

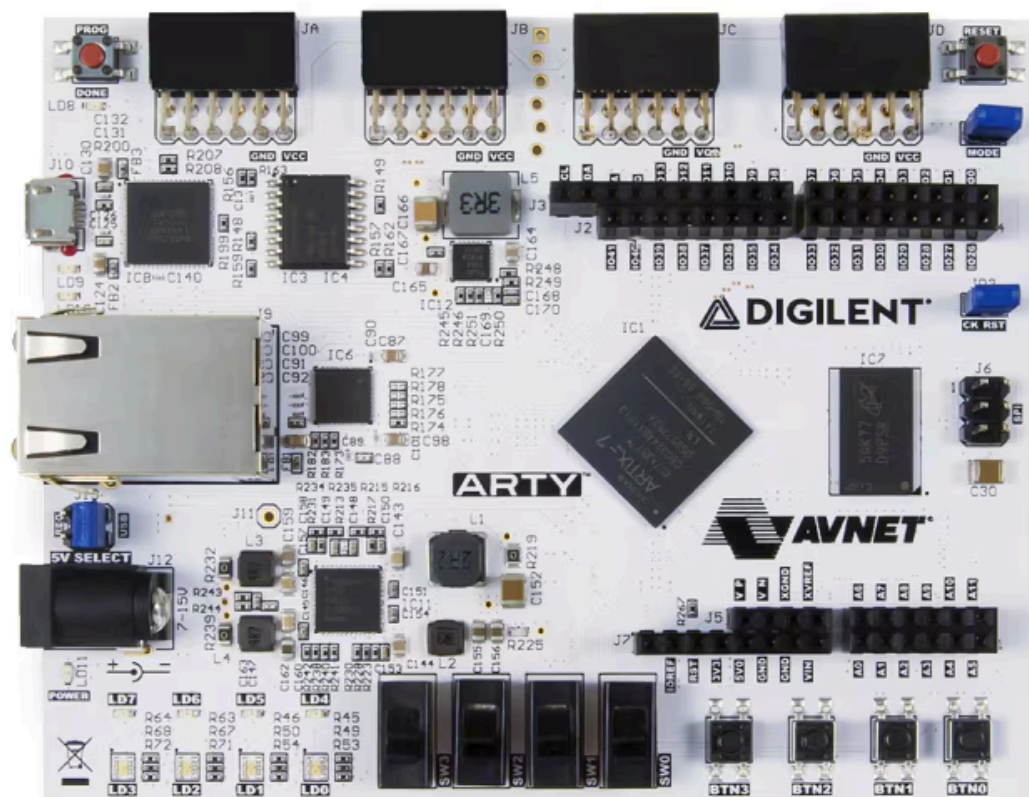
