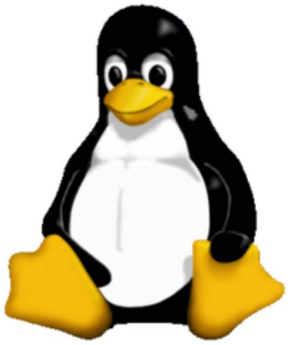
Slides: https://github.com/pdp7/talks/blob/master/rv-elc.pdf

Linux on RISC-V

with open source hardware and open source FPGA tools

Embedded Linux Conference 2020



Drew Fustini

drew@beagleboard.org

Twitter: @pdp7

RISC-V® AMBASSADOR

CC BY SA

CC BY SA

Open Source Hardware designer at OSH Park
(PCB manufacturing service in the USA)
drew@oshpark.com / Twitter: @oshpark

Board of Directors, BeagleBoard.org Foundation
drew@beagleboard.org

Board of Directors, Open Source Hardware Assoc.
Certification Program:
https://certification.oshwa.org/

RISC-V Ambassador for RISC-V International
https://riscv.org/

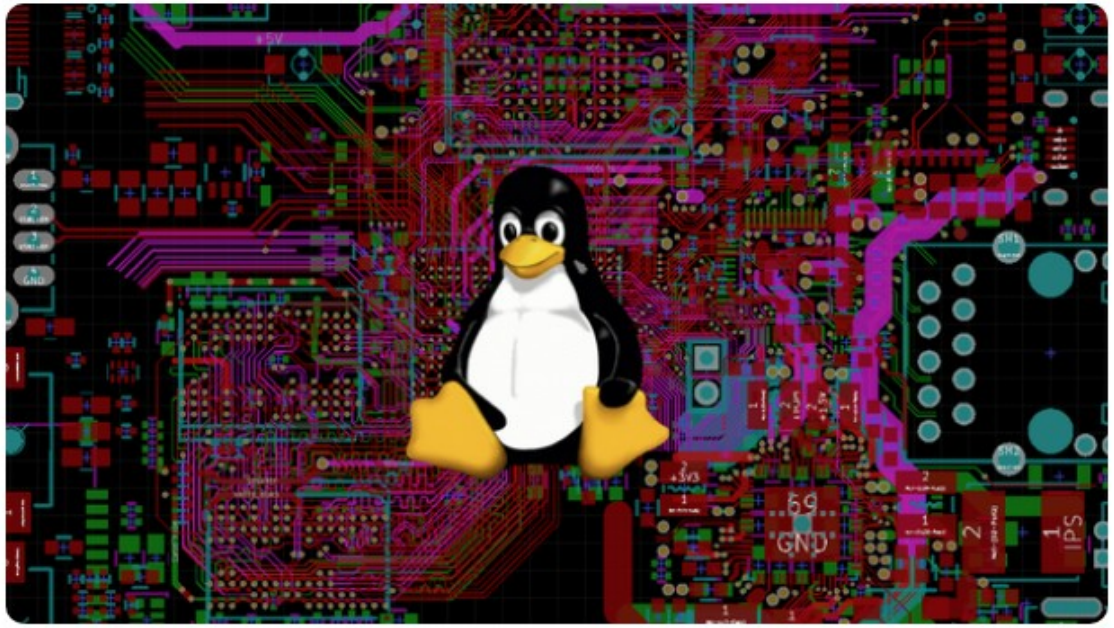
Berlin Embedded Linux M

meetup.com/Berlin-Embedded-Linux-Meetup/

Incognito

Meetup

Start a new group | Log in | Sign up



Berlin Embedded Linux Meetup

Berlin, Germany

140 members · Public group

Organized by **Drew F.** and **2 others**

Share:

About | Events | Members | Photos | Discussions

Join this group

What we're about

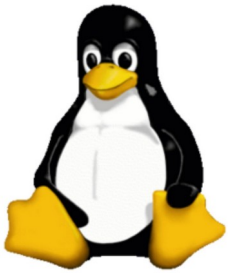
Meetup for those interested in embedded Linux development and Linux kernel development.

Organizers

Drew F. and **2 others**

Message

RISC-V at ELC



- Join **#2-track-riscv** on the OSS+ELC slack!
- RISC-V: Instruction Sets Want to be Free
  - Krste Asanovic (UC Berkeley)
  - Wednesday, 10:35am: https://sched.co/c4PV
- State of RISC-V Software Development Tools
  - Khem Raj (Comcast)
  - Wednesday 12:15pm: https://sched.co/c3Yn
- Ask the Expert Session with Calista Redmond
  - CEO of RISC-V International
  - Thursday 11:45am: https://sched.co/cosw

Munich RISC-V Group

München, Germany
359 members · Public group
Organized by **Flo W.** and **1 other**

Share:

Bay Area RISC-V Group

San Jose, CA
1,197 members · Public group
Organized by **Celeste Cooper** and **4 others**

Share:

Find many more at: https://riscv.org/local/

Open Source Hardware

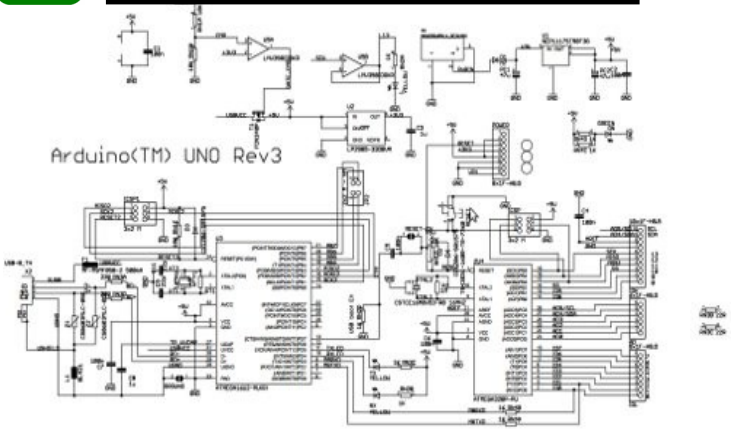
Statement of Principles:

Hardware whose **design** is made **publicly available** so that anyone can **study**, **modify**, **distribute**, **make**, and **sell** the design or hardware based on that design

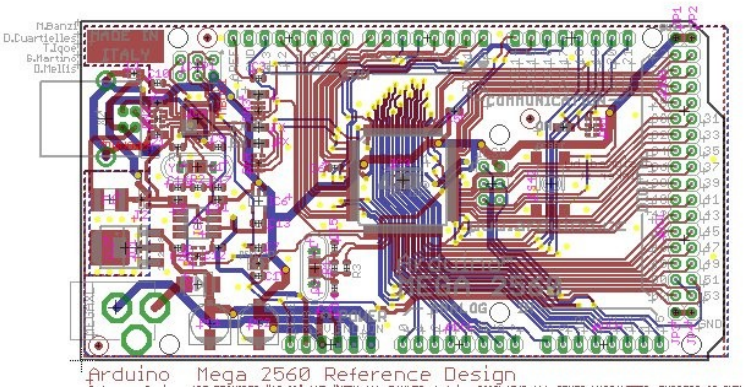
Open Source Hardware

Documentation required for electronics:

☑ Schematics



☑ Board Layout



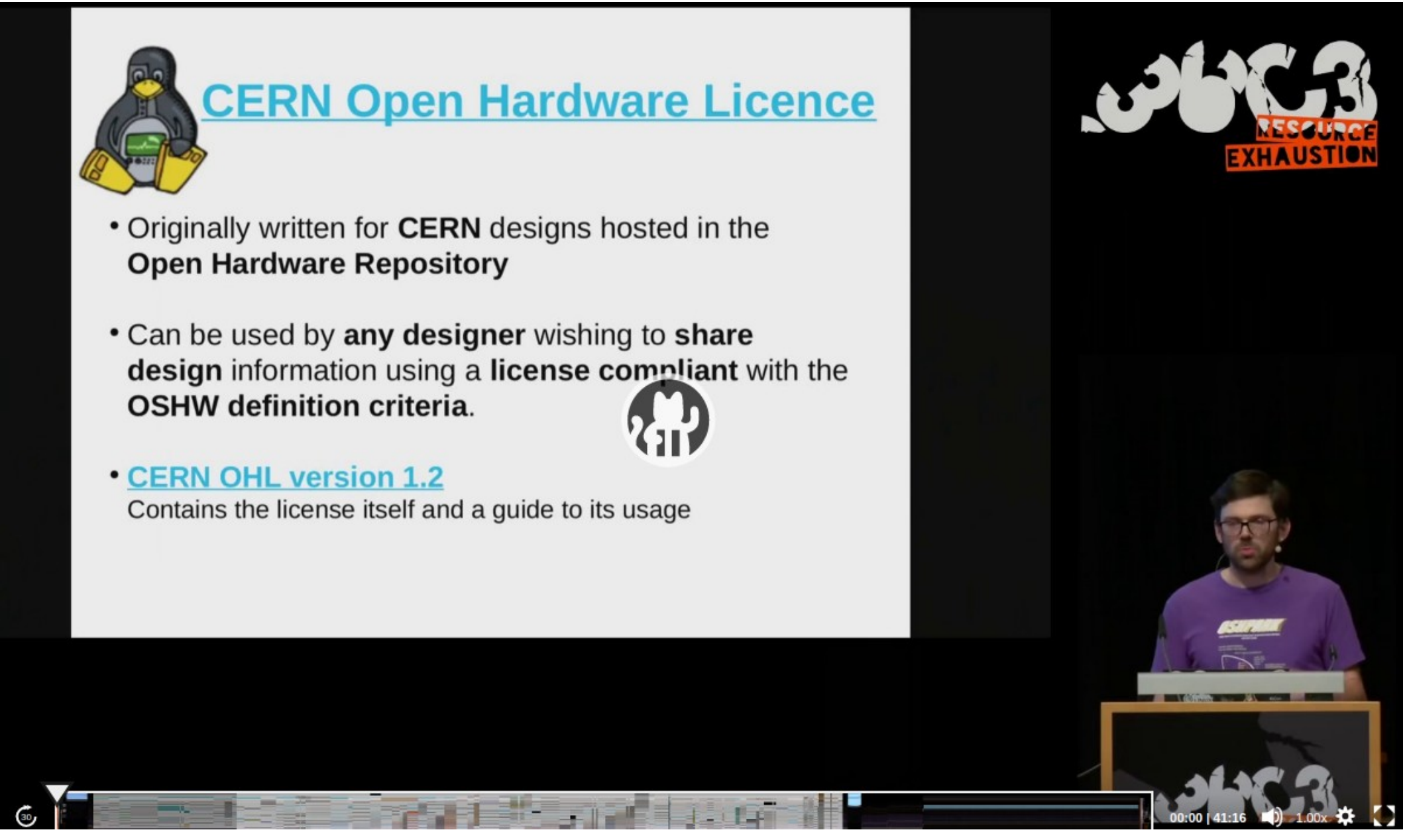
Editable source files for CAD software such as KiCad or EAGLE

☑ Bill of Materials *(BoM)*

Not strict requirement, but best practice is for all components available from distributors in **low quantity**

36c3 talk:

Linux on Open Source Hardware with Open Source chip design



Slides: https://github.com/pdp7/talks/blob/master/rv-elc.pdf

Section:
RISC-V
the instruction set for everything?

- When you write a C or C++ program, it is compiled into instructions for the microprocessor (CPU) to execute.

- How does the compiler know what instructions the CPU understands?
  - defined by the **Instruction Set Architecture**
  - The **ISA** is a standard, a set of rules that define the tasks the processor can perform.
  - Examples: x86 (Intel/AMD) and ARM
    - Both are proprietary and need commercial licensing

- **RISC-V: Free & Open RISC Instruction Set Arch**
  - “new instruction set architecture (ISA) that was originally designed to support computer architecture research and education and is now set to become a standard open architecture for industry”
  - The **5th** RISC instruction to come out out UC Berkeley
  - RISC = Reduced Instruction Set Computer

- **RISC-V: Free & Open RISC Instruction Set Arch**
  - Instruction Sets Want To Be Free: A Case for RISC-V
    - David Patterson, UC Berekely – *co-creator of the original RISC!*
    - https://www.youtube.com/watch?v=mD-njD2QKN0
  - RISC-V Summit 2019: State of the Union
    - Krste Asanovic, UC Berkeley
    - https://www.youtube.com/watch?v=jdkFi9_Hw-c
  - **Watch the Krste's KEYNOTE tomorrow!**
    - **"RISC-V: Instruction Sets Want to be Free"**
    - **Wednesday, July 1 • 10:35am – 10:55am**
    - **https://sched.co/c4PV**

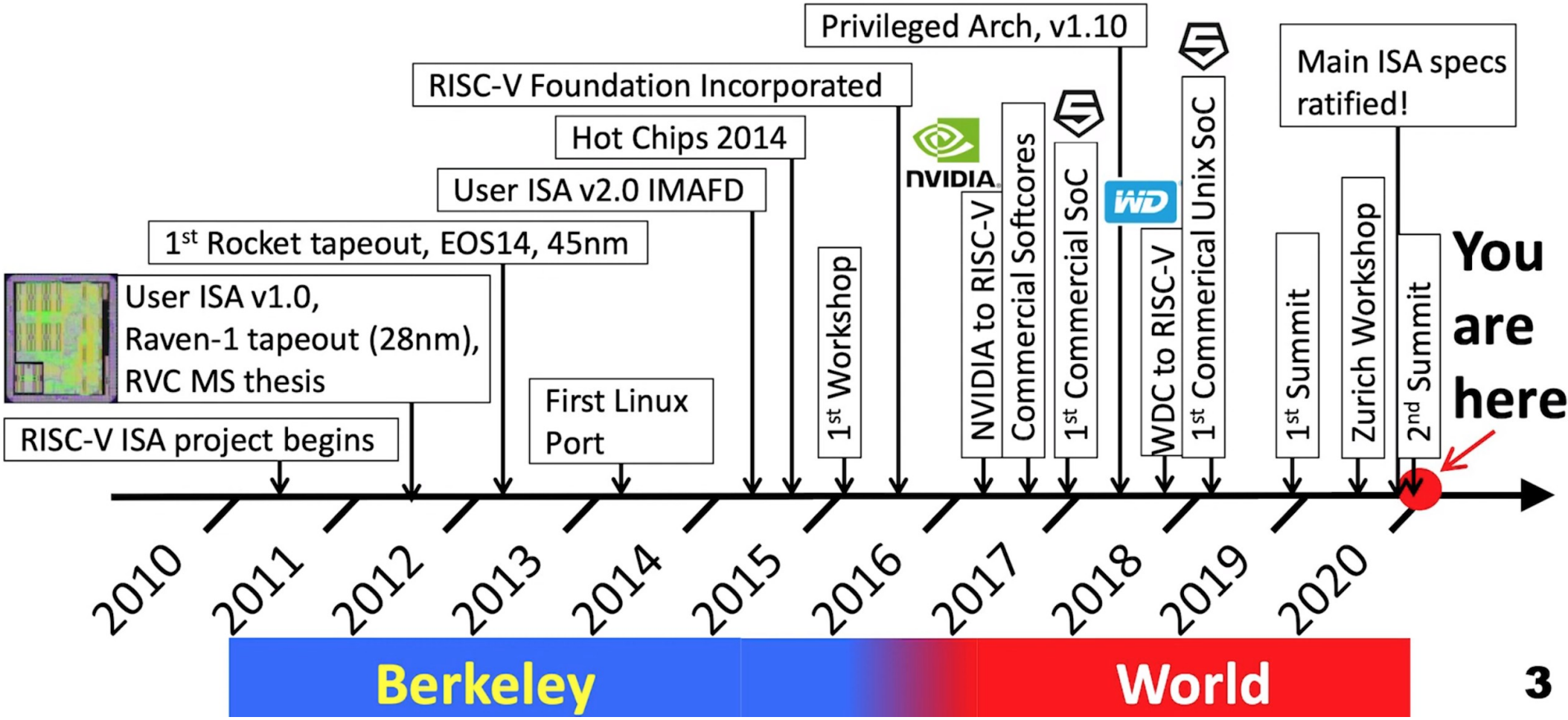
RISC-V Summit 2019: State of the Union https://youtu.be/jdkFi9_Hw-c

What's Different about RISC-V?

- ***Simple***
  - Far smaller than other commercial ISAs
- ***Clean-slate design***
  - Clear separation between user and privileged ISA
  - Avoids µarchitecture or technology-dependent features
- A ***modular*** ISA designed for ***extensibility/specialization***
  - Small standard base ISA, with multiple standard extensions
  - Sparse and variable-length instruction encoding for vast opcode space
- ***Stable***
  - Base and standard extensions are frozen
  - Additions via optional extensions, not new versions
- ***Community designed***
  - With leading industry/academic experts and software developers

RISC-V Summit 2019: State of the Union https://youtu.be/jdkFi9_Hw-c

RISC-V Timeline

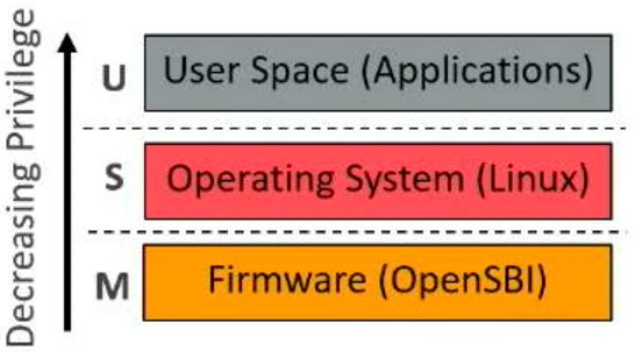


“Co-developing RISC-V Hypervisor Support”, Anup Patel
https://youtu.be/4GoikHB5Qcw

What is RISC-V ?

Free and Open Instruction Set Architecture (ISA)

- **Clean-slate and Extensible ISA**
- **XLEN (machine word length)** can be 32 (RV32), 64 (RV64), and 128 (RV128)
- **32 general purpose registers**
- **Variable instruction length** (instruction compression)
- **Three privilege modes:** Machine (M-mode), Supervisor (S-mode), and User (U-mode)
- **Control and Status Registers (CSR)** for each privilege mode



General Purpose Registers	
zero	Hardwired-zero register
ra	Return address register
sp	Stack pointer register
gp	Global pointer register
tp	Thread pointer register
a0-a7	Function argument registers
t0-t6	Caller saved registers
s0-s11	Callee saved registers

S-mode CSRs (Used By Linux)	
sstatus	Status
sie	Interrupt Enable
sip	Interrupt Pending
stvec	Trap Handler Base
sepc	Trap Progam Counter
scause	Trap Cause
stval	Trap Value
satp	Address Translation
sscratch	Scratch
NOTE: **sedeleg**, **sideleg**, and **scounteren** not used currently	

© 2020 Western Digital Corporation or its affiliates. All rights reserved.

"Co-developing RISC-V Hypervisor Support" - Anup Patel
https://youtu.be/4GoikHB5Qcw

RISC-V Extension

It is a set of ISAs and not one ISA

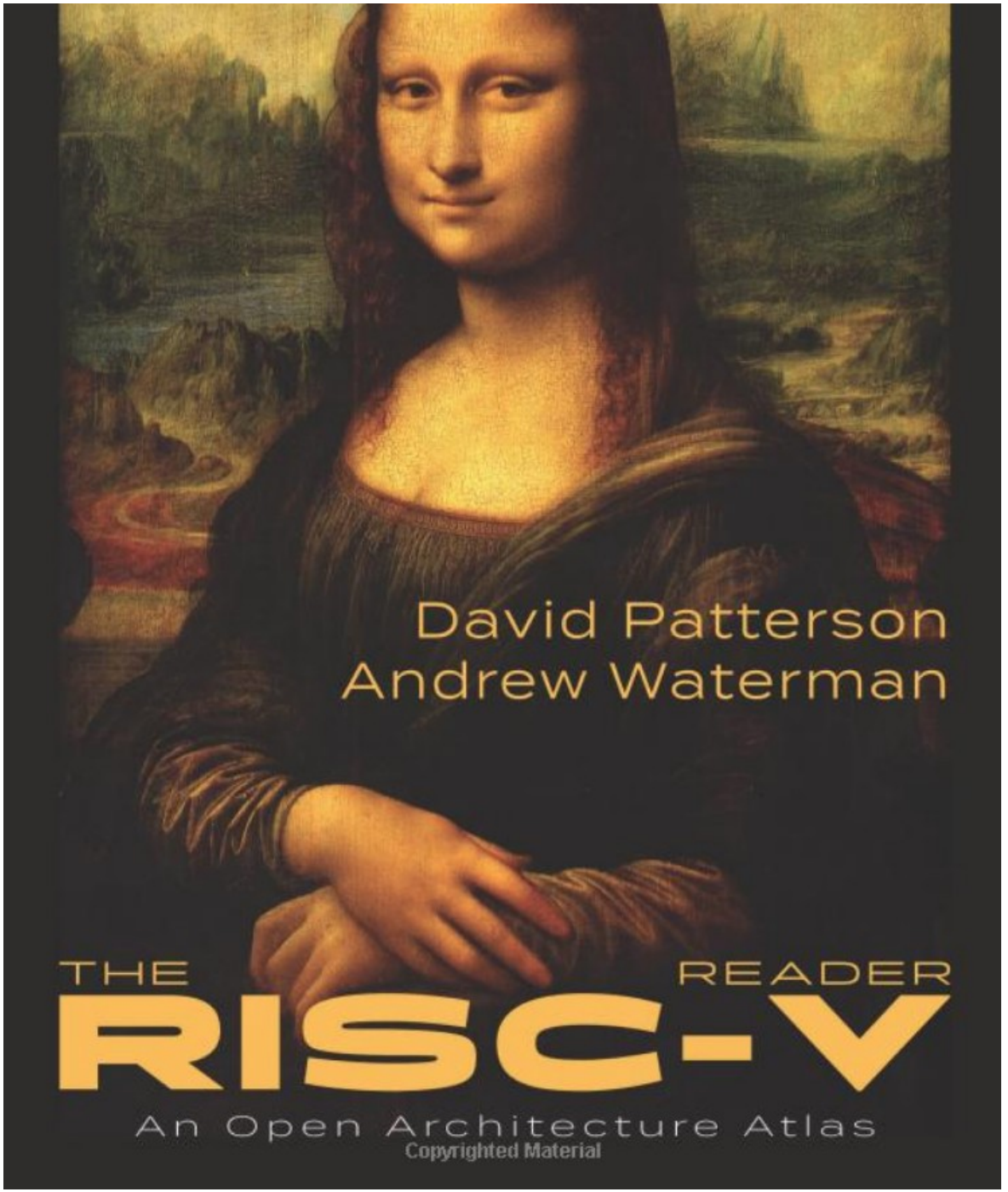
- RISC-V is a set of ISAs where each ISA is called an **RISC-V extension**
- **Standard naming notation** for ISAs supported by RISC-V implementation:
  - RV – Indicates RISC-V architecture
  - [###] – {32, 64, 128} Indicates register length
  - [abc..xyz] – Indicates set of extensions
- **Examples:** RV64IMA, RV64GC, RV32I, RV32IM
- **Linux needs:**
  - RV64IMAC
  - RV64IMAFDC (aka RV64GC)
  - RV32IMAC
  - RV32IMAFDC (aka RV32GC)

Extension	Description
I	Integer
M	Integer multiplication and division
A	Atomics
F	Single-precision floating point
D	Double-precision floating point
G	General purpose = IMAFD
C	16bit compressed instructions
H	Hypervisor
V	Vector
... And More ...	

