

Slides: <https://github.com/pdp7/talks/blob/master/berlin-elixx.pdf>

How to run Linux on RISC-V

with open hardware and open source FPGA tools

Berlin Embedded Linux meetup (2020-02-12)



Drew Fustini

drew@oshpark.com

Twitter: [@pdp7](https://twitter.com/pdp7)



Section:
RISC-V

the instruction set for everything?



- **RISC-V: Free and 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”



- **RISC-V: Free and Open RISC Instruction Set Arch**
 - Instruction Sets Want To Be Free: A Case for RISC-V
 - David Patterson, UC Berkeley – *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



State of the Union

Krste Asanovic

UC Berkeley, RISC-V Foundation, & SiFive Inc.

krste@berkeley.edu

RISC-V Summit
San Jose Convention Center, CA, USA
December 10, 2019

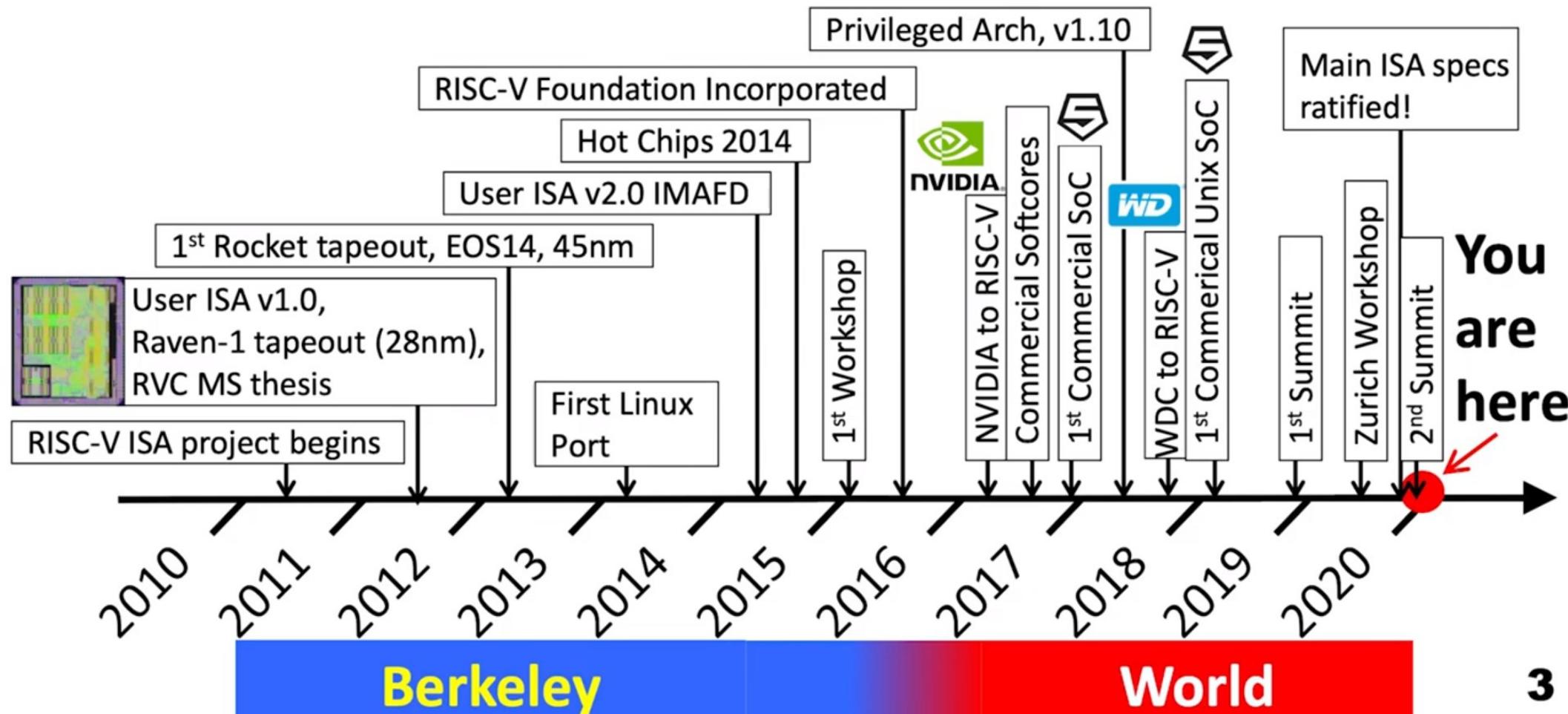


What's Different about RISC-V?

- ***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 Timeline





RISC-V Ecosystem

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 ...

Software



ISA specification

Golden Model

Compliance

Hardware

Open-source cores:

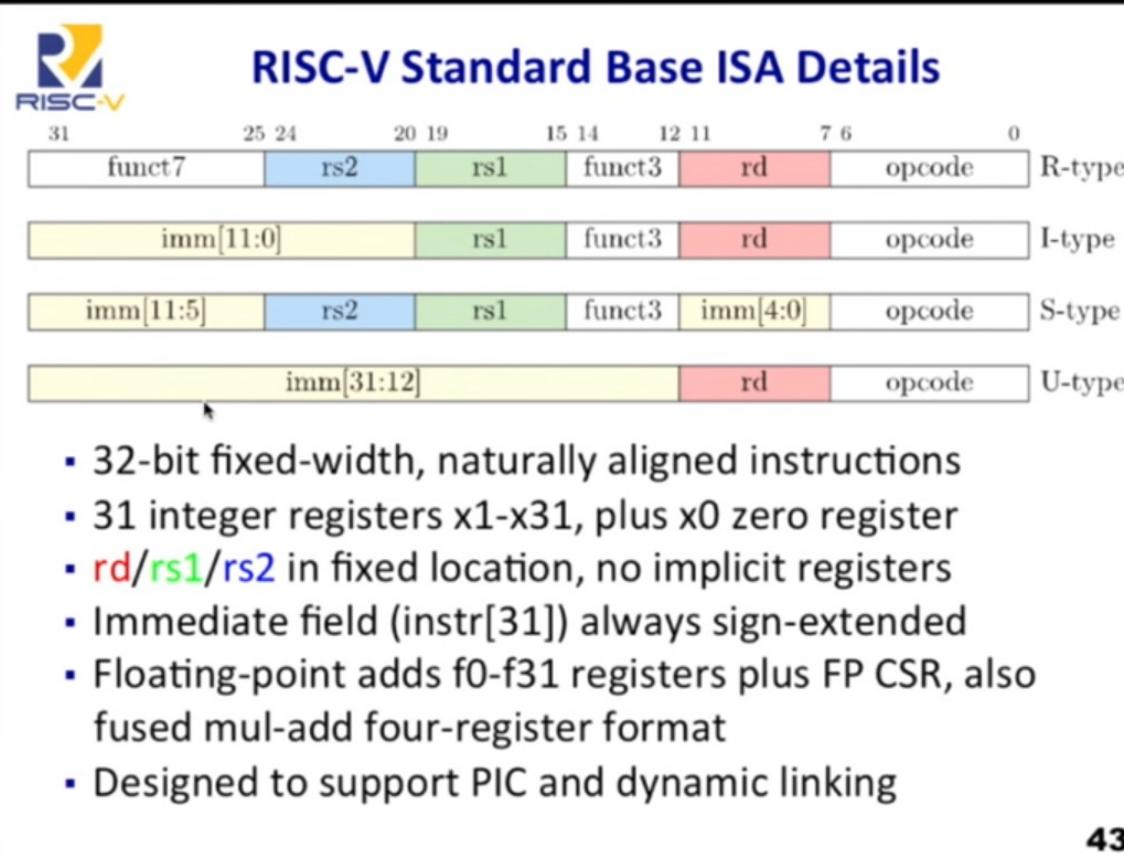
Rocket, BOOM, RI5CY, Ariane, PicoRV32, Piccolo, SCR1, Shakti, Serv, Swerv, Hummingbird, ...

Commercial core providers:

Alibaba, Andes, Bluespec, Cloudbear, Codasip, Cortus, InCore, Nuclei, SiFive, Syntacore, ...

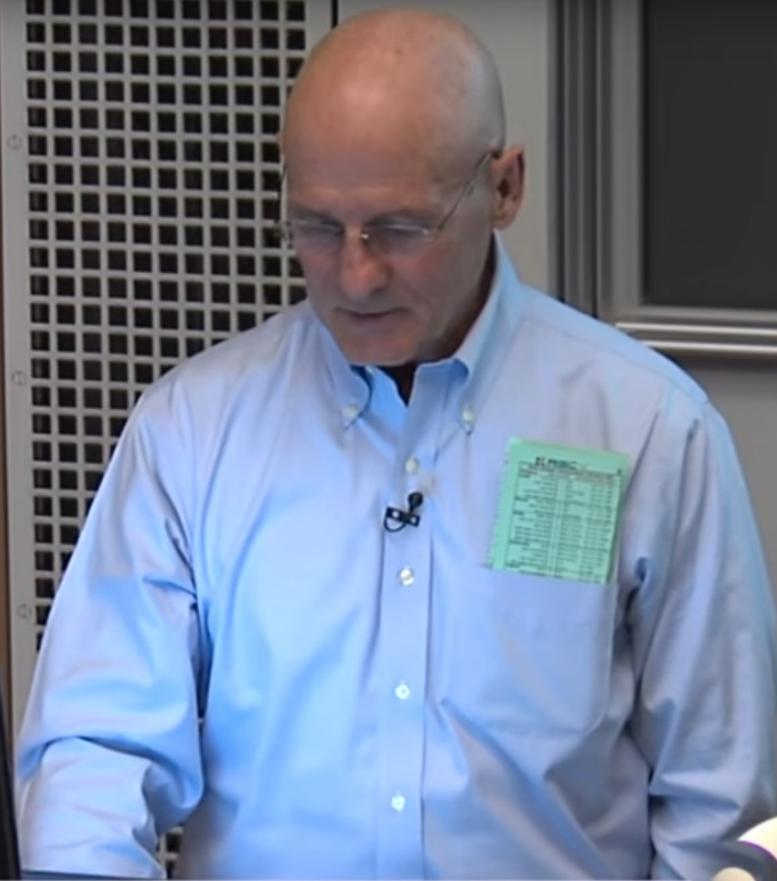
Inhouse cores:

