Slides: https://tinyurl.com/rv-linux-21

The Future of Linux on RISC-V

Drew Fustini <dfustini@baylibre.com>

$ whoami

- Linux kernel developer, BayLibre
  - Five team members presenting at ELC this week
  - Neil Armstrong yesterday: Introduction to Pin Muxing and GPIO Control Under Linux
  - Bartosz Golaszewski tomorrow, Wednesday, at 10:30am
    "Plan to Throw One Away" – Pitfalls of API Design for Low-level User-space Libraries and Kernel Interfaces
  - Kevin Hilman & Alexandre Mergnat tomorrow, Wednesday, at 2:45pm
    A New user(space): Adding RISC-V Support to Zephyr RTOS

$ whoami

- Board of Directors, BeagleBoard.org Foundation
  - BeagleV initiative to create open source hardware RISC-V boards
- Board of Directors, Open Source Hardware Association (OSHWA)
  - OSHW Certification Program (certification.oshwa.org)
- RISC-V Ambassador for RISC-V International

RISC-V this week *(virtual)*

- Perf on RISC-V: The Past, the Present and the Future
  - Monday, September 27, 11:15am - 12:05pm (Atish Patra & Anup Patel)
- Building a Low-key XIP-enabled RISC-V Linux System
  - Tuesday, September 28, 4:00pm - 4:50pm (Vitaly Vul)
- Initializing RISC-V: A Guided Tour for ARM Developers
  - Tuesday, September 28, 5:00pm - 5:50pm (Ahmad Fatoum & Rouven Czerwinski)
- A New user(space): Adding RISC-V Support to Zephyr RTOS
  - Wednesday, September 29, 2:45pm - 3:35pm (Kevin Hilman & Alexandre Mergnat)

RISC-V this week *(in-person)*

- How the Fastest Growing Open Hardware Project is Leveraging Visibility *[..]*
  - Tuesday, September 28, 11:25am - 11:50am (Kim McMahon, RISC-V International)
- Growing Diversity in Open Hardware: It's a Task for All of Us!
  - Tuesday, September 28, 12:25pm - 12:50pm (Kim McMahon, RISC-V International)
  - Open Hardware Diversity Alliance formed in August for "professional advancement of women and underrepresented individuals in open source hardware."
- Open Hardware: Skyrocketing Momentum and Global Adoption from Embedded to Enterprise
  - **Tuesday, September 28, 4:00pm - 4:50pm (Calista Redmond, CEO RISC-V Intl.)**

RISC-V this week *(in-person)*

- Open Software, Open Hardware with RISC-V and Zephyr communities
  - Thursday, September 30, 9:00 AM – 12:30 PM
  - Pre-registration is required. To register, add it to your OSS + ELC registration
  - 9:00 - 9:30: Coffee networking
  - 9:30 - 9:45: Welcome / keynote
  - 9:45 - 11:00: Breakouts to tables for table talks / BoF
  - 11:00 - 12:00: Demos and lightning talks
  - 12:00 - 12:30: Coffee networking

RISC-V *last* week

- Linux Plumbers Conference: RISC-V microconference
  - Live stream on YouTube, Detailed notes with links
  - Sessions
    - The RISC-V platform specification
    - ACPI for RISC-V
    - What's the problem with D1 Linux upstream?
    - Puzzle for RISC-V ifunc
    - Towards continuous improvement of code-generation for RISC-V


RISC-V
SUMMIT
December 6-8, 2021
San Francisco, CA

RISC-V (virtual) meetups around the world

AUSTIN AREA RISC-V GROUP, TX
United States

BAY AREA RISC-V GROUP, CA
United States

BOSTON RISC-V GROUP, MA
United States

NEW YORK CITY RISC-V GROUP, NY
United States

ROCKY MOUNTAIN AREA RISC-V GROUP, CO
United States

SAN DIEGO RISC-V GROUP, CA
United States

SEATTLE RISC-V GROUP, WA
United States

TORONTO RISC-V GROUP, ON
Canada

TWIN CITIES RISC-V GROUP, MN
United States

BENGALURU RISC-V GROUP, KA
India

HANGZHOU RISC-V GROUP, ZHEJIANG
China

HSINCHU RISC-V GROUP
Taiwan

KARACHI RISC-V GROUP, SINDH
Pakistan

MANILA-TAGUIG RISC-V GROUP, NCR
Philippines

NAVI MUMBAI RISC-V GROUP, MH
India

OSAKA RISC-V GROUP
Japan

PUNE RISC-V GROUP, MH
India

SHANGHAI RISC-V GROUP
China

TOKYO RISC-V GROUP
Japan

TOKYO-TECHNICAL STUDY RISC-V GROUP
Japan

BRISTOL RISC-V GROUP, ENGLAND
United Kingdom

CAMBRIDGE RISC-V GROUP, ENGLAND
United Kingdom

DUISBURG RISC-V GROUP, NRW
Germany

GÖTEBORG RISC-V GROUP, VÄSTRA GÖTALAND COUNTY
Sweden

ISRAEL RISC-V GROUP, TEL AVIV DISTRICT
Israel

LONDON RISC-V GROUP, ENGLAND
United Kingdom

LONDON-OPEN SOURCE RISC-V GROUP, ENGLAND
United Kingdom

MUNICH RISC-V GROUP, BY
Germany

VIENNA RISC-V GROUP
Austria

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

RISC-V Open Hours

- Bi-weekly meetup to provide the opportunity for the community to interact in real-time, with a particular focus on RISC-V support in open source software projects and RISC-V development boards.
- Join the mailing list for announcements and discussion
- Wednesday, Oct 13, 7:00 PM (US PDT) which is Thursday morning in Asia
- Wednesday, Nov 3, 8:00 AM (US PDT) which is European late afternoon

RISC-V: a Free and Open ISA

- Started in 2010 by computer architecture researchers at UC Berkeley
  - Krste Asanovic - RISC-V: The Next Ten Years
- Why "RISC"?
  - RISC = Reduced Instruction Set Computer
- Why "V"?
  - 5th RISC instruction set to come of out UC Berkeley
- Why is it "Free and Open"?
  - Specifications licensed as Creative Commons Attribution 4.0 International

What is different about RISC-V?

- Simple, clean-slate design
  - Avoids micro-architecture dependent features
- Small standard base, with multiple standard extensions
  - Suitable for everything from tiny microcontrollers to supercomputers
- Stable
  - Base and standard extensions are frozen
  - Additions via optional extensions, not new versions of base ISA

RISC-V Base Integer ISA

- RV32I: 32-bit
  - less than 50 instructions needed!
- RV64I: 64-bit
  - Most important for Linux
- RV128I: 128-bit
  - Future-proof address space

imm[31:12]				rd	0110111	LUI
imm[31:12]				rd	0010111	AUIPC
imm[20|10:1|11|19:12]				rd	1101111	JAL
imm[11:0]		rs1	000	rd	1100111	JALR
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ
imm[12|10:5]	rs2	rs1	001	imm[4:1|11]	1100011	BNE
imm[12|10:5]	rs2	rs1	100	imm[4:1|11]	1100011	BLT
imm[12|10:5]	rs2	rs1	101	imm[4:1|11]	1100011	BGE
imm[12|10:5]	rs2	rs1	110	imm[4:1|11]	1100011	BLTU
imm[12|10:5]	rs2	rs1	111	imm[4:1|11]	1100011	BGEU
imm[11:0]		rs1	000	rd	0000011	LB
imm[11:0]		rs1	001	rd	0000011	LH
imm[11:0]		rs1	010	rd	0000011	LW
imm[11:0]		rs1	100	rd	0000011	LBU
imm[11:0]		rs1	101	rd	0000011	LHU
imm[11:5]	rs2	rs1	000	imm[4:0]	0100011	SB
imm[11:5]	rs2	rs1	001	imm[4:0]	0100011	SH
imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	SW
imm[11:0]		rs1	000	rd	0010011	ADDI
imm[11:0]		rs1	010	rd	0010011	SLTI
imm[11:0]		rs1	011	rd	0010011	SLTIU
imm[11:0]		rs1	100	rd	0010011	XORI
imm[11:0]		rs1	110	rd	0010011	ORI
imm[11:0]		rs1	111	rd	0010011	ANDI
0000000	shamt	rs1	001	rd	0010011	SLLI
0000000	shamt	rs1	101	rd	0010011	SRLI
0100000	shamt	rs1	101	rd	0010011	SRAI
0000000	rs2	rs1	000	rd	0110011	ADD
0100000	rs2	rs1	000	rd	0110011	SUB
0000000	rs2	rs1	001	rd	0110011	SLL
0000000	rs2	rs1	010	rd	0110011	SLT
0000000	rs2	rs1	011	rd	0110011	SLTU
0000000	rs2	rs1	100	rd	0110011	XOR
0000000	rs2	rs1	101	rd	0110011	SRL
0100000	rs2	rs1	101	rd	0110011	SRA
0000000	rs2	rs1	110	rd	0110011	OR
0000000	rs2	rs1	111	rd	0110011	AND
fm	pred | succ	rs1	000	rd	0001111	FENCE
000000000000		00000	000	00000	1110011	ECALL
000000000001		00000	000	00000	1110011	EBREAK