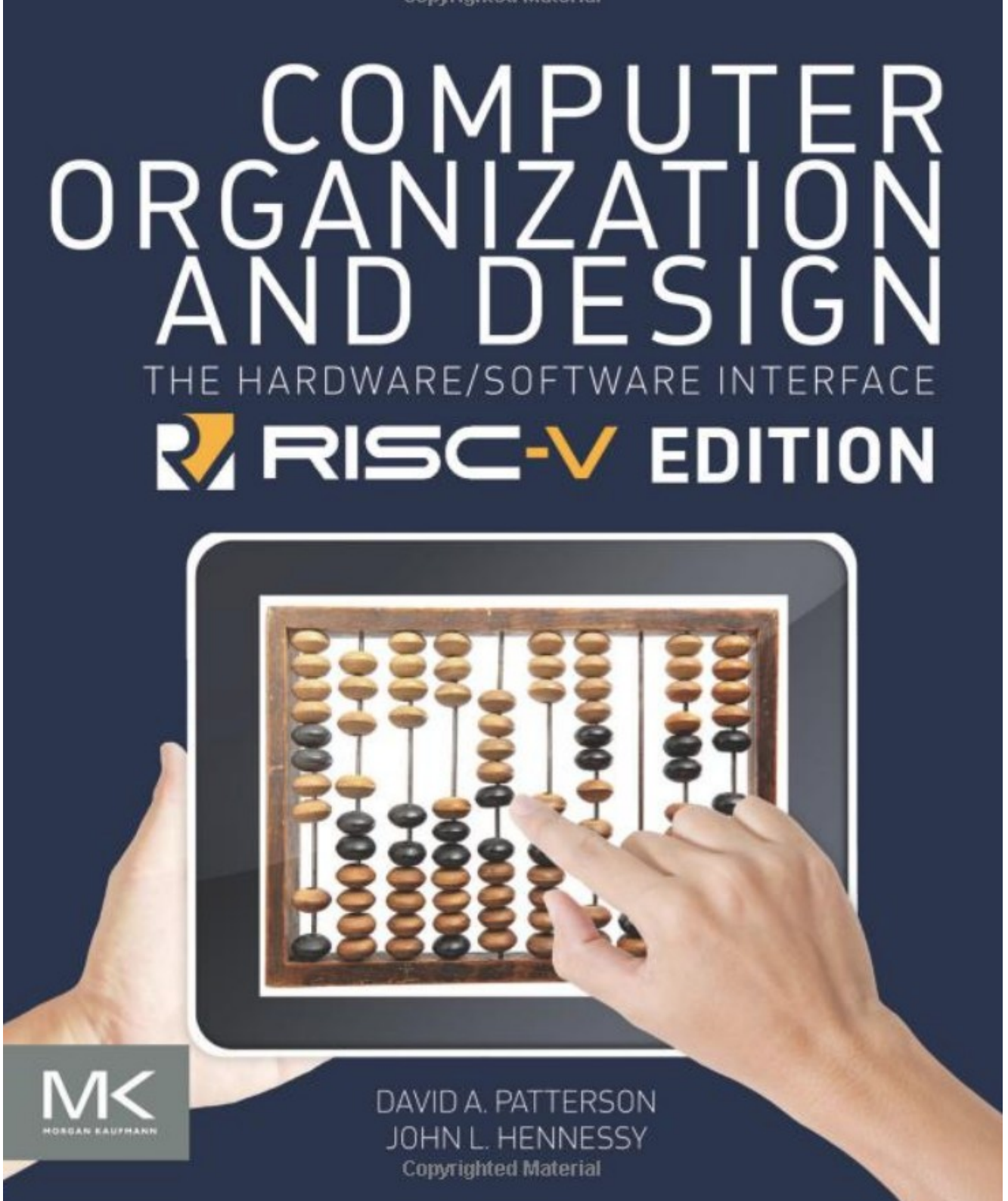
© 2020 Western Digital Corporation or its affiliates. All rights reserved.

Get up-to-speed quick with the **RISC-V Reader**

riscvbook.com



Deep dive with updated classic!



RISC-V Summit 2019: State of the Union https://youtu.be/jdkFi9_Hw-c

RISC-V Ecosystem

Software

Open-source software:
Gcc, binutils, glibc, Linux, BSD, LLVM, QEMU, FreeRTOS, ZephyrOS, LiteOS, SylixOS, …

Commercial software:
Lauterbach, Segger, IAR, Micrium, ExpressLogic, Ashling, AntMicro, Imperas, UltraSoC …

RISC-V Foundation

ISA specification | Golden Model | Compliance

Hardware

Open-source cores:
Rocket, BOOM, RI5CY, Ariane, PicoRV32, Piccolo, SCR1, Shakti, Serv, Swerv, Hummingbird, …

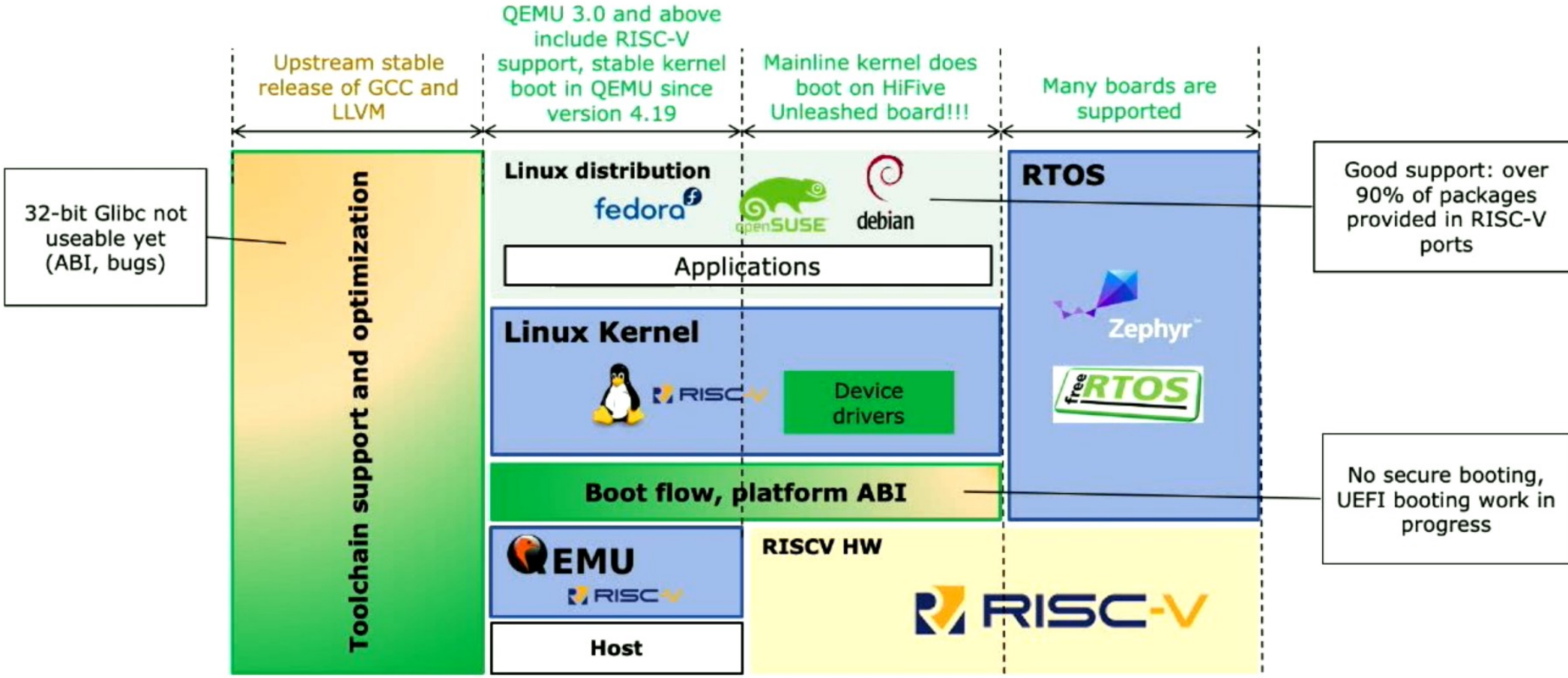
Commercial core providers:
Alibaba, Andes, Bluespec, Cloudbear, Codasip, Cortus, InCore, Nuclei, SiFive, Syntacore, …

Inhouse cores:
Nvidia, WD, +others

"RISC-V software ecosystem in 2020" - Atish Patra *(LCA 2020)*
https://youtu.be/qwkab2Z44pk

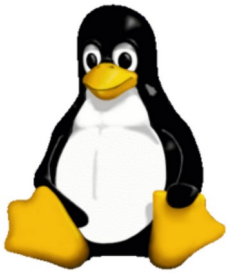
Linux Ecosystem: Status At-a-glance

Fast progress but more work needed



© 2020 Western Digital Corporation or its affiliates. All rights reserved.

RISC-V at ELC



- Go RISC-V Go:
  State of Software Development Tools for RISC-V
  - Khem Raj (Comcast)
  - TOMORROW Wednesday 12:15pm
  - https://sched.co/c3Yn

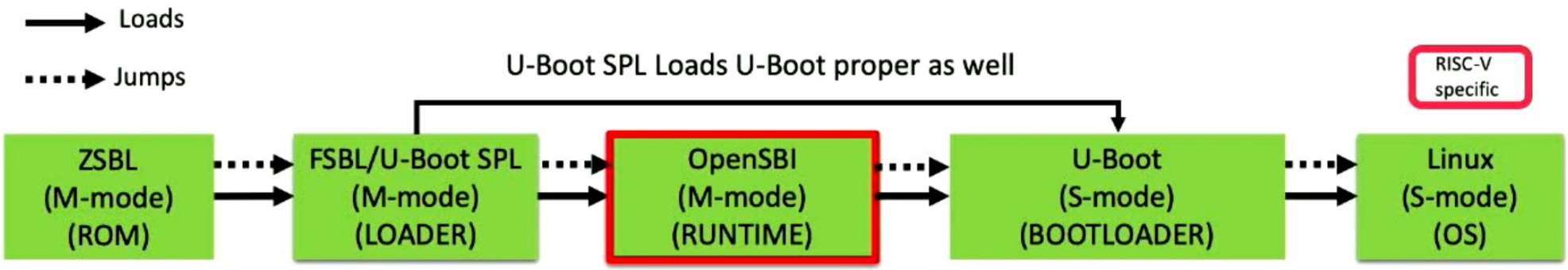
    “Clang/LLVM has newly added backends for RISC-V. In addition, there is now a golang port, which is still out of tree but headed towards upstream acceptance. Additionally, GCC is working towards the GCC-10 release, with several new enhancements to its RISC-V backend. The LLVM backend also means Rust can now cross-compile to the RISC-V architecture. The MUSL C library also has a new RISCV 64-bit port. Additionally, we will discuss the state of tools on the RV32 ecosystem, progress on the glibc port for RV32, and Yocto Project support for RV32 in its core, where QEMU RV32 port is usable for doing application port.”

"RISC-V software ecosystem in 2020" - Atish Patra *(LCA 2020)*

- https://youtu.be/qwkab2Z44pk

RISC-V Boot Flow

Follows commonly used multiple boot stages model



- Follows a standard boot flow
- Uses U-Boot proper as the last stage boot loader
- FSBL is SiFive specific and will be replaced by Coreboot/U-Boot SPL
- OpenSBI is a RISC-V specific runtime service provider
- All harts jump to Linux at the same time and a lottery-based approach chooses the booting hart

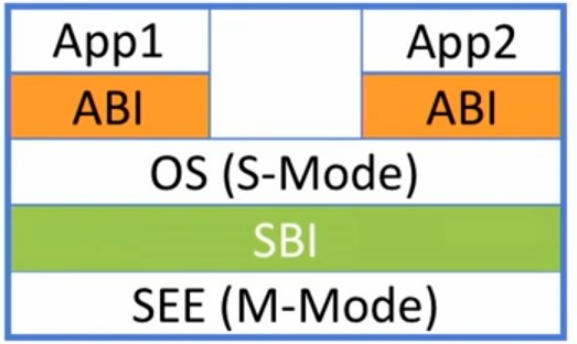
© 2020 Western Digital Corporation or its affiliates. All rights reserved.

"OpenSBI Deep Dive" - Anup Patel *(RV Workshop 2019)*

- https://youtu.be/jstwB-o9ll0

What is SBI ?

- SBI stands for RISC-V Supervisor Binary Interface
  - System call style calling convention between Supervisor (S-mode OS) and Supervisor Execution Environment (SEE)
- SEE can be:
  - A M-mode RUNTIME firmware for OS/Hypervisor running in HS-mode
  - A HS-mode Hypervisor for Guest OS running in VS-mode
- SBI calls help:
  - Reduce duplicate platform code across OSes (Linux, FreeBSD, etc)
  - Provide common drivers for an OS which can be shared by multiple platforms
  - Provide an interface for direct access to hardware resources (M-mode only resources)
- Specifications being drafted by the Unix Platform Specification Working group
  - Maintain and evolve the SBI specifications
  - Currently, SBI v0.1 in-use and SBI v0.2 in draft stage



© 2019 Western Digital Corporation or its affiliates. All rights reserved.

"OpenSBI Deep Dive" - Anup Patel *(RV Workshop 2019)*

- https://youtu.be/jstwB-o9ll0

What is OpenSBI ?

- OpenSBI is an open-source implementation of the RISC-V Supervisor Binary Interface (SBI) specifications
  - Licensed under the terms of the BSD-2 clause license
  - Helps to avoid SBI implementation fragmentation
- Aimed at providing RUNTIME services in M-mode
  - Typically used in boot stage following ROM/LOADER
- Provides support for reference platforms
  - Generic simple drivers included for M-mode to operate
    - PLIC, CLINT, UART 8250
  - Other platforms can reuse the common code and add needed drivers

© 2019 Western Digital Corporation or its affiliates. All rights reserved.

UEFI Support

- [RFC PATCH 00/11] Add UEFI support for RISC-V
  - Atish Patra (June 25, 2019)
  - “This series adds UEFI support for RISC-V”
    - Linux kernel: 5.8-rc2
    - U-Boot: master
    - OpenSBI: master

"OpenSBI Deep Dive" - Anup Patel *(RV Workshop 2019)*
https://youtu.be/jstwB-o9ll0

QEMU

In very good shape

- Use mainline QEMU ! No forks !
- QEMU can boot 32-bit and 64-bit mainline Linux since kernel 4.19
- QEMU can run OpenSBI, U-Boot and Coreboot
- OpenSBI v0.5 included with QEMU
- Hypervisor extensions v0.5 supported
  - Patches waiting for review
  - Specifications not frozen yet
- Vector extensions
  - Patches are in mailing list
  - Specifications not frozen yet
- Newest QEMU 4.2 brings many improvements
  - QEMU sifive_u machine can now boot same binaries that run on HiFive Unleashed
  - SPI flash on virt machine and HiFive Unleashe modelled. This allows FSBL (oreboot and coreboot) testing on QEMU
  - PMP access improvements
  - Support loading initrd for sifive_u machine

© 2020 Western Digital Corporation or its affiliates. All rights reserved.

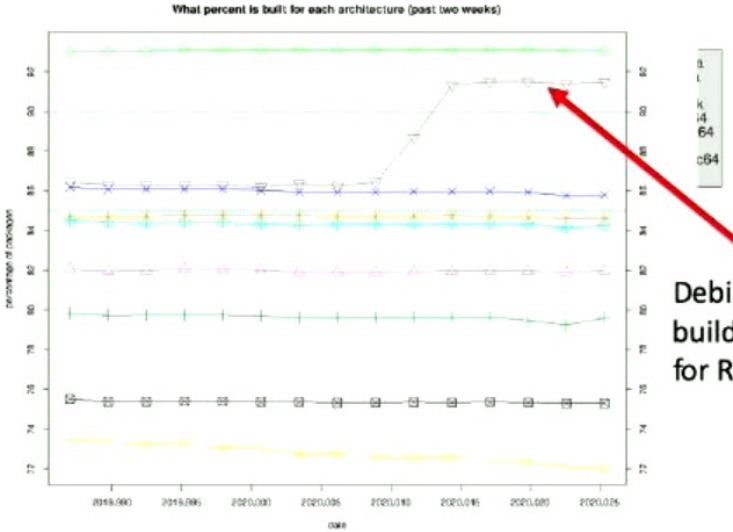
"OpenSBI Deep Dive" - Anup Patel *(RV Workshop 2019)*
https://youtu.be/jstwB-o9ll0

Linux Distributions

Fedora, Debian and OpenSuse are very active

- Fedora: stage 4 disk images released
  - https://fedoraproject.org/wiki/Architectures/RISC-V
  - Supports 3 boards + 142 QEMU VMs
  - Fedora 32/Rawhide including debug and source packages
- Debian
  - https://wiki.debian.org/RISC-V
  - ~92% of packages built
- OpenSuse Tumbleweed
  - https://download.opensuse.org/ports/riscv/
- OpenEmbedded has RISC-V support
  - Demos of running Plasma Mobile and Mycroft are available
  - https://github.com/riscv/meta-riscv

RISC-V graphical desktop on Fedora



Debian packages build percentage for RISC-V

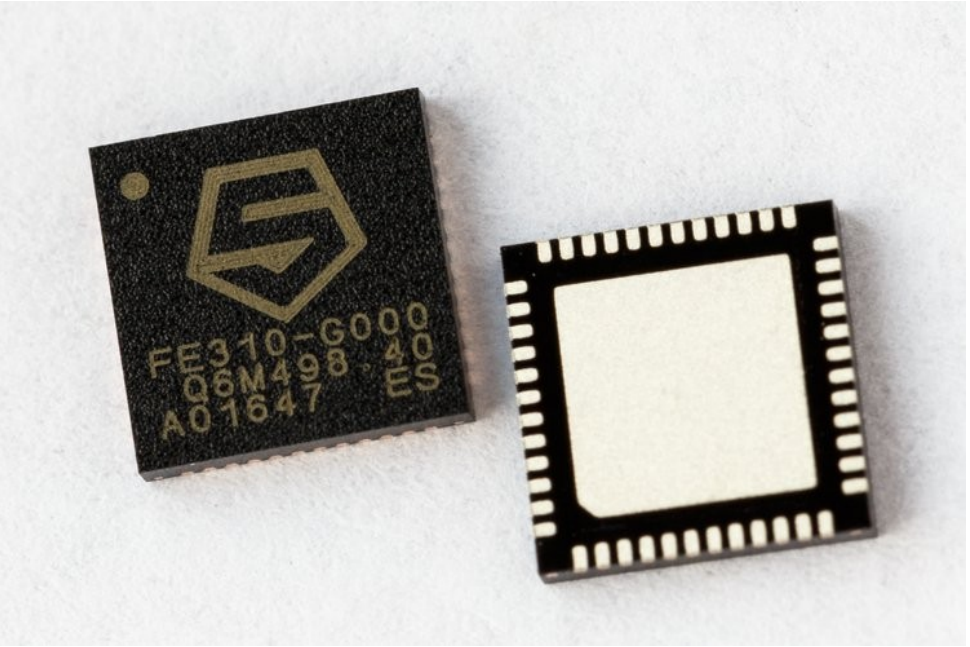
© 2020 Western Digital Corporation or its affiliates. All rights reserved.

