Open Source Compiler Tool Chains and Operating Systems for RISC-V

Jeremy Bennett
Mark Corbin
Agenda

• Open source compilers: Jeremy Bennett
  - what is available?
  - how good are they?
Agenda

• Open source compilers: Jeremy Bennett
  - what is available?
  - how good are they?

• Open source operating systems: Mark Corbin
  - what is available?
  - how good are they?
Agenda

• Open source compilers: Jeremy Bennett
  - what is available?
  - how good are they?

• Open source operating systems: Mark Corbin
  - what is available?
  - how good are they?

• Questions: Everyone
Tool Chain Components (GNU)

GCC Tool Chain Components (GNU)

- GCC
- GAS
- binutils
- objdump
- GDB

Source code

Ada
C/C++
Fortran
OpenMP

RV32GC
RV64GC

Compiler libraries

libc/libm

Newlib
Glibc
Musl

Complete
In progress
Not available

RV32GC
RV64GC

Glibc
Musl

libgcc
libstdc++v3

objdump

disassemble
target sim
Tool Chain Components (Clang/LLVM)

Source code
- Ada
- C/C++
- Fortran
- OpenMP

Clang/LLVM

LLVM int. asm

Object code

Compiler libraries
- CompilerRT
- libc++

libc/libm

LLDB

TableGen

binutils

llvm-objdump

RV32GC
RV64GC

Newlib
- BSD libc
- Musl
Tool Chain Components (Clang/LLVM)

- **Source code**: Ada, C/C++, Fortran, OpenMP
- **Clang/LLVM**
- **LLVM int. asm**
- **TableGen**
- **binutils**
- **compiler libraries**: CompilerRT
- **objc/libm**: BSD libc, Musl
- **Object code**: Newlib
- **LLDB**: ?

- **RV32GC**, **RV64GC**
- **Complete**, **In progress**, **Not available**
Tool Chain Components (Plan 9)

- Source code
- Compiler
- Assembler
- linker
- Object code
- compiler libraries
- emulation
- libc/libm
- libc
- disasm
- RV32IMA
- C99
Tool Chain Components (Plan 9)

- RV32IMA
- Source code: C99
- Compiler
- Assembler
- linker
- Object code
- Complete
- In progress
- Not available

Tool Chain Components (Plan 9)

- **Source code**
  - Compiler
    - Assembler
      - linker
        - Object code
          - libc/libm
            - emulation
  - RV32IMA

- **Complete**
- **In progress**
- **Not available**

- Glenda

### GCC Quality (C/C++)

<table>
<thead>
<tr>
<th></th>
<th>LIT</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>32-bit RISC-V</td>
<td>64-bit RISC-V</td>
</tr>
<tr>
<td><strong>LIT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expected passes</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Expected failures</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Unsupported tests</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Unexpected failures</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td><strong>GNU</strong></td>
<td></td>
<td>92,378</td>
<td>93,458</td>
</tr>
<tr>
<td>Expected passes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unexpected failures</td>
<td></td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>Unexpected successes</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Expected failures</td>
<td></td>
<td></td>
<td>209</td>
</tr>
<tr>
<td>Unresolved tests</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Unsupported tests</td>
<td></td>
<td></td>
<td>2,468</td>
</tr>
<tr>
<td></td>
<td>LIT</td>
<td>GNU</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32-bit RISC-V</td>
<td>64-bit RISC-V</td>
<td></td>
</tr>
<tr>
<td>LIT</td>
<td>Expected passes</td>
<td>46,135</td>
<td>46,136</td>
</tr>
<tr>
<td></td>
<td>Expected failures</td>
<td>171</td>
<td>171</td>
</tr>
<tr>
<td></td>
<td>Unsupported tests</td>
<td>427</td>
<td>427</td>
</tr>
<tr>
<td></td>
<td>Unexpected failures</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>GNU</td>
<td>Expected passes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unexpected failures</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unexpected successes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Expected failures</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unsupported tests</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unresolved tests</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unsupported tests</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Clang/LLVM Quality (C/C++)

<table>
<thead>
<tr>
<th></th>
<th>LIT</th>
<th>GNU</th>
<th>32-bit RISC-V</th>
<th>64-bit RISC-V</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Expected passes</td>
<td>Expected passes</td>
<td>46,135</td>
<td>46,136</td>
</tr>
<tr>
<td></td>
<td>Expected failures</td>
<td>Expected failures</td>
<td>171</td>
<td>171</td>
</tr>
<tr>
<td></td>
<td>Unsupported tests</td>
<td>Unsupported tests</td>
<td>427</td>
<td>427</td>
</tr>
<tr>
<td></td>
<td>Unexpected failures</td>
<td>Unexpected failures</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>GNU</td>
<td>GNU</td>
<td>28,579</td>
<td>28,568</td>
</tr>
<tr>
<td></td>
<td>Expected passes</td>
<td>Expected passes</td>
<td>28,579</td>
<td>28,568</td>
</tr>
<tr>
<td></td>
<td>Unexpected failures</td>
<td>Unexpected failures</td>
<td>4,863</td>
<td>4,949</td>
</tr>
<tr>
<td></td>
<td>Unexpected successes</td>
<td>Unexpected successes</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Expected failures</td>
<td>Expected failures</td>
<td>51</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>Unresolved tests</td>
<td>Unresolved tests</td>
<td>1,851</td>
<td>1,861</td>
</tr>
<tr>
<td></td>
<td>Unsupported tests</td>
<td>Unsupported tests</td>
<td>3,737</td>
<td>3,700</td>
</tr>
</tbody>
</table>
Compiled Code Speed (RV32)

