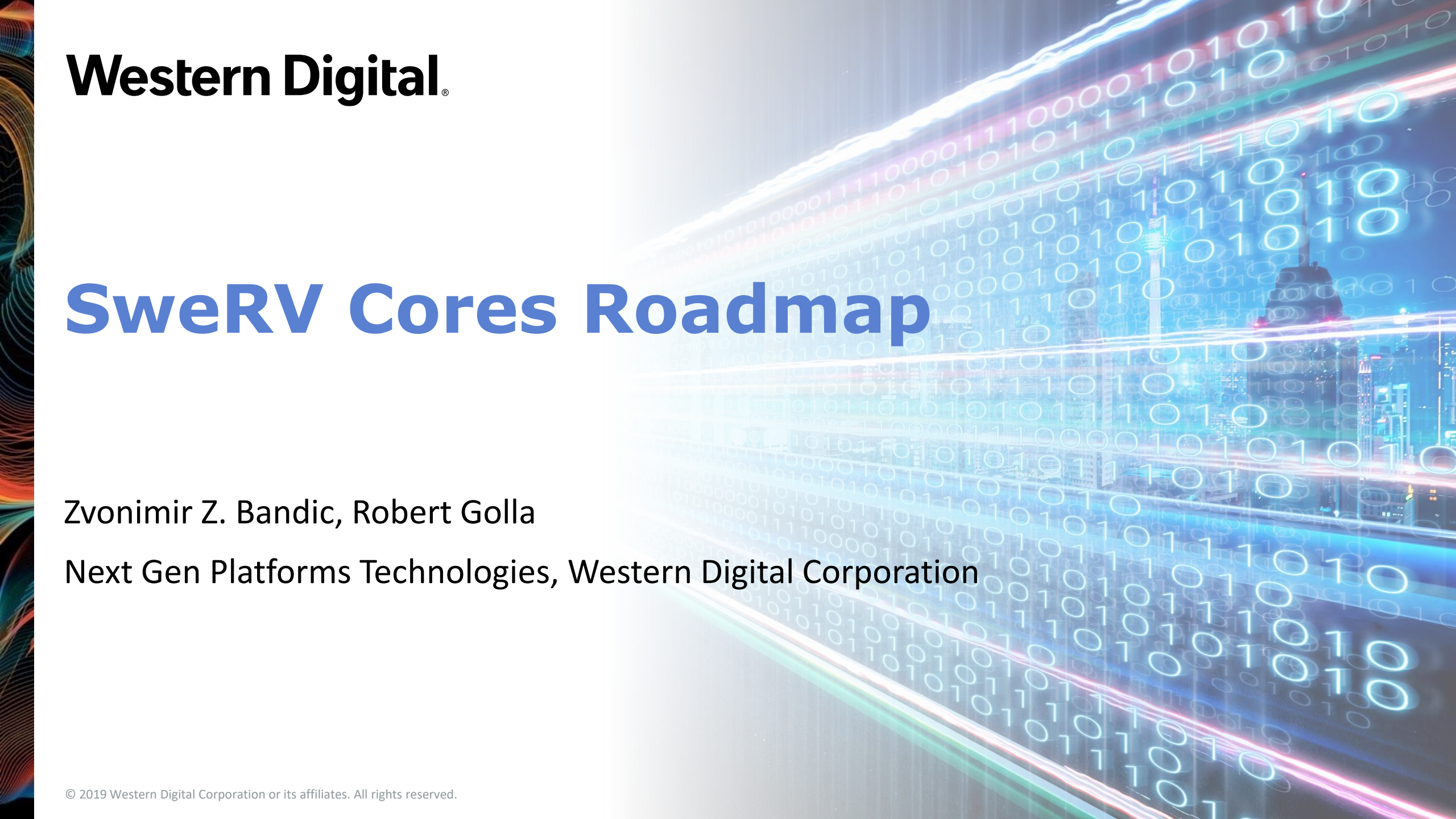
Western Digital®

SweRV Cores Roadmap

Zvonimir Z. Bandic, Robert Golla

Next Gen Platforms Technologies, Western Digital Corporation



Agenda

- Western Digital and RISC-V history
- SweRV Cores roadmap and CHIPS Alliance
- SweRV EH1 core recap:
  - Architecture and microarchitecture
  - Performance
- SweRV EH2 and EL2 cores:
  - Microarchitecture
  - Performance
  - Simultaneous multi-threading examples
  - Implementation areas
- RISC-V Flash Controller Vision
- Conclusions

