IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ASIC IMPLEMENTATION OF RV32IMAC RISC- V SOC

[1]Dr.Cyril Prasanna Raj P, [2]Divya R, [3]Geetha V, [4]Aishwarya K R, [5]Mahalakshmi H,

[1]Professor, [2,3,4,5]students

[1,] Electronics and Communication Engineering, Cambridge Institute of Technology, Bengaluru, India

[2,3,4,5]Electronics and Communication Engineering, Cambridge Institute of Technology, Bengaluru, India

ABSTRACT

This paper delves into the ASIC implementation of an RV32IMAC RISC-V System-on-Chip (SoC), focusing on its adaptation for diverse surveillance applications. By harnessing the capabilities of RISC-V architecture, the SoC is designed to offer a flexible and efficient platform for surveillance tasks in varied environments, including industrial sectors, war zones, and radioactive fields. Through meticulous architectural design and optimization strategies, the SoC achieves a balance between performance, power efficiency, and cost-effectiveness. Notably, it integrates specialized instructions tailored for surveillance operations, alongside robust support for sensor integration and real-time data processing. Furthermore, the SoC's implementation leverages advanced techniques to ensure reliability, scalability, and compatibility with emerging surveillance systems. With its ability to handle complex tasks autonomously and facilitate seamless communication via IoT-based services, the ASIC implementation of the RV32IMAC RISC-V SoC represents a significant advancement in the realm of surveillance technology, promising enhanced situational awareness and threat mitigation capabilities.

Key Word: RV32IMAC RISC-V System-on-Chip (SoC), cost-effectiveness

I. INTRODUCTION

The advent of the RISC-V instruction set architecture has ushered in a new era of openness and innovation in the field of computing. Among its various configurations, the RV32IMAC variant stands out due to its comprehensive set, encompassing integer multiplication and division (M), atomic operations (A), and compressed instructions (C). This versatile architecture, coupled with the scalability and customizability inherent in the RISC-V ecosystem, has sparked widespread interest in leveraging it for a myriad of applications. In parallel, Application-Specific Integrated Circuits (ASICs) have emerged as indispensable tools for realizing efficient and optimized hardware implementations of complex systems. ASICs offer unparalleled performance, power efficiency, and integration by tailoring the hardware at the silicon level to specific applications. The ASIC implementation of an RV32IMAC RISC-V System on Chip (SoC) represents a convergence of these two technological frontiers, promising to deliver high-performance computing solutions customized to diverse use cases. By translating the RV32IMAC architecture into a hardware design, integrating the RISC-V processor core with peripheral components, and optimizing for specific application requirements, ASICs unlock new opportunities for innovation in embedded systems, IoT devices, edge computing and beyond.This paper explores the intricacies of ASIC design methodologies and techniques employed in realizing an SoC based on the RV32IMAC architecture.

From architectural exploration to RTL design, synthesis, physical design, and verification, each stage of the ASIC implementation process demands meticulous attention to detail and optimization strategies to achieve the desired performance metrics within the constraints of power, area, and timing. Further more, the ASIC implementation of RV32IMAC RISC-V SoCs opens avenues for a wide range of applications, from low-power embedded systems to high-performance computing clusters. Its versatility, combined with the open-source nature of the RISC-V architecture, empowers developers to innovate and customize solutions tailored to specific use cases and market demands. By examining architectural considerations, optimization strategies, and performance evaluation metrics, we aim to provide insights into the evolving landscape of ASIC-based RISC-V SoCs and their impact on the future tecnology. The ASIC implementation of RV32IMAC RISC-V SoCs finds applications across a wide range of industries and domains, owing to their versatility, performance and customizability. The majorityof applications include:

1. Embedded systems.
2. Edge computing.
3. High-performance computing.
4. Storage solutions.
5. Security and cryptography

II LITERATURE REVIEW

Table 1.Literature Review of ASIC implementation of RV32IMAC RISC v soc

Published year	Author Name	Title of the paper	Proposed technique	Limitations
2019 [1]	S. Feng, J. Wu, S. Zhou, and R. Li,	The implementation of LeNet-5 with NVDLA on RISC-V SoC	-optimizations for performance and resource utilization	-Performance Trade offs -Compatibility -Scalability
2020 [2]	X. Zhong, C.-W. Sham, and L. Ma,	A RISC-V SoC for mobile payment based on visible light communication	-Power Efficiency -Security Features	-Security and Privacy -Mobility and Alignmen
2021 [3]	C. Duran, H. Gomez, and E. Roa	AES sbox acceleration schemes for lowcost SoCs	-Evaluation and Performance Analysis - Efficient Hardware Implementation	- Power Consumption - Cost - Scalability
2019 [4]	D. Jayasinghe, A. Ignjatovic, and S. Parameswaran	SCRIP	The technique involves implementing secure random clock execution	-Security Assurance -Complexity
2017 [5]	S. Hou, Y. Zhou, H. Liu, and N. Zhu	Wavelet Support Vector Machine Algorithm in Power Analysis Attacks	- Vector Machine (SVM) -Wavelet Transform	-Complexity -Generalization
2019 [6]	E. De Mulder, S. Gummalla, and M. Hutter	Protecting RISC-V against side-channel attacks	-Software-based Techniques -Instruction Set Modifications	-Performance Overhead -Area Overhead