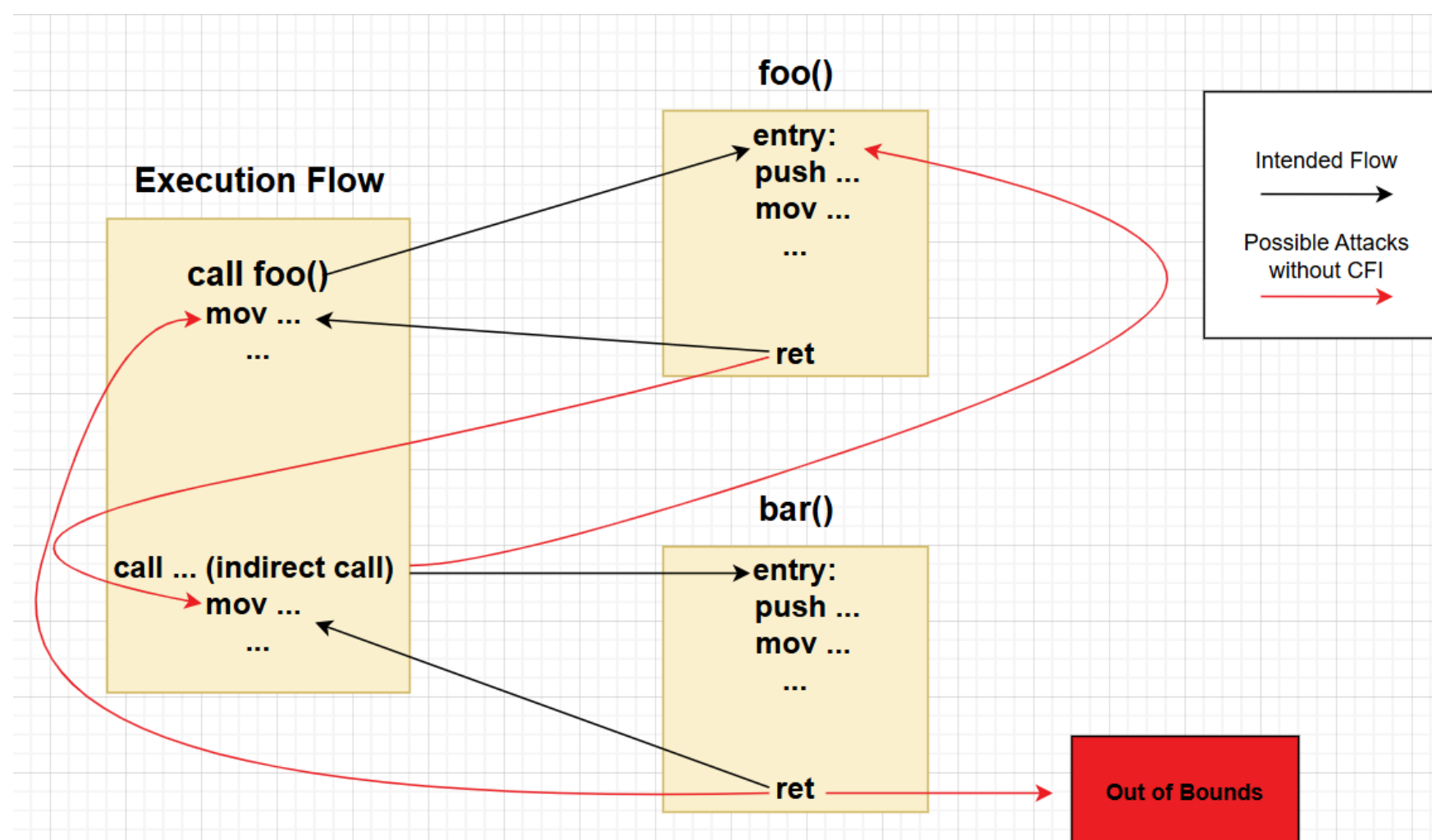
Objective & Background

