RISC-V Accelerated Processors on FPGA: Architecture, Design and Implementation

Author[1][1]*, Author[2][2], Author[3][3], Author[4][4], Author[5][5]*

[1]*Education Institute, Country*

[2]*Education Institute, Country*

[3]*Education Institute, Country*

[4]*Education Institute, Country*

[5]*Education Institute, Country*

example1@mail — *example2@mail*

Keywords: RISC-V, FPGA, Hardware Description Language (HDL), VHDL, 32-bit processor, ISA, APU, GPU-Accelerated Computing, RISC-V APU, 5 stage pipelining, System on Chip (SoC), Open Source, Digital Logic Design, Computer Architecture, RV32I, Vortex, e-GPU

Abstract: This survey paper reviews recent research and development efforts on Field Programmable Gate Array (FPGA) based implementations of Reduced Instruction Set Computing version Five (RISC-V) Accelerated Processing Units (APUs) that integrate CPU and GPU capabilities on a unified platform. Focused primarily on RV32I processors extended with SIMD/SIMT architectures and GPU-specialized accelerators, this work examines open-source Hardware Description Language (HDL) implementations targeting resource-efficient computing. Key topics include architectural trade-offs, programmability, and hardware-software co-design challenges. Architectures such as Vortex and e-GPU serve as case studies illustrating state-of-the-art RISC-V APU design strategies.

1 INTRODUCTION

The RISC-V Instruction Set Architecture (ISA) is a modern, open, and modular processor design standard that offers extensibility and customization beyond proprietary ISAs. Among its foundational subsets, the 32-bit RV32I base ISA provides a simple yet solid instruction framework, making it a popular choice for lightweight embedded processors and educational FPGA projects. The open-source nature and simplicity of RV32I enable rapid development and extension for diverse applications.

Recently, there has been growing research interest in designing Accelerated Processing Units (APUs) that merge traditional CPU cores with GPU-like parallel processors on a single chip. This integration addresses the need for heterogeneous computing platforms that leverage the sequential flexibility of CPUs alongside the data-parallel throughput of GPUs. RISC-V's modular ISA and open ecosystem facilitate the seamless extension of CPUs with SIMD and SIMT architectures, enabling such heterogeneous designs.

This survey aims to provide an overview of FPGA-based RISC-V APUs with a focus on architectural innovations, ISA extensions, and open-source hardware developments, using projects like Vortex and e-GPU as illustrative examples. The goal is to guide future research towards scalable, energy-efficient, and programmable heterogeneous computing solutions using RISC-V.

2 LITERATURE SURVEY

The landscape of RISC-V processor implementation on FPGAs is well-documented. Foundational work includes the design of 32-bit RISC-V CPUs targeting the RV32I base ISA with both single-cycle and pipelined versions to analyze throughput and resource trade-offs [1]. Others have focused on creating flexible, parameterized softcores that allow designers to balance performance and area for different applications [5]. Implementations often extend beyond the base ISA, with researchers demonstrating performance gains by adding application-specific custom instructions to an FPGA-based System on Chip (SoC) while maintaining ISA compliance [6]. Simpler implementations have also been verified in hardware, providing practical data on resource usage and performance [7].

A significant area of research is the extension of RISC-V for accelerated computing, particularly for graphics and general-purpose GPU (GPGPU) tasks. One approach integrates a fixed-pipeline graphics rendering extension into the ISA, optimized for low-power rendering on embedded AI chips [8]. More general-purpose architectures, such as Vortex, propose a minimal ISA extension for SIMT execution to support OpenCL and OpenGL, implementing a scalable multi-core soft GPU on FPGAs [2, 16]. Similarly, the e-GPU platform presents an open-source, configurable RISC-V GPU for ultra-low-power TinyAI edge devices, showing significant speedups on specialized workloads [3]. The FGPU project also demonstrates a scalable SIMT soft processor for FPGAs, focusing on integer processing and achieving notable speedups over traditional embedded processors [17]. To manage the complexity of these heterogeneous systems, scalable interconnects like the Open Bus Interface (OBI) are specified for integrating cores and accelerators [18]. The correctness of such complex parallel hardware is also being addressed through formal methods, with work on formalizing GPU instruction behavior in Coq to enable verifiable models for both software and hardware [15].