Nvidia, WD, +others





RV32I			
① RISC-V Reference Card			
Base Integer Instructions (32-bit/64-bit/128-bit Base)			
Category Name Format RV32I/32(64)/128(128)			
Loads			
Load Byte	I	LB	r1,r2,r3,imm
Load Halfword	I	LH	r1,r2,r3,imm
Load Word	I	LW (D/I/Q)	r1,r2,r3,imm
Load Byte Unsigned	I	LBU	r1,r2,r3,imm
Load Half Unsigned	I	LHU (W/D/Q)	r1,r2,r3,imm
Stores			
Store Byte	S	SB	r1,r2,r3,imm
Store Halfword	S	SH	r1,r2,r3,imm
Store Word	S	SW (D/Q)	r1,r2,r3,imm
Shifts			
Shift Left	R	SLLI (W/D)	r1,r2,r3,r4
Shift Left Immediate	I	SLSLT (W/D)	r1,r2,r3,shamt
Shift Right	R	SLR (W/D)	r1,r2,r3,r4
Shift Right Immediate	I	SLSLT (W/D)	r1,r2,r3,shamt
Shift Right Arithmetic	R	SLRA (W/D)	r1,r2,r3,r4
Shift Right Arithmetic Immediate	I	SLSLT (W/D)	r1,r2,r3,shamt
Arithmetic			
ADD	R	ADD (I/W/D)	r1,r2,r3,r4
ADD Immediate	I	ADDI (I/W/D)	r1,r2,r3,imm
Subtract	R	DSUB (I/W/D)	r1,r2,r3,r4
Load Upper Imm.	U	LDU	r1,imm
Load Upper Imm. to DC	U	LDUDC	r1,imm
Logical			
XOR	R	XOR	r1,r2,r3,r4
XOR Immediate	I	XORI	r1,r2,r3,imm
OR	R	OR	r1,r2,r3,r4
OR Immediate	I	ORI	r1,r2,r3,imm
AND	R	AND	r1,r2,r3,r4
AND Immediate	I	ANDI	r1,r2,r3,imm
Compare			
Set <	R	SLT	r1,r2,r3,r4
Set < Immediate	I	SLTI	r1,r2,r3,imm
Set < Unsigned	R	SLTU	r1,r2,r3,r4
Set < Unsigned Immediate	I	SLTUI	r1,r2,r3,imm
Branches			
Branch =	S	BNE	r1,r2,r3,imm
Branch =	S	BNEQ	r1,r2,r3,imm
Branch =	S	BNEQZ	r1,r2,r3,imm
Branch =	S	BNEQNZ	r1,r2,r3,imm
Branch <	S	BLT	r1,r2,r3,imm
Branch < Unsigned	S	BLTU	r1,r2,r3,imm
Branch < Unsigned Immediate	S	BLTUI	r1,r2,r3,imm
Jump & Link			
Jump & Link	BL	BL	r1,imm
Jump & Link Register	BLR	BLR	r1,r2,r3,imm
Systm			
Sync Thread	I	FENCE	
Sync Inst & Data	I	FENCE.I	
System			
System CALL	I	ECALL	
System BREAK	I	EBREAK	
Counters			
Read CYCLE	I	RCYCLE	#0
Anti-CYCLE upper Half	I	RCYCLESU	#0
Anti-CYCLE lower Half	I	RCYCLESL	#0
Read TIME	I	RTIME	#0
Read TIME upper Half	I	RTIMESU	#0
Read TIME lower Half	I	RTIMESL	#0
Read INSTR Retired	I	RICNTRMT	#0
Read INSTR upper Half	I	RICNTRMTH	#0
32-bit Instruction Formats			



Dave Patterson
UC BERKELEY



**RV32I / RV64I / RV128I + M, A, F, D, Q, C
RISC-V “Green Card”**



Base Integer Instructions (32/64/128)

Category	Name	Format	RV32/64/128 Basic
Loads	Load Byte	I	LB
	Load Halfword	I	LH
	Load Word	I	LD
	Load Halfword Unaligned	I	LH.U
	Load Word Unaligned	I	LD.U
Stores	Store Byte	S	SB
	Store Halfword	S	SH
	Store Word	S	SD
Shifts	Shift Left	R	SL
	Shift Left Immediate	I	SLI
	Shift Right	R	SR
	Shift Right Immediate	I	SRI
	Shift Right Arithmetic	R	SA
	Shift Right Arithmetic Immediate	I	SRI.A
Arithmetic	Add	R	ADD
	Add Immediate	I	ADDI
	Subtract	R	SUB
	Subtract Immediate	I	SUBI
	Mul	R	MUL
	Mul Immediate	I	MULI
	Divide	R	DIV
	Divide Immediate	I	DIVI
Logical	Not	R	NOT
	Not Immediate	I	NOTI
	Or	R	OR
	Or Immediate	I	ORI
	And	R	AND
	And Immediate	I	ANDI
Compare	Set =	R	SSET
	Set < Immediate	I	SSETI
	Set < Unsigned	I	SSETU
Branches	Branch	I	BEQ
	Branch +	I	BNE
	Branch -	I	BGE
	Branch -	I	BGT
	Branch < Unsigned	I	BGEU
	Branch < Unsigned	I	BGTU
Jump & Link	Jump	I	JAL
	Jump & Link Register	I	JALR
Synch	Synch Branch	I	FSYNC
	Synch Branch & Status	I	FSYNC_I
System	System CALL	I	SYSCALL
	System RETRET	I	SYCRETRET
Counters	Read CYCLE	I	RSVCYC
	Read CYCLE upper Half	I	RSVCYCH
	Read TIME	I	RSVTIME
	Read TIME upper Half	I	RSVTIMEH
	Read INSTR RETVAL	I	RSVINSTRRET
	Read INSTR RETVAL half	I	RSVINSTRRETH

RV Privileged Instructions (32/64/128)

Category	Name	Format	RV32/64/128 Basic
C/SW Access	Atomic R/W	R	CSWRW
	Atomic Read & Set R/W	R	CSSRW
	Atomic Read & Clear R/W	R	CSCRW
	Atomic R/W Imm	R	CWRIM
	Atomic Read & Set R/W Imm	R	CSSRIM
	Atomic Read & Clear R/W Imm	R	CSCRIM
Change Level	Env. Call	R	RCALL
	Environment Breakpoint	R	REB
Trap Redirect	To Supervisor	R	RTS
	Redirect Trap to Supervisor	R	RTTS
	Processor Trap to Supervisor	R	PTTS
Interrupt	Wait for Interrupt	R	RIWT
	Processor PENDING	R	RPND
	Processor PENDING, 16-bit	R	RPND16
Optional Multiplication-Division Extension: RV32M			
Multiply	Multiply	R	MULL
	Multiply upper half	R	MULLH
	Multiply half signs/ints	R	MULLHSI
	Multiply upper half ints	R	MULLHSU
Divide	Divide	R	DIVU
	Divide Unsigned	R	DIVUS
	Divide	R	DIVS
	Divide Unsigned	R	DIVUSI
	Divide S	R	DIVSI
Optional Atomic Instruction Extension: RVA			
Load	Load Reserved	R	LR
	More Store Condition	R	LMSC
	Swap	R	ASWAP
And	And	R	AMR
Logical	Not	R	ANOT
	And	R	ANAND
	Or	R	ANOR
	Maximum	R	AMAX
	Minimum	R	AMIN
	Maximum Unsigned	R	AMAXU
	Minimum Unsigned	R	AMINU
3 Optional FP Extensions: RV32/64/128(F/D/Q)			
Moves	Move from Integer	R	FMV.I
	Move to Register	R	FMV.R
Convert	Convert from Int	R	FPOV.I
	Convert from Int Unsigned	R	FPOV.U
	Convert to Int	R	FPOV.I
	Convert to Int Unsigned	R	FPOV.U
3 Optional FP Extensions: RV64/128(F/D/Q)			
Moves	Move from Integer	R	FMV.I
	Move to Register	R	FMV.R
Convert	Convert from Int	R	FPOV.I
	Convert from Int Unsigned	R	FPOV.U
	Convert to Int	R	FPOV.I
	Convert to Int Unsigned	R	FPOV.U
Optional Compressed Instructions: RV32			
Load	Load Word	R	CL
	Load Word SP	R	CLSP
	Load Double	R	CD
	Load Double SP	R	CDSP
	Load Quad	R	CQ
	Load Quad SP	R	CQSP
	Load Byte Unsigned	R	CB
	Float Load Word	R	CF
	Float Load Double	R	CFD
	Float Load Word SP	R	CFSP
	Float Load Double SP	R	CFDSP
Stores	Store Word	S	CS
	Store Word SP	S	CSSP
	Store Double	S	CD
	Store Double SP	S	CDSP
	Store Quad	S	CQ
	Store Quad SP	S	CQSP
	Float Store Word	S	CF
	Float Store Double	S	CFD
	Float Store Word SP	S	CFSP
	Float Store Double SP	S	CFDSP
Arithmetic	Add	R	CR
	Add Word	R	CAWD
	Add Immediate	I	CAWI
	Add Double	R	CDWD
	Add Double Imm	I	CDWI
	Add SP Imm	I	CDSP
	Add SP Imm * 4	I	CDSP4
	Load Immediate	R	CL
	Load Double	R	CD
	Load Double Imm	I	CDI
	Move	R	CM
	Sub	R	CS
	Sub Word	R	CW
Logical	Not	R	CN
	Or	R	CO
	And	R	CA
	And Immediate	I	CAWI
Shifts	Shift Left	R	CSLL
	Shift Right Immediate	I	CSRMI
	Shift Right Arith Int	I	CSRMI
Branches	Branch	R	CB
	Branch to Int	R	CBINT
	Branch to Int Unsigned	R	CBINTU
Jump	Jump	I	J
	Jump Register	I	JR
Jump & Link	Jump	I	JAL
	Jump & Link Register	I	JALR
System	Env. Break	I	SE

RISC-V and Industry

- Designed to be extensible
 - Microcontroller to supercomputer
- RISC-V Foundation now controls standard: riscv.org
 - Over 400 members: companies, universities and more
 - [YouTube channel has hundreds of talks!](#)
 - <https://www.youtube.com/channel/UC5gLmcFuvdGbajs4VL-WU3g>
- Companies like Nvidia and Western Digital will ship millions of devices with RISC-V
- Avoid ARM licensing fees
- Freedom to leverage open source implementations
 - BOOM, Rocket, PULP, SweRV, and many more

[Join the Mailing Lists](#)info@riscv.org [fb](#) [tw](#) [in](#) [D](#) [w](#) | [Member Login](#)[ABOUT](#) [MEMBERSHIP](#) [SPECS & SUPPORT](#) [CORES & TOOLS](#) [NEWS](#) [EVENTS](#)

RISC-V® Summit

THANKS FOR ATTENDING!

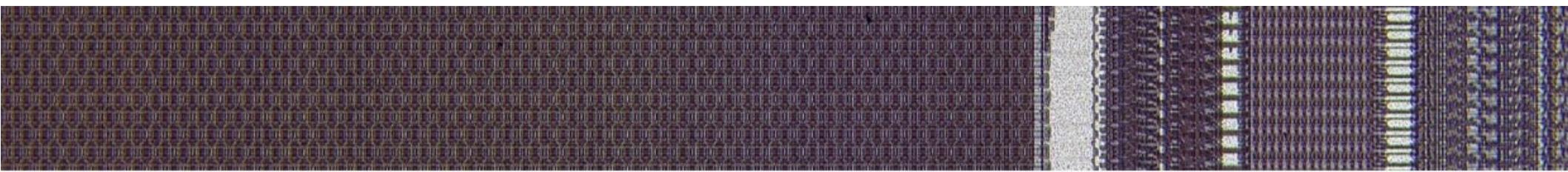
The **RISC-V Summit 2019** was held in San Jose, California on December 9-12. Slides are **now available**, videos coming soon.