RISC-V Standard Extensions

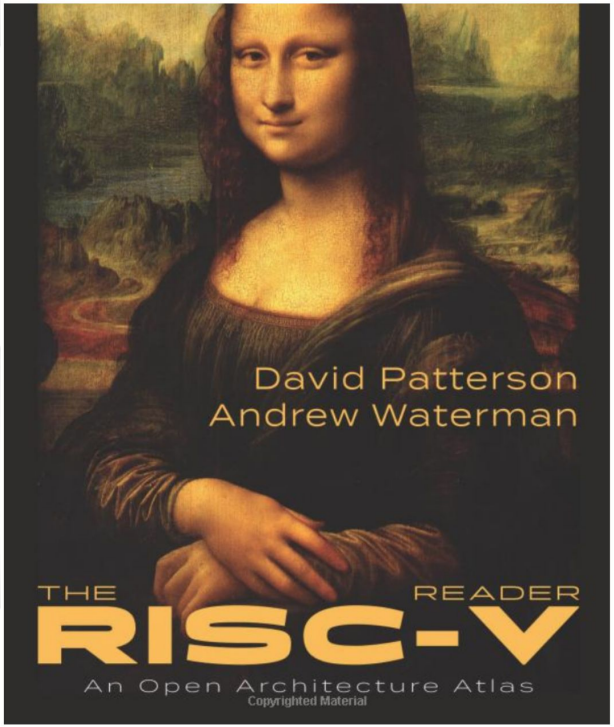
- M: integer multiply/divide
- A: atomic memory operations
- F, D, Q: floating point, double-precision, quad-precision
- **G**: "general purpose" ISA, equivalent to IMAFD
- **C**: compressed instructions conserve memory & cache like ARM Thumb
- Linux distros like Debian and Fedora target **RV64GC**

Learn more about RISC-V

- Get up-to-speed quick with

  the **RISC-V Reader**:

  riscvbook.com

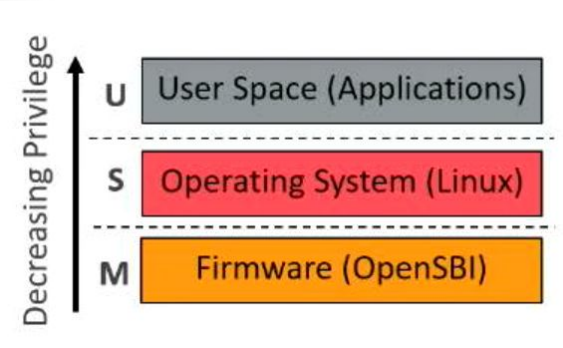


“Is RISC-V an Open Source processor?”

- RISC-V is a set of specifications under an open source license
- RISC-V implementations can be open source or proprietary
- **Open specifications make open source implementations possible**
- An open ISA enables open source processor implementations

RISC-V Privileged Architecture

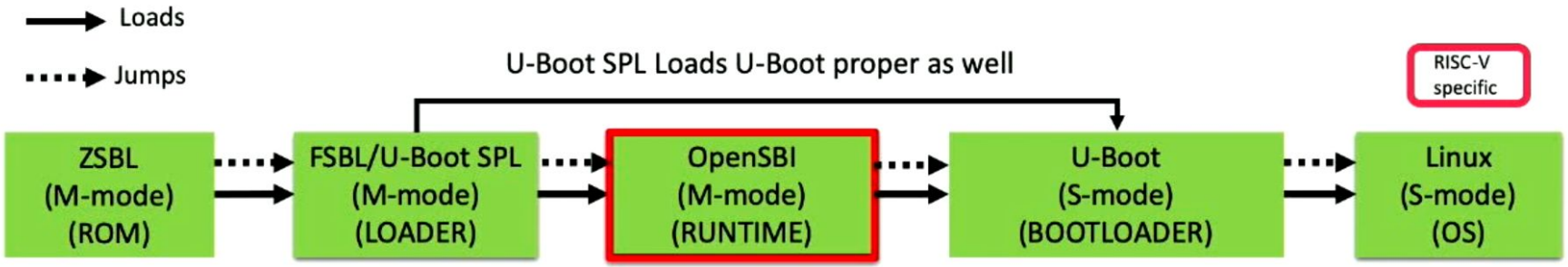
- Ratified specification
  - RISC-V Instruction Set Manual Volume II: Privileged Architecture
- Three privilege modes
  - User (U-mode): applications
  - Supervisor (S-mode): OS kernel
  - Machine (M-mode): firmware



(source: Co-developing RISC-V Hypervisor Support, Anup Patel)

RISC-V Boot Flow

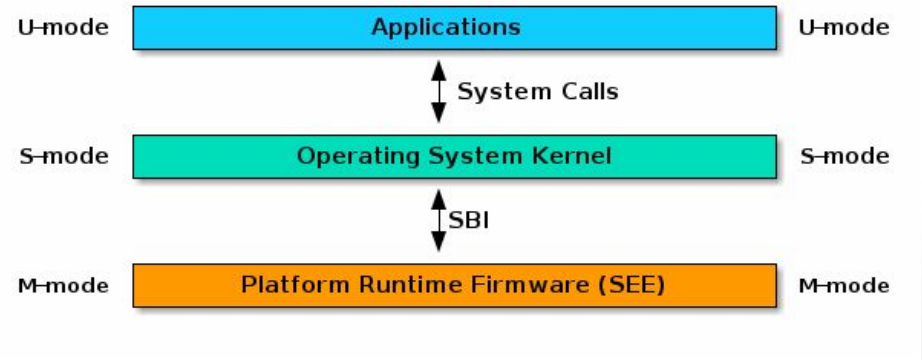
- M-mode: boot ROM and first-stage bootloader
- S-Mode: U-Boot loads and jumps to the Linux kernel
- Similar flow to ARM SoC but something new in the middle: SBI



(source: RISC-V software ecosystem in 2020, Atish Patra)

What is SBI?

- Supervisor Binary Interface is a non-ISA RISC-V specification
- The calling convention between S-mode and M-mode
- allows S-mode software like the Linux to be portable across RISC-V implementations by abstracting platform specific functionality.

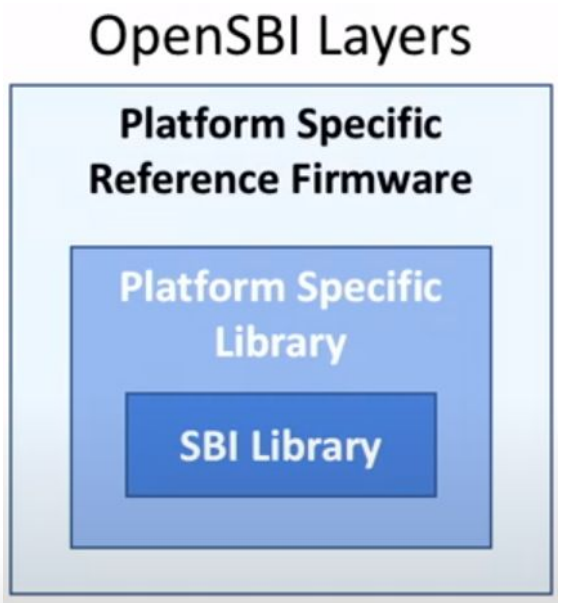


What is SBI?

- SBI is required by the UNIX-Class Platform Specification
  - Mailing list: tech-unixplatformspec
  - This will be replaced by upcoming RISC-V Platform Specification
- Base Extension
  - query SBI specification version and implementation
  - query machine vendor, architecture and implementation
- Timer Extension, IPI Extension, RFENCE Extension

What is OpenSBI?

- OpenSBI is an open source SBI implementation
- Layers of implementation
- Provides run-time service in M-mode



(source: OpenSBI Deep Dive, Anup Patel)

OpenSBI Generic Platform

- Preference is to use Generic platform when possible
- Device Tree based platform where all platform specific functionality is based on DT passed by the previous boot stage.
- Generic platform allows us to use the same OpenSBI firmware binaries across a variety of emulators and dev boards
- RISC-V systems using Generic Platform
    - QEMU, Spike simulator, SiFive HiFive Unleashed, Alibaba T-HEAD C9xx based boards

Hypervisor extension

- Hypervisor Supervisor (HS-mode) and Virtualized Supervisor (VS-mode)
  - Included in Privileged 1.12 spec, currently in [45 day public review before ratification]

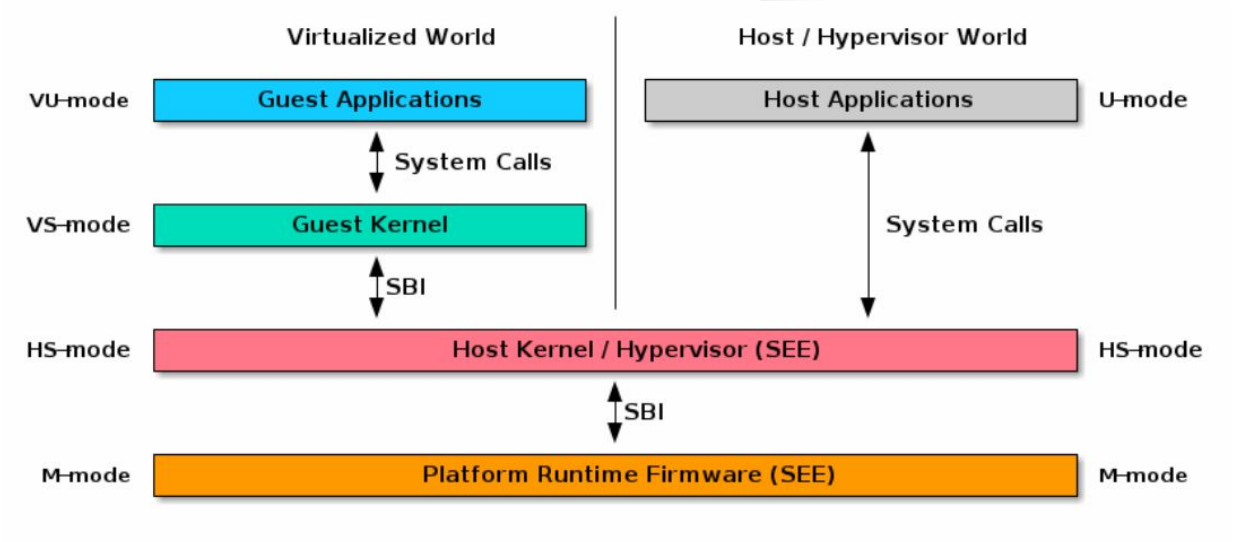


Figure 2. RISC-V System with H-extension

What is new in SBI?

