Toolchain Projects

This page gives an overview of upstream projects. If you miss information or find mistakes, please edit.

ABIs, APIs, and other conventions

The main document for RISC-V ABI/ELF-related information is the RISC-V ELF psABI document, which can be found here: https://github.com/riscv/riscv-elf-psabi-doc/blob/master/riscv-elf.md

An overview of other RISC-V ABI/API related documents can be found here: https://github.com/riscv/riscv-elf-psabi-doc/blob/master/README.md

The following RISC-V ABIs are currently defined:

- 32-bit
  - ILP32, ILP32F, ILP32D, ILP32E
- 64-bit
  - LP64, LP64F, LP64D, LP64Q, LP64E

Calling conventions

There are two calling conventions for GP registers in the RISC-V ecosystem:

- Standard ABI (sometime also called UABI): https://github.com/riscv/riscv-elf-psabi-doc/blob/master/riscv-elf.md#integer-register-convention-
- EABI (targeting embedded systems): https://github.com/riscv/riscv-eabi-spec/blob/master/EABI.adoc

Default library path

The rich set of APIs has the consequence, that glibc's default library path includes a subdirectory for the actual ABI (e.g. "/usr/lib64/lp64d" for LP64D). These default paths are defined here: https://sourceware.org/git/?p=glibc.git;a=blob;f=sysdeps/unix/sysv/linux/riscv/configure.ac;

Binutils

The GNU Binutils are a collection of binary tools (GNU linker, GNU assembler, many other excellent tools such as gprof).

- Binutils Homepage
- Source Release Page

RISC-V maintainers

- Andrew Waterman (SiFive)
- Palmer Dabbelt (Google)
- Jim Wilson (SiFive)
- Nelson Chu (RIVOSINC)

Releases

Rule of thumb: Binutils (GNU linker, GNU assembler, tons of other excellent tools) releases twice per year (mid July and mid January).

- Binutils 2.41 (30 Jul 2023)
- Binutils 2.40 (14 Jan 2023)

RISC-V status

- RV32GC = RV32IMAFDC_Zicsr_Zifencei is implemented
- RV64GC = RV64IMAFDC_Zicsr_Zifencei is implemented
- Unratified extension support is kept in staging branches here: https://github.com/riscv/riscv-binutils-gdb

---

GCC

The GNU Compiler Collection (GCC) includes front ends for C, C++, Objective-C, Fortran, Ada, Go, and D, as well as libraries for these languages (e.g. libstdc).

- GCC Homepage
- Source Release Page

RISC-V maintainers

- Andrew Waterman (SiFive)
- Palmer Dabbelt (RIVOSINC)
- Jim Wilson
- Kito Cheng (SiFive)

GCC RISC-V community meetings

- Schedule: biweekly, Thursday 7 am (PST), starting on Mar 11, 2021
- Organized by Wei Wu (PLCT)
- Public announcement: https://www.mail-archive.com/gcc@gcc.gnu.org/msg94197.html

RISC-V GCC Patchwork Sync meetings

- Schedule: weekly, Tuesday UTC +0 02:30 p.m, starting on 2023-04-18
- Organized by Palmer Dabbelt (RIVOSINC)
- Public announcement: https://docs.google.com/document/d/1bW2jgRmhYdHz7oVw5EUcXVAv4_cfuBaZ5bhVG1pUnIo

Releases

Rule of thumb: GCC closes the merge window for the next release in mid-November (once per year).

- GCC 14 schedule
  - GCC 14 Stage 1 (starts 2024-04-30)
  - GCC 14 Stage 3 (starts 2024-11-16)
  - GCC 14 Stage 4 (starts 2025-01-17)
  - GCC 14.1 release (not defined; possibly May/June)

After stage 3 has started, new functionality may not be introduced.

The upstream release schedule can be found here.

The upstream release timeline can be found here.

