Introduction to CHIPS Alliance

(Common Hardware for Interfaces, Processors and Systems)

Zvonimir Z. Bandic, Chairman, CHIPS Alliance
Sr. Director, Western Digital Corporation

CHIPS ALLIANCE

Agenda

- Who are we?
- Project goals and deliverables
- Organization structure
- Governance model
- Membership, workgroups and events
- Conclusions
- (CHIPS Alliance example projects)

CHIPS Alliance – who are we?

- Open source hardware and open source design and verification tools
  - Fully open design methodology: from high level synthesis, to P&R, synthesis, physical design, PDks
- Founding members: Google, Western Digital, Esperanto, SiFive

- Extraordinary individuals: Wilson Snyder, Olof Kindgren

What is CHIPS Alliance?

- Organization which develops and hosts:
  - high quality, open source hardware code (IP cores)
  - Interconnect IP (phy and logical protocols)
  - open source software development tools - design, verification,…
- A barrier free environment for collaboration:
  - Standards organization framework for collaboration and development
  - Roadmap definition for IP and tools
  - Legal framework - Apache v2 license
- Shared resources ($ and time) which lower the cost of hardware development:
  - For IP and tools

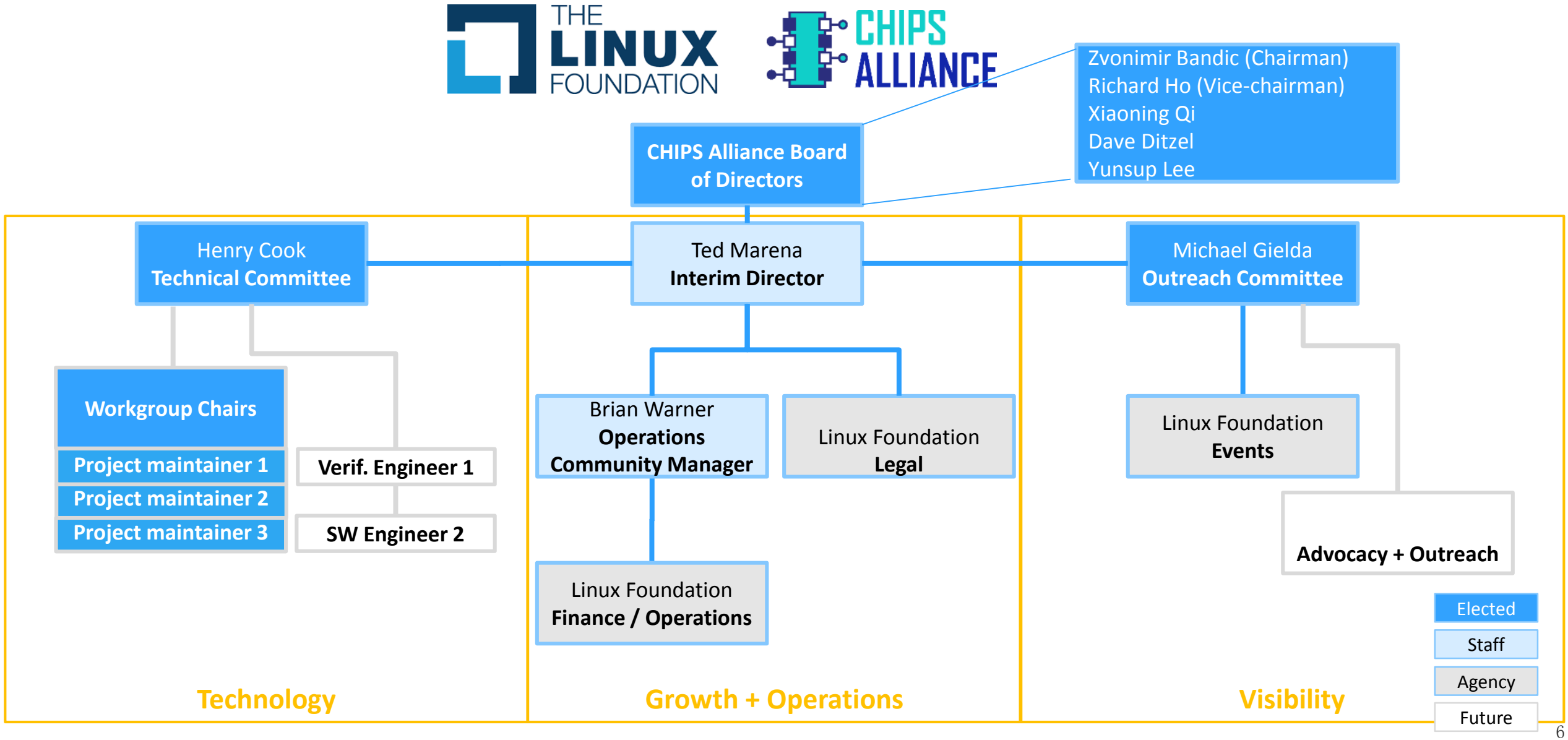
CHIPS ALLIANCE

Project Goals and deliverables

- Leverage common hardware development efforts:
  - IP blocks can be broadly used – RISC-V cores, Neural network accelerator cores, Uncore components (PCIe, DDR…), Interconnects
  - Verification contributions benefit all - joint resources on design verification
- Deliver high quality, open source CPU designs, peripherals and complex IP blocks
  - Known validated blocks that can be quickly adopted in silicon and/or FPGAs
- Develop and improve software development tools:
  - Open source RTL simulators – such as Verilator
  - Deploy cloud based design verification
  - Enable radically new design verification models, such as Python based design verification
- Explore and develop RedHat models for open source hardware

CHIPS ALLIANCE

CHIPS Alliance – organizational structure



Governance Model

Governing Board
oversees business decisions, budgets, outreach, marketing/events, trademarks, etc.

Technical Steeri ommittee
proposes projects t approved, top level coordin across projects

Project lifecycle document

Outreach Committee
coordinate evangelism, communication, outreach, events, training

THE APACHE SOFTWARE FOUNDATION
Apache v2

GitHub

Project Maintainers and Technical Team Workgroups
deliver verified design and design verification test benches, design and verification software tools and more

CHIPS ALLIANCE

Membership

- Like other projects of the Linux Foundation, this project is funded through membership dues and contributed engineering resources
- Membership levels include:
  Platinum, Gold, Silver, Auditor, Individual

CHIPS ALLIANCE

Events

- First CHIPS Alliance workshop:
  - Held in Mountain View, June 19 2019
- In preparation:
  - Design verification workshop (Munich, Nov 14-15) - announced
  - January 29th - CHIPS Alliance 1-day workshop and CHISEL workgroup Workshop
  - 2nd workshop (Shanghai, early March 2020)

