SiFive

RISC-V

Palmer Dabbelt, SiFive

Why Instruction Set Architecture matters

- Why can't Intel sell mobile chips?
  - 99%+ of mobile phones/tablets are based on ARM's v7/v8 ISA
- Why can't ARM partners sell servers?
  - 99%+ of laptops/desktops/servers are based on the AMD64 ISA (over 95%+ built by Intel)
- How can IBM still sell mainframes?
  - IBM 360 is the oldest surviving ISA (50+ years)

ISA is the most important interface in a computer system

ISA is where software meets hardware

Open Software/Standards Work!

Field	*Standard*	*Free, Open Impl.*	*Proprietary Impl.*
Networking	Ethernet, TCP/IP	Many	Many
OS	POSIX	Linux, FreeBSD	M/S Windows
Compilers	C	gcc, LLVM	Intel icc, ARMcc
Databases	SQL	MySQL, PostgresSQL	Oracle 12C, M/S DB2
Graphics	OpenGL	Mesa3D	M/S DirectX
ISA	??????	-----------	x86, ARM, IBM360

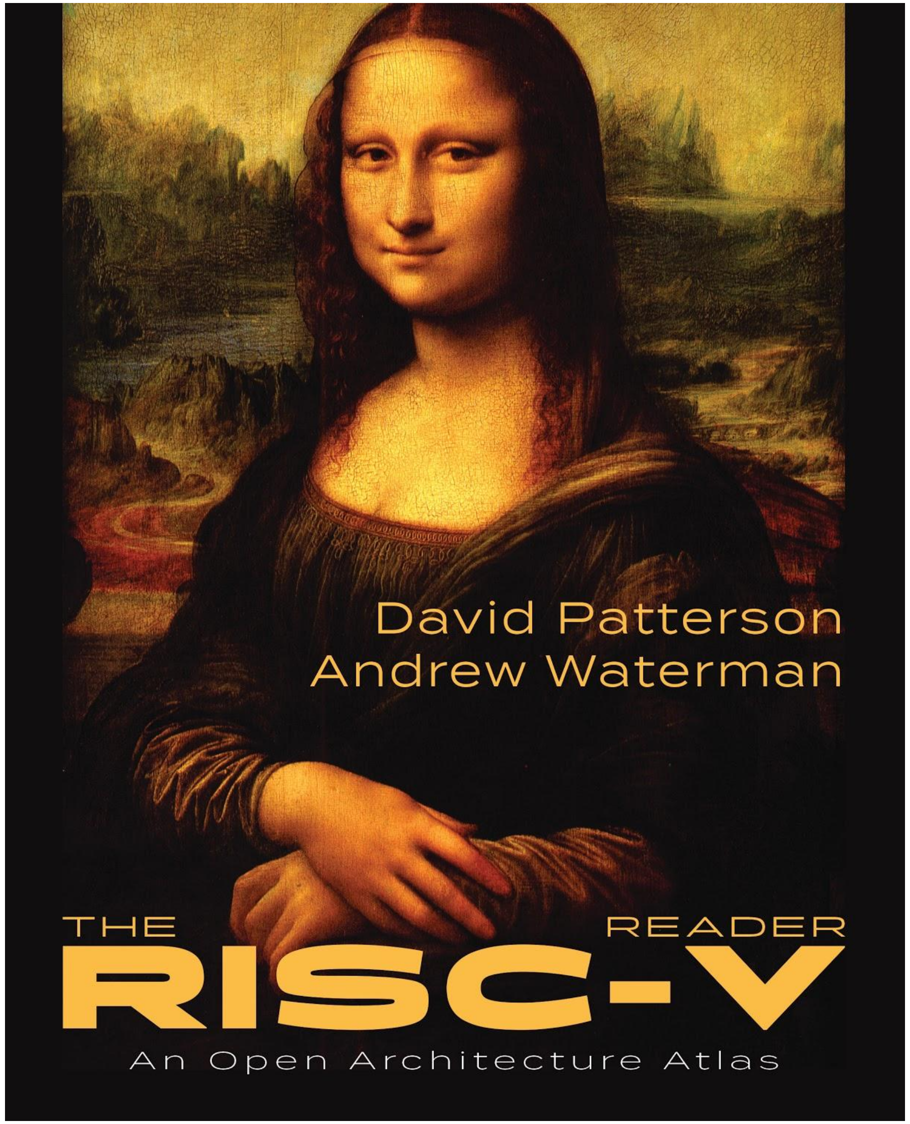
- Why not have successful free & open standards and free & open implementations, like other fields?
- Dominant proprietary ISAs are not great designs

What is RISC-V?

- A high-quality, license-free, royalty-free RISC ISA specification originally designed at UC Berkeley
- Standard maintained by the non-profit RISC-V Foundation
- Suitable for all types of computing system, from microcontrollers to supercomputers
- Numerous proprietary and open-source cores
- Experiencing rapid uptake in industry and academia
- Supported by a growing shared software ecosystem
- A work in progress…

RISC-V Reader

- Authored by Andrew and Dave
  - Andrew Waterman: SiFive co-founder and co-inventor of the RISC-V ISA
  - Dave Patterson: UC Berkeley professor, co-author of “Computer Organization and Design”, and co-inventor of RISC-V
- “An Open Architecture Atlas”
  - Concise introduction and reference
  - Aimed at embedded systems programmers, students, and the curious



SiFive

Origin of RISC-V

RISC-V Origins

- In 2010, after many years and many projects using MIPS, SPARC, and x86 as the bases of research at Berkeley, it was time to choose an ISA for next set of projects
- Obvious choices: x86 and ARM

Intel x86 “AAA” Instruction

- ASCII Adjust After Addition
- AL register is default source and destination

- If the low nibble is > 9 decimal, or the auxiliary carry flag AF = 1, then
  - Add 6 to low nibble of AL and discard overflow
  - Increment high byte of AL
  - Set CF and AF
- Else
  - CF = AF = 0

- Single byte instruction

ARM v7 LDMIAEQ Instruction

```
LDMIAEQ SP!, {R4-R7, PC}
```

- LoaD Multiple, Increment-Address
- Writes to 7 registers from 6 loads
- Only executes if EQ condition code is set
- Writes to the PC (a conditional branch)
- Can change instruction sets

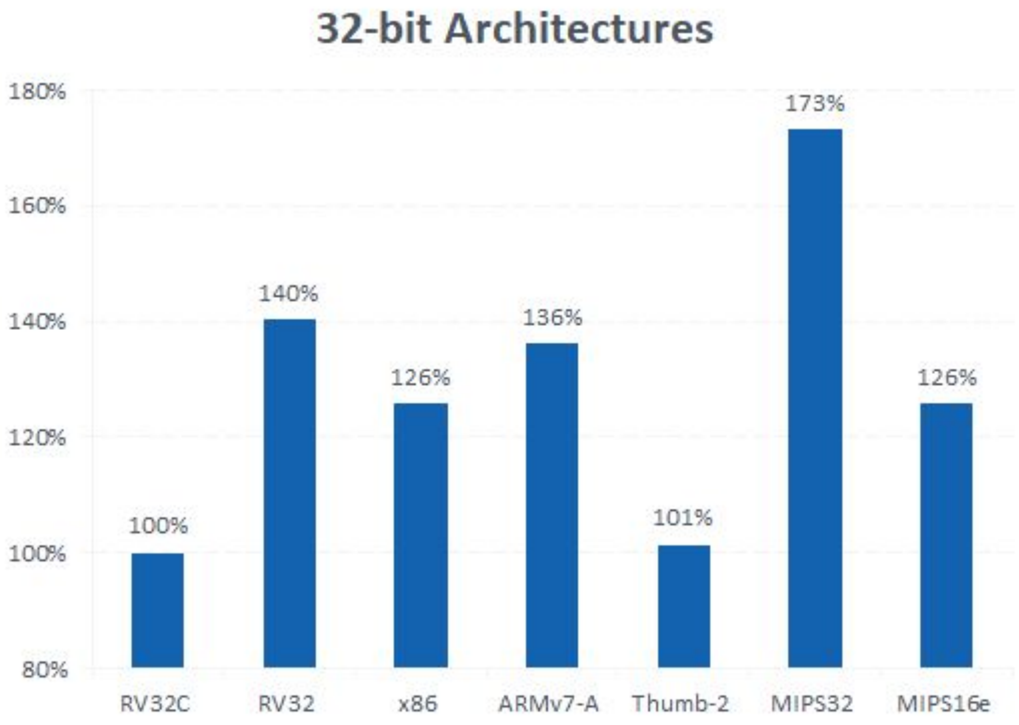
- Idiom for "stack pop and return from a function call"