Develop a RISC-V CFI Extension validation framework to test security against control-flow vulnerabilities (e.g., buffer overflows and control-flow hijacking), enhancing processor resilience for aerospace/defense applications and supporting Idaho Scientific's cybersecurity goals.

Idaho Scientific, a government-focused embedded security leader, develops high-assurance solutions like the Helios Secure Processor. New RISC-V CFI extensions (Zicfiss for shadow stacks/ROP mitigation; Zicfilp for jump validation/JOP defense) lack validation tools, hindering secure adoption in aerospace and defense.

Value Proposition

Developing a RISC-V CFI test suite (zicfiss/zicfilp) via FPGA prototype and software simulation tests to fill verification gaps, prevent control-flow vulnerabilities, and meet aerospace/defense reliability needs.



Key Requirements

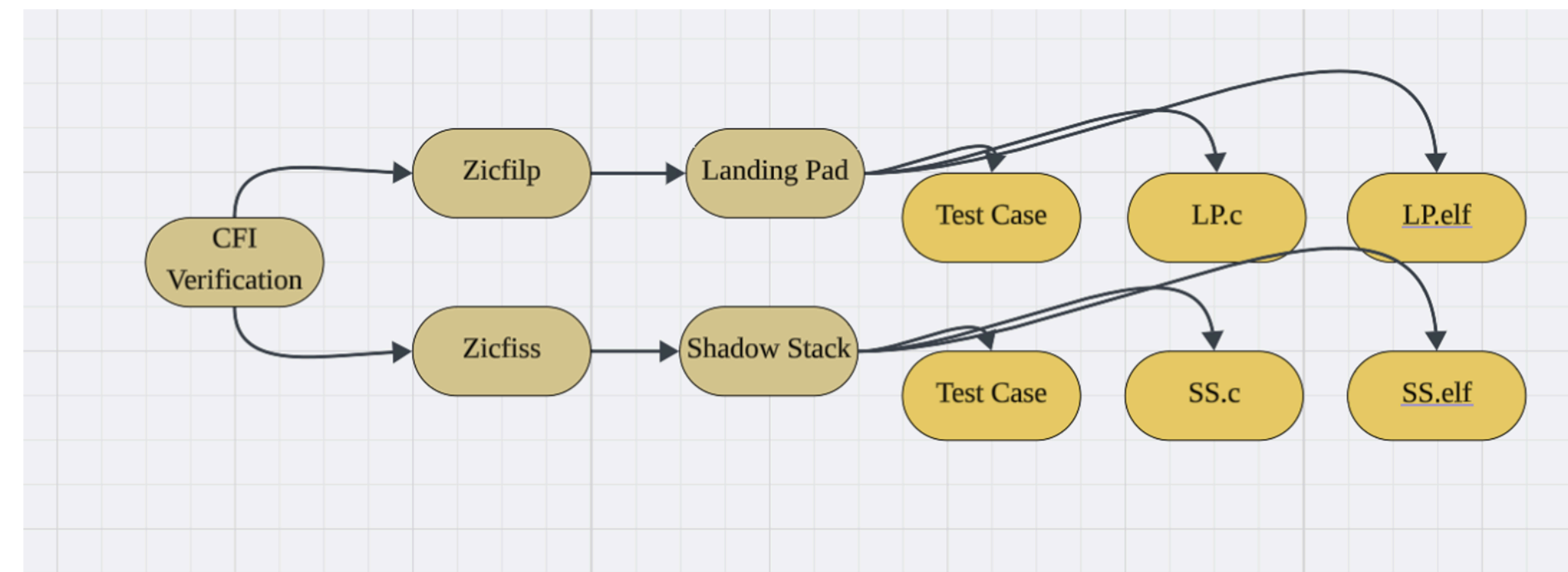
The processor should synthesize, implement, and be placed onto a Arty A7 FPGA. The suite should catch CFI vulnerabilities present in the system. It should have a usable user interface to run tests.