- v0.3
  - Suspend added to Hart State Management (HSM)
  - Performance Monitoring Unit (PMU)
  - System reset extension
- NOTE: Hart is a hardware thread
  - Hardware execution context that contains state mandated by ISA: PC and registers
  - My laptop has 4 cores, each of core has two hyperthreads, so it could be described as having 8 harts if it was RISC-V (e.g. the number of penguins on the boot screen)

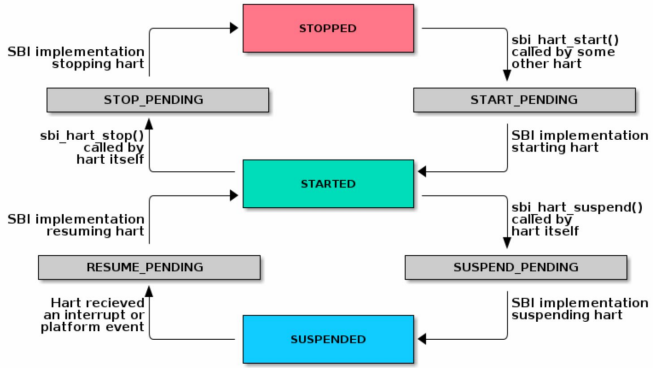
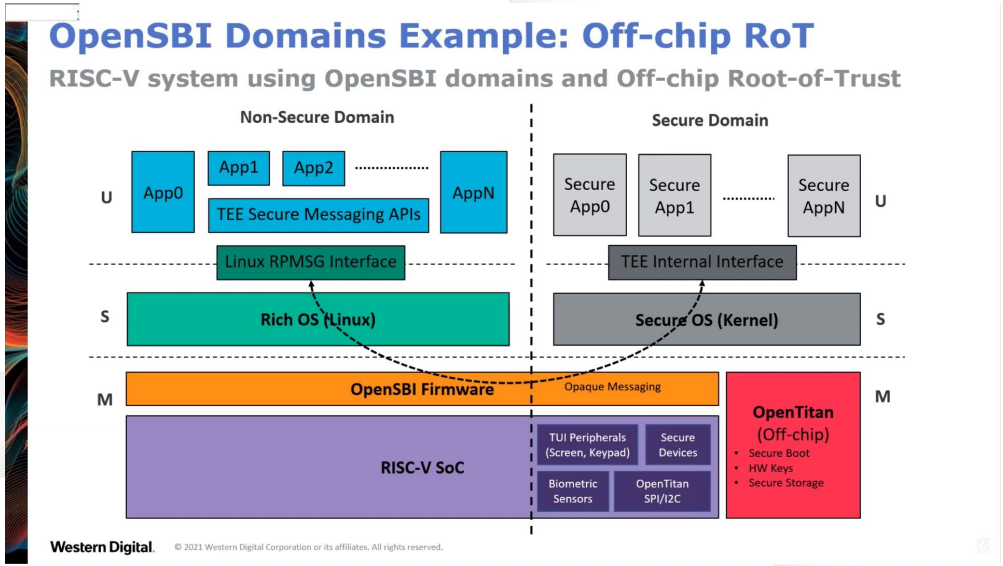


Figure 3. SBI HSM State Machine

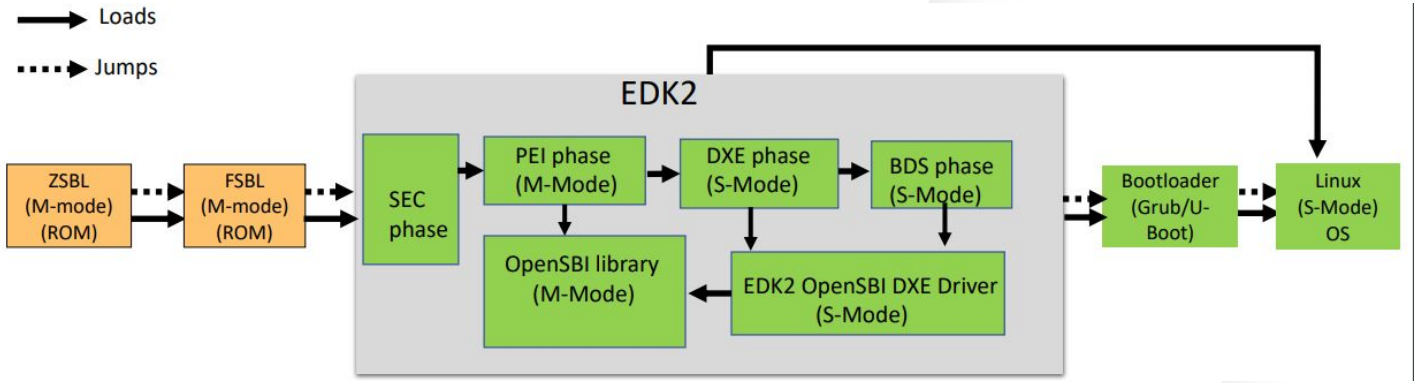
OpenSBI Domain Support

- An OpenSBI domain is a system-level partition (subset) of underlying hardware having its own memory regions and HARTs
- Talk by Anup Patel



UEFI Support

- U-Boot and TianoCore edk2 both have UEFI implementations on RISC-V
- Grub2 can be an UEFI payload on RISC-V
- UEFI support for RISC-V added in Linux 5.10



(source: Introduction to RISC-V Boot Flow, Atish Patra and Anup Patel)

RISC-V emulation in QEMU

- Support for RISC-V in mainline QEMU
  - QEMU can boot 32-bit and 64-bit mainline Linux kernel
  - QEMU sifive_u machine can boot same binaries as the HiFive Unleashed board
  - Draft versions of Hypervisor and Vector extensions supported
- Embedded Linux from scratch in 45 minutes on RISC-V!
  - Tutorial by By Michael Opdenacker from Bootlin at FOSDEM 2021

RISC-V in the Linux kernel

- Initial port by Palmer Dabbelt merged in Linux 4.15
  - Mailing list: linux-riscv@lists.infradead.org (archive)
- "What's missing in RISC-V Linux, and how YOU can help!"
  - Björn Töpel at Munich RISC-V meetup (jump to 43:25)
  - "It's a fun, friendly, and still pretty small community"
  - "A great way to learn the nitty gritty details of the Linux kernel"

Kernel feature support for riscv arch *(v5.12-rc2)*

```
pdp7@x1:~/linux$ ./Documentation/features/list-arch.sh riscv |ack TODO --passthru
#
# Kernel feature support matrix of the 'riscv' architecture:
#
      core/ cBPF-JIT             : TODO |                      HAVE_CBPF_JIT #  arch supports cBPF JIT optimizations
      core/ eBPF-JIT             :  ok  |                      HAVE_EBPF_JIT #  arch supports eBPF JIT optimizations
      core/ generic-idle-thread  :  ok  |             GENERIC_SMP_IDLE_THREAD #  arch makes use of the generic SMP idle thread facility
      core/ jump-labels          :  ok  |                HAVE_ARCH_JUMP_LABEL #  arch supports live patched, high efficiency branches
      core/ thread-info-in-task  :  ok  |                  THREAD_INFO_IN_TASK #  arch makes use of the core kernel facility to embedd thread_info in task_struct
      core/ tracehook            :  ok  |                 HAVE_ARCH_TRACEHOOK #  arch supports tracehook (ptrace) register handling APIs
     debug/ debug-vm-pgtable     :  ok  |           ARCH_HAS_DEBUG_VM_PGTABLE #  arch supports pgtable tests for semantics compliance
     debug/ gcov-profile-all     :  ok  |           ARCH_HAS_GCOV_PROFILE_ALL #  arch supports whole-kernel GCOV code coverage profiling
     debug/ KASAN                :  ok  |                    HAVE_ARCH_KASAN #  arch supports the KASAN runtime memory checker
     debug/ kcov                 :  ok  |                      ARCH_HAS_KCOV #  arch supports kcov for coverage-guided fuzzing
     debug/ kgdb                 :  ok  |                    HAVE_ARCH_KGDB #  arch supports the kGDB kernel debugger
     debug/ kmemleak             :  ok  |                 HAVE_DEBUG_KMEMLEAK #  arch supports the kernel memory leak detector
     debug/ kprobes              :  ok  |                       HAVE_KPROBES #  arch supports live patched kernel probe
     debug/ kprobes-on-ftrace    :  ok  |              HAVE_KPROBES_ON_FTRACE #  arch supports combined kprobes and ftrace live patching
     debug/ kretprobes           :  ok  |                    HAVE_KRETPROBES #  arch supports kernel function-return probes
     debug/ optprobes            : TODO |                     HAVE_OPTPROBES #  arch supports live patched optprobes
     debug/ stackprotector       :  ok  |                  HAVE_STACKPROTECTOR #  arch supports compiler driven stack overflow protection
     debug/ uprobes              :  ok  |                 ARCH_SUPPORTS_UPROBES #  arch supports live patched user probes
     debug/ user-ret-profiler    : TODO |           HAVE_USER_RETURN_NOTIFIER #  arch supports user-space return from system call profiler
        io/ dma-contiguous       :  ok  |                  HAVE_DMA_CONTIGUOUS #  arch supports the DMA CMA (continuous memory allocator)
  locking/ cmpxchg-local         : TODO |                   HAVE_CMPXCHG_LOCAL #  arch supports the this_cpu_cmpxchg() API
  locking/ lockdep               :  ok  |                    LOCKDEP_SUPPORT #  arch supports the runtime locking correctness debug facility
  locking/ queued-rwlocks        : TODO |               ARCH_USE_QUEUED_RWLOCKS #  arch supports queued rwlocks
  locking/ queued-spinlocks      : TODO |              ARCH_USE_QUEUED_SPINLOCKS #  arch supports queued spinlocks
     perf/ kprobes-event         :  ok  |      HAVE_REGS_AND_STACK_ACCESS_API #  arch supports kprobes with perf events
     perf/ perf-regs             :  ok  |                      HAVE_PERF_REGS #  arch supports perf events register access
     perf/ perf-stackdump        :  ok  |             HAVE_PERF_USER_STACK_DUMP #  arch supports perf events stack dumps
    sched/ membarrier-sync-core  : TODO |       ARCH_HAS_MEMBARRIER_SYNC_CORE #  arch supports core serializing membarrier
    sched/ numa-balancing        :  ok  |        ARCH_SUPPORTS_NUMA_BALANCING #  arch supports NUMA balancing
  seccomp/ seccomp-filter        :  ok  |             HAVE_ARCH_SECCOMP_FILTER #  arch supports seccomp filters
     time/ arch-tick-broadcast   :  ok  |              ARCH_HAS_TICK_BROADCAST #  arch provides tick_broadcast()
     time/ clockevents           :  ok  |                   !LEGACY_TIMER_TICK #  arch support generic clock events
     time/ context-tracking      :  ok  |               HAVE_CONTEXT_TRACKING #  arch supports context tracking for NO_HZ_FULL
     time/ irq-time-acct         :  ok  |             HAVE_IRQ_TIME_ACCOUNTING #  arch supports precise IRQ time accounting
     time/ virt-cpuacct          : TODO |             HAVE_VIRT_CPU_ACCOUNTING #  arch supports precise virtual CPU time accounting
       vm/ batch-unmap-tlb-flush: TODO |  ARCH_WANT_BATCHED_UNMAP_TLB_FLUSH #  arch supports deferral of TLB flush until multiple pages are unmapped
       vm/ ELF-ASLR              :  ok  |               ARCH_HAS_ELF_RANDOMIZE #  arch randomizes the stack, heap and binary images of ELF binaries
       vm/ huge-vmap             : TODO |                  HAVE_ARCH_HUGE_VMAP #  arch supports the arch_vmap_pud_supported() and arch_vmap_pmd_supported() VM APIs
       vm/ ioremap_prot          : TODO |                    HAVE_IOREMAP_PROT #  arch has ioremap_prot()
       vm/ PG_uncached           : TODO |                ARCH_USES_PG_UNCACHED #  arch supports the PG_uncached page flag
       vm/ pte_special           :  ok  |                 ARCH_HAS_PTE_SPECIAL #  arch supports the pte_special()/pte_mkspecial() VM APIs
       vm/ THP                   :  ok  |       HAVE_ARCH_TRANSPARENT_HUGEPAGE #  arch supports transparent hugepages
```

