Journal of Climate and Community Development
A Double-Blind, Peer-Reviewed, HEC recognized Y-category Research Journal
E-ISSN: 3006-7855 P-ISSN: 3006-7847

A Comparative Performance Analysis of RISC (ARM/RISC-V) and CISC (x86) Architectures in Modern Computing Systems: Power Efficiency and Workload-Specific Benchmarking

Zeeshan Safdar[1] **Saleem Zubair**[2] **Ayesha Saddiqa**[3]

[1] PhD Advisor, Environment & Climate Change, EcoCare Consultancy Services, Pvt. Ltd, Lahore, Punjab, Pakistan.

[2] Masters Student, RUDN Institute OF Environmental Engineering, Specialization '' Ecology and Nature Management, People's Friendship University of Russia (RUDN University).

[3] Assistant Professor, Department Law School, National University of Sciences and Technology, Punjab, Pakistan.

Corresponding Author: Zeeshan Safdar, **Correspondence through:** zeeshansafdar752@gmail.com

Vol. *5,* ***Issue*** *2, 2026*

Article Information

Received:

2026-06-01

Revised:

2026-07-29

Accepted:

2026-08-14

ABSTRACT

This paper overcomes that gap by methodically consolidating and comparing published benchmark data that has been confirmed by hardware review sites, as well as by SPEC and the official hardware manufacturers' specifications, for various mobile-, desktop-, cloud-, embedded- and artificial-intelligence-related workload types. However, the break is very clear: ARM-based RISC processors, such as those used in Apple's M4 and Qualcomm's Snapdragon X-series, offer significantly better efficiency performance and performance per Watt for thermally restricted machines, where x86-based CISC processors from Intel and AMD remain competitive in sustained throughput, legacy software support and specialized high-performance computing tasks. When it comes to cloud infrastructure, AWS Graviton4 instances showing significantly better latency than X86 based instances in various workloads, however others showing either matched or improved performance per watt when compared to X86 based instances; indicating that while architecture may be superior in some workloads, it's not in others. While still a bit of a green land in the world of chips compared to ARM and x86, the open-source RISC-V ecosystem has seen an eight-fold performance boost over the last 5 years, and is being embraced by edge and embedded application makers. The paper argues that one should not be influenced by the preconceived notions about the superiority of RISC or CISC in selecting architecture since it must depend on application-specific requirements, power constraints and overall software-ecosystem cost.

Keywords: *Computer Architecture, RISC, CISC, ARM, x86, RISC-V, Processor Design, SPEC CPU2017, Geekbench, Performance-per-watt, Quantitative Benchmarking, Embedded Systems, Cloud Computing.*

Citation: APA

Safdar, Z., Zubair, S & Saddiqa, A (2026). A Comparative Performance Analysis of RISC (ARM/RISC-V) and CISC (x86) Architectures in Modern Computing Systems: Power Efficiency and Workload-Specific Benchmarking, Journal of Climate and Community Development, 5(2), 137-148

1. Introduction

A. Background

Over the last 50 years, computers have transformed virtually all aspects of human life. Over the last half a century, computing has influenced almost all parts of human life. From smart phones to laptops, autonomous systems to hyperscale Cloud Data Centers, processors are able to deliver billions of instructions per second of processing power. Every such processor is built around a particular Instruction Set Architecture (ISA), the way in which software instructs the behavior of the hardware.

There have been two dominant ISAs in the design of processors: Reduced Instruction Set Computer (RISC), and Complex Instruction Set Computers (CISC). While the idea first emerged as a competition in the early 1980s, the actual dividing line between the two — both practical and economic — is becoming more and more of a technical one than an ideological one.

With the development of cloud computing, artificial intelligence, edge computing, and mobile technology, the choice of the processor has become a strategic issue that takes into account power efficiency, size, security, and software ecosystem costs, as well as processing power. So, a hard examination on RISC and CISC, based on evidences is still relevant for system designers, computer engineers, and researchers.

B. Evolution of Processor Architectures

Most processor designs done in the 1960s and 1970s were CISCs, in part because of the cost of memory and because of the limitations of the compiler. Complex, multi-operation instructions saved the programs but increased the complexity in decoding the instructions [4]. The CISC lineage of when Intel first announced the 8086 (in 1978) was the most widely used in business, cemented for decades due to its continuity of backward compatibility [8][9].

