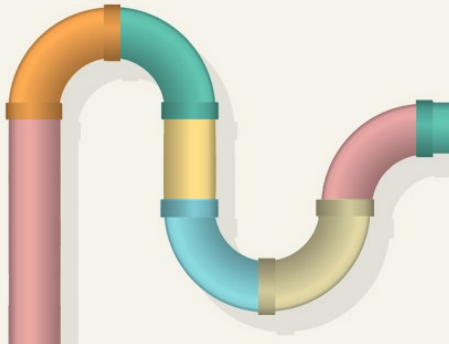
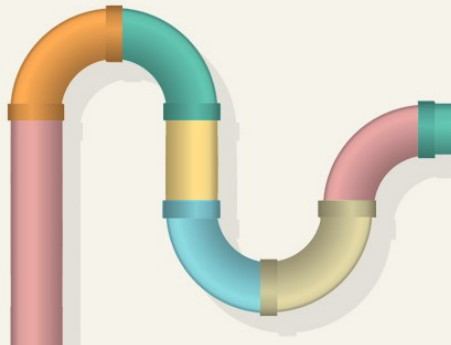
Why RISC-V Is Not Nearly Boring Enough

Al Stone
Principal Software Engineer
Platform Enablement
Red Hat



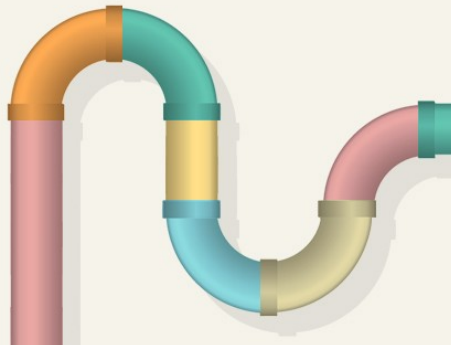
When RISC-V Grows Up...

- The ISA is only a small part of a product
- What we need is to be dead boring
- To get there, we need:
  - a clear vision
  - a clear process
  - a clear – and complete – specification



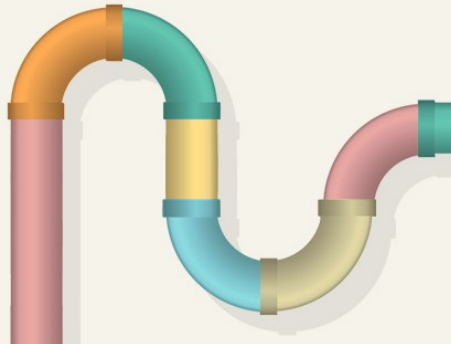
Discussion Topics

- So, what about the Vision Thing?
- Getting things done
- Filling in all the blanks
  - what do we have?
  - an outline for what we need
- Even more discussion



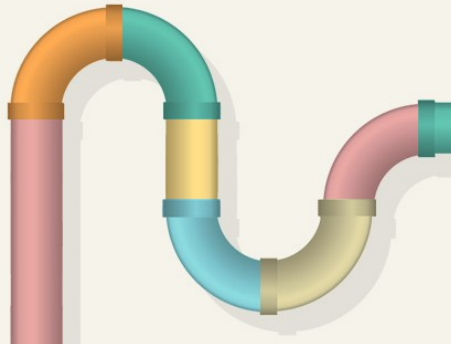
The Vision Thing

- Unix-class platform specification …..
  - First thought: ***too*** boring
    - What about various BSDs, RTOSs, and yes, even Windows?
    - Suggestion: make it an OS Platform Spec
  - Second thought: what's the goal?
    - Set expectations for OSs: processors, devices and firmware
    - Set expectations for platform providers: what they need to provide



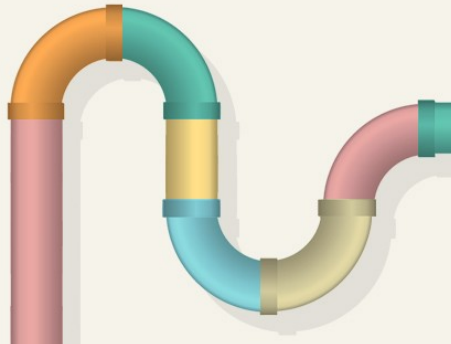
The Vision Thing

- Operating System Platform Spec (OSPS)
  - Clearly define terminology
  - Clearly identify RISC-V ISA in use, and what to do when something is missing
  - Clearly define I/O: required buses, required devices, required behavior
  - Detailed specification of interface between OS and firmware
    - and between OS and hardware via firmware
    - and so that virtualization is possible
  - Keep it Simple
  - Keep it Small



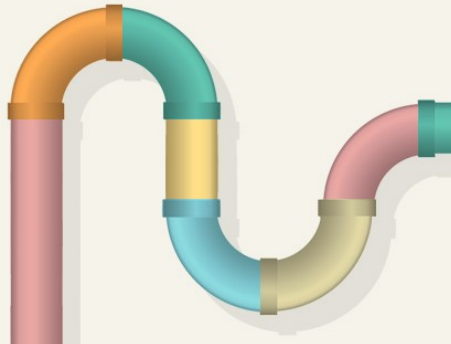
The Vision Thing

- Compliance will be an issue
  - Humans are involved (inadvertent errors)
  - Humans are involved (intentional errors)
  - Tools to help:
    - Reference QEMU implementation
    - A Test Suite
    - Official certification (“Meets OSPS x.y”)