RISC-V in the Linux kernel

- KVM support for the Hypervisor spec (Anup Patel/Atish Patra)
  - [PATCH v20 00/17] KVM RISC-V Support
- Vector ISA support based on the draft vector extension (Greentime Hu)
  - [RFC PATCH v8 00/21] riscv: Add vector ISA support

Linux distro: Fedora

- "This project, informally called Fedora/RISC-V, aims to provide a complete Fedora experience on the RISC-V (RV64GC)"

4

(source: Fedora on RISC-V, Wei Fu)

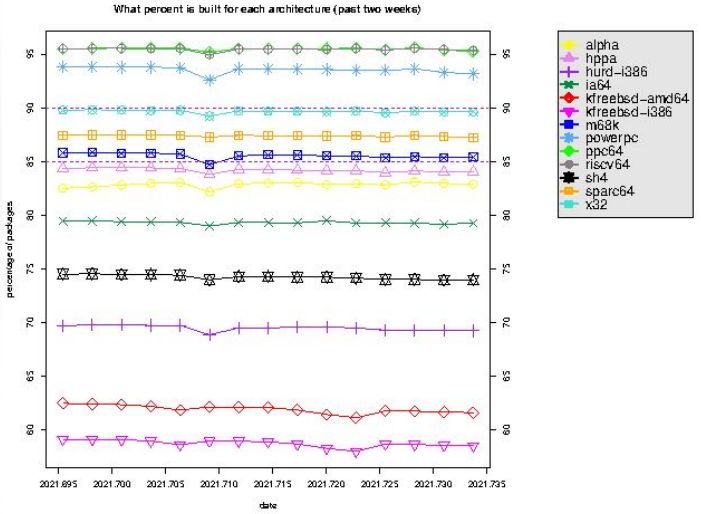
Linux distro: Fedora

- QEMU and libvirt
  - Fedora images can run on QEMU with graphics
- Real hardware
  - HiFive Unleashed, HiFive Unmatched, and more
- Installation instructions

(source: Fedora on RISC-V, Wei Fu)

Linux distro: Debian

- Port of Debian for the RISC-V architecture called riscv64
  - “a port in Debian terminology means to provide the software normally available in the Debian archive (over 20,000 source packages) ready to install and run”
- 95% of packages are built for RISC-V
  - The Debian port uses RV64GC as the hardware baseline and the lp64d ABI



Additional Linux distros

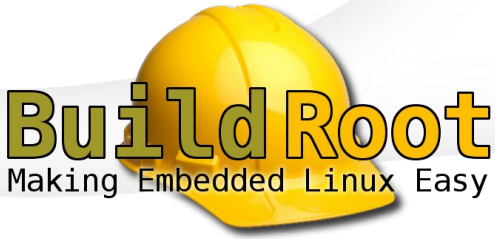
- Ubuntu 21.04 supports QEMU and SiFive boards
- OpenSuSE is under development and considered an early preview
- Gentoo has riscv64 stages available to download
- Arch Linux is in experimental development

OpenEmbedded / Yocto

- meta-riscv: general hardware-specific BSP overlay for RISC-V devices
  - works with different OpenEmbedded/Yocto distributions and layer stacks
  - Supports both QEMU and real boards like SiFive HiFive Unleashed

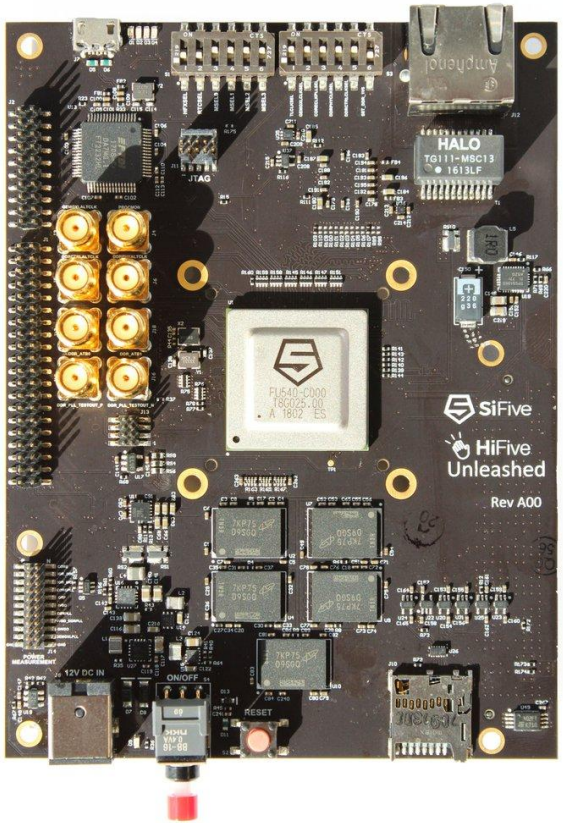
BuildRoot

- RISC-V port is now supported in the upstream BuildRoot project
- "Embedded Linux from scratch in 45 minutes (on RISC-V)"
  - Tutorial by Michael Opdenacker of Bootlin
  - Use Buildroot to compile OpenSBI, U-Boot, Linux and BusyBox
  - Boot the system in QEMU



SiFive Freedom Unleashed

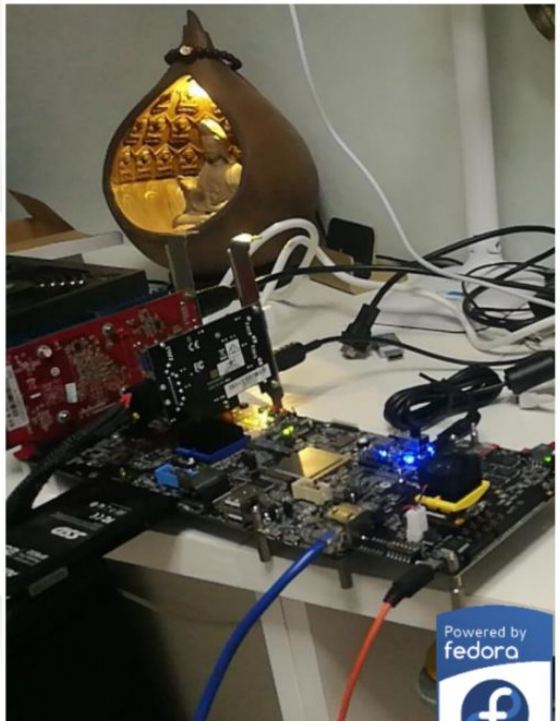
- The first Linux-capable RISC-V dev board
  - And the board design is Open Source Hardware!
- High performance compared to FPGA
  - FU540 SoC clocked over 10x faster than FPGA ‘soft’ cores
- Too expensive for widespread adoption
  - Sold for $999 on CrowdSupply and no longer available
  - FU540 SoC chip is not sold separately
  - SiFive core business is designing cores, not SoC’s or boards