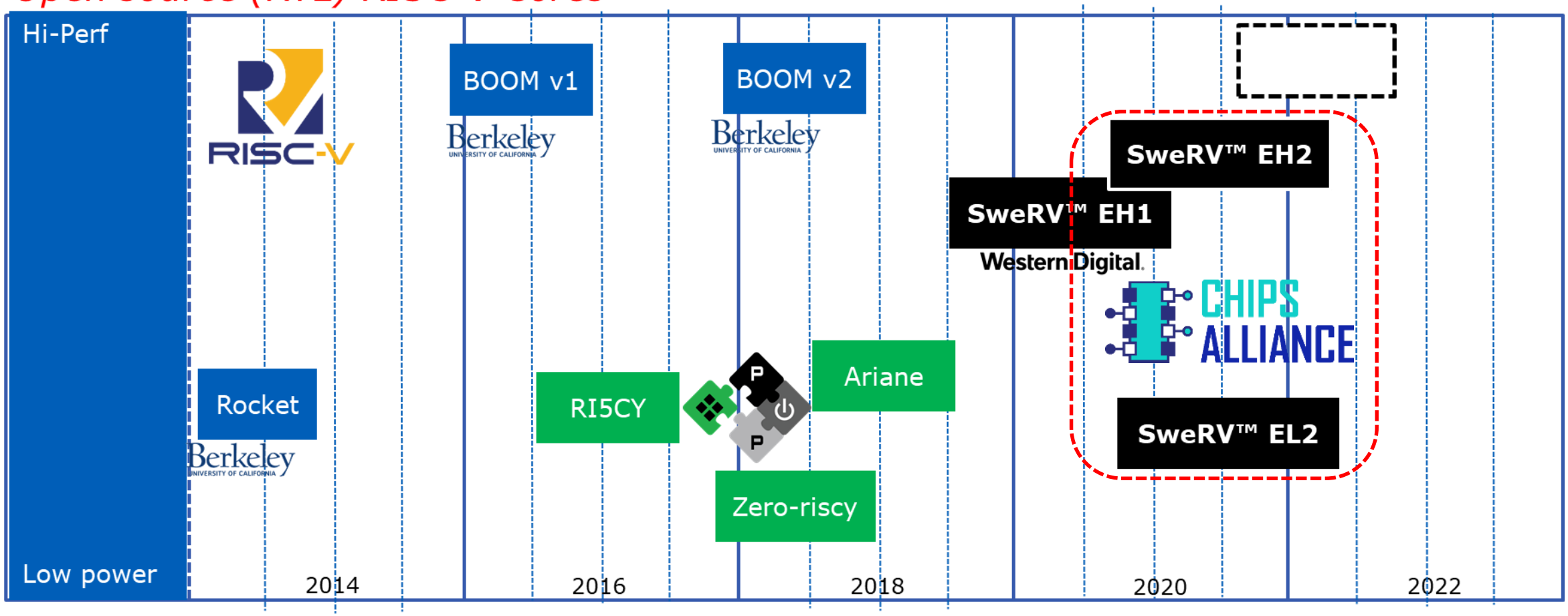
Western Digital RISC-V History

Motivated by our desire for open standard interfaces

2015	2016	2017	2018	2019
Platinum member	Explore RISC-V solutions	Announce plans to ship 1B cores	Contribute to Fedora Linux® & Arduino Cinco ports	Contribute SweRV™ Core EH1,PlatformIO, OmniXtend™, SBI, QEMU…

