

# GNU tool chain for CORE-V

*Thursday 23 September 2021 08:45 (30 minutes)*

CORE-V is a family of RISC-V processor cores developed to commercially robust standards by the Open Hardware Group, a consortium of industrial and academic organizations.

In the first part of this talk we give an update on the work on the GNU tool chain for the CV32E40P, the first of the CORE-V family with custom extensions for branching, autoincrement load/store, hardware loops, multiply accumulate and general CPU use. This is a joint effort by Embecosm and the University of Bologna, and has relied on the GVSoC simulator developed as part of the PULP project.

The second part of the talk looks at the use of GVSoC as a GCC tool chain test target. GVSoC is a RISC-V virtual platform, which is a fully open-sourced tool designed to drive future architectural research in the area of highly parallel and heterogeneous RISC-V based IoT platforms. Consisting of a highly configurable event-driven full-platform simulator, GVSoC is capable of performing extremely accurate timing simulations. By reaching 25 MIPS and 100% functional accuracy, the virtual platform supports simulating a broad range of hardware IP blocks, including standalone RISC-V cores, multi-core accelerator Clusters, memories, DMAs, and many other components. While efficient C++ models describe hardware IP blocks and flexible Python scripts instantiate components, a powerful built-in Instruction Set Simulator (ISS) enables simulating complete Parallel Ultra-Low-Power (PULP) systems.

To support the GNU tools test suite targetting CV32E40P core execution, we expanded GVSoC ISS integrating the CORE-V Instruction Set Architecture (ISA) extensions. Along with it, we extended the DejaGnu testing framework, adding a custom baseboard that describes linker and compiler options. Lastly, we relied on a pre-compiled platform-dependent runtime linked by the DejaGnu tool at testing time to enable a faster execution.

A central part of this work is that the tool chain should be upstreamed as a vendor variant, thus riscv32-corev-elf-gcc rather than riscv32-unknown-elf-gcc. We shall conclude this talk by looking at the work remaining before this can be submitted.

A central part of this work is that the tool chain should be upstreamed as a vendor variant, thus riscv32-corev-elf-gcc rather than riscv32-unknown-elf-gcc. In this talk we will look at the work remaining before this can be submitted.

## I agree to abide by the anti-harassment policy

I agree

**Primary authors:** BENNETT, Jeremy (Embecosm); Ms MILLS, Jessica (Embecosm); Prof. TAGLIAVINI, Giuseppe (University of Bologna); Mr BRUSCHI, Nazareno (University of Bologna); TABANELLI, Enrico (University of Bologna)

**Presenters:** BENNETT, Jeremy (Embecosm); Ms MILLS, Jessica (Embecosm); Prof. TAGLIAVINI, Giuseppe (University of Bologna); Mr BRUSCHI, Nazareno (University of Bologna); TABANELLI, Enrico (University of Bologna)

**Session Classification:** GNU Tools Track

**Track Classification:** GNU Tools Track