Early work during the 1980s at the University of California, Berkeley and IBM Research found that compilers seldom use most of the complex instructions available to them; most of the instructions being executed use only a small subset of simple instructions that occur frequently [1, 2, 14]. This observation led to the RISC philosophy of fixed length instructions, load and store memory addressing, large register files and a deep pipeline [1, 2, 6]. These RISC principles later impacted on ARM, SPARC, PowerPC, MIPS, and - more recently - the open-source RISC-V ISA [2], [3], [6], [7].

The RISC program found significant success in ARM smartphone chips and, subsequently, in Apple Silicon devices and RISC-V chips on AWS Graviton servers [7,10] has turned the design around and has established it as one of the most widely adopted families of processors to date.

C. Problem Statement

Over the past 40 years, comparisons of RISC and CISC architectures have appeared in many different publications, but there are many instances of comparisons being made only at the general level; for instance, that "RISC intrinsically is more power-efficient", or that "CISC intrinsically is more performant," without reference to recent, published and source-verified data. Today's Intel and AMD processors break down complex instructions into simpler RISC-like-like operations internally [8] [9] and the ARM architectures have been increasingly adding complex vector, security and AI-acceleration extensions [7]. Comparing either design to its historical usage can thus yield findings that are today no longer relevant. In this research, the gap – based on the comparison between RISC and CISC is addressed by basing the comparison on verifiable and numerically-expressed benchmark evidence from mobile, desktop, cloud and embedded computing domains.

D. Research Objectives

This study aims to:

- Tell about the Principles of Architecture of RISC and CISC processor; Compare mechanisms for instruction-set design and execution, Compare the performance of a processor with unbiased and known benchmark test results of various workloads; Evaluate the amount of energy consumed – given a documented TDP and real-world power consumption; Explore use cases from mobile and desktop applications to cloud, embedded and AI-decelerated systems; Understand the current Industry trends such as RISC-V growth;
- Advise on criteria for selecting architecture for given application requirements.

E. Research Questions

This is the first question. This is the first question: What are the basic differences between RISC and CISC?

RQ2: In today's computing world, which architecture achieves improved measurable performance, and for which workloads?

RQ3: What are the differences in energy use, memory use, and hardware complexity based on the documented benchmarks and powers?

RQ4: What evidence in terms of benchmarks can be presented to answer the dominance of ARM in mobile computing?

RQ5: What is it about x86 that continues to be the foundation of enterprise and desktop computing?

RQ6: What is the likely impact of the measured performance trend of RISC-V on the future architectures of processors?

F. Scope of the Research

This research takes a close look at ARM, x86, AMD64, Apple Silicon and RISC-V architectures when they exist in real systems. Analysis points are instruction set design, single core and multi core performance, power efficiency, hardware complexity, cloud server efficiency, embedded suitability, and capability of acceleration of AI. There is no independent hardware testing done as part of this study, but instead, the results presented here are secondary benchmark data found in the published literature, which is normally used for comparative computer-architecture studies [1, 4].

G. Significance of the Study

This research is useful to computer-science students because it demonstrates that architectural theory is supported by current, numerically verifiable evidence [1, 4]; to software architects and system engineers as it will help them in platform-selection processes; and to embedded-systems developers because of the ability to quantify the power advantages that make RISC the dominant architecture in that space [7]; and to enterprise decision-makers as it will help serve the purpose of deciding whether to keep using x86 in mission-critical workloads and what the current have to offer [8, 9]. It also provides an evidence-based basis for the current discussions around RISC-V adoption, heterogeneous computing and AI-accelerated processor design [14, 17].

II. Literature Review

A. Introduction

This RISC–CISC comparison issue has been the major issue for the computer architecture discipline for over 40 years now [1]. Although the two philosophies were initially framed as opposing approaches, advances in semiconductor fabrication, compiler optimization, cache hierarchy, and parallel execution have progressively reduced many of the original distinctions between them. ARM-based RISC processors today power the overwhelming majority of smartphones and embedded devices and are expanding into personal computing [7], [10], while x86-based CISC processors continue to dominate desktop computing, enterprise servers, and legacy enterprise software [8], [9]. This review synthesizes foundational and recent literature and identifies the research gap motivating the present study.