Open source RISC-V cores

Open source (RTL) RISC-V Cores

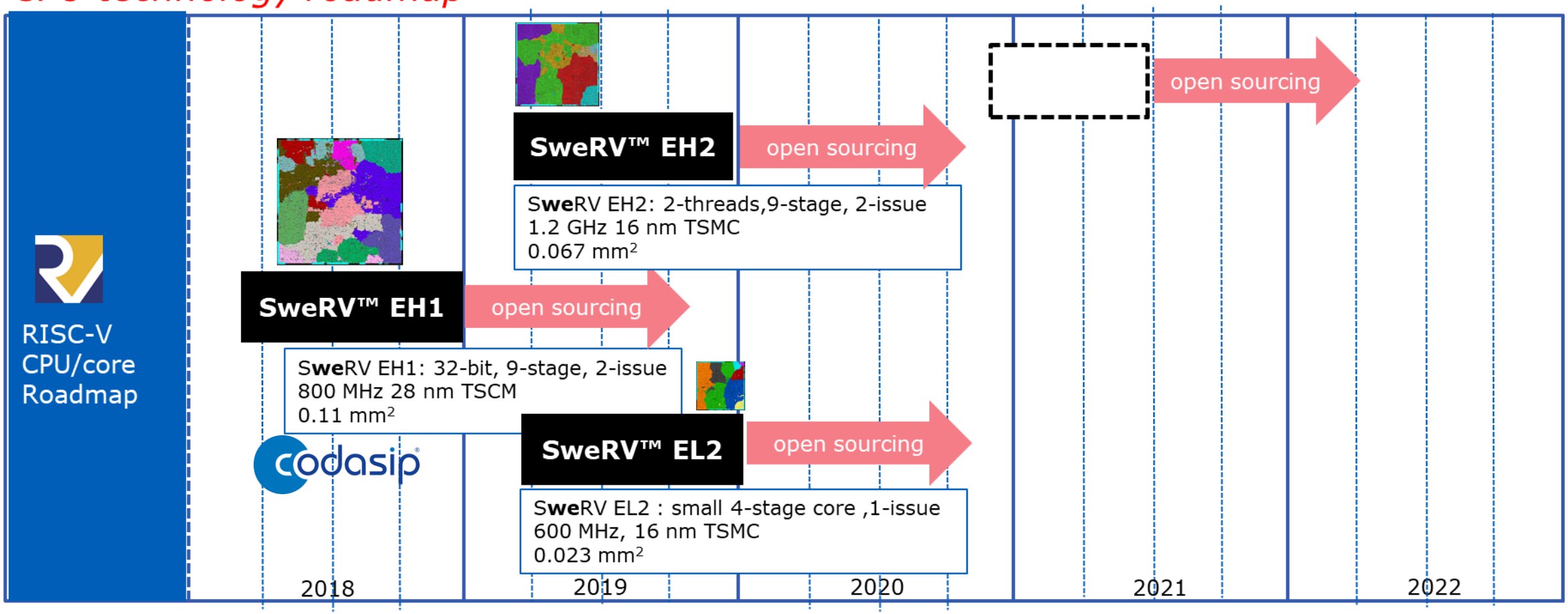


- SweRV EH1 core addresses high performance embedded requirements, increasing performance to 4.9 CM/MHz while keeping size in 0.1 mm^2 range at 28 nm TSMC. SweRV EH2 introduces multi-threading, while EL2 represents ultra-low power core for variety of SoC applications.