RISC-V: The Free and Open RISC Instruction Set Architecture

RISC-V is a free and open ISA enabling a new era of processor innovation through open standard collaboration. Born in academia and research, RISC-V ISA delivers a new level of free, extensible software and hardware freedom on architecture, paving the way for the next 50 years of computing design and innovation.

[NEW TO RISC-V? LEARN MORE](#)[Privacy - Terms](#)



RISC-V

7.86K subscribers

SUBSCRIBED



HOME

VIDEOS

PLAYLISTS

COMMUNITY

CHANNELS

ABOUT



Uploads PLAY ALL

SORT BY



RISC-V Summit 2019: 69
SafeRV Building Blocks for...

150 views • 1 week ago

RISC-V Summit 2019: 61
Andes RISC V Processor...

210 views • 1 week ago

RISC-V Summit 2019: 47
Production ready RISC V ...

563 views • 1 week ago

RISC-V Summit 2019: 64 Ara
2 0 64 bit RISC V Vector...

237 views • 1 week ago

RISC-V Summit 2019: 63
Working Towards a Commo...

467 views • 1 week ago

RISC-V Summit 2019: 59
RISC V Processor Verificati...

76 views • 1 week ago



RISC-V Summit 2019: 62
Ruby Sponsor SiFive presen...

36 views • 1 week ago

RISC-V Summit 2019: 66
Rambus presents Challenge...

27 views • 1 week ago

RISC-V Summit 2019: 49
SweRV Cores Roadmap

151 views • 1 week ago

RISC-V Summit 2019: 50
RISC V Enclaves A Clean...

77 views • 1 week ago

RISC-V Summit 2019: 54
RISC V A New Zero Trust...

52 views • 1 week ago

RISC-V Summit 2019: 48
Ruby Sponsor SiFive presen...

122 views • 1 week ago



RISC-V Summit 2019: 51
sel4 on RISC V Verified OS...

108 views • 1 week ago

RISC-V Summit 2019: 60
Headline Sponsor Western...

54 views • 1 week ago

RISC-V Summit 2019: 46
RISC V For Heterogeneous...

54 views • 1 week ago

RISC-V Summit 2019: 58
Innovation in CPU...

180 views • 1 week ago

RISC-V Summit 2019: 53
Integrate RISC V to build...

58 views • 1 week ago

RISC-V Summit 2019: 56
OneSpin presents More tha...

104 views • 1 week ago

RISC-V and the world

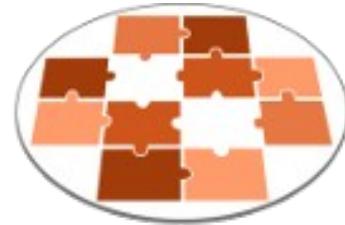
- RISC-V Foundation moving from US to Switzerland
- Nations such as India have RISC-V initiatives
 - Desire for sovereign technology and avoid backdoors from other nations
- Strong interest from chipmakers in China
 - U.S. companies have been banned from doing business with Huawei... who's next?
 - ARM deemed UK-origin tech so ok to do business with Huawei, but what will brexit-govt bring?



- **lowRISC** is a not-for-profit organisation 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

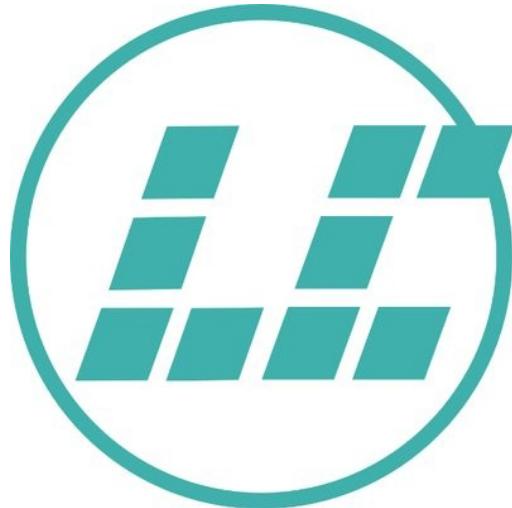


- Tutorial for the v0.7 lowRISC release
 - By Jonathan Kimmitt (lead developer), and Alex Bradbury (lead reviewer)
 - <https://www.cl.cam.ac.uk/~jrrk2/docs/ariane-v0.7/tutorial/>
 - Digilent Nexys A7-100T: \$265
 - This tutorial adds further functionality towards the final SoC design:
 - Graphical Colour Console with X-windows support incorporating mouse and keyboard events.
 - Choice of SD-Card, Quad-SPI or Ethernet TFTP boot-loader with DHCP support.
 - Linux 5.3.8 RISCV kernel and updated Debian userland with advanced package tool.
 - Choice of RV64-GC Rocket (Chisel) or Ariane (SystemVerilog) CPU



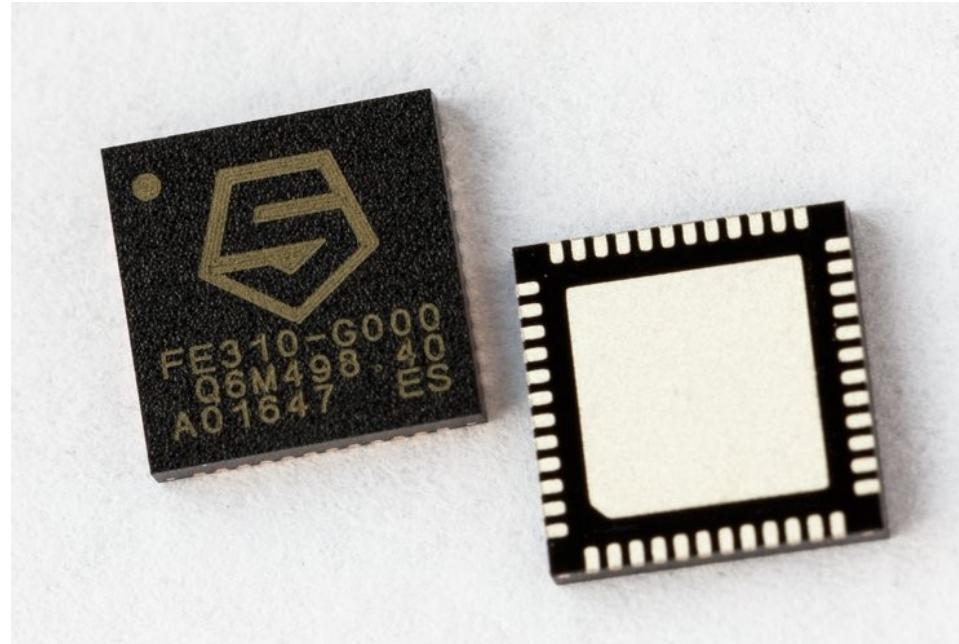
FOSSi
Foundation

- **FOSSi Foundation**
 - The Free and Open Source Silicon Foundation
 - “non-profit foundation with the mission to promote and assist free and open digital hardware designs”
 - Events: ORConf, Latch-up, Week of OSHW
 - **Open Source Silicon Design Ecosystem**
 - Talk by FOSSi co-founder Julius Baxter



- **LibreCores**
 - Project of the FOSSi Foundation
 - “**gateway to free and open source digital designs** and other components that you can use and **re-use in your digital designs**”
 - “advances the idea of OpenCores.org”

SiFive

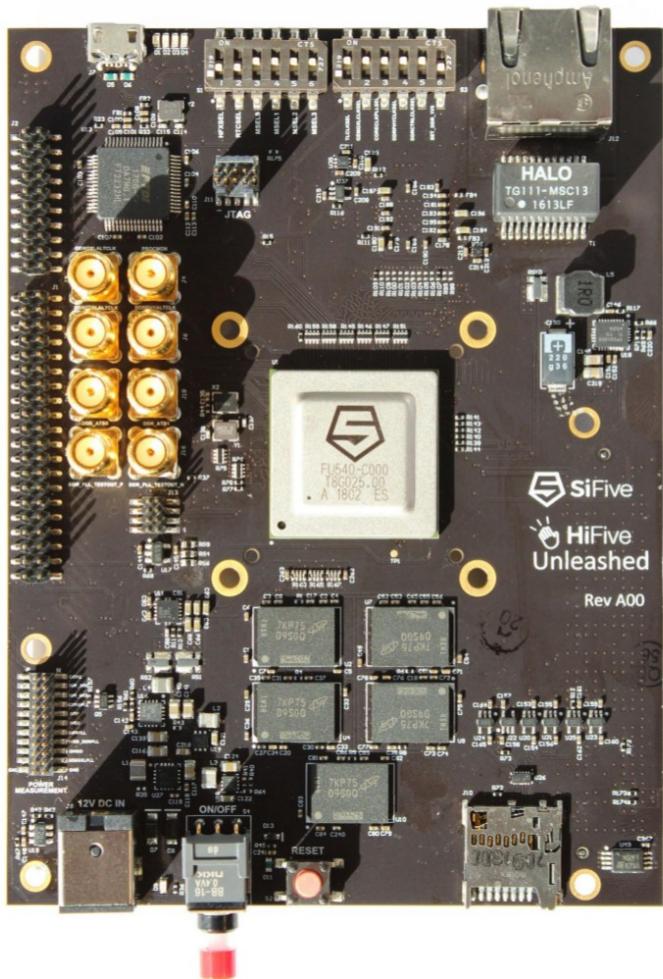


- “founded by the creators of the free and open RISC-V architecture as a reaction to the end of conventional transistor scaling and escalating chip design costs”