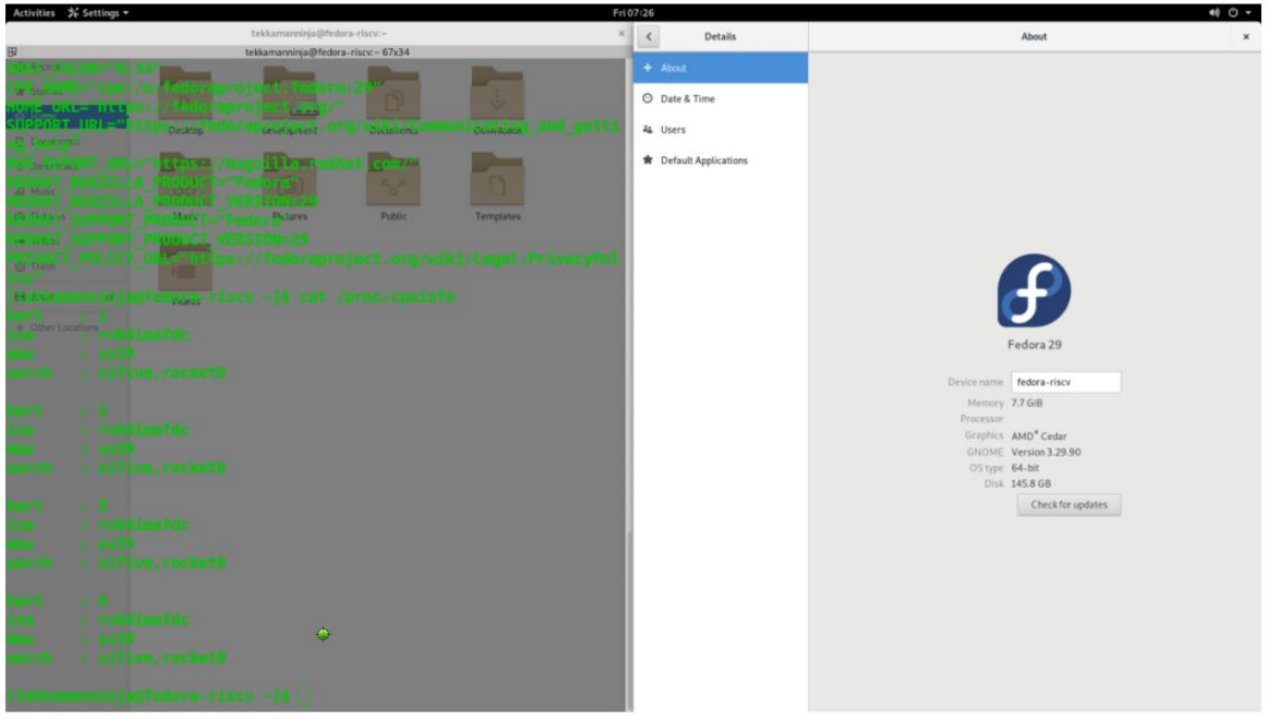


NOTE: ASIC often used to indicate that an SoC (System-on-Chip) has a “hard” processor core constructed by silicon fab instead of “soft” core in an FPGA

SiFive Freedom Unleashed

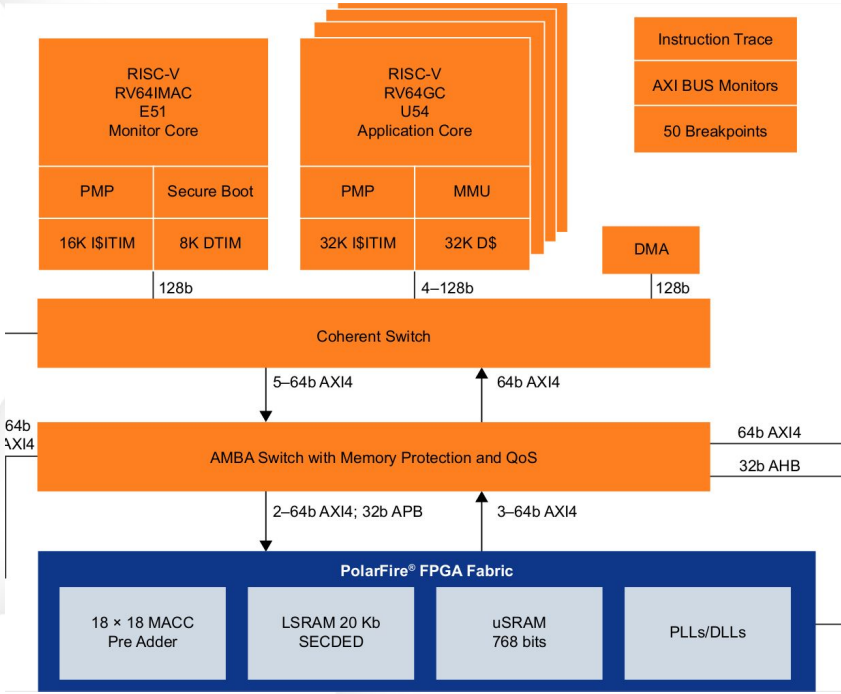
- Fedora GNOME image [running on Unleashed](#) with PCIe graphics card





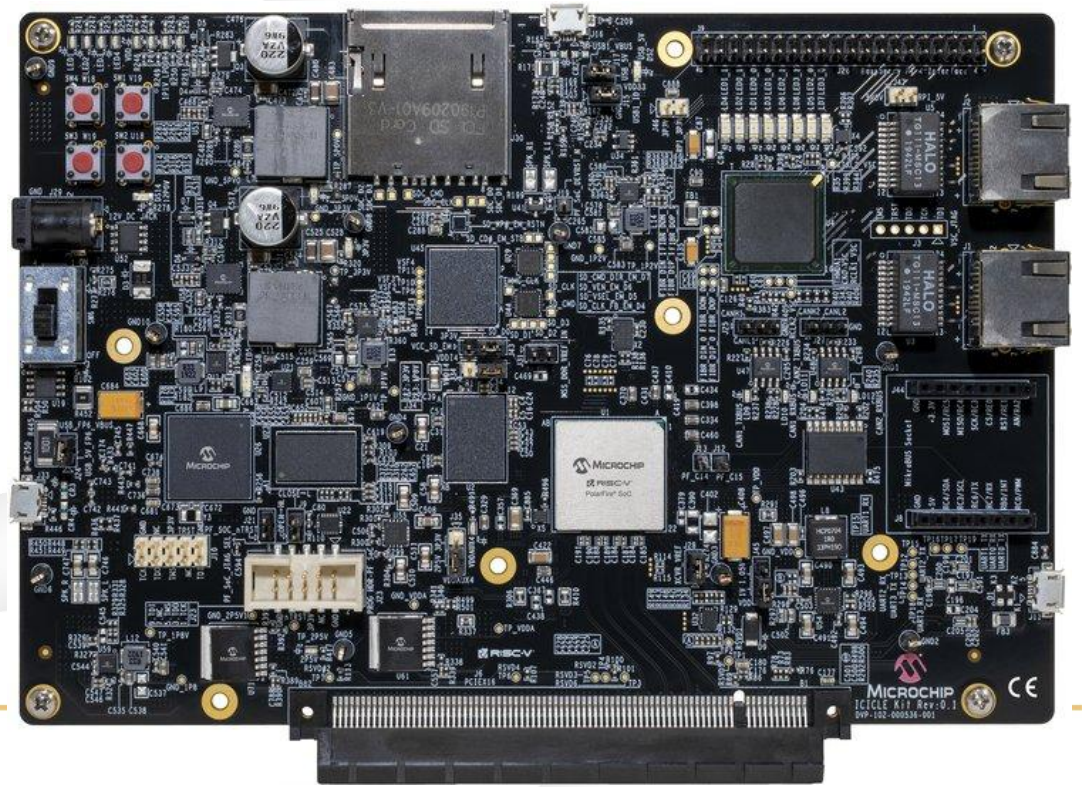
Microchip PolarFire SoC

- Same cores as the SiFive FU540 but adds a FPGA fabric
  - 4x 667 MHz U54 cores, 1x E51 core
  - FPGA with 25k to 460k logic elements (LEs)
  - Supports DDR4 and PCIe Gen2
- Full commercial product family
  - Available from distributors
  - From the former Microsemi business unit



Microchip Icicle board

- PolarFire SoC dev board
  - $499 on CrowdSuppy
  - MPFS250T-FCVG484EES
  - 600 MHz clock RISC-V cores
  - 254K logic element FPGA
- Memory
  - 2 GB LPDDR4 x 32
  - 8 GB eMMC flash and SD card

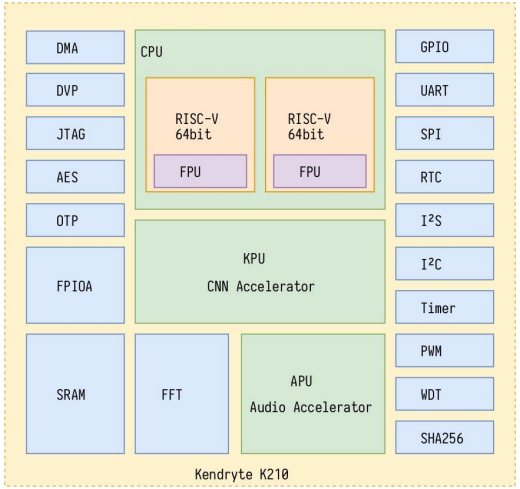


Kendryte K210

- 400MHz dual core RV64GC
  - 8MB SRAM and no DRAM interface
- Affordable dev boards
  - Sipeed MAiX BiT is only $13
- Full support added in Linux 5.8
  - RISC-V NOMMU and M-Mode Linux talk by Damien Le Moal at LPC 20219
- 5 boards supported in u-boot