3 SURVEY OVERVIEW

3.1 Relevance and Timeliness

This survey addresses a critical juncture in the semiconductor industry where RISC-V architecture is experiencing unprecedented growth and adoption. As of 2024, over two billion System-on-Chips (SoCs) are currently using RISC-V cores, with projections indicating this number will surge to 20 billion by 2031. The global RISC-V market was valued at USD 1.76 billion in 2024 and is expected to reach USD 25.73 billion by 2034, growing at a CAGR of 30.7%. The timing of this survey is particularly significant as major semiconductor players including Bosch, Infineon, Nordic, NXP, and Qualcomm have jointly invested in advancing RISC-V adoption globally. The ratification of the RVA23 Profile standard in October 2024 marks a watershed moment for software portability across RISC-V implementations, accelerating toolchain and operating system adoption across diverse processor configurations.

3.2 Impact on VLSI Design and APU Development

The emergence of RISC-V APUs represents a fundamental shift in VLSI design methodologies, challenging the decades-long dominance of proprietary instruction set architectures. By eliminating costly licensing fees and vendor lock-in constraints, RISC-V enables semiconductor companies to innovate more freely while reducing development costs and accelerating time-to-market. FPGA-based prototyping platforms, as examined in this survey, provide ideal testbeds for exploring RISC-V extensions and heterogeneous computing architectures. The flexibility of FPGAs enables rapid iteration on microarchitectural options, ISA extensions, and system integration strategies without the prohibitive costs associated with ASIC fabrication.

APU architectures that integrate CPU and GPU capabilities on unified platforms address the fundamental challenge of balancing sequential processing efficiency with massive data-parallel throughput. The RV32I base ISA serves as a robust foundation for these heterogeneous designs, providing a simple yet extensible instruction framework that facilitates the addition of SIMD and SIMT execution capabilities. This modularity enables designers to customize processor configurations for specific application domains while maintaining ISA compliance and software compatibility.

3.3 Addressing Architectural Challenges in Heterogeneous Processing

The semiconductor industry faces mounting pressures to deliver computing solutions that efficiently handle diverse workload characteristics spanning graphics rendering, parallel computation, and embedded control. Traditional CPU-only architectures struggle with data-parallel throughput requirements, while standalone GPU designs lack the flexibility needed for sequential control logic and irregular computation patterns. RISC-V APUs directly address these challenges by integrating CPU and GPU capabilities on unified platforms with shared memory hierarchies and coherent interconnects.

This survey provides critical insights into architectural trade-offs, programmability models, and hardware-software co-design strategies essential for practical APU implementation. By examining state-of-the-art projects like Vortex and e-GPU, the work establishes design patterns and best practices for SIMT/SIMD extensions, warp scheduling, memory subsystem optimization, and divergent branch handling. The Vortex architecture demonstrates minimal ISA extensions for SIMT execution supporting OpenCL and OpenGL, implementing a scalable multi-core soft GPU on FPGAs. Similarly, the e-GPU platform presents an open-source, configurable RISC-V GPU showing significant speedups on specialized workloads while maintaining ultra-low-power operation.

The methodological approaches documented in this survey span hardware description language selection (VHDL, Verilog, Chisel), pipeline microarchitecture design, GPU execution models, and system-level integration using standardized interconnects like the Open Bus Interface. These contributions provide practical guidance for researchers and practitioners developing RISC-V-based heterogeneous processors, addressing critical design decisions regarding resource allocation, thread scheduling, memory bandwidth optimization, and execution divergence handling.

3.4 Implications for the RISC-V Processor Ecosystem

The technical content of this survey directly supports RISC-V's strategic positioning across computing domains requiring customizable, extensible processor architectures. The RV32I base ISA examined here serves as the foundation for countless embedded processors and educational FPGA projects, while the GPU acceleration techniques presented enable RISC-V to compete in computationally demanding workloads traditionally dominated by proprietary architectures. Major technology companies are already leveraging RISC-V at scale—NVIDIA alone is estimated to ship between one and two billion RISC-V cores in 2024, utilizing the architecture in its CUDA microcontroller cores. Qualcomm has shipped over 650 million RISC-V cores in its Snapdragon processors.