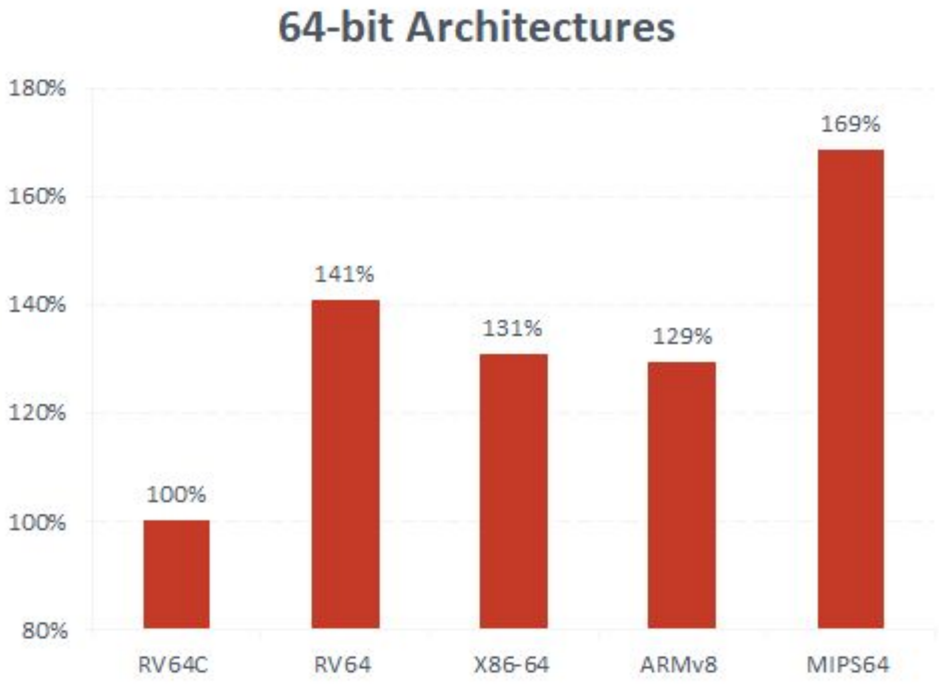
RISC-V Origin Story

- x86 impossible – IP issues, too complex
- ARM mostly impossible – no 64-bit, IP issues, complex
- So we started "3-month project" in summer 2010 to develop our own clean-slate ISA
  - Principal designers: Andrew Waterman, Yunsup Lee, Dave Patterson, Krste Asanovic
- Four years later, we released the frozen base user spec
  - First public specification released in May 2011
  - Several publications, many tapeouts, lots of software along the way

Static Code Size

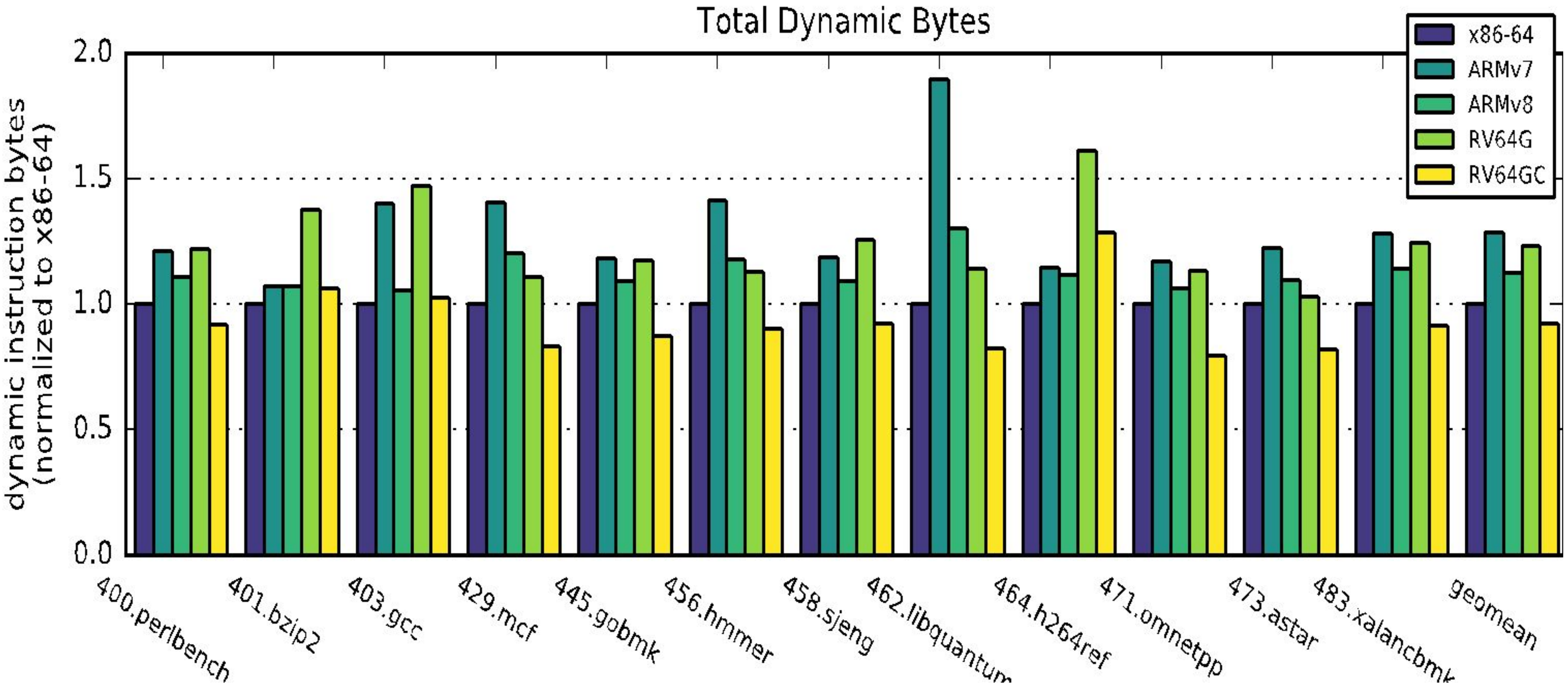
- RISC-V is now the smallest ISA for 32- and 64-bit addresses
- All results are with the same GCC compiler and options