RISC-V and Industry

- Designed to be extensible: microcontroller to supercomputer!
- RISC-V International now controls standard: riscv.org
  - Over 400 members: companies, universities and more
  - Transitioning from RISC-V Foundation to Swiss-based RISC-V International
  - YouTube channel has hundreds of talks!
- Companies like Nvidia and Western Digital will ship millions of devices
- Avoid ISA licensing fees
- Freedom to choose micro-architecture implementation
  - only a few companies like Apple and Qualcomm has licenses to do with ARM
- Freedom to leverage existing open source implementations
  - Berkeley's Rocket and BOOM, ETH Zurich's PULP cores, Western Digital SweRV

- lowRISC is a not-for-profit organization whose goal is to produce a fully open source System-on-Chip (SoC) in volume
  - “We will produce a SoC design to populate a low-cost community development board and to act as an ideal starting point for derivative open-source and commercial designs”
- OpenTitan project with Google
  - Announcing OpenTitan, the First Transparent Silicon Root of Trust

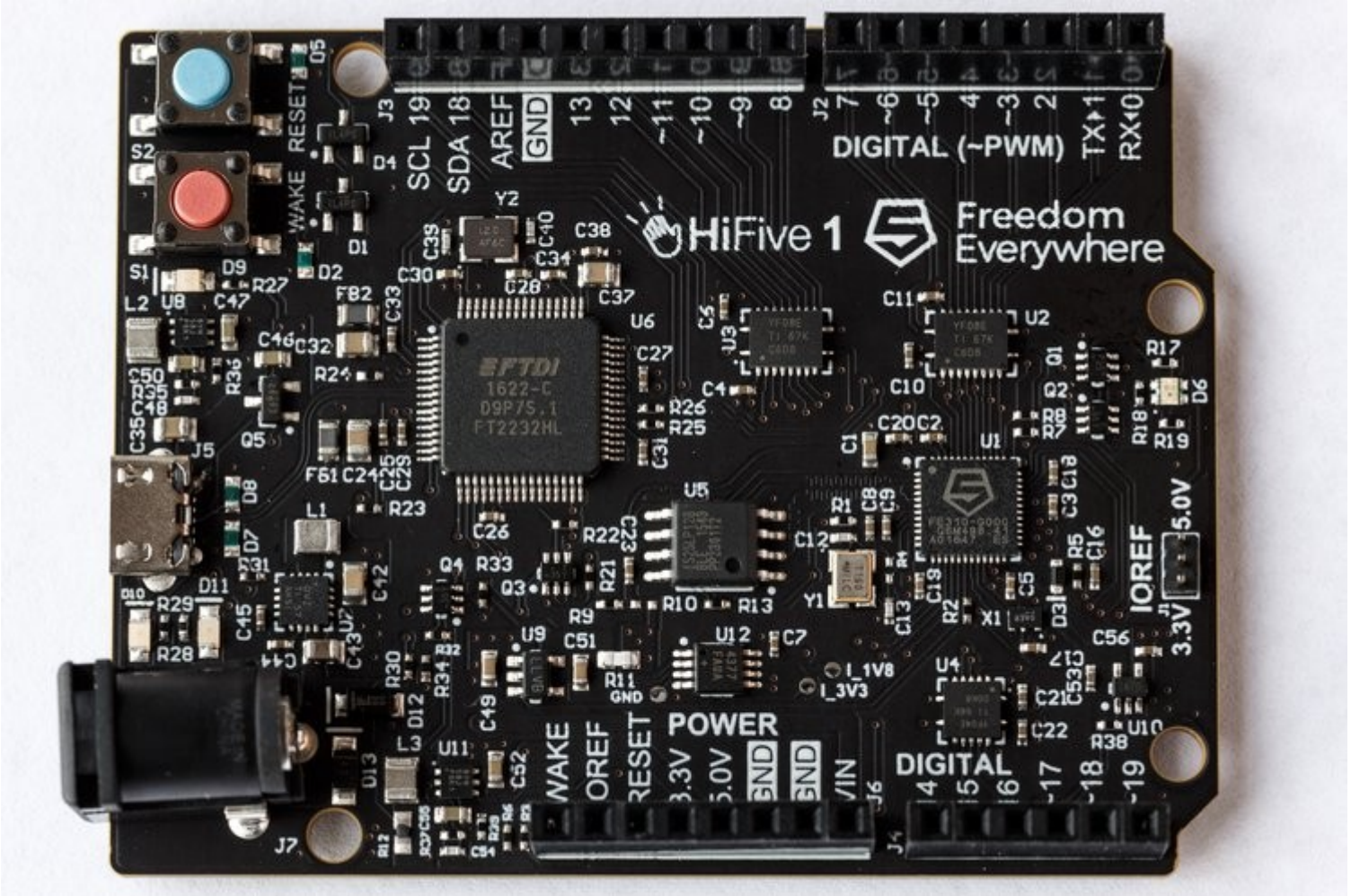
SiFive



- “founded by the creators of the free and open RISC-V architecture as a reaction to the end of conventional transistor scaling and escalating chip design costs”

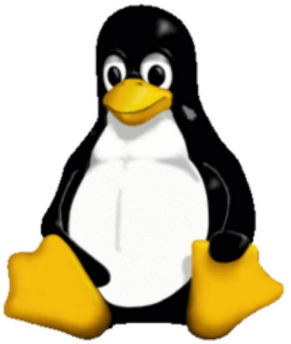
SiFive FE310 microcontroller

- **HiFive1**: Arduino-Compatible RISC-V Dev Kit



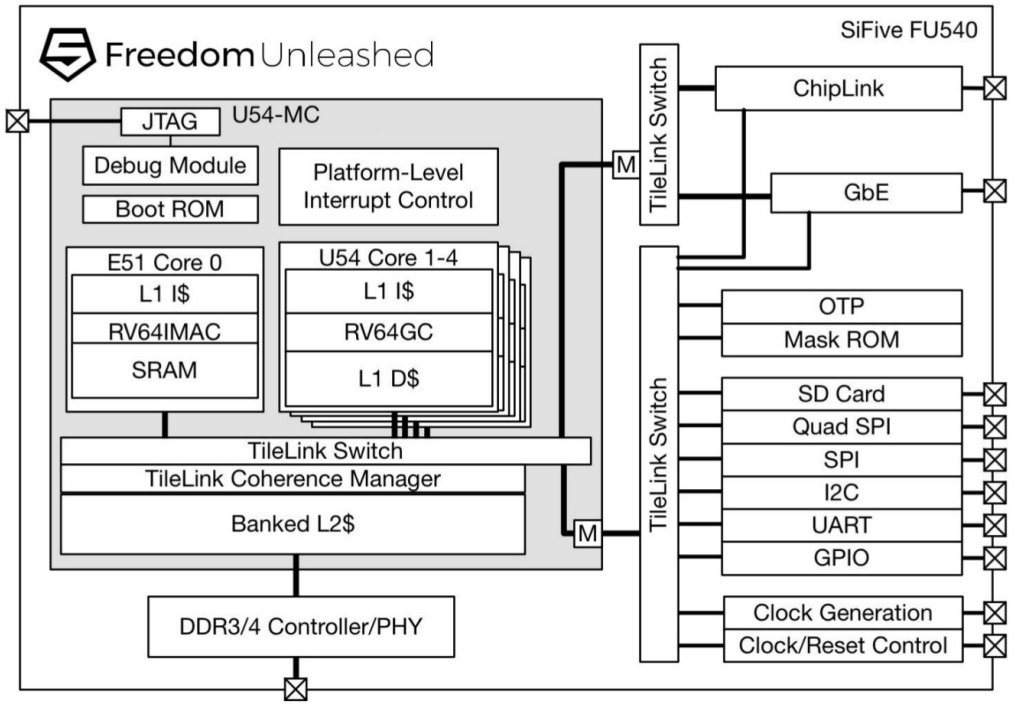
Slides: https://github.com/pdp7/talks/blob/master/rv-elc.pdf

Section:
Linux-capable RISC-V chips



- **SiFive Freedom FU540 SoC**
  - FOSDEM 2018 talk: “Igniting the Open Hardware Ecosystem with RISC-V”

FU540: Penta-Core 64-bit RISC-V Linux SoC



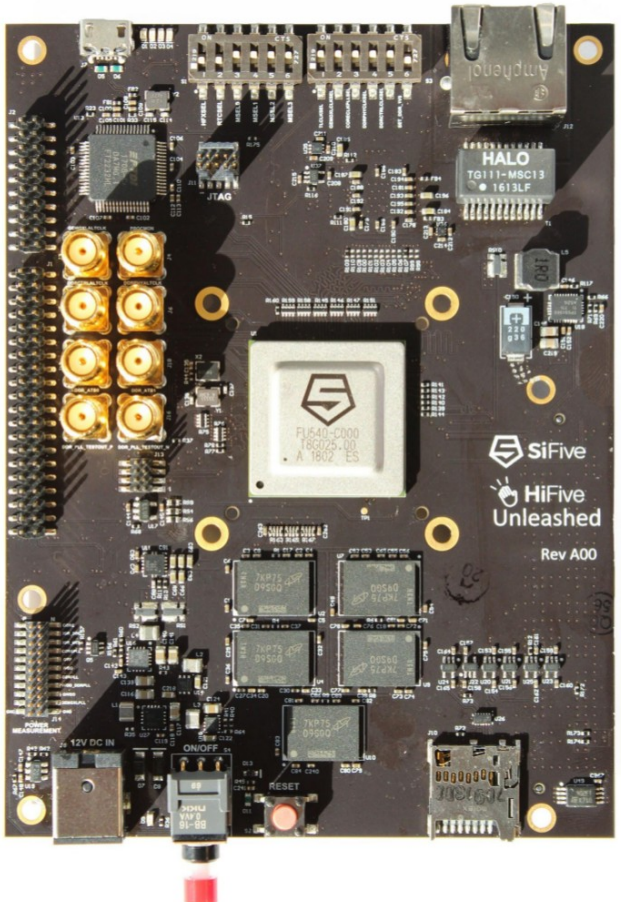
SiFive

36

COPYRIGHT 2018 SIFIVE. ALL RIGHTS RESERVED.

- **SiFive Freedom FU540 SoC**
  - HiFive Unleashed board:
    - powerful but expensive ($1,000) and very limited

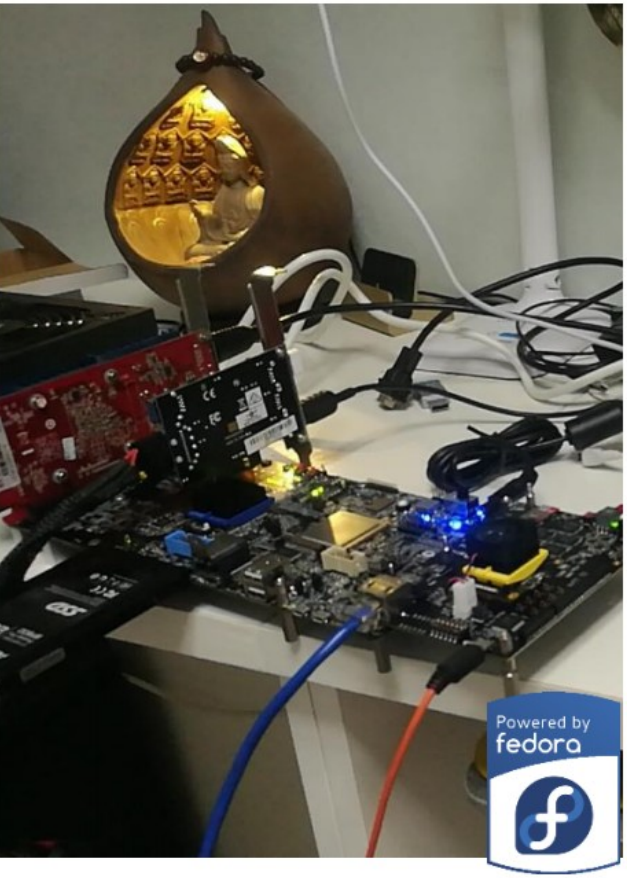
HiFive Unleashed

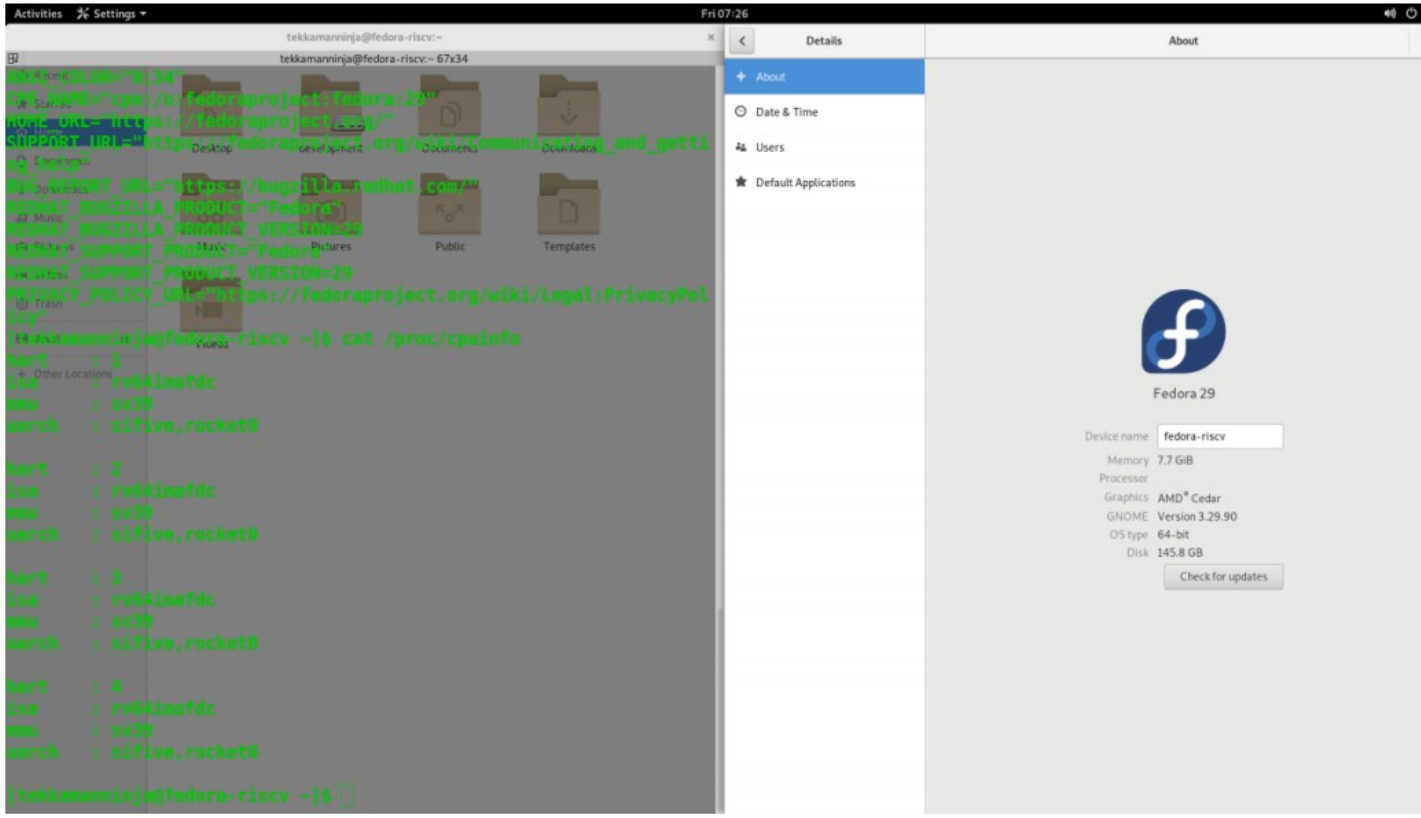


- World's First Multi-Core RISC-V Linux Development Board
  - SiFive FU540-C000 (built in 28nm)
    - 4+1 Multi-Core Coherent Configuration, up to 1.5 GHz
    - 4x U54 RV64GC Application Cores with Sv39 Virtual Memory Support
    - 1x E51 RV64IMAC Management Core
    - Coherent 2MB L2 Cache
    - 64-bit DDR4 with ECC
    - 1x Gigabit Ethernet
  - 8 GB 64-bit DDR4 with ECC
  - Gigabit Ethernet Port
  - 32 MB Quad SPI Flash
  - MicroSD card for removable storage
  - FMC connector for future expansion with add-in cards

“Fedora on RISC-V”, Wei Fu (RV Summit 2019)
https://www.youtube.com/watch?v=WC6e3g8uWdk

Fedora GNOME Image on SiFive Unleashed





Red Hat

CLUB-
MATE
SiPEED
RISC-V

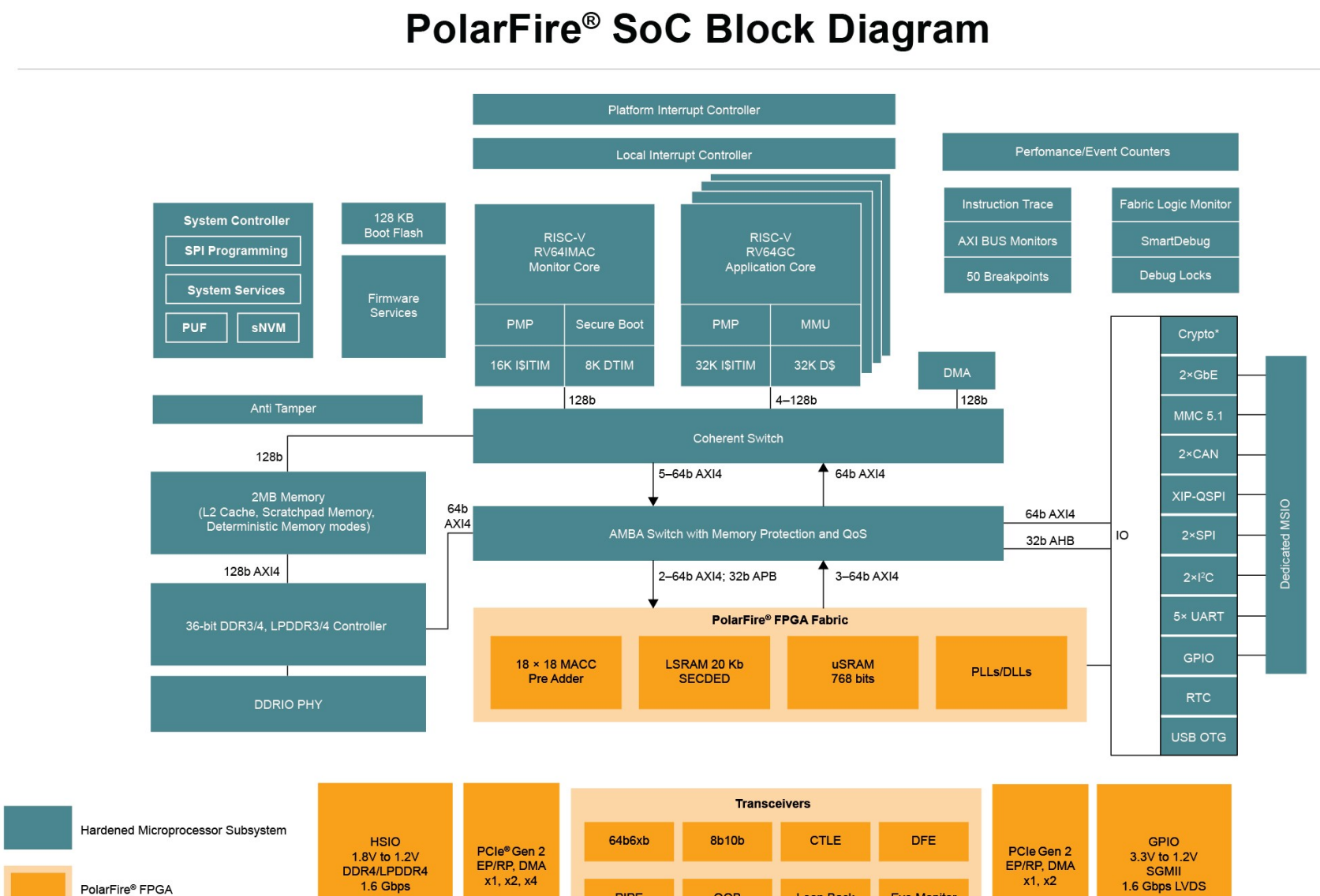
- **Kendryte K210**

  - dual core 64-bit RISC-V at 400MHz with 8MB SRAM
  - Sipeed MAix BiT for RISC-V is only $13!
  - Damien Le Moal at Linux Plumbers 2019
    - RISC-V NOMMU and M-mode Linux
  - Full support coming in Linux 5.8
  - Buildroot with busybox (https://git.io/JJflC)
  - need NOMMU/FDPIC support for better userspace
    - https://youtu.be/GydyykyNjxs (Maciej W. Rozycki)
    - 8MB runs out very quick!
  - there is a MMU but the wrong spec :(
  - u-boot patch series
    - [PATCH v14 00/20] riscv: Add Sipeed Maix support
    - Sean Anderson (June 24th)

File Edit Log Configuration Control signals View Help
/ # uname -a
Linux k210 5.6.0-rc1vowstar #1 SMP Mon Feb 17 23:
/ # cat /proc/meminfo |head
MemTotal: 6656 kB
MemFree: 2496 kB
MemAvailable: 2080 kB
Buffers: 0 kB
Cached: 1916 kB
SwapCached: 0 kB
Active: 0 kB
Inactive: 0 kB
Active(anon): 0 kB
Inactive(anon): 0 kB
/ # cat /proc/cpuinfo
processor : 0
hart : 0
isa : rv64imafdc
processor : 1
hart : 1
isa : rv64imafdc
/ # tcc -run -nostdlib hello.c
hello.c:2: warning: implicit declaration of functi
hello show 'n tell
/ #
TYPE-C
/dev/ttyUSB0 115200-8-N-1

Microchip PolarFire SoC FPGA

- Hard RISC-V cores with FPGA fabric, similar to Xilinx Zync for ARM. Coming 2nd half 2020.



