Developing a RISC-V Instruction Set Architecture for edge IOT Platform

G.Venkateswarulu [1], A.poojitha [2], G.Mustaqeem [3], M.Ashwini [4], K.B.Siri Chandana [5], B.Sunilkumar [6]

1Research Supervisor, Assistant Professor, Dept. Of ECE, ALTS, Anantapuram.

[2, 3,4,5,6] UG Scholar, Dept. Of ECE, ALTS, Anantapuramu.

ABSTRACT

This paper details the creation of a fully synthesizable 32-bit processor built around the open-source RISC-V (RV32I) instruction set architecture (ISA). The primary goal of this project was to design a cost-effective processor suitable for embedded systems. The design process includes developing a framework for RISC-V processor implementation and validation, incorporating essential tools and automated testing procedures. The resulting processor is a single-core design with straightforward hardware, created using Verilog HDL. The prototype is tested on an "Artix-7" FPGA platform, achieving a peak operating frequency of 32 MHz. This project begins with an emphasis on FPGA design and will eventually extend to a System on Chip (SoC) design. The development workflow covers the entire process from architectural conceptualization to RTL design, progressing through logic synthesis and physical layout, ultimately preparing for tapeout in a 90 nm CMOS technology. This approach demonstrates both the scalability and feasibility of implementing RISC-V processors in cost-sensitive applications.

Keywords: RISC-V, 32-bit processor, Verilog HDL, Embedded systems, FPGA prototyping, Artix-7, Low-cost design, System on Chip (SoC)

I. INTRODUCTION

The RISC-V instruction set architecture (ISA), originally developed to foster research and education in computer architecture, has rapidly evolved into a widely adopted open-source standard for both academic and industrial applications. RISC-V's modular design supports address spaces of 32-bit, 64-bit, and 128-bit, with the base integer ISA offering a minimal set of essential instructions. This core set of instructions is ideal for building operating systems, linkers, compilers, and assemblers. The RISC-V Foundation has provided a comprehensive ecosystem, including a tool chain that ensures interoperability across different components, facilitating the use of RISC-V in a variety of applications.

RISC-based architectures have long been integral to embedded systems, with ARM-based processors dominating the market for low-cost, energy-efficient mobile devices such as smartphones, tablets, and wearables. Other RISC architectures, including MIPS, Super H, and Open RISC, have also been utilized in embedded and IoT devices. Notably, MIPS processors have powered devices like the Nintendo 64 and PlayStation Portable, while Hitachi's SuperH architecture continues to evolve as open-source hardware due to the expiration of related patents.

32-bit RV32I processor tailored for embedded systems, with a particular emphasis on applications such as the Internet of Things (IoT), real-time embedded systems, and sensor technologies. The processor's design prioritizes low cost, low power consumption, and design simplicity, making it suitable for resource-constrained environments. The design process includes developing a complete framework for RISC-V processor design, validation, and implementation, with the goal of creating a processor capable of running Linux. The key stages of the design process, which are discussed in this paper, include architectural design, Register-Transfer Level (RTL) implementation, verification, and post-silicon validation. Additionally, the design leverages FPGA prototyping to evaluate performance and functionality, ensuring the processor meets the stringent requirements of

embedded systems. Furthermore, the paper presents a comprehensive analysis of the tool chain, simulation, and synthesis steps involved in developing the processor, highlighting optimizations made for power efficiency, cost reduction, and design complexity.

II. EXISTING METHOD

The Dominant Architectures: Examining the Status Quo

The embedded systems domain has long been dominated by a few key architectures, each with its own strengths and weaknesses:

ARM (Advanced RISC Machine): Ubiquitous in mobile devices and low-power applications, ARM's success stems from its energy efficiency, wide adoption, and a rich ecosystem of tools and software. However, its proprietary nature and licensing fees can be a barrier for some applications, particularly in cost-sensitive markets

MIPS (Microprocessor without Interlocked Pipeline Stages): Known for its simple architecture and widespread use in networking devices, MIPS have also seen use in embedded systems. However, its adoption has waned in recent years, partly due to competition from ARM and a less vibrant ecosystem.