Concept Development

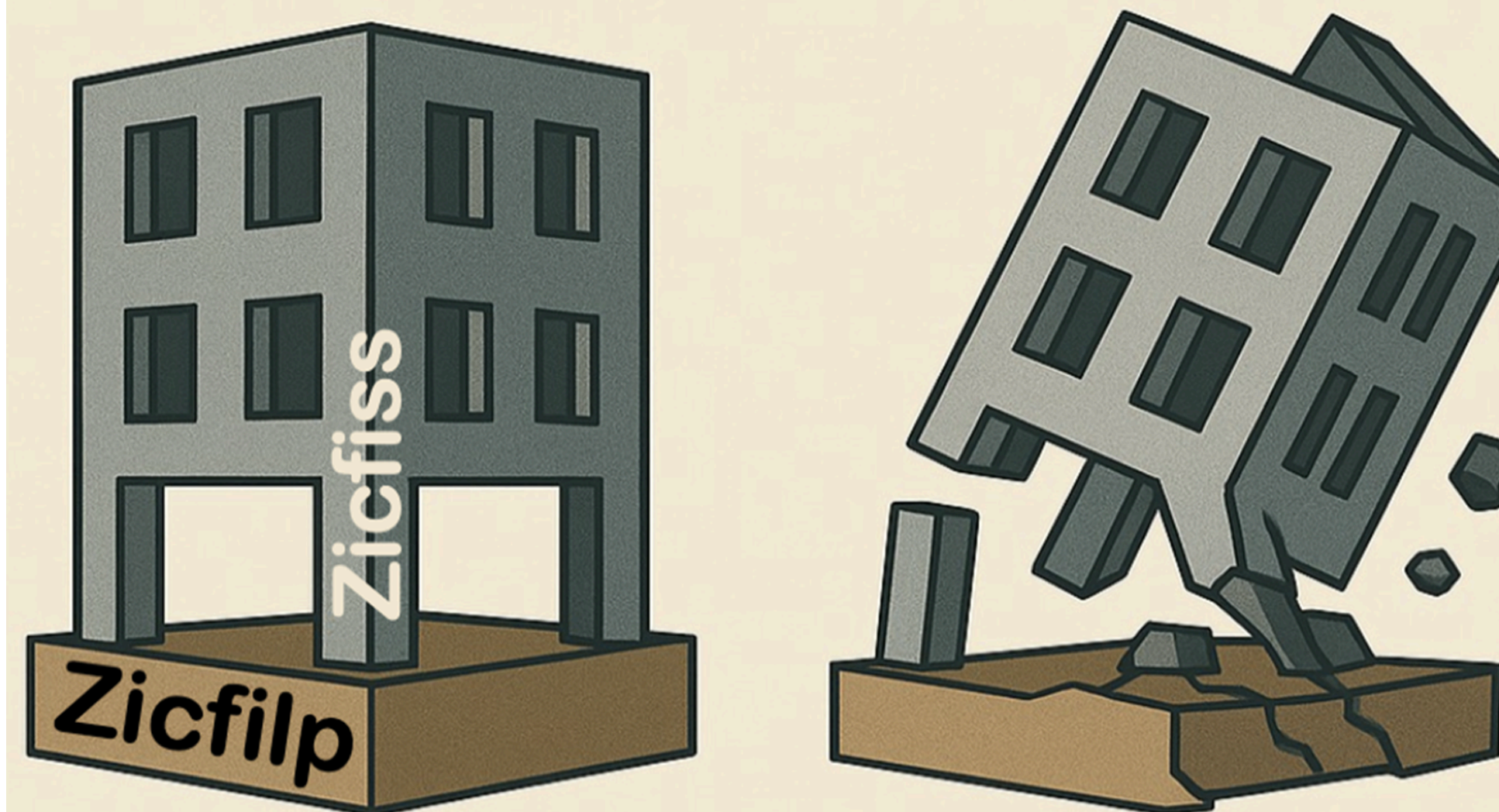
We are placing the pieces of a processor together for implementation onto an FPGA, then developing tests to hold the original designers of the processor accountable to the CFI specification. Developing a simulator to effectively analyze behavior of C and Assembly Code against the processor in an emulated environment.

The Final Design

The final design is to consist of a git repository consisting of a verification suite compromised of zicfiss and zicfilp verification tests. Within these subfolders, there will be tests filled with their respective .elf files, .c files, and documentation itemizing and describing the respective test.



CONTROL FLOW INTEGRITY (CFI)

