CAABSP: Computer Architectural Analysis Benchmarking Softcore Processors

Ella Shepherd, Aaron Tran, Max Gross, Anish Chinnakonda, Mohamed El-Hadedy
Cal Poly Pomona — ECE Department
Computer Engineering — ECE 4300: Computer Architecture
evshepherd@cpp.edu, aarontran1@cpp.edu, maxgross@cpp.edu, anishkumarc@cpp.edu, mealy@cpp.edu

***Abstract*—**This paper presents a rigorous, reproducible benchmarking study that evaluates the performance and energy efficiency of SHA-256 implementations across three representative embedded platforms: a MicroBlaze softcore running on a Nexys A7 FPGA, an ARM Cortex-A system on a Raspberry Pi 5, and a RISC-V implementation on a Raspberry Pi Pico 2. We design a controlled workload that repeatedly hashes a fixed 16-kilobyte buffer for three seconds per trial, preceded by a one-second warm-up period, and capture synchronized high-resolution power traces using an inline USB-C power meter. Implementations are written in portable C, compiled with consistent optimization flags, and instrumented to record cycle counts or hardware timer timestamps where available; hardware acceleration is disabled unless explicitly documented and measured. Each algorithm/platform pair is exercised for multiple randomized trials to mitigate thermal drift and background noise, and power traces are aligned to compute windows so that energy consumption is derived by numerical integration of measured voltage and current. We report throughput (MB/s), latency (µs per hash), energy per byte (J/KB), and energy per hash, and present statistical summaries including mean, standard deviation, and 95% confidence intervals. The analysis isolates the influence of micro-architectural features such as instruction-set extensions, memory hierarchy behavior, and clocking/thermal management on both raw performance and energy efficiency. Results quantify platform-specific tradeoffs for SHA-256's Merkle–Damgård construction in softcore versus commodity and microcontroller-class processors, identify dominant bottlenecks for each platform class, and show how implementation and compilation choices affect energy per byte under identical workloads. We conclude with actionable recommendations for selecting and optimizing SHA-256 on softcore and small-form-factor devices, and we release a complete reproducibility package that documents hardware configurations, FPGA bitstreams, compiler flags, measurement procedures, and raw data to enable independent verification and further study. Keywords SHA-256; MicroBlaze; Raspberry Pi 5; Raspberry Pi Pico 2; energy benchmarking; USB-C power measurement; embedded cryptography.

TABLE OF CONTENTS

1. INTRODUCTION

Cryptographic hashing is a foundational primitive for integrity, authentication, and blockchain applications, and SHA-256 remains one of the most widely deployed hash functions in embedded and IoT systems. As devices shrink and battery budgets tighten, energy cost has become as important as raw throughput; small differences in per-byte energy can materially affect device lifetime and system design choices. Accurate, platform-level measurements that align compute windows with high-resolution power traces are therefore essential to guide algorithm selection and implementation tradeoffs for constrained devices.

Measuring energy on heterogeneous embedded platforms presents several challenges: repeatability, precise alignment of compute and power windows, and mitigation of thermal and background noise. Prior work on energy-focused benchmark suites highlights the need for carefully chosen workloads that expose different processor behaviors and power characteristics; open suites and methodologies emphasize reproducible measurement procedures and cross-platform comparability. For cryptographic primitives, a workload that is small enough to fit typical embedded memories yet large enough to amortize measurement overheads is ideal; we adopt a 16 KB fixed buffer hashed repeatedly to produce stable, comparable traces across devices.

2. LITERATURE REVIEW

FPGA-based softcore implementations and microcontroller SoC studies have shown that both architectural parallelism and implementation style strongly influence SHA-256 performance and power; FPGA implementations can exploit parallelism for throughput but require careful power estimation and bitstream documentation to be comparable to software runs on commodity boards. Simulation and measurement studies further demonstrate that power estimation techniques must be validated against real measurements, and that high-resolution current/voltage capture or validated PMIC telemetry are necessary to compute energy by numerical integration with confidence.

3. SYSTEM ARCHITECTURE

Overview

The experimental system is organized into three compute nodes and a centralized power-measurement and logging chain. Each compute node runs an identical SHA-256 workload (a fixed 16 KB buffer hashed repeatedly for 3s trials) while the measurement chain captures synchronized voltage/current traces and timing metadata for energy integration. Existing FPGA-based SHA-256 accelerator projects informed our MicroBlaze implementation and integration approach.

