ASIC Implementation of PULP RISC-V Core

Anirudh Rao M[1], Monish V[1], Manoj B M[1], N K Sindhu[1], Dr. Geethashree.A[2]

[1] B.E Students, [2] Associate Professor, [1,2]Department of Electronics and Communication Engineering, Vidyavardhaka College of Engineering, Mysore, Karnataka.

***Abstract*— The microprocessor used in this project is built using the RISC-V CV32E40P architecture. According to the literature review, we discovered that the previously modified 32-bit CV32E40P manufactured at non-RISC-V has a frequency of 160 MHz, so the performance is insufficient for many out-of-the-box applications. Therefore, we chose to work on the processor's performance aspect in order to better support other applications of this type. By making various modifications to the physical design of the CV32E40P, we are attempting to increase the CPU clock speed in this project from 500MHz to 600MHz. In essence, by compromising on the power and surface area components of the CPU architecture, we are essentially sacrificing performance. In this project, we won't change the CV32E40P's RTL code because we'll improve processor performance by altering physical design factors like floor planning, proper placement, etc. We'll also put more of an emphasis on resolving issues that came up when we increased the CPU clock speed, such as noise, in addition to other problems.**

Keywords—RISC-V (Reduced Instruction Set), Harvard Architecture, ALU (Arithmetic and Logical Unit), Verilog, Cadence, Genus.

I. INTRODUCTION

PULP - An Open Parallel Ultra-Low-Power Processing Platform. RISC-V - It is an open-source instruction set architecture (ISA). CV32E40P is a 4-stage in-order 32-bit RISC- processor core. CV32E40P started its life as a fork of the OR10N CPU core basedon the Open RISC-V ISA.

The simplest PULP based systems are our microcontrollers that can be configured to use any of the 32-bit RISC-V cores we have developed as the RI5CY which can add memory and has a certain number of peripherals.

II. LITERATURE SURVEY

Shashi Kumar V, Gurusiddayya Hiremath [1] One of the most significant challenges in system-on-a-chip (SoC) designs today is the increase in the rate of power consumption. Therefore, it is very important for chip designers to improve power management at the architectural level itself. As the chip sizedecreases, the leakage current increases rapidly. It is necessary to control the power of all design of 90 nm and below. As leakage current control is a high priority in design and implementation, as in many libraries and designs, the source dissipates power in CMOS. RISCV is a recently released ISA that is free and open source for industrial implementations underthe control of RISCV.

The main goal is to obtain an efficient low-power RISCV processor with DFT. The design adopts the following power control methods to divert and reduce leakage power.

i. Multi V_{th}

ii. Clock Tree Optimization and Clock Gating

iii. Multi supply voltage

iv. Power shut Off

Andrew S. Waterman [2] Power dissipation and energy efficiency are more concentrated in both simple and complex types of processors. Instruction set designers made extensive use of both techniques to reduce the relative energy cost of carrying instruction streams.

i. Increase the amount of work performed by one instruction.

ii. Reduce instruction size.

Finally, we discuss the impact of RVC on power efficiency, performance, and processor design. And compare the RVC code size with other commercial ISAs. RISC ISA variable-length codes can reduce the size of static and dynamic codes compared to fixed-length codes. It also avoids the degradation of low-capacity ISAs containing only short instructions. RVC instruction set analysis and RVC performance evaluation requires the collection of static and dynamic metrics from a subset of the SPEC CPU2006 benchmark. Static measurements are taken directly from object code and execution results. Dynamic measurements are obtained on the RISCV instruction set simulator by performing the test using a small input set.

Timothy Saxe, Pasquale Davide Schiavone, Frank Gürkaynak, Davide Rossi, Alfio Di Mauro, Mao Wang, Ket Chong Yap, Luca Benini Fellow [3] RISC-V Microcontroller unit (MCU) show the ability of the System-On Chip (SoC) to address the demanding situations of many rising IoT applications, such as

i. Interfacing sensors and accelerators with non- widespreadinterfaces.
ii. Performing on-the-fly pre-processing duties on statisticsstreamed from peripherals
iii. Accelerating near-sensor analytics, encryption, and systemmastering duties.