PolarFire SoC Icicle Kit

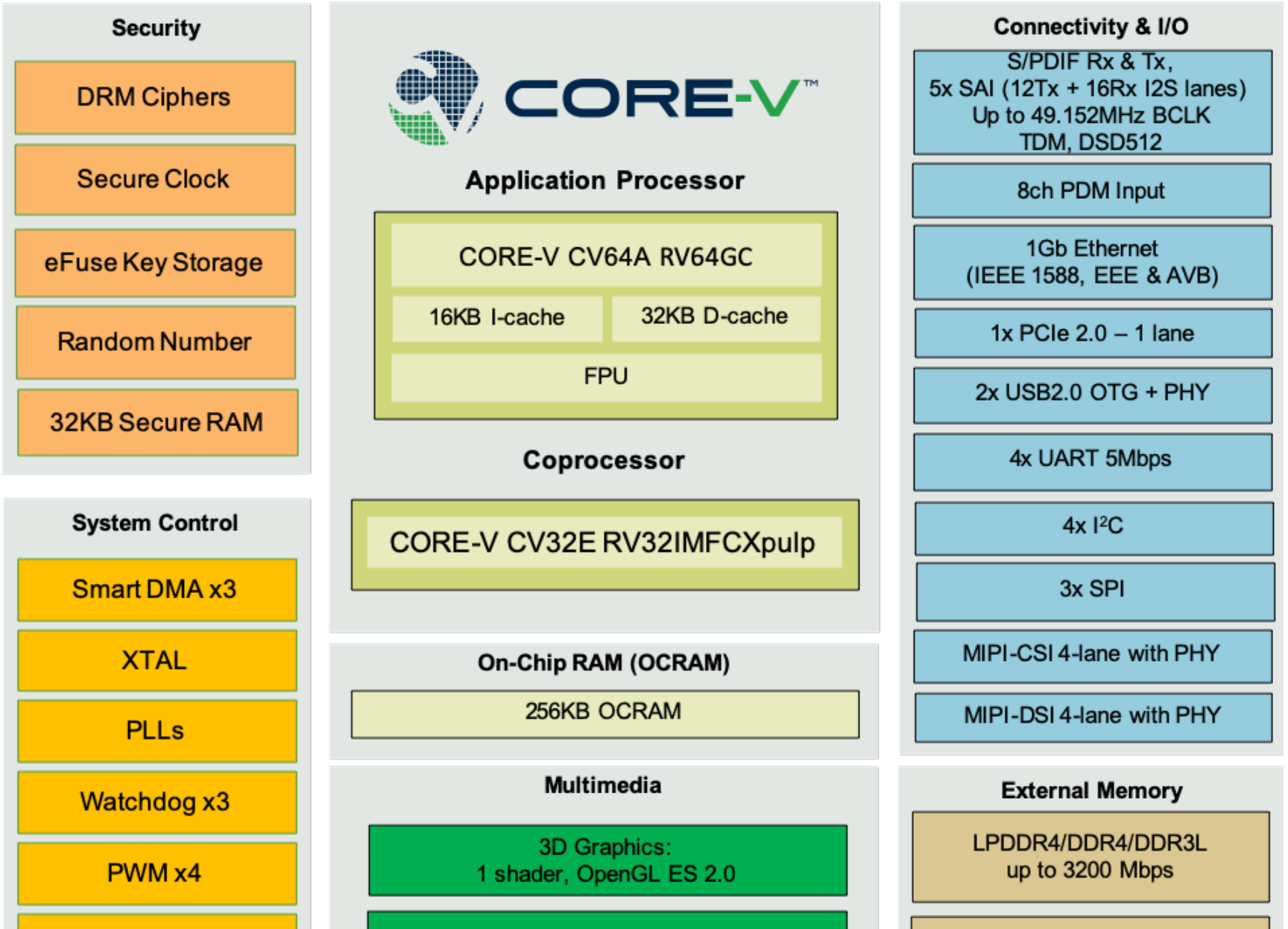
A low-cost dev kit for Microchip's PolarFire SoC, a low-power FPGA integrated with a complete hardened quad core 64-bit RISC-V processor

This project is coming soon. Sign up to receive updates and be notified when this project launches.

Subscribe to Updates

OpenHW Group: Core-V Chassis SoC

- similar to NXP iMX but with RISC-V cores
- tape-out 2nd half of 2020



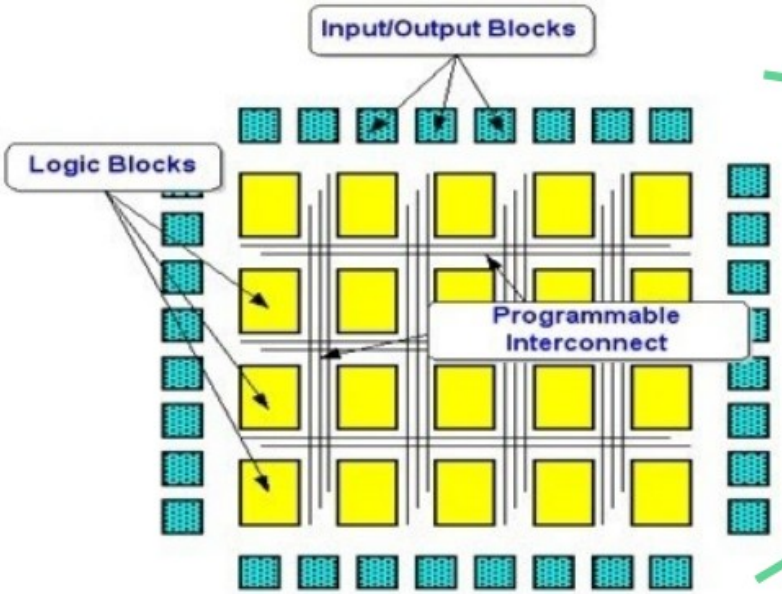
Section:

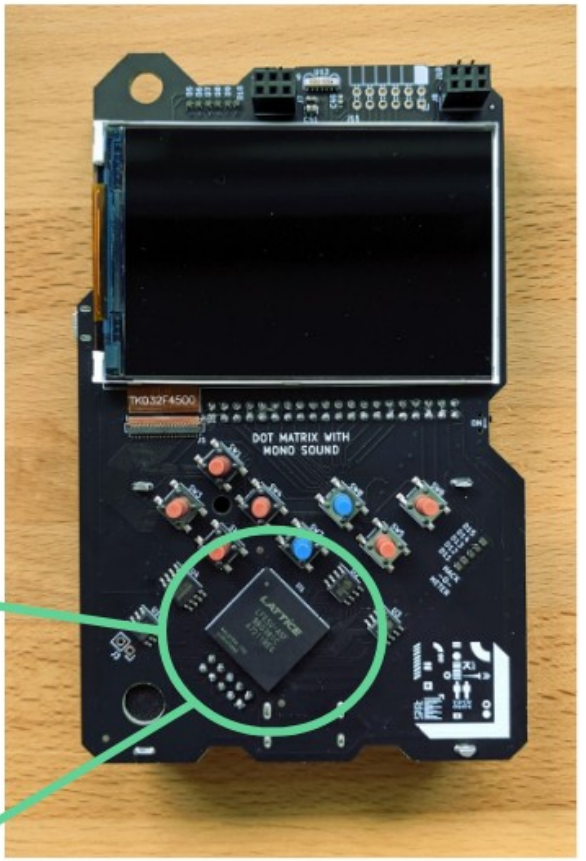
Open Source FPGA tools

- Keynote at Hackday Supercon 2019 by Dr. Megan Wachs of SiFive
- **“RISC-V and FPGAs: Open Source Hardware Hacking”**
  - https://www.youtube.com/watch?v=vCG5_nxm2G4

Where do FPGAs Come In?

- Field Programmable Gate Array
- Change a chip's HARDWARE in a few minutes
- Make it act like a new chip!





Open Source toolchains for FPGAs

- Project IceStorm for Lattice iCE40
  - “A Free and Open Source Verilog-to-Bitstream Flow for iCE40 FPGAs”
    by Claire Wolf (oe1cxw) at 32c3

media.ccc.de

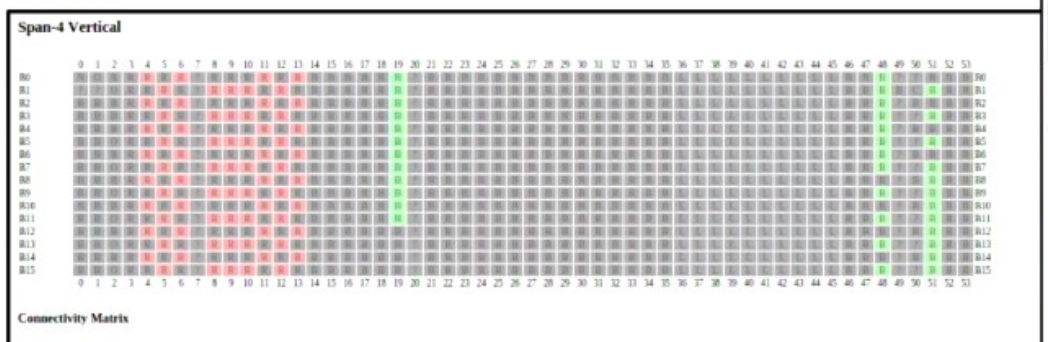
Search...

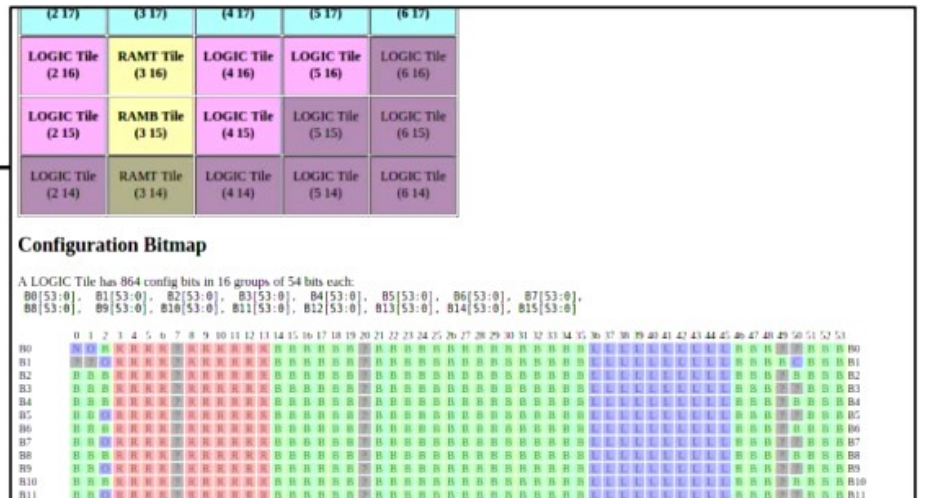
browse › congress › 2015 › event

A Free and Open Source Verilog-to-Bitstream Flow for iCE40 FPGAs

Clifford

Some screenshots from IceStrom Docs:



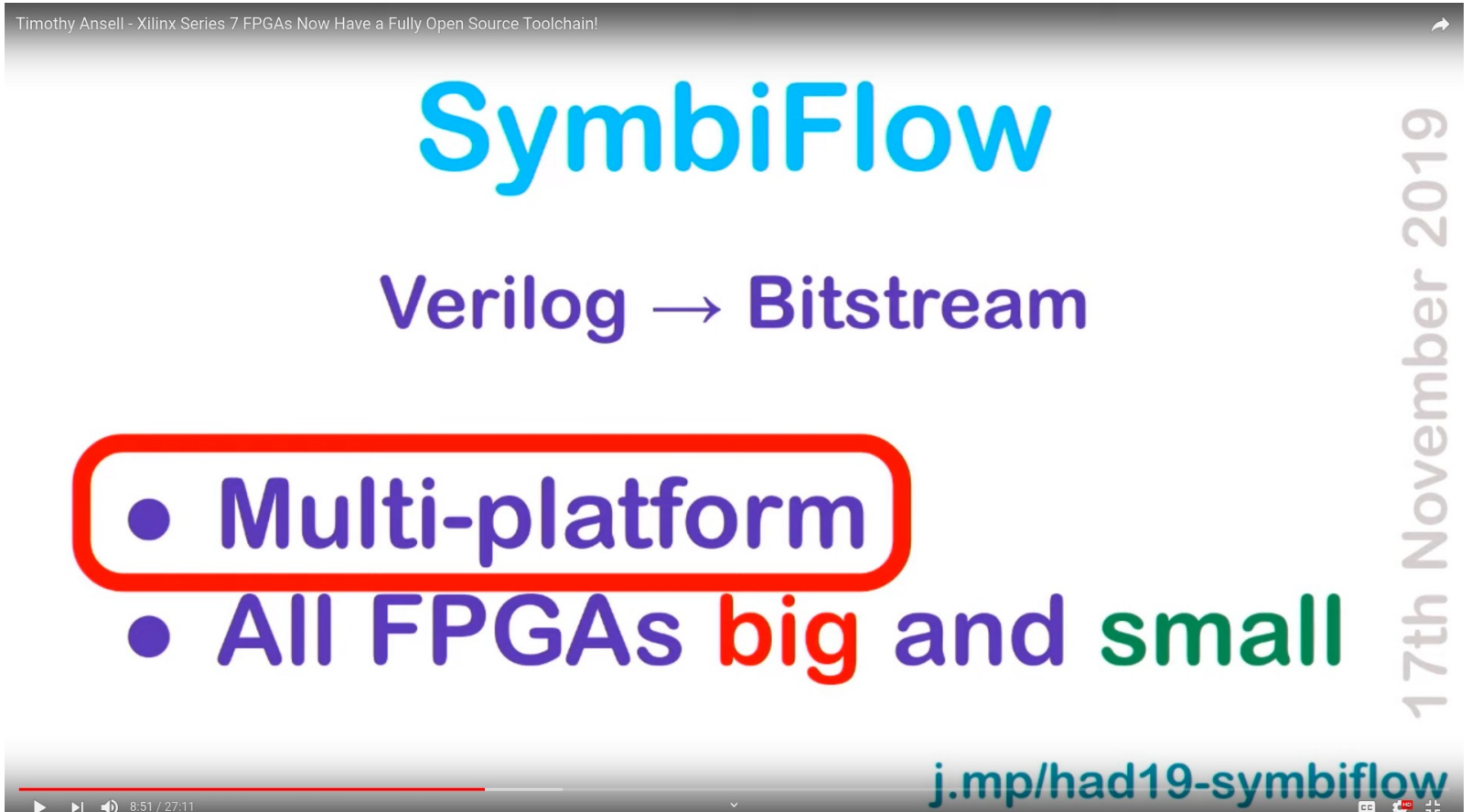


Open Source toolchains for FPGAs

- Project Trellis for Lattice ECP5
  - “Project Trellis and nextpnr FOSS FPGA flow for the Lattice ECP5”
    - David Shah (@fpga_dave)
      - youtube.com/watch?v=0se7kNes3EU

Open Source toolchains for FPGAs

- Project X-Ray & SymbiFlow for Xilinix Series 7
  - Timothy ‘mithro’ Ansell: “Xilinx Series 7 FPGAs Now Have a Fully Open Source Toolchain!” *(almost)*
    - youtube.com/watch?v=EHePto95qoE



Open Source and FPGAs

- Hackspace Magazine column about how about open source FPGA tools developed by Claire Wolf (oe1cxw), David Shah and others have made FPGAs more accessible than ever before to makers and hackers:
  - hackspace.raspberrypi.org/issues/26/

The rise of the FPGA

Reconfigure your chips to suit your project

Drew Fustini

@pdp7

Drew Fustini is a hardware designer and embedded Linux developer. He is the Vice President of the Open Source Hardware Association, and a board member of the BeagleBoard.org Foundation. Drew designs circuit boards for OSH Park, a PCB manufacturing service, and maintains the Adafruit BeagleBone Python library.

FPGAs have been the talk of the town at many of this year's hacker conferences. But what exactly is an FPGA, and why are they so hot right now?

FPGA stands for Field Programmable Gate Array, a digital logic chip that can be programmed to reconfigure the internal hardware. An FPGA does not run software – it physically changes the configuration of its gate arrays to adapt to the task at hand, making an FPGA an incredibly versatile tool. Need 25 PWM pins for a project? No problem. Want to replicate the functionality of a vintage CPU? Your FPGA has you covered. Not only is an FPGA versatile, but it is also better at handling timing-critical tasks than a microcontroller. You can filter high-speed sensor data before it's read by your processor, or offload repetitive tasks like debouncing buttons, reducing the burden on your microcontroller.

FPGAs are hot right now, but they're not a new technology – they've been used in industry for decades. However, pricey FPGA development boards, and the massive proprietary software suites needed to develop FPGA designs, means that there has historically been limited adoption in the maker community.

When you write an Arduino program, you're using a language called Wiring, a variation of C++. When you press compile, this high-level code is used to generate a binary file that the processor on the Arduino board executes to achieve the purpose of your program. Similarly, we describe the circuits we want in an FPGA using a high-level, text-based format known as a hardware description language (HDL), such as Verilog. The HDL design is then transformed by a synthesis tool into the basic building blocks that exist in the FPGA.

This process is normally done with proprietary software from the FPGA vendor, but these tools can take ages to download and devour disk space. That is, until Project IceStorm, led by Clifford Wolf, enabled a complete open-source workflow for the Lattice iCE40 FPGA.

FPGAs are hot right now but they're not a new technology – they've been used in industry for decades

This opened the door for low-cost, open hardware boards such as myStorm Blackice, TinyFPGA, iCEBreaker, and Fomu, which are great tools for teaching workshops and building projects.

The Lattice ECP5 FPGA is capable of more advanced features than the iCE40, and it now has an open-source workflow too, thanks to Project Trellis led by David Shah. This enabled the ECP5-powered Supercon badge to have cool features like HDMI video, while still being open for anyone to hack on without requiring proprietary tools.

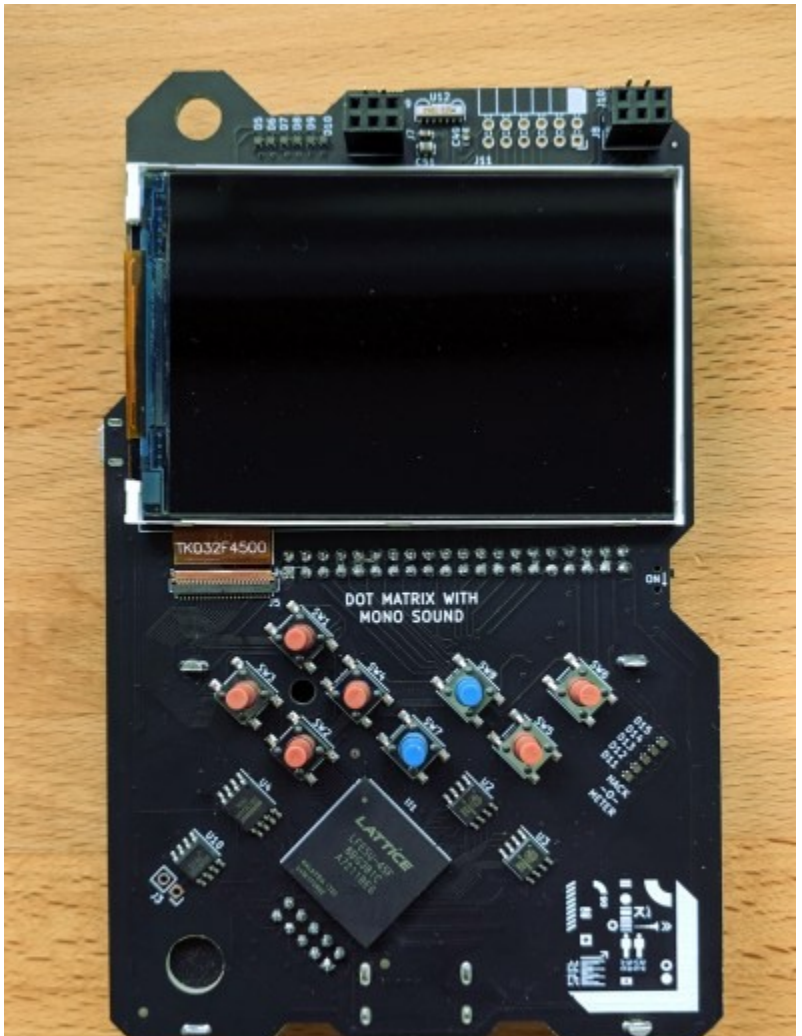
FPGAs are a fascinating technology with lots of awesome applications. If you want to find out more, start off by reading Luke Valenty's *The Hobbyists Guide to FPGAs* on **Hackaday.io** (**hsmag.ce/G0AQwR**), and watch Tim Ansell's Supercon talk to learn about the exciting future of open-source FPGA tools (**hsmag.cc/kyStPD**).