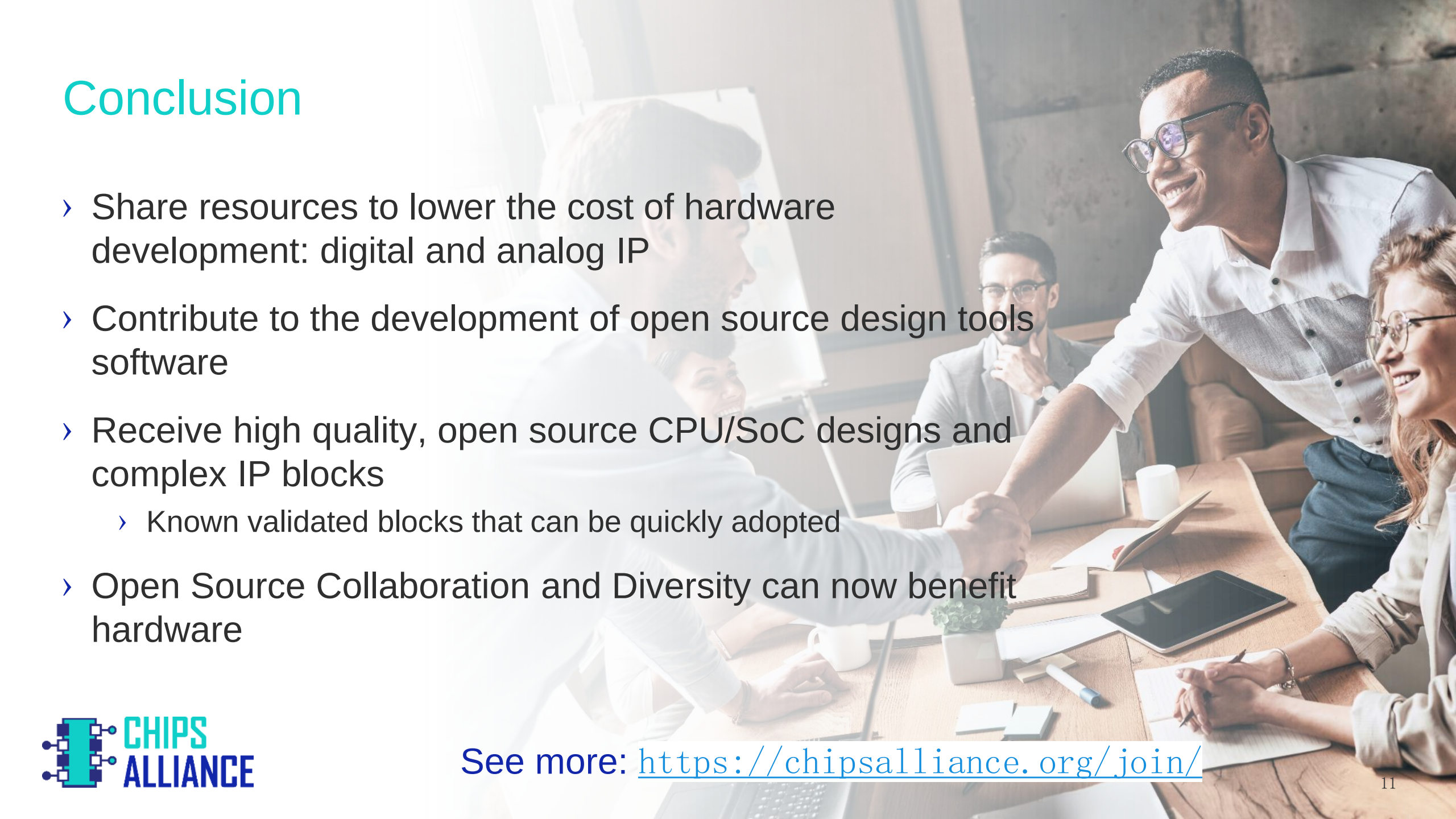
CHIPS ALLIANCE

Workgroups

- Chisel-WG
- Tools-WG:
  - Verilator
  - FuseSOC
  - Cocotb-verilator
- Cores-WG:
  - SweRV
- Interconnect:
  - TileLink 2.0
  - OmniXtend

CHIPS ALLIANCE

Conclusion



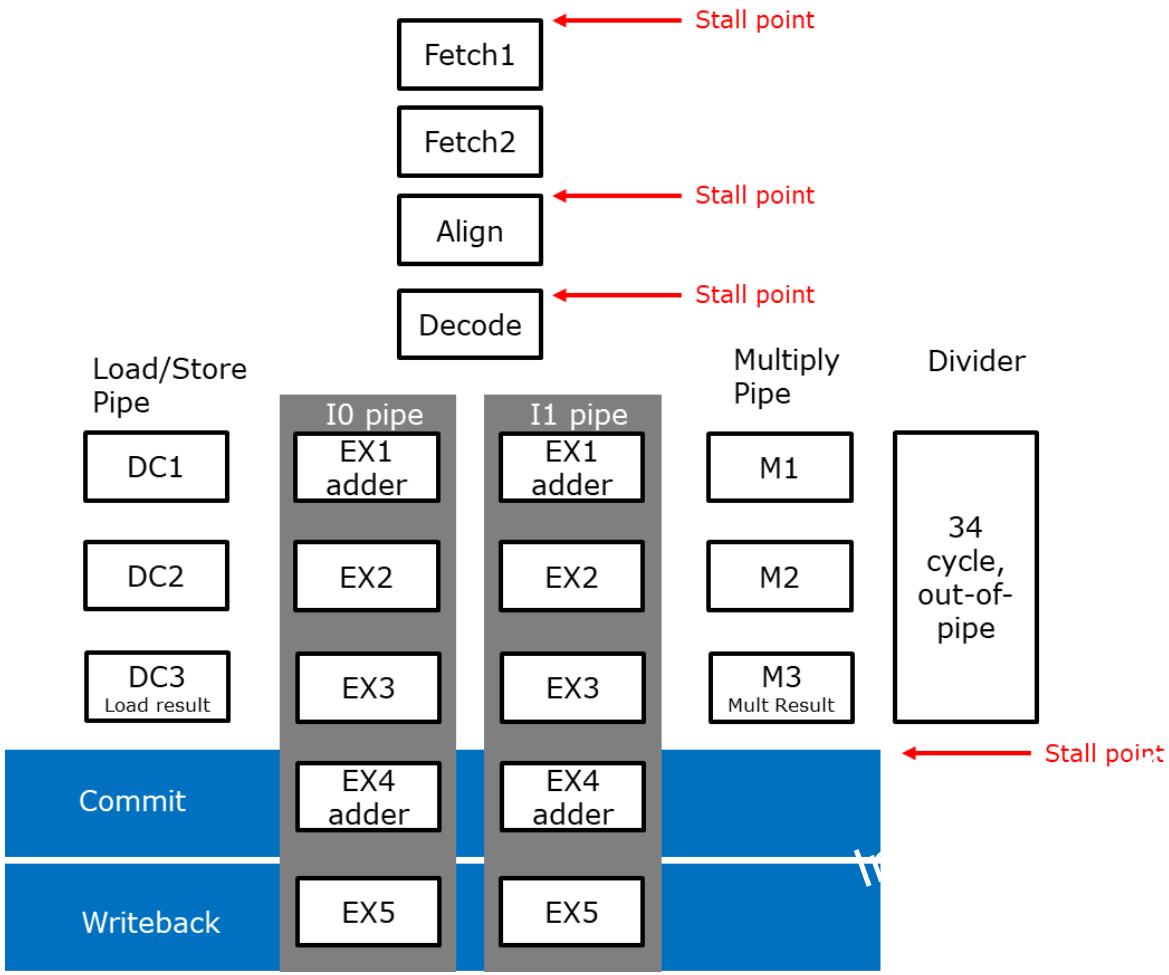
- Share resources to lower the cost of hardware development: digital and analog IP
- Contribute to the development of open source design tools software
- Receive high quality, open source CPU/SoC designs and complex IP blocks
  - Known validated blocks that can be quickly adopted
- Open Source Collaboration and Diversity can now benefit hardware