The contribution of the supplied heterogeneous SoC layout andsilicon demonstrator is summarized as follows.

1. Flexibility of Architecture
2. Power management
3. Leading area overall performance and power efficiency

Zekiye Eda Sataner, Etki Gür, Salih Bayar, Yusuf H. Durkaya [4] The proposed RISC V processor is deliberate making use of Verilog and it's far accomplished on Cyclone IV 4CE115 FPGA system reachable on Altera DE2-one hundred fifteen Board. Additionally, digital building agent and disassembler gadgets are created and dispensed as a chunk of this task.

Before making use of the goal RISC-V processor, the consumer can create system code making use of the digital building agent instrument. At that point, the created system code may be downloaded onto the RISC-V processor making use of UART. The on-line building agent and disassembler apparatuses are created with advancements, for example, HTML5, CSS and JavaScript. The proposed processor is a totally beneficial processor that makes use of RV32I base wide varietyeducational set with 37 guidelines.

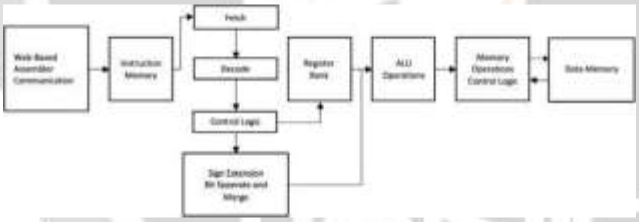

Web Based Assembler Communication
Instruction Memory
Fetch
Decode
Control Logic
Sign Extension Bit Separate and Merge
Register Bank
ALU Operations
Memory Operations Control Logic
Data Memory

Fig: Processor Design

Without an electronic assembler and RISCV disassembler, people waste pointless energy interpreting assembly code into machine language. At this point, an application that accepts RISCV assembly code as input in the "text area" of the websitechanges it to native code when output is generated. It uses the online build agent and disassembler communication between this structure and the FPGA provided by the UART. This information goes to the FPGA and playback works based on this information. All playback systems are built with Logisim. Logisim is an educational device for planning andreconstructing logic circuits. A 32-cycle RISCV processor is implemented on the base and fully supports the base RV32I integer control system. An LCD display on the FPGA was usedto show the output.

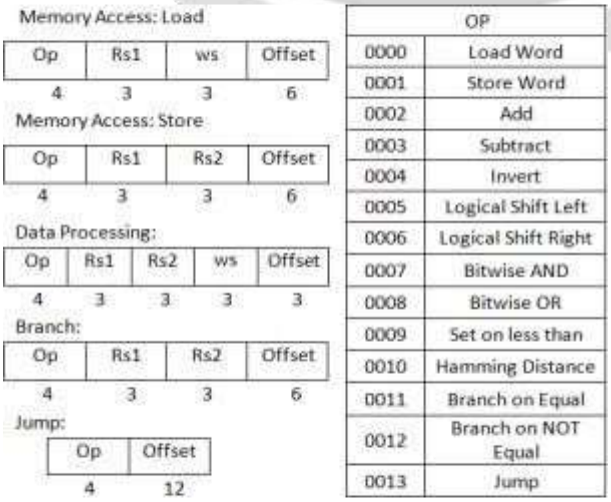
Saeid Moslehpour, Chandrasekhar Puliroju, Akram Abu-aisheh [5] It is done using the VHDL synthesis model. First, the processor's Datapath module shows how each module sends data from one module to the next, and then a screen showing how each bit of the processor is displayed from an external point. The shot is displayed in the top module view. Of the view. The full processor contains 9 blocks. This includes program counters (PCs), clock generators, command registers, ALUs, accumulators, decoders, I / O buffers, multiplexers, and memory. Together with these modules, a suitable bus is formed to form a processor that can store, load, and execute arithmetic and logical operations. The microinstruction code is put on the data bus, and the address bus and read cycle are initiated. To point to the subsequent instruction at the control memory address, the program counter is increased. The instruction register receives micro instructions from the data bus. There are two types of the instruction register.

a) Opcode, data operand: The opcode is sent to the ALUand decoder and decoded to generate a series of microoperations. The data operands are loaded into the data bus and moved to the ALU for special microoperations determined by the opcode.
b) Opcode, data operand address: The address of the data operand is loaded into the address bus and the memory read cycle begins. Here, the memory location in main memory specified by the address line is read, the information is transferred to the data bus, and the ALU performs the operation specified in its opcode.