X86 (Intel Architecture): While x86 dominates the desktop and server markets, its relatively high power consumption and complexity make it less suitable for deeply embedded applications where resources are limited.

These established architectures, while successful, present limitations for certain use cases, paving the way for alternative solutions like RISC-V.

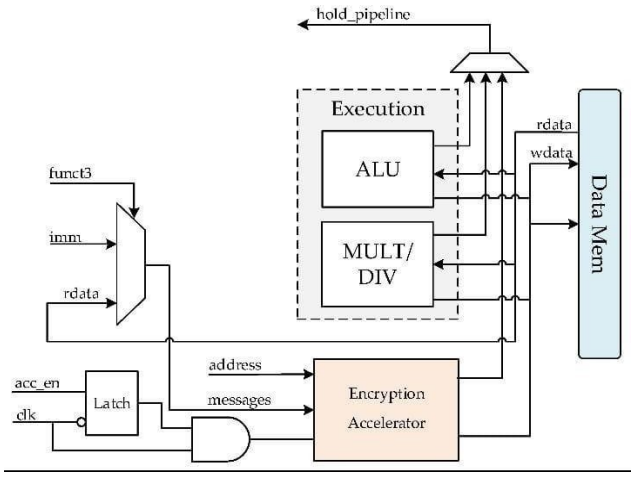

Fig 1: Low Power IoT RISC-V Processor

Proprietary CPU architectures like ARM, MIPS, and x86, each with unique advantages and disadvantages, have dominated the embedded system market. These architectures have been essential to general-purpose computing, networking devices, and low-power applications. They are less appropriate for resource-constrained and cost-sensitive applications like the Internet of Things (IOT), nevertheless, because to their drawbacks, which include licencing fees, architectural complexity, and power inefficiencies.

RISC-V, an open-source and modular Reduced Instruction Set Computing (RISC) architecture has become a potent substitute to address these issues. In contrast to conventional architectures, RISC-V eliminates the need for proprietary constraints and enables customisation and optimisation for particular applications.

The majority of the current system is based on proven embedded designs, although these systems have the following problems:

ARM and other proprietary ISAs have high licencing fees.

Limited adaptability in altering instruction sets for optimisations specific to a certain application.

Tradeoffs between performance and power that is not necessarily optimal for edge computing and the Internet of Things.

The search for an affordable, low-power, and adaptable embedded processor has prompted research on RISC-V, which gets rid of these limitations and offers a scalable solution for contemporary applications. Design Challenges for Resource-Constrained Environments

IoT and other embedded applications often operate within stringent resource constraints, posing unique design challenges:

Power Consumption: Minimizing power consumption is paramount for battery-powered or energy-harvesting devices. Every aspect of the design, from the ISA to the circuit level, must be optimized for energy efficiency.

Code Size: Limited memory resources in embedded devices necessitate efficient instruction encoding and compact code size. A streamlined ISA with minimal overhead is crucial.

Real-Time Operation: Many embedded systems, particularly in industrial control or automotive applications, require predictable and deterministic real-

time behaviour. This necessitates careful consideration of timing constraints and potential pipeline stalls.

Cost Sensitivity: Cost is often a critical factor in high-volume embedded deployments. A processor's complexity and manufacturing costs must be carefully balanced against its performance capabilities.

Addressing these challenges requires a holistic design approach, considering the interplay between the chosen ISA, microarchitecture, and circuit-level implementation.