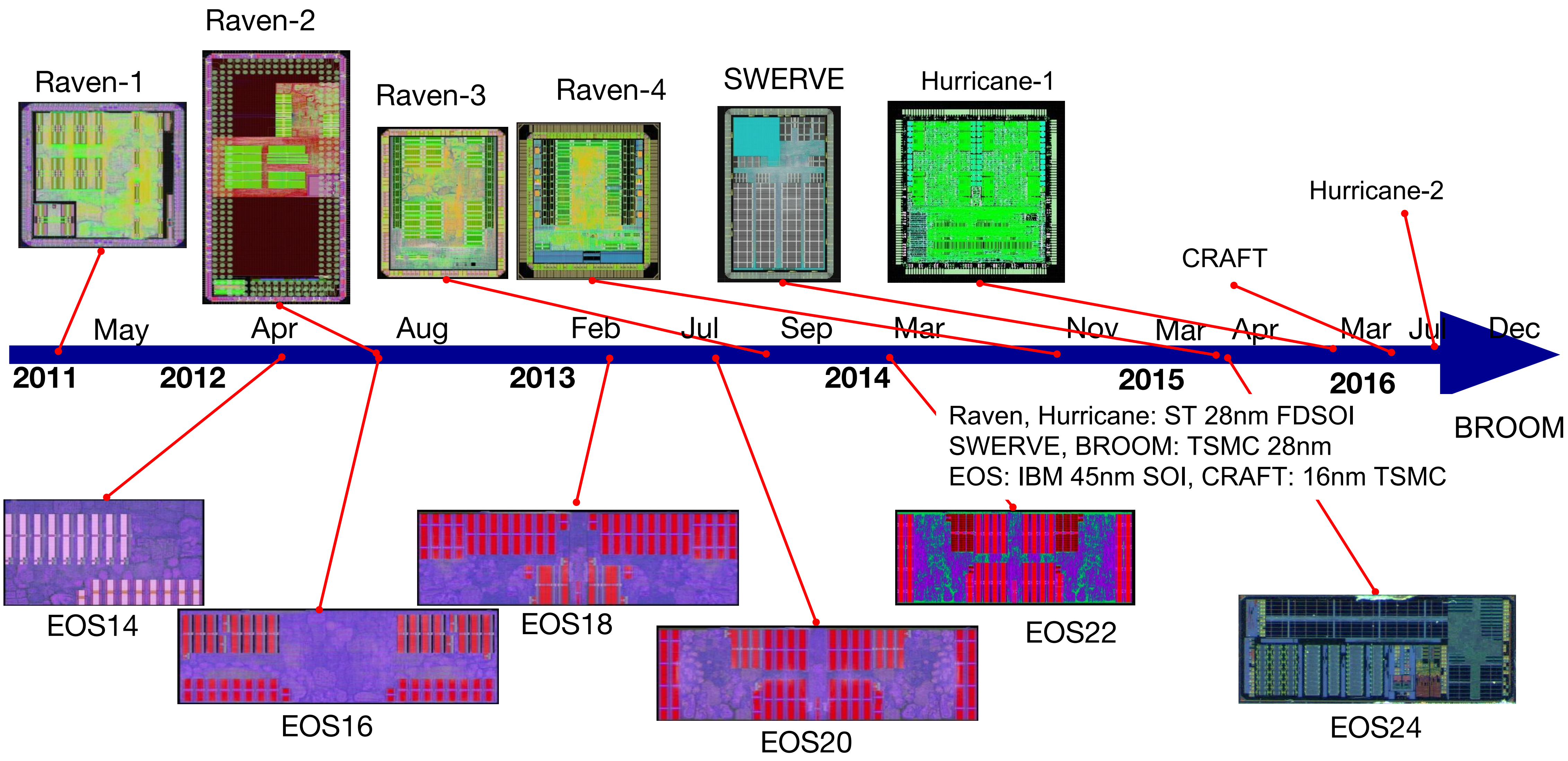


Dynamic Bytes Fetched

- RV64GC is lowest overall in dynamic bytes fetched

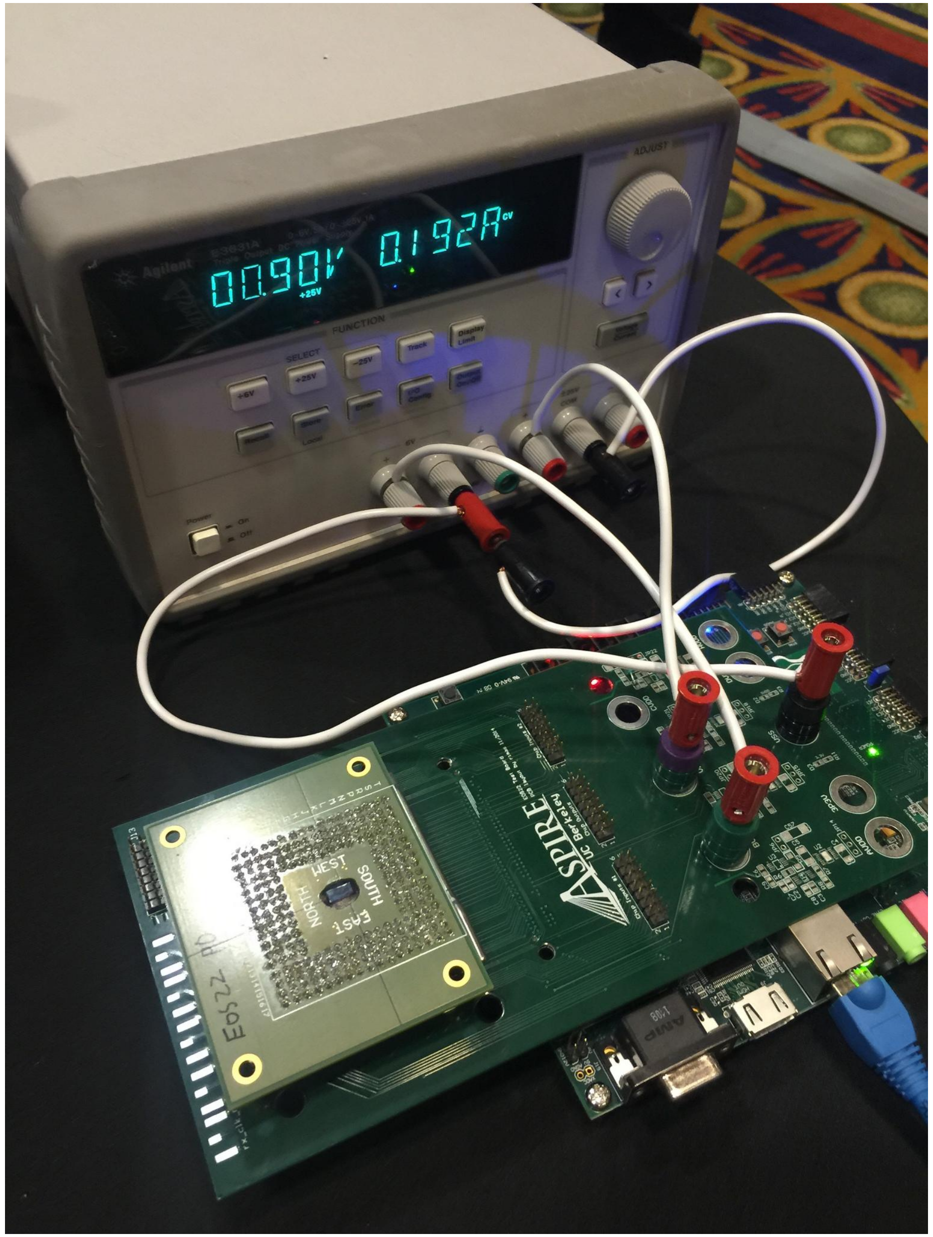


Silicon Implementations



Raven, Hurricane: ST 28nm FDSOI
SWERVE, BROOM: TSMC 28nm
EOS: IBM 45nm SOI, CRAFT: 16nm TSMC