CHIPS ALLIANCE

See more: https://chipsalliance.org/join/

CHIPS Alliance projects

SweRV™ core microarchitecture

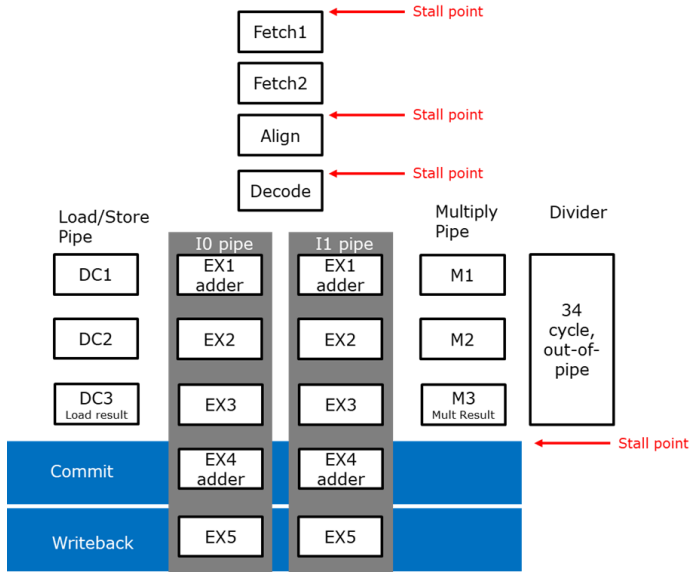


- 9 stage pipeline
- 4 stall points
  - Fetch1
    - Cache misses, line fills
  - Align
    - Form instructions from 3 fetch buffers
  - Decode
    - Decode up to 2 instructions from 4 instruction buffers
  - Commit
    - Commit up to 2 instructions / cycle
- EX pipes
  - ALU ops statically assigned to I0, I1 pipes
  - ALU's are symmetric
- Load/store pipe
  - Load-to-use of 2
- Multiply pipe
  - 3 cycle latency
- Divide pipe
  - 34 cycles, out-of-pipe

Pipeline diagram

L1: ld x11,8(x10)	
L2: ld x13, 8(x12)	
L3: ld x14, 8(x11)	# L1
A4: addi x15,x13,1	# L2
A5: add x16, x13, x14	# L2, L3
L6: ld x17,8(x16)	# A5
A7: addi x17,x17,1	# L6

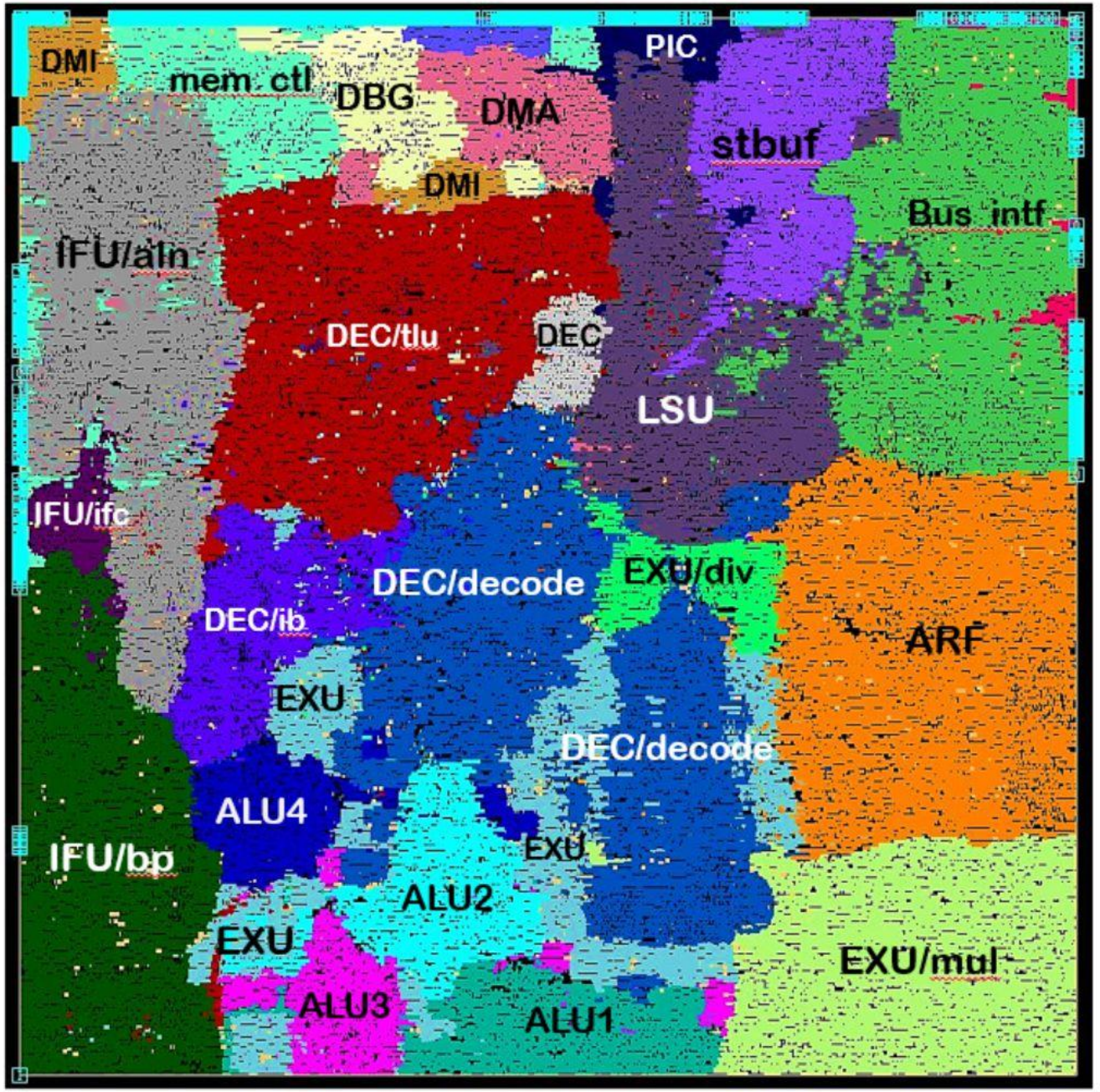
depends on



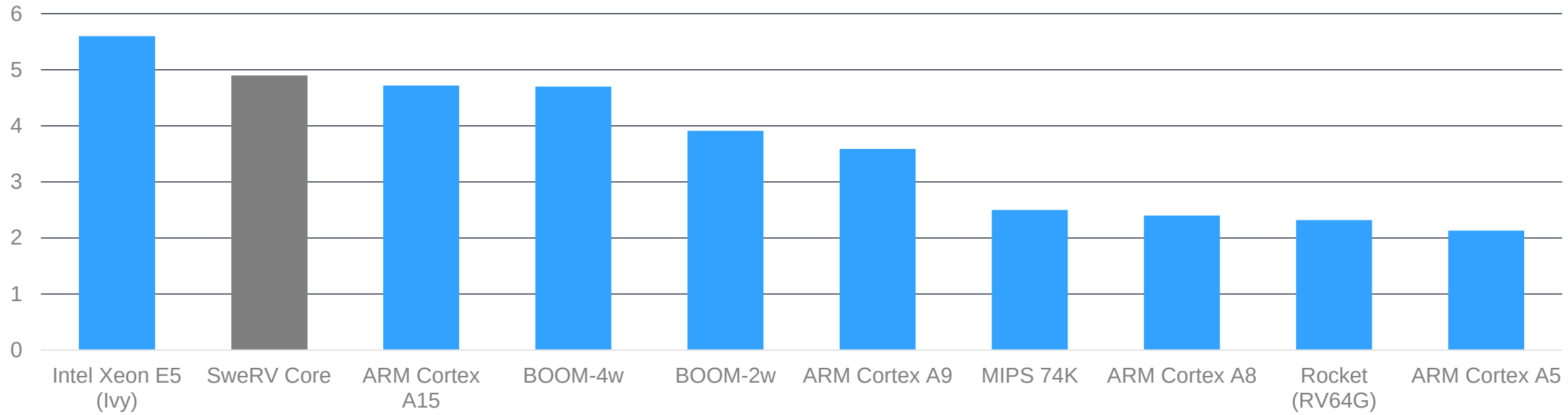
	1	2	3	4	5	6	7	8	9	10	11	12
DECODE	L1	L2	L3,A4			A5	L6,A7					
EX1/DC1		L1	L2	L3,A4			A5	L6,A7				
EX2/DC2			L1	L2	L3,A4			A5	L6,A7			
EX3/DC3				L1	L2	L3, A4			A5	L6,A7		
EX4/COM					L1	L2	L3, A4			A5	L6,A7	
EX5/WB						L1	L2	L3, A4			A5	L6,A7