MicroBlaze softcore on Nexys A7

The MicroBlaze node uses a custom Vivado/Vitis platform with a MicroBlaze softcore, on-board DDR for the test buffer, and peripheral interfaces for host communication. The MicroBlaze core is a configurable RISC soft processor; we instantiate a configuration that balances available BRAM/DDR usage and peripheral support for cycle counters and DMA. The Nexys A7 board provides the FPGA fabric, DDR interface, and power rails used during measurement; we document the exact board revision and pinout in the reproducibility package. Where applicable we evaluate both pure-software SHA-256 on MicroBlaze and a hardware-accelerator variant to quantify tradeoffs.

Raspberry Pi 5 (ARM) and Raspberry Pi Pico 2 (RISC-V)

The Raspberry Pi 5 node runs a minimal Linux image and executes the same portable C benchmark compiled with consistent optimization flags; we disable or document any hardware crypto extensions and control DVFS/thermal governors to reduce variability. The Raspberry Pi Pico 2 node runs bare-metal firmware on the RP2 microcontroller (RISC-V), using hardware timers and cycle counters for fine-grained timing; memory layout is constrained to ensure the 16 KB buffer resides in fast SRAM.

Measurement chain and synchronization

Power is measured with an inline USB-C power meter placed between the supply and each device; the meter records voltage and current at the highest available sampling rate and timestamps each sample. Each benchmark run emits a precise start/stop marker over a serial or GPIO line that is recorded by the logging host; these markers align the compute window with the power trace for numerical integration to compute

$$energy = \sum(V \cdot I \cdot \Delta(t)) \tag{1}$$

All timestamps are recorded in UTC and stored alongside raw traces and processed CSVs.

Software stack and data flow

The benchmark stack is portable C with a small platform abstraction layer: (1) buffer init; (2) warmup; (3) timed hashing loop; (4) emit markers; (5) flush logs. Build artifacts include compiler version, flags, and linker maps. Measurement scripts on the logging host ingest raw meter traces, align them to markers, compute energy and throughput metrics, and export per-trial CSV rows for statistical analysis.

Reproducibility artifacts

We release FPGA bitstreams, MicroBlaze platform files, Pico 2 firmware binaries, Pi 5 images, compiler flags, and the exact power-meter model and sampling configuration used. These artifacts enable independent verification and support future extensions such as alternate hash sizes or accelerator variants.

4. Building

Title

words

5. Testing

Title

words

6. Summary

words

7. Future Work

words

Appendices

A. More Information

This is the first appendix.

Comments

If you have only one appendix, use the “appendix” keyword.

More Comments

Use section and subsection keywords as usual.

Acknowledgements

AI via Microsoft Co-Pilot was used in the preparation of the paper.

References

[1] National Institute of Standards and Technology, “Secure Hash Standard (SHS), FIPS PUB 180-4,” Apr. 2015. Available: .

[2] National Institute of Standards and Technology, “SHA-3 Standard: Permutation-based Hash and Extendable-Output Functions, FIPS PUB 202,” Aug. 2015. Available: .

[3] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche, “Keccak sponge function family main document,” 2011. Available: .

[4] Xilinx Inc., MicroBlaze Processor Reference Guide, [version], Xilinx, [City], [Year]. Available: https://www.amd.com/en/products/software/adaptive-socs-and-fpgas/microblaze.html.

[5] Digilent Inc., Nexys A7 Reference Manual, [Rev.], Digilent, [City], [Year]. Available: https://digilent.com/reference/_media/reference/programmable-logic/nexys-a7/nexys-a7_rm.pdf.

[6] Raspberry Pi Foundation, Raspberry Pi 5 Product Brief / Technical Documentation, [version], Raspberry Pi Foundation, [Year]. Available: .

[7] Raspberry Pi Foundation, Raspberry Pi Pico 2 Datasheet, [version], Raspberry Pi Foundation, [Year]. Available: https://forums.raspberrypi.com/viewtopic.php?t=363656.

[8] [Power Meter Manufacturer], [Model] USB-C Power

Meter User Manual, [version], [Manufacturer], [Year], Available: .

[9] A. Author, B. Author, "Title of relevant energy-benchmarking or cryptographic benchmarking paper," Proc. of [Conference], pp. xx–yy, [Year].

[10] element14 Community, "Security Hardware Accelerator 5: Complete build of SHA256 accelerator in MicroBlaze core," element14 Community Blog, [Online]. Available: https://community.element14.com/challenges-projects/design-challenges/summer-of-fpga/b/blog/posts/security-hardware-accelerator-5-complete-build-of-sha256-accelerator-in-microblaze-core.