RISC-V Cores roadmap

CHIPS ALLIANCE

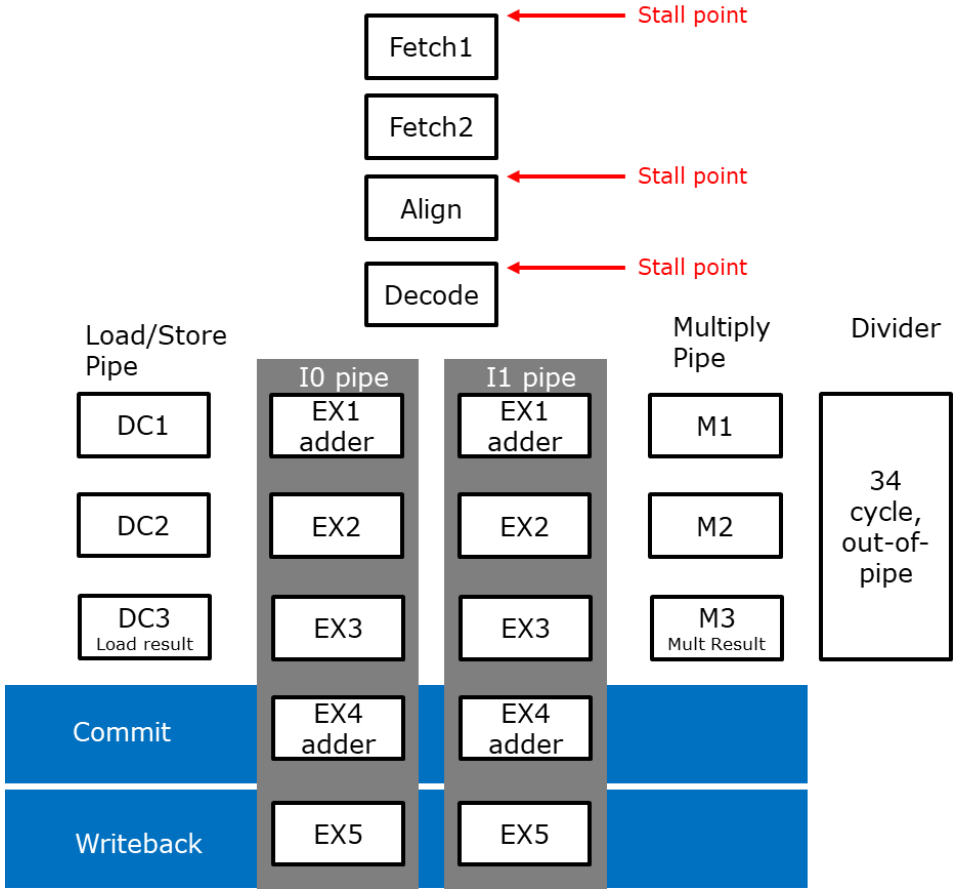
CPU technology roadmap



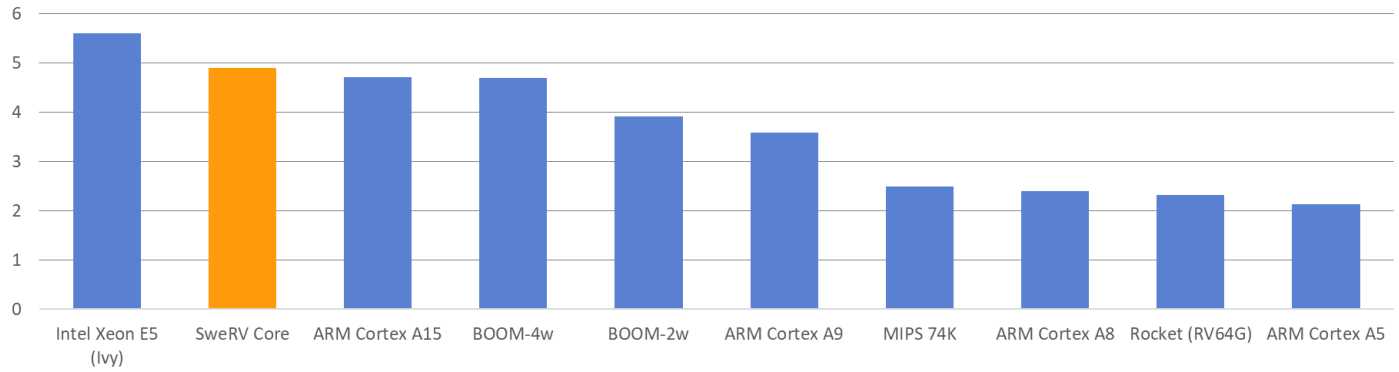
https://github.com/chipsalliance/Cores-SweRV