SweRV Core Physical Design

- TSMC 28 nm
  - 125 C, SVT, 150 ps clock skew
- SSG corner w/out memories
  - 1 GHZ
    - .132 mm^2
  - 800 MHZ
    - .100 mm^2
  - 500 MHZ
    - .093 mm^2
- TT corner w/out memories
  - 1 GHZ
    - .092 mm^2
  - 800 MHZ
    - .091 mm^2
  - 500 MHZ
    - .088 mm^2



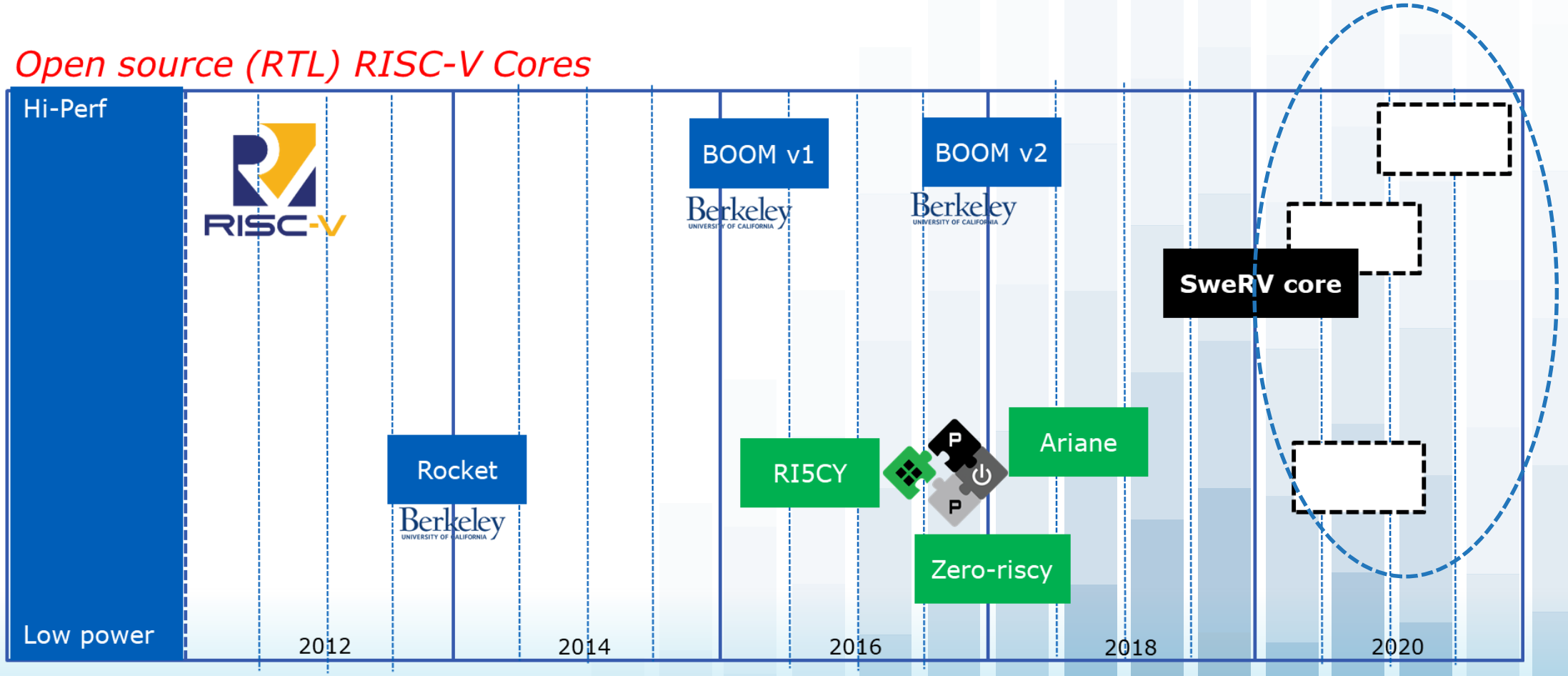
SweRV Core Performance



- 4.9 CoreMark/MHz
  - Additional performance gains are possible with compiler optimizations
  - Multi-threaded/multi-core results are always renormalized to a single execution context
- 2.9 Dhrystone MIPs/MHz
  - Using optimized strcpy function

CoreMark data from C.Celio, D.Patterson, K.Asanovic,https://www2.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-167.pdf

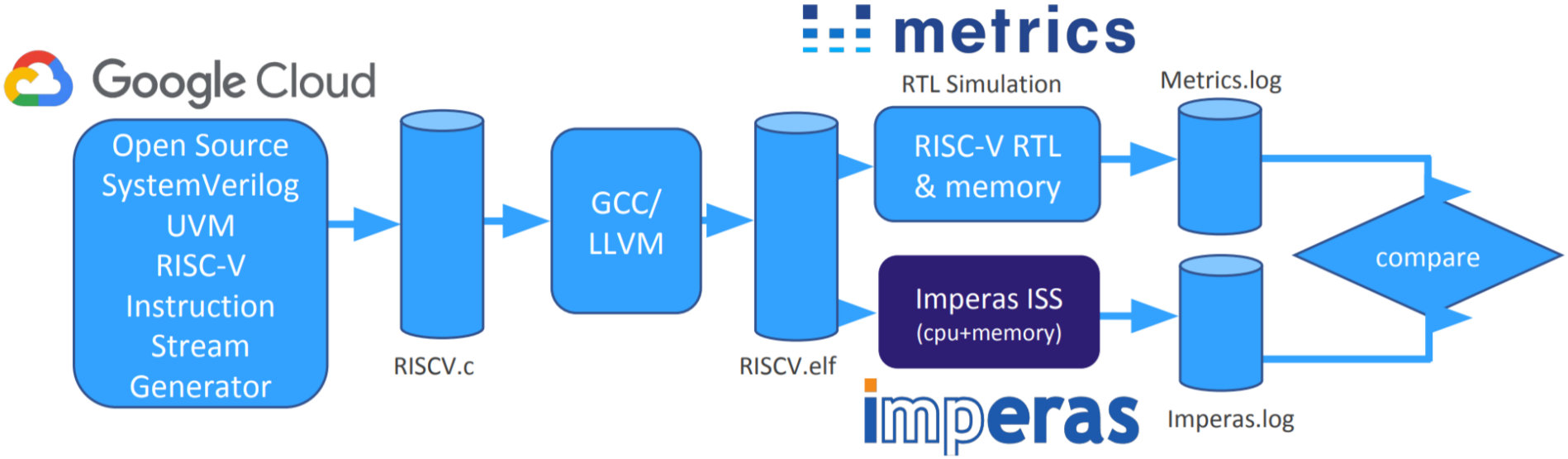
SweRV™ line of open-sourced cores fills important market segments



- SweRV core addresses high performance embedded requirements, increasing performance to 5 CM/MHz while keeping size in 0.1 mm^2 range