A clock signal is produced by the clock generator. This signal serves as the decoder's input, which manages how the processor operates. It produces a reset pulse, which needs to be dynamically low. Instructions are retrieved from memory and kept in the instruction register. Only the rising edge of the clock is used to execute it. The ALU output is temporarily stored in the accumulator, a form of register that is only triggered on the positive edge of the clock. Data is output from memory, while mem rd, mem wr, and address are its inputs. Data will be written to memory if mem wr is high and scanned from memory to the data register if mem rd is high. Standard mathematical operations are carried out by the multiplexer known as the Arithmetic and Logic Unit (ALU). The clock's negative edge must be synced with ALU operations. The address multiplexer chooses one output from two inputs that are provided. The program counter's address is sent to the address buses when the fetch signal is high, which causes the instruction to be fetched. The operand address supplied in the instruction register's address field will always be sent to the address bus and subsequently accessed, supposing its low. The decoder makes sure the system is running in the right order. After exiting the instruction register, it will be transformed in accordance with the opcode. In essence, a decoder is a restricted state machine composed of states. After converting VHDL to PSPICE library and object files. The next step is composition. After converting VHDL to PSPICE library and object file. The next step is aggregation.

Chandran Venkatesan, Thabsera Sulthana, M Sumithra M.G [6] The MIPS processors are usually identical sort of designs. However, its modifications withinside the execution tiers like pipelining, unmarried or multiple. The operations are completed on chip registers instead of reminiscence locations, due to the fact the get admission to time differs for sign in as compared to reminiscence location. Due to the operation pace cell phones, pills and transportable gadgets are the use of ARM RISC processor. The drawback in transportable gadgets turned into that takes excessive energy which results in much less battery existence and reasons failure in silicon components of the gadgets. This drawback has been reduced on this task for the duration of by-by skip the pipelining tiers, however it reasons Dynamic energy dissipation. The energy dissipation is particularly because of undesirable switching tiers or a extra range of transitions gift withinside the device.

The pipelining level consists of fetch, decode, execute and reminiscence read/write operations.4-level pipelining and Clock gating is applied to lessen the overall performance and energy. Mainly this layout reduces the Dynamic energy dissipation as clock flip off clock sign whilst now no longer want and it load as much as 4 clock cycles in order thatconcurrently challenge may be done. Every output of the pipelining level is the subsequent nation input. The microinstructions completed on this layout had been separated.



Memory Access: Load

Op	Rs1	ws	Offset
4	3	3	6

Memory Access: Store

Op	Rs1	Rs2	Offset
4	3	3	6

Data Processing:

Op	Rs1	Rs2	ws	Offset
4	3	3	3	3

Branch:

Op	Rs1	Rs2	Offset
4	3	3	6

Jump:

Op	Offset
4	12

OP	
0000	Load Word
0001	Store Word
0002	Add
0003	Subtract
0004	Invert
0005	Logical Shift Left
0006	Logical Shift Right
0007	Bitwise AND
0008	Bitwise OR
0009	Set on less than
0010	Hamming Distance
0011	Branch on Equal
0012	Branch on NOT Equal
0013	Jump

Fig: Instruction Format of RISC

AGINETI ASHOK1, V. RAVI [7] RISC makes use of pipelining idea and range of check in to keep the intermediate facts values. The execution of a preparation is split into range of stages.

- The IF degree receives the following preparation from reminiscence with the cope with gift withinside the Program Counter (PC) and in a while it will likely be saved withinside the preparation checks in (IR).
- In ID degree preparation are decoded and evaluates this system counter preparation, and reads if any operand is neededfrom check in.
- In EXE degree the execution of ALU operation oncommands takes location.
- Memory Access degree takes location handiest if any cutting-edge preparation calls for the reminiscence access.

