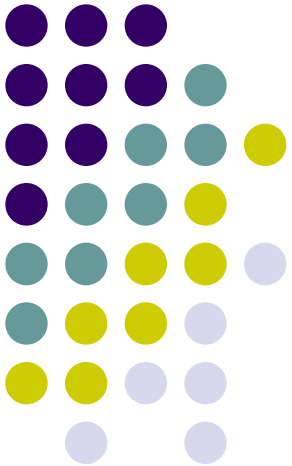
RISC-V ISA Architectures and Eco Systems

劉靖家
清華大學電機系

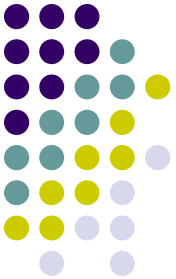


Design Technology Center

References

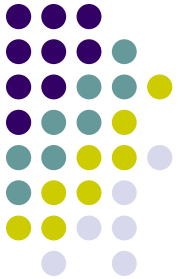
- Why RISC-V? “Instruction Sets Want to be Free”
  - Krste Asanovic, Professor UC Berkeley, Chairman RISC-V Foundation, Co-Founder SiFive
  - https://riscv.org/2017/05/6th-risc-v-workshop-proceedings/
- RISC-V Instruction Set Manual
  - User-Level ISA
  - Privileged Architecture
  - https://riscv.org/specifications/
- RISC-V Summit and Workshop Proceedings https://riscv.org/category/workshops/proceedings/ /

What is RISC-V?



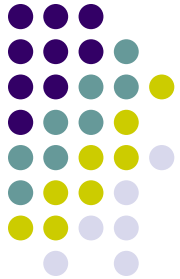
- Fifth generation of RISC design from UC Berkeley
- A high-quality, license-free, royalty-free RISC ISA Specification
- Standard maintained by non-profit RISC-V Foundation
- Appropriate for all levels of computing system, from microcontrollers to supercomputers
- 498 (138 companies & 35 universities) for 7th workshop (Nov, 2017) at WD, CA
- >1100 attendees for 2018 RISC-V summit!

RISC-V Foundation



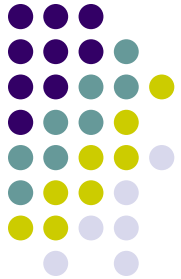
- Mission statement
  - “to standardize, protect, and promote the free and open RISC-V instruction set architecture and its hardware and software ecosystem for use in all computing devices.”
- Established as a 501(c)(6) non-profit corpora(on on August 3, 2015
- First year, 41 “founding” members.
- >325 members
  - 140 members at Q3 2018
  - 60 at Q1 2017

What's Different about RISC-V?



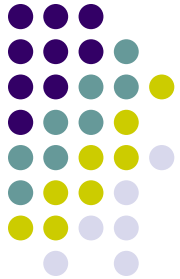
- Simple
  - Far smaller than other commercial ISAs
- Clean-slate design
  - Clear separation between user and privileged ISA
  - Avoids μ-architecture or technology-dependent features
- A modular ISA
  - Small standard base ISA
  - Multiple standard extensions
- Designed for extensibility/specialization
  - Variable-length instruction encoding
  - Vast opcode space available for instruction-set extensions
- Stable
  - Base and standard extensions are frozen
  - Additions via optional extensions, not new versions

RISC-V Base Plus Standard Extensions



- Four base integer ISAs
  - RV32E, RV32I, RV64I, RV128I
  - RV32E is 16-register subset of RV32I
  - Only <50 hardware instructions needed for base
- Standard extensions
  - M: Integer multiply/divide
  - A: Atomic memory operations (AMOs + LR/SC)
  - F: Single-precision floating-point
  - D: Double-precision floating-point
  - G = IMAFD, “General-purpose” ISA
  - Q: Quad-precision floating-point