B. Historical Development of CISC Architecture

CISC architecture started in the 1960s and 1970s when memory was too costly, stimulating a reduction of program size by instructions that could execute many operations [4, 5]. This method showed simpler decode logic with our added instruction count for a particular program [4]. The most commercially successful philosophy for the 8086 was released first in 1978, and Intel's continued backward compatibility over the years has been attributed as a key component of the dominance of x86 in enterprise and desktop computing [8, 9]. The modern CISC implementations mitigate the decoding costs by using pipelining, superscalar execution, branch prediction and speculative execution [1].

C. Emergence of RISC Architecture

It was found in the early ‘80s that compilers did not use most of the CISC instructions, and that by far most of the programs currently being run were using a very limited number of simple instructions [1, 14]. The Berkeley RISC project by Patterson, which will be further described later on, and the 801 project of IBM [14] showed that fewer instructions, not more, can lead to more through-put [2]. The design features that emerged as a result of this (fixed-length instructions, load-store architecture, large register windows and deep pipelines) later

influenced ARM, SPARC, PowerPC, MIPS, and RISC-V [2], [3], [6] and [7].

D. Comparison of Instruction Set Design

CISC architectures offer variable length instructions and multiple addressing modes with large instruction sets, viable to express complex operation in small instructions [4] [5]. In RISC architectures, however, dealing with much smaller and uniform instruction sets, in most clock cycles, they execute within a few cycles [1, 2, 6]. By interpreting x86 instructions into simplified micro-operations for execution [8] and [9] inside the processor, many modern applications of CISC have been called "insider's view RISC" by researchers, one of the most important and significant strides in the modern design of processors.

E. Performance Evaluation in Previous Studies

There are many measures conventionally used to evaluate processor performance, such as Instructions Per Cycle (IPC), Cycles Per Instruction (CPI), clock frequency, throughput, latency, energy consumption, and cache efficiency [1]. In mobile and battery-limited applications, independent tests show that ARM processors deliver better performance per Watt than x86 processors, whereas in applications such as virtualization, database and large-scale scientific applications, Intel Xeon and AMD EPYC processors have an edge for performance [8] [9]. The M-series processors by Apple have also confirmed that ARM based RISC processor designs can achieve desktop class performance yet consume significantly less power [10] than a x86 design, thereby defying the earlier notion of fitting RISC in embedded usage scenarios.

F. Energy Efficiency

A vast improvement in energy efficiency is becoming a crucial evaluation factor, as the smartphone, wearables, drones, and IoT market continues to expand. ARM Limited [7] reports a significant reduction in power consumption relative to traditional desktop processors; since these reports are self-made, independent reviews performed as described in Section IV confirm significant power reductions (down to single digits of watts) for ARM based SoCs that enable full processing power without the need to use a fan. It is the reason why ARM's ubiquity is such that just about every iPhone 1, 2, and 3.0 supposedly running iOS 5.0 or later out there will be powered by a Qualcomm, Samsung, MediaTek, or Apple smartphone chipset.

G. Software Compatibility

One of the best features of CISC is backward compatibility. The legacy codebases have been around for decades and are still being used on Intel and AMD chips [8,9] and enterprises are not driven towards changing their environments by cost considerations. This compatibility, though, complicates decoding/validation at each new generation of processor [1]. The situation for ARM is vastly improved with translation layers like Apple's Rosetta 2 [10] but processing native x86 software is still CISC's more robust feature.

H. Modern Architectural Convergence

The RISC–CISC difference is no longer very evident at the micro architectural level, according to recent literature. ARM processors have added some extensions to SIMD, NEON extensions, cryptographic extensions, and machine-learning extensions that were thought to carry with them the complexity of CISC chips have adopted [7]; ARM also shrunk the cost of its simple, low-power instruction sets while x86 has earned its reputation for complexity by breaking tasks down into micro-operations something much closer to RISC architecture [8, 9]. As such, several researchers say that the fact of the matter is that modern processor performance is more dependent on the micro architectural implementation than its philosophy with regard to the ISA.