DIY Chip Tapeout Effort



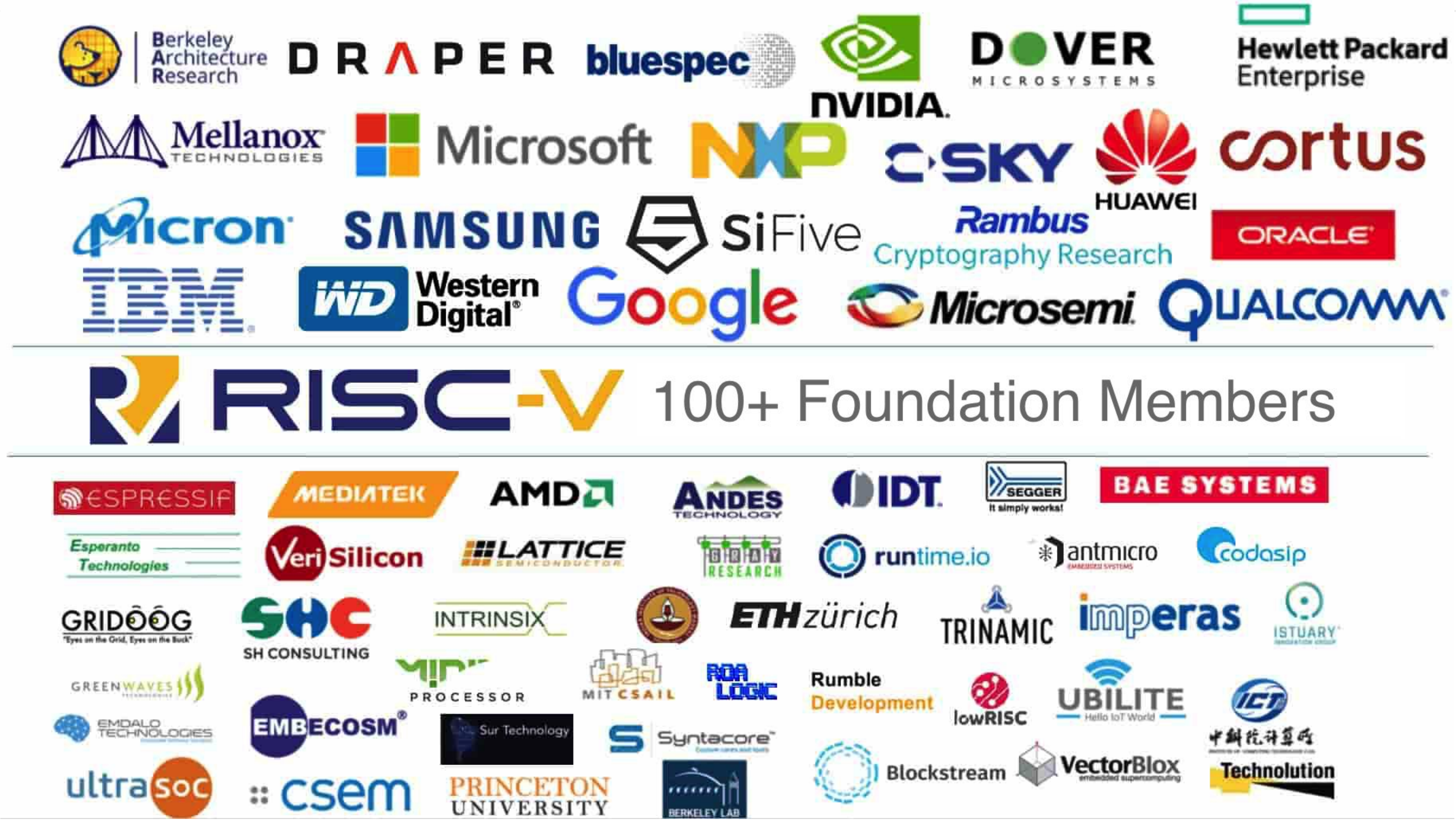
RISC-V Software

There’s a lot of software!

```
From: Palmer Dabbelt <palmer.dabbelt@eecs.berkeley.edu>
To: config-patches@gnu.org
Subject: config.sub patch for RISC-V
Date: Wed, 10 Sep 2014 19:20:31 -0700
Message-Id: <1410402032-9184-1-git-send-email-palmer.dabbelt@eecs.berkeley.edu>
X-Mailer: git-send-email 1.8.5.5

This patch provides support for the RISC-V ISA: http://riscv.org/

Not yet upstreamed ports of the binutils, GCC, LLVM, glibc, and Linux
exist for RISC-V, and a number of hardware implementations exist --
more more information can be seen at http://riscv.org .  We'd like to
start getting RISC-V recognized by configure so it's easier for people
to start porting stuff.

Thanks!
```

Current State of RISC-V

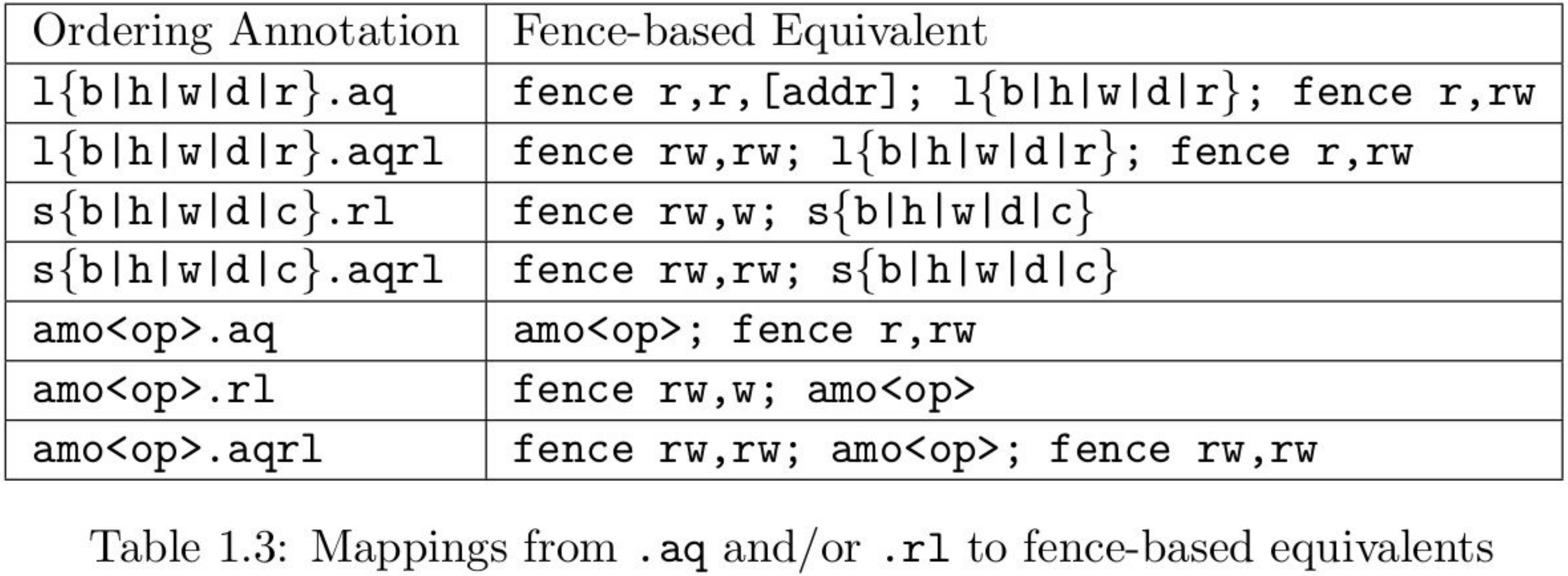
RISC-V Foundation



RISC-V Specifications

- User Mode ISA Specification
  - RV32I/RV64I: ALU, branches, and memory
  - M extension for multiplication
  - A extension for atomics
  - F and D extensions for single and double precision floating-point
  - C extension for compressed instructions (16-bit)
- Privileged Mode ISA Specification
  - Supervisor mode
  - Hypervisor mode
  - Machine mode
- External Debug Specification
  - Debug machine-mode software over JTAG

RISC-V Weak Memory Model

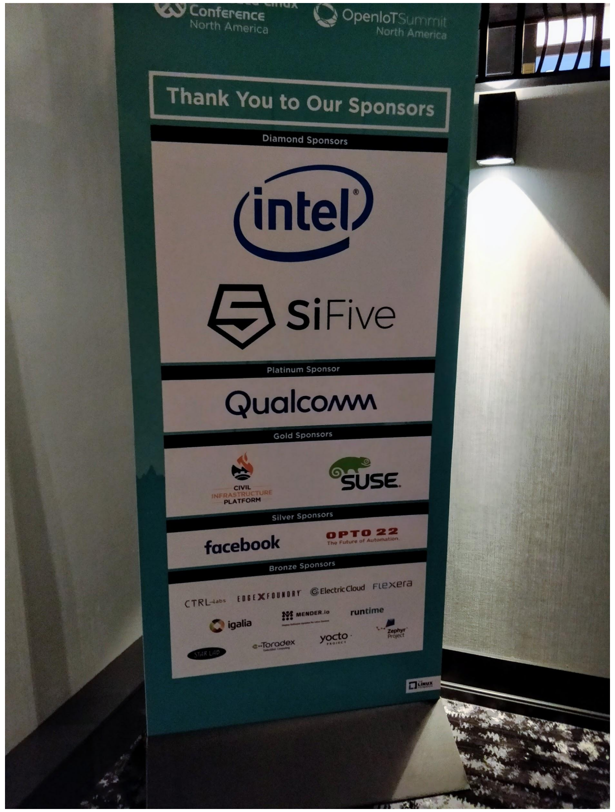


Ordering Annotation	Fence-based Equivalent
l{b|h|w|d|r}.aq	fence r,r,[addr]; l{b|h|w|d|r}; fence r,rw
l{b|h|w|d|r}.aqrl	fence rw,rw; l{b|h|w|d|r}; fence r,rw
s{b|h|w|d|c}.rl	fence rw,w; s{b|h|w|d|c}
s{b|h|w|d|c}.aqrl	fence rw,rw; s{b|h|w|d|c}
amo<op>.aq	amo<op>; fence r,rw
amo<op>.rl	fence rw,w; amo<op>
amo<op>.aqrl	fence rw,rw; amo<op>; fence rw,rw