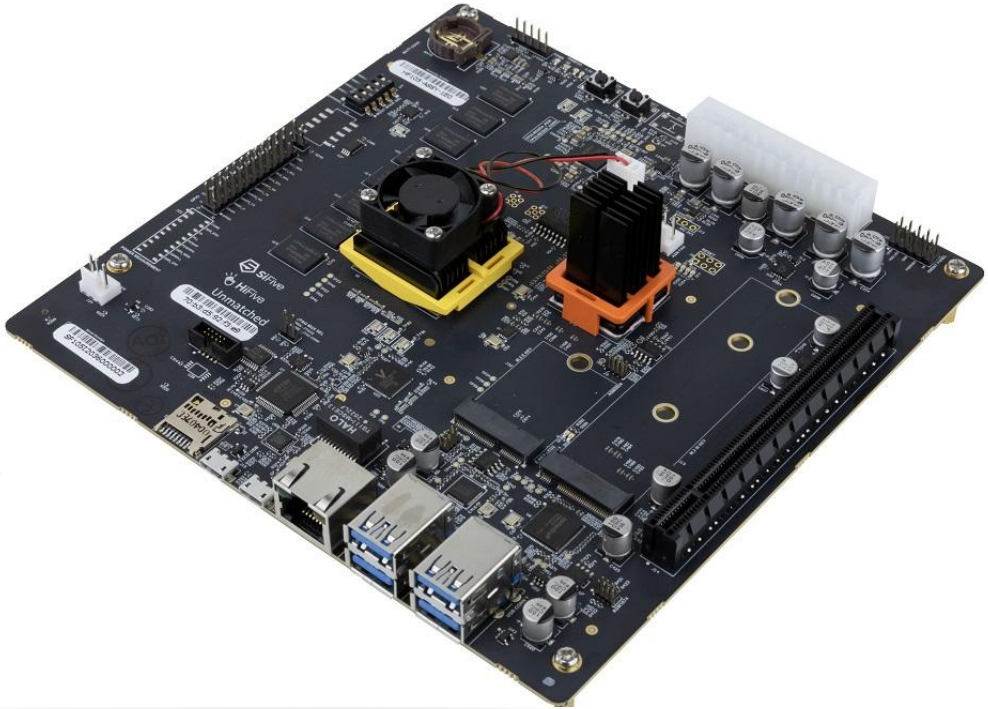
Kendryte K210

- Buildroot with busybox for rootfs
  - upstreaming in progress by Damien on buildroot list: Add RV64 NOMMU and Canaan K210 SoC support
- Damien Le Moal created a 6 DoF robotic arm
- 8MB of RAM runs out very quick!
  - No shared libraries as MMU implements draft spec not supported by Linux
  - Add risc-v support to elf2flt: ELF to bFLT (binary flat) converter for no-mmu Linux targets



SiFive Unmatched

- $665 on CrowdSupply
  - Shipping from Mouser too
- SiFive Freedom FU740 SoC
  - 4x U74 RV64GC application cores
  - 1x S7 RV64IMAC embedded core



SiFive Unmatched

- Mini-ITX PC form factor
  - 8GB DDR4 RAM
  - 4x USB 3.2 Gen 1 ports
  - Gigabit Ethernet
  - x16 PCIe Gen 3 Expansion Slot
  - M.2 connector for NVMe SSD
  - M.2 connector for WiFi/Bluetooth

Alibaba T-Head XuanTie 910

- T-Head is a subsidiary of Alibaba
- 16-core 2.5 GHz RISC-V processor
  - implementation of draft Vector extension
  - Performance comparable to Arm Cortex-A73
  - Paper: Xuantie-910: A Commercial Multi-Core 12-Stage Pipeline Out-of-Order 64-bit High Performance RISC-V Processor with Vector Extension

Android on T-Head C910 SoC

- T-Head has ported Android 10 (AOSP) to RISC-V architecture!

T-Head ICE Evaluation Board

- ICE EVB is a XuanTie C910 based high performance SoC board developed by T-Head

- Dual core T-Head XuanTie C910@1.2GHz
- Extra C910V@1.2GHz core with vector extension, up to 128bit
- DDR4 with speed up to 2400Mbps
- Support GMAC interface
- Support GPU and 3D
- Display: RGB888 LED, 1080P
- Chip size: 15x15mm
- Process: 28HPC+

T-Head XuanTie C906

- RV64GCV, 5-stage, in-order pipeline, up to 1GHz, single-core

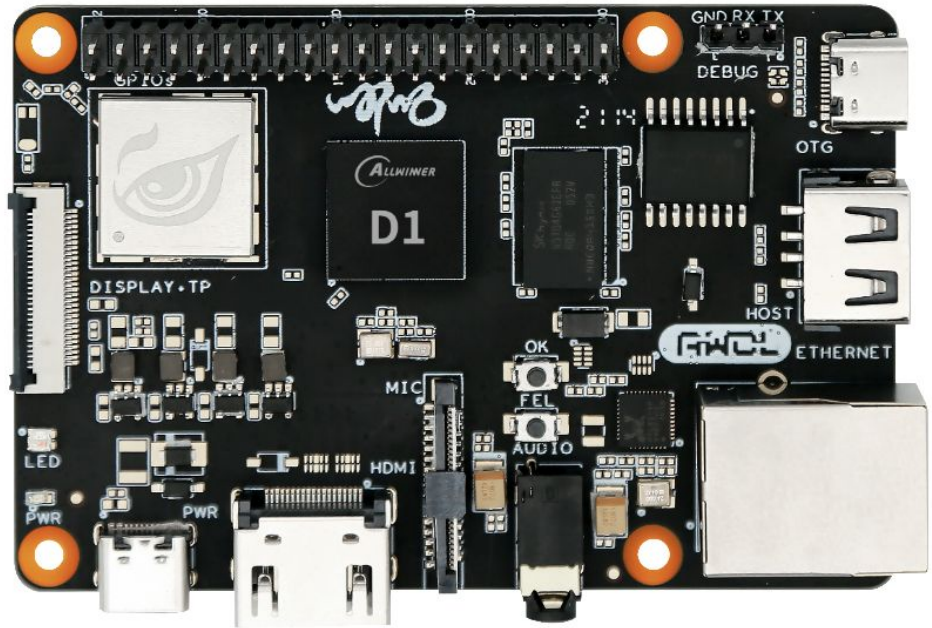
Feature	Description
Architecture	RV64GCV
Pipeline	5-stages
T-Head extension	T-Head instruction extension（TIE） T-Head memory attributes extension（TMAE）
Floating point Unit	Support RISCV Half、F、D instruction extension Support IEEE 754-2008 standard
Vector Unit	Support RISCV V instruction extension(configurable) vector register width 128bit element size support 8/16/32/64bit support INT8/INT16/INT32/INT64/BFP16/FP16/FP32/FP64
Bus interface	AXI4.0 128-bit
Instruction Cache	Up to 64KB（configurable）
Data Cache	Up to 64KB（configurable）
Interrupt controller	Flexibly configurable Platform-Level Interrupt Controller (PLIC)
Memory Management Unit (MMU)	Sv39 virtual memory translation
Physical Memory Protection (PMP)	Up to 16 regions
Debug	RV debug

Allwinner D1 SoC

- Allwinner D1 has single T-Head C906 (RV64GCV) core at 1 GHz

Allwinner Nezha D1 dev board

- Nezha dev board made by AWOL (Allwinner Online) with Allwinner D1
- Starter kit bundle for at $115 on AliExpress



- Main control: Allwinner D1 C906 RISC-V 1GHz
- DRAM: DDR3 1GB/2GB
- Storage: Onboard 256MB spi-nand, support USB external U disk and SD card to expand storage
- Network: Support Gigabit Ethernet, support 2.4G WiFi and Bluetooth, onboard antenna
- Display: Support MIPI-DSI+TP screen interface, support HDMI output, support SPI screen
- Audio: Microphone daughter board interface * 1, 3.5mm headphone jack * 1 (CTIA)
- Board size: length 85mm*width 56mm*thickness 1.7mm
- PCB layer: 6 layers
- Support Tina Linux，based on Linux 5.4 kernel

RISC-V International developer board program

- RISC-V Developer Boards initiative from RISC-V International to get Linux capable boards into developers' hands!
  - Launched with the Allwinner D1 Nezha board and SiFive Unmatched *(limited qty)*
- Fill out the RISC-V Developer Boards form
  - Preference is to be RISC-V International member or from a RISC-V International member organization. NOTE: individuals can join RVI free of cost!
  - Explain why you are interested in a RISC-V board and what you plan to do with it
  - For example, adding RISC-V support to an upstream open source software project
  - Don't overestimate the hardware specs you actually need like RAM

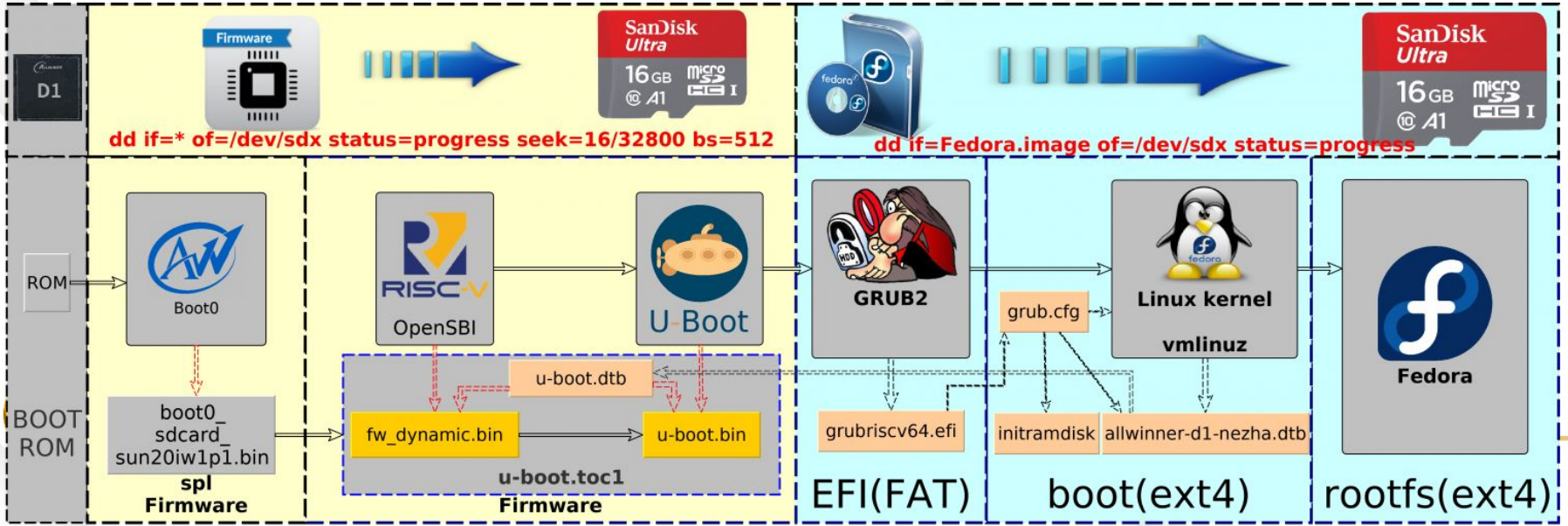
Allwinner D1 open source community

- linux-sunxi: strong open source community for Allwinner SoCs
  - D1 wiki page
  - Allwinner Nezha board wiki page
- Samuel Holland has been working on getting mainline to run
  - SPL: https://github.com/smaeul/sun20i_d1_spl
  - OpenSBI: https://github.com/smaeul/opensbi
  - U-Boot 2021.10-rc3: https://github.com/smaeul/u-boot/tree/d1-wip
  - Linux 5.14-rc4: https://github.com/smaeul/linux/tree/riscv/d1-wip