CHIPS ALLIANCE

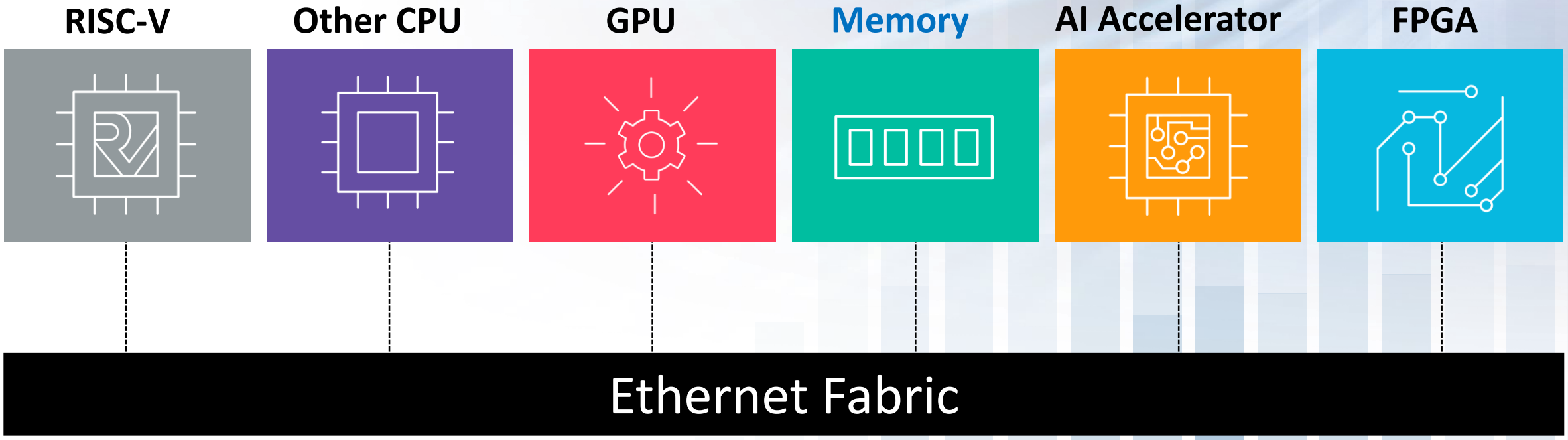
Design Verification using Co-Sim with reference model



- Google: open source Stressful Transaction & Instruction Generator (STIG):
  - STIG will drive RISC-V core under test through corner cases and push it to the limit
  - A high quality SystemVerilog, UVM DV infrastructure
- Metrics : SystemVerilog design + UVM simulator for RTL
- Imperas: model and simulation golden reference of RISC-V CPU

CHIPS ALLIANCE

OmniXtend: a truly open high performance memory fabric



Data is the center of the architecture

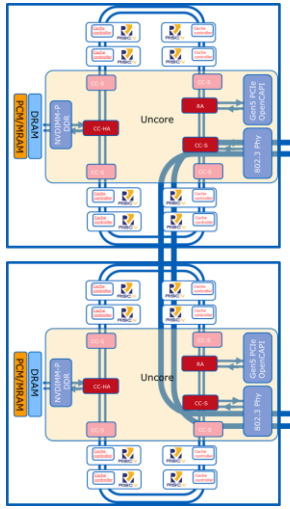
No established hierarchy – CPU doesn't 'own' the GPU or the Memory

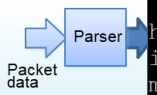
Cache Coherency preserved system-wide over the Fabric

CHIPS ALLIANCE

OmniXtend architecture overview

RISC-V node

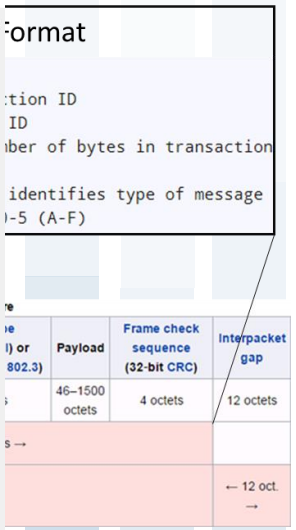


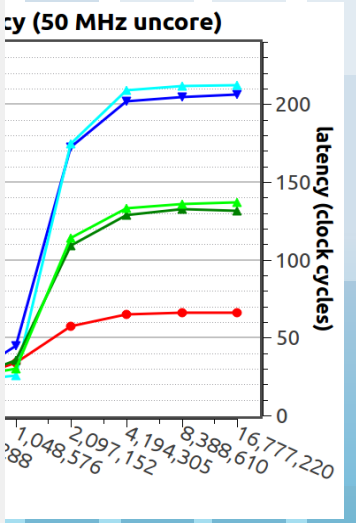


10.11.46.137 - PuTTY

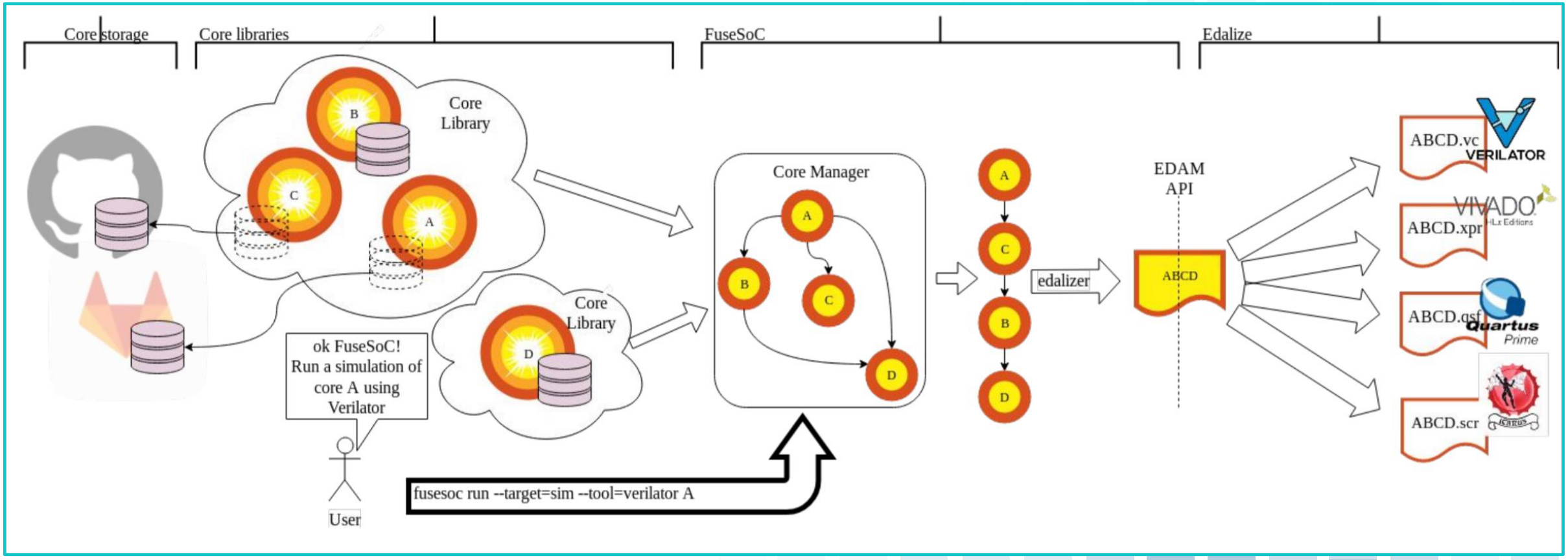
```
0
# cat /proc/cpuinfo
hart    : 1
isa     : rv64imafdc
mmu     : sv39
uarch   : sifive,rocket0

hart    : 2
isa     : rv64imafdc
mmu     : sv39
uarch   : sifive,rocket0

hart    : 3
isa     : rv64imafdc
mmu     : sv39
uarch   : sifive,rocket0

hart    : 4
isa     : rv64imafdc
mmu     : sv39
uarch   : sifive,rocket0

hart    : 9
isa     : rv64imafdc
mmu     : sv39
uarch   : sifive,rocket0

hart    : 10
isa     : rv64imafdc
mmu     : sv39
uarch   : sifive,rocket0

hart    : 11
isa     : rv64imafdc
mmu     : sv39
uarch   : sifive,rocket0

hart    : 12
isa     : rv64imafdc
mmu     : sv39
uarch   : sifive,rocket0

# cat /proc/cpuinfo | grep hart | wc -l
8
#
```

```
hart    : 12
isa     : rv64imafdc
mmu     : sv39
uarch   : sifive,rocket0

# cat /proc/cpuinfo | grep hart | wc -l
8
#
```