The adoption patterns demonstrate RISC-V's viability across processor complexity spectrums, from simple embedded microcontrollers to sophisticated multi-core APU configurations. Infineon, the world's leading automotive microcontroller company, launched its first RISC-V-based AURIX automotive microcontroller in March 2025, signaling mainstream acceptance in safety-critical applications. The European Space Agency has partnered with semiconductor companies to develop RISC-V processors for space industry microcontrollers, citing the architecture's configurability, security, and cost-effectiveness. These adoption patterns underscore the practical relevance of APU research presented in this survey.

The RV32I subset's simplicity makes it particularly well-suited for FPGA implementation, enabling resource-efficient designs that can be extended with GPU-like acceleration capabilities. Research documented in this survey demonstrates performance gains achievable through application-specific custom instructions while maintaining ISA compliance. The open-source nature of RISC-V facilitates collaborative innovation, allowing designers to contribute to architectural evolution while sharing advancements with the broader community. As RISC-V International expands to over 4,120 members across 52 countries with more than 80 technical working groups, surveys documenting existing APU implementations and architectural patterns become increasingly valuable for coordinating community efforts and establishing reference designs.

3.5 Future Directions in RISC-V APU Research

Looking forward, this survey establishes a foundation for next-generation research in scalable, energy-efficient, and programmable heterogeneous computing using RISC-V. Critical open questions include optimizing architectural complexity in multi-core APU configurations, improving synchronization mechanisms between CPU and GPU execution contexts, enhancing communication bandwidth between heterogeneous cores through advanced interconnect designs, and ensuring energy-efficient operation for resource-constrained FPGA implementations. The formal methods approaches mentioned for verifying complex parallel hardware behavior represent promising directions for ensuring correctness as APU designs grow in sophistication, with work on formalizing GPU instruction behavior in Coq enabling verifiable models for both software and hardware.

FPGA-based prototyping will continue to play a crucial role in exploring APU design spaces before committing to silicon implementation. The configurable nature of FPGAs allows researchers to experiment with pipeline depths, cache hierarchies, thread scheduling policies, and memory subsystem organizations without the financial risk associated with ASIC fabrication. Hardware description languages such as VHDL, Verilog, and Chisel enable modular, parameterizable hardware construction facilitating iterative design and verification cycles across different FPGA platforms including Xilinx and Altera families.

The convergence of RV32I-based CPU cores with

GPU-like SIMT/SIMD acceleration presents opportunities for specialized processor designs optimized for specific computational patterns. Thread-level parallelism, multi-port caching, and parallel memory banks collectively balance area, bandwidth, and performance while maintaining programmable flexibility. Software frameworks leverage execution models through lightweight kernel dispatch abstractions, allowing scalable control over thread blocks and warps in a manner reminiscent of CUDA-style execution. As the RISC-V community grows to encompass over 16,000 engineers worldwide through education initiatives, mentorship programs, and technical training, surveys like this one become essential educational resources for onboarding new contributors and establishing shared understanding of APU architectural principles.

In conclusion, this survey arrives at a pivotal moment when RISC-V is transitioning from academic curiosity to industrial reality across processor design domains. By documenting current APU implementations spanning RV32I CPU cores and GPU acceleration units, analyzing design trade-offs in heterogeneous architectures, and identifying future research directions in FPGA-based prototyping, this work contributes to the collective knowledge base enabling RISC-V to fulfill its promise as an open, modular, and transformative computing platform.

4 METHODOLOGY

4.1 Hardware Overview

The hardware methodology focuses on FPGA-based prototyping platforms for implementing heterogeneous RISC-V Accelerated Processing Units (APUs). The FPGA environment enables flexible exploration of microarchitectural options, ISA extensions, and system integration. Key FPGA platforms used include Xilinx and Altera FPGA families, providing sufficient resources to prototype multi-core CPUs alongside GPU accelerators with shared memory and interconnect subsystems. Hardware description languages such as VHDL, Verilog, and Chisel have been employed for modular, parameterizable hardware construction facilitating iterative design and verification cycles.