SiFive: Linux on RISC-V



HiFive Unleashed



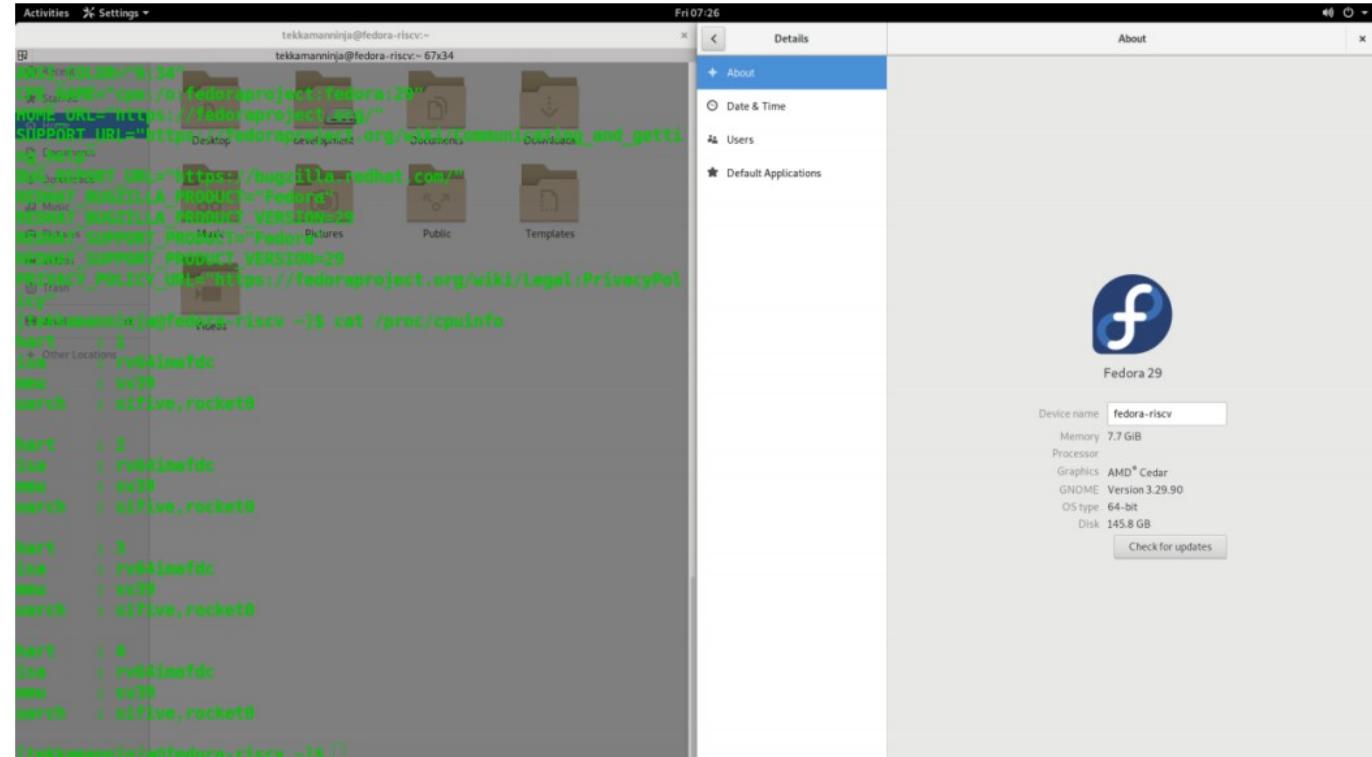
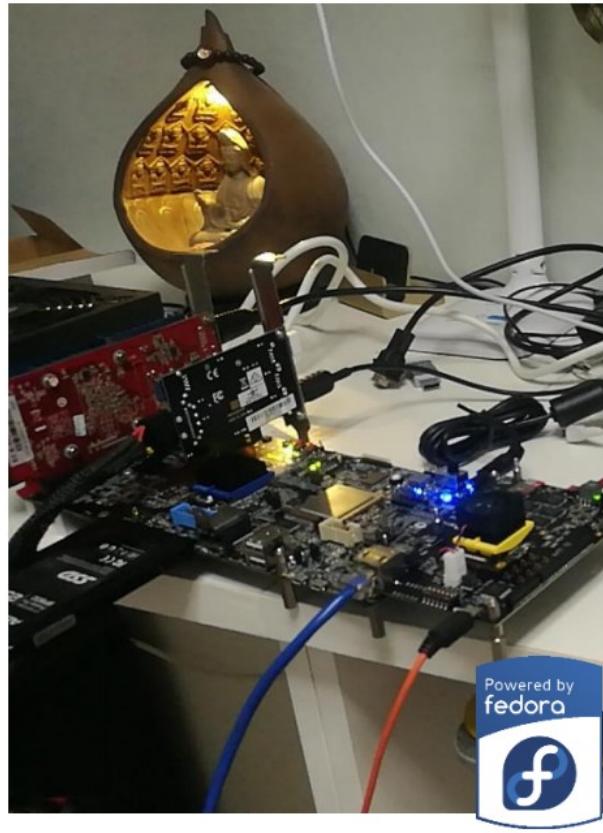
- World's First Multi-Core RISC-V Linux Development Board
 - SiFive FU540-C000 (built in 28nm)
 - 4+1 Multi-Core Coherent Configuration, up to 1.5 GHz
 - 4x U54 RV64GC Application Cores with Sv39 Virtual Memory Support
 - 1x E51 RV64IMAC Management Core
 - Coherent 2MB L2 Cache
 - 64-bit DDR4 with ECC
 - 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

RISC-V Summit 2019: Linux on RISC V Fedora and Firmware Status Update

- <https://www.youtube.com/watch?v=WC6e3g8uWdk>
- Wei Fu – Software Engineer, Red Hat

The screenshot shows a YouTube video player interface. At the top, the title is "RISC-V Summit 2019: 10 Linux on RISC V Fedora and Firmware Status Update". Below the title, the channel name "RISC-V" is visible with "7.91K subscribers". The video duration is 9:56 / 18:37. The main content is a slide titled "Fedora on RISC-V". The slide features a diagram with a green arrow pointing from a "Rich Software EcoSystem" box to a "Fedora always OPEN" logo with a penguin. Another blue arrow points from a "Rich Hardware EcoSystem" box to two cartoon penguins dressed as Uncle Sam and a cowboy. The text "I WANT YOU! I WANT YOU" is overlaid on the penguins. A footer on the slide reads "We would like to support more targets based on standard RISC-V Spec.". The video player has standard controls (play/pause, volume, etc.) and a progress bar. The URL "#RISCVSUMMIT | tmt.knect365.com/risc-v-summit/" is displayed at the bottom of the slide. The Red Hat logo is also present. The bottom of the screen shows typical YouTube video player metrics: likes (6), dislikes (0), share, save, and a "SUBSCRIBED" button with a bell icon.

Fedora GNOME Image on SiFive Unleashed



#RISCVSUMMIT | tmt.knect365.com/risc-v-summit/

Targets

Supported



Virtual: libvirt + QEMU

with graphics parameters (Spice).



Real Hardware: SiFive Unleashed

with Expansion Board, PCI-E graphic Card & SATA SSD

Tested



QEMU for AndeStar V5 && ADP-XC7KFF676

Andes QEMU and AndeShape **FPGA** board



中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES



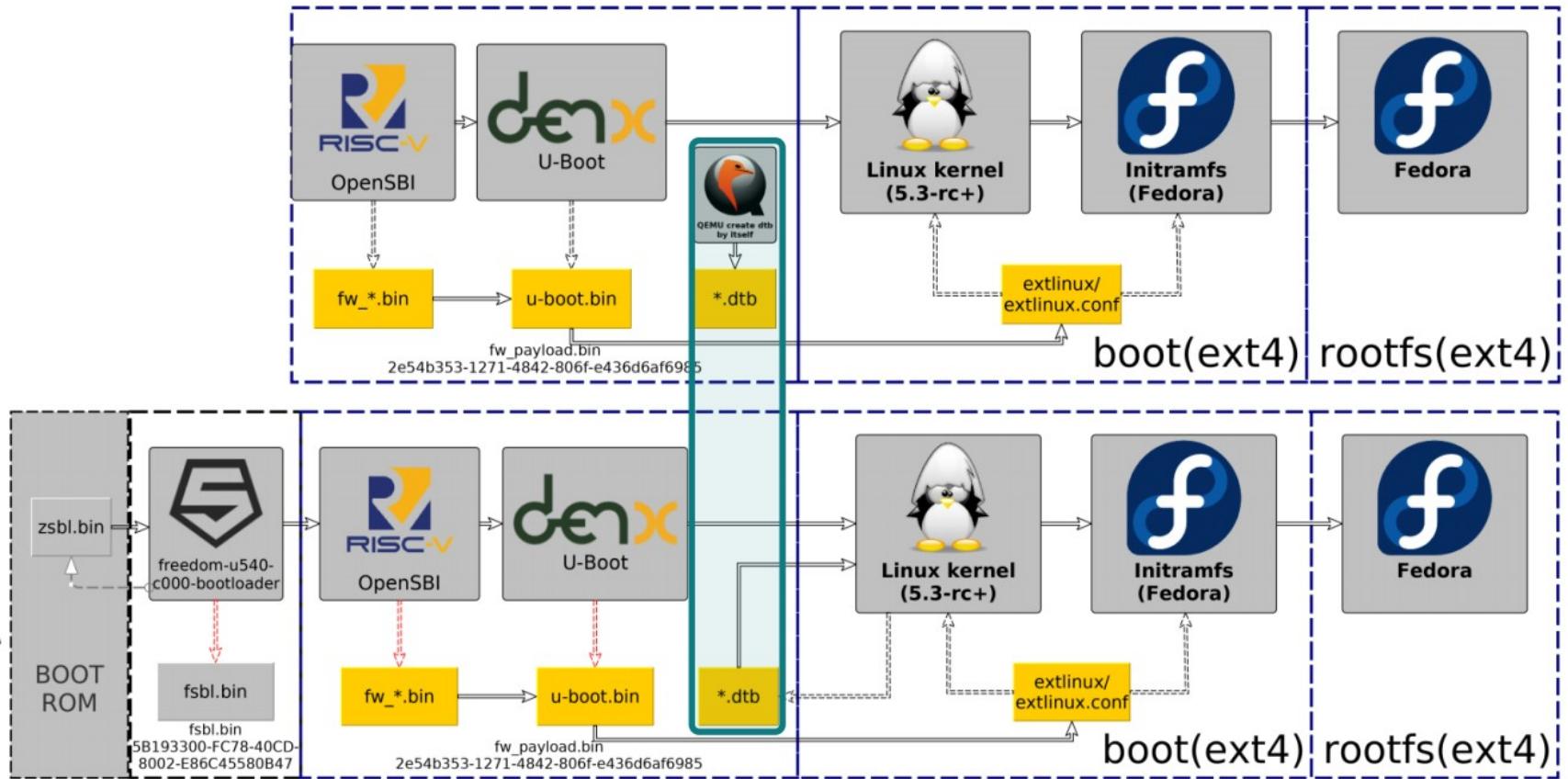
ICT SERVE Platform: FlameCluster

FPGA Cloud development platform (with PCI-E SSD and graphic Card)



#RISCVSUMMIT | tmt.knect365.com/risc-v-summit/

The current boot flow for Fedora on RISC-V





RISC-V

This page contains details about a port of Debian for the RISC-V architecture called **riscv64**.