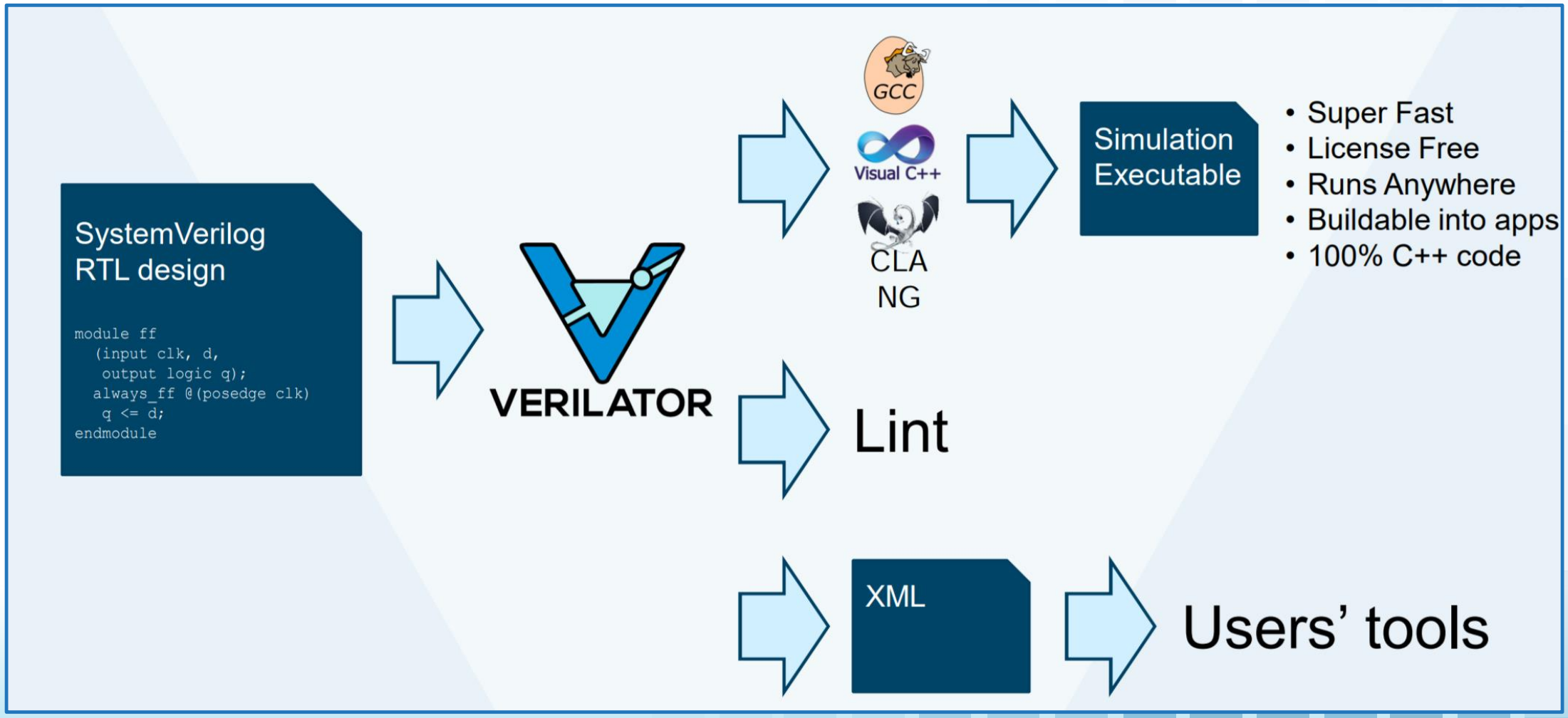


FuseSOC (and SweRV support!)



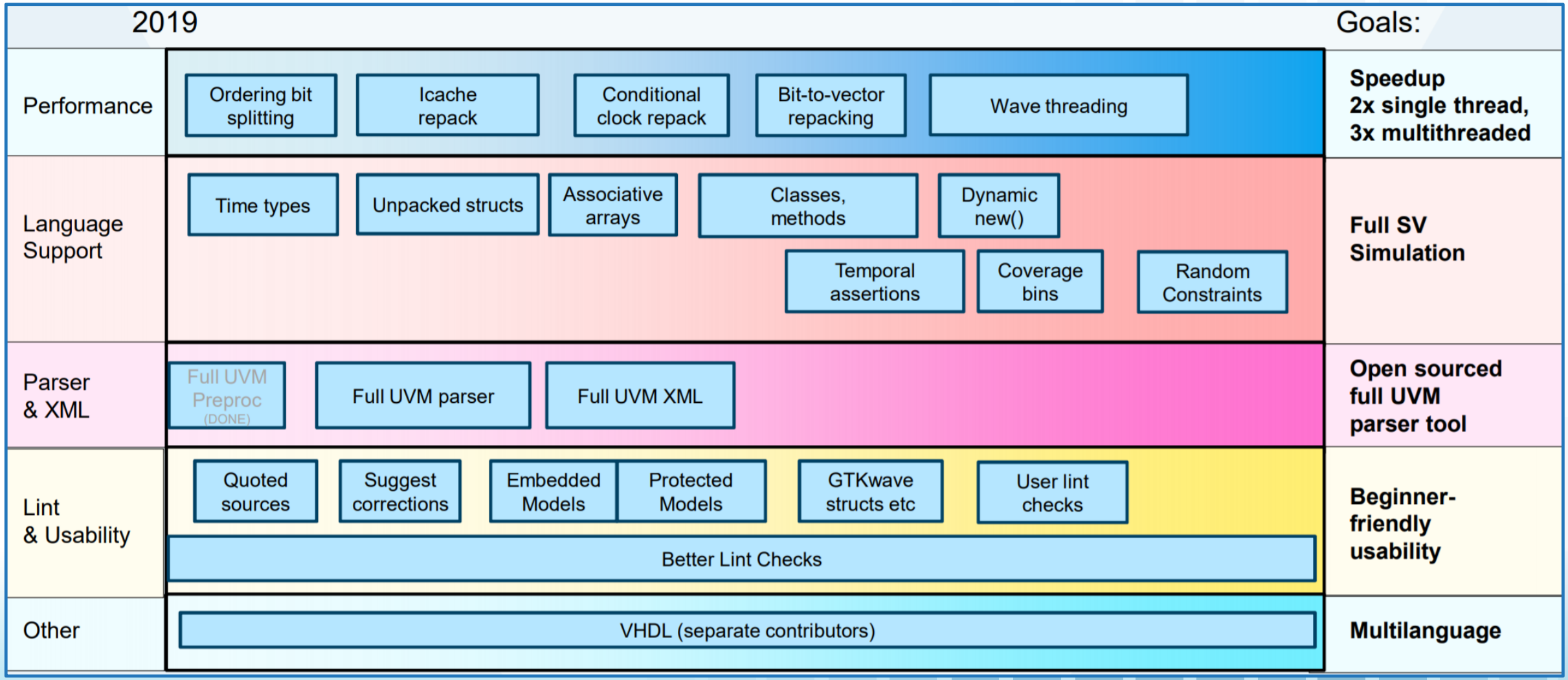
› FuseSoC is a package manager… ...and a build tool for HDL