I. Emerging Architecture: RISC-V

Developed by China's open-source and royalty-free ISA, RISC-V, can be used to create custom-designed processors for the universities and startups, or semiconductor companies, and without the burden of licensing fees [3, 14]. AI, embedded systems, robotics, edge computing and automotive electronics are all areas poised for adoption growth [6,16]. Support for virtualization on RISC-V cores, e.g. designs like CVA6, also shows how ready the architecture is for server class and multi-tenant applications [18] whilst at the same time it hasn't yet reached commercial maturity compared to ARM and x86 [14].

J. RISC and CISC in Artificial Intelligence

AI workloads are performed using simple matrix multiplication and parallel vector processing, which is very different from traditional computer use.

Consequently, none of the big processor vendors—whether RISC or CISC—actively avoid any kind of dedicated AI predictor in their new coprocessor design, and Apple's Neural Engine [10], Intel's AI Boost NPU [8] and AMD's XDNA engines [9] represent good examples of that.

K. Alternative RISC Implementations

The other RISC-based architectures continue to influence the design space in general: Beyond ARM and RISC-V. RISC benefits have not been limited to the mobile and embedded domain as shown in IBM's POWER architecture that has been documented by the Open POWER Foundation [11].

L. Research Gap

From this literature review, the following four persistent gaps were identified: (i) many studies focus on theoretical architectural differences at the exclusion of the recent convergence in the world of microarchitectures, (ii) benchmarking studies mostly focus on one computing domain either mobile or desktop, while multiple domains can be evaluated at the same time, (iii) microarchitectures, such as Apple Silicon and RISC-V, are still under-benchmarked in the academic literature and (iv), most importantly, comparative claims are often expressed rather than supported by current, numerate and properly cited benchmarks. Addressing gap (iv) directly, this research features a new section, Quantitative Bench Result (Section IV), consisting solely of verifiable, dateable and cited secondary sources.

M. Summary of Foundation Literature

Table I: *Summary of Foundational Literature*

Author(s) / Source	Core Focus	Key Finding
Hennessy & Patterson [1]	Quantitative Architecture	Microarchitectural techniques (pipelining, speculation) allow CISC to remain competitive despite decode overhead.
Patterson & Hennessy [2]	RISC Design	Simplified instruction sets improve pipeline throughput and register efficiency.
Stallings [4]	Computer Organization	CISC provides powerful instructions and strong backward compatibility.
Tanenbaum & Austin [5]	Structured Organization	System performance depends closely on compiler-microarchitecture interaction.
ARM Ltd. [7]	Energy-Efficient Design	Simplified decode structures enable superior performance-per-watt.
Intel Corp. [8]	Enterprise Systems	x86 preserves ecosystem compatibility via macro-to-micro-op translation.
Apple Inc. [10]	High-Performance RISC	Custom ARM silicon competes directly with premium x86 desktop chips.
RISC-V International [13]	Open ISA	Open architecture lowers market-entry barriers for custom silicon.

III. Research Methodology

A. Introduction

The methodology used in this work is comparative, i.e. comparing RISC and CISC architectures within various current fields of computing is based on secondary data analysis. Considering the fact that the architecture of processors is well documented by standardized processors' benchmarking, the research is based on the published or proven data rather than on original hardware experimentations, following the general practice in computer science comparative researches [1].

B. Research Design

The study employs mixed design with a qualitative and quantitative approach using comparative design style. Sections I through II provide descriptive

analysis thereby defining architecture, whereas Sections IV—Measure able differences in performance, power and efficiency—provides a quantitative secondary data analysis, comparing documented, standard published numbers to establish measurable differences.

C. Technique: Quantitative Secondary Data Comparative Analysis

In this research the technique applied is a core technique which is a quantitative secondary data comparative analysis. In doing this, instead of computer manufacturers doing original hardware testing, published benchmark data from standardized testing organizations, and independent hardware-review websites is typed up systematically, cross-checked extensively, and assembled into a common numeric comparison metric. This approach has proven useful in computer architecture research; when credible, when-possible and when-verified published data exists, it is typically neither practical nor desirable to repeat the merely large-scale testing of systems that can be simulated in order to recharacterize their performance.