32-bit RV32I fundamental integer instruction set hardware design architecture is presented. The execution employs a single-core, in-order, single-cycle architecture that is not bus-based and completely supports the RV32I instruction set. Among the application domains are acoustic signal processing, real-time embedded systems, sensor technologies, and many others. Target applications for the Internet of Things (IoT) and other embedded low-cost devices required a focus on price, power, and design complexity optimisation over strict timing constraints. The intended CPU is being prototyped on an FPGA device. With 32-bit architectures in mind, a set of RISC-V-based tools and a test framework were designed to facilitate the completion of our work on the RISCV-based compact processor. This work describes the created framework and its application to the original design.

III. PROPOSED METHOD

This research introduces an innovative approach to designing processors for embedded systems, particularly in the context of resource-constrained Internet of Things (IoT) devices. The study focuses on utilizing RISC-V, an open-source and modular processor architecture, to address the unique challenges of power consumption, hardware complexity, and scalability, which are often limiting factors for traditional proprietary architectures like ARM or x86. By leveraging RISC-V's open-source and modular nature, developers are able to customize the processor design to meet specific application requirements, something that is not easily achievable with proprietary solutions. The royalty-free design of RISC-V further eliminates licensing fees, which are a significant burden for cost-sensitive IoT applications.

Power efficiency is a primary concern for IoT devices, especially those operating on battery power or energy-harvesting systems. The proposed processor architecture addresses this issue by employing a single-cycle execution pipeline, which reduces power consumption by completing each instruction in a single clock cycle. This results in energy savings while maintaining high performance, thus extending the operational lifetime of IoT devices. Additionally, the simplicity of the RISC-V architecture, with its reduced instruction set, makes it highly suitable for energy-efficient designs without compromising performance, further contributing to lower power consumption and reduced hardware complexity. These design choices ensure that the processor is well-suited for IoT devices that operate in remote or energy-constrained environments.

The focus on the RV32I instruction set in the proposed processor ensures small code size and efficient memory utilization, making it ideal for embedded IoT systems with limited storage. By minimizing the instruction set, the design also reduces the hardware footprint, simplifying the processor and leading to cost savings while still providing the necessary computational capabilities for embedded tasks. The single-cycle execution model enhances the processor's efficiency by reducing latency, simplifying control logic, and ensuring deterministic timing, all of which are crucial for real-time IoT applications that require predictable behaviour.

In terms of the processor's microarchitecture, it is modular and optimized for low power and efficient resource utilization. The key components of the architecture include a fetch/decode unit, a register file for fast operand access, an arithmetic logic unit (ALU) for computational tasks, a memory control unit for data transfers, and a control unit that coordinates the operation of the processor. Each of these components is designed to contribute to low power consumption while maintaining high performance and functionality, ensuring the processor is capable of handling a variety of embedded IoT tasks.

In conclusion, this study presents a low-power, customizable processor architecture based on the RISC-V instruction set, specifically tailored for embedded IoT applications. By addressing the power, complexity, and scalability issues common in traditional processor architectures, the proposed design

provides a cost-effective and flexible solution for modern IoT devices. The research paves the way for future developments in RISC-V-based processors, highlighting their potential for enabling more efficient and scalable solutions for the ever-evolving IoT landscape

TABLE I: CISC&RISC Comparison

s.no	Feature	CISC(e.g.,X86)	RISC(e.g. ARM, RISC-V)
1.	Instruction	Complex,multi-step instructions	simple, single-step instruction
2.	Instruction	variable	Fixed
3.	memory access	can access memory directly	requires load/store instructions
4.	execution speed	slower per instruction but versatile	faster, more predictable
5.	power efficiency	generally lower	generally higher
6.	use cases	desktops,laptops,general -purpose CPUs	mobile, embedded, low-power applications

Proprietary CPU architectures like ARM, MIPS, and x86, each with unique advantages and disadvantages, have dominated the embedded system market. These architectures have been essential to general-purpose computing, networking devices, and low-power applications. They are less appropriate for resource-constrained and cost-sensitive applications like the Internet of Things (IOT), nevertheless, because to their drawbacks, which include licensing fees, architectural complexity, and power inefficiencies.

