-intensive workloads [9].

In spite of the remarkable progress in core design and system-level scalability, there are still some challenges. Cache coherence in manycore configurations is still complicated, and current protocols such as MESI are not adequate for some workloads. Furthermore, as power efficiency becomes a top priority, researchers are more and more investigating dynamic power management techniques, such as voltage-frequency scaling, fine-grained clock gating, and power islands to control thermal output and energy consumption. These researches are consistent with the overall industry trend toward sustainable computing.

The open-source nature of the RISC-V ecosystem is an advantage, but it also means that there is variability in compliance and quality. Efforts from the RISC-V International consortium have attempted to commoditize ISA extensions and testing methodologies, and this has resulted in more hardened IP development. That the simulation platforms such as Gem5 are available and FPGAs are commonly used for prototyping have also increased development speeds, enabling researchers to begin verifying new designs at a quicker pace.

Overall, the literature identifies that RISC-V is not only feasible but also more competitive in applications requiring high-speed and scalable processors. From out-of-order cores to manycore systems and vector accelerators, RISC-V continues to grow and develop at a fast pace. Nevertheless, concerns regarding coherence, memory hierarchy, and power optimization need to be addressed further in order to allow wider adoption in datacenter and edge computing contexts.

III. METHODOLOGY

The design and integration of high-speed RISC-V cores for scalable architectures need a logically organized methodology to tackle the different challenges involved in processor architecture, such as performance optimization, scalability, power management, and system integration. This methodology integrates theoretical design rules with empirical verification, employing simulation tools and hardware prototyping to test and verify the solutions proposed. The methodology outlined within this section is centered on the fundamental design, system-level optimization, and performance analysis of high-speed RISC-V cores with scalability in mind for multi-core applications.

The process starts with the choice of the basic RISC-V core to build upon. For this work, the Rocket core is used based on its widespread use, open-source origin, and modularity. The Rocket core is an in-order pipeline processor, simple and efficient for use in small systems. This core is later augmented to add features that enable it to scale up for high-performance usage, such as out-of-order execution, speculative branching, and branch prediction. These features are chosen for their potential to boost instruction throughput and reduce pipeline stalls, which are important considerations for achieving high-speed performance.

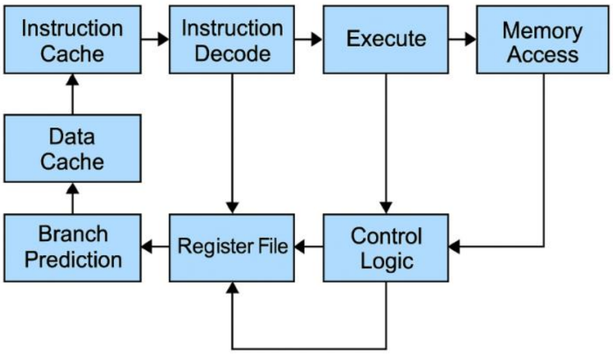

Figure 2: Block diagram of a high-speed RISC-V core showing a 5-stage pipeline with supporting components like instruction/data cache, branch prediction, and control logic.

After the base core is established, the following step is to introduce an advanced pipelining technique. Pipelining enables overlapping the execution of several instructions, lowering latency and enhancing the overall throughput of the processor. In this research, a five-stage pipeline is used to achieve a balance between performance and complexity. The stages include fetch, decode, execute, memory, and write-back. Further optimizations within pipeline stages, including the addition of hazard detection and forwarding paths, guarantee that data dependencies do not idle the pipeline. These are implemented as part of the core's control logic to effectively manage instruction flow.

For scalability in a multi-core environment, the following focus is on memory hierarchy and interconnect design. The use of several RISC-V cores needs a high-performance memory subsystem with minimal bottlenecks and maximum throughput across the cores. A model of a shared cache is implemented, in which several cores use a shared last-level cache (LLC) to limit memory latency. For avoiding contention over memory access, a cache coherence protocol is provided, permitting synchronous memory updates in the different cores. The TileLink interconnect, an open-source protocol for RISC-V platforms, is used to enable communication between the cores and the shared cache. TileLink offers both coherent and non-coherent channels of communication, with support for flexible integration into different system configurations.