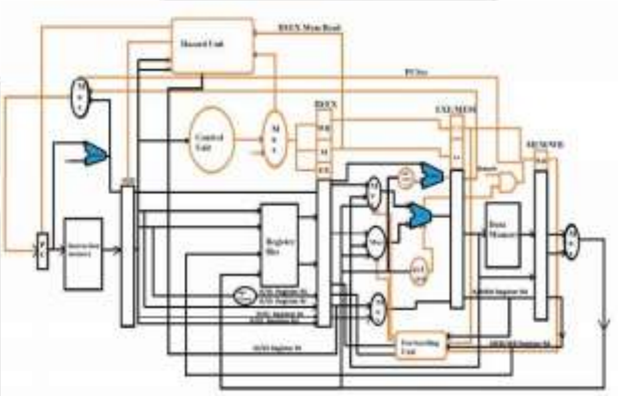


Fig: MIPS Processorwith forwarding unit

Hazard Unit

• The CPU's processing of pipelined instructions poses a risk. Incorrect compilation results if the next instruction cannot beexecuted at a specific clock cycle.
• The control unit determines whether a hazard occurs when commanded. If a hazard occurs, the control unit does not takeany action at this moment.

Forwarding Unit

Processors execute millions of instructions per second, creating a problem called pipeline blocking. This can besolved by stopping a stage in the pipeline. In other words, it splits the pipeline into two parts: getting commands and executing commands. The ALU forwarder is used to stop the pipeline to consume the ALU result directly. At risk, operands come from pipelined MEM/WB registers or EX/MEM. If there is no risk, the register file provides the operands to the ALU.

Execution unit

There is an ALU in the MIPS execution unit that executes operations using opcodes. To obtain the jump address, add the program counter value to the character extension unit that has been moved two units to the left. By adding the most important bit to hold the binary number's sign, the extended code unit increases the number. The ALU controller produces the control signal for the ALU. Two inputs are followed by a 2-bit data output in a circuit known as an ALU controller, which tells the ALU what kind of arithmetic and logical operations to carry out on the two inputs. results of simulations performed using the Xilinx stool.

Shubhodeep Roy Choudhury, Shajid Thiruvathodi, Vaidyanathan Seetharaman, Matt Cockrell, Jon Michelson, Jason Redgrave [8] STING is a patented design verification tool. For RISCV-based implementation developed by Provided by Valtrix Technologies Private Limited, India. STING is a lightweight program. (Similar to an embedded operating system). It can be used to create and run. Different workloads (oriented, algorithmic, or random) on the device under test (DUT) Check architecture compliance and function test. is software Guided Test Creation Methodology allowing portability of test stimuli throughout. SoC design lifecycle.

Hardware resource usage and the test command and memory size are it is also customizable and easy to control. You can easily target any SoC using parameters. Configuration on IoT microcontroller multi-core server. The figure shows the various components STING software stack.

It consists of test generators, checkers, device drivers, verification libraries/APIs and a microkernel. These are built into a bare metal ELF image which can be booted seamlessly on the DUT in any verification environment, such as in simulations, in circuit emulation, The simplest PULP-based system is a microcontroller. It can be configured to use the developed 32-bit RISC-V core, RI5CY. This allows you to add memory and use some peripherals. FPGA prototype and silicon. The user can control the generation of intelligent, self- checking, and architecturally correct test sequences in the portable test program using intuitive test configurations. After it is booted, the program can run tests targeting a specific SoC/CPU feature and report any anomalies detected during execution. Orientation tests (for scenarios such as mutual exclusion, code cross-modification, memory ordering, etc.) can be performed. Designed using a programming environment that allows users to write stimuli in assembly language as a phrase. This framework forms the basis for architectural conformance testing in RISCV.