- GCC Baseline
- LLVM
- GCC EABI (old)
- GCC LTO
- GCC new size opts

Embench Score

Large is good
Compiled Code Size (RV32)

- ARM GCC
- GCC comb. elim.
- GCC new size opts
- LLVM save-restore
- GCC LTO
- LLVM EABI (new)
- GCC EABI (old)
- LLVM
- GCC Baseline

**Small** is good
Operating Systems

Mark Corbin
Debian

- Upstream 64-bit support as a ‘ports architecture'
- Platforms - QEMU. HiFive Unleashed and LowRISC (with custom kernels).
- 88% of the Debian package archive builds for RISC-V
  - https://buildd.debian.org/stats/
- On-going work – change boot flow from BBL → U-Boot/OpenSBI
- Future plans – proper Debian Installer support, work towards 100% archive coverage (requires LLVM and Rust support), look at RV32 support
- Further details - https://wiki.debian.org/RISC-V

Information kindly provided by
Karsten Merker and Manuel A. Fernandez Montecelo
Fedora

- Upstream 64-bit support as a ‘alternative architecture’ (no 32-bit support planned)
- Platforms - QEMU and HiFive Unleashed
- Bootloader flow still in development – BBL → U-Boot/OpenSBI
- Further details - https://fedoraproject.org/wiki/Architectures/RISC-V

Information kindly provided by Richard W.M. Jones
Gentoo

- Upstream experimental 64-bit support
- Platforms - QEMU userspace only – no bootable systems yet
- Default ‘system set’ is complete
- Outstanding tasks – test more packages, automate build and update of installation stages/media

Information kindly provided by
Andreas K. Huettel
openSUSE

- Upstream experimental 64-bit support
- Platforms - QEMU userspace only
- Build of openSUSE Factory, but missing some packages including LLVM and Rust.
- Outstanding tasks – add grub support to standardise the installation and boot process
- Further details - https://en.opensuse.org/openSUSE:RISC-V

Information kindly provided by
Andreas Schwab
FreeBSD

- Upstream 64-bit support
- Platforms - QEMU and HiFive Unleashed
- HiFive Unleashed
  - Ethernet and SMP
  - No SD card support yet (ramdisk or NFS root)
  - See https://www.youtube.com/watch?v=nZsBp0Gbg_8
- Future plans - TODO list on Wiki page?
- Further details - https://wiki.freebsd.org/riscv

HiFive Unleashed details from mailing list post by Ruslan Bukin
NetBSD

• 64-bit support (not upstream)
• Platforms – Spike (not fully booting)
• Future plans – bootable system, userland, QEMU and HiFive Unleashed support
• Further details - http://wiki.netbsd.org/ports/riscv/

Information kindly provided by
Zachary McGrew and Maxime Villard
Buildroot

- Upstream 32-bit and 64-bit support
- Platforms
  - QEMU
  - Coming soon ... HiFive Unleashed (patches submitted)
- Most packages build - http://autobuild.buildroot.net/
- On-going work – maintain and remove forks
- Future plans – musl/uclibc support, additional board support, migrate to U-Boot/OpenSBI
- Further details - https://buildroot.org/
OpenEmbedded/Yocto

- Upstream 32-bit and 64-bit support in OE-Core layer
- Platforms - QEMU and HiFive Unleashed
- All standard packages work for 64-bit (and most work on 32-bit)
- Future plans – maintain and remove forks, add qemu-riscv32/64 machines to OE-Core
- Further details - https://layers.openembedded.org/layerindex/branch/master/layer/meta-riscv/

Information kindly provided by Alistair Francis
OpenWrt

- Upstream 64-bit support (in staging tree)
- Platforms - QEMU and HiFive Unleashed
- Supports musl (default C library for OpenWrt)
- Builds 99% of the OpenWrt package repo
- Further details -
  https://openwrt.org/docs/techref/hardware/soc/soc.sifive

Information kindly provided by
Alex Guo and Zoltan Herpai
Apache MyNewt

• Platforms - HiFive1
• Further details - https://mynewt.apache.org/
FreeRTOS

• Upstream 32-bit and 64-bit support

• Platforms – QEMU (sifive_e), Microsemi M2GL025 Mi-V and OpenISA Vega Board

• Future plans
  – will be adding support for IAR compilers
  – new features driven by user requests
    • Maybe ...floating point? ...memory protection? ...

• Further details - https://www.freertos.org/Using-FreeRTOS-on-RISC-V.html

Information kindly provided by
Richard Barry
Zephyr

• Upstream 32-bit support

• Platforms
  - QEMU (sifive_e), HiFive1, Microsemi M2GL025 Mi-V and OpenISA Vega Board
  - Coming soon...HiFive1 Rev B and LiteX VexRiscv

• Future plans – SMP support, RV64 support, memory protection.

• Further details - https://www.zephyrproject.org/

Information kindly provided by
Nathaniel Graff
Plan 9

• Plan 9 C compiler has been re-targeted for RV32 (see https://www.geeklan.co.uk/files/oshug69-Miller-criscv.pdf)
• Compiler has been tested with a small subset of libc on a bare metal picoRV32 core
• Work not started on kernel

Information kindly provided by Richard Miller
Summary

• Common Issues
  - Waiting for mainline kernel to catch-up for HiFive Unleashed support
  - Need upstream 32-bit glibc for 32-bit Linux support
  - LLVM/Rust support needed for a number of package builds
  - Bootloader flow changing from BBL → U-Boot/OpenSBI
  - Broader range of (low cost) development platforms needed (especially Linux-capable 32-bit platform).

• Please let me have any changes or comments for updating:
  - https://github.com/riscv/riscv-wiki/wiki/RISC-V-Software-Status