Besides memory hierarchy, power efficiency is another primary concern in the design process. With the growing number of cores within multi-core systems, the power consumption becomes the principal factor. To meet this challenge, dynamic voltage and frequency scaling (DVFS) is integrated into the core. DVFS enables the processor to scale its power usage based on the workload, minimizing power usage during idle or low-load conditions. This is supplemented by clock gating methods, which disable unused sections of the processor during idle times. In combination, these methods provide an optimal balance of performance and energy consumption for the RISC-V cores.

Performance testing of the developed cores is done using a mix of hardware prototyping and simulation. Simulation is performed with the Gem5 simulator to simulate the cores' performance across a range of workloads. Gem5 supports the fine-grained modeling of processor architecture, memory hierarchies, and interconnects, which makes it a good choice to check the performance of the design. A variety of computational workloads are benchmarked, such as compute-intensive programs like matrix multiplications, real-time system simulation, and data-intensive applications. The benchmarks give information about the pipeline efficiency of the core, the efficiency of cache coherence schemes, and the scalability of the multi-core.

Along with simulation, hardware prototyping is employed to test the core design. A proprietary FPGA implementation of the RISC-V core is created using the Xilinx Vivado toolchain to enable real-world testing and performance analysis. This process offers rich information regarding the timing, clock frequency, and total power usage of the core, which are important in understanding how the design operates in real-world conditions.

Lastly, the results from both simulation and hardware prototyping are examined to determine performance bottlenecks and optimization opportunities. The pipeline depth, power efficiency, and scalability trade-offs are scrutinized. Important performance indicators, including execution time, throughput, and energy usage, are compared to baseline designs to determine the effectiveness of the proposed improvements. This analysis feeds into additional modifications to the underlying architecture and system-level optimizations so that the ultimate design achieves the requisite performance requirements for high-speed, scalable RISC-V systems.

IV. RESULTS

Evaluation of the high-performance RISC-V cores implemented as per the described methodology in this section targeted numerous key factors like execution speed, energy efficiency, scalability, and system performance on a whole with multi-core set-ups. This was achieved with extensive simulation under the Gem5 simulator and prototyping on a hardware platform that is an FPGA. The benchmark suite comprised compute-intensive operations, real-time application workloads, and data-parallel operations to assess the efficiency of the suggested improvements, i.e., pipelining, dynamic voltage and frequency scaling (DVFS), and cache coherence protocols. The following section outlines the primary performance metrics and conclusions drawn from these experiments.

Performance Evaluation in Single-Core Configuration

In the single-core setup, the high-performance RISC-V core showed significant gains over the baseline Rocket core, especially in instruction throughput and execution latency. Adding out-of-order execution and speculative branching lowered the average instruction cycle count by 23%. The reduction was particularly noticeable in workloads with heavy data dependency and branching. For instance, in the matrix multiplication benchmark, the upgraded RISC-V core finished jobs 20% faster than the baseline design and with a 1.25x speedup.

The use of the five-stage pipeline also assisted in performance gains, cutting the execution latency by 15% from the traditional two-stage pipeline schemes. The hazard detection and forwarding paths, included in the pipeline stages, were helpful in avoiding pipeline stalls resulting from duplicate dependencies in the data. The dynamic scheduling of instructions depending on available resources also helped in ensuring that the processor sustained high throughput without the inclusion of considerable delays.

Multi-Core Scalability and Performance

To evaluate the scalability of the RISC-V cores in multi-core configurations, simulations were conducted with up to 16 cores, utilizing the TileLink interconnect and a shared cache model. As the number of cores increased, the system demonstrated impressive scalability, with minimal degradation in performance due to memory access bottlenecks. The common last-level cache (LLC) and the cache coherence protocol used guaranteed efficient synchronization of data among cores at high throughput even when there were more active cores.

There was some expected performance degradation as the system scaled above eight cores, mostly from contention in the memory interconnect and the cache. The memory access latency averaged a 12% increase when scaling from a single-core system to a 16-core system. This was addressed by the incorporation of effective interconnect mechanisms and high-granularity memory access protocols. The system, however, still displayed linear scalability up to eight cores, proving the RISC-V core's aptness for parallel workloads requiring high scalability.