Slides: https://github.com/pdp7/talks/blob/master/rv-elc.pdf

Section:
Linux on the Hackaday Badge

Hackaday 2019 Supercon badge

- RISC-V “soft” core on ECP5 FPGA
- Gigantic FPGA In A Game Boy Form Factor



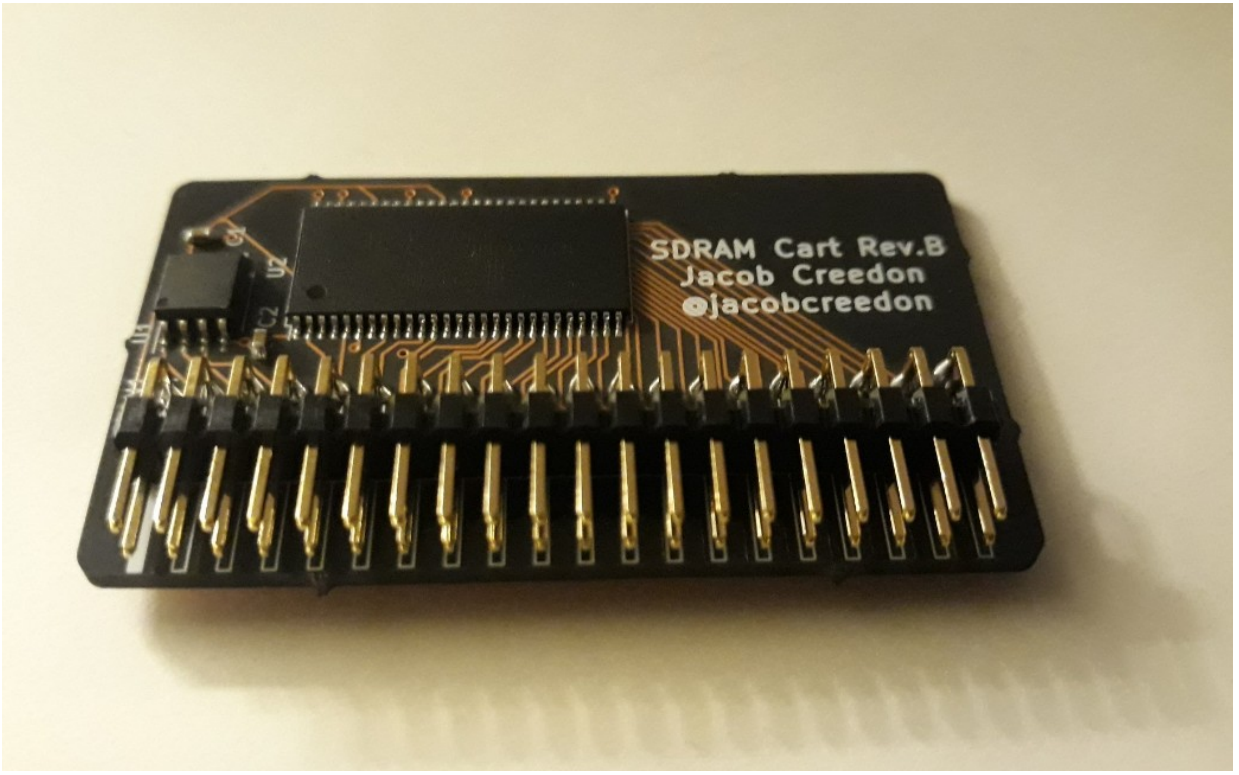
“Team Linux on Badge”

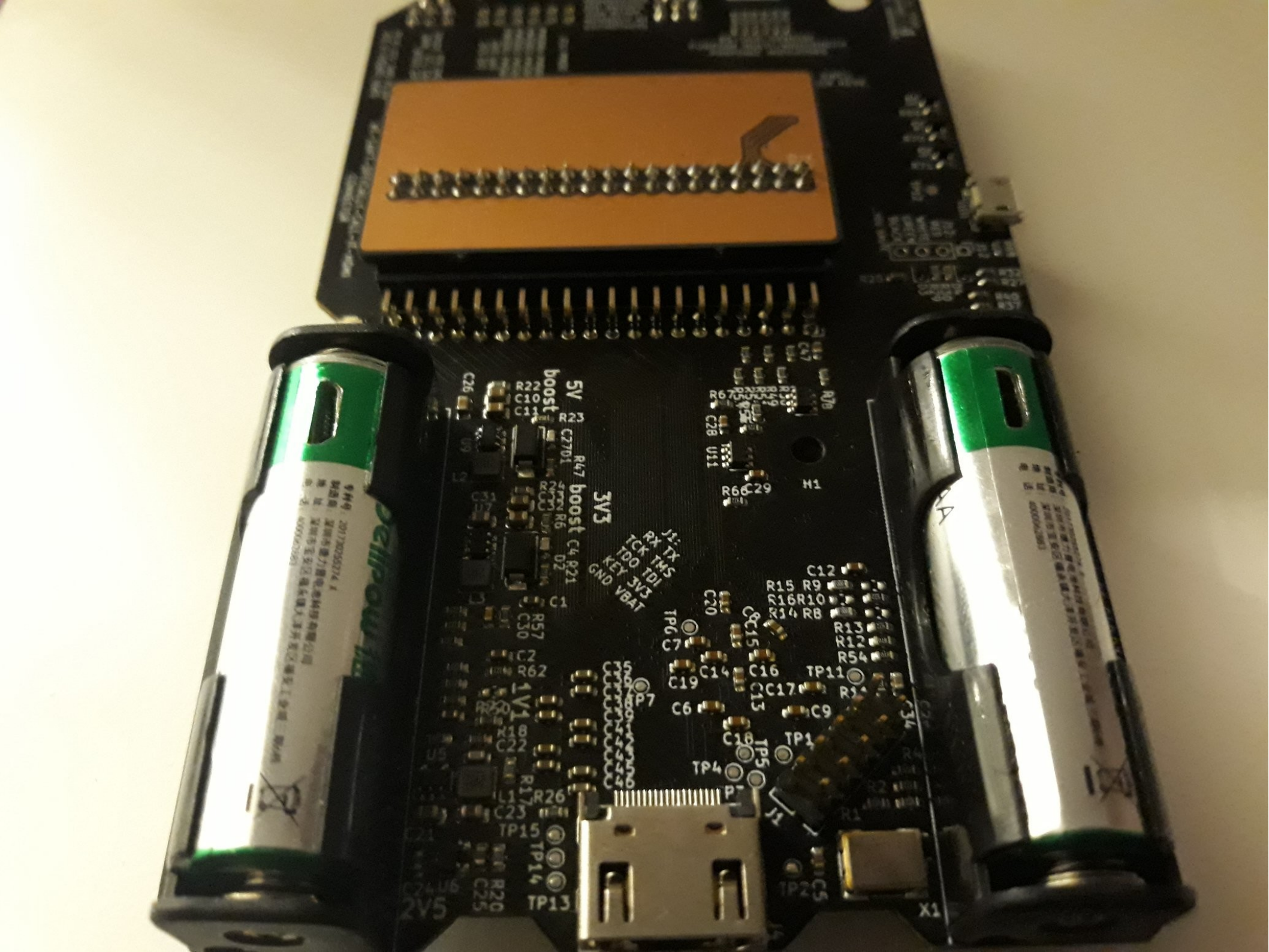
“Team Linux on Badge”

- Blog post: Hackaday Supercon badge boots Linux using SDRAM cartridge
- Michael Welling (@QwertyEmedded), Tim Ansell (@mithro), Sean Cross (@xobs), Jacob Creedon (@jacobcreedon)
- First attempt: use the built-in 16MB SRAM… no luck :(

“Team Linux on Badge”

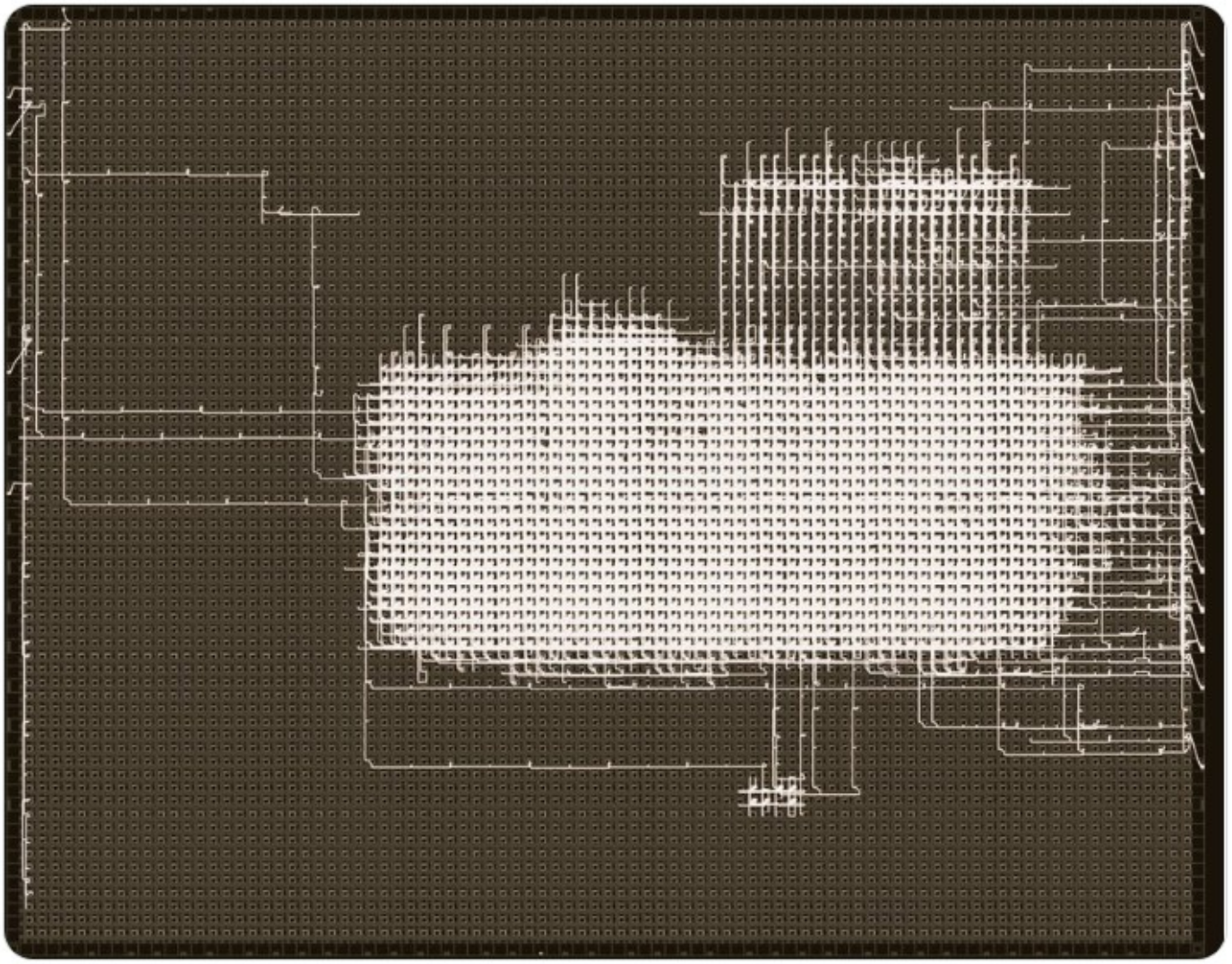
- Second attempt:
  - Jacob Creedon designed an a cartridge board that adds 32MB of SDRAM to the Hackaday Supercon badge… before the event!




5V boost
3V3
1V1
2V5
J1: TX RX TMS TCK TDI TDO KEY 3V3 GND VBAT
X1
H1

OSS FPGA & EDA tools
@ico_TC

This is how a Linux capable core looks like on an FPGA.#nextpnratwork

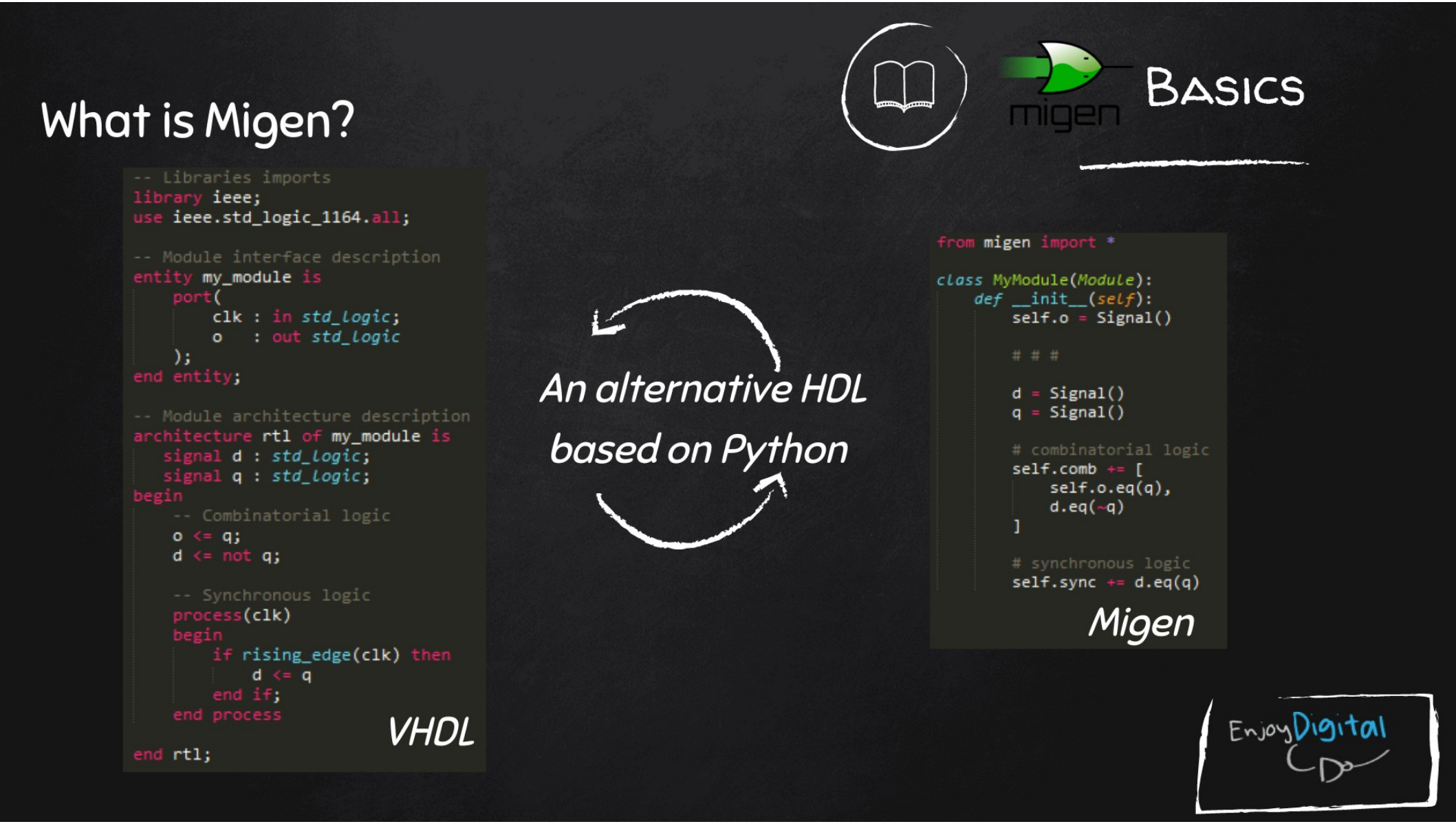


12:42 AM · Dec 15, 2019 · Twitter for Android

Designing Hardware in Python?

- Yes!
- “Using Python for creating hardware to record FOSS conferences!”
- Tim “mithro” Ansell
- youtube.com/watch?v=MkVX_mh5dOU

Slides: https://github.com/litex-hub/fpga_101

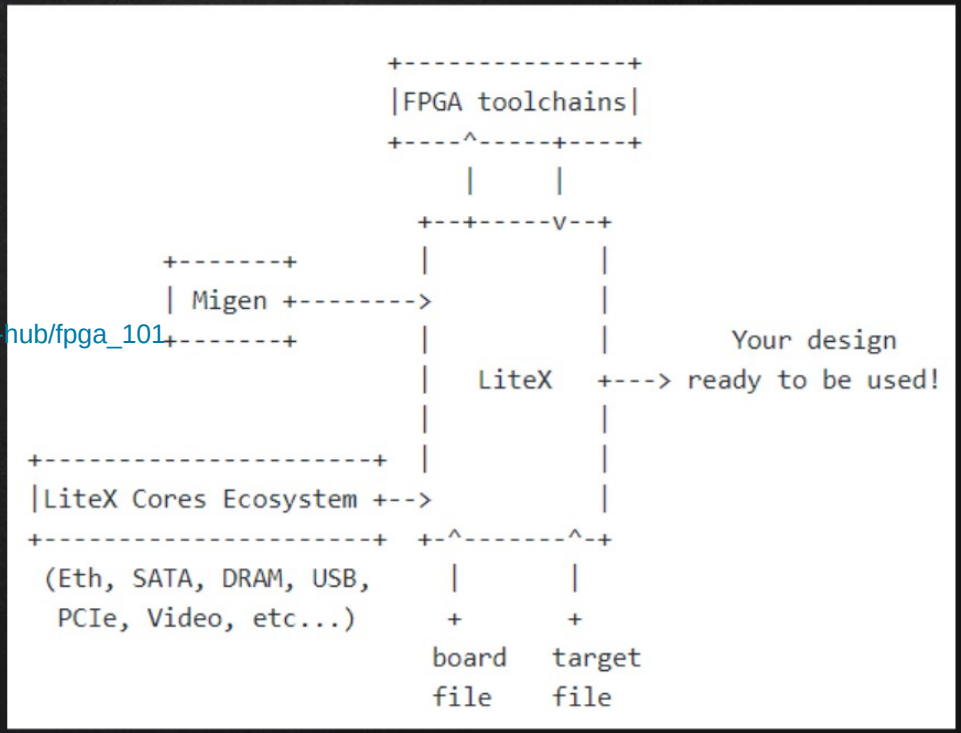


- LiteX used to build cores, create SoCs and full FPGA designs.
- LiteX is based on Migen
- Migen lets you do FPGA design in Python!
- https://github.com/enjoy-digital/litex

Slides: https://github.com/litex-hub/fpga_101

FPGAs for custom System on Chip

- *LiteX automates parts of the SoC design (buses, registers, software) and allow being more efficient.*
- *It provides most of the base elements required in a modern SoC (buses, streams, fifos, arbiters, muxes, etc...)*
- *It is compatible with all the LiteX core ecosystem (DRAM, Ethernet, PCIe, SATA, USB controllers...)*
- *Can be found at: github.com/enjoy-digital/litex*