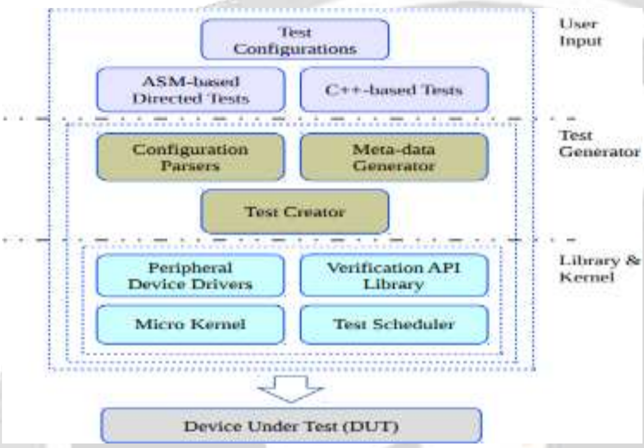


Fig: Different components of the software stack of STING

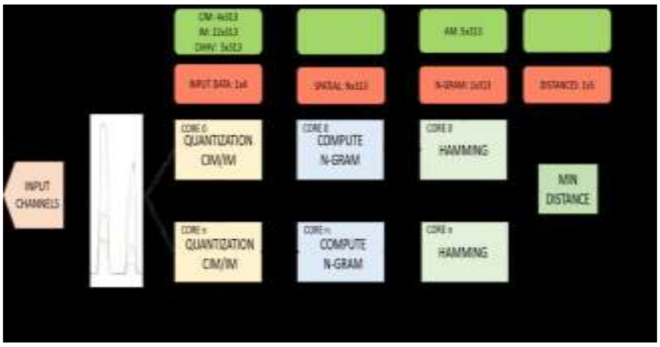
Set up for testing on STING. For tests that require complex programming configurations, e.g., CRC computations, fast Fourier transforms, etc.) or algorithms in nature (e.g., L3caches clash), STING provides a C++ based programming platform for developing incentives in advanced code. An abstraction was created on top of this structure to develop drivers for peripherals.

Devices in the SoC. Code snippets from OS, applications and tests are also portable to STING. Easy to use C++ based framework. New tests and feature enhancements were added to STING to cover the scenarios identified in the test plan for the PULPino RISCY core.

STING was successfully used for the functional verification ofPULPino RISCY core.

Fabio Montagna, Abbas Rahimi, Simone Benatti, Davide Rossi, Luca Benini [9] The circuits of the brain are huge in terms of the number of neurons and synapses, suggesting that large circuits are the basis of the processing power of the brain. High-dimensional (HD) computing, also known as super dimensional computing, is based on the understanding that the brain uses patterns of neural activity that are not easily associated with numbers. In fact, the ability of the brain to calculate numbers is weak.

We introduce an accelerator for all HD compute operations and optimize their memory access on the PULP platform. We target a silicon prototype of the PULP platform with 4 cores operating at 0.5 V, fabricated by



28 nm FD-SOI aka PULPv3.

• Our accelerator preserves the semantics of HD computation by avoiding any lossy optimization on binary hyper vectors, and its classification accuracy (average 92.4%) is similar. corresponds to the golden MATLAB.
• We are further investigating how HD compute acceleration can benefit from the new generation of PULPs with extended RISC-V-based processors (called Wolf) for efficient digital signal processing. energy results such as bit manipulation.
This instruction expansion, together with more than 8 cores, achieves 18.4 times the speed of single core PULPv3. Sequence diagram of HD computer processing with three main cores: map to HD Space encoder, time encoder, AM for classification. Each core in the processing chain is individually parallelized using an optimized version of the OpenMP directive, efficiently distributing the workload across multiple cores.

Davide Rossi, Francesco Conti, Antonio Pullini, Igor Loi, Luca Benini [10] In terms of embedded computer vision standards, multicore systems with closely connected cluster topologies have demonstrated promising results, providing optimum performance with constrained power budgets. We provide PULP (Ultra Low Power Parallel Processing Platform), a closely connected Open RISC ISA core cluster architecture with cutting-edge methods for quick and dependable performance. Utilizing the 28nm UTB FDSOI technology from STMicroelectronics, expandable power is possible.