Other RISC-V Extensions

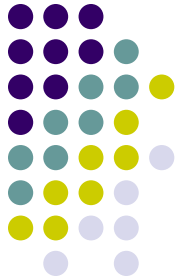


- “A”: Atomic Operations Extension
  - Fetch-and-op
  - Load–Reserved/Store Conditional
- “C”: Compressed Instruction Extension
- “V” Vector Extension State (almost done)
- L: decimal floating points
  - GCC decimal floating types: _Decimal32, _Decimal64, and _Decimal128
- B: Bit Manipulation
  - Insert, extract, and test bit fields, rotations, funnel shifts, and bit and byte permutations, etc.
- J: Dynamically Translated Languages
  - Dynamic checks and garbage collection.
- T: Transactional Memory
- P: Packed-SIMD

Variable-Length Encoding

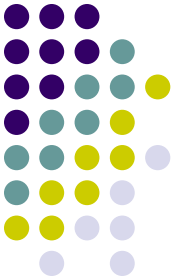
- Extensions can use any multiple of 16 bits as instruction length
- Branches/Jumps target 16-bit boundaries even in fixed 32-bit base

RISC-V Privileged Architecture



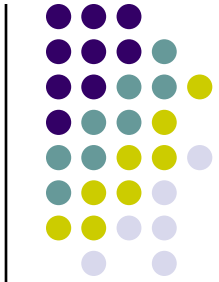
- Three privilege modes
  - User (U-mode)
  - Supervisor (S-mode)
  - Machine (M-mode)
- Supported combinations of modes:
  - M --- simple embedded systems
  - M, U --- embedded systems with protection
  - M, S, U --- systems running Unix-style OS

Virtual Memory Architectures (M, S, U modes)

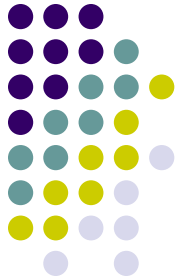


- Designed to support current Unix-style OS
- Sv32 (RV32)
  - Demand-paged 32-bit virtual-address spaces
    - 10+10+12 bits
  - 2-level page table
  - 4 KiB pages or 4 MiB megapages
- Sv39 (RV64)
  - Demand-paged 39-bit virtual-address spaces
    - 9+9+9+12 bits
  - 3-level page table
  - 4 KiB pages, 2 MiB megapages, 1 GiB gigapages
- Sv48, Sv57, Sv64 (RV64)

On-going Efforts

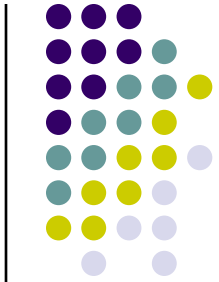


- Formal spec
- Hypervisor
- Crypto
- J (dynamic translation / runtimes)
- Packed SIMD
- Vector
- Security
- Fast interrupts
- Trace



RISC-V SOFTWARE TOOLS

Simulators



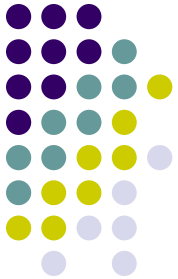
- Spike (ISS)
- RISCVEMU (ISS)
- RV8 (RISC-V simulator for x86-64)
- Qemu (ISS with dynamic translation)
  - upstreamed
- Imperas OVP (ISS with dynamic translation)

- C++ model generated by Verilator
  - Cycle accurate

- Gem5 model

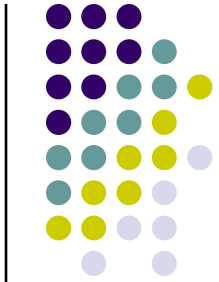
https://github.com/riscv/riscv-software-list

Toolchains

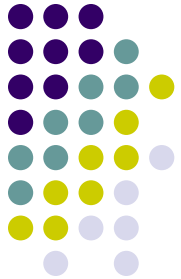


- GNU-based toolchains
  - binutils, gcc, glibc, newlib all upstreamed.
- LLVM port is making rapid progress.
  - RV32IMFDC upstream.
- Bootloader
  - U-boot is upstream
- Debugger
  - gdb upstream (SiFive)
  - OpenOCD (SiFive)
  - Commercial: Segger, Lauterbach, UltraSoC, IAR