Fedora on Allwinner Nezha D1 board

- Fedora wiki: [Architectures/RISC-V/Allwinner]

  - [Wei Fu] *(RISC-V Ambassador and RedHat engineer)* has created rawhide XFCE image and produced great documentation of the boot flow and SD card partition layout



Challenges for Allwinner D1 and T-Head C906

- "What's the problem with D1 Linux upstream?" last week at Plumbers
  - Guo Ren (Alibaba T-Head), Liu Shaohua (Allwinner), Wei Fu (Red Hat/Fedora)
  - Slides in Google Slides and PDF, jump to 2h 28 min the live stream recording
  - Peripherals are mostly reused from existing ARM SoC so not much worked needed
  - T-Head C9xx core performance functionality that's not critical to boot upstream:
    - Instructions to accelerate I-cache synchronization
    - Instructions to accelletrate TLB synchronization
    - Vector 0.7.1 draft spec

How to handle non-coherent interconnects?

- T-Head designed C9xx cores in 2019 and there was no spec on how to handle DMA on a RISC-V system with non-coherent interconnects
- non-coherent interconnects can make it possible to have low cost SoC
- However, the RISC-V Privileged spec wrote: “In RISC-V platforms, the use of hardware-incoherent regions is discouraged due to software complexity, performance, and energy impacts”
- Guo Ren posted [PATCH] riscv: Support non-coherency memory model in 2019 but a RISC-V extension for this was still in an early phase

Page-Based Memory Types (PBMT) extension

- A proposal arose in the RISC-V Virtual Memory Task Group
  - "PBMT" extension proposal to support Page-Based Memory Types (aka "page-based attributes")
- SvPBMT extension is now frozen and in a 45 day public review period

PTE format: T-Head vs. PBMT

- Standard PBMT and D1 custom PTE use the highest bits to determine memory type. But the encoding and semantics are different.

```
Svpbmt PTE format:
| 63 | 62-61 | 60-8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0
  N     MT     RSW    D   A   G   U   X   W   R   V
        ^
RISC-V
Encoding &
MemType     RISC-V Description
----------  -----------------------------------------
00 - PMA    Normal Cacheable, No change to implied PMA memory type
01 - NC     Non-cacheable, idempotent, weakly-ordered Main Memory
10 - IO     Non-cacheable, non-idempotent, strongly-ordered I/O memory
11 - Rsvd   Reserved for future standard use
```

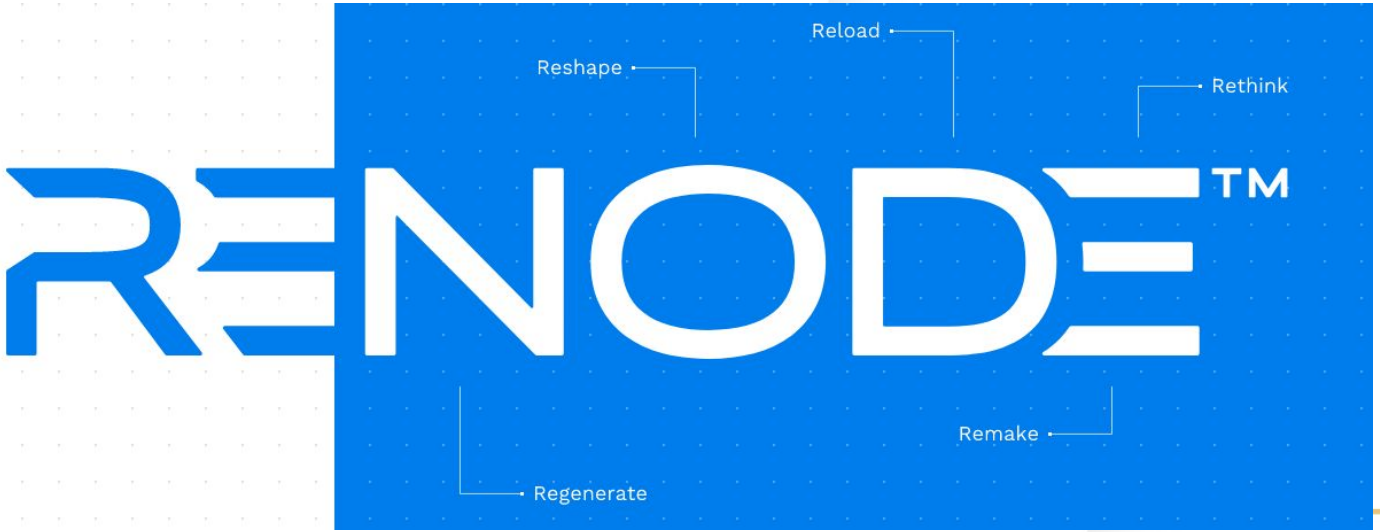
```
T-HEAD C9xx PTE format:
| 63 | 62 | 61 | 60 | 59-8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0
  SO   C    B    SH   RSW    D   A   G   U   X   W   R   V
  ^    ^    ^    ^
BIT(63): SO - Strong Order
BIT(62): C  - Cacheable
BIT(61): B  - Bufferable
BIT(60): SH - Shareable

MT_MASK : [63 - 59]
MT_PMA  : C + SH
MT_NC   : (none)
MT_IO   : SO
```

PTE format: T-Head vs. PBMT

- Guo Ren has posted implementation for standard PBMT:
  - [PATCH V2 1/2] riscv: Add RISC-V svpbmt extension
- Watch linux-riscv mailing list for further discussion

No hardware? Try Renode!

- Renode can simulate physical hardware systems including CPU, peripherals, sensors, and networking



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

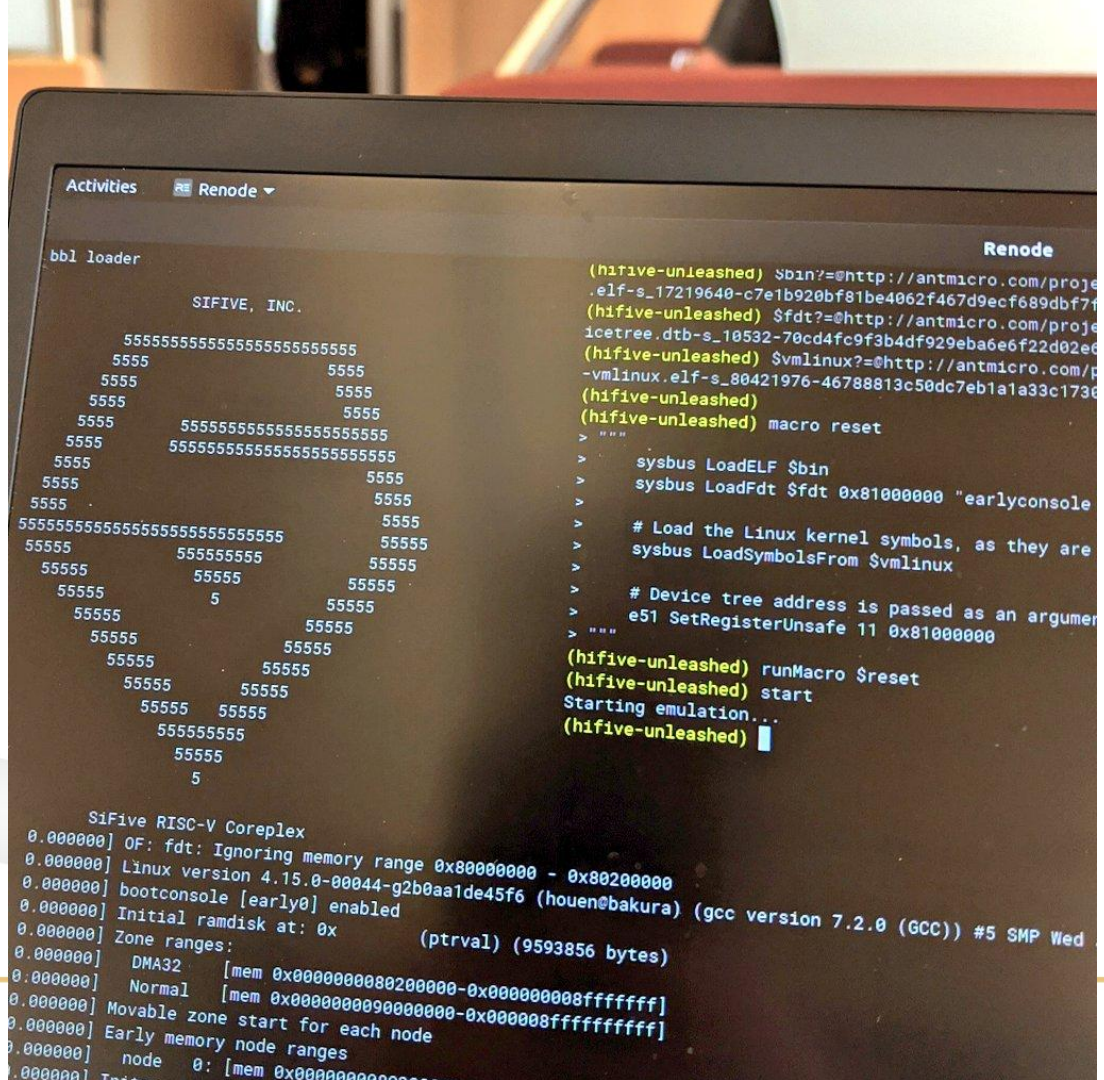
single-node/quark_c1000.resc

Fomu

LiteX/VexRiscv on Digilent Arty

Xilinx ZedBoard

source: [Renode supported boards]


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 0x0000000080200000-0x00000008fffffff]
0.000000] Normal [mem 0x0000000090000000-0x00000bffffffffff]
0.000000] Movable zone start for each node
0.000000] Early memory node ranges
0.000000] node 0: [mem 0x0000000080200000-0x00000008fffffff]
(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)

How to get involved with RISC-V International?