In order to achieve great power efficiency through parallel processing, we made a significant contribution by introducing the PULP (Ultra Low Power Parallel Processing) platform, which consists of closely connected Open RISC core clusters. We examined the platform and found that it had a maximum power efficiency of 211 GOPS/W and an outstanding performance multiplier of 354x. As a CCTV-based camera method, it demonstrates how to switch between a low-power state that uses just 1.18 mJ per frame at 0.7 frames per second and a high-performance state that uses 27 frames per second and uses 12.6 mJ per frame. In order to compete with specialized mixed-signal accelerators like the 1.57 TOPS/W at Kim et al., our future work will be focused on pushing the PULP architecture to its theoretical limit of 1 GOPS/mW. While maintaining the software's general programmability, in terms of energy efficiency.

Noam Gallmann, Pirmin Vogel, Pasquale Davide, Schiavone Luca Benin [11] Ibex and CV32E40P, the two 32bit in order RISCV microprocessor cores explored in this work, both originate from a single parent design: RI5CY is among the earliest and most well-known opensource RISCV processor cores and was originally developed by The University of Bologna and ETH Zurich as main processing element for milliwatt range edge computing devices .The extensive use by the industrial community urged the two IPs to be moved to not- for-profit organizations to provide high-quality verification and industrial maintenance, still maintaining their permissive license and opensource policy. Ibex (originally known as Zeroriscy) is a 2stage RV32E, I[M]C core optimized for low- cost and low power. It has been contributed to low RISC in December 2018. Since then, Ibex has been extended with new optional features including support for a separate branch and jump target ALU (BTALU), an additional writeback pipeline stage (WBStage), static branch prediction (SBP), a single cycle integer multiplication unit (SCMult) and the RISCV draft bit manipulation extension (RV32B). In this paper, a Power Performance Area (PPA) comparison between the two cores is provided. Application performance, silicon area, performance, and energy efficiency are analyzed using various RTL parameters and application benchmarks. This article provides an update on the comparison of the two types of carrots Since the publication of Schiavone et al. based on earlier academic versions of the designs.

Simone Benatti, Davide Rossi, Andrea Bartolini, Antonio Mastrandrea, Christian Conficoni, Andrea Tilli, Luca Benini [12] In a variety of computer domains, power management of digital circuits is becoming more and more crucial. Dennard scaling limits the power and thermal capabilities of high-performance computing systems. The use of an open source, RISCV-based controller for the high-performance computing sector is assessed in this study. The power density needed to run the latest generation of processors at peak performance has continuously grown over the past ten years after Dennard Scaling came to an end. This rise in power density hinders supercomputer systems and has driven up power and cooling costs over time. improved calculational energy efficiency. The power controller must take the following actions to fulfil these objectives:

(i) Interface with a variety of on-chip and off-chip sensors, as well as power and actuator management interfaces.
(ii) Perform complex computational tasks, such as control automation, signal processing,optimization, and

machine learning algorithms.

ETHZ and the University of Bologna collaborated to create PULP, an open-source parallel computing platform that was first created to address the processing demands of demanding IoT applications. flexible data stream management, generally produced by several sensors. It comes with a collection of transfer-level IP addresses that have been registered and licenced from the Solder Pad to build a whole system-on-chip architecture. The processor, communication system, memory system, and peripheral systems are all included. PULP's GCC 7.1 toolchain supports the OpenMP programming paradigm, and real-time operating systems like ZephirOS and FreeRTOS also support it. This enables quick application porting, development, performance optimization, and debugging. This project suggests using a PULP-based controller for HPC computer node power management. Power management concerns for HPC systems are discussed in Part II. The history of the PULP is covered in Part III. Part V assesses the advantages of employing PULP-based power controllers for the most recent microcontrollers, and Part IV gives the firmware specifications for HPC system power controllers. This article examines how to utilise the Clay project to develop a power controller for an HPC system and defines the use of power controllers in HPC systems.