D. Data Collection Method

The sources were selected using explicit inclusion and exclusion criteria as follows, (i) standardized benchmark authorities (SPEC CPU2017 [28]); (ii) independent hardware-testing publications that comply with a well-established methodology and are cross-referenced across at least 2 independent outlets per figure (Notebookcheck [21], [22]; Beebom [19]; XDA Developers [20]; Tom's Guide [23]; Phoronix [24] and [27]; Dedicatted [25]); (iii) official technical documentation released by the manufacturer (ARM [7], Intel [8], AMD [9] and [26], Apple [10], RISC-V International [13]); (iv) peer-reviewed and academic publications (IEEE Xplore, ACM Digital Library, arXiv [15]–[18]). Sources were removed if they failed to report a test methodology, a baseline without which figures could not be compared, or if they could not be matched with the results from one or more independent sources. Figures for the current generation silicon (2024-2026) were selected for the data collection involving the 1990s.

E. Data Analysis Technique

Benchmark numbers were gathered and presented in tables to compare with each other inside the individual domains, based on mobile/laptop, power consumption, cloud/server, and embedded/RISC-V. Where sources reported percentage differences these were used as reported; percentage differences were calculated directly from the raw scores presented in sources where only raw scores were available and these are all presented here for uniformity of presentation across the tables.

F. Structural Overview of Evaluation Parameters

Table II: *System Comparison Matrix*

Evaluation Parameter	RISC Architecture	CISC Architecture
Instruction Set Structure	Small, uniform, simple	Large and complex
Instruction Length	Fixed (32/64-bit)	Variable (1–15 bytes)
Execution Model	Single-cycle focus	Multi-cycle, micro-op translated
Memory Access	Load/Store only	Direct memory-to-register
Hardware Complexity	Low decode complexity	High decode complexity
Typical Power Envelope	4–40 W (mobile/laptop class)	45–125+ W (laptop/desktop class)
Software Compatibility	Improving via translation layers	Extensive native legacy support

IV. Quantitative Benchmark Results

A. Overview

Here are the numbers – verified and cited – the core of the numbers that you need to remain focused on and not move away from, from mobile/laptop computing, power consumption, embedded/RISC-V performance, and cloud/server infrastructure. These are meant to replace the subjective scores found in research, with measured, source-related data, thus solving the research gap identified in Section II-L.

B. Mobile and Laptop Performance (Geekbench 6)

Independent test results confirm that across all ARM-based SoC single-core throughput is considerably high when compared to its competitors, and that the gap in multi-core performance is going in the opposite direction, depending on the number of cores. There's no denying that Apple has taken a solid shot at out dubbing Google's Snapdragon X Elite with its own Apple M4 processor, taking a bucks-up 3,781 single-core scores versus 2,441 with the Snapdragon X Elite, and multi-core scores weren't far behind at 14,858 versus 14,050 respectively [19]. The second-generation Snapdragon X2 Elite Extreme then clocked in at 4,074 single core points and 23,449 multi core points [23, see yearly progression] – a jump of 270% in the latter mark from the previous year.

Table III: *Geekbench 6 Benchmark Comparison (Verified Published Scores)*

Processor (ISA)	Single-Core Score	Multi-Core Score	Source
Apple M4 (ARM/RISC)	3,781	14,858	[19]
Snapdragon X Elite Gen 1 (ARM/RISC)	2,441	14,050	[19]
Snapdragon X2 Elite Extreme (ARM/RISC)	4,074	23,449	[23]
Intel Core Ultra 7 258V (CISC)	~2,700*	~9,300*	[20]

*Approximate figures derived from percentage differences reported relative to Apple M4 in the cited source [20], where Apple M4 single-core performance was measured at 39% higher than the Intel Core Ultra 7 258V.

C. Power Consumption Analysis

Known power numbers further support the efficiency that is commonly attributed to RISC designs with ARM processors. The Apple M4, when hosted in an iPad, requires only 4-5W to function normally, increasing only to 14-20W for high loads as the effect of active cooling takes place in Mac mini and MacBook Pro models respectively [21]. In contrast, the Intel Core Ultra 5 245HX has a documented TDP of 55 W [22] and higher end x86 desktop and x86 workstation processors that have a rating of 100 W or more under full load will be the norm.

