Preface of Literature Review

Preface to the Literature Review

This literature review serves as the foundational blueprint for the entire project, providing the critical context that informed its design, implementation, and evaluation. The surveyed works revealed a significant gap in the existing research: while individual studies offered excellent in-depth analyses of either the Bit-Manipulation ('B') or Scalar Cryptography ('K') extensions, a unified, comparative evaluation on a consistent hardware platform was absent. This project was conceived to directly address that gap, and the literature provided an indispensable guide for the tools, coding techniques, and performance benchmarks necessary to achieve this goal.

1. Guiding the Research Methodology and Toolchain

To ensure a scientifically rigorous comparison, a controlled, full-system emulation environment was required. The body of research confirmed this approach, validating the project's selection of QEMU and the RISC-V GCC toolchain as industry-standard tools for this type of analysis. The work of Gewehr et al. on low-power embedded systems [7], for example, exemplifies the kind of rigorous evaluation this project sought to emulate. This methodology effectively isolates the performance impact of the ISA extensions from physical hardware variables. The literature provides the practical blueprint for leveraging the RISC-V GCC toolchain, specifically guiding the use of the -march compiler flag to target the rv32imac_zbb_zbc and rv32imac_zksh_zknh ISA strings, which was essential for the compiler to generate the specialized instructions at the core of this investigation.

2. From Theoretical Gains to Practical Implementation

The prior art provided a direct roadmap for translating theoretical performance gains into the tangible C code implemented in this project. The strategy involved creating two versions of each cryptographic function—a baseline and an accelerated version—to enable direct, quantitative comparison.

Optimizing AES with the 'B' Extension: Following the path laid out by the work of Babu, Sivaraman, Sarma, and Warrier [2] and the FPGA implementation by Markov and Romanov [5], this project's accelerated AES implementation replaces slow, memory-intensive software routines with single, efficient hardware instructions. In the ShiftRows function, expensive byte-wise memory swaps are replaced by the rori (rotate right immediate) instruction. For the MixColumns step, the complex and computationally intensive xtime() helper function for Galois Field multiplication is substituted with a direct call to the clmul (carry-less multiply) instruction, a core feature of the Zbc sub-extension validated by these papers.

Accelerating SHA-256 with the 'K' Extension: The detailed analysis by Gewehr and Moraes [3, 7] is the definitive guide for accelerating SHA-256. Their research demonstrated the power of 'fused instructions,' a principle this project applied by using dedicated instructions like sha256sig0 and sha256sum0. This allowed for the replacement of complex, multi-instruction C macros with single, high-performance hardware operations, directly targeting the algorithm's computational hotspots in the message schedule and compression rounds.

3. Establishing a Framework for Validation and Context

Crucially, the literature provided not only a method but also the quantitative benchmarks needed to validate and contextualize this project's findings. The success of the project is measured by its ability to replicate and explain its results in the context of the prior art.

For AES, the observed ~50% throughput improvement (~1.5x speedup) aligns perfectly with the moderate yet significant gains demonstrated for the general-purpose 'B' extension in the works of Babu et al. [2] and Markov & Romanov [5]. This result also underscores a fundamental design trade-off by standing in useful comparison to the transformative 42.57x speedup reported by Gewehr et al. for the domain-specific Zkne (AES) extension [3], confirming that while the 'B' extension offers broad utility, specialized instructions yield orders-of-magnitude greater performance.

For SHA-256, the measured 10-20% throughput improvement (~1.1x-1.2x speedup) successfully verifies the consistent, albeit more modest, benefits of the Zksh instructions reported by Gewehr and Moraes [3, 7]. This result confirms the efficiency of the fused instructions while also reflecting the performance constraints imposed by the serially dependent nature of the hashing algorithm.

In summary, this literature review was not a passive survey but an active and comprehensive blueprint. It provided the 'why' (the research gap), the 'how' (the tools and coding practices), and the 'what to expect' (the performance targets), enabling the rigorous and comparative evaluation that forms the core contribution of this work.

References

[1] https://roalogic.github.io/RV12/DATASHEET.html

[2] P. S. Babu, S. Sivaraman, D. N. Sarma, and T. S. Warrier, "Evaluation of Bit Manipulation Instructions in Optimization of Size and Speed in RISC-V," in Proc. 34th Int. Conf. on VLSI Design (VLSID), 2021, pp. 54-59, doi: 10.1109/VLSID51830.2021.00014.

[3] C. G. de A. Gewehr and F. G. Moraes, "Improving the Efficiency of Cryptography Algorithms on Resource-Constrained Embedded Systems via RISC-V Instruction Set Extensions," in Proc. 36th Symp. on Integrated Circuits and Systems Design (SBCCI), 2023, doi: 10.1109/S-BCCI60457.2023.10261964.

[4] K. Kim, D. Harris, and K. Macsai-Goren, "Design and Synthesis of RISC-V Bit Manipulation Extensions," in Proc. 57th Asilomar Conf. on Signals, Systems, and Computers, 2023, doi: 10.1109/IEEECONF59524.2023.10477073.

[5] D. Markov and A. Romanov, "Implementation of the RISC-V Architecture with the Extended Zbb Instruction Set," in Proc. Int. Ural Conf. on Electrical Power Engineering (UralCon), 2022, doi: 10.1109/UralCon54942.2022.9906776.

[6] M. A. Elmohr et al., "Hardware Implementation of A SHA-3 Application-Specific Instruction Set Processor," in Proc. Int. Conf. on Microelectronics (ICM), 2016, pp. 109-112, doi: 10.1109/ICM.2016.7847846.

[7] C. Gewehr et al., "Hardware Acceleration of Authenticated Encryption with Associated Data via RISC-V Instruction Set Extensions in Low Power Embedded Systems," Feb. 2024, doi: https://doi.org/10.1109/lascas60203.2024.10506132.

[8] M. Namazi Rizi, N. Zidaric, L. Batina, and N. Mentens, "Optimised AES with RISC-V Vector Extensions," in Proc. 27th Int. Symp. on Design & Diagnostics of Electronic Circuits & Systems (DDECS), 2024, doi: 10.1109/DDECS60919.2024.10508919.