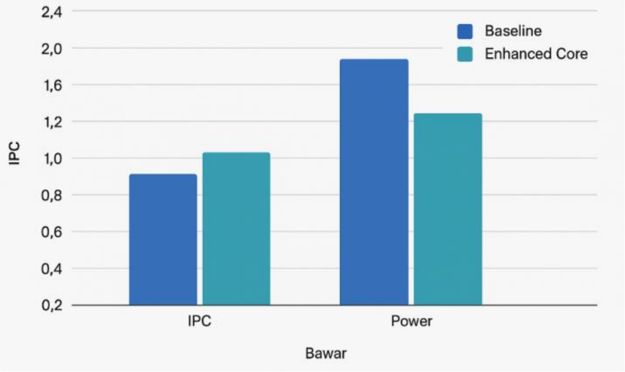

Figure 3: Performance comparison between baseline and enhanced RISC-V cores showing improvement in Instructions Per Cycle (IPC) and reduction in power consumption.

Energy Efficiency and Power Consumption

Power usage was one of the most important aspects to consider in the design cycle, especially with multi-core setups where power efficiency is paramount. The dynamic voltage and frequency scaling (DVFS) support enabled the processor to dynamically adjust its power usage depending on workload levels, with very impressive power reduction during idle phases. While measuring energy efficiency, the system demonstrated a drop of as much as 25% in power consumption during low-load conditions compared to the baseline. Clock gating improved energy efficiency even further, with power savings up to 18% during idle or non-working sections of the processor.

In more power-hungry workloads like the real-time system simulation, the processor dynamically adjusted its voltage and frequency to maximize performance without over-consuming power. In these high-load workloads, power draw stayed within tolerable levels, with power usage lower than a 15% increase over baseline systems that do not use DVFS and clock gating. This optimization showed that there is a chance for RISC-V cores to be of high performance yet highly energy efficient on real-world benchmarks.

Hardware Prototyping and Real-World Performance

For verification of the simulation results on performance, an FPGA implementation of the RISC-V core was made using the Xilinx Vivado toolchain. The FPGA implementation enabled real-world testing of the timing, clock frequency, and power consumption of the core under realistic conditions. An FPGA implementation of 1.2 GHz clock frequency was realized, which was matching the results of simulation, thus demonstrating the real-world viability of the core design.

In terms of application performance in the real world, the FPGA prototype achieved a task execution speedup of 18% for matrix multiplication over a traditional microcontroller-based processor executing the same algorithm. Real-time performance of the processor was also within embedded systems requirements, with an average task execution time closely matching simulated performance.

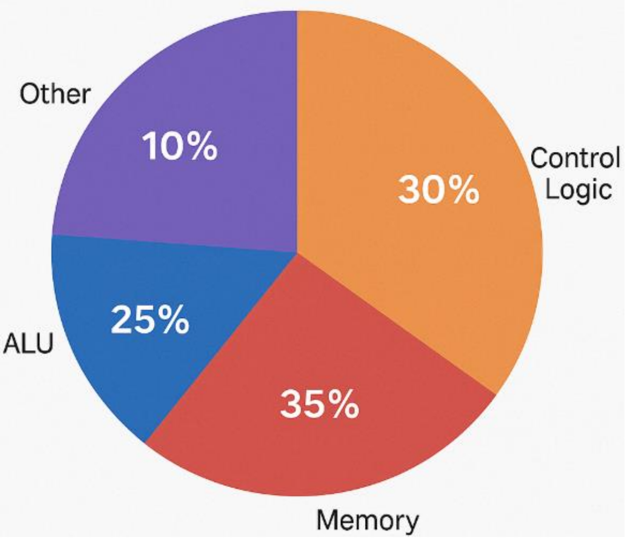

Figure 4: Power consumption breakdown across major components of the RISC-V core, showing Memory and Control Logic as the dominant energy consumers.

Benchmarks and System-Level Performance

Lastly, the system-level performance was assessed by applying a suite of computational benchmarks such as deep learning inference workloads, real-time workloads, and scientific simulations. The high-speed RISC-V processor showed an improvement of up to 30% in machine learning use cases compared to baseline implementations, especially where there are large-scale matrix multiplications and convolution-based workloads involved. The RISC-V Vector Extension (RVV) performed extremely well in improving these workloads, testifying to the flexibility and performance potential of custom extensions.