Table 1.3: Mappings from .aq and/or .rl to fence-based equivalents

RISC-V Linux Kernel Port

- **Upstream Kernel boots**
  - Upstream boots on QEMU



GNU-Based Toolchains

- **binutils, GCC: May, 2017**
- **glibc: February, 2018**
  - only supports rv64i-based ISAs
- **newlib: August, 2017**
- **“Probably not a compiler bug”**

SiFive

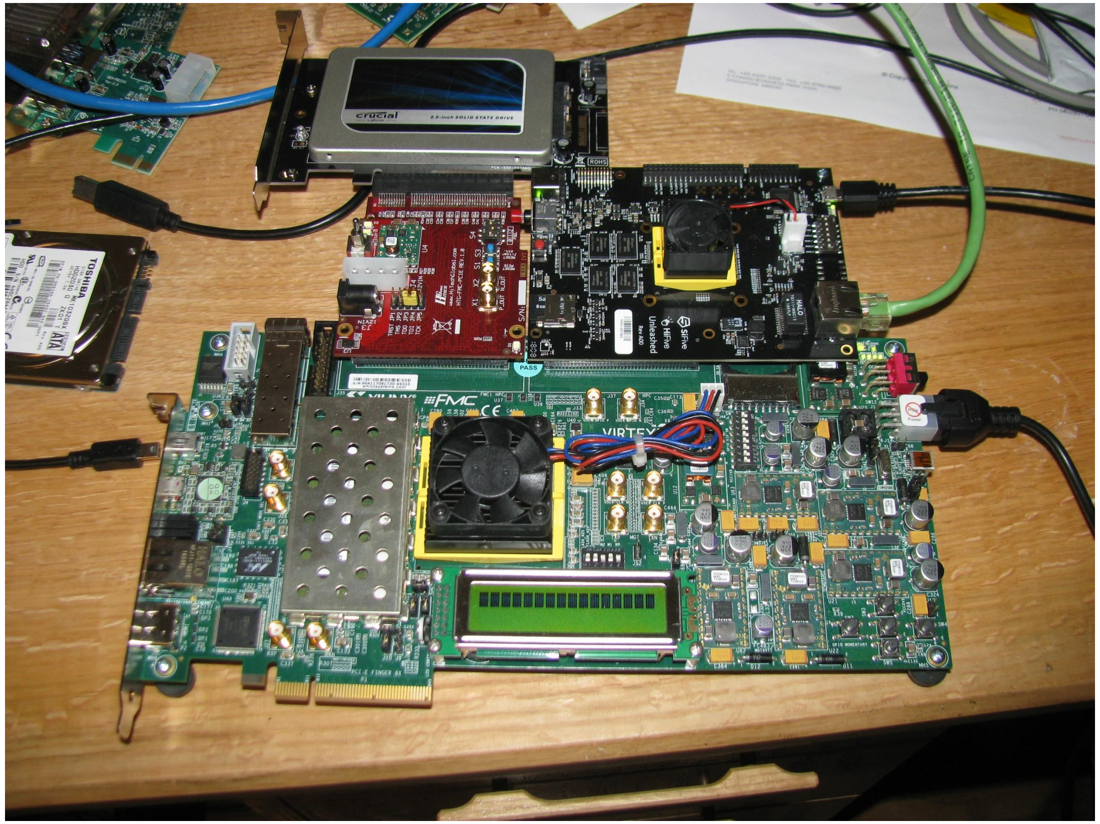
bluespec

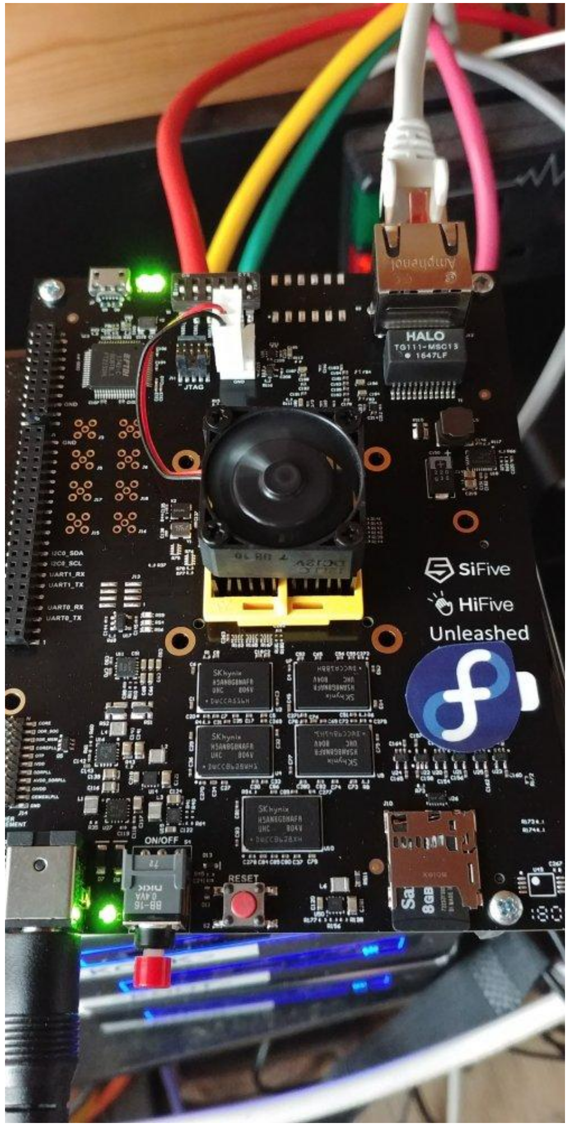
redhat

ANDES TECHNOLOGY

Open Source Hardware

- Fedora runs on open source hardware
- Reproduced open-source FPGA shell
  - http://github.com/sifive/freedom





Open Standards Work!

- Kito Cheng (Andes Technology): GCC and newlib
- Jim Wilson (SiFive): binutils and GCC
- Darius Rad (Bluespec): glibc
- Andrew Waterman (SiFive): binutils, GCC, and glibc
- DJ Delorie (RedHat): glibc