RISC-V, an open-source and modular Reduced Instruction Set Computing (RISC) architecture has become a potent substitute to address these issues. In contrast to conventional architectures, RISC-V eliminates the need for proprietary constraints and enables customisation and optimisation for particular applications.

The majority of the current system is based on proven embedded designs, although these systems have the following problems:

ARM and other proprietary ISAs have high licencing fees.

Limited adaptability in altering instruction sets for optimisations specific to a certain application.

Tradeoffs between performance and power that is not necessarily optimal for edge computing and the Internet of Things.

The search for an affordable, low-power, and adaptable embedded processor has prompted research on RISC-V, which gets rid of these limitations and offers a scalable solution for contemporary applications. Since every instruction is executed in a single cycle, the processor's clock period must be long enough to accommodate the worst-case path delay. Detailed timing analysis ensures that the adder for the program counter, instruction decoding, ALU operations, and memory accesses all meet the required timing constraints.

Redundancy for Branch Calculations: A second adder is employed specifically for branch and jump target address calculations, reducing the load on the primary ALU and helping achieve the required single-cycle performance.

Clock Frequency Optimizations:

The target clock frequency is determined by the worst-case delay across all components. Techniques such as logic partitioning and optimized routing in ASIC design are used to shorten these delays, thereby enabling higher operating frequencies

IV. RESULTS AND DISCUSSION

The Internet of Things (IoT) requires efficient, low-power computing solutions for edge processing. RISC-V, an open-source and customizable Instruction Set Architecture (ISA), offers a flexible and energy-efficient approach for IoT devices. Its adaptability allows processors to be tailored for specific applications, making it ideal for edge computing. This paper explores the implementation of RISC-V in IoT platforms, evaluating its performance, energy efficiency, scalability, and security. The analysis highlights RISC-V's potential to meet the evolving demands of the IoT ecosystem.

Test Environment:

A testbench created to give the ALU range of input circumstances is used to run the simulation in Vivado 2023.1.The testbench validates several ALU operating using a set of control signals and operand values.

The clock signal, or clk, regulates when things happen.

Enable: Establishes the timing of the ALU’s operations.

32-bit operands a [31:0] and [3:0] are employed in ALU operands.

Timing and Performance Analysis:

a) Static Timing Analysis (STA) Results

Proper data transfers were ensured by meeting setup and hold time requirements.

The system's effectiveness in real-time processing was confirmed by the calculation of the maximum operating frequency.

b) Memory Read/Write Operations

To confirm memory transactions, the data_read and data_write signals were used.

Accurate processing of RISC-V parameters was ensured by the system's proper data.

Successful data processing was confirmed by the memory enable (mem_write, mem_read) signals toggling appropriately.

c) Power Consumption Analysis

On the basis of clock frequency and switching activity, dynamic power consumption was examined.

Low-power design methods were used to reduce static power, also known as leakage power.

Result Behavioural Simulation

1. In the Flow Navigator, go to Simulation → Run Simulation → Run Behavioural Simulation.
2. Wait for the Waveform Viewer to open. Process to launch the project in Xilinx Vivado

Step 1: Open an Existing Project

1. In the Vivado Start Page, click Open Project.
2. Navigate to the folder where your project is saved.
3. Select the .xpr (Vivado Project File) and click Open.
4. The project will load, showing

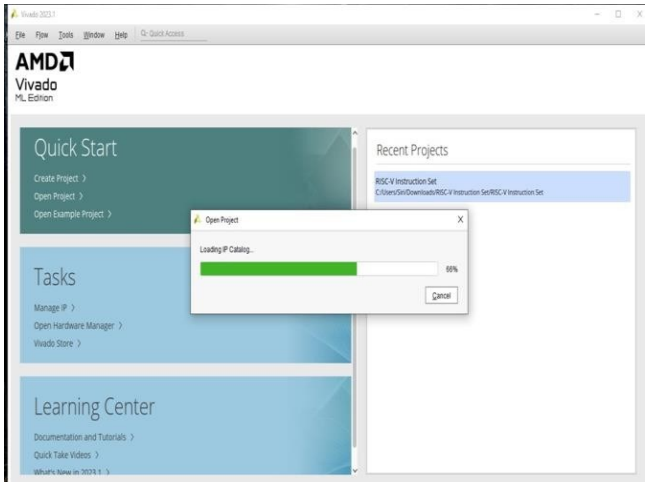

Fig 2: Opening an existing project

Result Synthesis:

1. In the Flow Navigator, click Synthesis → Run Synthesis.
2. Vivado will convert your HDL code into a netlist.
3. Once synthesis completes, review any warnings or errors.
4. Click Open Synthesized Design to inspect the generated hardware structure.

Result Implementation:

1. Click Implementation → Run Implementation in the Flow Navigator.
2. Vivado will perform placement and routing of the design. 3. Once completed, review timing reports to check for violations.

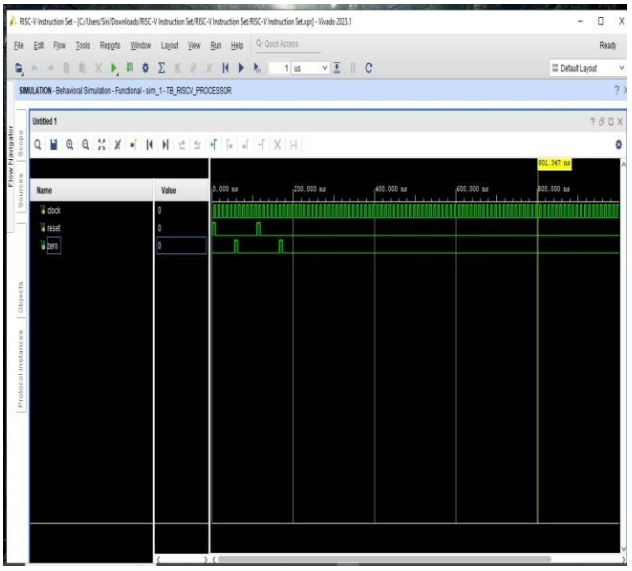

Fig 3: Simulation Results

Timing issues: Timing issues in a RISC-V processor can significantly impact the performance and reliability of an edge IoT computing platform. Common issues include instability in the clock signal, improper reset behaviour, and delays caused by pipeline stalls or data hazards. These problems can result in undefined system states or slower processing speeds.