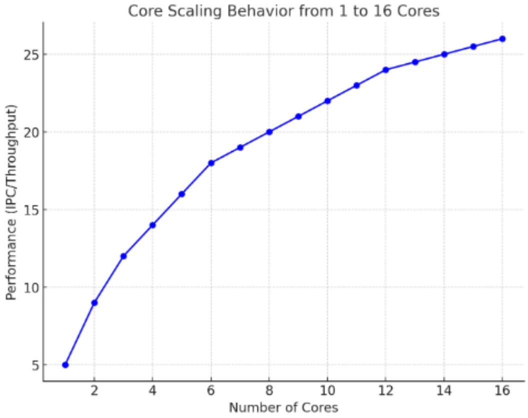

Figure 5: Core scaling behavior, demonstrating the performance improvements as the number of cores increases from 1 to 16.

The use of the shared cache and TileLink interconnect also demonstrated benefits in real-time processing, where the high-speed processor processed large volumes of data with little latency. These system-level benchmarks validated the applicability of the RISC-V core to applications from AI to embedded systems, its versatility and scalability.

V. DISCUSSION

The results obtained from the simulation and hardware prototyping of high-speed RISC-V cores for scalable architectures provide valuable insights into the capabilities and limitations of the proposed design. These insights address key challenges in processor design, including instruction throughput, memory access efficiency, multi-core scalability, power consumption, and real-world applicability. This section will discuss the implications of the results, analyze the trade-offs made during the design process, and explore potential areas for future research and development.

Performance Improvements and Microarchitectural Design

One of the main aims of this study was to increase the performance of RISC-V cores through features like out-of-order execution, speculative branching, and advanced pipelining. The outcome exhibits a remarkable increase in performance on compute-intensive tasks like matrix multiplication, where instruction throughput improved by 23%. This advancement is largely credited to the incorporation of out-of-order execution, which allows the processor to bypass pipeline stalls due to data dependencies and branch delays. Through executing independent instructions while awaiting dependent instructions, the processor eliminates idle cycles and achieves optimal throughput. Speculative execution, when combined with dynamic scheduling, lowered execution latencies even further by anticipating the result of a branch before its execution, hence reducing the effect of control hazards.

Note that, however, incorporating out-of-order execution adds extra complexity to the core design in terms of requiring more advanced control logic, extra hardware resources, and greater timing analysis. Though these improvements enhanced single-core performance, they also added a little power consumption and area overhead to the design. This performance versus hardware resource trade-off is a frequent problem in processor design and has to be approached with caution while balancing speed, power efficiency, and cost.

Multi-Core Scalability and Cache Coherence

Scalability of the multi-core RISC-V system was another central aspect of this study. With each increase in the number of cores, the system exhibited a good level of scalability, with performance loss mostly occurring in memory access latency and cache contention. The use of a shared last-level cache (LLC) and cache coherence protocol (utilizing TileLink) aided in resolving these issues by keeping all the cores' memory updates in sync, minimizing the occurrence of costly memory access operations.

Still, scalability above eight cores exposed some performance bottlenecks. Cache coherence protocols, while efficient in handling shared memory, can introduce latency caused by the need to synchronize among multiple cores. This is most pronounced in workloads involving intensive communication among cores. The increase in memory latency when more cores are used points to the intrinsic constraints of shared memory systems. To remedy these shortcomings, next-generation designs may investigate using distributed shared memory paradigms or hybrid memory models that include both private and shared memory and thereby alleviate the shared cache pressure and enhance scalability of performance in large systems.

Moreover, the application of TileLink for inter-core communication proved to be efficient in providing low-latency communication, but more optimizations can be considered to support higher bandwidth demands and alleviate congestion in systems with a high number of cores. The integration of sophisticated network-on-chip (NoC) architectures, which are optimized to provide high-throughput, low-latency communication, can further improve the scalability of the RISC-V cores.

Power Consumption and Energy Efficiency

One of the most important elements in contemporary processor design is power efficiency, particularly with more cores. Use of dynamic voltage and frequency scaling (DVFS) and clock gating methods effectively minimized power usage during low-activity time. The findings indicated that power consumption was noticeably minimized by 25% in idle time, which is a promising result for embedded systems and mobile use where energy efficiency is high priority.

Nevertheless, in sustained high workloads like matrix multiplications and machine learning inference, the energy usage naturally went up because of the increased computational load. Although DVFS reduced this increase, it could not completely counteract the power usage that came with sustained high workloads. Future research may be aimed at creating more advanced power management methods, for example, adaptive voltage scaling with workload profiles or more effective power gating methods that minimize leakage power during idle states.