RISC-V status

RV32G and RV64G are mostly implemented. However, there is still some optimization potential.

- RV32GC = RV32IMAFDC_Zicsr_Zifencei is implemented

- RV64GC = RV64IMAFDC_Zicsr_Zifencei is implemented
- RV32/64E is supported
- Supported calling conventions: ilp32, ilp32f, ilp32d, lp64, lp64f, lp64d and lp64e
- GCC supports -msave-restore -mno-relax
- Unratified extension support is kept in staging branches here: https://github.com/riscv/riscv-gcc
- RISC-V specific command-line options: https://gcc.gnu.org/onlinedocs/gcc/RISC-V-Options.html

---

GDB

GDB is the GNU Project debugger.

- GDB Homepage
- Source Release Page

RISC-V maintainers

- Andrew Burgess (Embecosm)
- Palmer Dabbelt (Google)

Releases

GDB major releases are approximately annually. There are typically one or two minor releases each year. This is the typical schedule:

- major release branch/pre-release approximately 1 month before release
- first minor release ("re-spin") approximately 3 months after major release

At the time of writing the most recent release was 14.1, released on 2020-12-03.

RISC-V status

Debugging works on top of *PTRACE* syscalls. HW-Breakpoint or HW-Watchpoint support is missing.

Glibc

Glibc is the GNU C library.

- Glibc Homepage
- Source Release Page

RISC-V maintainers

- Palmer Dabbelt (Google)
- Andrew Waterman (SiFive)
- DJ Delorie (Red Hat)
- Darius Rad(Bluespec)

Releases

Rule of thumb: Glibc releases twice per year (February and August).

Previous releases:

- glibc 2.38 (2023-07-31)
- glibc 2.37 (2023-02-01)
- glibc 2.36 (2022-07-30)

RISC-V status

- The following ABIs are supported:
    - ILP32, ILP32D, LP64, LP64D

LLVM

The LLVM Project is a collection of modular and reusable compiler and toolchain technologies.

- LLVM Homepage
- Source Release Page

RISC-V maintainer

- Alex Bradbury (lowRISC)

LLVM RISC-V meetings

- Schedule: biweekly, Thursday 8 am (PST), since Sept 2019
- Organized by Alex Bradbury (lowRISC) and Ana Pazos (Qualcomm)
- Public announcement: https://lists.llvm.org/pipermail/llvm-dev/2021-February/148345.html

Releases

- LLVM 17 schedule
  - LLVM 17.0.6 (28 Nov 2023)

The upstream release page can be found here.

RISC-V status

- RV32GC = RV32IMAFDC_Zicsr_Zifencei is implemented
- RV64GC = RV64IMAFDC_Zicsr_Zifencei is implemented
- RV32E is **not** supported
- Some non-ratified extensions have been merged mainline (they need the flag -menable-experimental-extensions to enable them):
  - Bitmanip v1.0
    - b, zba, zbb, zbc, zbe, zbf, zbm, zbp, zbr, zbs, zbt, zbproposedc
  - Vector v1.0
    - Example command for building: clang --target=riscv64-unknown-elf -march=rv64gcv -menable-experimental-extensions
  - Floating-point v0.1
    - zfh
- LLVM supports -msave-restore -mno-relax
- RISC-V specific command-line options: https://clang.llvm.org/docs/ClangCommandLineReference.html#riscv
- LLVM's extension parsing code: https://github.com/llvm/llvm-project/blob/main/clang/lib/Driver/ToolChains/Arch/RISCV.cpp

---

Newlib

Newlib is a C standard library implementation intended for use on embedded systems.

- Newlib Homepage
- Source Release Page

RISC-V maintainer

- Kito Cheng (SiFive)

Releases

Rule of thumb: Newlib releases once per year.

Last releases:

- Newlib 4.3.0 (20 Jan 2023)

RISC-V status

RV32 and RV64 are supported. Still, there is optimization and completeness potential.