SweRV EH1 core recap

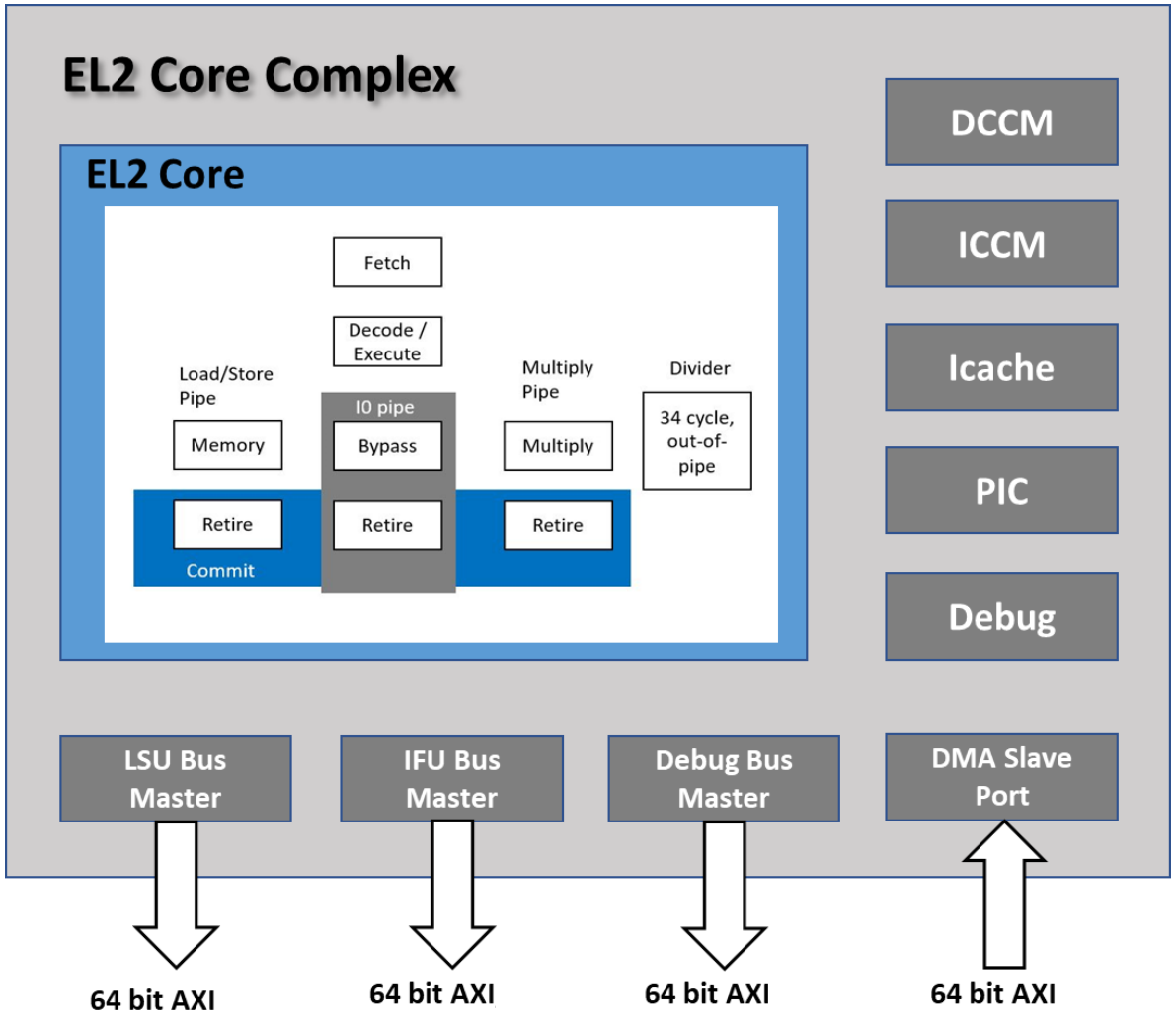
https://github.com/chipsalliance/Cores-SweRV



- 8 stage pipeline
  - Fetch to commit
- 3 stall points
- Execution 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
  - Up to 34 cycles, out-of-pipe
- Coremark score of 4.9 CM/MHz