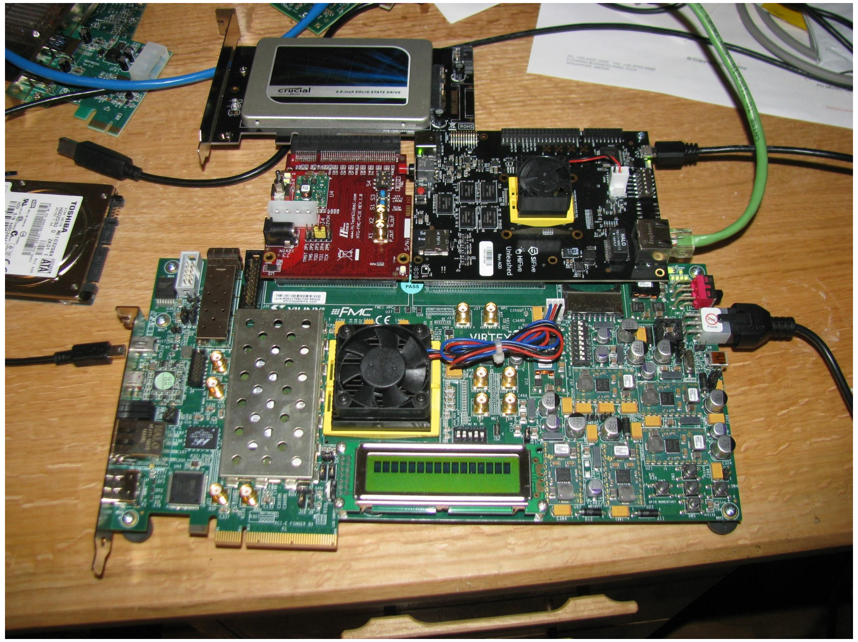
Zeyhpr

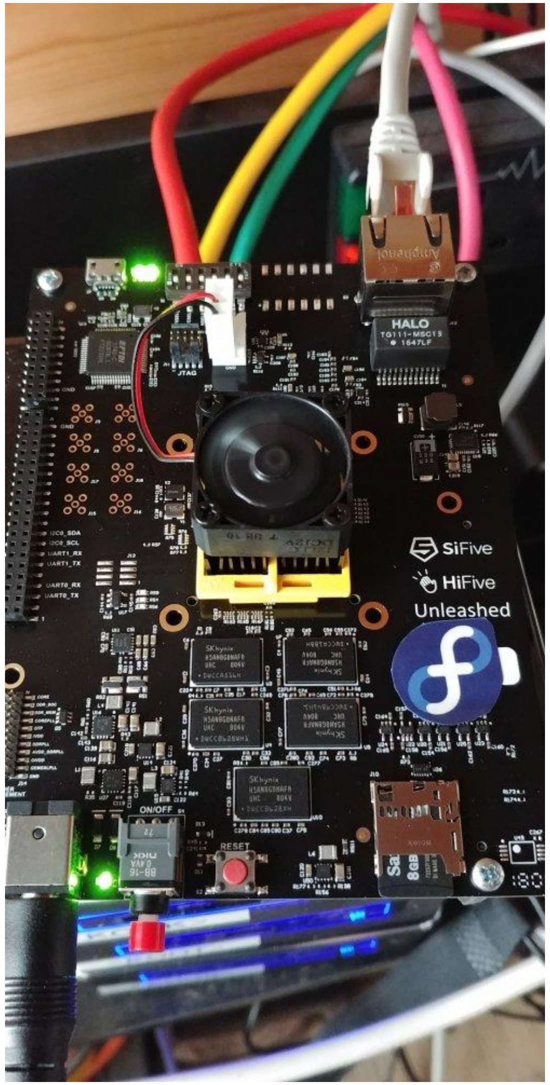
- http://docs.zephyrproject.org/boards/riscv32/index.html
- Upstream support for HiFive1
- Zephyr SDK comes with RISC-V toolchains
- Used in a real product
  - http://badge.antmicro.com/
- Released 1.13.0 recently

antmicro
EMBEDDED SYSTEMS

Fedora

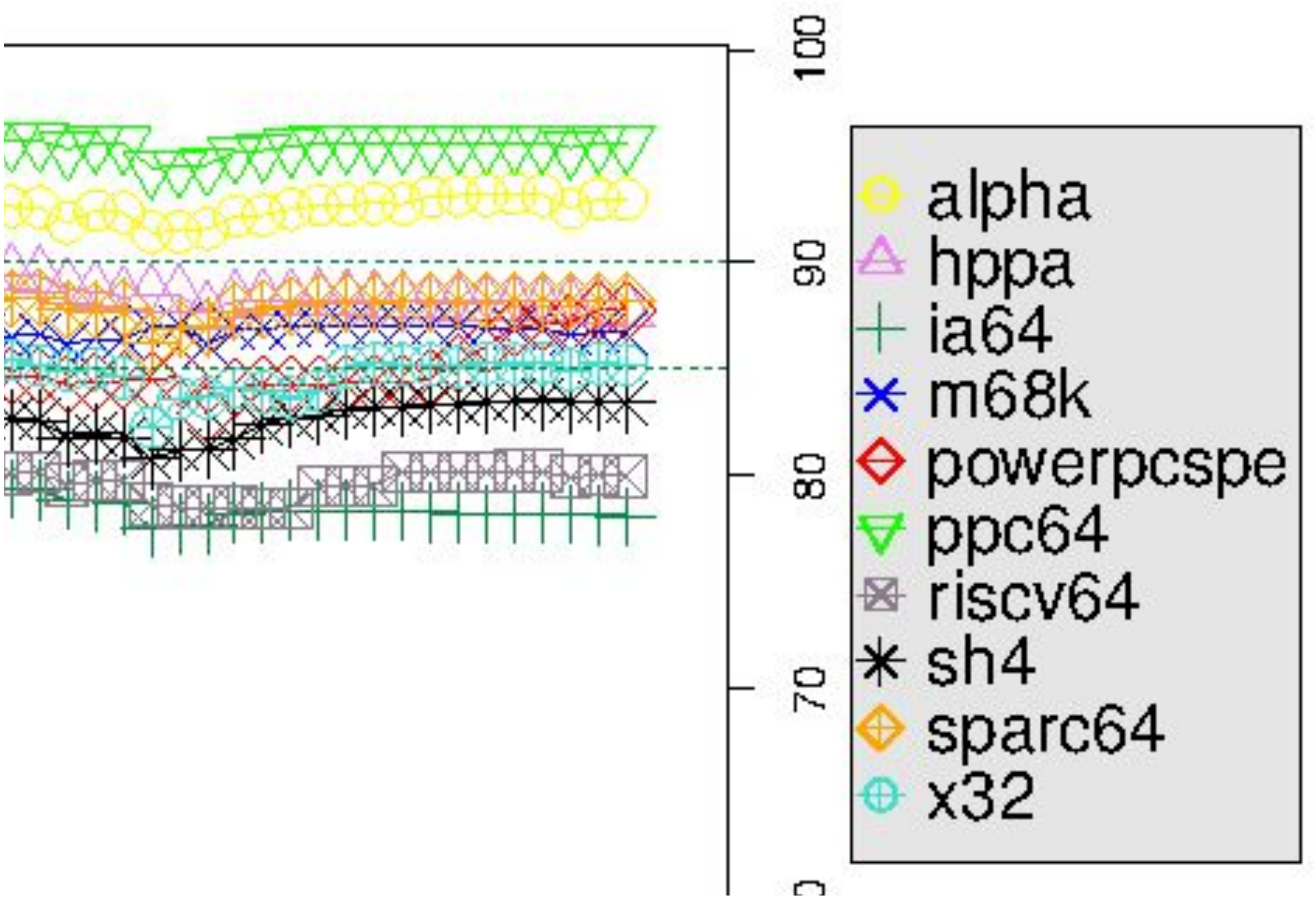
- https://fedoraproject.org/wiki/Architectures/RISC-V
- **Self hosting**
  - HiFive Unleashed
  - QEMU
- **Build farms on HW**