Table IV: *Documented Power Consumption (TDP / Measured Draw)*

Processor (ISA)	Power Envelope	Context	Source
Apple M4 (ARM/RISC)	4–5 W (fanless); 30–40 W (cooled)	iPad / Mac mini	[21]
Intel Core Ultra 5 245HX (CISC)	55 W (TDP)	Gaming notebook	[22]
AMD Ryzen AI 9 HX 370 (CISC)	Higher sustained draw than M4 under load	Laptop	[17 - nanoreview cited in text]

D. Cloud and Server Performance

For cloud infrastructure, outcome is more workload related than for mobile computing. According to a third-party test, the AWS Graviton4 (ARM, Neoverse-V2 cores) instance had around 38% lower latency for the read-write sysbench OLTP test compared to a similarly sized Intel Xeon 8488C instance, and around 20% lower latency compared to an AMD EPYC 9R14 instance on identically configured 64-vCPU instances [25]. Neither of these independent bare-metal testings has performed well, as Graviton4 was core-for-core and, depending on workload, equal to or superior to fourth-generation AMD EPYC.

According to ARM's own published efficiency figures, however, when measured by performance per watt, the AMD EPYC 9965 processors offer up to 3.3x higher performance in individual workloads than the ARM based NVIDIA Grace superchip [26] signaling that the efficiency advantage of ARM is not consistent across the top of the server spectrum and is heavily dependent on workload type, cost-

per-core, and vendor optimization.

Table V: *Cloud/Server Benchmark Comparison*

Comparison	Result	Source
Graviton4 (ARM) vs. Intel Xeon 8488C	~38% lower latency (Graviton4)	[25]
Graviton4 (ARM) vs. AMD EPYC 9R14	~20% lower latency (Graviton4)	[25]
Graviton4 (ARM) vs. AMD EPYC (4th Gen)	Matching or exceeding, workload-dependent	[24]
AMD EPYC 9965 (CISC) vs. NVIDIA Grace (ARM)	Up to 3.3x higher performance-per-watt (EPYC)	[26]

E. Embedded and RISC-V Performance Trajectory

In a short time, the performance of RISC-V has improved significantly. Independent benchmarking showed an around 8x performance boost over the five-year period between the SiFive HiFive Unmatched board and the SpacemiT K3 RVA23 system-on-chip [27]. Commercial RISC-V IP, including the SiFive Performance P650, is expected to surpass 11 SPECInt2006/per-GHz forwards and make a case for RISC-V as a viable chip choice for edge AI, embedded and commercial IoT applications, although below ARM and x86 in terms of overall software ecosystem maturity [13], [14].

F. AI Acceleration Comparison

Table VI: *AI Co-Processor Implementations by Host Architecture*

Host Processor (ISA)	AI Co-Processor	Primary Function
Apple M-Series (ARM/RISC)	Neural Engine	On-device neural inference
Intel Core Ultra (CISC)	AI Boost NPU	Matrix operation acceleration
AMD Ryzen AI (CISC)	XDNA Engine	Local client inference
Qualcomm Snapdragon (ARM/RISC)	Hexagon NPU	Edge multi-modal inference

These figures confirm that AI-workload competitiveness today depends primarily on dedicated accelerator silicon rather than the host processor's underlying ISA, a finding consistent with the architectural convergence discussed in Section II-H.

G. Consolidated Findings

Table VII: *Consolidated Quantitative Comparison Summary*

Evaluation Domain	Quantitative Evidence	Favoured Architecture
Single-core mobile/laptop performance	Apple M4 leads Snapdragon X Elite by ~55% (Geekbench 6) [19]	RISC (ARM)
Multi-core mobile/laptop performance	Gap narrows to <6% or reverses with higher core counts [19]	Workload-dependent
Fanless power envelope	4–5 W (M4) vs. 55 W TDP (Intel Ultra 5 245HX) [21], [22]	RISC (ARM)
Cloud latency (OLTP)	Graviton4 20–38% lower latency vs. Xeon/EPYC [25]	RISC (ARM)
Server performance-per-watt (select workloads)	EPYC up to 3.3x vs. ARM Grace [26]	CISC (x86)
5-year embedded/RISC-V performance growth	~8x improvement (SiFive to SpacemiT K3) [27]	RISC (RISC-V, trend)

Evaluation Domain	Quantitative Evidence	Favoured Architecture
Legacy software ecosystem	Decades of native x86 compatibility [8], [9]	CISC (x86)

V. Discussion

A. Interpretation of Findings

In summarizing the quantitative evidence provided in Section IV the conclusion drawn is that the RISC–CISC comparisons can no longer be decided on sup/craf arguments. A clear and measurable benefit has been seen for ARM-based RISC processors in terms of single-core efficiency and thermally constrained power envelopes as seen in the independently sourced mobile and laptop benchmarks [19] – [23]. This, however, is tempered significantly when it comes to multi-core processing for x86-based contenders have added up core count and sustained cooling capacity to their processors; and when it comes to the highest end of server workloads, where AMD's published proof powering the NVIDIA Grace processor actually indicates that EPYC outperforms the ARM-based option on performance per watt [26] in certain configurations.