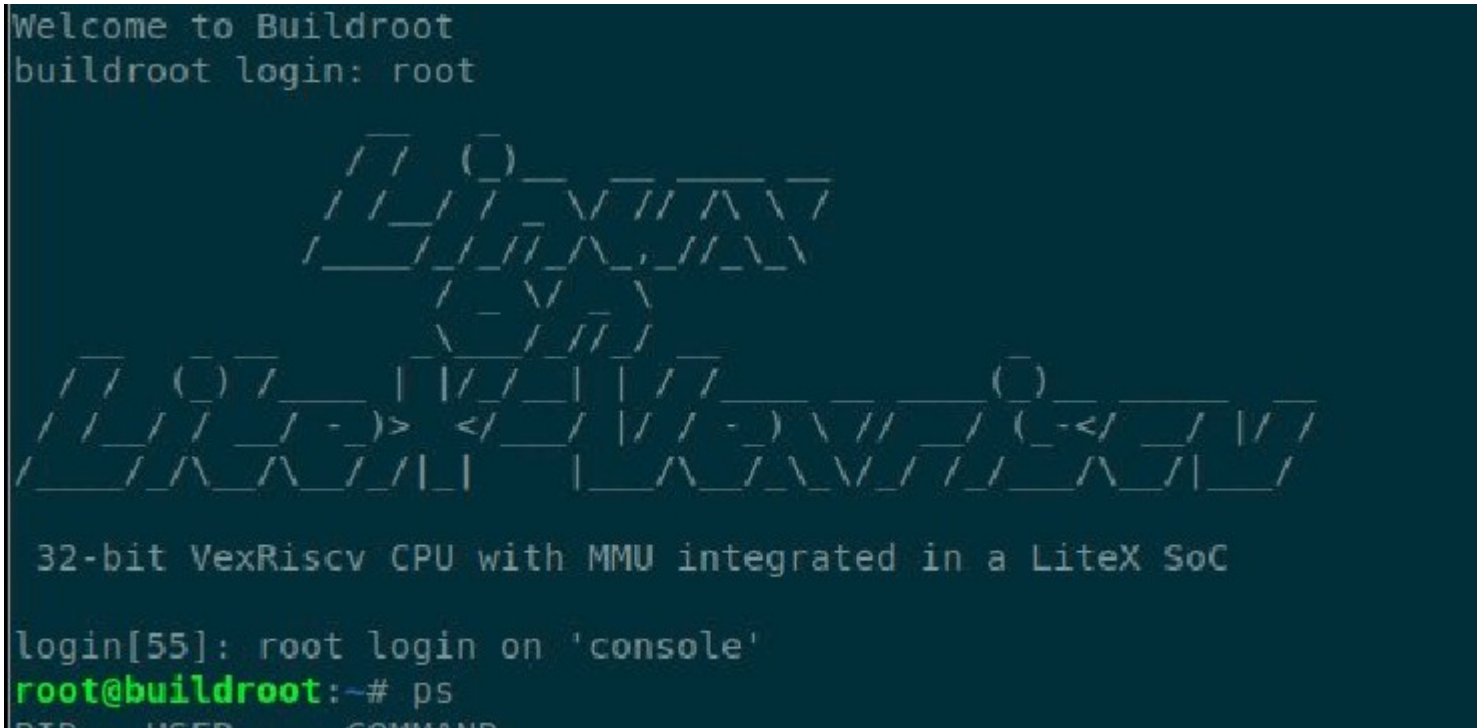


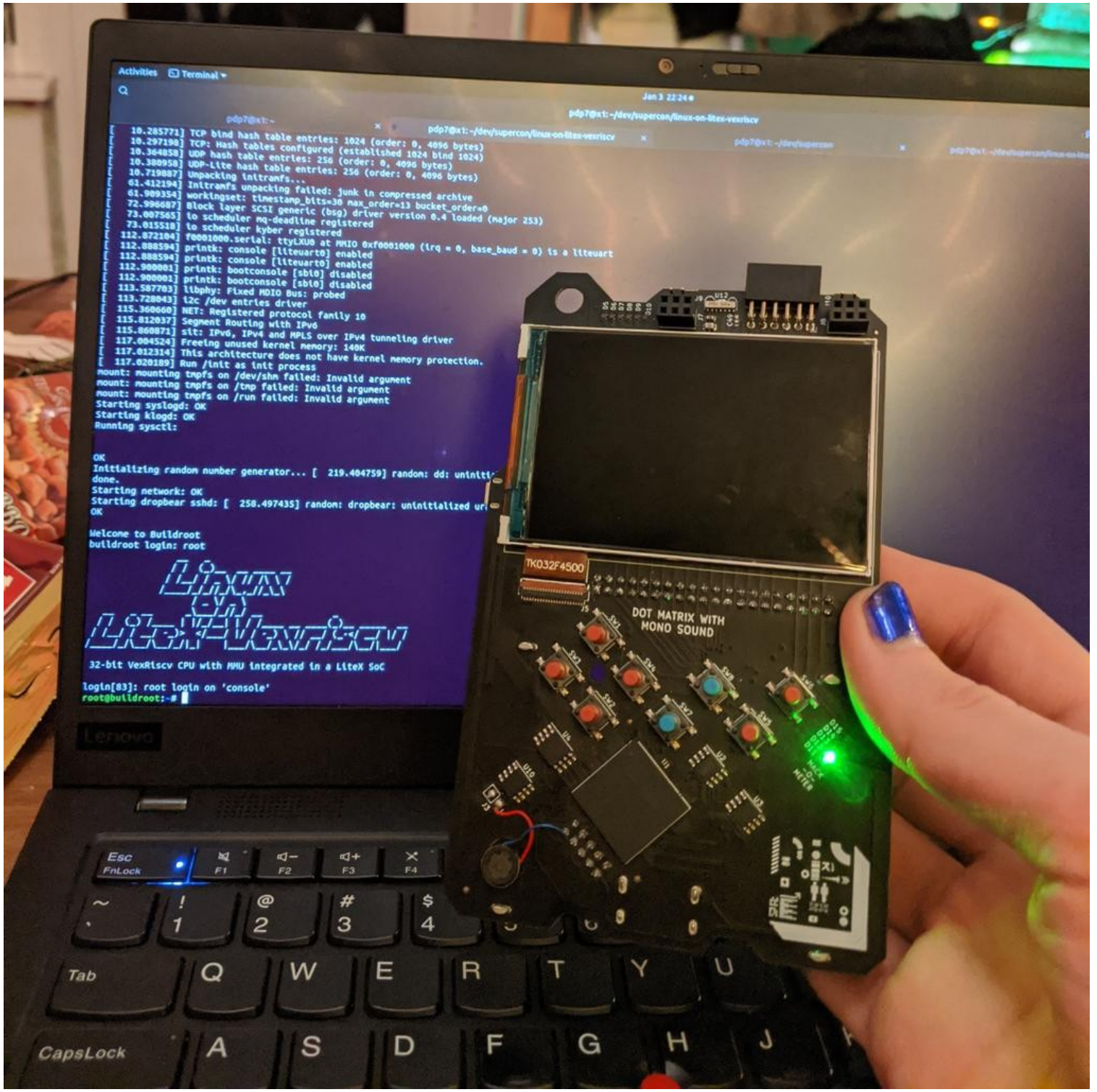
https://github.com/litex-hub/fpga_101

EnjoyDigital

Linux on LiteX-VexRiscv

- VexRiscv: 32-bit Linux Capable RISC-V CPU
- SoC built using VexRiscv core and LiteX moduels like LiteDRAM, LiteEth, LiteSDCard, ...
  - github.com/litex-hub/linux-on-litex-vexriscv




Activities
Terminal
Jan 3 22:24
pdp7@x1: ~/dev/supercon/linux-on-litex-vexriscv
[10.285771] TCP bind hash table entries: 1024 (order: 0, 4096 bytes)
[10.297198] TCP: Hash tables configured (established 1024 bind 1024)
[10.364858] UDP hash table entries: 256 (order: 0, 4096 bytes)
[10.380958] UDP-Lite hash table entries: 256 (order: 0, 4096 bytes)
[10.719887] Unpacking initramfs...
[61.412194] Initramfs unpacking failed: junk in compressed archive
[61.909354] workingset: timestamp_bits=30 max_order=13 bucket_order=0
[72.996687] Block layer SCSI generic (bsg) driver version 0.4 loaded (major 253)
[73.007565] io scheduler mq-deadline registered
[73.015518] io scheduler kyber registered
[112.872104] f0001000.serial: ttyLXU0 at MMIO 0xf0001000 (irq = 0, base_baud = 0) is a liteuart
[112.888594] printk: console [liteuart0] enabled
[112.888594] printk: console [liteuart0] enabled
[112.900001] printk: bootconsole [sbi0] disabled
[112.900001] printk: bootconsole [sbi0] disabled
[113.587703] libphy: Fixed MDIO Bus: probed
[113.728043] i2c /dev entries driver
[115.360660] NET: Registered protocol family 10
[115.812037] Segment Routing with IPv6
[115.860871] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
[117.004524] Freeing unused kernel memory: 140K
[117.012314] This architecture does not have kernel memory protection.
[117.020189] Run /init as init process
mount: mounting tmpfs on /dev/shm failed: Invalid argument
mount: mounting tmpfs on /tmp failed: Invalid argument
mount: mounting tmpfs on /run failed: Invalid argument
Starting syslogd: OK
Starting klogd: OK
Running sysctl:
OK
Initializing random number generator... [219.404759] random: dd: uninitialized
done.
Starting network: OK
Starting dropbear sshd: [258.497435] random: dropbear: uninitialized
OK
Welcome to Buildroot
buildroot login: root
Linux on LiteX-VexRiscv
32-bit VexRiscv CPU with MMU integrated in a LiteX SoC
login[83]: root login on 'console'
root@buildroot:~#
TK032F4500
DOT MATRIX WITH
MONO SOUND
Lenovo
Esc
FnLock
F1
F2
F3
F4
Tab
CapsLock

- upstream support for Hackaday Supercon badge:
  - https://github.com/litex-hub/litex-boards/pull/31

litex-hub / **litex-boards**

Unwatch 7 | Unstar 17 | Fork 24

Code | Issues 2 | Pull requests 1 | Actions | Projects 0 | Wiki | Security | Insights

add the Hackaday Supercon ECP5 badge #31

Edit

Merged **enjoy-digital** merged 1 commit into `litex-hub:master` from `pdp7:master` 21 days ago

Conversation 18 | Commits 1 | Checks 1 | Files changed 2 | +461 −0

pdp7 commented 22 days ago • edited — Contributor

Add the Hackaday Supercon 2019 badge which has an ECP5 FPGA.

These changes are from a fork by Michael Welling (**@mwelling**)

During Supercon, we tried two approaches:

- use the built-in 16MB QSPI SRAM
- use add-on cartiridge with 32MB SDRAM by Jacob Creedon

We were not able to get the QSPI SRAM working so I've removed those changes, and I have just added the changes that are needed to boot Linux with the 32MB SDRAM.

In addition to **@mwelling**, thank you to Jacob Creedon (**@jcreedon**), **@gregdavill**, Tim Ansell (**@mithro**), and Sean Cross (**@xobs**) who all helped get Linux working on this badge.

Reviewers

No reviews

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

- upstream support for Hackaday Supercon badge:
  - https://github.com/litex-hub/litex-boards/pull/31

Merged **add the Hackaday Supercon ECP5 badge** #31
Changes from **all commits** File filter... Jump to... Review changes

215 litex_boards/partner/platforms/hadbadge.py

```
@@ -0,0 +1,215 @@
+ from litex.build.generic_platform import *
+ from litex.build.lattice import LatticePlatform
+
+ # IOs ----------------------------------------------------------------------------------------------
+
+ _io = [
+     ("clk8", 0, Pins("U18"), IOStandard("LVCMOS33")),
+     ("programn", 0, Pins("R1"), IOStandard("LVCMOS33")),
+     ("serial", 0,
+         Subsignal("rx", Pins("U2"), IOStandard("LVCMOS33"), Misc("PULLMODE=UP")),
+         Subsignal("tx", Pins("U1"), IOStandard("LVCMOS33")),
+     ),
+     ("led", 0, Pins("E3 D3 C3 C4 C2 B1 B20 B19 A18 K20 K19"), IOStandard("LVCMOS33")),  # Anodes
+     ("led", 1, Pins("P19 L18 K18"), IOStandard("LVCMOS33")), # Cathodes via FET
+     ("usb", 0,
+         Subsignal("d_p", Pins("F3")),
+         Subsignal("d_n", Pins("G3")),
+         Subsignal("pullup", Pins("E4")),
+         Subsignal("vbusdet", Pins("F4")),
+         IOStandard("LVCMOS33")
+     ),
+     ("keypad", 0,
+         Subsignal("left", Pins("G2"), Misc("PULLMODE=UP")),
```

- upstream support for Hackaday Supercon badge:
  - https://github.com/litex-hub/litex-boards/pull/31

Merged **add the Hackaday Supercon ECP5 badge** #31
Changes from **all commits** File filter... Jump to... Review changes

246 litex_boards/partner/targets/hadbadge.py Viewed

```
@@ -0,0 +1,246 @@
+ #!/usr/bin/env python3
+
+ # This file is Copyright (c) 2018-2019 Florent Kermarrec <florent@enjoy-digital.fr>
+ # This file is Copyright (c) 2018 David Shah <dave@ds0.me>
+ # License: BSD
+
+ import argparse
+ import sys
+
+ from migen import *
+ from migen.genlib.resetsync import AsyncResetSynchronizer
+
+ from litex_boards.platforms import hadbadge
+
+ from litex.soc.cores.clock import *
+ from litex.soc.integration.soc_sdram import *
+ #from litex.soc.integration.soc_core import *
+ from litex.soc.integration.builder import *
+
+ #from .spi_ram_dual import SpiRamDualQuad
+
+ from litedram import modules as litedram_modules
+ from litedram.phy import GENSDRPHY
```

Merged **add the Hackaday Supercon ECP5 badge** #68

Changes from **all commits** ▾ **File filter...** ▾ **Jump to...** ▾ ⚙ ▾

13 make.py

```
@@ -160,6 +160,16 @@ def __init__(self):
160 160         def load(self):
161 161             os.system("ujprog build/ulx3s/gateware/top.svf")
162 162
163     ++ # HADBadge support ---------------------------------------------------------------------
164     + +
165     + + class HADBadge(Board):
166     + +     def __init__(self):
167     + +         from litex_boards.targets import hadbadge
168     + +         Board.__init__(self, hadbadge.BaseSoC, {"serial"})
169     + +
170     + +     def load(self):
171     + +         os.system("dfu-util --alt 2 --download build/hadbadge/gateware/top.bit --reset")
172     + +
163 173   # OrangeCrab support ---------------------------------------------------------------------
164 174
165 175   class OrangeCrab(Board):
@@ -209,6 +219,7 @@ def load(self):
209 219         # Lattice
```

add 32MB SDRAM for hadbadge #97

Merged **enjoy-digital** merged 1 commit into `enjoy-digital:master` from `pdp7:master` 21 days ago

Conversation 2 | Commits 1 | Checks 0 | Files changed 1

pdp7 commented 22 days ago — Contributor

Add AS4C32M8SA-7TCN 32MB SDRAM used on cartridge PCB by Jacob Creedon (**@jcreedon**) for the Hackaday Supercon 2019 badge which has an ECP5 FPGA.

These changes are from a fork by Michael Welling (**@mwelling**)

In addition to **@mwelling**, thank you to Jacob Creedon (**@jcreedon**), **@gregdavill**, Tim Ansell (**@mithro**), and Sean Cross (**@xobs**) who all helped get Linux working on this badge.

KiCad design files by **@jcreedon** for the SDRAM cartridge are available on GitHub.

There is also a shared project to order the SDRAM cartridge PCB.

Refer to my blog post for more information

Merged

add 32MB SDRAM for hadbadge #97

Changes from **all commits** File filter... Jump to...

9 litedram/modules.py

```
@@ -190,6 +190,15 @@ class AS4C32M16(SDRAMModule):
      technology_timings = _TechnologyTimings(tREFI=64e6/8192, tWTR=(2, None), tCCD=(1,
      speedgrade_timings = {"default": _SpeedgradeTimings(tRP=18, tRCD=18, tWR=12, tRFC=

+ class AS4C32M8(SDRAMModule):
+     memtype = "SDR"
+     # geometry
+     nbanks = 4
+     nrows  = 8192
+     ncols  = 1024
+     # timings
+     technology_timings = _TechnologyTimings(tREFI=64e6/8192, tWTR=(2, None), tCCD=(1,
+     speedgrade_timings = {"default": _SpeedgradeTimings(tRP=20, tRCD=20, tWR=15, tRFC=

  # DDR ----------------------------------------------------------------------------
```

ProTip! Use n and p to navigate between commits in a pull request

optimize performance on Hackaday Badge #35

Edit | New issue

Open **pdp7** opened this issue 17 days ago · 7 comments

pdp7 commented 17 days ago — Contributor

Related to comments in PR #31 (comment)

The performance of the ECP5 Hackaday Badge with 32MB SDRAM is "painfully" slow.

@mithro suggested there could be some issue with the configuration.

@enjoy-digital has attempted some optimizations:
#31 (comment)

> With enjoy-digital/litedram@ `34e6c24` and enjoy-digital/litex@ `fa22d6a` we have a ~10% boot time speedup on designs using SDRAM:
>
> - De0Nano: 94.6.s to 84.6s
> - ULX3S: 75.9s to 68.2s
>
> On Arty with DDR3 the gain is effect more limited: 8.7s to 8.4s. That would be interesting to test this on the badge.

I will measure if the boot time improve

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications Customize

Unsubscribe

You're receiving notifications because you're watching this repository.

Open

optimize performance on Hackaday Badge #35

pdp7 opened this issue 17 days ago · 7 comments

pdp7 commented 15 days ago • edited

Contributor Author

@enjoy-digital WOW! much faster! It gets to login in 28 seconds (previous version was 258 seconds).

Recording:
https://asciinema.org/a/Pcm3vd1BEdEKY9srYX6MsNfCE

Text:

```
pdp7@x1:~/dev/enjoy/linux-on-litex-vexriscv$ lxterm --images=images.json /dev
[LXTERM] Starting....
lBIOS CRC passed (561ab1e2)

 Migen git sha1: 063188e
 LiteX git sha1: --------

--========== SoC ==========--
CPU:       VexRiscv @ 48MHz
ROM:       32KB
SRAM:      4KB
```

enjoy-digital committed 15 days ago 1 parent 2317519 commit 39ce39a298f5

Showing **1 changed file** with **5 additions** and **3 deletions**.

8 litex/soc/integration/soc_sdram.py

```
@@ -26,12 +26,13 @@ class SoCSDRAM(SoCCore):
        }
        csr_map.update(SoCCore.csr_map)

-       def __init__(self, platform, clk_freq, l2_size=8192, **kwargs):
+       def __init__(self, platform, clk_freq, l2_size=8192, l2_data_width=128, **kwargs):
            SoCCore.__init__(self, platform, clk_freq, **kwargs)
            if not self.integrated_main_ram_size:
                if self.cpu_type is not None and self.csr_data_width > 32:
                    raise NotImplementedError("BIOS supports SDRAM initialization only for c
-           self.l2_size = l2_size
+           self.l2_size       = l2_size
+           self.l2_data_width = l2_data_width

            self._sdram_phy    = []
            self._wb_sdram_ifs = []
```

- Greg Davill got the screen working with LiteVideo!
  - twitter.com/GregDavill/status/1231082623633543168

Slides: https://github.com/pdp7/talks/blob/master/rv-elc.pdf

Open Source boards with ECP5 FPGA *(can run Linux)*

Open Source ECP5 boards

- Radiona.org ULX3S
  - https://www.crowdsupply.com/radiona/ulx3s

crowdsupply.com/radiona/ulx3s

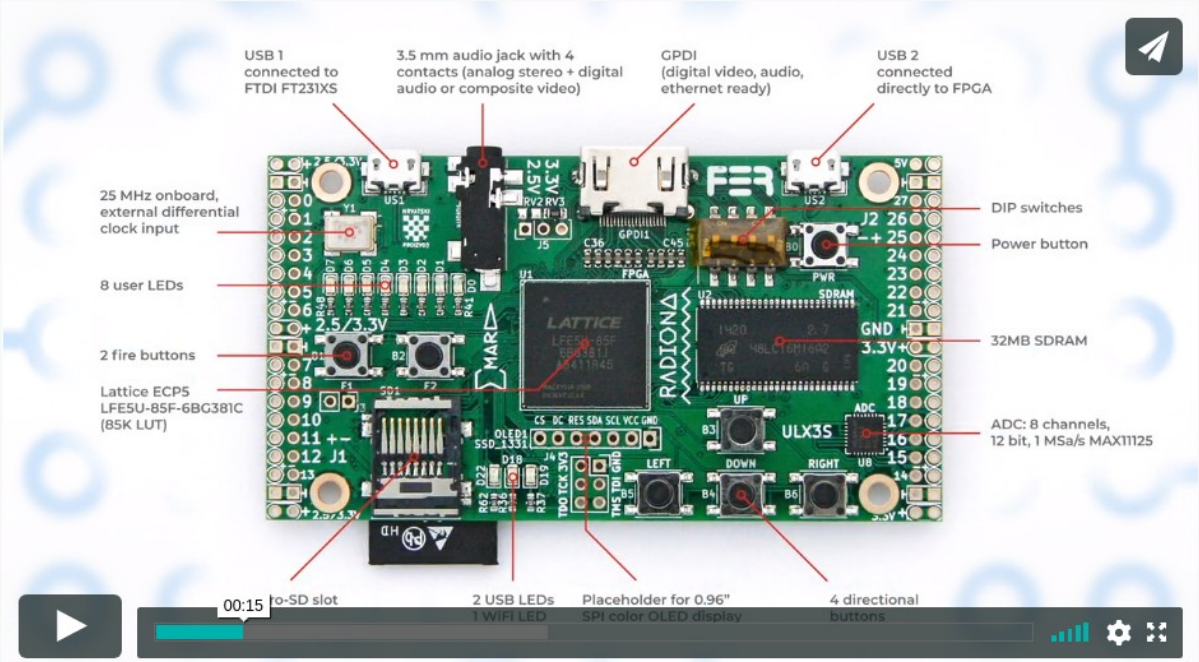
CROWD SUPPLY BROWSE LAUNCH ABOUT US Search

Open Hardware
Development Kits

ULX3S

by Radiona.org / Zagreb Makerspace

A powerful, open hardware ECP5 FPGA dev board



$50,793 raised
of $15,000 goal

338% Funded!

