How Well Does RISC-V Perform?

Recent Comparisons Against Other Architectures

Jeremy Bennett, Craig Blackmore, Paolo Savini (Embecosm)
Hugh O'Keeffe (Ashling)

How to measure compiler performance

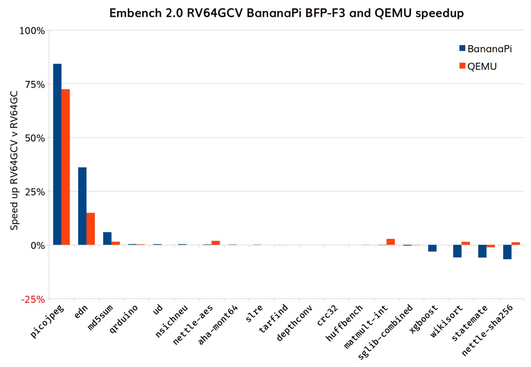
Key benchmark criteria:

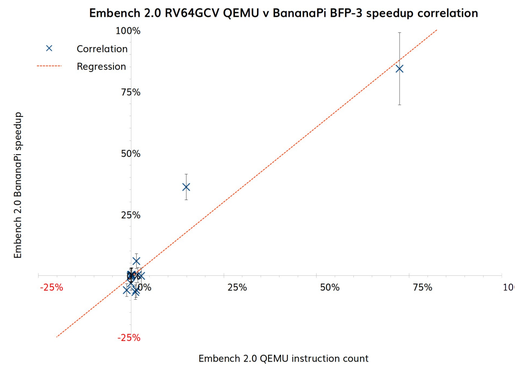
- based on multiple real programs, which are open source, or at least readily available
- able to benchmark code speed and code size
- an easily understood and reproducible score, fairly based on the performance of all programs
- updated regularly, to avoid being gamed by compilers.

This year we focus exclusively on Embench and comparison to Arm, primarily for microcontrollers. There are not yet enough application class RV64 platforms to usefully run SPEC CPU, particularly for RISC-V Vector (RVV) 1.0. QEMU is widely used for this, but we show, using Embench, that, the correlation between QEMU instruction count and real hardware performance with RVV is not close enough to be useful for compiler development.

RISC-V	Arm
CV32E40Pv2 on Nexys A4 FPGA @ 15MHz BananaPi BPI-F3 @1.6GHz	STM32F407 @ 16MHz

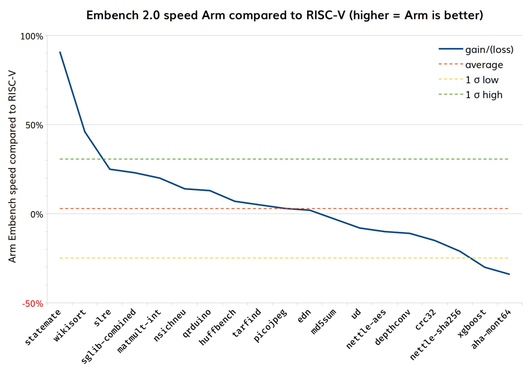
The impact of RISC-V Vector (RVV)

