Wally RISC-V Processor

Project Launch

OpenHW Technical Working Group Meeting

November 8, 2022 (updated January 25, 2023)

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

Components of the Project

- Component 1: Single-issue 5-Stage Pipeline microarchitecture including Privileged Unit (Machine, User, Supervisor Modes)
- Component 2: Multiply/Divide Unit (MDU) : Unsigned/Signed Multiply, Integer Division using division by recurrence algorithms
- Component 3: Configurable Half/Single/Double/Quad precision IEEE 754 Floating-Point Units (FPU): Fused-Multiply Add/Subtract, Divide/Square Root using division by recurrence. Units are configurable to adhere to IEEE 754 or for RISC-V ISA.
- Component 4: L1 Instruction and Data cache configurable for different ways

- Component 5: Virtual Memory with Memory Management Unit (MMU), Translation Lookaside Buffer (TLB), Page Table Walker, and L2 Cache Integration
- Component 6: Core-Local Interruptor (CLINT), Platform-Level Interrupt Controller (PLIC), 16550D University Asynchronous Receiver/Transmitter (UART), General-Purpose Input/Output (GPIO), Tightly Integrated Memory (TIM), AHB with APB Bridge.
- Component 7: CoreMark and Embench benchmark testing and scripts to run
- Component 8: Verification scripts for riscv-arch-tests, 754 UCB TestFloat, and Custom privileged and peripheral tests.

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 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.

Target Date for Project Launch

January 31, 2023 (or earlier)

Industry Landscape

Wally is most similar to the OpenHW CVA6. Primary differences are that CVA6 has been around longer with greater adoption, while Wally is configurable and associated with a textbook.

OpenHW Members committed to participate

David Harris, James Stine, Sarah Harris, Ross Thompson

Technical Output

Wally explores RISC-V processor design using an open-source configurable pipelined processor affectionately known as Wally. Wally is described in SystemVerilog with a configuration file specifying features such as:

- XLEN (32 or 64)
- Supported Extensions
  - A: Atomic
  - C: Compressed
  - D: Double-precision floating-point
  - E: Embedded 16 registers
  - F: Single-precision floating-point
  - M: Multiplication and division
  - Q: Quad-precision floating-point
  - Zicsr: CSRs
  - Zfencei: FENCE.I instruction synchronization
  - Counters: performance counting
- Privileged Modes and Features
  - Supervisor (S) and User (U) Modes
  - Reset vector address
  - Virtual memory (including instruction and data TLB (ITLB and DTLB) sizes)
  - Physical memory protection
  - Vectored interrupts
- Caches and Branch Prediction
- Supported Peripherals
  - Core Local Interrupt Controller with timers (CLINT)
  - Platform Level Interrupt Controller (PLIC)
  - Universal Asynchronous Receiver/Transmitter serial port (UART)
  - General-Purpose Input/Output pins (GPIO)
  - Tightly Integrated Memory (TIM)

Wally presently does not support:

- Hypervisor
- Advanced Microarchitecture: Superscalar, Out-of-Order, Multithreading, Multicore
- Unfrozen Extensions
  - L: Decimal Floating-Point
  - B: Bit Manipulation
  - J: Dynamically Translated Languages
  - T: Transactional Memory
  - P: Packed SIMD
  - V: Vector Instructions
  - H: Hypervisor

Wally is named in honor of the Harvey Mudd College mascot, Wally Wart. Wally is verified with a variety of test cases.

- The riscv-arch-test suite of tests from the RISC-V foundation show that each instruction in isolation performs its basic functions.
- The wally-riscv-arch-test suite adds tests for the memory management unit and privileged functions not covered by riscv-arch-test.
- The TestFloat suite of tests from Berkeley show that floating-point units pass a comprehensive set of tests.
- Wally boots Linux in simulation and on an FPGA, confirming that it correctly executes 550M instructions including using privileged modes, CSRs, interrupts, peripherals, and virtual memory.

Wally is designed to balance performance, power, and area rather than favor one at great expense to the others. The SystemVerilog has been synthesized in 90 and 28 nm processes. The results were reviewed to detect and eliminate grossly inefficient structures caused by poor coding. The design was also tuned to shorten the cycle time by balancing the amount of logic in each pipeline stage and by optimizing cycle-limiting paths.

We hope Wally will continue to evolve. It could be a good platform for evaluating implementation tradeoffs in RISC-V functional units. For example, we hope it can serve as a baseline against which improved implementations can be compared. Most of the SystemVerilog was written by five people working part time over 18 months, and we are confident that many important optimizations still should be made.

Engineering Resources Required and Availability

The authors are continuing to refine Wally while completing the textbook. No other resources are required for launch.

Wally currently passes riscv-arch-test, UCB Softfloat, and custom tests using the Questa/ModelSim simulators. It also boots Linux on a Xilinx FPGA board. We believe it is currently at or near Technical Readiness Level (TRL) 3 Repository Requirements.

GitHub repository for source code.

Secondary repository for binary Docker container.

Project Distribution Model

Free and open-source with Solderpad Hardware license via OpenHW site.