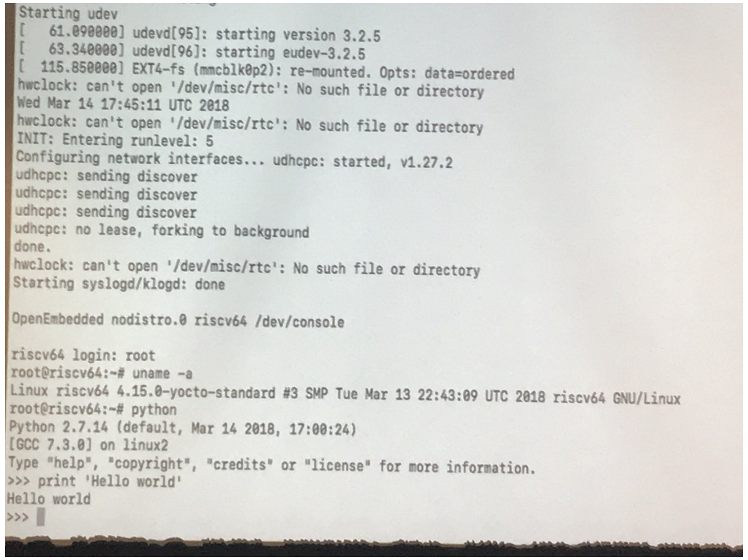
Debian

- https://wiki.debian.org/RISC-V



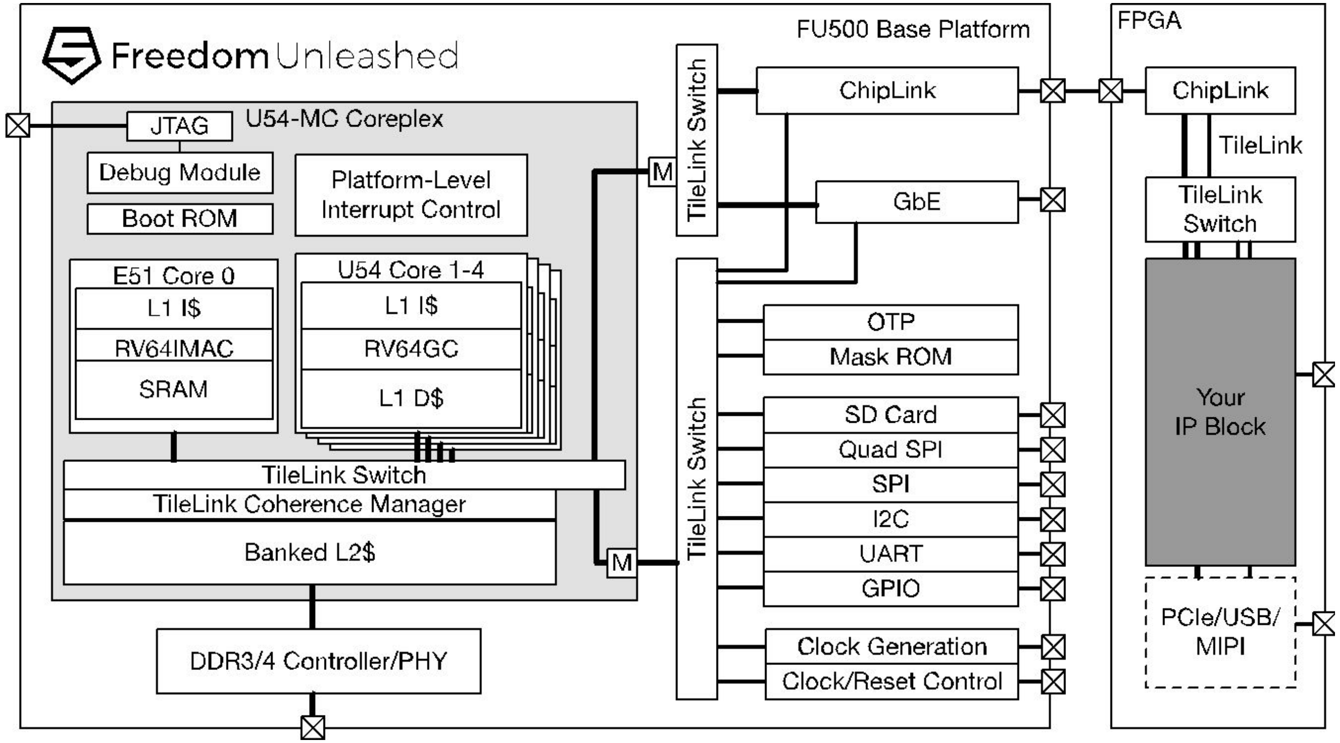
OpenEmbedded/Yocto

- https://github.com/riscv/meta-riscv
- **Port is in good shape**
  - ○ Supported upstream, runs X
  - ○ Works on HiFive Unleashed and QEMU
- **Khem Raj gave a talk at ELC**
  - ○ https://www.youtube.com/watch?v=TdsmjqWJmfc

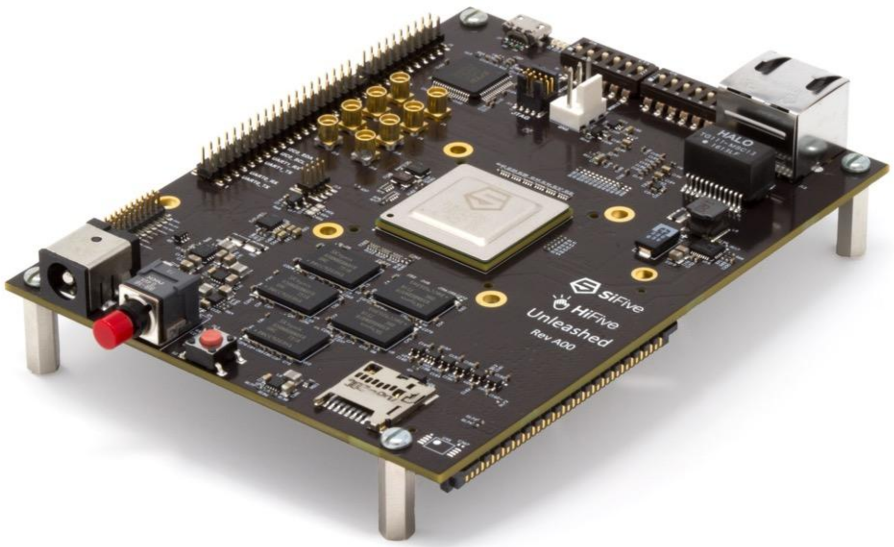


```
Starting udev
[   61.090000] udevd[95]: starting version 3.2.5
[   63.340000] udevd[96]: starting eudev-3.2.5
[  115.850000] EXT4-fs (mmcblk0p2): re-mounted. Opts: data=ordered
hwclock: can't open '/dev/misc/rtc': No such file or directory
Wed Mar 14 17:45:11 UTC 2018
hwclock: can't open '/dev/misc/rtc': No such file or directory
INIT: Entering runlevel: 5
Configuring network interfaces... udhcpc: started, v1.27.2
udhcpc: sending discover
udhcpc: sending discover
udhcpc: sending discover
udhcpc: no lease, forking to background
done.
hwclock: can't open '/dev/misc/rtc': No such file or directory
Starting syslogd/klogd: done

OpenEmbedded nodistro.0 riscv64 /dev/console

riscv64 login: root
root@riscv64:~# uname -a
Linux riscv64 4.15.0-yocto-standard #3 SMP Tue Mar 13 22:43:09 UTC 2018 riscv64 GNU/Linux
root@riscv64:~# python
Python 2.7.14 (default, Mar 14 2018, 17:00:24)
[GCC 7.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print 'Hello world'
Hello world
>>> 
```