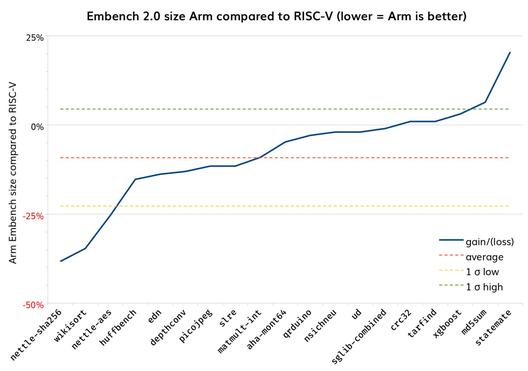




Results obtained using GCC 14.1. Correlation analysis: r = 0.97, σ_r = 6.2%

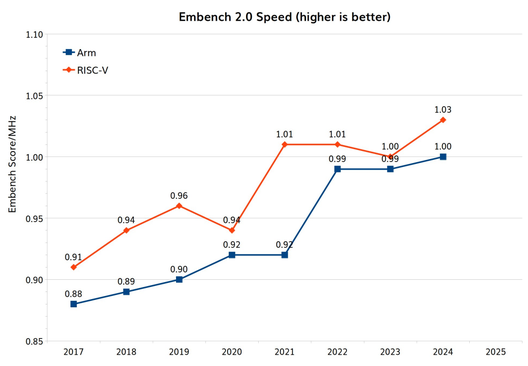
GCC Arm Cortex M4 (no FPU) v RV32IMC

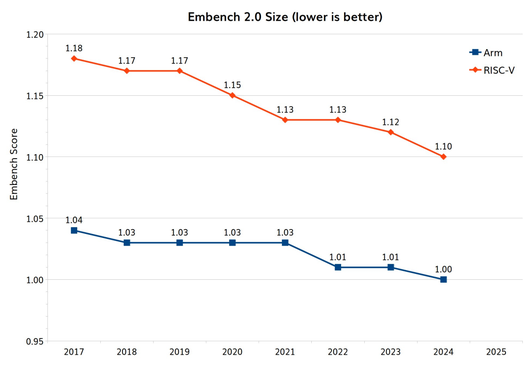




Embench 2.0 scores using GCC 14.1

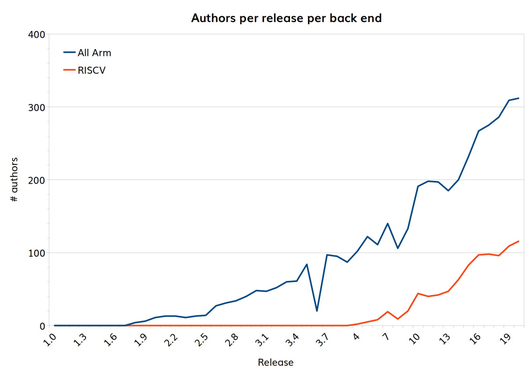
Compiler performance over time: Arm v RISC-V

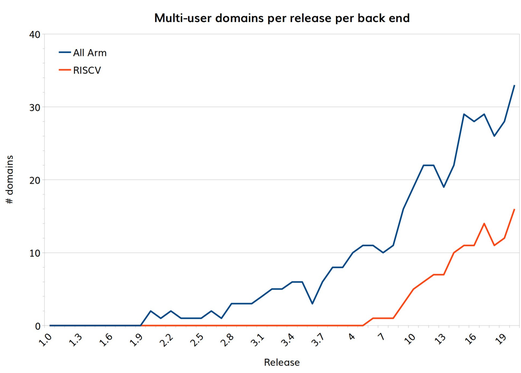




GCC 14.1 Arm Cortex M4 (no FPU) v RV32IMC

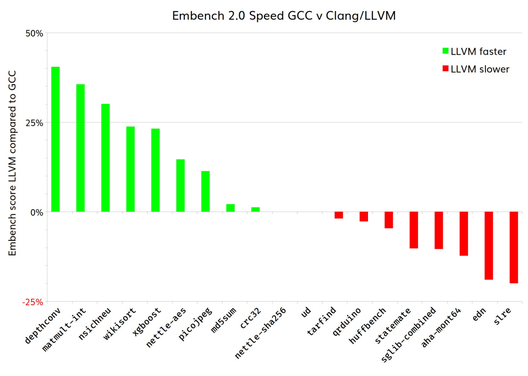
LLVM community activity RISC-V v Arm

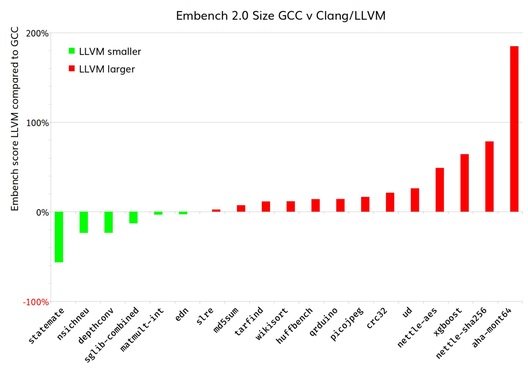




RISC-V includes both RV32 and RV64, Arm includes both Arm and AArch64

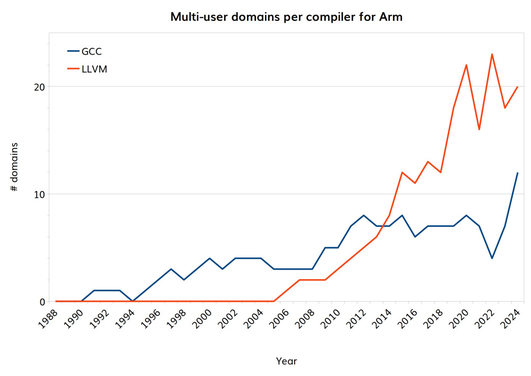
Comparing compilers: GCC v Clang/LLVM

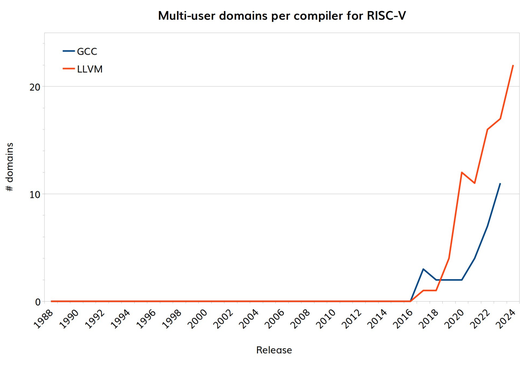




Overall Clang/LLVM is 3.9% faster and 11.7% larger

Community activity GCC v LLVM





RISC-V includes both RV32 and RV64, Arm includes both Arm and AArch64

A huge thank you to our collaborators and supporters: The GCC and LLVM communities, the OpenHW Foundation and the Embench community.

www.embecosm.com **www.ashling.com**