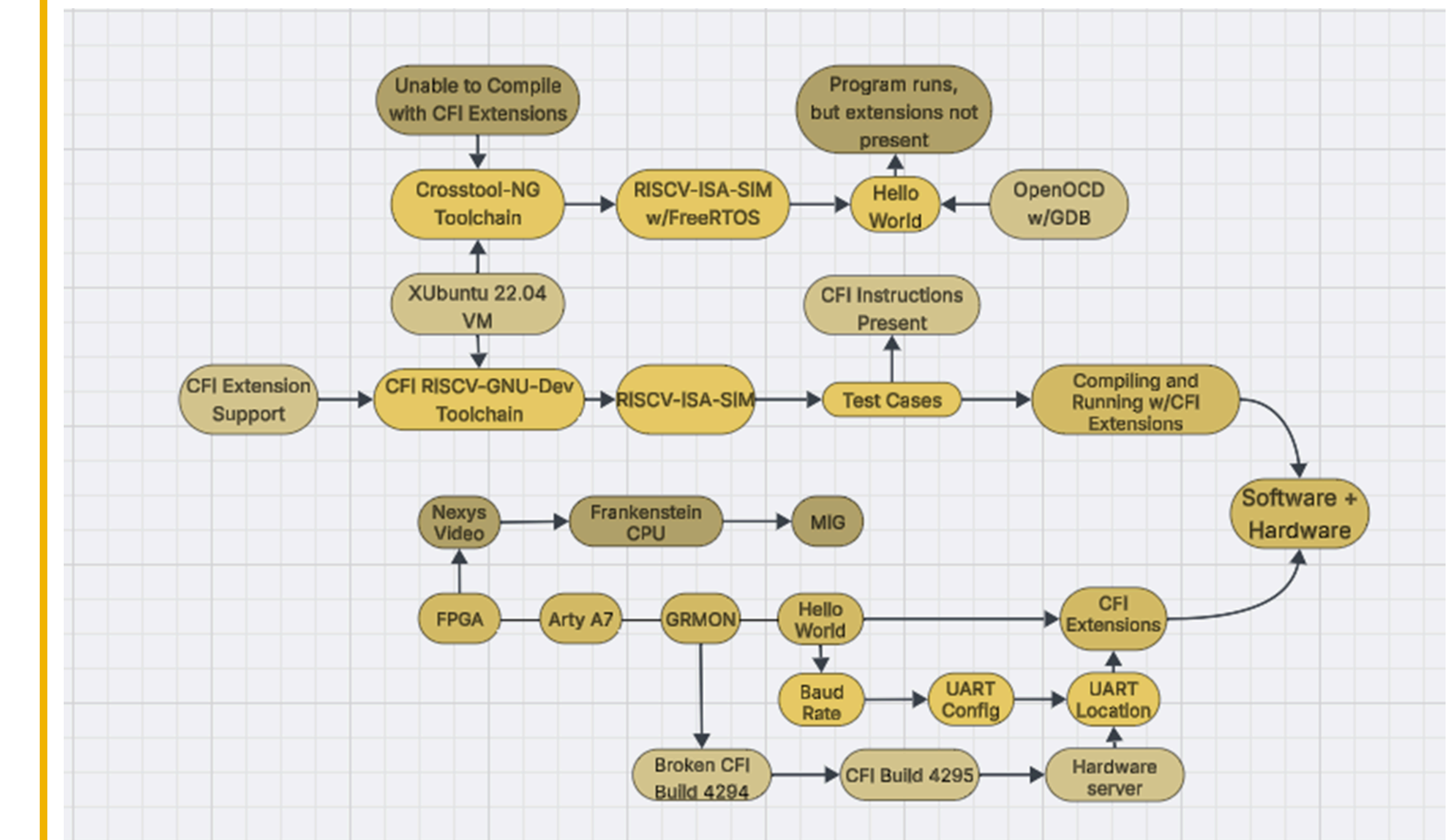


Zicfiss
immovable load-bearing beams
Zicfilp
foundation for beams

Missing Zicfiss
collapses
Missing Zicfilp
no place to stand

Validation & Results

Our project is a verification project intended to validate the functionality of a RISC V CFI specification. If the project verifies the integrity of a CFI core we were successful.



Conclusions & Future Direction

We adapt existing Gaisler RISC-V architectures for hardware compatibility and leverage open-source emulation attacks to stress-test CFI specification resilience. Future work for this project will include more development on the actual implementation of the CFI extensions as well as being able to test future ratifications that come from the RISC-V team. Lastly the project should be made to be open-source so other developers can continue refining the suite for specific needs and different implementations.

Acknowledgements

Kip Sikes and Andrew Weiler's leadership with an unprecedented project and goal. Andrew Weiler and Claude Garrett's expertise in hardware, software, and embedded systems.



2025 Capstone Project