The pattern does suggest, as the research gap in Section II-L suggested, that the absolute superiority of either RISC or CISC is domain/ workload dependent and that the current, cross-checked configuration of benchmark programs has little bearing on this

B. RISC and CISC in Contemporary Computing

The figures of latency and power above illustrate that ARM's shift to the cloud with AWS Graviton and to the desktop/laptop with Apple Silicon is warranted. The fact that RISC-V is open and royalty-free, not only continues to make it easier to bring custom silicon to market for AI, robotics, automotive, and IoT, but given that performance has actually been measured at eight times higher over five years, might this be accelerating and not slowing down? [3]

The continued relevance of CISC is also well demonstrated. CISC still offers some real technical benefits today, not just inertia from entrenched software bases, as AMD's lead in certain high-density server workloads demonstrates in terms of performance-per-watt [26].

C. Implications for Software Architects

Do not choose a processor because of a preference for a particular architecture. ARM's documented power efficiency makes it an ideal choice for embedded controllers, smartphones and wearables. The x86 platform continues to enjoy the mature ecosystem of enterprise resource planning systems, database servers, and virtualization platforms, where-and-in-what configurations-its superior performance-per-watt [8] is shown to translate into superior performance. More architects should expect that more will rely on this approach of a system using a mix of different types of computing devices, such as CPUs, GPUs and/or dedicated NPUs, that each contribute to the overall performance without regard to the host ISA [10].

D. Limitations of the Study

This is all secondary published benchmark data; no facilities were used for independent testing of the hardware, and the numbers were the result of test sets under specific hardware configurations, compiler configurations, and for specific workloads by the original testing organizations and may not be representative of every practical application [25]. The sources mentioned in Section IV-B [20] also admit to the fact that benchmark scores like Geekbench 6 reward different workloads (such as memory bound workloads) differently on different platforms. The architecture of processors has also changed rapidly over the years, and numbers as of 2026 can be outdated in a matter of as quickly as a few months, especially for the rapidly growing RISC-V architecture [27]. Lastly, this research omits comparisons to new paradigms like quantum computing, neuromorphic processors and chiplet-based heterogeneous architectures which could significantly alter future architectural comparisons.

VI. Conclusion

This study aimed to take the discussion on the RISC–CISC controversy out of the general theoretical realm and to make it more concrete by relying on empirical, current, and properly referenced quantitative benchmark numbers in mobile, desktop, cloud, and embedded computing areas. The evidence strongly shows that there is no clear, measurable victory in either domain; ARM-based RISC processors that rely on fewer

instructions per Watt to achieve higher CPU performance excel in power-critical applications such as mobile, while x86-based CISC processors offer better legacy support, as well as multi-threaded, sustained throughput in some architectures and configurations with server-based applications. Although it is young, RISC-V is rapidly gaining momentum with a known and rapidly improving performance curve that appears likely to increase its use in embedded, edge and ultimately general-purpose computing.

What this means is that, at the micro architectural level, RISC and CISC are converging, and the dedicated silicon for AI in the future combined with the growing number of workloads for which it is optimized will increasingly give a processor its edge, over balancing adherence to RISC or CISC. Therefore, depending on application requirements, measured power requirements, and total cost of software-ecosystem, the choice of processor should be based on existing benchmark for this and should not rely on inherited assumptions.

Future Work

RISC-V independent experimental benchmarking on hardware (ARM and x86 under same, controlled, test conditions).

The performance of AI (NPU/TPU)-accelerator over different processors generations.

Expanded performance-per-watt comparison of ARM-based cloud servers to x86 cloud servers running a variety of enterprise workloads.

Modern Processor Architecture Security against Side Channel Attacks (Spectre & Meltdown).

Comprehensive analysis of Heterogeneous architectures done by chiplets from AMD, Intel and Apple.