Verilator



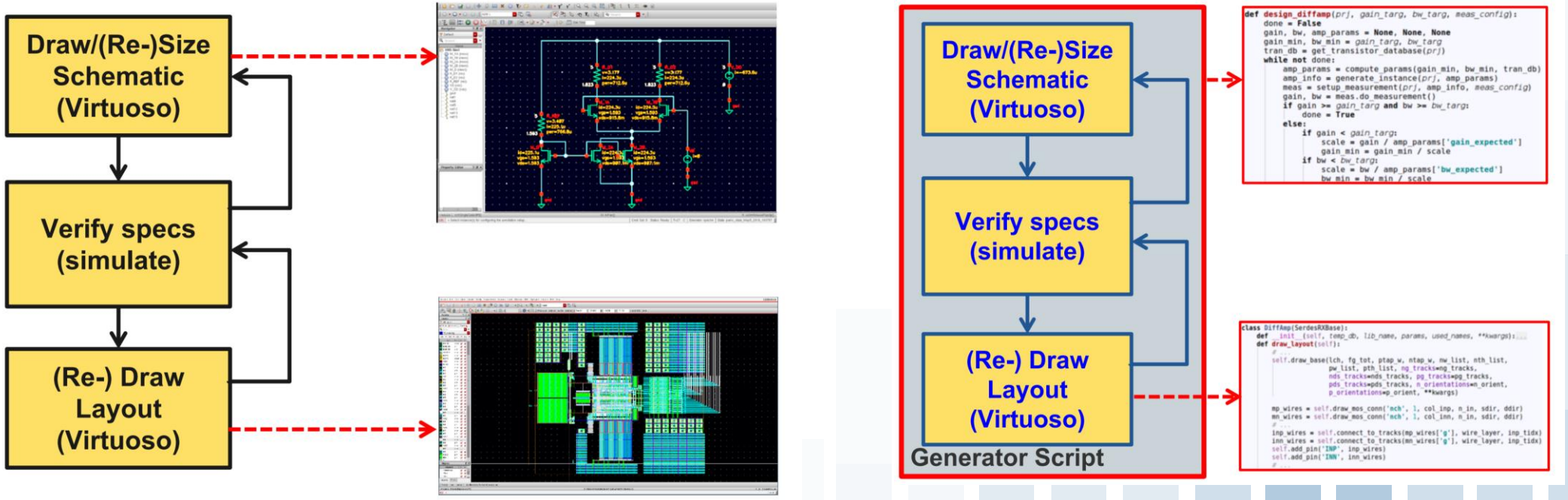
CHIPS ALLIANCE

Verilator roadmap



CHIPS ALLIANCE

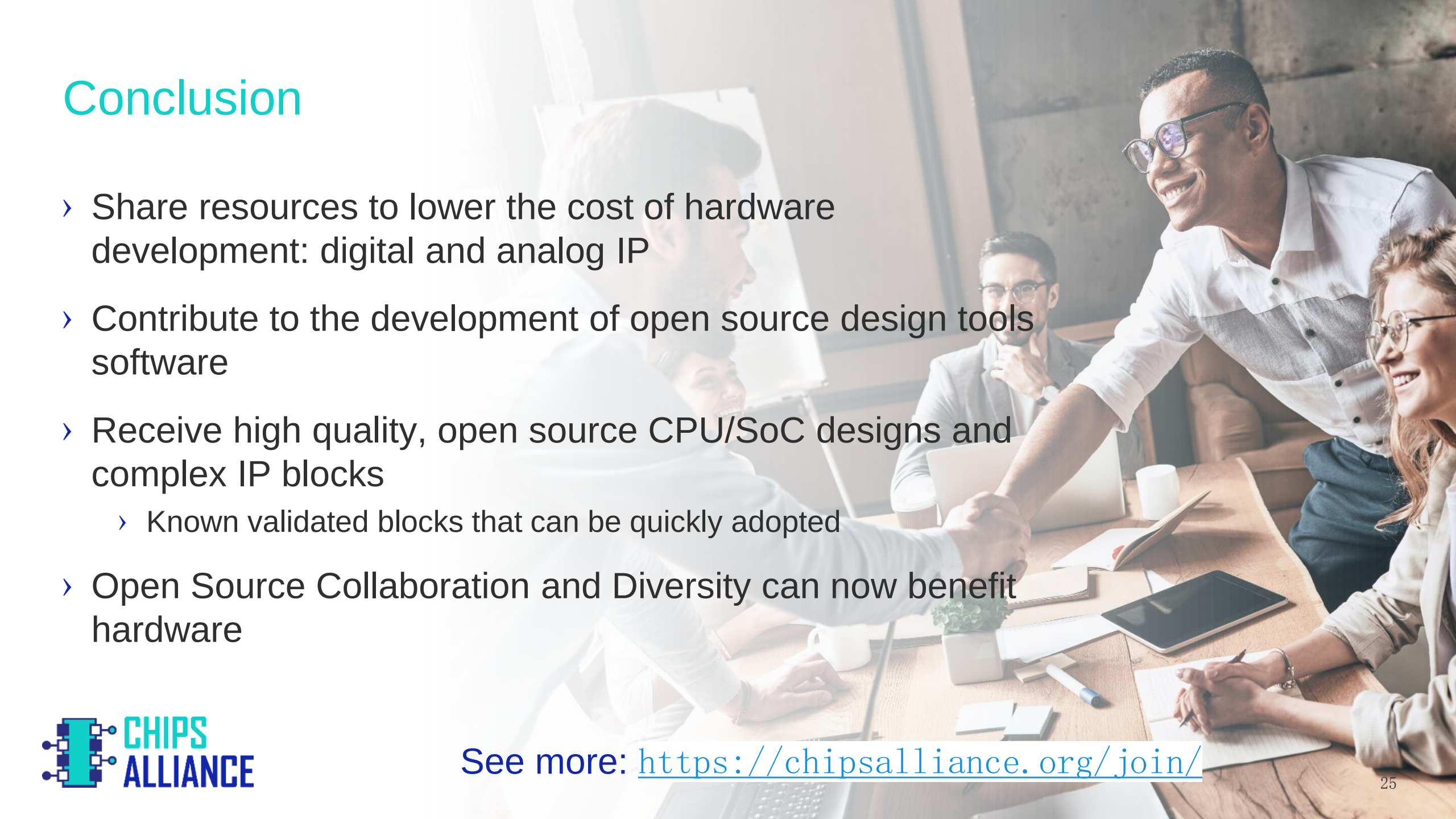
BAG: Berkeley Analog Generator



- Core design loop has not changed in 30+ years
- Captures design knowledge in an executable generator