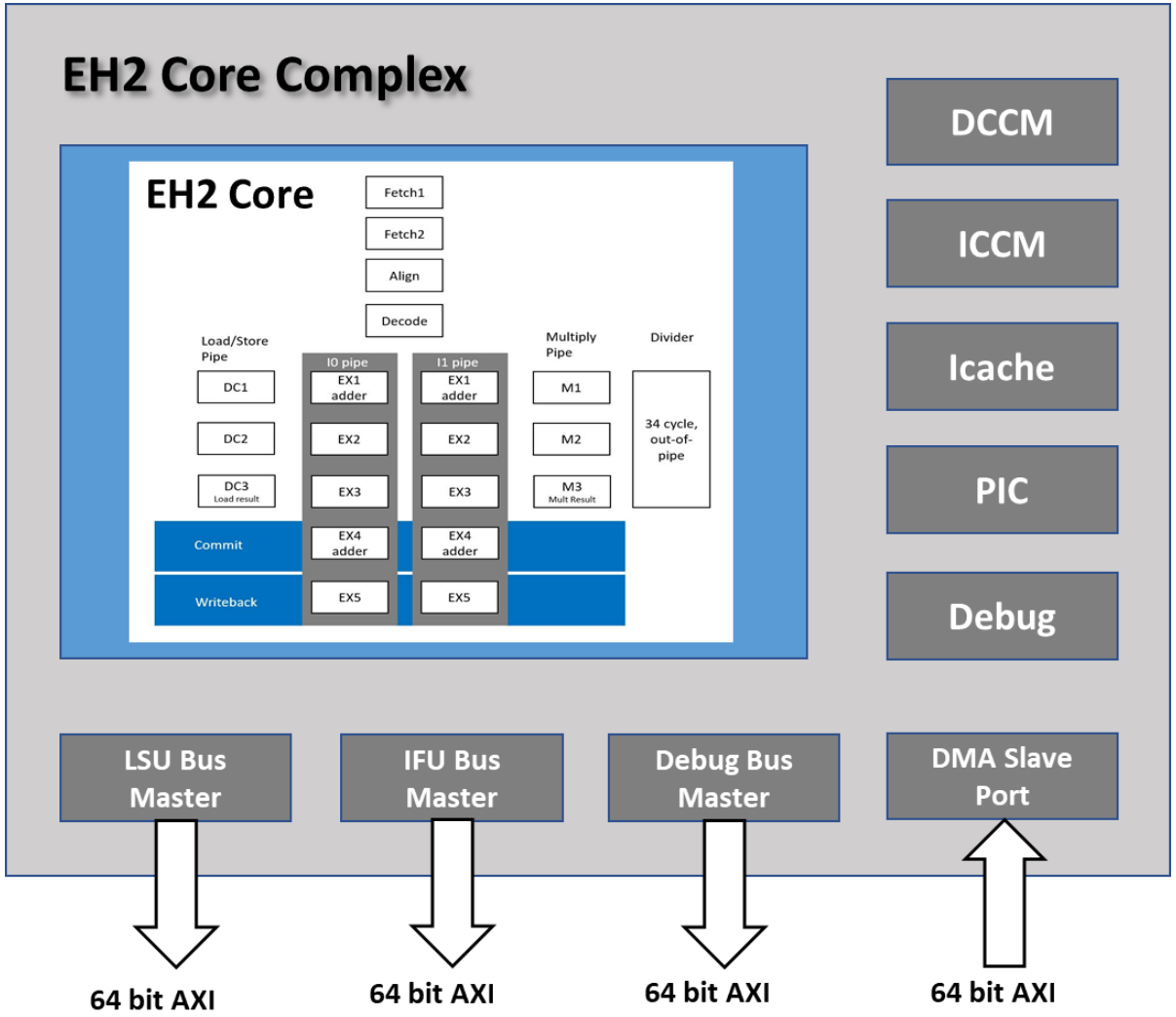


SweRV EL2 core



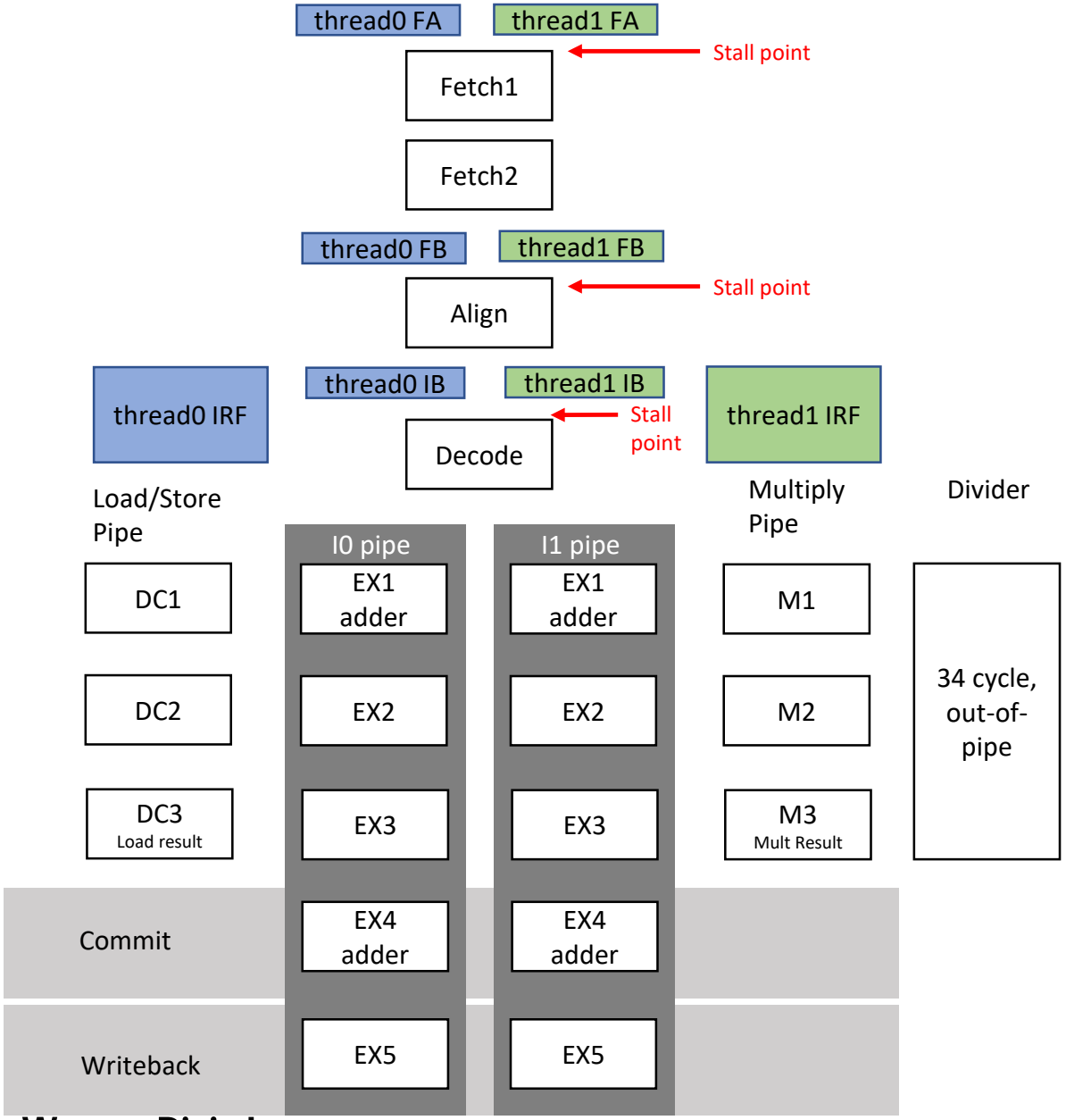
- EL2
  - 2nd generation design based on EH1
  - RV32IMC+Zbb+Zbs
  - 4 stage pipe to commit
  - Non-blocking microarchitecture
- Zero cycle branch target buffer
- Worst case branch mispredict penalty of 1 cycle
  - Close to 1 IPC on wide range of workloads
- Fast Interrupt support
  - Redirect directly to interrupt service routine upon interrupt
- Enhanced DMA support
  - Fully pipelined
- Dhrystone of 2.0
- Coremark/MHz of 3.6
- Target frequency of 600 MHz
  - 16 nm, worst case corner