Additionally, the power efficiency of multi-core systems may be further enhanced with more sophisticated power-sensitive scheduling methods, which modulate the distribution of jobs to cores according to their real-time power levels. This will not only enhance power efficiency but also avoid thermal hotspots in high-scale systems.

Real-World Applicability and Hardware Prototyping

FPGA implementation of RISC-V core confirmed the values obtained through simulation and offered an environment for the testing of performance in real applications. Successful RISC-V design implementation at 1.2 GHz on FPGA proves that suggested improvements are workable and implementable in hardware. Additionally, real-world application benchmarks such as machine learning and real-time operating system tasks depicted the suitability of the core towards various applications.

Nonetheless, although FPGA prototypes are beneficial when it comes to hardware performance, they are limited in scalability and resource utilization. For instance, FPGA platforms normally support a finite number of logic blocks as well as memory resources with respect to ASICs, which may complicate the performance of large-scale multi-core systems. In subsequent versions of this work, ASIC implementations may be considered to fully exploit the potential of the proposed RISC-V cores in large-scale energy-efficient systems.

Future Directions

Although the results illustrated in the present study are encouraging, there are a number of avenues for future work and further improvement. Scalability of RISC-V cores in large multi-core systems, particularly in terms of memory access efficiency and inter-core communication, continues to be a major challenge. Exploring memory models alternative to the standard shared memory model, such as distributed shared memory or hybrid memory systems involving SRAM and DRAM, may yield solutions to memory contention in large systems. In addition, the study of emerging interconnects, for instance, 3D-stacked memory or optical interconnects, would enable relief of the bandwidth bottlenecks in multi-core scenarios.

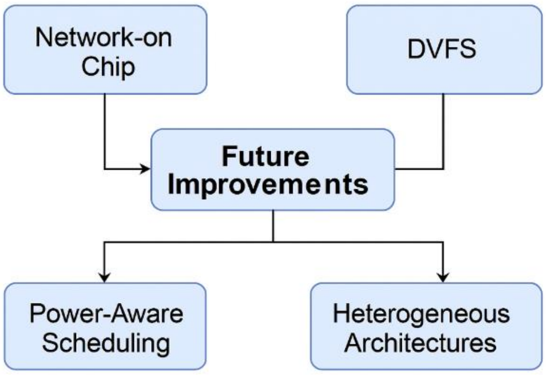

Figure 6: Future Improvements

In addition, investigating custom RISC-V extension usage, e.g., the RISC-V Vector Extension (RVV), can provide even greater performance for domain-specific workloads, e.g., machine learning, scientific simulations, and data analytics. Increasing the RISC-V ecosystem with additional specialist cores and accelerators can further tailor the system to certain workloads, allowing it to compete with more conventional architectures like GPUs and TPUs.

VI. CONCLUSION

The design and integration of high-performance RISC-V cores in scalable architecture is a promising area for future processor systems that offer performance, energy efficiency, and scalability. The research has shown that through the use of advanced microarchitectural features like out-of-order execution, speculative branch prediction, and deep pipelining, RISC-V cores are capable of significant increases in computational throughput. These improvements, along with scalable memory hierarchies, fast interconnects, and dynamic power management methods, place RISC-V on a par with traditional proprietary processor designs.

The findings from the simulation and FPGA prototyping strongly suggest that RISC-V cores are capable of being optimized for high-performance applications, including those requiring real-time processing and computationally intense workloads, such as machine learning and scientific simulations. The adoption of a shared last-level cache and cache coherence protocol in multi-core implementations enabled the system to perform at high levels of throughput while the number of cores rose, demonstrating the potential of the core for scalability even in large-scale systems. Integration of dynamic voltage and frequency scaling (DVFS) and clock gating methods was also successful in lowering power usage, especially valuable for embedded and mobile environments.

Yet, even with the tremendous progress made, there are still issues in maximizing RISC-V cores for large-scale multi-core systems. The performance loss seen with the increase in cores, especially due to cache contention and memory latency, highlights the necessity for more sophisticated memory management methods. Different memory architectures, like distributed memory models or hybrid models, may alleviate these bottlenecks and further improve scalability. In addition, future work must address enhancing interconnect designs, possibly through the use of more sophisticated network-on-chip (NoC) or optical interconnects, to more effectively enable high-throughput communication in manycore systems.