Embedded and Linux

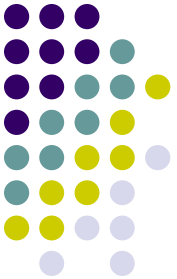


- Embedded runtimes
  - Zephry (upstream), seL4 (upstream), FreeRTOS exists (not upstream), Micrium uC/OS and ThreadX.
- Linux kernel port was upstreamed in January 2018.
  - Only supports RV64I now
- Fedora and Debian support is in progress



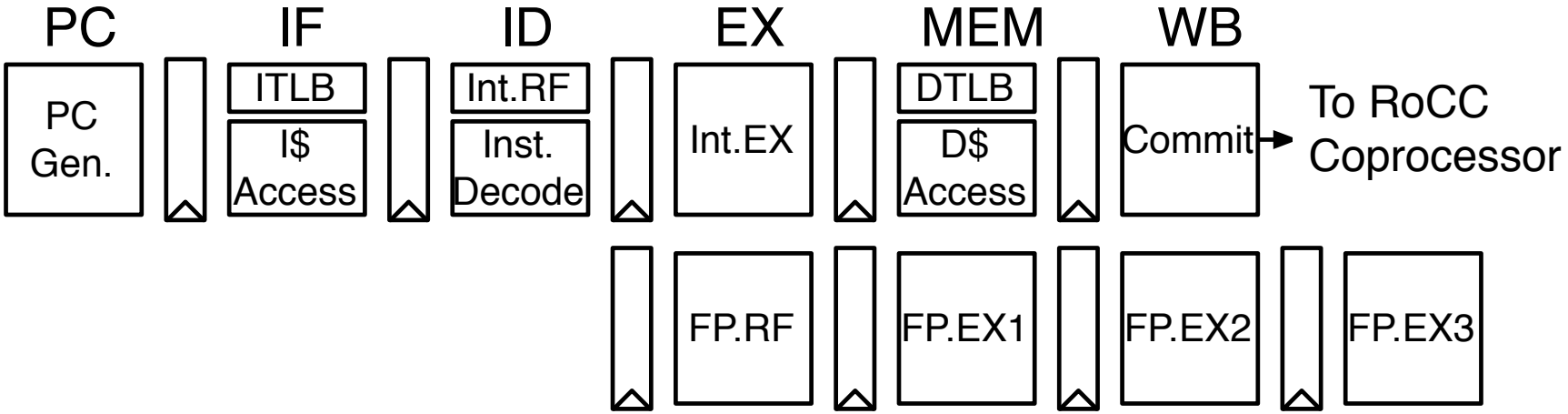
RISC-V CORES AND CHIPS

UC Berkeley RISC-V Core Generators



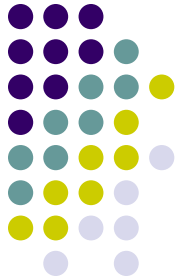
- **Rocket:** Family of In-order Cores
  - Supports 32-bit and 64-bit single-issue only
  - Dual-issue soon
  - Similar in spirit to ARM Cortex M-series and A5/A7/A53
- **BOOM:** Family of Out-of-Order Cores
  - Supports 64-bit single-, dual-, quad-issue
  - Similar in spirit to ARM Cortex A9/A15/A57
- All based on Chisel language

RISC-V Rocket In-Order Core



- 64-bit 5-stage single-issue in-order pipeline
- Design minimizes impact of long clock-to-output delays of compiler-generated RAMs
- MMU supports page-based virtual memory
- 64-entry BTB, 256-entry BHT, 2-entry RAS
- IEEE 754-2008-compliant FPU
  - Supports SP, DP fused multiply-adds with hardware support for subnormals
- Currently working on dual-issue in-order Rocket