Acknowledgment

The author gratefully acknowledges the guidance and supervision provided by the Department of Computer Science, Superior University Lahore (Gold Campus), in the preparation of this research.

Conflict of Interest

The author declares no conflict of interest. This study did not involve human subjects, animal testing, or original experimental hardware testing; all benchmark data were obtained from previously published, publicly available secondary sources.

References

J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, 6th ed. Cambridge, MA, USA: Morgan Kaufmann, 2017.

D. A. Patterson and J. L. Hennessy, Computer Organization and Design: RISC-V Edition. San Francisco, CA, USA: Morgan Kaufmann, 2017.

D. A. Patterson and A. Waterman, The RISC-V Reader: An Open Architecture Atlas. Berkeley, CA, USA: Strawberry Canyon LLC, 2017.

W. Stallings, Computer Organization and Architecture, 11th ed. New York, NY, USA: Pearson, 2022.

A. S. Tanenbaum and T. Austin, Structured Computer Organization, 6th ed. Upper Saddle River, NJ, USA: Pearson, 2013.

D. M. Harris and S. L. Harris, Digital Design and Computer Architecture: RISC-V Edition. Codeocean, USA: Morgan Kaufmann, 2021.

Arm Architecture Reference Manual, Arm Ltd., Cambridge, UK, 2024.

Intel 64 and IA-32 Architectures Software Developer's Manual, Intel Corporation, Santa Clara, CA, USA, 2024.

AMD64 Architecture Programmer's Manual, Advanced Micro Devices, Austin, TX, USA, 2024.

Apple Silicon Overview, Apple Inc., Cupertino, CA, USA, 2024.

Power Architecture Documentation, OpenPOWER Foundation, Austin, TX, USA, 2024.

Snapdragon Technical Documentation, Qualcomm Technologies, San Diego, CA, USA, 2024.

RISC-V Specifications, RISC-V International, Geneva, Switzerland, 2024.

K. Asanović and D. Patterson, "Instruction sets should be free: The case for RISC-V," EECS Dept., Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2014-146, Aug. 2014.

C. Celio, P. Dabbelt, D. Patterson, and K. Asanović, "The renewed case for the reduced instruction set computer: Avoiding ISA bloat with macro-op fusion for RISC-V," arXiv preprint arXiv:1607.02318, 2016.

T. Lu, "A survey on RISC-V security: Hardware and architecture," arXiv preprint arXiv:2104.14811, 2021.

M. Perotti et al., "A new Ara for vector computing: An open source highly efficient RISC-V vector processor design," arXiv preprint arXiv:2206.04378, 2022.

B. Sá et al., "CVA6 RISC-V virtualization: Architecture, microarchitecture, and design space exploration," arXiv preprint arXiv:2309.01142, 2023.

"Apple M4 vs Snapdragon X Elite: Benchmark Comparison," Beebom, Nov. 2024. [Online].

"We tested it: Here's how Apple's M4 compares to Intel Lunar Lake and Snapdragon X Elite," XDA Developers, Nov. 2024. [Online].

"Apple M4 (10 cores) vs Intel Core Ultra 9 288V vs AMD Ryzen AI 9 HX 370," Notebookcheck, 2024. [Online].

"Apple M4 (10 cores) vs Intel Core Ultra 5 245HX vs Intel Core Ultra 7 255HX," Notebookcheck, 2024. [Online].

"I benchmarked the Snapdragon X2 Elite Extreme — here's how it compares to Apple M4, Intel Core Ultra 9 and more," Tom's Guide, Sept. 2025. [Online].

M. Larabel, "AWS Graviton4 96-Core Performance vs. AMD EPYC & Intel Xeon CPU Benchmarks Review," Phoronix, 2024. [Online].

"ARM vs x86: How AWS Graviton Can Save up to 40% on Cloud Costs," Dedicatted, 2026. [Online].

"Choose AMD EPYC CPUs for Hybrid Cloud Efficiency," Advanced Micro Devices, technical infographic. [Online].

M. Larabel, "RISC-V CPU Performance Up 8x in Five Years: SiFive HiFive Unmatched to SpacemiT K3," Phoronix, Jun. 2026. [Online].

SPEC CPU 2017 Benchmarks, Standard Performance Evaluation Corporation. [Online]. Available: https://www.spec.org/cpu2017/.