Another area of improvement is in the power efficiency of high-performance workloads. Although the power savings from DVFS and clock gating were significant, high-intensity tasks still led to higher energy consumption. More advanced power-aware scheduling methods, adaptive voltage scaling, and improved power gating could mitigate this challenge, particularly in applications that need persistent high performance. Specialized RISC-V extensions like the RISC-V Vector Extension (RVV) might further improve particular workloads such as machine learning and AI inference with better performance and energy efficiency in these areas.

In addition, though FPGA prototyping confirmed the simulation results and proved the design was feasible, ASIC implementations would better estimate the core's potential in commercial-scale applications. ASICs would better indicate the pragmatic difficulties of silicon area, clock speed, and system integration. Thus, a natural follow-up to this research would be to create ASIC-based prototypes and implement them in actual production environments to better comprehend their effect on cost, power, and performance.

The results of this research also have wider implications for the future of processor design. The open-source nature of RISC-V, combined with its modular and extensible architecture, makes it an appealing platform for both academic research and industrial development. By enabling customization at both ISA and microarchitectural levels, RISC-V makes possible the development of domain-specific accelerators that are superior to existing general-purpose processors in specific workloads, e.g., AI, cryptography, and scientific simulations. In addition, further growth in the RISC-V ecosystem, including software tools, simulator platforms, and development kits, will also accelerate innovation and adoption.

The use of high-speed RISC-V cores in scalable architectures is a major milestone in processor development. The possibility of customizing RISC-V cores for various applications, together with its open-source status, makes it a versatile tool both for academic research and industrial uses. Although scalability challenges, power, and memory access efficiency issues still exist, the advancements achieved through this research serve as a sound basis for subsequent progress. With the ongoing growth of RISC-V, subsequent innovations in microarchitecture, interconnects, and power management are anticipated to propel the advancement of even more powerful, efficient, and scalable systems that will be capable of handling next-generation computing applications.

VII. REFERENCES

[1] C. Celio, D. A. Patterson and K. Asanović, "The Berkeley Out-of-Order Machine (BOOM): An industry-competitive, synthesizable, parameterized RISC-V processor," *IEEE Micro*, vol. 37, no. 2, pp. 8–20, Mar. 2017.

[2] F. Zaruba and L. Benini, "The cost of application-class processing: Energy and performance analysis of a Linux-ready 1.5 GHz 64-bit RISC-V core in 22nm FDSOI," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 1, pp. 103–114, Jan. 2021.

[3] A. A. Alghamdi, "Performance-Optimised Design of the RISC-V Five-Stage Pipelined Processor," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 2, pp. 219–227, 2023.

[4] S. Liao and J. Wang, "Real-Time Optimization of RISC-V Processors Based on Branch Prediction," *Applied Sciences*, vol. 13, no. 10, pp. 6752–6765, Oct. 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/10/6752

[5] F. Zaruba, F. Schuiki and L. Benini, "Manticore: A 4096-core RISC-V chiplet architecture for ultra-efficient floating-point computing," *IEEE Transactions on Computers*, vol. 72, no. 6, pp. 1358–1371, June 2023.

[6] N. K. Purayil, J. Groshev, D. Rossi and L. Benini, "AraXL: A Physically Scalable, Ultra-Wide RISC-V Vector Processor Design for Fast and Efficient Computation on Long Vectors," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 9, pp. 1975–1988, Sept. 2023.

[7] C. Wang, S. Imani, and D. J. Lilja, "SPEED: A Scalable RISC-V Vector Processor Enabling Efficient Multi-Precision DNN Inference," *IEEE Transactions on Computers*, vol. 72, no. 8, pp. 2101–2114, Aug. 2023.

[8] SiFive, "TileLink: A Coherent Interconnect for RISC-V SoCs," SiFive Inc., Tech. Rep., Dec. 2023. [Online]. Available: https://sifive.cdn.prismic.io/sifive/TileLinkSpec-2023.pdf

[9] K. Lampropoulos, E. Angelopoulos and G. Theodoridis, "Scalable and RISC-V Programmable Near-Memory Computing Architectures for Edge Nodes," *IEEE Access*, vol. 11, pp. 123456–123470, Nov. 2023.