5	8	300
updates	days left	backers

Last update posted Apr 03, 2020

me@example.com Subscribe to Updates

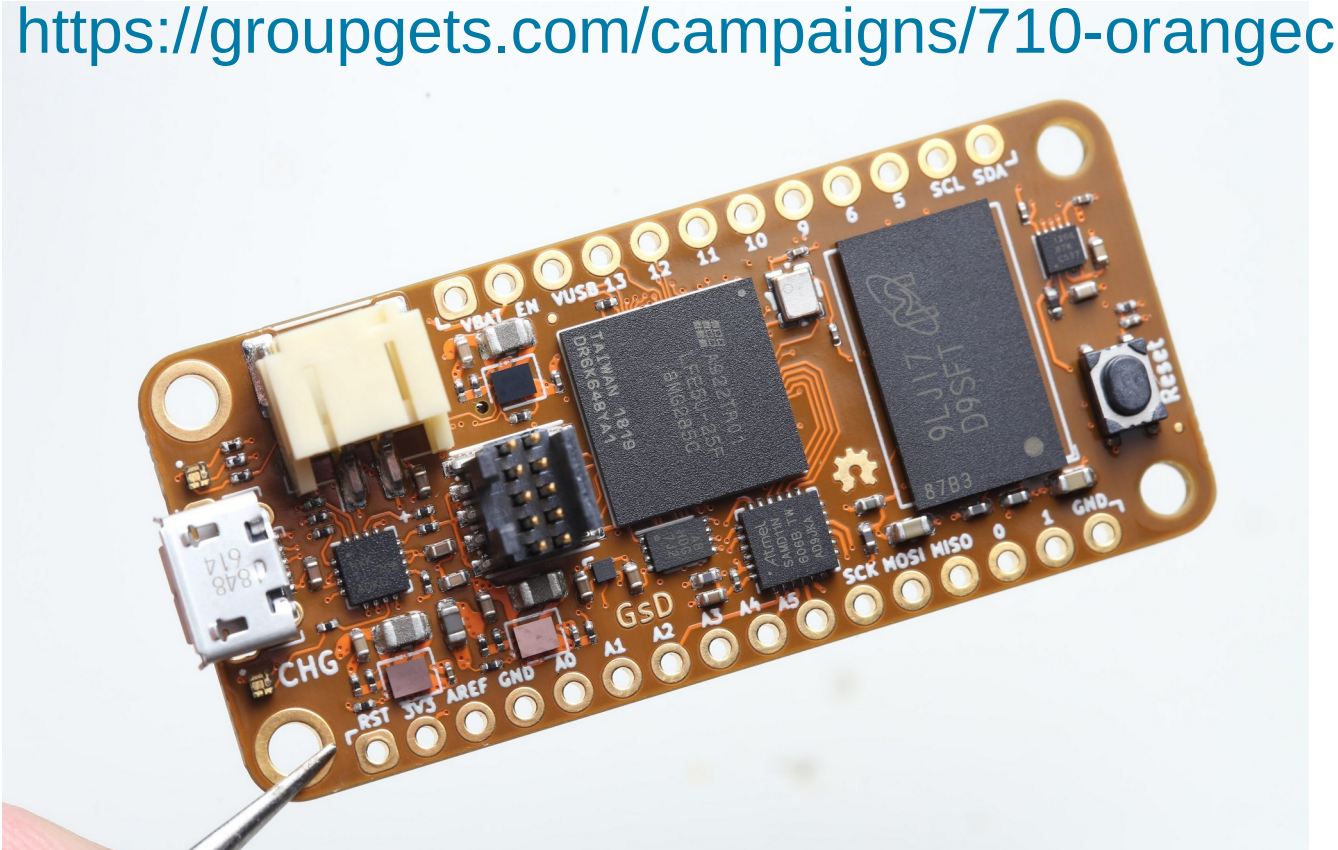
Recent Updates View all 5 updates.

PMOD Set $36

A set of three PMODs for ULX3X: one dual-USB board with support for IR send and receive, one Digital Video board for HDMI input or as a

Open Source ECP5 boards

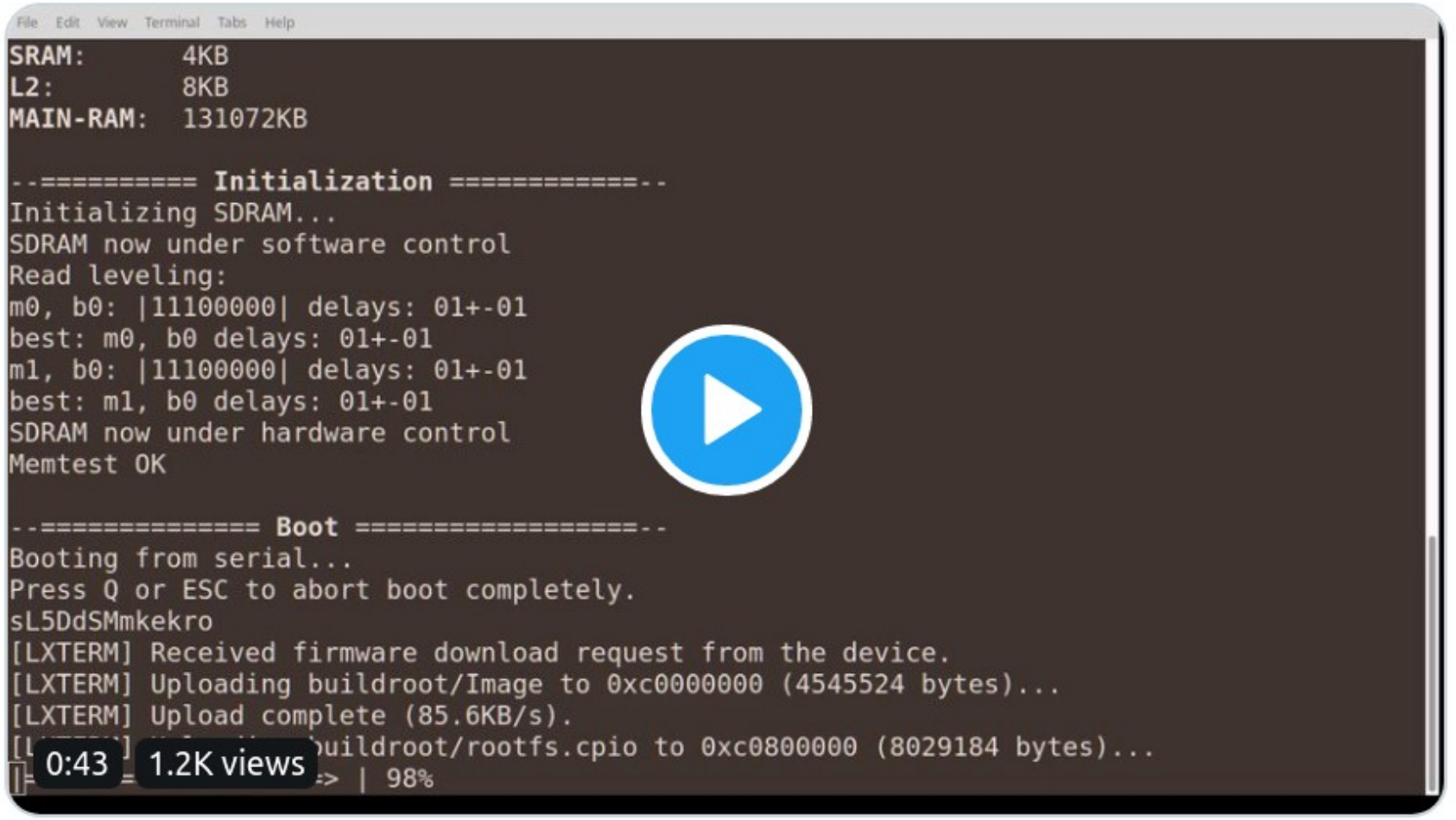
- Lattice ECP5 FPGA in Adafruit Feather form factor and 128MB DDR RAM:
  - Orange Crab by Greg Davill
    - https://github.com/gregdavill/OrangeCrab
    - https://groupgets.com/campaigns/710-orangecrab



Greg @ #36c3
@GregDavill

Replying to @mithro @pdp7 and 2 others

Done. 😃

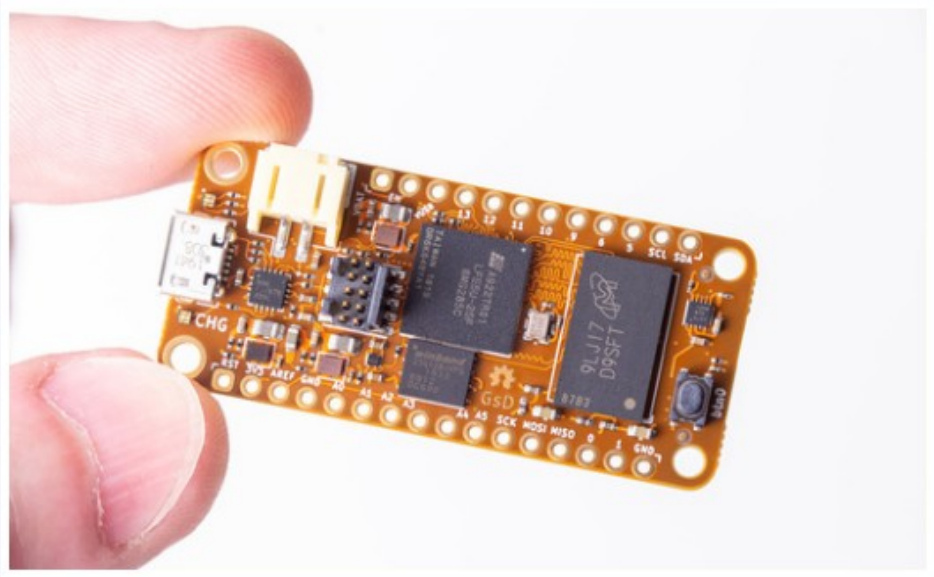


4:55 PM · Dec 28, 2019 · Twitter Web App

Open Source ECP5 boards

- Orange Crab by Greg Davill
  - https://groupgets.com/campaigns/710-orangecrab

GroupGets Campaigns Manufacturers Store Blog About Create Campaign Sign In



OrangeCrab

Single-unit price: **$99.00** + shipping

Campaign scheduled end date: Mon, 13 Apr 2020 16:19:00 PDT

Backers: 174

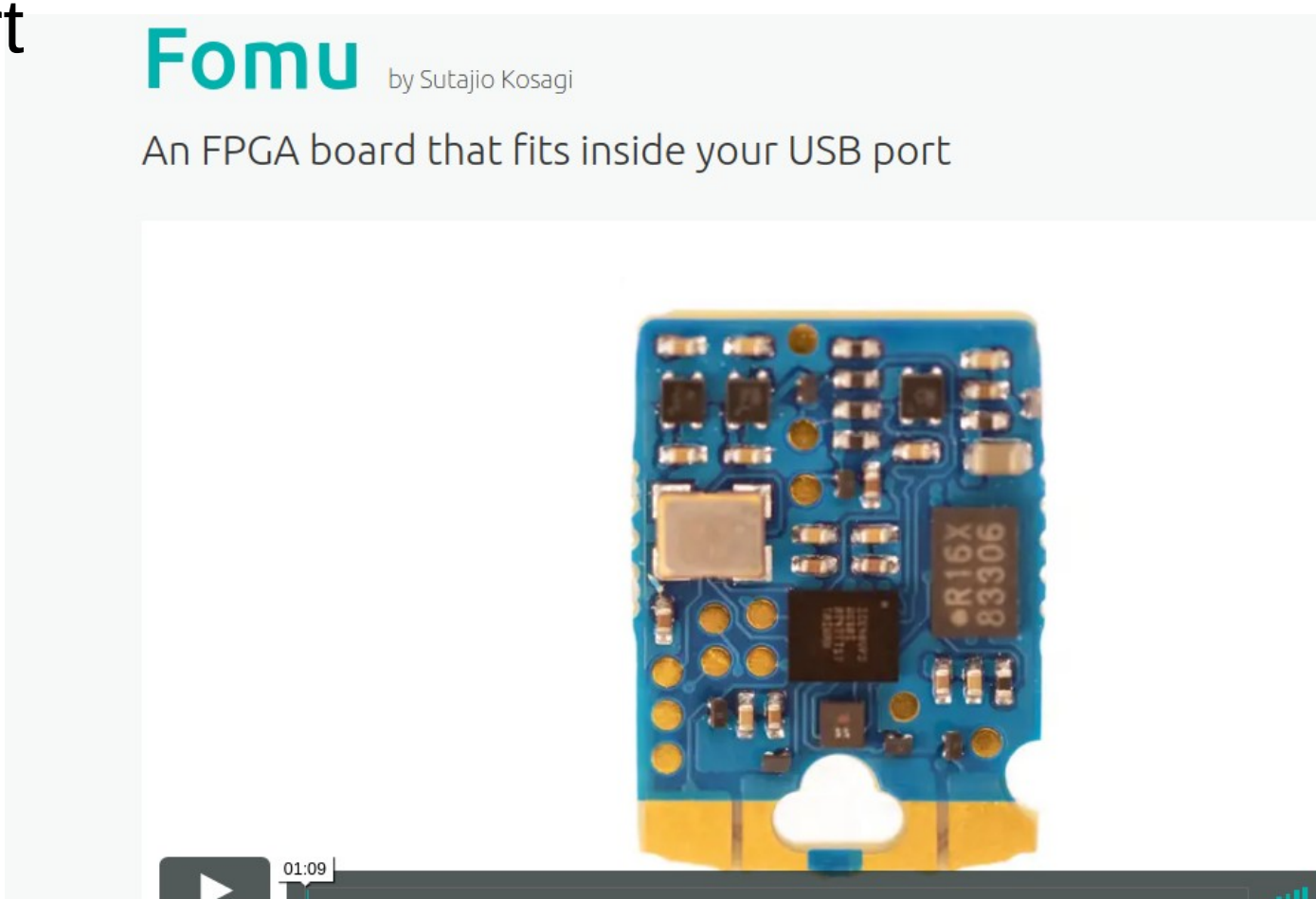
Target: 200

Funded: 286

Join this Buy

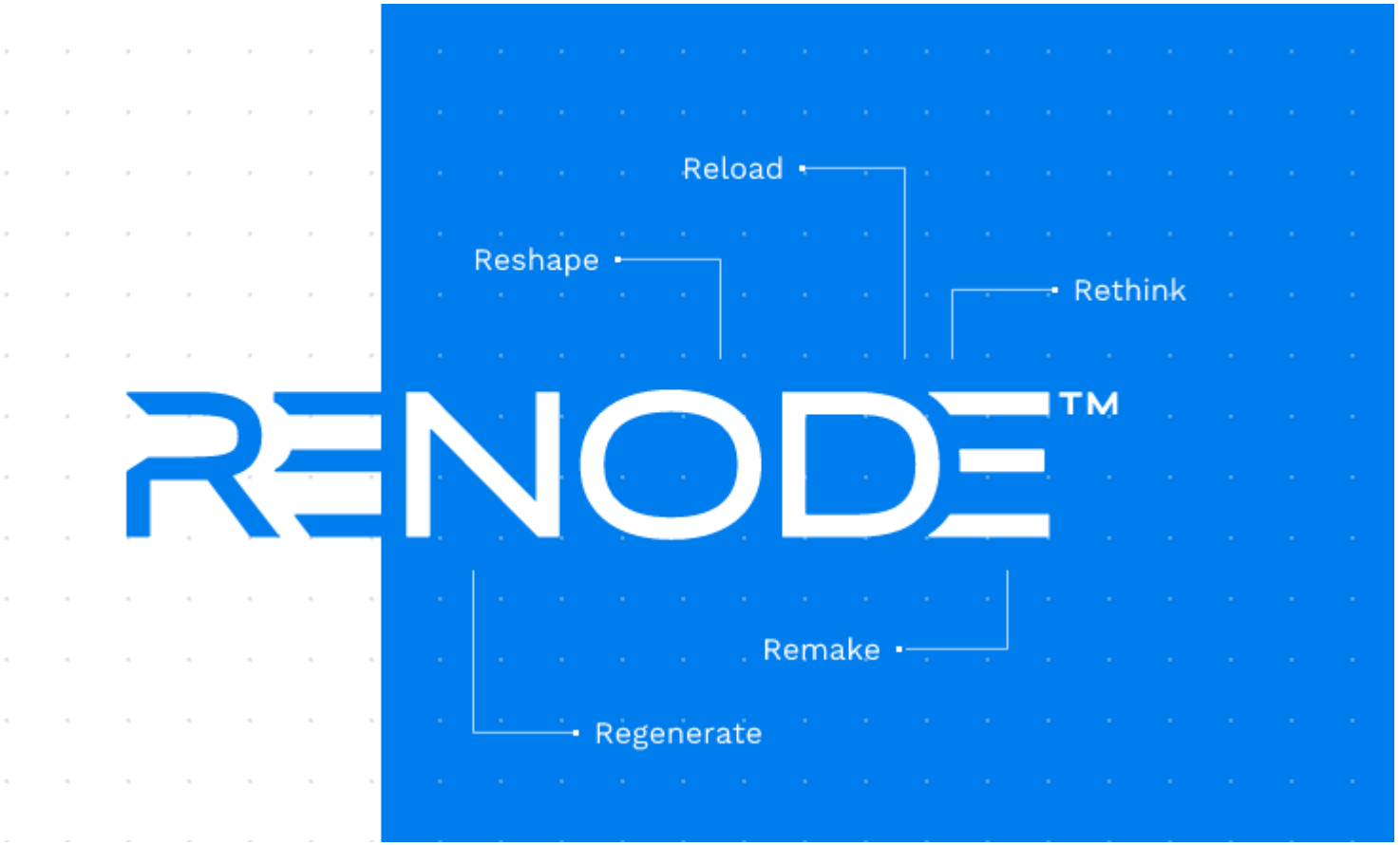
Want to learn FPGAs? Try Fomu!

- workshop.fomu.im
- crowdsupply.com/sutajio-kosagi/fomu
- Fits in USB port
- RGB LED
- Learn:
  - MicroPython
  - Verilog
  - LiteX



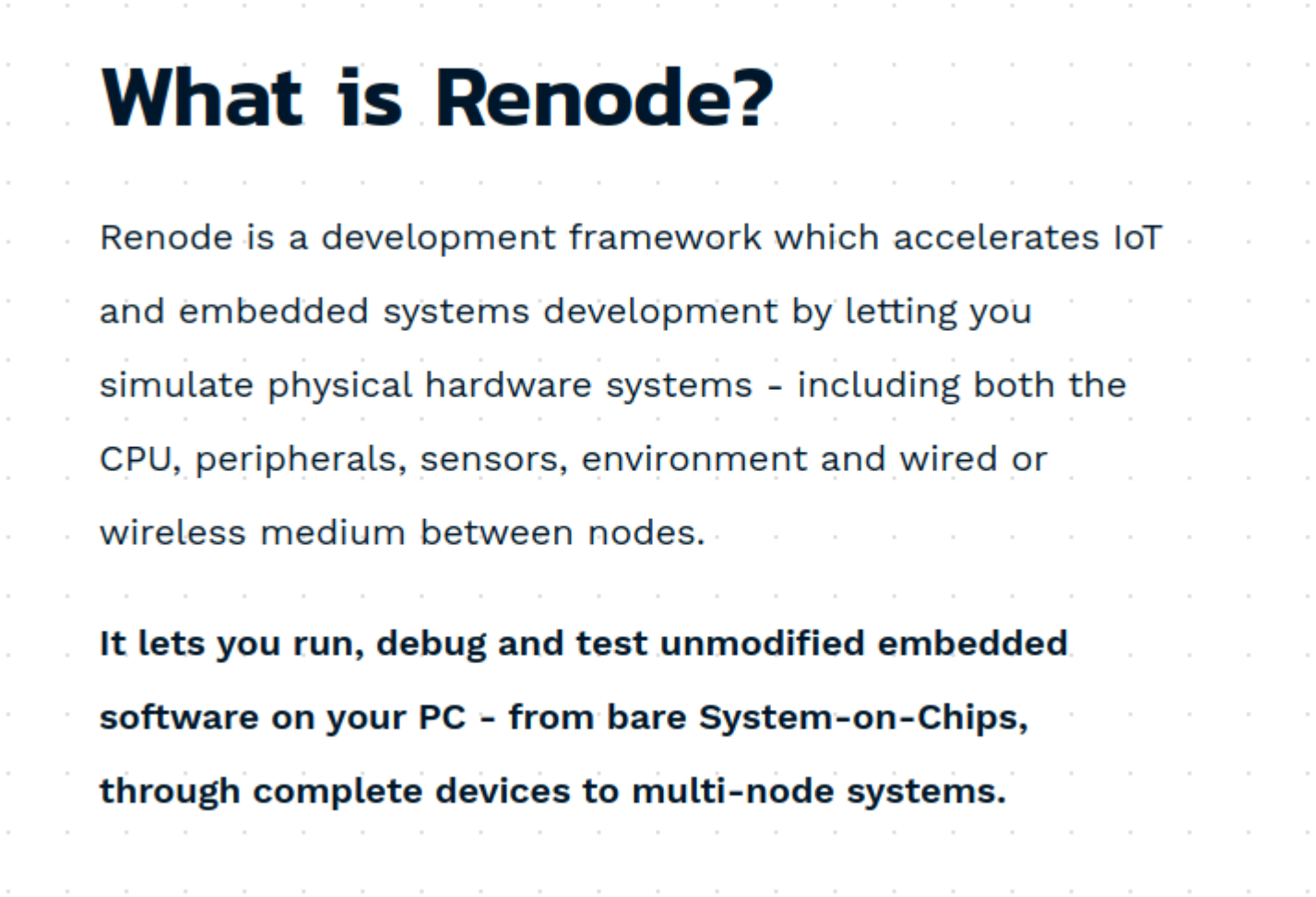
No hardware? Try Renode!