4.2 CPU Design

The CPU cores in these APUs primarily implement the 32-bit RISC-V RV32I base ISA with pipeline microarchitectures ranging from simple five-stage classic pipelines to more advanced, configurable implementations supporting extensions like multiplication and division. Design efforts emphasize modular register files, control units, and arithmetic logic units to maintain compliance with RISC-V specifications while prioritizing ease of extension. The CPU cores support standard CSR registers, memory management, and interrupt handling as per the specification and are tailored for FPGA optimization to balance throughput and resource usage. Configurability for applications ranges from embedded control tasks to general computational work.

4.3 GPU Design

Modern GPUs extend their base instruction set to efficiently support SIMT and SIMD execution, allowing thousands of lightweight threads to process data in parallel within organized groups called warps. Architectures like Vortex, combine multiple shader-like compute cores, warp schedulers, and deeply pipelined memory systems that overlap execution and data fetching to sustain throughput. Thread execution follows a lockstep model, where each warp shares an instruction stream but operates on distinct data values. Divergent branches are resolved using execution masks, enabling selective thread activity without heavy control overhead. Thread-level parallelism, multi-port caching, and parallel memory banks collectively help balance area, bandwidth, and performance while maintaining programmable flexibility. Designs with configurable compute units or fixed-function pipelines further trade off power efficiency and workload adaptability depending on their deployment context.

At the ISA and microarchitecture level, minimal hardware changes can introduce scalar and vector instruction classes alongside dedicated register files, program counters, and mask handling logic, enabling flexible thread management within a familiar RISC-like environment. Parallel execution units share functional resources between warps, while latency is hidden through rapid context switching. Software frameworks leverage this model through lightweight kernel dispatch abstractions, allowing scalable control over blocks and warps in a manner reminiscent of OpenCL or CUDA-style execution—illustrating the underlying principles behind modern GPU programming and edge-parallel computation.

4.4 APU System Design

The RISC-V-based APU architecture integrates CPU and GPU elements on a unified platform through a