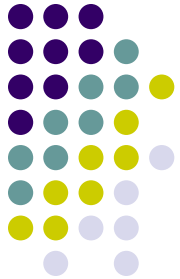
ARM Cortex-A5 vs. RISC-V Rocket



Category	ARM Cortex-A5	RISC-V Rocket
ISA	32-bit ARM v7	64-bit RISC-V v2
Architecture	Single-Issue In-Order	Single-Issue In-Order 5-stage
Performance	1.57 DMIPS/MHz	1.72 DMIPS/MHz
Process	TSMC 40GPLUS	TSMC 40GPLUS
Area w/o Caches	0.27 mm^2	0.14 mm^2
Area with 16K Caches	0.53 mm^2	0.39 mm^2
Area Efficiency	2.96 DMIPS/MHz/mm^2	4.41 DMIPS/MHz/mm^2
Frequency	>1GHz	>1GHz
Dynamic Power	<0.08 mW/MHz	0.034 mW/MHz

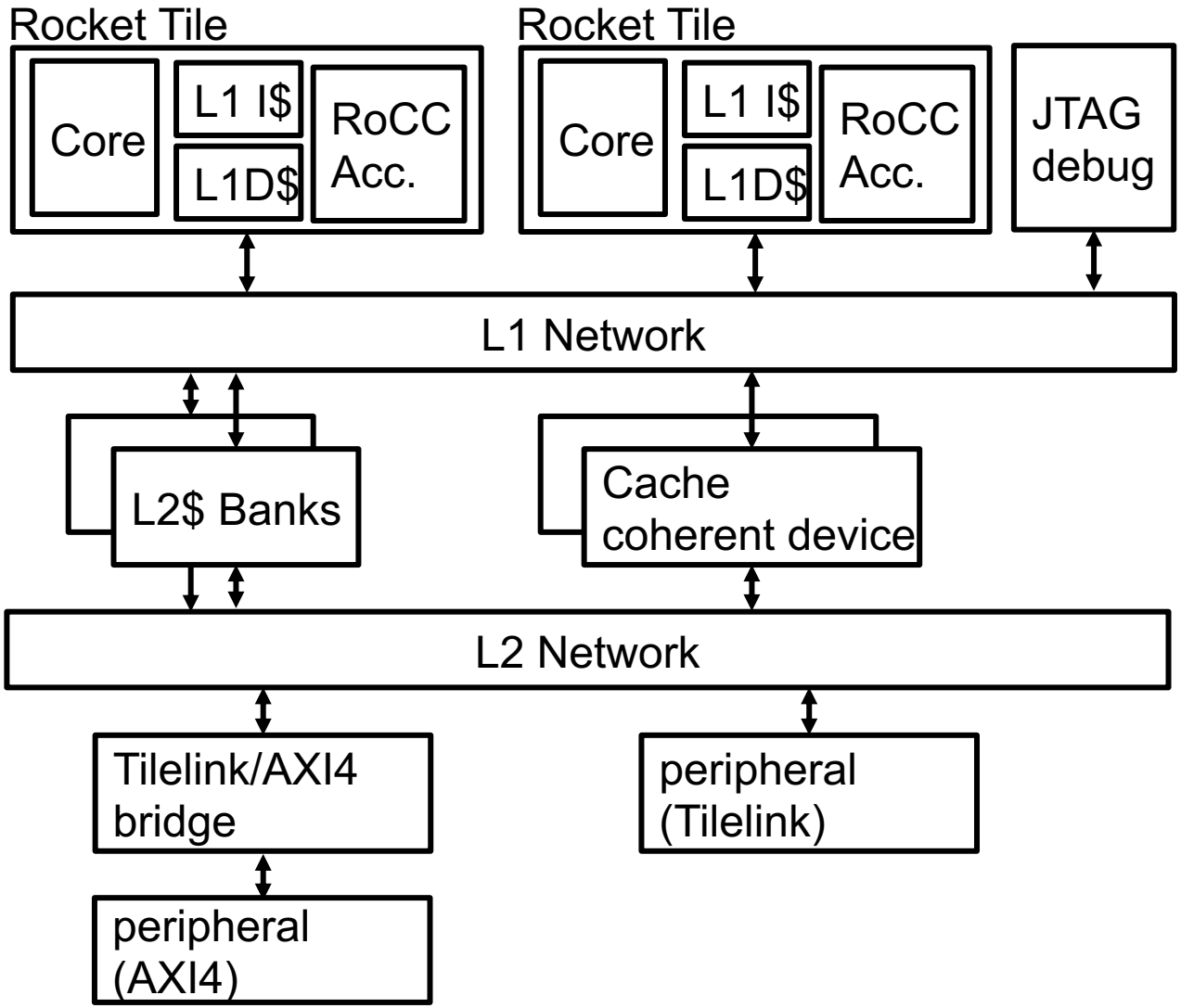
- PPA reporting conditions
  - 85% utilization, use Dhrystone for benchmark, frequency/power at TT 0.9V 25C, all regular VT transistors
- 10% higher in DMIPS/MHz, 49% more area-efficient

