Enabling RISC-V Development with QEMU

Alistair Francis <Alistair.Francis@wdc.com>
RISC-V Workshop - ETH
12th of June 2019
What is QEMU?

It’s FREE!

• QEMU is a very quick open source (mostly GPLv2) emulator

• It is not cycle accurate, but it is functionally accurate

• It uses the Tiny Code Generator (TCG) to translate different guest architecture instructions to host executable code
  – Supports full system (softMMU) emulation
  – Also supports just Linux/BSD user space translation

• It works similarly to GCC with separate host and target support
  – Currently mainline has RISC-V guest and host support
Basics of Tiny Code Generator (TCG)

• TCG began as a backend for a C compiler

• TCG can convert TCG ops to target (host) instructions
  – It also performs some optimisations and liveness analysis to improve performance

• TCG will combine blocks of guest code into a TB blocks
  – The end of a block occurs when a branch/jump instruction is encountered
  – Running QEMU with single step turned on results in every TB block being a single guest instruction

• TCG caches each instruction so that future decoding is extremely fast
  – Fast path memory actions don’t need to have addresses recalculated for example

• TCG natively supports these targets (hosts)
  – AArch64, ARMv7, x86, AMD64, MIPS, PPC, PPC64, RISC-V, S390 and Sparc

• TCG supports even more guest architectures
What do we have in mainline today?

Mainline QEMU has full RISC-V support, don’t use forks

• Support for running 32-bit and 64-bit RISC-V operating systems and Linux user space applications on all supported QEMU platforms

• Support for running 32-bit and 64-bit operating systems and Linux user space applications of all supported guest QEMU architectures on 64-bit RISC-V platforms

• Support for the QEMU virt machine, HiFive Unleased, HiFive One and Spike machines

• ISA extensions can be enabled/disabled via command line (pending pull request)
Getting started with QEMU

• The best way to get started is to follow your distros guide
  • Fedora, Debian, Buildroot and OpenEmbedded all have guides on running on QEMU
  • OpenSBI documentation also describes booting on QEMU
• The QEMU wiki has a RISC-V page
QEMU Demo

Debugging OpenSBI with instruction output
QEMU Demo

Connecting GDB to QEMU and setting break points
QEMU Demo

Enabling RISC-V ISA extensions, Hypervisor (H) extension
QEMU Demo
Running RISC-V Linux User Mode