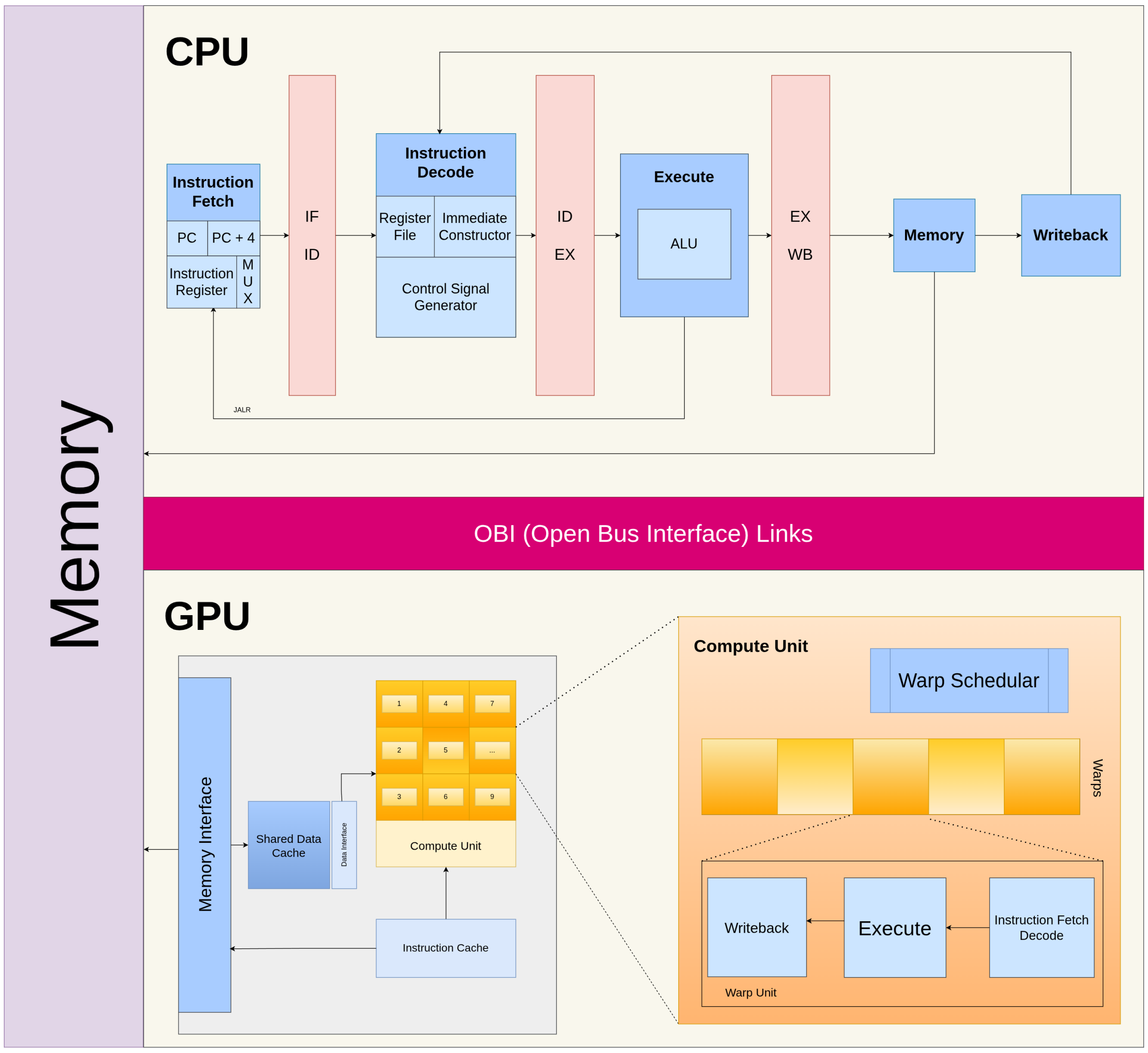

Figure 1: APU Pipeline Design

shared interconnect fabric, exemplified by the Open-Bus Interface (OBI), which enables seamless communication between heterogeneous cores. As show in e-GPU, The system supports coherent memory sharing while balancing latency and bandwidth to sustain efficient task execution across compute units. The e-GPU operates as an independent accelerator with dedicated OBI master and slave ports for memory transactions and configuration, along with an interrupt line for real-time synchronization with the host. Power and clock gating mechanisms tied to the host power manager allow dynamic energy scaling, ensuring optimal performance and power efficiency for embedded and TinyAI-class workloads.

The host subsystem features a lightweight RISC-V processor optimized for control-intensive tasks and acts as the orchestration layer for the GPU, managing kernel dispatch, memory allocation, and execution flow through a high-bandwidth interconnect and low-latency SRAM banks. A minimalist runtime environment, built on Newlib and a compact OpenCL-like API, enables host–accelerator coordination without dependence on full operating systems or multithreading. This hardware-software co-design approach allows applications to be compiled into single executables with static resource definition, simplifying deployment and runtime management. FPGA-based validation of the system supports experimentation with ISA-level modifications, pipeline tuning, and memory hierarchies, providing a flexible framework for evaluating performance-power tradeoffs in processing environments.

5 CONCLUSION

The RISC-V ecosystem has demonstrated significant progress towards realizing flexible and extensible processor designs suitable for a wide range of embedded and parallel computing applications. The foundational RV32I implementations continue to provide a robust baseline for efficient core processors, as shown by various FPGA-based single-cycle and pipelined architectures that balance throughput and implementation complexity effectively.

Recent advancements in heterogeneous RISC-V platforms, exemplified by projects such as Vortex and e-GPU, illustrate the practical integration of GPU-like acceleration into traditional CPUs to form APUs capable of exploiting data-parallel workloads across domains like AI, graphics, and edge computing. These designs highlight the feasibility of minimal ISA extensions, scalable multi-core configurations, and novel cache and memory hierarchies that maintain programmability via open-source toolchains and standardized interfaces.

The combined body of work emphasizes the importance of modularity, configurability, and hardware/software co-design in addressing evolving performance and application-specific requirements. Challenges remain in optimizing architectural complexity, synchronization, and communication across heterogeneous cores while ensuring energy-efficient operation suitable for resource-constrained environments. Overall, these efforts present an encouraging future trajectory for RISC-V APUs, promoting more versatile, open, and high-performance computing architectures tailored to contemporary embedded and parallel computation needs.