CHIPS ALLIANCE

Conclusion



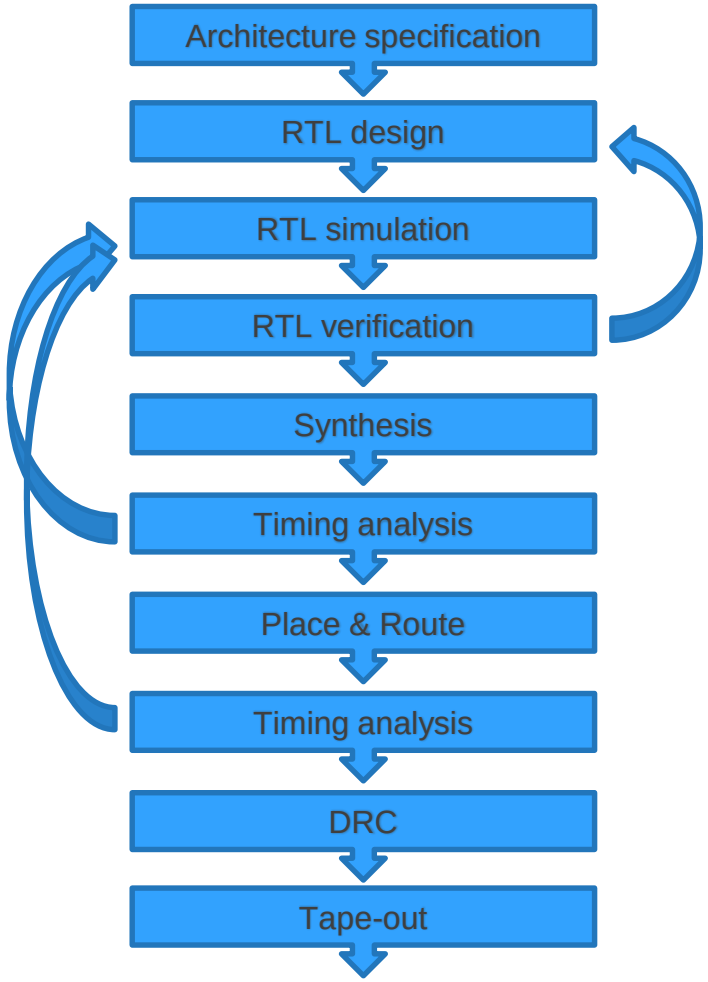
- Share resources to lower the cost of hardware development: digital and analog IP
- Contribute to the development of open source design tools software
- Receive high quality, open source CPU/SoC designs and complex IP blocks
  - Known validated blocks that can be quickly adopted
- Open Source Collaboration and Diversity can now benefit hardware

CHIPS ALLIANCE

See more: https://chipsalliance.org/join/

backup

Examples of open source hardware and design tools contributions



- Open source RTL designs:
  - Compute cores (SweRV, Rocket)
  - Key interfaces (OmniXtend)
  - AI blocks
  - CPUs - (Linux of computers)
- Interconnects:
  - OmniXtend cache coherence over Ethernet
  - Phy for Chiplets
- Modern design tools:
  - Chisel and FIRRTL
  - FuseSOC
- Addressing RTL simulation and design verification:
  - UVM Stressful Instruction generation
  - Verilator and System Verilog roadmap
  - Cocotb project in collaboration with FOSSI
- Long term goal:
  - a collaborative and innovative open source hardware ecosystem

Project Deliverables

- The scope of the Project includes hardware and software design and development under an open source (Apache v2) license:
  - Verified IP blocks (compute cores, accelerators etc)
  - Verified SoC designs (based on RISC-V and other open source cores)
  - Open source software development tools for ASIC development
  - Other high value IP including analog:
    - Peripherals, Mixed Signal Blocks and Compute Acceleration
- New design flows exploration:
  - Python based design verification

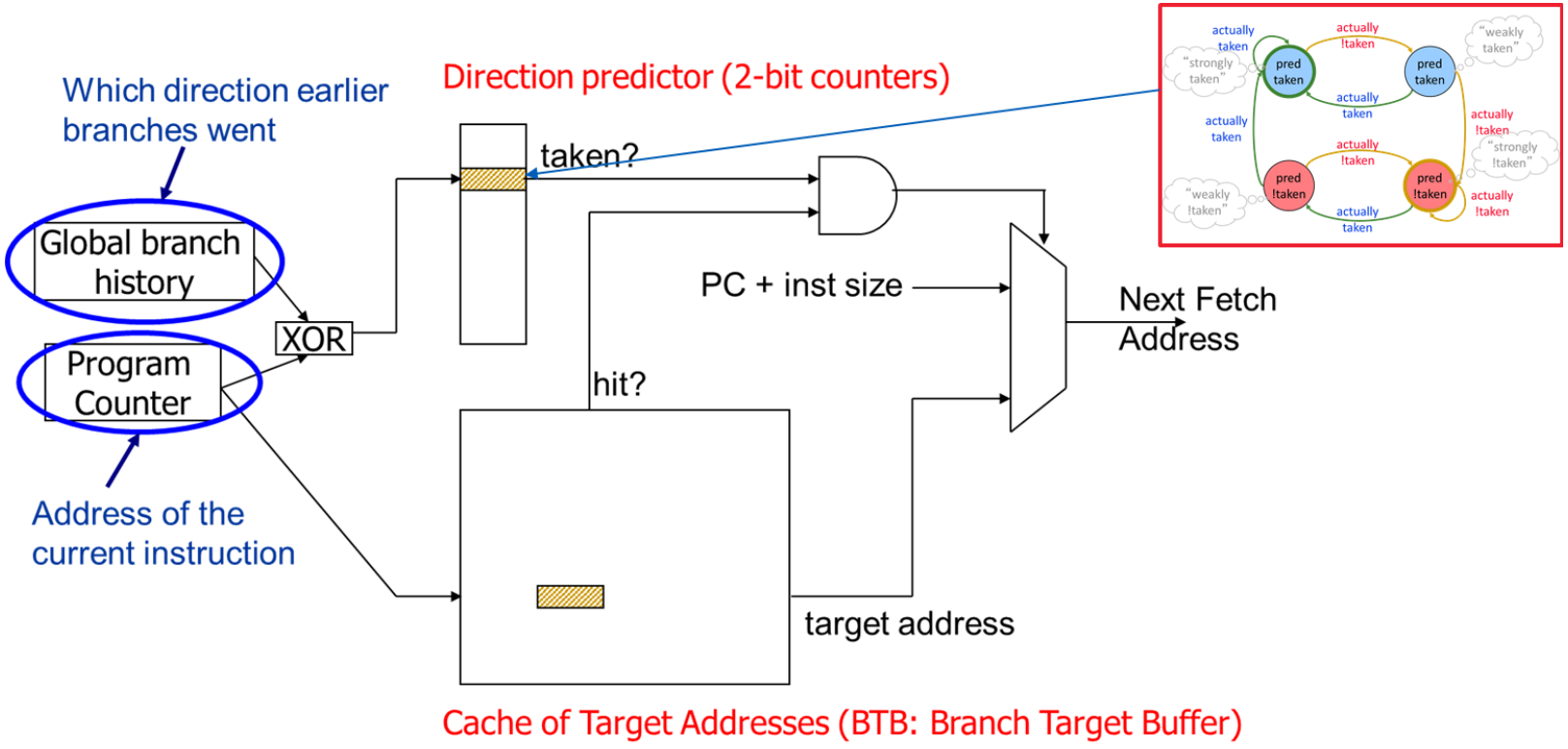
FOSSi Foundation and CHIPS collaboration

Open Hardware Ecosystems

- RISC-V, CHIPS Alliance and OpenPower Foundations are working together with their members to standardize tools addressing the common requirements for open microprocessor design, development and production
  - This will include IP, compliance, design, validation and open source tools
  - Builds on top of the associated open source software ecosystems
- OpenCAPI and OMI offer an architecture agnostic interconnect
  - Can be used across all microprocessor architectures including RISC-V, POWER, x86 and ARM?
- The RISC-V and POWER ISAs are both highly capable RISC architectures, the choice between them a matter of engineering requirements and use cases
- Both ecosystems also come together through CHIPS Alliance where organizations are working on open designs for IP blocks, cores, interconnects, and open source software design and verification tools

SweRV Core Branch Prediction / Branch Handling I

- Branch direction is predicted using GSHARE algorithm
  - XOR of global branch history and PC
    - Used to lookup branch direction in branch history table (BHT)
  - PC hash
    - Used to lookup branch target in branch target table (BTB)
  - The sizes of the branch target buffer (BTB) and the branch predictor table (BPT) are independently configurable with up to 512 and 2048 registers



SweRV Core Branch Prediction / Branch Handling II

- Branch direction is predicted using GSHARE algorithm
- Branches that hit in the BTB result in 1 cycle branch penalty
- Branches that mispredict in primary alu's result in 4 cycle branch penalty
- Branches that mispredict in secondary alu's result in 7 cycle branch penalty