國立陽明交通大學
NATIONAL YANG MING CHIAO TUNG UNIVERSITY

RISC-V

Revolutionizing Computing with RISC-V :
An In-Depth Analysis of its Processor and its Extensive Capabilities

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING, NATIONAL YANG MING CHIAO TUNG UNIVERSITY, TAIWAN

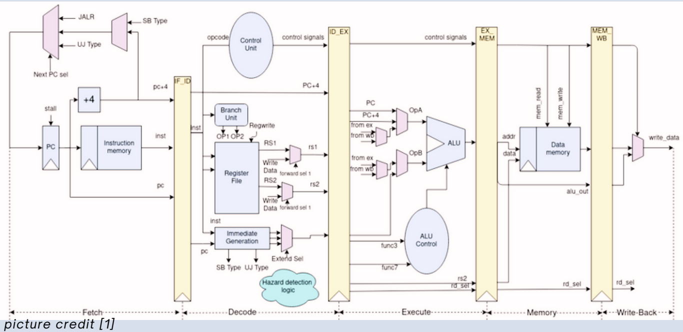
AUTHOR: HENRY BETSALEEL WILSONSOM DIEUDONNE (李松磊) | SUPERVISOR: PROF. BO-CHENG CHARLES LAI (賴伯承)

ABSTRACT

RISC-V is an open-source Instruction Set Architecture (ISA) that adheres to the principles of Reduced Instruction Set Computing (RISC). It serves as a platform for developing processors suited for a wide range of applications, spanning from embedded systems to high-performance computing. This project focuses on the design and synthesis implementation of a RISC-V processor using Verilog. Furthermore, the study explores OpenLane, a state-of-the-art open-source environment, which enables the development of a System on Chip (SoC). The successful creation of a fully open-source toolchain for RTL simulation, generation, GDS-II conversion, PDKs, IP libraries, and I/O blocks is a significant milestone in bridging the historical gap between hardware and software development. The project also involves a comprehensive review of a noteworthy paper/project claiming to have achieved a groundbreaking RISC-V-based SoC, generated using Chisel and employing fully open-source technologies in the tape-out process.

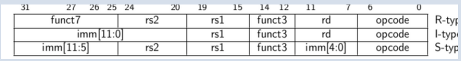
PROCESSOR DATAPATH

The RISC-V architecture incorporates a datapath that comprises several essential stages. The system's design includes the following stages: fetch, decode, execute, memory, and write back. Special attention has been dedicated to managing and addressing potential structural hazards that may arise during operation. The Instruction Set Architecture used is the RISC (RV32IM) which means 32 bits per instruction.



picture credit [1]

Instruction type's overview | supported instructions



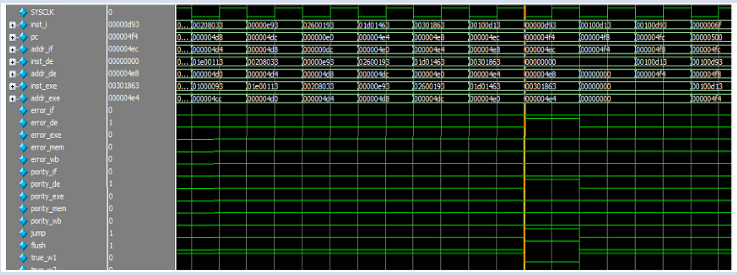
31 27 / 26 25	24 20	19 15	14 12	11 7	6 0	
funct7	rs2	rs1	funct3	rd	opcode	R-typ
imm[11:0]		rs1	funct3	rd	opcode	I-typ
imm[11:5]	rs2	rs1	funct3	imm[4:0]	opcode	S-typ

- J_types
- **Jalr**
- **Jal**

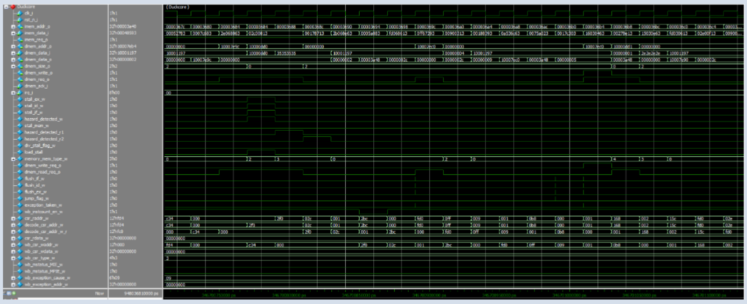
- R_types
- **add**
- **sub**
- **xor**
- **or**
- **and**
- **sll**
- **slt**
- **sltu**
- **sra**