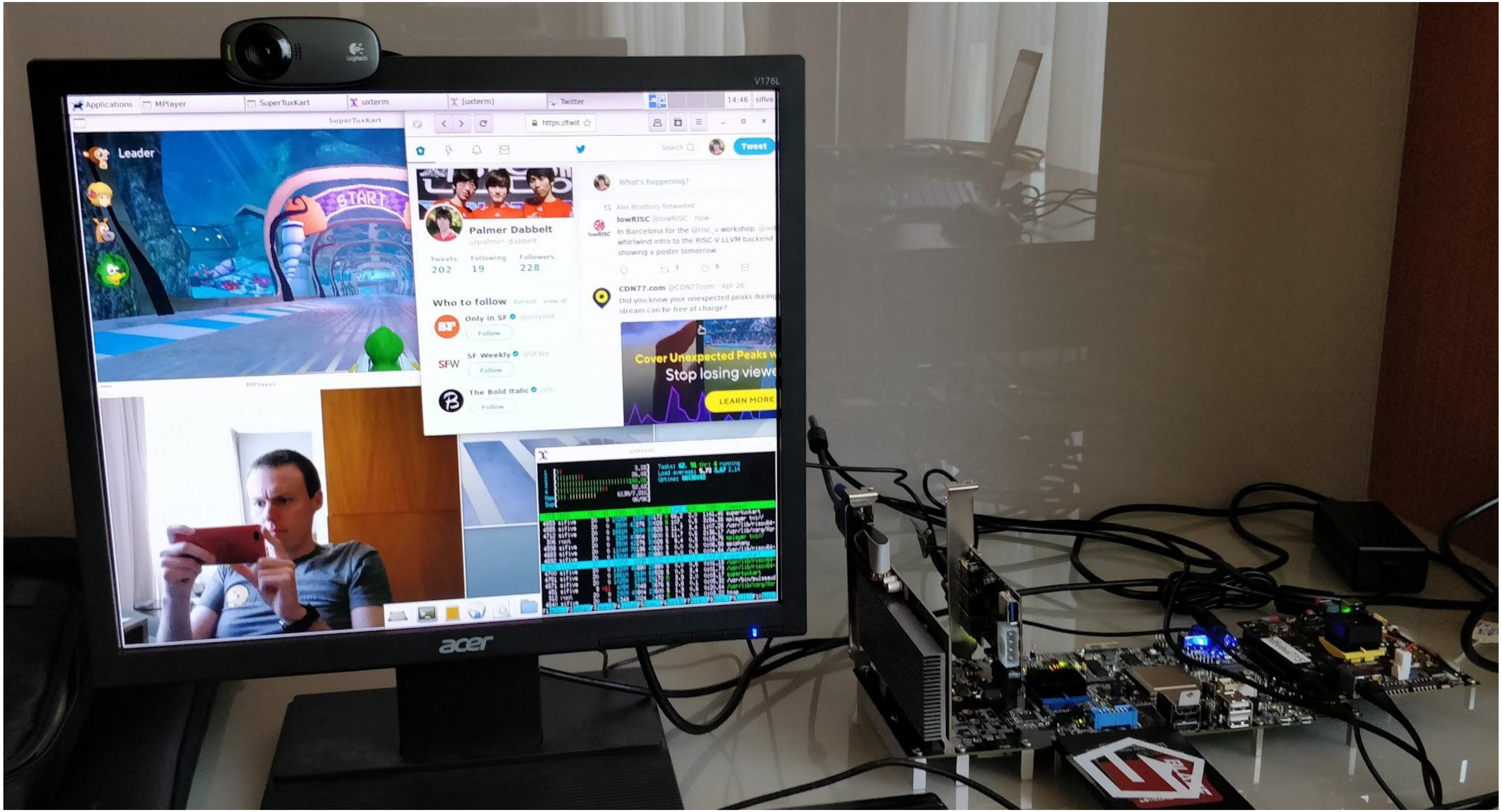
SiFive's HiFive Unleashed



- 1.5GHz Quad Core on 28nm
- 2MiB L2 cache and 64-bit DDR4
- ChipLink board-to-board interconnect

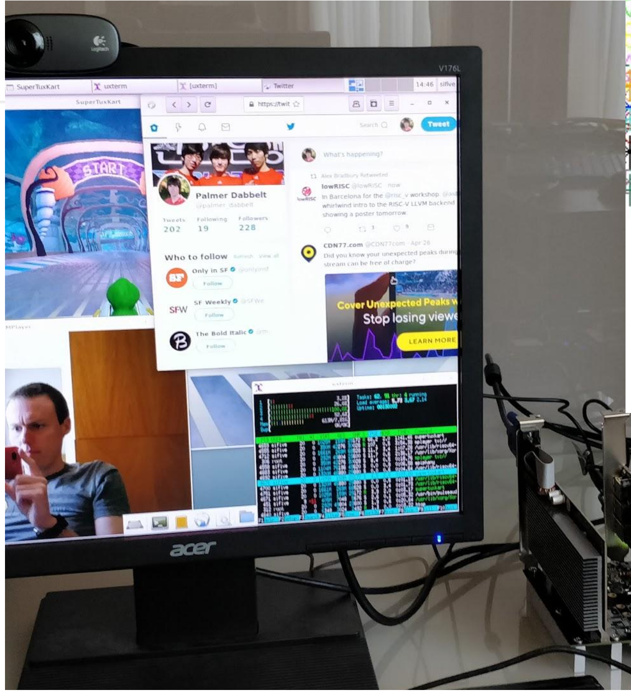


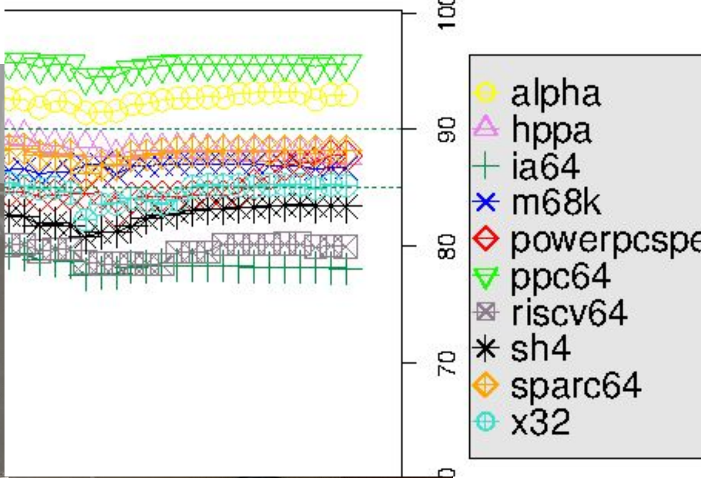
The State of RISC-V Software

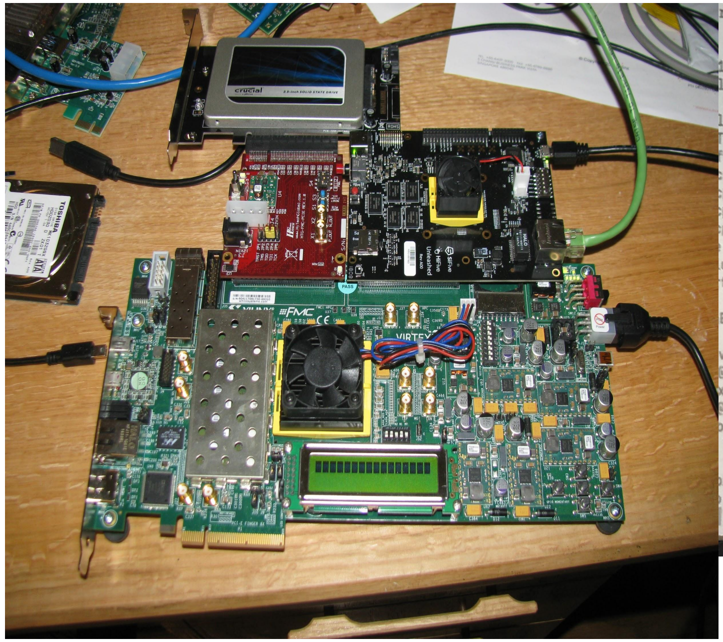


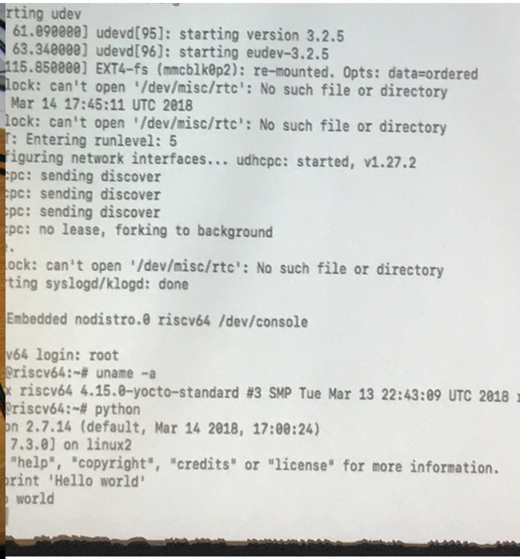
Getting Started

- http://sifive.com/blog
    - “Last Week in RISC-V”
- sw-dev@groups.riscv.org
- #riscv on freenode
- Per-project mailing lists
    - Most are upstream
    - linux-riscv@lists.infradead.org
    - linux-qemu@nongnu.org
    - binutils, GCC, glibc, GDB at FSF









```
rting udev
 61.090000] udevd[95]: starting version 3.2.5
 63.340000] udevd[96]: starting eudev-3.2.5
115.850000] EXT4-fs (mmcblk0p2): re-mounted. Opts: data=ordered
lock: can't open '/dev/misc/rtc': No such file or directory
 Mar 14 17:45:11 UTC 2018
lock: can't open '/dev/misc/rtc': No such file or directory
T: Entering runlevel: 5
figuring network interfaces... udhcpc: started, v1.27.2
cpc: sending discover
cpc: sending discover
cpc: sending discover
cpc: no lease, forking to background
e.
lock: can't open '/dev/misc/rtc': No such file or directory
rting syslogd/klogd: done

Embedded nodistro.0 riscv64 /dev/console

v64 login: root
@riscv64:~# uname -a
x riscv64 4.15.0-yocto-standard #3 SMP Tue Mar 13 22:43:09 UTC 2018 r
@riscv64:~# python
on 2.7.14 (default, Mar 14 2018, 17:00:24)
 7.3.0] on linux2
 "help", "copyright", "credits" or "license" for more information.
print 'Hello world'
 world
```