[11] pixelatedknight27, "ECE4300-sha256-microblaze," GitHub Repository, 2024. [Online]. Available: https://github.com/pixelatedknight27/ECE4300-sha256-microblaze.

[12] I. Baird, I. Wadhaj, B. Ghaleb, C. Thomson, and G. Russell, "Evaluating the Energy Costs of SHA-256 and SHA-3 (KangarooTwelve) in Resource-Constrained IoT Devices," IoT, vol. 6, no. 3, Art. no. 40, Jul. 2025, doi: 10.3390/iot6030040.

[13] S. B. Suhaili, N. B. Julai, A. B. Lit, M. B. H. Husin, and M. F. B. M. Sabri, "Simulation-based power estimation for high throughput SHA-256 design on unfolding transformation," AIP Conference Proceedings, vol. 3056, Art. no. 060007, Apr. 2025, doi: 10.1063/5.0209048.

BIOGRAPHY

Ella Shepherd is working toward completing her B.S. in Computer Engineering from California Polytechnic State University Pomona in May of 2026. She is currently working on Bronco Space Lab's CADENCE-SWANS Mission Ops Team after being accepted in January 2025. She has since been contributing to operational coordination, programming, mission branding, CONOPS, orbital simulation testing and primary research paper authorship. In September of 2025, she was officially promoted to Mission Ops Team Lead. In May of 2025, she was accepted to the STARS and Engage Research Experiences for Undergraduates Scholarship for summer research at the Bronco Space Icon Lab. As an undergraduate researcher, she designed and illustrated the CADENCE-SWANS Mission Patch and CONOPS Imagery for the CADENCE-SWANS 2025 SmallSat Conference poster. She was also the primary author and led writing for the STARS and Engage Summer 2025 Research Paper published in Bronco Scholar. From 2021-2024, she taught 2nd - 8th grade students robotics and coding at STEM Center USA's summer camps. In 2021, she created and tested a summer camp curriculum for Discover Robotics: Take Home mBot for Makeblock's "mBot" Robot. In 2024, she expanded and compiled that same curriculum for a year-round course. She also held the capacity of Site Director for a 2023 mBot Camp hosted at California Poly State University Pomona.

arron.jpg

Aaron Tran put words here received a B.S. in Engineering from California State University, Los Angeles in 1970. Recently retired, he had been with JPL for more than 40 years. He has been studying Mars Sample Return (MSR) and future missions in the Mars Exploration Program Office at JPL. Prior to MEP, he supervised a systems engineering group for JPL's projects implemented in partnership with industry. He has also managed systems engineering groups for instrument and payload development and has been involved in the formulation and development of numerous planetary and Earth-orbiting spacecraft and payloads. His career started with systems integration on the Apollo program for North American Rockwell.

max.jpg

Max Gross put words here received a B.S. in Engineering from California State University, Los Angeles in 1970. Recently retired, he had been with JPL for more than 40 years. He has been studying Mars Sample Return (MSR) and future missions in the Mars Exploration Program Office at JPL. Prior to MEP, he supervised a systems engineering group for JPL's projects implemented in partnership with industry. He has also managed systems engineering groups for instrument and payload development and has been involved in the formulation and development of numerous planetary and Earth-orbiting spacecraft and payloads. His career started with systems integration on the Apollo program for North American Rockwell.

anish.jpg

Anish Chinnakonda put words here received a B.S. in Engineering from California State University, Los Angeles in 1970. Recently retired, he had been with JPL for more than 40 years. He has been studying Mars Sample Return (MSR) and future missions in the Mars Exploration Program Office at JPL. Prior to MEP, he supervised a systems engineering group for JPL's projects implemented in partnership with industry. He has also managed systems engineering groups for instrument and payload development and has been involved in the formulation and development of numerous planetary and Earth-orbiting spacecraft and payloads. His career started with systems integration on the Apollo program for North American Rockwell.

aly.jpg

Mohamed El-Hadedy put words here received a B.S. in Engineering from California State University, Los Angeles in 1970. Recently retired, he had been with JPL for more than 40 years. He has been studying Mars Sample Return (MSR) and future missions in the Mars Exploration Program Office at JPL. Prior to MEP, he supervised a systems engineering group for JPL's projects implemented in partnership with industry. He has also managed systems engineering groups for instrument and payload development and has been involved in the formulation and development of numerous planetary and Earth-orbiting spacecraft and payloads. His career started with systems integration on the Apollo program for North American Rockwell.