ACKNOWLEDGEMENTS

The authors acknowledge the use of AI-based tools, including Perplexity AI and Gemini, for assistance in generating some portions of the text.

REFERENCES

[1] Rao, M., Niranjan, P., & MJ, D. K. (2024, October). Design and Implementation of 32-bit RISC-V Processor Using Verilog. In *2024 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)* (pp. 181-186). IEEE.

[2] Tine, B., Elsabbagh, F., Yalamarthy, K., & Kim, H. (2021). Vortex: Extending the RISC-V ISA for GPGPU and 3D-Graphics Research. In *54th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)* (pp. 1-13). ACM.

[3] Machetti, S., Schiavone, P. D., Orlandic, L., Huang, D., Kasap, D., Ansaloni, G., & Atienza, D. (2025). e-GPU: An Open-Source and Configurable RISC-V Graphic Processing Unit for TinyAI Applications. *IEEE Transactions on Embedded Systems*.

[4] Goh, J. K., & Uttraphan, C. (2024). An Enhanced UTHM RISC-V Processor Core Architecture Implemented on FPGA. *Evolution in Electrical and Electronic Engineering, 5*(2), 32-41.

[5] Rodrigues, J. F. M. (2019). *Configurable RISC-V softcore processor for FPGA implementation* (Master's thesis, Instituto Superior Técnico).

[6] Li, Z., Hu, W., & Chen, S. (2019, August). Design and implementation of CNN custom processor based on RISC-V architecture. In *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)* (pp. 1945-1950). IEEE.

[7] Begum, S., & Kumar, T. (2023). FPGA-based Implementation of 32-bit RISC-V Processor. *International Journal of Engineering Research and Technology*.

[8] Zhou, Y., Jin, X., & Xiang, T. (2020). RISC-V Graphics Rendering Instruction Set Extensions for Embedded Chips. In *Proceedings of the 2020 ACM Symposium on Embedded Computing*. ACM.

[9] Matthews, E. et al. (2021). Soft-Processor Parameterized RISC-V. *IEEE Transactions on Computers*.

[10] Pol, L. (2021). Lightweight Open-Source RISC-V Processor for IoT. In *Mobility, Sensing, and Networking Conference*.

[11] Miyazaki, H., Kanamori, T., Islam, M. A., & Kise, K. (2020). RVCoreP: An optimized RISC-V soft processor of five-stage pipelining. *IEICE TRANSACTIONS on Information and Systems, 103*(12), 2494-2503.

[12] Barriga, A. et al. (2021). Synthesis and Verification of RISC-V Cores. In *International Symposium on Embedded Systems*.

[13] Jo, S. et al. (2023). Single Cycle RISC-V Core Implementation. *International Journal of Engineering Research and Technology*.

[14] Jinyu, J. et al. (2022). Implementing RISC-V ISA on FPGA. *International Journal of Advanced Engineering*.

[15] Bhatia, N., D'Souza, M., & Chakrabarti, S. K. (2019, February). Formalizing GPU Instruction Set Architecture in Coq. In *Proceedings of the 12th Innovations in Software Engineering Conference* (pp. 1-5).

[16] Elsabbagh, F., Tine, B., Roshan, P., Lyons, E., Kim, E., Shim, D. E., ... & Lim, S. K. (2020). Vortex: opencl compatible risc-v gpgpu. *arXiv preprint arXiv:2002.12151*.

[17] Al Kadi, M., Janssen, B., & Huebner, M. (2016, February). FGPU: An SIMT-architecture for FPGAs. In *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays* (pp. 254-263).

[18] OpenHW Group. (2019). *OpenBus Interface (OBI) Specification*. Retrieved from `https://openhwgroup.org/specifications/`

[19] Vortex GPGPU. (n.d.). GitHub repository. Retrieved from `https://github.com/vortexgpgpu/vortex`

[20] Smol GPU. (n.d.). GitHub repository. Retrieved from `https://github.com/Grubre/smol-gpu`

[21] RISCY. (n.d.). GitHub repository. Retrieved from `https://github.com/mongrelgem/RISCY`