- I_types
- **addi**
- **lw**
- **slli**
- **subi**
- **xori**
- **ori**
- **andi**
- **slli**
- **slti**

DECODING STAGE



CORE SIGNALS WAVEFORMS



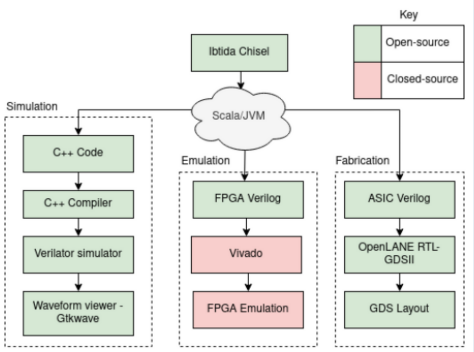
RELATED STUDIES

Paper/Project Title	Main Contributions	Methodology	Performance Metrics
"RISC-V: The Free and Open RISC Instruction Set Architecture"	Introduction to RISC-V architecture	Conceptual analysis	ISA specification
"A 28nm SoC with a 1.16 GHz 6-Core RISC-V Processor, a 4 Tbps Network-on-Chip, and a 1024-Core Neural Network Accelerator"	Integration of RISC-V in multi-core SoC and neural network accelerator	Case study, implementation	Frequency, throughput, area utilization
"Boom: A RISC-V Out-of- Order Processor Core"	Design and implementation of out-of-order RISC-V core	Design analysis, synthesis	IPC, cycle time, area utilization
"PULPino: An Open-Source Single-Core RISC-V SoC Including a Time- Predictable Processor and FPGA-Based Fast Prototyping"	Open-source single- core RISC-V SoC design	Prototype development, FPGA implementation	Predictability, performance, power consumption
"lowRISC"	Open-source RISC-V SoC design	Design, verification	Integration, functionality, configurability
"VexRiscv"	Highly configurable RISC-V CPU core	Implementation, synthesis	Customization options, area utilization

OPENLANE AND CHISEL

Among the journal papers on RISC-V core implementations, one notable study is the Chisel paper. Chisel was utilized as a frontend tool to enhance the design's openness and functionality. Chisel is a domain-specific language embedded within Scala, which offers advanced programming language features for designing circuits, in contrast to traditional hardware description languages (HDLs) like Verilog or VHDL. In our implementation, Chisel served as a powerful tool and model for circuit design, allowing for improved flexibility and expressiveness in creating the RISC-V core.

Overview of different stages used in Ibtida during the tape-out of the paper design



picture credit [1]

CONCLUSION

This project has provided me with a great opportunity to learn from an open-source implementation of a RISC-V processor core. RISC-V has the potential to revolutionize the world of computing with its openness and extensive capabilities. It fosters a collaborative community of developers and researchers, driving innovation in the field. This project has been a valuable learning experience that deepened my understanding of RISC-V. However, there is still much more to explore and discover. The Verilog-based processor core developed is still in its early stages.

Looking ahead, further exploration and advancements in RISC-V hold great promise for shaping the future of computing technologies.

I try my best to stay up to date with the latest literature. The possibilities are endless, and by embracing the potential of RISC-V, we can contribute to the ongoing evolution of the computing landscape.

REFERENCES

[1] Muhammad Shahzeb Khan, Aireen Amir Jalal, Ahmed, S., Ali Ahmed Ansari and Raza, A. (2021). IBTIDA: Fully open-source ASIC mplementation of Chisel-generated System on a Chip. doi:https://doi.org/10.36227/techrxiv.16663738.v1.

[2]Patterson, D. A., & Hennessy, J. L. (2017). Computer Organization and Design RISC-V Edition: The Hardware Software Interface. Morgan Kaufmann.

[3] Holler, R., Haselberger, D., Ballek, D., Rossler, P., Krapfenbauer, M. and Linauer, M. (2019). Open-Source RISC-V Processor IP Cores for FPGAs — Overview and Evaluation. 2019 8th Mediterranean Conference on Embedded Computing (MECO). doi:https://doi.org/10.1109/meco.2019.8760205.