Michael Gautschi, Pasquale Davide Schiavone [13] Not only do Internet of Things devices have to operate within very strict power limits of a few milliwatts, but they also have to be flexible in their compute capacity from a few kOPS to GOPS. Operation closer to threshold (NT) can achieve higher power efficiency and parallel schemes can achieve capacity scalability. This article describes the design of a specific open source RISCV processor core developed for NT operation in tightly coupled multicore clusters. Introduces microarchitecture optimization and instruction scaling to increase compute density and reduce load on shared memory hierarchy. Over the past decade, we have been exposed to small, battery-powered Internet of Things (IoT) devices controlled by a microcontroller (MCU) that interact with the environment and communicate via public wireless channels. low capacity. Such devices require an ultra-low power (ULP) circuit that interacts with the sensor. In the next months [1 year], it's anticipated that demand for sensors and processors for platforms in the IoT market will rise. It is feasible to combine sensors since modern IoT terminals incorporate several sensors and are constructed around an MCU, which is primarily utilised for control and mitigation processing. The terminals typically don't need much upkeep or operation, need ULP operation, and are self-contained. These devices must be scalable in terms of performance and power efficiency since bandwidth needs range from electrocardiogram sensors and cameras to microphone arrays and necessary computer power. quantity.

Mike Thompson, Jingliang (Leo) Wang, Steve Richmond, Lee Moore, David McConnell, Greg Tumbush [14] The OpenHW Group is a global, member-run organization with the common goal of developing RISCV-compliant, open- source IP that meets commercial standards for delivery and quality. Open HW Group is a global non-profit organization whose members share the common goal of developing RISCV-compliant open-source intellectual property that meets commercial standards for distribution and quality.

The original designs of the processor originated from the PULP foundation of the University of Bologna and ETH Zurich under the direction of Professor Luca Benini. This initial RISCV implementation became well known in the RISCV community and quickly attracted the interest of many early SoC users. The Open HW team renamed the specific PULP kernels "COREV". The verification environment for these cores is collectively known as "corevverif". COREV's specifications, design, and verification code are open-source artifacts, licensed under Solder pad 2.0, an authorized license extension of Apache 2.0. reproducible documentation and verification.

This document details an open HW verification method using commercial tools with open scripting and a distributed team collaboration experience. Snippets from benchmarks, steps and comparisons, coverage, scripts, and more. Will be provided. Instructions for downloading and running the test suite to achieve 100% functionality coverage on the RISCV core will be shown. Readers will gain the ability to use the Open HW verification environment as the basis for designing and verifying a custom RISCV kernel with its own value-added features.

Francesco Conti, Pasquale Davide Schiavone, Davide Rossi [15] For inexpensive, battery-powered Internet of Things (IoT) end nodes, achieving a few milliwatts of power while meeting stringent performance requirements has emerged as one of their main problems. IoT end nodes must meet the demands for powerful computation, extreme energy efficiency, and affordability. These devices must be able to process, process, and transmit a wide range of signals, including biometric signals like ECG, EEG, or EMG (electronic health), signals from low-power cameras or webcams, vibration or audio signals in industries, or merely low bandwidth data like temperature or humidity (intelligent monitoring).

IoT devices must be programmable and are increasingly becoming whole node-based systems with sensors, microprocessors, specialized hardware, memory, and transmitter-wireless receiver that can run continuously for a number of years in order to support such a broad range of use cases. In order to accomplish this, IoT end nodes often function with extremely time-varying behaviour: they spend the majority of their time asleep but must react to external events by waking up. When this occurs, you frequently have to carry out a variety of quite diverse responsibilities, including B. Control interfaces that process information from environmental sensors, store it in either volatile or non-volatile memory, and send it through radio waves. In most cases, the name also has to execute rather intensive digital signal processing in order to perform semantic extraction of the relevant sensor data. Various core types may be appropriate for varying portions of the workload since application workloads have different compute needs. It can process proximity sensor data thanks to a robust digital signal processing (DSP) capability. In order to address various power, performance, capacity, and surface area requirements, certain base suppliers provide a range of alternative designs.

III. CONCLUSION

RISC-V allows small device manufacturers to create hardware without licensing fees and allows developers and researchers to design and test instruction set architectures for free Proven. Is important to. Software running on PULP/PULPino is written with RISC-V and C assembler.

Open-source ISA and industry people popularly refer RISC V as Linux of hardware development. This is the new revolution and future.

IV. REFERENCE

[1] Shashi Kumar V, Gurusiddayya Hiremath. "Low Power Implementation of RISC-V Processor". IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 6, Issue 3, Ver. II (May. -Jun. 2016), PP 59-64 e-ISSN: 2219 – 4200, p-ISSN No.: 2219 – 4197.