ARM Cortex-A9 vs. RISC-V BOOM

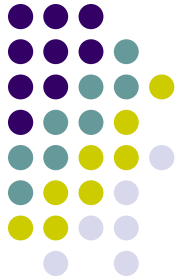


Category	ARM Cortex-A9	RISC-V BOOM-2w
ISA	32-bit ARM v7	64-bit RISC-V v2 (RV64G)
Architecture	2 wide, 3+1 issue Out-of-Order 8-stage	2 wide, 3 issue Out-of-Order 6-stage
Performance	3.59 CoreMarks/MHz	3.91 CoreMarks/MHz
Process	TSMC 40GPLUS	TSMC 40GPLUS
Area with 32K caches	2.5 mm^2	1.00 mm^2
Area efficiency	1.4 CoreMarks/MHz/mm^2	3.9 CoreMarks/MHz/mm^2
Frequency	1.4 GHz	1.5 GHz

Rocket Chip Generator



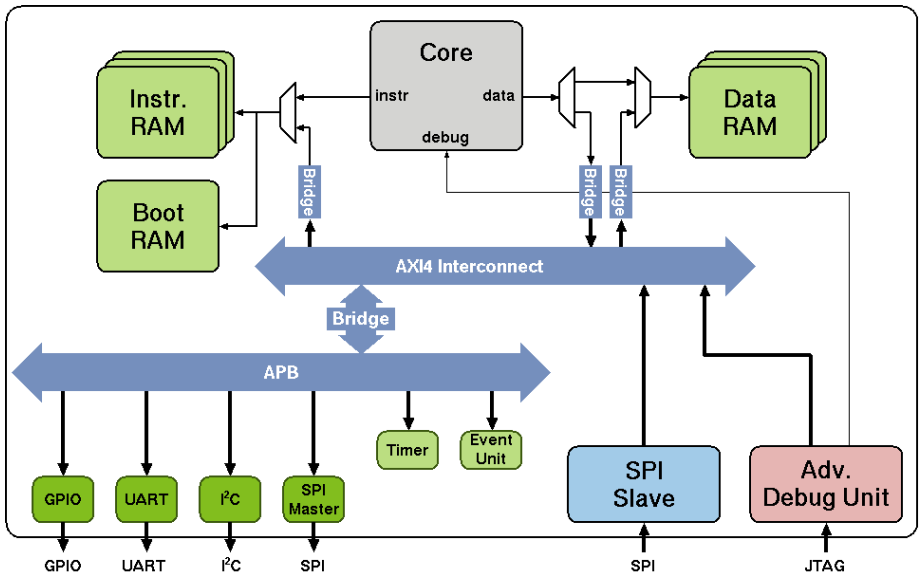
Rocket Chip Configuration Parameters



- Tune the design under different performance, power, area constraints, and diverse technology nodes
  - No. of Rocket tiles
  - No. of banks
  - No. of MSHRS
  - No. of sets in L1D & No. of ways in L1D
  - No. of sets in L1I & No. of ways in L1I
  - Coherence protocol: MI, MEI, MSI, MESI
  - Size of data TLB
  - Size of instruction TLB
  - Size of BTB
  - No. of trackers in coherence manager
  - Instantiate FPU?
  - No. of floating-point pipeline stages
  - Width of off-chip I/O
  - …