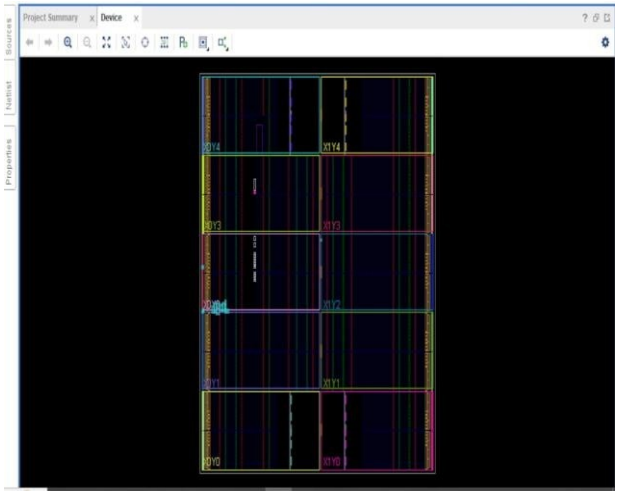

Fig 4: Implemented Design

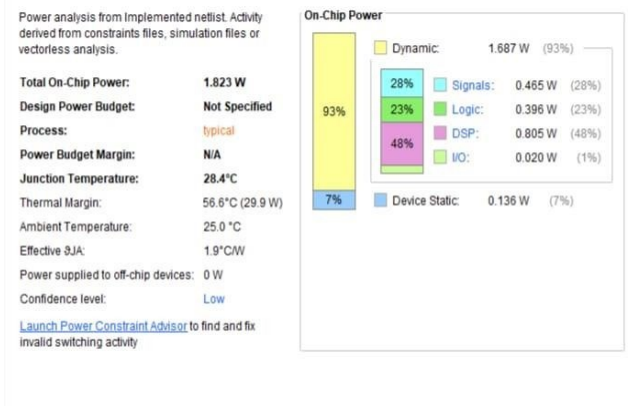

Fig 5: Power Consumption

Low power consumption in IoT edge devices is critical for extending battery life and ensuring efficient operation in resource-constrained environments. To achieve low power consumption, strategies such as voltage and frequency scaling, clock gating, and power gating can be implemented to reduce energy usage during idle and non-active periods. Additionally, optimizing the hardware, such as using energy-efficient RISC-V cores with simple pipelines and low-power memory, ensures minimal energy expenditure. Software optimizations, including event-driven processing and using lightweight operating systems can further reduce unnecessary computations. Combining these techniques allows IoT devices to function for extended periods without frequent recharging or external power sources, making them ideal for continuous operation in remote or off-grid environments.

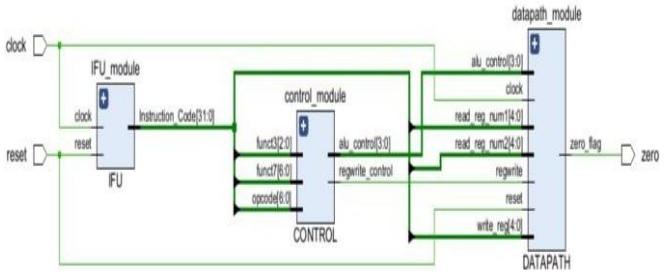

Fig 6: RTL Schematic

The RTL (Register Transfer Level) schematic in the image represents the internal structure of a RISC-V processor. It shows the data flow between functional units, including registers, ALU, memory, and control logic. The power consumption diagram highlights the processor's efficiency in low-power applications, making it suitable for IoT and embedded systems. This design ensures optimized performance with minimal energy usage, ideal for remote and battery-powered environments.

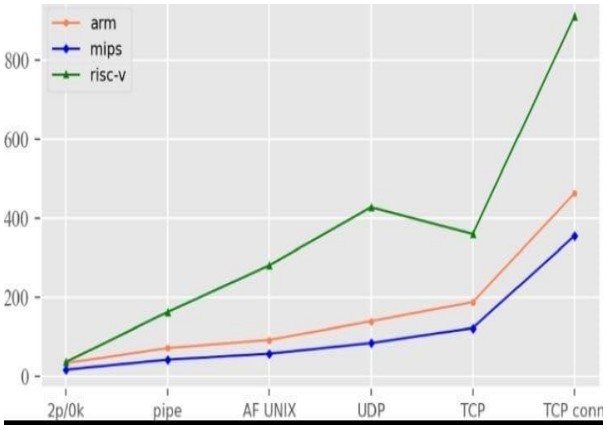

Fig 7: Performance Comparison of RISC-V, ARM, and MIPS Processors

As a synthesis tool, Cadence Genus is utilised. We Employed a multi-mode multi-corner flow with three corners, ten tracks, and regular-Vt cells of varying velocities. Detailed description of the NLDM under typical, ideal, and catastroph conditions. The synthesis results show that the given timing constraints (200 MHz clock) are met by all-time paths. The design process went through multiple iterations before reaching this level of quality. Making sure that reset signals affected all relevant flip-flops was the most time-consuming problem; and it was only shown via gate level simulations as opposed to RTLsimulations. In contrast, we didn't employ clock domaincrossing, another common ASIC design error.