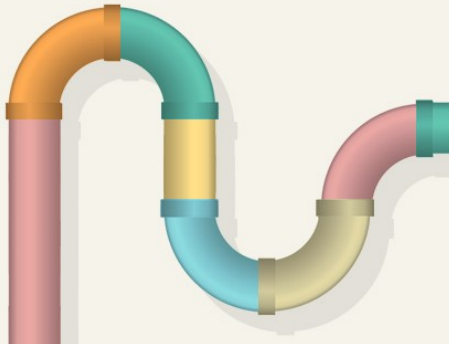
Getting Things Done

- Current: UNIX-Platform Spec TG; drop the “UNIX”?
- Github: https://github.com/riscv/riscv-platform-specs
- Member's portal: https://lists.riscv.org/g/tech-unixplatformspec
- Current change process: Discuss *ad infinitum* on ML?
- Keep it Simple:
  - RFC on the ML
  - On reasonable consensus, submit MR
  - Commits must have SoB
  - Each MR discussed/voted on in TG
  - Pass to TSC?
- Versioning: x.y? Once a quarter with RCs?



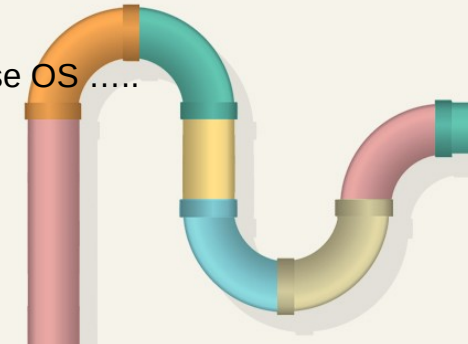
What We Have

- Github: https://github.com/riscv/riscv-platform-specs
- Can you build an SBC, or a laptop, or a server to be used with any general purpose OS based on this list?
- Can you modify an operating system, either Linux or That Other One, that will reliably boot on a platform meeting these requirements?



What We Need

- Fair Warning:
  - ML discussion typically very detailed
  - This author thinks from the general to the specific
  - And he has much to do
- Overall Structure
  - HBI, SBI, ABI ….
  - Hardware: ISA, CPU, memory, I/O devices and buses
  - Boot Sequence: hardware → firmware → boot loader → kernel (the protocols)
  - Kernel: device enumeration and management
  - Profiles/Use Cases: dev boards, embedded, RTOS, general purpose OS …..
- Compliance Levels?
  - Accept what has been done as L0?
  - Jump straight to what we want?

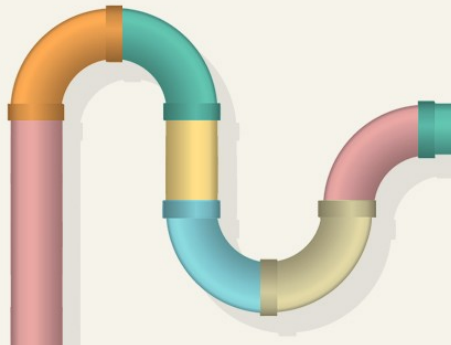


What We Need

(with apologies to Jack Kerouac)

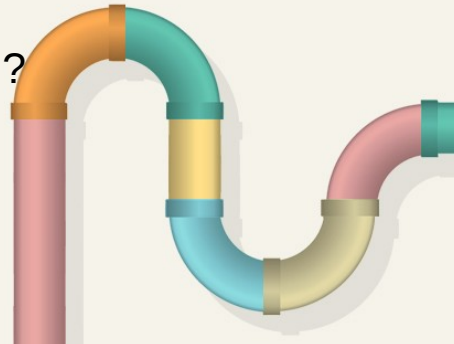
- Hardware
  - CPU
    - Required ISA Components
    - Privilege Levels and their Usage
    - Identification: make, model, modules, topology
    - Performance Monitoring
    - Debug Instructions, Trace Instructions
    - Timers
    - Virtualization
  - Memory
    - MMU
    - Addressability (tags?)
    - Page Sizes
    - EDAC
  - I/O
    - IPL
    - Interrupt Controllers
    - MMIO
    - IOMMU and virt-iommu
    - Buses
    - Serial Console
    - Base Management Controller
    - TPM
    - Debug port (JTAG)

- H- or M-mode
  - Trusted execution environment
  - CPU services (e.g., provided to UEFI)
    - power on/off
    - frequency managemenr
    - power management
    - thermal management
    - Does identification go here or the ISA?
  - Booting the platform
    - IPL
    - Network boot
    - More console details?
  - Kernel (S-mode)
    - device management
    - processor management
    - enumeration
    - firmware update
  - User space (U-mode)
    - Identification (e.g., DMI)
    - Firmware update



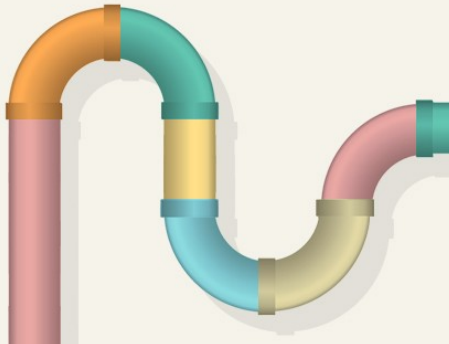
What We Need

- Profiles/Use Cases
  - Over time (L0, L1, ….)
  - By target (dev board, embedded, general OS ….)
  - Compliance should be by target, then by level
  - How do we determine compliance?
    - More importantly, who?
- One last random thought …
  - What about form factors such as mini-iTX and such?



What did we just do?

- The Vision Thing
- Getting things done
- What could/should we do?
  - what do we have?
  - what do we need?
- What happens next ….



Red Hat

Thank You!

Platform spec: https://github.com/riscv/riscv-platform-specs

Mailing list: tech-unixplatformspec@lists.riscv.org

IRC: Freenode #fedora-riscv, #riscv