PULPINO Core

- ETH Zurich and University of Bologna
- PULP: Parallel Ultra-Low-Power Processor
- PULPino: A single-core RISC-V SoC
  - RV32I, partial RV32M, Compressed
  - Custom instructions
    - Hardware loops
    - Post-incr. ld and st
    - Multiply-Accumulate
    - ALU extensions (min, max, abs, ...)
- Use SystemVerilog
  - Built from grounds up
- Use in NXP small cores



https://github.com/pulp-platform/pulpino

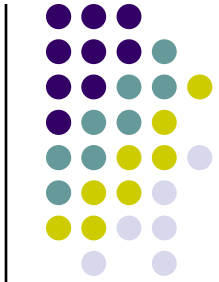
ETH Zurich Ariane: An open-source 64-bit RISC-V

- RV64-IC(MA)
- Full privileged specification
  - Linux
- 6-stage pipeline
  - In order issue, out-of-order write-back, in-order commit
  - Branch prediction
  - Scoreboarding
- Boot into Linux user on FPGA

SiFive

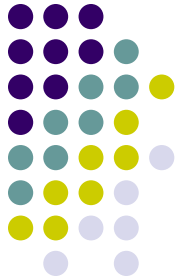
- Core members from UC Berkeley.
- https://www.sifive.com/risc-v-core-ip
- 3 series
  - E Cores
    - 32-bit embedded cores
    - MCU, edge computing, AI, IoT
  - S Cores
    - 64-bit embedded cores
    - Storage, AR/VR, machine learning
  - U Cores
    - 64-bit application processors
    - Linux, datacenter, network baseband

Andes Cores --- AndeStar V5 Architecture



- Baseline extension instructions
  - Memory accesses and branches with fewer instructions
  - Code size reduction on top of C-extension
- DSP/SIMD based on GPR
  - P-extension proposal
- Simplified custom instructions
- Non-instruction extensions: CSR-based
  - Vectored PLIC with priority preemption (Fast interrupts proposal)
  - Stack protection mechanism
  - Power management
  - Cache management in finer granularity
  - Simultaneous support for write-back and write-thru

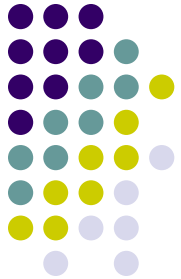
nVidia Use Case: Replacing in-house Core



Item	Requirement	ARM A53	ARM A9	ARM R5	RISC-V Rocket	NV RISC-V
Core perf	>2x falcon	Yes	Yes	Yes	Yes	Yes
Area (16ff)	<0.1mm^2	No	No	Yes	Yes	Yes
Security	Yes	TZ	TZ	No	Yes	Yes
TCM	Yes	Yes	No	Yes	No	Yes
L1 I/D $	Yes	Yes	Yes	Yes	Yes	Yes
Addressing	64bit	Yes	No	No	Yes	Yes
Extensible ISA	Yes	No	No	No	Yes	Yes
Safety (ECC/Parity)	Yes	Yes	Yes	Yes	Yes	Yes
Functional Simulation model	Yes	Yes	No	No	No	Yes

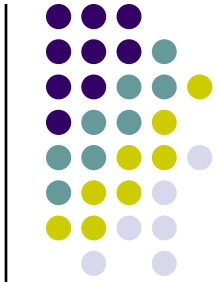
Flexibility to address both lower cost and higher performance.

WD RISC-V Core



- 2-way, superscalar, mostly in- order core with 9 stages pipeline:
  - Support for RV32IMC
  - 1 Load/Store pipe
  - 1 MLY
  - 1 DIV
  - 4 ALU engines
- First RISC-V based SoC for NAND controller applications
  - Full advantage of open source software ecosystem for RISC-V
  - Instruction optimization for NAND media handling
  - Freedom of power and performance optimization for end application

Summary



- Modern ISA design
  - Lean and modular
- Excellent (educational) source on processor design
  - Silicon-verified cycle-accurate
    - Rocket chip and Pulpino
  - Andes Cores
  - Flexible and easy to extend
- Fast progress on toolchain, simulators and software stack!