Table II. Synthesis Results

Type	Instances	Area (μm^2)		Leakage power (μW)	
macros	21	1,841,871.1	(79.7%)	75,100.8	(72.1%)
sequential	25,664	263,667.6	(11.4%)	14,051.6	(13.5%)
inverter	4,572	6,449.2	(0.3%)	681.0	(0.7%)
buffer	6,344	11,842.4	(0.5%)	1,385.3	(1.3%)
logic	52,359	185,915.2	(8.0%)	12,917.4	(12.4%)
Total	88,960	2,309,745.5	(100%)	104,136.1	(100%)

IV. CONCLUSION

A unique single-cycle small embedded processor design that is modular and greatly expandable has been implemented by the authors of the current research. For data memory and instruction memory, on-chip block RAMs of 16KB and 64KB are used. A 32MHz maximum clock speed is achieved by the CPU using a Xilinx Vivado FPGA chip with a total power consumption of 7.9 mW. The current processor design paves the way for future implementations for specific and general IoT and embedded applications, considering the advantages of the expanding RISC V community as well as the existing tool-chain and software around this new instruction set and based upon this design paradigm. Future objectives include including an I/O bus, a multi-level cache system, interrupts, and a floating-point co-processor. These findings may be discussed in a subsequent study for application in research, education, and business while also opening the door to quicker designs. Additionally, it is true that porting a processor to silicon requires a lot of work, and at the moment, there are significant technical and financial obstacles to overcome that make this operation impractical. Common ones, such as in-depth familiarity with specialist design processes, pricey and exclusive EDA tools, and expensive or limited access to libraries, memory macros, and analog intellectual property.

V. REFERENCES

[1] A. Waterman, Y. Lee, D. A. Patterson, and K. Asanovic, “The RISCV instruction set manual, volume I: User-level ISA, version 2.0,” EECS Dept., Univ. California at Berkeley, Berkeley, CA, USA, Rep. UCB/EECS-2014-54, May 2014. [Online]. Available: http://www2.eecs.
berkeley.edu/Pubs/TechRpts/2014/EECS-2014-54.html

[2] B. Zimmer et al., “A RISC-V vector processor with simultaneous switching switched-capacitor DC–DC converters in 28 nm FDSOI,” IEEE J. Solid-State Circuits, vol. 51, no. 4, pp. 930–942, Apr. 2016.

[3] Y. Lee et al., “A 45 nm 1.3 GHz 16.7 double-precision GFLOPS/W RISC-V processor with vector accelerators,” in Proc. 40th Eur. Solid-State Circuits Conf. (ESSCIRC), Sep. 2014, pp. 199–202. [Online]. Available:
http://ieeexplore.ieee.org/document/6942056/I.S. 1963, pp. 271-350.

[4] J. C. Furgal and C. U. Lenora, ``Green routes to silicon-based materials and their environmental implications," Phys. Sci. Rev., vol. 5, no. 1, p. 24,

[5] N. M. Qui, C. H. Lin, and P. Chen, ``Design and implementation of a 256 bit RISC-V-based dynamically scheduled very long instruction word on FPGA," IEEE Access, vol. 8, pp. 172996 173007, 2020.

[6] K. Asanovi and D. A. Patterson, “Instruction sets should be free: The case for risc-v,” EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2014-146, Aug 2014.

[7] E. Farquhar and P. Bunce, The Mips Programmer’s Handbook, 1st ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. S. Williams, “vvp (1) - linux man page,” 2001, [Online; accessed 8 January-2017].[Online].Available:
https://linux.die.net/man/1/vvp.

[8] V. Patil, A. Raveendran, P. M. Sobha, A. D. Selvakumar, and D. Vivian, “Out of order floating point coprocessor for risc v isa,” in 2015 19th International Symposium on VLSI Design and Test, June 2015, pp. 1–7.

[9] Andrew Waterman, Yunsup Lee, David A. Patterson, and Krste Asanovic. “The RISC-V Instruction Set Manual, Volume I: Base User- Level ISA”. In Tech. Report UCB/EECS-2011-62, EECS Dept., UC Berkeley, pages 97–116, 2011.