- https://renode.io/



No hardware? Try Renode!

- https://renode.io/



SiFive HiFive1

single-node/sifive_fe310.resc

SiFive HiFive Unleashed

single-node/hifive_unleashed.resc

Microchip PolarFire SoC Hardware Development Platform

single-node/polarfire-soc.resc

Toradex Colibri T30

single-node/tegra3.resc

OpenISA VEGAboard

single-node/vegaboard_ri5cy.resc

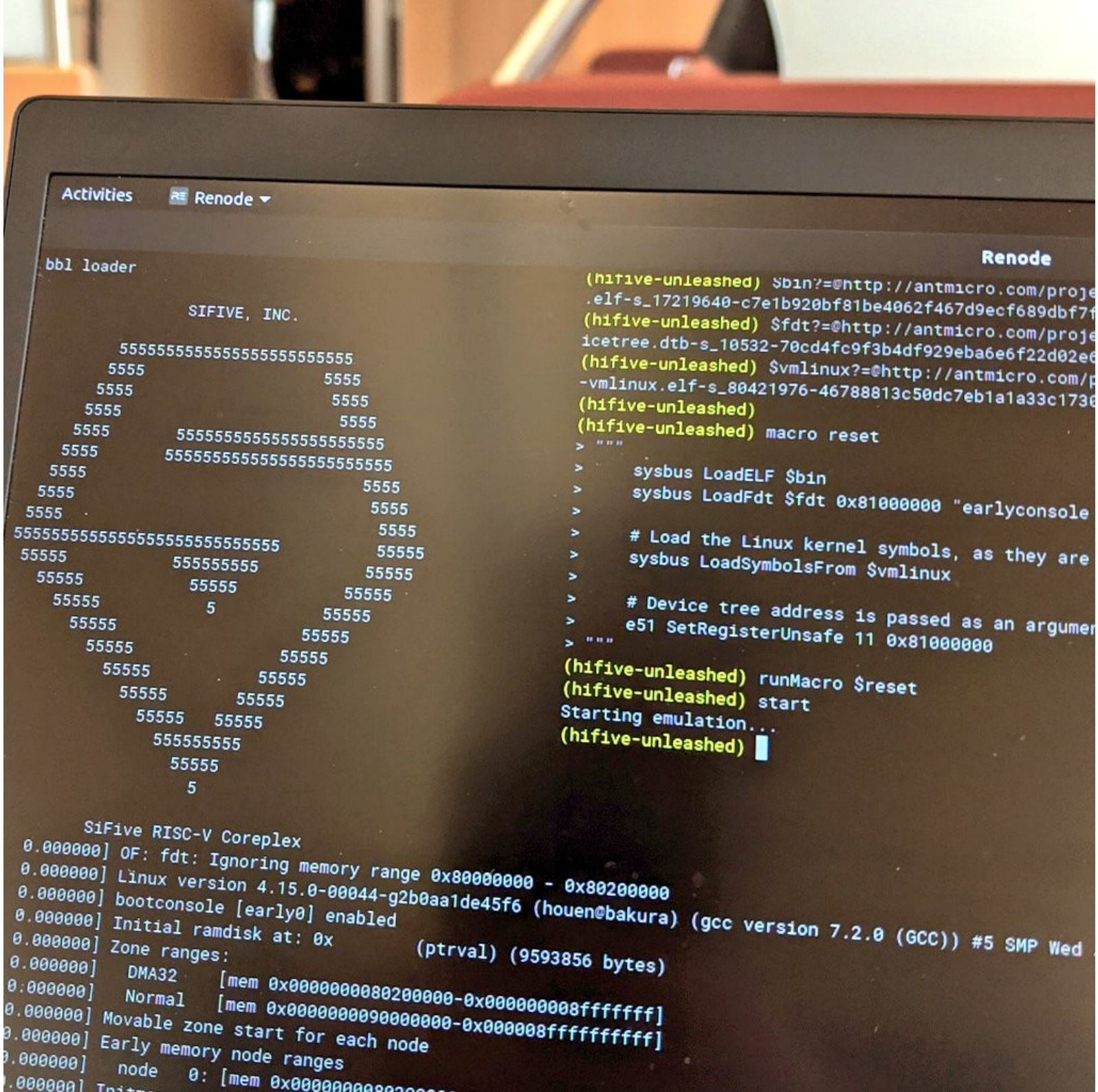
Intel Quark SE Microcontroller Evaluation Kit C1000

single-node/quark_c1000.resc

Fomu

LiteX/VexRiscv on Digilent Arty

Xilinx ZedBoard


Activities
Renode
Renode
bbl loader
SIFIVE, INC.
SiFive RISC-V Coreplex
0.000000] OF: fdt: Ignoring memory range 0x80000000 - 0x80200000
0.000000] Linux version 4.15.0-00044-g2b0aa1de45f6 (houen@bakura) (gcc version 7.2.0 (GCC)) #5 SMP Wed
0.000000] bootconsole [early0] enabled
0.000000] Initial ramdisk at: 0x (ptrval) (9593856 bytes)
0.000000] Zone ranges:
0.000000] DMA32 [mem 0x0000000080200000-0x00000000ffffffff]
0.000000] Normal [mem 0x0000000090000000-0x000008ffffffffff]
0.000000] Movable zone start for each node
0.000000] Early memory node ranges
0.000000] node 0: [mem 0x
(hifive-unleashed) $bin?=@http://antmicro.com/proje
.elf-s_17219640-c7e1b920bf81be4062f467d9ecf689dbf7f
(hifive-unleashed) $fdt?=@http://antmicro.com/proje
icetree.dtb-s_10532-70cd4fc9f3b4df929eba6e6f22d02e6
(hifive-unleashed) $vmlinux?=@http://antmicro.com/p
-vmlinux.elf-s_80421976-46788813c50dc7eb1a1a33c173
(hifive-unleashed)
(hifive-unleashed) macro reset
> """
> sysbus LoadELF $bin
> sysbus LoadFdt $fdt 0x81000000 "earlyconsole
>
> # Load the Linux kernel symbols, as they are
> sysbus LoadSymbolsFrom $vmlinux
>
> # Device tree address is passed as an argumen
> e51 SetRegisterUnsafe 11 0x81000000
> """
(hifive-unleashed) runMacro $reset
(hifive-unleashed) start
Starting emulation...
(hifive-unleashed)

- Björn Töpel for RISC-V Munich meetup last week

Getting your hands dirty

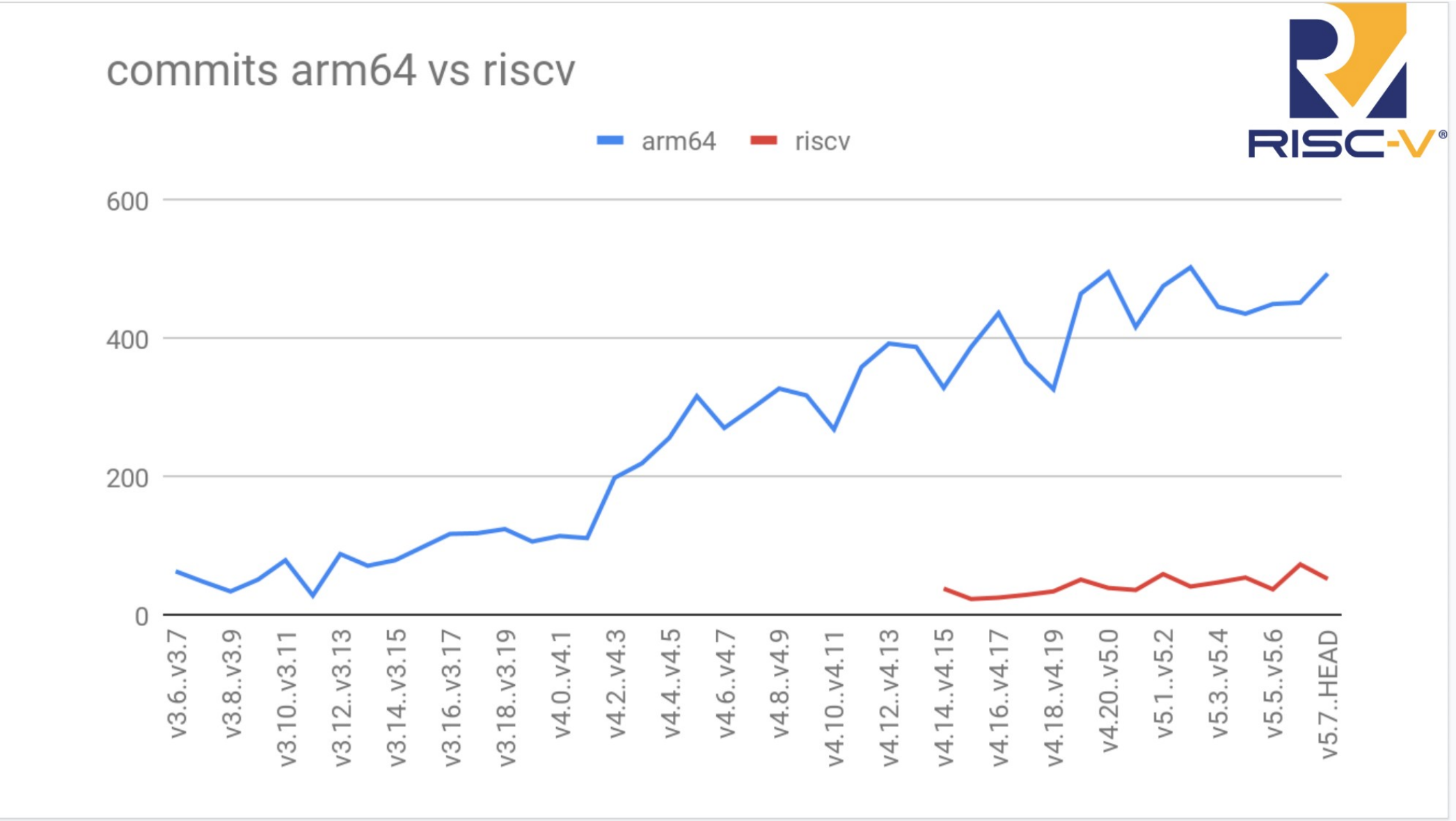
Enter RISC-V land!

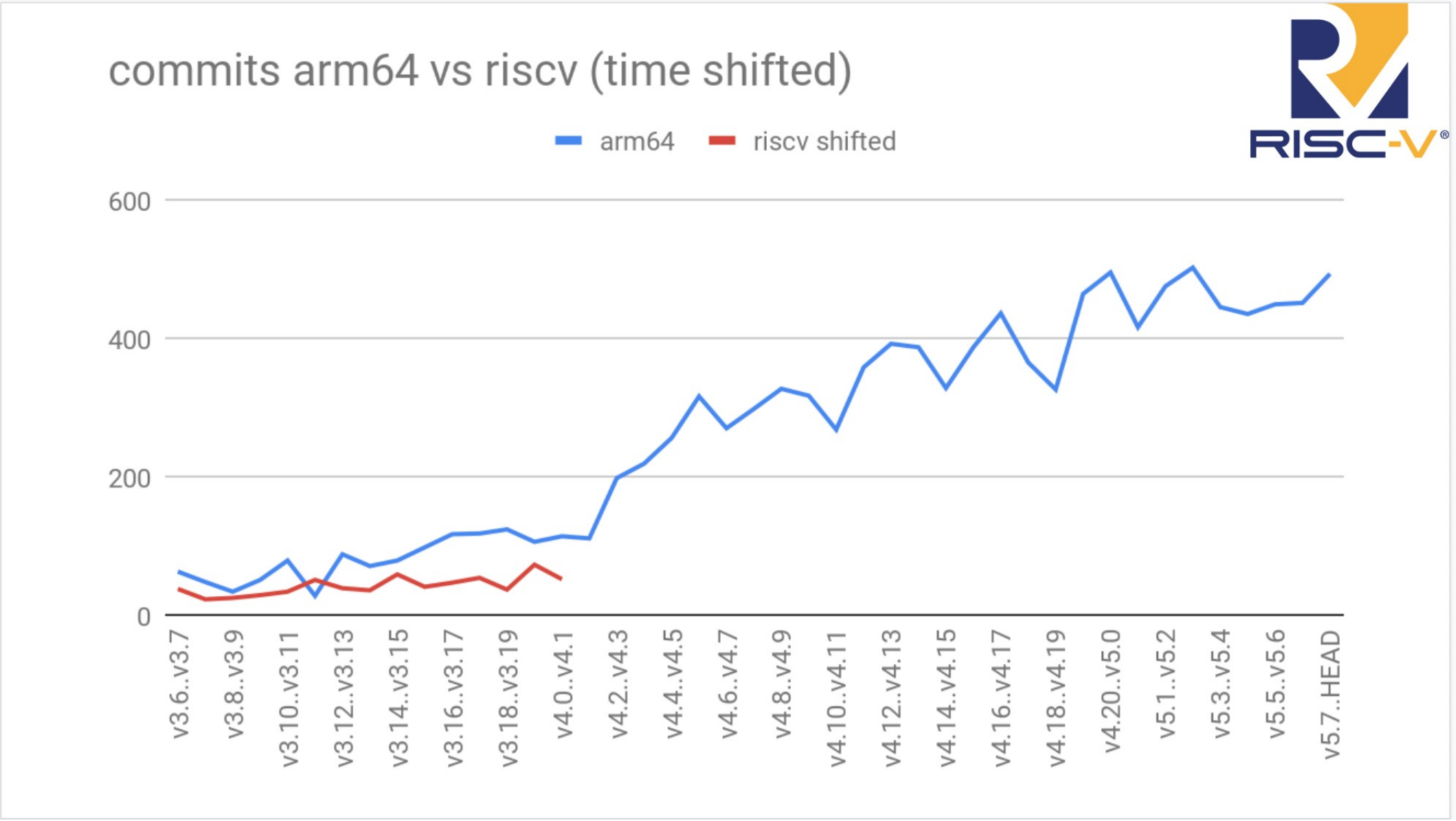
A great way to learn the nitty gritty details of the Linux kernel

Watching RISC-V grow up (from MCU to server core)

It’s a fun, friendly, and still pretty small community

linux-riscv@lists.infradead.org
https://lore.kernel.org/linux-riscv/


commits arm64 vs riscv
arm64
riscv
600
400
200
0
v3.6..v3.7
v3.8..v3.9
v3.10..v3.11
v3.12..v3.13
v3.14..v3.15
v3.16..v3.17
v3.18..v3.19
v4.0..v4.1
v4.2..v4.3
v4.4..v4.5
v4.6..v4.7
v4.8..v4.9
v4.10..v4.11
v4.12..v4.13
v4.14..v4.15
v4.16..v4.17
v4.18..v4.19
v4.20..v5.0
v5.1..v5.2
v5.3..v5.4
v5.5..v5.6
v5.7..HEAD


commits arm64 vs riscv (time shifted)
arm64
riscv shifted
600
400
200
0
v3.6..v3.7
v3.8..v3.9
v3.10..v3.11
v3.12..v3.13
v3.14..v3.15
v3.16..v3.17
v3.18..v3.19
v4.0..v4.1
v4.2..v4.3
v4.4..v4.5
v4.6..v4.7
v4.8..v4.9
v4.10..v4.11
v4.12..v4.13
v4.14..v4.15
v4.16..v4.17
v4.18..v4.19
v4.20..v5.0
v5.1..v5.2
v5.3..v5.4
v5.5..v5.6
v5.7..HEAD

Recent/ongoing work

Clang/LLVM built kernel boots with QEMU (HW?) (Palmer Dabbelt/Nathan Huckleberry)
KGDB support (Vincent Chen)
SPARSEMEM (physical memory model) (Logan Gunthorpe)

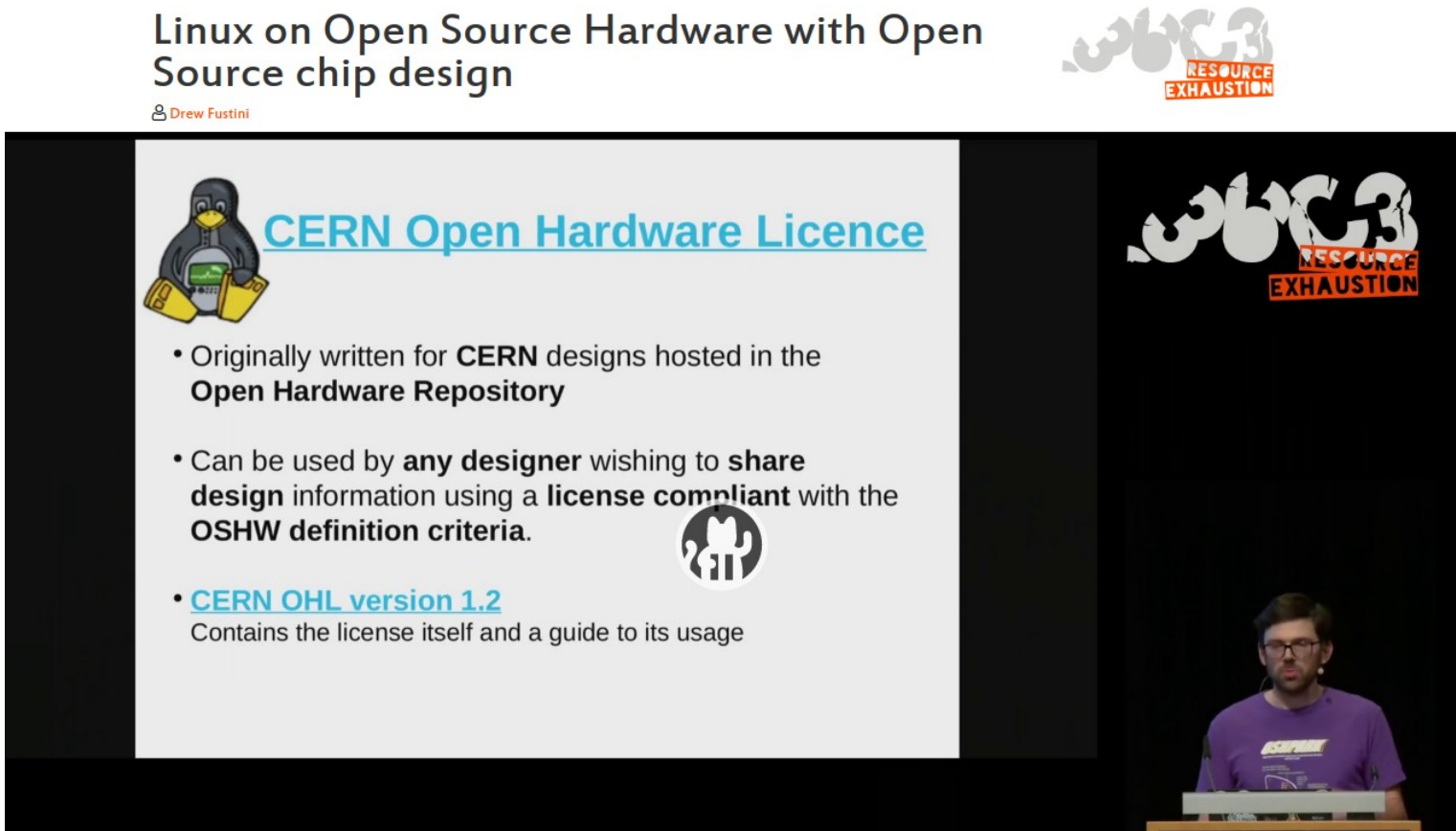
KVM (Anup Patel/Atish Patra)
kexec/kdump support (Nick Kossifidis)
kprobes/kretprobes (Patrick Stählin/Guo Ren)

Finding a project

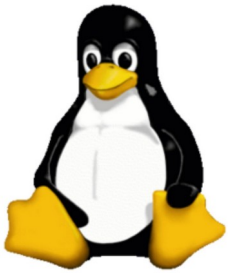
```
$ ./Documentation/features/list-arch.sh riscv | grep -c TODO
28

$ /Documentation/features/list-arch.sh riscv|grep -c ok\ \ \|
12

$ ./Documentation/features/list-arch.sh riscv | grep TODO
```

- Become a RISC-V® AMBASSADOR
- Slides:

  github.com/pdp7/talks/blob/master/rv-elc.pdf
- Contact: @pdp7 || drew@beagleboard.org
- 36c3 talk



RISC-V at ELC



- Join **#2-track-riscv** on the OSS+ELC slack!
- RISC-V: Instruction Sets Want to be Free
  - Krste Asanovic (UC Berkeley)
  - Wednesday, 10:35am: https://sched.co/c4PV
- State of RISC-V Software Development Tools
  - Khem Raj (Comcast)
  - Wednesday 12:15pm: https://sched.co/c3Yn
- Ask the Expert Session with Calista Redmond
  - CEO of RISC-V International
  - Thursday 11:45am: https://sched.co/cosw