Wally RISC-V Processor

Project Concept

David Harris, James Stine, Sarah Harris, Ross Thompson

OpenHW Technical Working Group Meeting

November 8, 2022

High-Level Summary

Wally is an open-source configurable RISC-V microprocessor and system-on-chip project including SystemVerilog files, test suites, benchmarking, peripherals, Linux boot, and a design flow for an implementation on FPGA boards and for implementation as a System on Chip (SoC) targeting a *28* nm process. This implementation initially targets RV32I, RV32E, and RV64I with a 5-stage pipeline, support for A, C, D, F, and M extensions, and optional caches, branch prediction, virtual memory, AHB, RAMs, and peripherals.

The standard Wally configurations that have been primarily validated include:

RV32E: rv32e tiny core, AHB bus, no core memories, no priv unit, smallest size and highest frequency

RV32I: simple 32-bit CPU, DTIM, IROM, no bus or privileged unit

RV32IC: rv32ic microcontroller with DTIM, IROM, AHB bus, peripherals, MU privilege modes.

RV32GC: RV32GC application processor with caches, branch predictor, FPU, virtual memory, AHB bus, peripherals

RV64I: simple 64-bit CPU, DTIM, IROM, no bus or privileged unit

RV64GC: RV64GC application processor. Like RV32GC, boots BuildRoot Linux on VCU108/118 boards

Summary of Market

There are numerous RISC-V projects available including some wonderful ones from the OpenHW Group Members. Wally is unique in that it is associated with a textbook (D. Harris, J. Stine, S. Harris, R. Thompson, *RISC-V System-on-Chip Design*, to be published by Elsevier, 2023), which both documents its principles of operation and relates it to broader concepts of computer architecture. Moreover, Wally is configurable allowing designs to easily change microarchitecture and compare the costs of various features and implementations.

Since this implementation is targeted at education, it has the possibility of being expanded easily into other research areas that could be beneficial to students learning topics in computer architecture. It could also lead to good research into tradeoffs between functional units or peripherals for a given microarchitecture.

Who would make use of OpenHW output

This design is well-suited to academia for education and research because it is heavily documented. It would also be suitable as an open core for industrial designs. The maximally configured version overlaps substantially with CVA6 in microarchitecture, performance, and application domains.

Why is this project important?

This project is important in that it is associated with a textbook and that it is highly configurable. No existing SystemVerilog RISC-V cores have either of these attributes.

In the maximum configuration, it meets all application processor requirements.

The SystemVerilog code is written considering performance and readability. The floating-point and muldiv units target two cycle latency.

High-Performance Floating-Point Arithmetic

An important element to Wally is its ability to tackle computer arithmetic algorithms that are found in high-performance computer architectures. This includes division by recurrence architectures, signed multiplication, and division and square root as well as addressing the IEEE 754 arithmetic. Wally is configurable to handle both RISC-V handling of floating-point arithmetic (e.g., NaN boxing) as well as configuring for IEEE 754 compliance.

Verification

Wally is designed to be configurable but also integrated into standard testbenches that RISC-V supports. This includes the Imperas RISCV, riscof RISC-V test suite, and IEEE 754 TestFloat environments. Additional tests are also provided for verification of tricky issues with floating-point, branch prediction, privileged unit, virtual memory, and other useful elements.

Benchmarking

Wally runs the Embench and CoreMark benchmarks.

OpenHW Members/Participants

David Harris (HMC), James Stine (OSU), Sarah Harris (UNLV), Ross Thompson (OSU)

We are interested in collaboration with other OpenHW members, including with the Intel Pathfinder environment.

We are also interested in collaboration to adopt the OpenHW verification environment.

Project Leaders

Technical Project Leaders : David Harris, James Stine, Sarah Harris, Ross Thompson

Next Steps/Investigation towards Project Launch

Item1: We would like to integrate the repository into GitHub's OpenHW repository, so that it can get wider acceptance.

Item 2: The boot loader is utilizing a version of QEMU in flash. We are in the process of creating our own boot loader similar to what SweRV does.

Item 3: High-performance Unified FP/Integer division hardware (present implementation uses separate hardware)

Item 4: Docker container for easier deployment.

Item 5: Code beautification

Target Date for Project Launch

January 31, 2023 (or earlier)