III METHODOLOGY

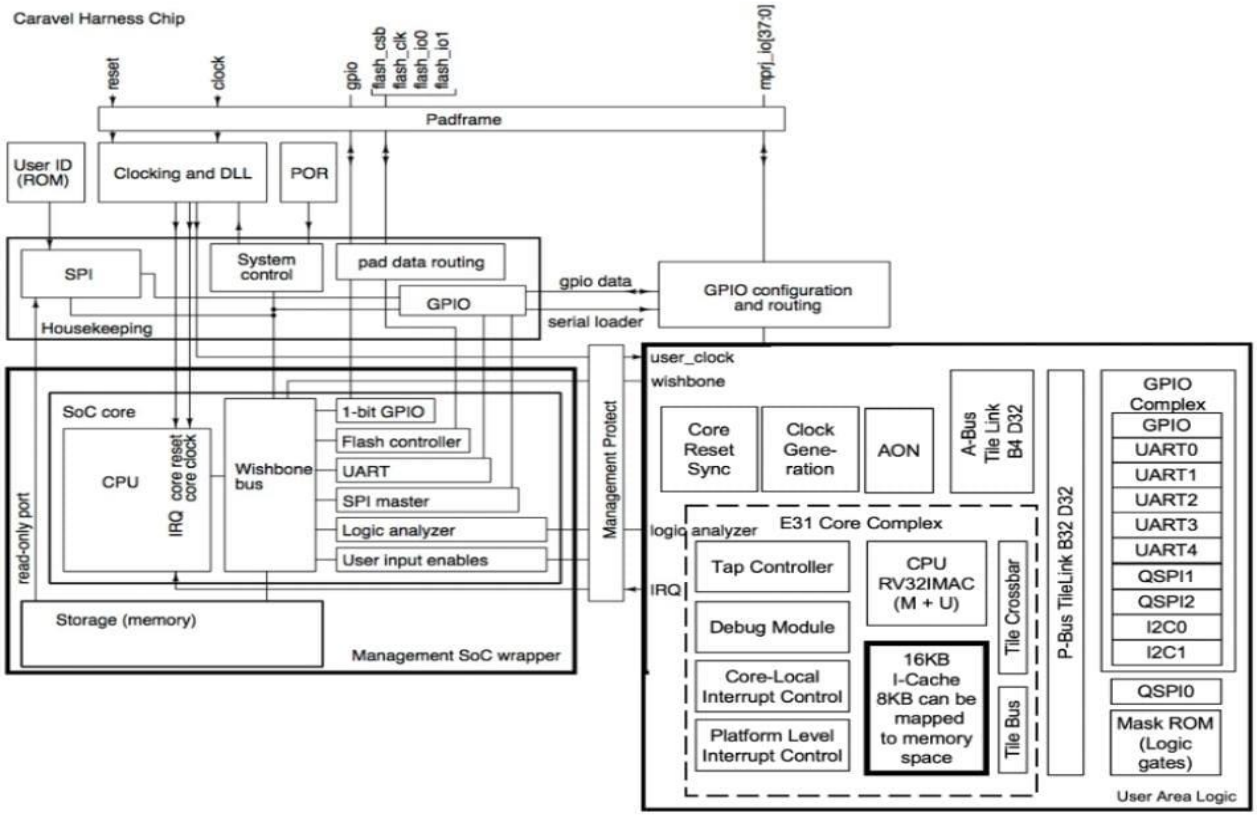

Fig 1. Block Diagram of Caravel and Rocket E31

The realm of embedded systems faces challenges in achieving optimal performance, power efficiency and flexibility while adhering to stringent design constraints. Existing solutions often struggle to balance these factors, leading to compromises in functionality and scalability. There is a need for a robust and versatile hardware platform that can meet the evolving demands of embedded applications without sacrificing efficiency or customizability.Current embedded systems depend on a range of architectures and implementations, each with its own set of strengths and limitations. While some systems leverage proprietary architectures for optimized performance, they often lack flexibility and scalability, limiting their applicability to specific use cases. Other systems utilize open-source architectures like RISC-V to achieve greater flexibility, but may face challenges in achieving optimal performance and efficiency due to the lack of tailored hardware implementations.The proposed system involves the ASIC implementation of an RV32IMAC RISC-V SoC, offering a balance of performance, power efficiency and flexibility tailored for embedded applications. By customizing the hardware at the silicon level, the ASIC implementation can optimize resource utilization, minimize power consumption and maximize performance for a wide range of applications.