Contents

[1. In a nutshell](#)

- [1. What is RISC-V?](#)
- [2. What is a Debian port?](#)
- [3. What are the goals of this project in particular?](#)
- [4. Progress](#)

[2. Upstream project / Architecture](#)

- [1. Upstream project / Community](#)
- [2. Architecture details](#)
- [3. Toolchain upstreaming status](#)

[3. Hardware](#)

- [1. ASIC implementations, i.e. "real" CPU chips](#)
 - [1. SiFive "Freedom U540" SoC \(quad-core RV64GC\) / "HiFive Unleashed"](#)
 - [2. Planned](#)
- [2. FPGA implementations](#)

- Experiment to get Linux on the low cost Kendryte K210 RISC-V microcontroller
 - dual core 64-bit RISC-V at 400MHz with 8MB SRAM
 - Sipeed MAix BiT for RISC-V \$13
 - PDF: RISC-V NOMMU and M-mode Linux
 - youtube.com/watch?v=ycG592N9EMA&t=10394
 - jump to 2h 53m
- Many RISC-V Improvements Ready For Linux 5.5: M-Mode, SECCOMP, Other Features



RISC-V NOMMU and M-Mode Linux

Damien Le Moal, Western Digital

Linux Plumbers Conference, September 9th, 2019

Kendryte K210 SoC + Busybox

Sipeed MAIX Go Board (6+2 MB SRAM)

```
[ 0.00000] Linux version 5.1.0-rc5-00314-g375c2321604f (damien@washi) (gcc version 8.2.0 (Buildroot 2018.11-rc2-00003-ga0787e9)) #221 SMP Fri May 10 15:17:17 JST 2019
[ 0.00000] earlycon: sbi0 at I/O port 0x0 (options '')
[ 0.00000] printk: bootconsole [sbi0] enabled
[ 0.00000] initrd not found or empty - disabling initrd
[ 0.00000] Zone ranges:
[ 0.00000]   DMA32    [mem 0x00000008000000-0x0000000807ffff]
[ 0.00000]   Normal    empty
[ 0.00000] Movable zone start for each node
[ 0.00000] Early memory node ranges
[ 0.00000]   node  0: [mem 0x00000008000000-0x0000000807ffff]
[ 0.00000] Initmem setup node 0 [mem 0x00000008000000-0x0000000807ffff]
[ 0.00000] elf_hwcap is 0x112d
[...]
[ 0.00000] Built 1 zonelists, mobility grouping off. Total pages: 2020
[ 0.00000] Kernel command line: console=hvc0 earlycon=sbi init=/bin/bash
[ 0.00000] Dentry cache hash table entries: 1024 (order: 1, 8192 bytes)
[ 0.00000] Inode-cache hash table entries: 512 (order: 0, 4096 bytes)
[ 0.00000] Sorting __ex_table...
[ 0.00000] Memory: 6284K/8192K available (920K kernel code, 101K rwdta, 158K rodata, 393K init, 95K bss, 1908K reserved, 0K cma-reserved)
[ 0.00000] SLUB: HWalign=64, Order=0-3, MinObjects=0, CPUs=2, Nodes=1
[ 0.00000] rcu: Hierarchical RCU implementation.
[ 0.00000] rcu: RCU calculated value of scheduler-enlistment delay is 25 jiffies.
[ 0.00000] NR_IRQS: 0, nr_irqs: 0, preallocated irqs: 0
[...]
[ 0.251433] Freeing unused kernel memory: 392K
[ 0.254361] This architecture does not have kernel memory protection.
[ 0.259473] Run /bin/bash as init process
```

```
BusyBox v1.30.1 (2019-05-10 14:49:46 JST) hush - the humble shell
```

```
# mount -t proc none /proc
# cat /proc/cpuinfo
processor : 0
hart : 0
isa : rv64imafdc

processor : 1
hart : 1
isa : rv64imafdc
```

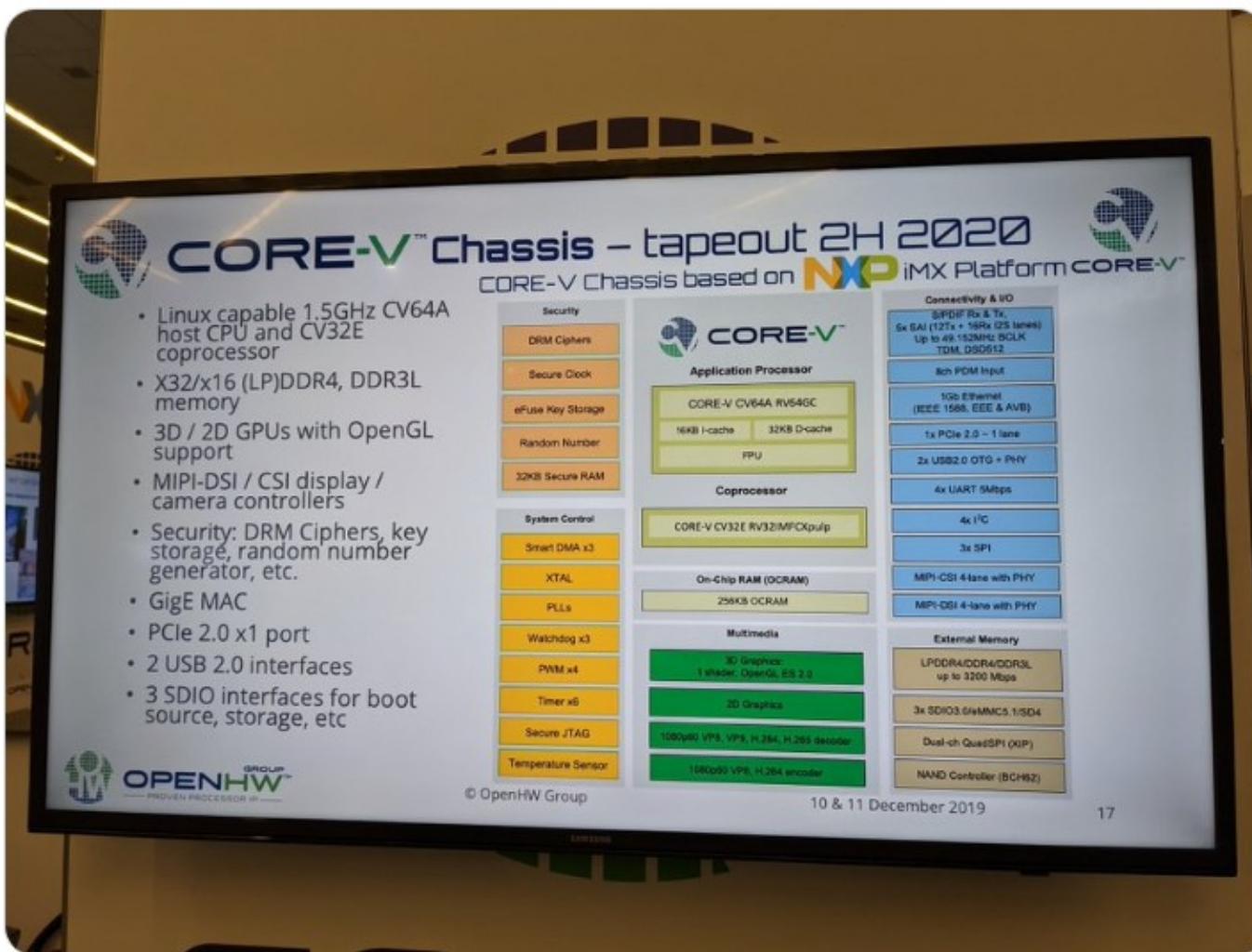
Coming in 2020?

- Andes 27-series CPU
 - “32-bit A27 and 64-bit AX27 and NX27V cores, which will enter production in Q1 2020.”
 - [Andes’ RISC-V SoC debuts with AI-ready VPU as Microchip opens access to its PolarFire SoC](#)
- Microchip PolarFire SoC FPGA
 - Hard RISC-V with FPGA fabric... like the Xilinx Zync for ARM
- NXP iMX with RISC-V instead of ARM!
 - [OpenHW Group Unveils CORE-V Chassis SoC Project, Building on PULP Project IP](#)



Karim Yaghmour
@karimyaghmour

Spotted at RISC V summit: an iMX chip where the ARM core was ripped out and replaced with a RISC V/PULP - just wow



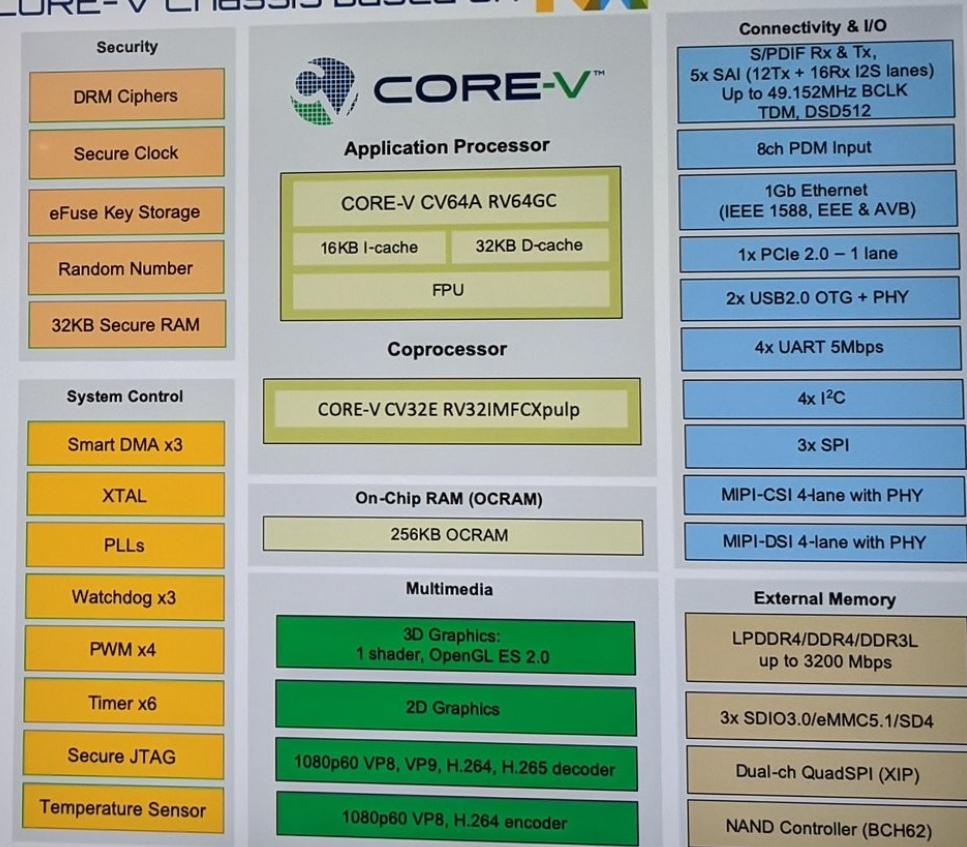


CORE-V™ chassis – tapeout 2H 2020



CORE-V Chassis based on **NXP** iMX Platform **CORE-V**

- Linux capable 1.5GHz CV64A host CPU and CV32E coprocessor
- X32/x16 (LP)DDR4, DDR3L memory
- 3D / 2D GPUs with OpenGL support
- MIPI-DSI / CSI display / camera controllers
- Security: DRM Ciphers, key storage, random number generator, etc.
- GigE MAC
- PCIe 2.0 x1 port
- 2 USB 2.0 interfaces
- 3 SDIO interfaces for boot source, storage, etc



© OpenHW Group

10 & 11 December 2019

17

Section:

Open Source FPGA tools

Open Source and FPGAs

- 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

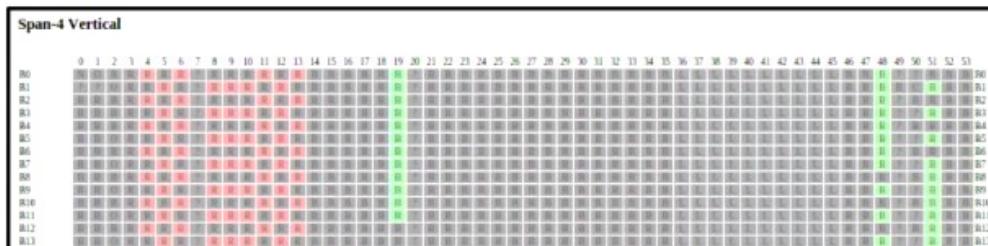


A Free and Open Source Verilog-to-Bitstream Flow for iCE40 FPGAs



by [Clifford](#)

Some screenshots from IceStrom Docs:



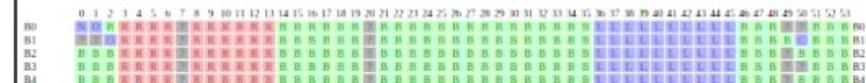
(2 17)	(3 17)	(4 17)	(5 17)	(6 17)
LOGIC Tile (2 16)	RAMT Tile (3 16)	LOGIC Tile (4 16)	LOGIC Tile (5 16)	LOGIC Tile (6 16)
LOGIC Tile (2 15)	RAMB Tile (3 15)	LOGIC Tile (4 15)	LOGIC Tile (5 15)	LOGIC Tile (6 15)
LOGIC Tile (2 14)	RAMT Tile (3 14)	LOGIC Tile (4 14)	LOGIC Tile (5 14)	LOGIC Tile (6 14)

Configuration Bitmap

A LOGIC Tile has 64x config bits in 16 groups of 54 bits each:

B0[53:0], B1[53:0], B2[53:0], B3[53:0], B4[53:0], B5[53:0], B6[53:0], B7[53:0],

B8[53:0], B9[53:0], B10[53:0], B11[53:0], B12[53:0], B13[53:0], B14[53:0], B15[53:0]



Open Source and FPGAs

- 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

Project Trellis and nextpnr FOSS FPGA flow for the Lattice ECP5

Project Trellis & nextpnr

FOSS Tools for ECP5 FPGAs

David Shah
@fpga_dave

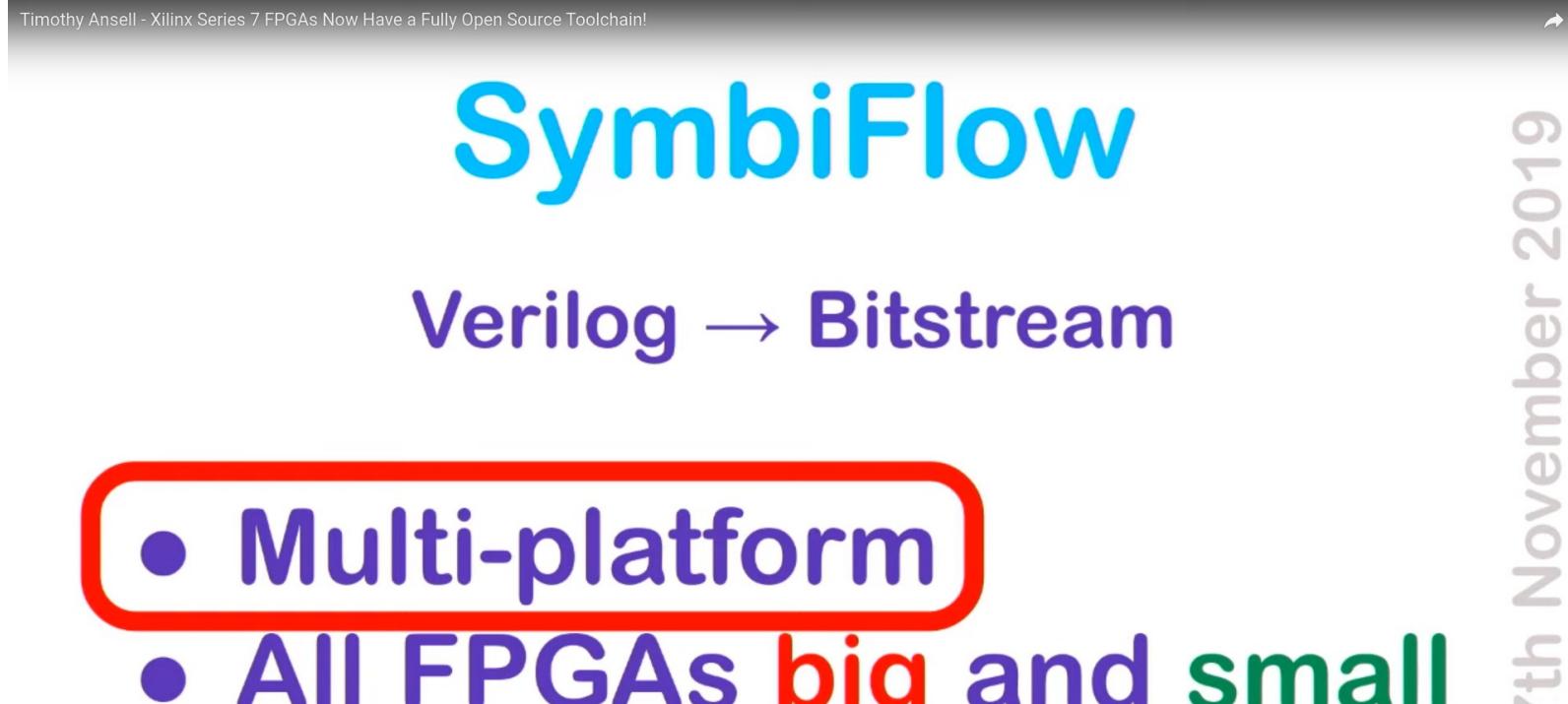
Symbiotic EDA || Imperial College London

FOSDEM 19
org



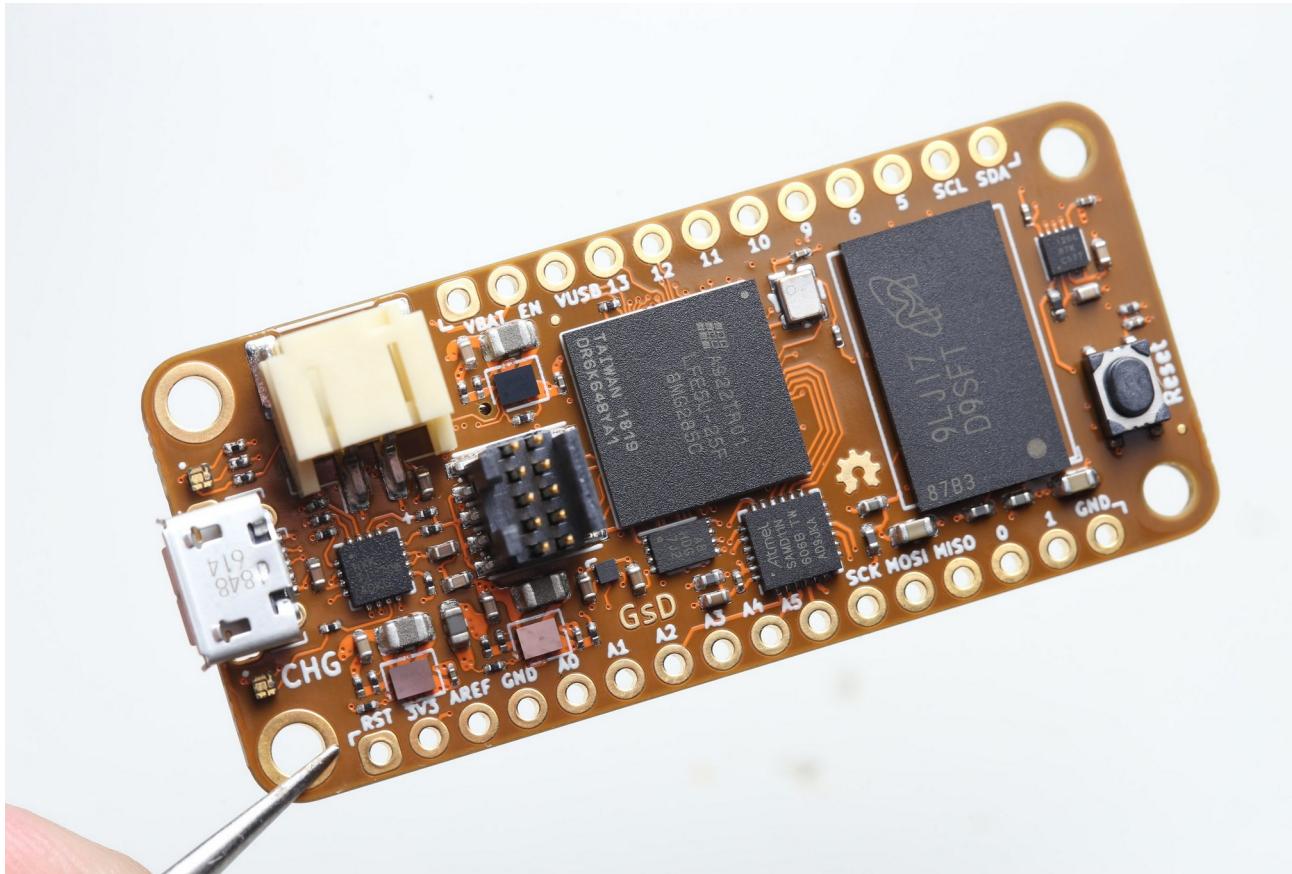
Open Source and FPGAs

- Open Source toolchains for FPGAs!
 - Project X-Ray and SymbiFlow for Xilinx Series 7
 - [Timothy 'mithro' Ansell](#): “Xilinx Series 7 FPGAs Now Have a Fully Open Source Toolchain!” (*almost*)
 - [youtube.com/watch?v=EHePto95qoE](https://www.youtube.com/watch?v=EHePto95qoE)



Open Source and FPGAs

- Open Source Hardware boards with Lattice ECP5 FPGA with open RISC-V “soft” CPU:
 - Orange Crab by Greg Davill
 - <https://github.com/gregdavill/OrangeCrab>





Greg @ #36c3
@GregDavill



Replying to @mithro @pdp7 and 2 others

Done. 😊

```
File Edit View Terminal Tabs Help
SRAM:      4KB
L2:        8KB
MAIN-RAM: 131072KB

----- Initialization -----
Initializing SDRAM...
SDRAM now under software control
Read leveling:
m0, b0: |11100000| delays: 01+-01
best: m0, b0 delays: 01+-01
m1, b0: |11100000| delays: 01+-01
best: m1, b0 delays: 01+-01
SDRAM now under hardware control
Memtest OK

----- Boot -----
Booting from serial...
Press Q or ESC to abort boot completely.
sL5DdSMmkekro
[LXTERM] Received firmware download request from the device.
[LXTERM] Uploading buildroot/Image to 0xc0000000 (4545524 bytes)...
[LXTERM] Upload complete (85.6KB/s).
[LXTERM] Uploading buildroot/rootfs.cpio to 0xc0800000 (8029184 bytes)...
[ 0:43 ] 1.2K views => | 98%
```