[2] Andrew S. Waterman. "Improving Energy Efficiency and Reducing Code Size with RISC-V Compressed".
University of California, Berkeley Technical Report No. UCB/EECS-2011-63 May 13, 2011.

[3] Pasquale Davide Schiavone, Davide Rossi, Alfio Di Mauro, Frank Gürkaynak, Timothy Saxe, Mao Wang, Ket Chong Yap, Luca Benini Fellow. "Arnold: an eFPGA- Augmented RISC-V SoC for Flexible and Low-Power IoT End-Nodes". IEEE Transactions on VLSI Systems, Vol. 29, No. 4, April 2021.

[4] Etki Gür, Zekiye Eda Sataner, Yusuf H. Durkaya, Salih Bayar. "FPGA Implementation of 32-bit RISC-V Processor with Web-Based Assembler-Disassembler". IEEE 2018 International Symposium on Fundamentals of Electrical Engineering (ISFEE) - Bucharest, Romania.

[5] Saeid Moslehpour, Chandrasekhar Puliroju, Akram Abuaisheh. "Design of RISC Processor Using VHDL and Cadence". K. Elleithy (ed.), Advanced Techniques in Computing Sciences and Software Engineering, DOI 10.1007/978-90-481-3660-5_89, Springer Science + Business Media B.V. 2010.

[6] Chandran Venkatesan, Thabsera Sulthana, M Sumithra
M.G. "Design of a 16-Bit Harvard Structure RISC Processor in Cadence 45nm Technology". 2019 5th International Conference on Advanced Computing & Communication.
[7] Agineti Ashok, V. Ravi. "ASIC Design of MIPS Based RISC Processor for High Performance". 2017 International Conference on Nextgen Electronic Technologies.

[8] Shubhodeep Roy Choudhury, Shajid Thiruvathodi, Vaidyanathan Seetharaman, Matt Cockrell, Jon Michelson, Jason Redgrave. "Verifying PULPino RISCY Core for a Google Accelerator with STING".

[9] Fabio Montagna, Abbas Rahimi, Simone Benatti, Davide Rossi, Luca Benini. "PULP-HD: Accelerating Brain-Inspired High-Dimensional Computing on a Parallel Ultra-Low Power Platform." IEEE/ACM Design Automation Conference (DAC), 2018. arXive preprint arXive: 1804.09122.

[10] Francesco Conti, Davide Rossi, Antonio Pullini, Igor Loi, Luca Benini. "Energy-Efficient Vision on

the PULP Platform for Ultra-Low Power Parallel Computing”. IEEE Xplore: 18 December 2014
DOI: 10.1109/SiPS.2014.6986099.

[11] Noam Gallmann, Pirmin Vogel, Pasquale Davide, Schiavone Luca Benini. “A Cost-Benefit Analysis of Ibex and CV32E40P Regarding Application Performance, Power and Area”. CARRV 2021, June 2021, worldwide © 2021 Association for Computing Machinery.

[12] Andrea Bartolini, Davide Rossi, Antonio Mastrandrea , Christian Conficoni , Simone Benatti, Andrea Tilli , Luca Benini. “A PULP-based Parallel Power Controller for Future Exascale Systems”. A. Bartolini et al, "A PULP-based Parallel Power Controller for Future Exascale Systems" 2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Genoa, Italy, 2019.

[13] Michael Gautschi, Pasquale Davide Schiavone.
“Near-Threshold RISC-V Core with DSP Extensions for Scalable IoT Endpoint Devices”. This work was supported in part by the FP7 ERC Advance Project MULTITHERMAN under Grant 291125.

[14] Mike Thompson, Jingliang (Leo) Wang, Steve Richmond, Lee Moore, David McConnell, Greg Tumbush. “Jump start your RISC-V project OpenWH.

[15] Pasquale Davide Schiavone, Francesco Conti, Davide Rossi. “Slow and Steady Wins the Race? A Comparison of Ultra-Low-Power RISC-V Cores for Internet-of-Things Applications”. 2017 27th International Symposium on Power and Timing Modeling, Optimization and Simulation.Systems