To set up the environment for running Docker containers and QEMU, the first step involves configuring QEMU and Docker Buildx. Next, environment variables defining the paths to the Process Design Kit (PDK) and OpenLANE directories are exported. Following this, the PDK tarball artifact, generated in a prior job, is downloaded and unpacked into the designated PDK directory. Necessary dependencies are then installed using the make command, including various tools essential for the project's execution. Subsequently, the project undergoes hardening using OpenLANE, a tool tailored for digital ASIC design. To ensure the correctness of the design, RTL tests are executed, validating input/output ports, logic analyzer functionality, and stimulus response. Additionally, Gate Level (GL) tests are run to verify the gate-level implementation aligns with the intended behavior specified at the RTL level, ensuring the robustness and reliability of the project's implementation.

The System-on-Chip (SoC) architecture serves as the backbone of many electronic devices, seamlessly integrating various components to enable their operation. At the heart of this architecture lies the RV32IMAC RISC-V CPU, which acts as the central processing unit responsible for executing instructions and managing computations. This CPU is equipped with a robust instruction set, encompassing fundamental integer operations, as well as advanced functionalities like multiplication/division, atomic operations, and compressed instructions, catering to a wide range of computing tasks.Memory management is a critical aspect of SoC design, and it involves separate blocks dedicated to storing program instructions (I-Cache) and data. These memory blocks are optimized to enhance performance, often incorporating techniques such as caching to reduce access latency and improve overall system efficiency.Peripheral Interfaces play a crucial role in facilitating communication between the SoC and external devices. These interfaces support a variety of protocols and standards, enabling seamless interaction with peripherals such as sensors, actuators, and communication modules. Common examples include GPIOs for general-purpose input/output operations, UARTs for serial communication, and SPI/I2C interfaces for connecting to lower-speed peripherals.

Ensuring coordinated operation and error handling within the SoC are Clocking and Reset mechanisms, which provide synchronized timing signals and a means to restart the system in case of errors. Meanwhile, Management Logic oversees various system tasks, including interrupt handling, power management, and clock synchronization, to ensure smooth and efficient operation.Memory Blocks, which may include Static Random-Access Memory (SRAM) for high-speed data access and embedded Non-volatile Memory (eNVM) like Flash for program storage, play a vital role in data storage and retrieval within the SoC.To describe and design the functionality of these components, Hardware Description Languages (HDLs) such as Verilog or VHDL are utilized. These languages allow engineers to specify the behavior and interaction of the CPU, memory blocks, and peripheral interfaces in a concise and structured manner.Furthermore, Electronic Design Automation (EDA) Tools play a pivotal role in the design process, enabling tasks such as simulation, synthesis, place and route, and verification. These tools ensure that the SoC design meets specified requirements and standards, leading to the development of reliable and efficient electronic systems.