- Individuals and non-profits can join free of cost
- RISC-V Technical wiki landing page is the single best place to remember
  - Technical Organization charts
  - ISA Extensions On Deck for Freeze Milestone for 2021 ratification
  - RISC-V Technical Working Groups
  - RISC-V extension and feature support
  - RISC-V Software Ecosystem

How to get involved with RISC-V International?

- RISC-V Working Groups mailing list server
  - From riscv.org: The work done within RISC-V International is organized on our groups server at lists.riscv.org. This includes mailing lists, file storage, meetings and calendar invitations, and archives, among other things. Groups are organized into a hierarchy by functional area. New groups are proposed and approved through the TSC (technical groups) or the Board of Directors (non-technical groups)"
  - All RISC-V technical committees and work groups are non-confidential – list traffic, meeting minutes, and deliverables are public
  - Active participation on lists and meetings is limited to RISC-V members

How to get involved with RISC-V International?

- Many groups have bi-weekly or monthly meetings
- Technical Meetings Calendar (NEW!)
  - ICS File
  - Google Subscribe Link
  - Note: Tech Groups Calendar is OLD and not used anymore

RISC-V Technical Meetings

Today | September 2021 | Print | Week | Month | Agenda

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30 8am tech-config meeting	31 9am Fast Interrupts TG Meeting	Sep 1	2 7:05am RISC-V Security HC 10am RISC-V Cryptographic Extensions wee 10am RISC-V: Graphics SIG	3	4
5	6 7am psABI TG 8am Platform HSC Bi-Weekly Meeting	7 6am RISC-V Development Partners 7am RISC-V Labs - Colocation / CI / Testing 9am Security Technologies SIG	8 6am RISC V DataCenter SIG 8am Platform HSC Bi-Weekly Meeting	9 7am Bi-Weekly RISC-V Dev Boards Meeting 8am Arch Compat. Test SIG SemiMonthly M 9am RISC-V Advanced Interrupt Controller M 10am RISC-V Cryptographic Extensions wee	10	11
12	13 7am Toolchain & Runtime TS Bi-Weekly Mee 8am tech-config meeting 9am CMO TG Meeting	14 8am TEE TG Weekly 9am Debug Task Group call 9am Fast Interrupts TG Meeting	15 10am Virtual Memory TG Meeting	16 7am RISC-V SIG-HPC monthly meeting 7:05am RISC-V Security HC meeting 8am RiscV Embedded SIG Monthly Meeting 10am RISC-V Cryptographic Extensions wee 10am RISC-V: Graphics SIG	17	18
19	20 8am Platform HSC Bi-Weekly Meeting 9am J Extension Meeting	21 6am RISC-V Development Partners 7am RISC-V Labs - Colocation / CI / Testing 8am TEE TG Weekly 9am Security Technologies SIG	22	23 7am Bi-Weekly RISC-V Dev Boards Meeting 8am Arch. Compat. Test SIG SemiMonthly M 10am RISC-V Cryptographic Extensions wee	24	25
26	27 7am Toolchain & Runtime TS Bi-Weekly Mee 8am tech-config meeting	28 8am Processor Trace Call 8am TEE TG Weekly 9am Fast Interrupts TG Meeting	29 10am Virtual Memory TG Meeting	30 7:05am RISC-V Security HC 9:05am Software Horizontal Committee 10am RISC-V Cryptographic Extensions wee 10am RISC-V: Graphics SIG	Oct 1	2

Events shown in time zone: Pacific Time - Los Angeles

GoogleCalendar

Cache Management Operations (CMO) TG

- Important for SoC's that lack cache coherent interconnects
- Zicmobase extension adds a base set of instructions and CSRs to handle Cache Block operations like invalidate, clean and flush
- Now frozen and under 45 day public review
- What about existing SoCs?
    - It is possible trap and emulate these new instructions in SBI once frozen
- Mailing list: tech-cmo, next task group meeting: Oct 11

Advanced Interrupt Architecture SIG

- RISC-V Advanced Interrupt Architecture (AIA)
  - APLIC: Advanced Platform-Level Interrupt Controller
  - IMSIC: Incoming Message-Signaled Interrupt Controller
  - Mailing list: tech-aia
- ACLINT (Advanced Core Local Interruptor)
  - Backwards compatible with the SiFive CLINT but restructured as 3 devices: MTIMER (M-mode timer), MSWI (M-mode software interrupts), SSWI (S-mode software interrupts)
  - Discussion of ACLINT occurs on the RISC-V Unix Platform Mailing list

Advanced Interrupt Architecture SIG

- Next Generation RISC-V Interrupt Support talk by Anup Patel
  - Last week at Plumbers: live stream recording

AIA & ACLINT for OS-A platforms

Possible uses of AIA and ACLINT in OS-A platforms

RISC-V AIA Specification | RISC-V ACLINT Specification | RISC-V SBI Specification | RISC-V Privilege Specification

OS-A Platforms	MSIs			Wired Interrupts			IPIs			Timer		
	M-level	S-level	VS-level	M-level	S-level	VS-level	M-level	S-level	VS-level	M-level	S-level	VS-level
Legacy Wired IRQs	NA	NA	NA	PLIC	PLIC	PLIC (Emulate)	MSWI (CLINT) Phase1	SBI IPI	SBI IPI	MTIMER (CLINT) Phase1	SBI Timer	SBI Timer
Only Wired IRQs	NA	NA	NA	APLIC M-level Phase1	APLIC S-level Phase1	APLIC S-level (Emulate) Phase2	MSWI Phase1	SSWI Phase1	SBI IPI	MTIMER Phase1	Priv Sstc In-progress	Priv Sstc In-progress
MSIs and Wired IRQs	IMSIC M-file Phase1	IMSIC S-file Phase1	IMSIC S-file (Emulate) Phase2	APLIC M-level Phase1	APLIC S-level Phase1	APLIC S-level (Emulate) Phase2	IMSIC M-file Phase1	IMSIC S-file Phase1	SBI IPI	MTIMER Phase1	Priv Sstc In-progress	Priv Sstc In-progress
MSIs, Virtual MSIs and Wired IRQs	IMSIC M-file Phase1	IMSIC S-file Phase1	IMSIC VS-file Phase2	APLIC M-level Phase1	APLIC S-level Phase1	APLIC S-level (Emulate) Phase2	IMSIC M-file Phase1	IMSIC S-file Phase1	IMSIC VS-file Phase2	MTIMER Phase1	Priv Sstc In-progress	Priv Sstc In-progress

RISC-V Platform Specification

- Platform horizontal steering committee (HSC)
  - standardize the interface between hardware platforms and OS like Linux
  - agenda and minutes on wiki for bi-weekly meetings chaired by Kumar Sankaran
  - Recordings for past meetings available in Google Drive
  - Slides from past meetings are online
  - Mailing list: tech-unixplatformspec

RISC-V Platform Specification

- OS-A Platform: to run full OS like Linux, BSD, Windows
  - RVA22U profile for user-mode.
  - RVA22S profile for supervisor-mode.
  - RVM20M64 profile for machine-mode.
  - Must comply with Embedded Base Boot Requirements (EBBR) Specification
  - Optional server extension mandates additional requirements like ACPI
- M Platform: to run bare-metal applications or RTOS on microcontroller

RISC-V Platform Specification

- Session in RISC-V microconference at Linux Plumbers Conference

RISC-V Platform Specification

- ACPI for RISC-V at Linux Plumbers Conference

Proof of Concept

QEMU

- **ACPI Tables –** RSDP, XSDT, FADT, DSDT, MADT, RTDT, MCFG
- **MADT:**
  - Per-hart INTC
  - IMSIC
  - Per-socket APLIC
- **DSDT:**
  - Processors
  - APLIC with _MAT
  - Generic 16550a UART(PNP0501) with _DSD method
  - Virtio

EDK2

- **Integrated OpenSBI with AIA support**
- **ACPI enablement** (AcpiTableDxe, QemuFwCfg)
- **SMBIOS enablement**

Linux

- **Basic ACPI enablement for RISC-V** (ACPICA and ARCH specific ACPI)
- **ACPI based timer driver** (RTDT)
- **ACPI based INTC Driver**
- **ACPI based IMSIC driver**
- **ACPI based APLIC driver**
- **SMBIOS enablement**
- **Hart capabilities using SMBIOS table 44**

RISC-V at Linux Plumbers Conference

- Live stream on YouTube, Detailed notes with links
  - Sessions
    - The RISC-V platform specification
    - ACPI for RISC-V
    - What's the problem with D1 Linux upstream?
    - Puzzle for RISC-V ifunc
    - Towards continuous improvement of code-generation for RISC-V