Open Source and FPGAs

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

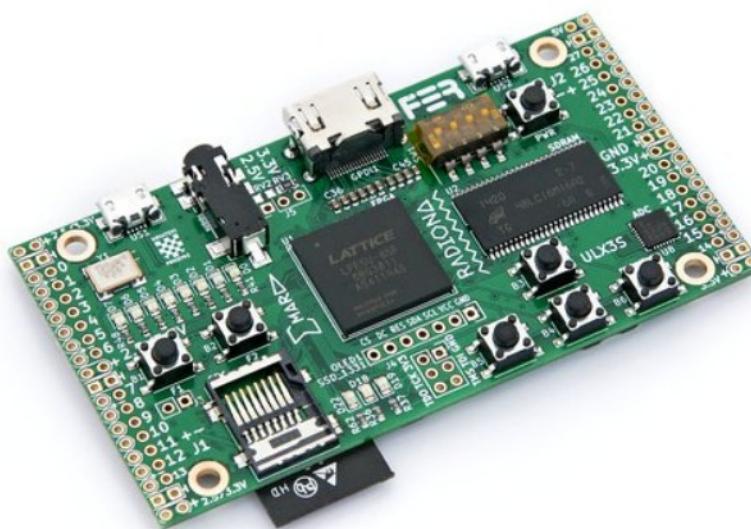
ULX3S

A powerful, open hardware ECP5 FPGA dev board

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

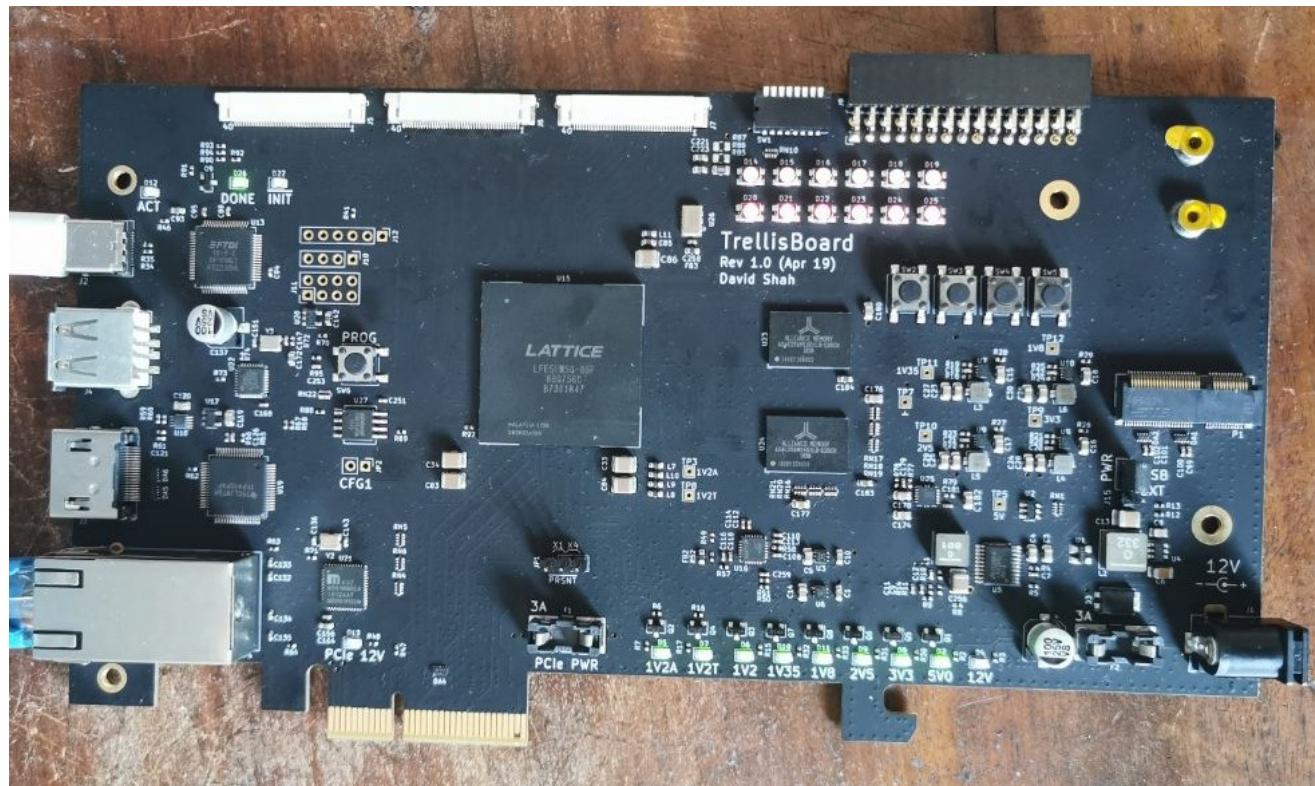
me@example.com

Subscribe



Open Source and FPGAs

- Open Source Hardware boards with Lattice ECP5 FPGA with open RISC-V “soft” CPU:
 - David Shah's Trellis board (Ultimate ECP5 Board)
 - <https://github.com/daveshah1/TrellisBoard>

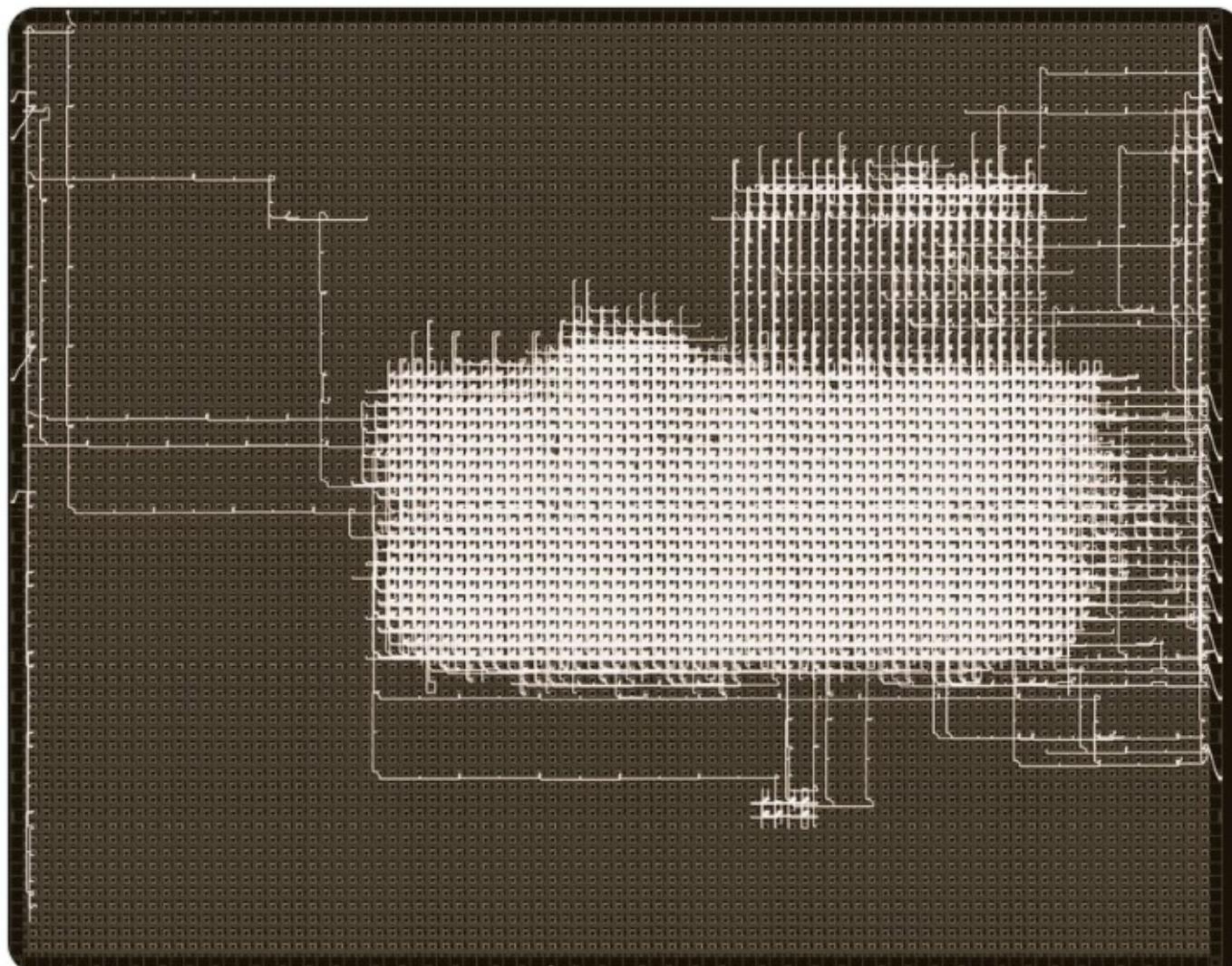




OSS FPGA & EDA tools
@ico_TC



This is how a Linux capable core looks like on an
FPGA. [#nextpnratwork](#)

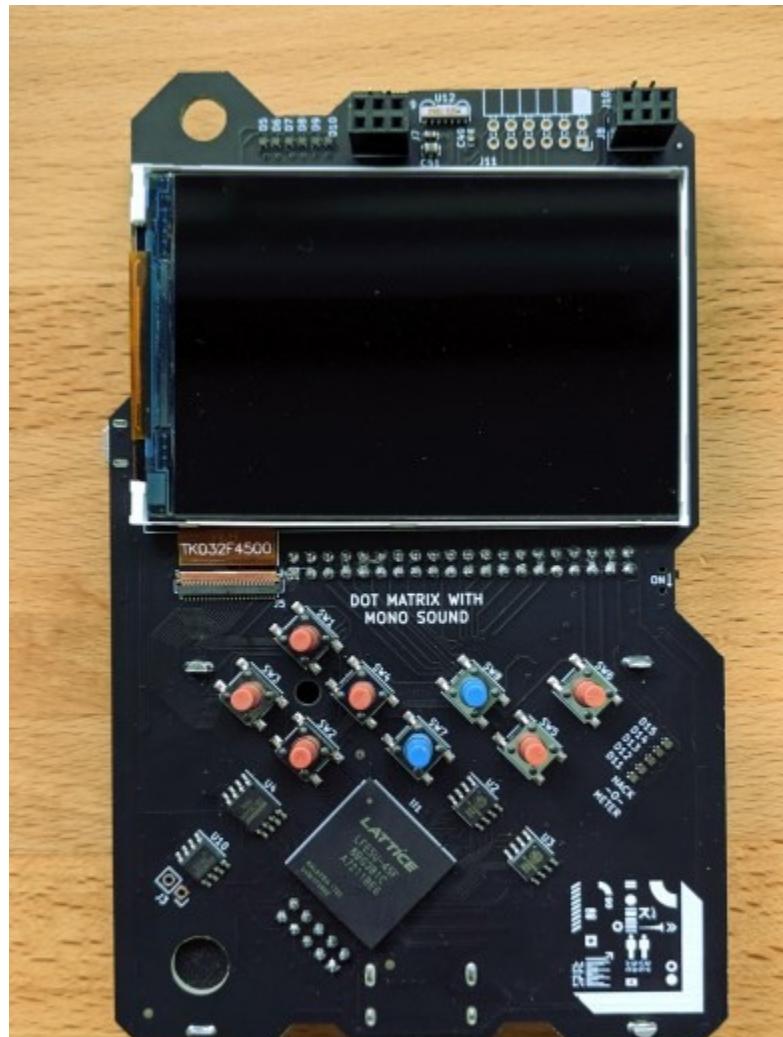


Section:

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”

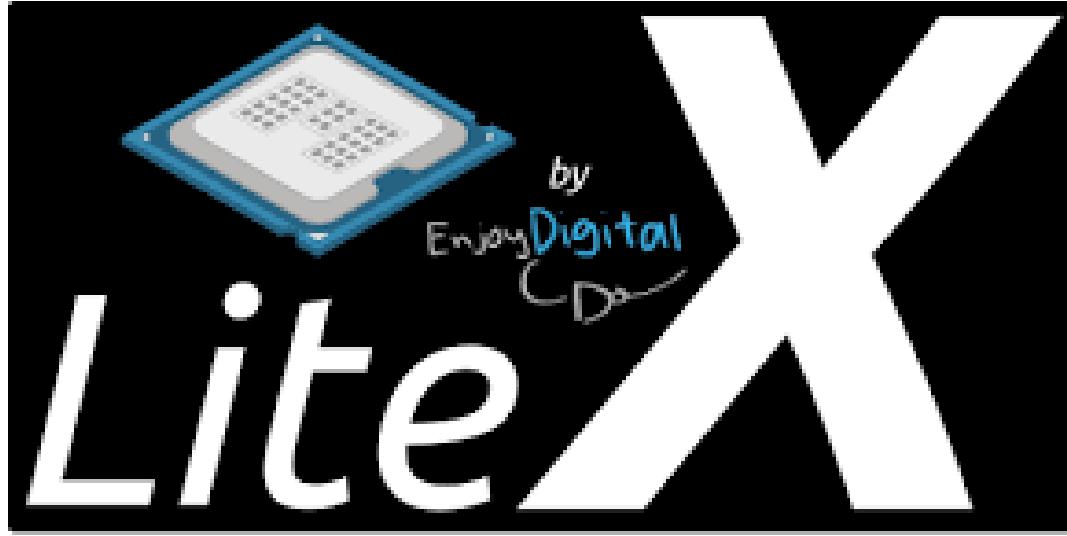
- Blog post: Hackaday Supercon badge boots Linux using SDRAM cartridge
 - <https://blog.oshpark.com/2019/12/20/boot-linux-on-this-hackaday-supercon-badge-with-this-sdram-cartridge/>
- Michael Welling (@QwertyEmdedded), Tim Ansell (@mithro), Sean Cross (@xobs), Jacob Creedon (@jacobcreedon)
- First attempt: use the built-in 16MB SRAM... no luck :(
 - (*though xobs now might have a way to do it*)

“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!



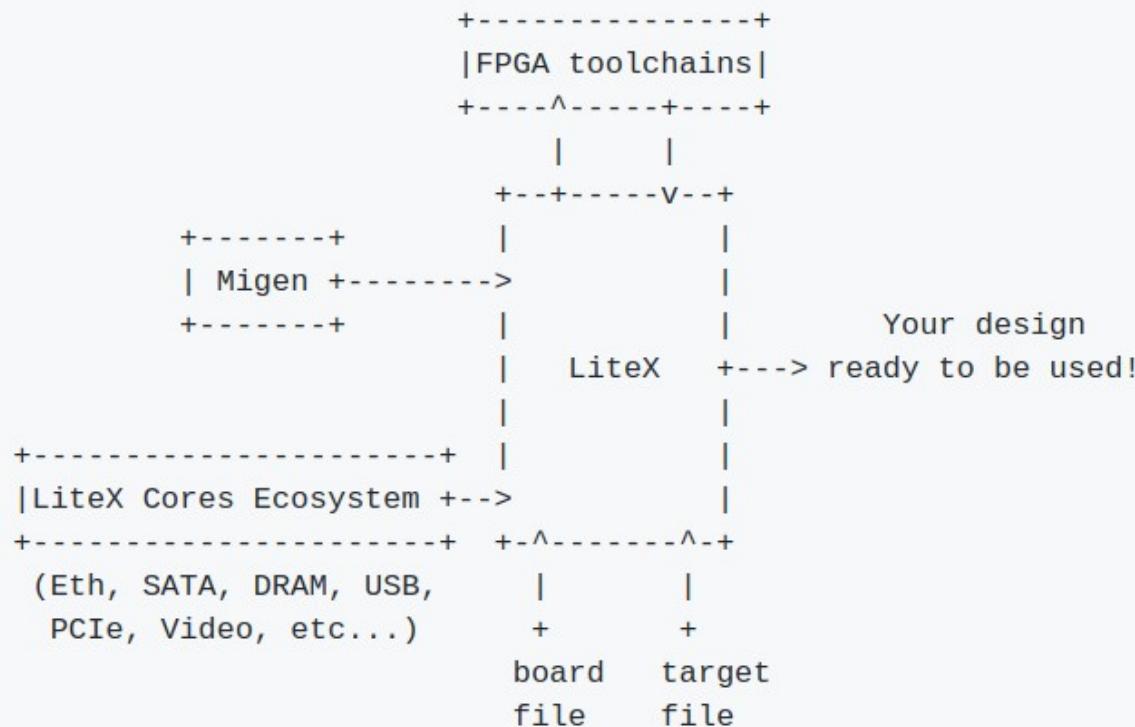




Build your hardware, easily!

- 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>

Typical LiteX design flow:



LiteX already supports various softcores CPUs: LM32, Mor1kx, PicoRV32, VexRiscv and is compatible with the LiteX's Cores Ecosystem:

Name	Build Status	Description
LiteDRAM	build passing	DRAM
LiteEth	build passing	Ethernet
LitePCIe	build passing	PCIe
LiteSATA	build passing	SATA

Linux on LiteX-VexRiscv

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



- 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

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 cartridge 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

```

[Code](#)[Issues 17](#)[Pull requests 2](#)[Actions](#)[Security](#)[Insights](#)

add the Hackaday Supercon ECP5 badge #68

Mergedenjoy-digital merged 1 commit into [litex-hub:master](#) from [pdp7:master](#)  21 days ago[Conversation 2](#)[Commits 1](#)[Checks 0](#)[Files changed 1](#)

pdp7 commented 22 days ago

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 cartridge 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

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
```

[Code](#)[Issues 21](#)[Pull requests 3](#)[Actions](#)[Projects 0](#)[Wiki](#)[Security](#)[In](#)

add 32MB SDRAM for hadbadge #97

Mergedenjoy-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):
190      technology_timings = _TechnologyTimings(tREFI=64e6/8192, tWTR=(2, None), tCCD=(1,
191      speedgrade_timings = {"default": _SpeedgradeTimings(tRP=18, tRCD=18, tWR=12, tRFC=
192
193 +     class AS4C32M8(SDRAMModule):
194 +         memtype = "SDR"
195 +         # geometry
196 +         nbanks = 4
197 +         nrows  = 8192
198 +         ncols   = 1024
199 +         # timings
200 +         technology_timings = _TechnologyTimings(tREFI=64e6/8192, tWTR=(2, None), tCCD=(1,
201 +         speedgrade_timings = {"default": _SpeedgradeTimings(tRP=20, tRCD=20, tWR=15, tRFC=
193
194     # DDR -----
195
```



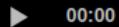
- Opened GitHub issue:
 - optimize performance on Hackaday Badge #35
 - <https://github.com/litex-hub/litex-boards/issues/35>
- Now 10x faster!
 - <https://asciinema.org/a/Pcm3vd1BEdEKY9srYX6MsNfCE>
 - Thanks to [enjoy-digital](#)



Asciicast updated.

```
[ 7.896093] Initramfs unpacking failed: junk in compressed archive
[ 7.953813] workingset: timestamp_bits=30 max_order=13 bucket_order=0
[ 9.236463] Block layer SCSI generic (bsg) driver version 0.4 loaded (major 253)
[ 9.239004] io scheduler mq-deadline registered
[ 9.240102] io scheduler kyber registered
[ 13.977920] f0001000.serial: MMIO 0xf0001000 (irq = 0, base_baud = 0) is a liteuart
[ 13.980290] printk: console [liteuart0] enabled
[ 13.980290] printk: console [liteuart0] enabled
[ 13.982965] printk: bootconsole [sbi0] disabled
[ 13.982965] printk: bootconsole [sbi0] disabled
[ 14.058778] libphy: Fixed MDIO Bus: probed
[ 14.074959] i2c /dev entries driver
[ 14.247461] NET: Registered protocol family 10
[ 14.307974] Segment Routing with IPv6
[ 14.315214] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
[ 14.455698] Freeing unused kernel memory: 140K
[ 14.457905] This architecture does not have kernel memory protection.
[ 14.459170] 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... [ 23.063050] random: dd: uninitialized urandom read (512 bytes read)
done.
Starting network: OK
Starting dropbear sshd: [ 27.336210] random: dropbear: uninitialized urandom read (32 bytes read)
OK
```

Welcome to Buildroot



00:00



Linux boots on Hackaday Supercon FPGA badge [10x faster!]

[Code](#)[Issues 2](#)[Pull requests 1](#)[Actions](#)[Projects 0](#)[Wiki](#)[Security](#)[Insights](#)

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



...

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.

[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

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  
Lo:
```



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):

26 26 }

```
27      27      csr_map.update(SoCCore.csr_map)
```

28

```
-      def __init__(self, platform, clk_freq, l2_size=8192, **kwargs):
```

```
29 +     def __init__(self, platform, clk_freq, l2_size=8192, l2_data_width=128, **kwargs):
```

```
30      30          SoCCore.__init__(self, platform, clk_freq, **kwargs)
```

```
31      31          if not self.integrated_main_ram_size:
```

```
32      32          if self.cpu_type is not None and self.csr_data_width > 32:
```

```
raise NotImplementedError("BIOS supports SDRAM initialization only for c
```

34 - self.l2_size = l2_size

34 + self.l2_size = l2_size

35 + self.l2_data_width = l2_data_width

35 36

36 37 self, s dram phy = []

```
37     38         self.wh_sdram_ifs = []
```