IV RESULT

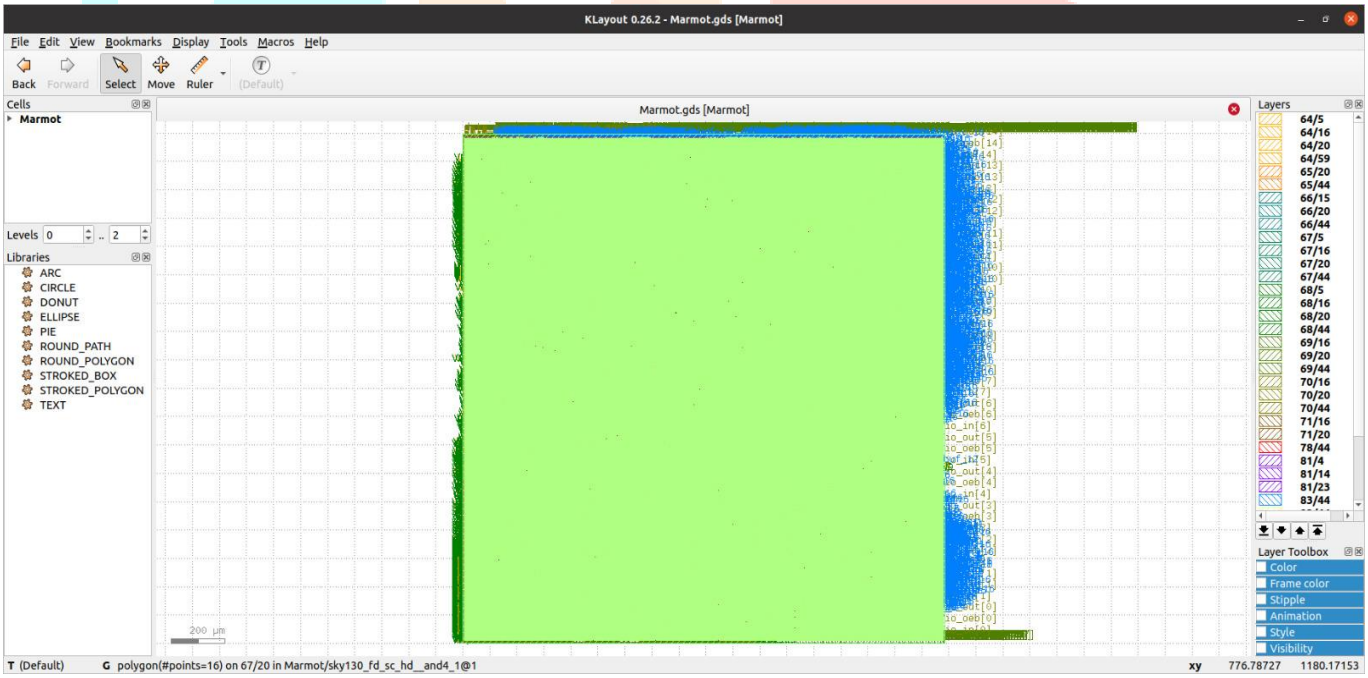

Fig 2 : Klayout

The KLayout tool was utilized to analyze the GDS2 file, providing valuable insights into the layout design. The output revealed detailed information about the layout structure, including key features and potential areas for optimization. Analysis of the GDS2 file identified any anomalies or errors present, allowing for informed decision-making regarding design improvements. Furthermore, KLayout's versatile capabilities enabled efficient navigation and visualization of the layout, facilitating a comprehensive understanding of its intricacies. Overall, the utilization of KLayout enhanced the quality and efficiency of the layout analysis process, contributing to the overall success of the project.

V CONCLUSION AND FUTURE SCOPE

The ASIC implementation of RV32IMAC RISC-V SoC sets the stage for a new era of efficiency, flexibility, and innovation in embedded systems design. By addressing the shortcomings of current solutions and offering a robust hardware platform optimized for performance, power efficiency, and scalability, it promises to unlock new opportunities for advancement in embedded computing and technology. As we continue to push the boundaries of hardware design and innovation, the ASIC implementation of RV32IMAC RISC-V SoC remains at the forefront of driving progress and shaping the future of embedded systems.Exploring novel approaches to system-on-chip (SoC) integration and interconnect architectures can further enhance the scalability and efficiency of ASIC implementations of RV32IMAC RISC-V SoCs. Moreover, fostering collaboration between academia, industry, and standardization bodies will facilitate the development of comprehensive toolchains and ecosystems, accelerating the adoption and deployment of RISC-V-based solutions worldwide. In this rapidly evolving landscape, the ASIC implementation of RV32IMAC RISC-V SoCs holds the potential to revolutionize embedded systems and drive innovation across diverse application domains, paving the way for a future of unparalleled efficiency, performance, and versatility.

VI REFERENCES

1 S. Feng, J. Wu, S. Zhou, and R. Li, ‘‘The implementation of LeNet-5 with NVDLA on RISC-V SoC,’’ in Proc. IEEE 10th Int. Conf. Softw. Eng. Service Sci. (ICSESS), Oct. 2019, pp. 39–42.

2 X. Zhong, C.-W. Sham, and L. Ma, ‘‘A RISC-V SoC for mobile payment based on visible light communication,’’ in Proc. IEEE Asia Pacific Conf. Circuits Syst. (APCCAS), Dec. 2020, pp. 102–105.

3 C. Duran, H. Gomez, and E. Roa, ‘‘AES sbox acceleration schemes for lowcost SoCs,’’ in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2021, pp. 1–5.

4 D. Jayasinghe, A. Ignjatovic, and S. Parameswaran, ‘‘SCRIP: Secure random clock execution on soft processor systems to mitigate powerbased side channel attacks,’’ in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), Nov. 2019, pp. 1–7.

5 S. Hou, Y. Zhou, H. Liu, and N. Zhu, ‘‘Wavelet support vector machine algorithm in power analysis attacks,’’ Radioengineering, vol. 26, no. 3, pp. 890–902, 2017.

6 E. De Mulder, S. Gummalla, and M. Hutter, ‘‘Protecting RISC-V against side-channel attacks,’’ in Proc. 56th ACM/IEEE Annu. Design Autom. Conf. (DAC), Jun. 2019, pp. 1–4.