

## Goal

- To **design** and synthesize a **software solution** to fill an academic **market void** for **RISC-V** computer architecture simulation.
- Provide an integrated **development environment** equipped with **useful educational tools**, such as a real-time code **simulator**.
- Employ bleeding-edge **cross-platform** and open-source frameworks while staying completely **open-source**.
- Adapt** to **industry** changes and demand as well as the nature of RISC-V architecture.

## Motivations and Objectives

- Motivations
  - RISC-V is increasingly appearing in the **spotlight** due to increased **versatility and functionality**.
  - An **open ISA**, its roots in reduce instruction set computing allow for an **open environment** for design, expansion, and manufacturing of RISC-V based **chips and software**.



- Objectives
  - To fill the **vacancy** in the market for a lightweight simulator of RISC-V architecture for **education and industry** usage.
  - Be an open-source, Java based, **IDE and simulator**, with the intention to be an **intuitive cross-platform tool** under an open license.

## Research Challenges

- Integration**, due to the nature of collaborative programming, and ensuring each class and package **cohered** with the others.
- Comprehension and implementation of numerous **extensions**, such as Quad-Precision Floating-Point arithmetic, Dynamically Translated Languages, as well as Transactional Memory.
- Relative **freshness** of RISC-V allows for efficient and innovative approaches while also resulting in **limited resources**.

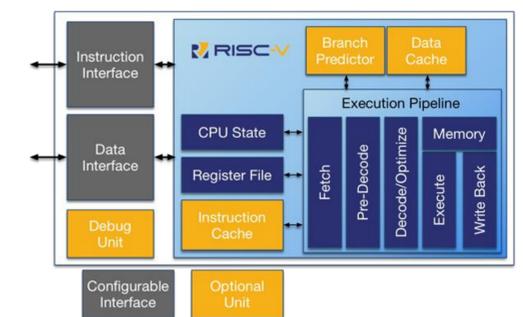
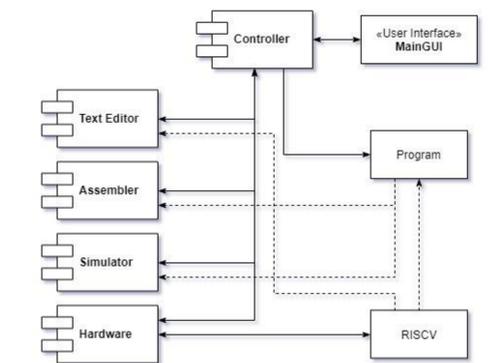
## Acknowledgement

We would like to thank our advisor, Professor Striki, for her continuous insight, guidance, and support. We would also like to thank Professor Burdea for his feedback and inspiration.

## Methodology

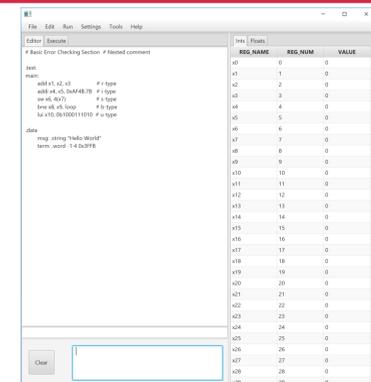


- Determine and configure **standardized development tools** — Eclipse IDE, Oracle JDK, Gluon SceneBuilder — based upon consensus research and need.
- Design and implementation of **core modules** — controller, text editor, assembler, hardware — via collaborated efforts facilitated by **file control** software such as Github and GitKracken.
- Creation of a real-time code **simulator** and implementation of **bases and extensions**, which provide instruction processing of RISC-V programs and ensure comprehensive functionality.
- Continuous integration** of components and packages, concluding with unit, system, and integration tests. The scope of the software necessitated **extensive testing and debugging**.



## Results

- The design and **implementation** of an integrated development environment for RISC-V architecture was **achieved**. It possesses the desired **functionalities** while fulfilling the objectives and motivations.
- The software allows for **user selection** of **hardware** and **core extension preferences**. Options include the hardware choice of 32 or 64 bit based architecture, as well as atomic, multiplication and division, and single or double precision floating point instructions.
- The software **auto completes** instructions and **simulates** RISC-V programs while displaying memory and registers. Algorithmic keywords allow for instruction prediction and the **graphical user interface** provides a natural, visual representation of program logic.
- The software encourages experimentation and **education** of RISC-V by providing an open and **intuitive** platform with additional and accessible academic **resources**.



### RISC-V Instruction Set



## Future Work

- The expansion or implementation of advanced extensions and **further educational features**, such as the creation and embedment of **labs** to introduce users to RISC-V and gain familiarity by solving problems.

## References

[1] "RISC-V ISA." *RISC-V Foundation*, riscv.org/risc-v-isa/. N.p., 2018. Web. 3 April 2018,  
 [2] Waterman, Andrew. Asanovic, Kryste. May 7, 2017. "The RISC-V Instruction Set Manual Volume I: User-Level ISA". Document Version 2.2. EECS Department, University of California, Berkeley.  
 [3] Wasserman, Aaron. "Why We Need RISC-V – Hacker Noon." *Hacker Noon*, Hacker Noon, 4 Jan. 2018, hackernoon.com/why-we-need-risc-v-f94e3929891b. N.p., 2018. Web. 3 April 2018.