SweRV EH2 core



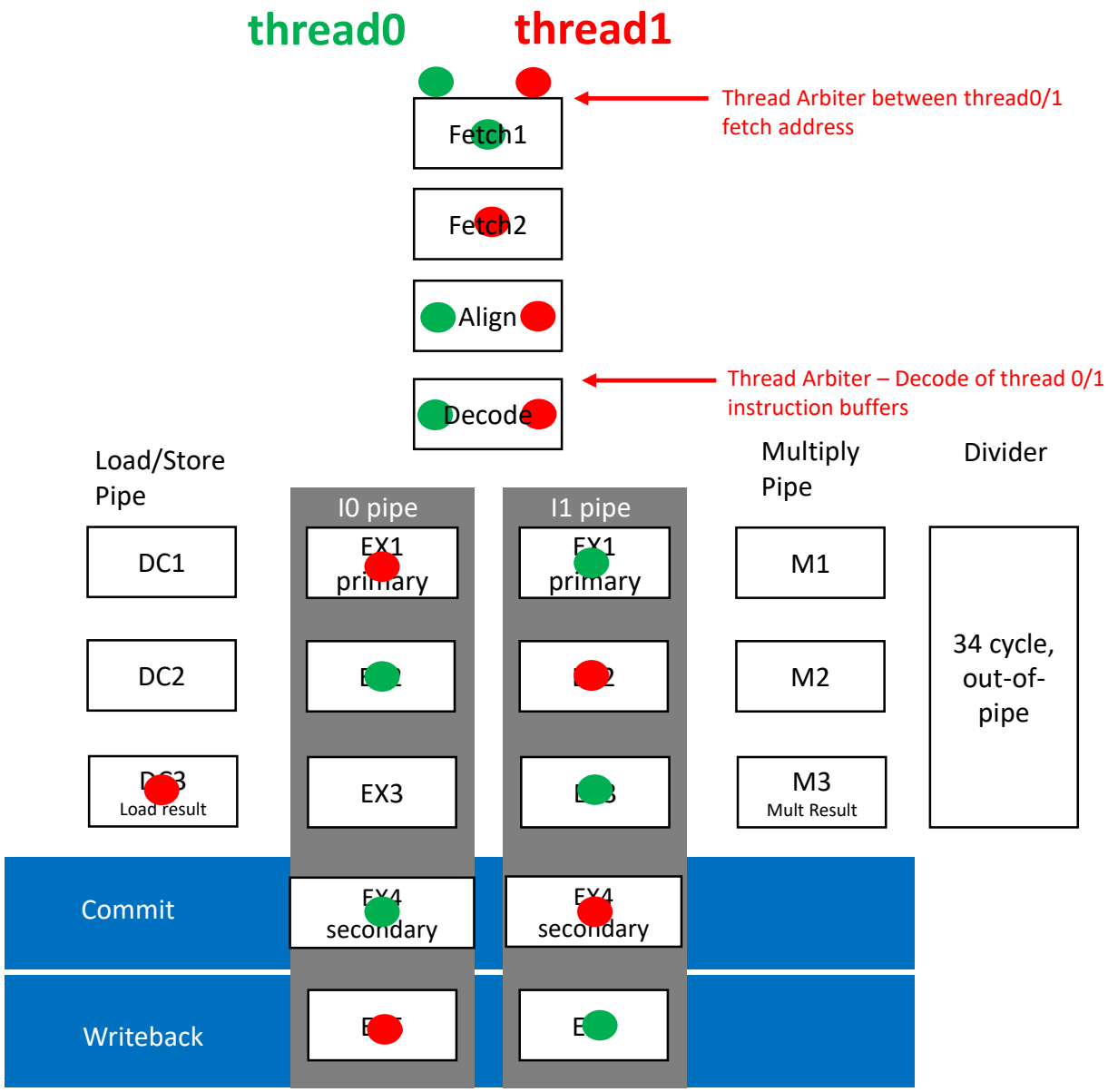
- EH2
  - 2nd generation design based on EH1
  - RV32IMAC+Zbb+Zbs
  - 8 stage pipe to commit
  - Non-blocking microarchitecture
- Multithread support
  - Up to 2 threads
  - Typical throughput gains of 1.6 to 2X
    - When ST IPC is 1.0 or less
- Fast Interrupt support
  - Redirect directly to interrupt service routine upon interrupt
- Enhanced DMA support
  - Fully pipelined
- Dhrystone of 2.9
- Coremark/MHz
  - 4.9 for 1 thread
  - 6.3 for 2 threads
- Target frequency of 1.2 GHz
  - 16 nm, worst case corner

SweRV EH2 pipeline



- Fetch Address (FA)
  - Each thread has a dedicated fetch address register
  - An LRU arbiter selects up to 1 thread to fetch each cycle
- Fetch Buffers (FB)
  - Each thread has 4 fetch buffers
    - 8B wide
    - Holds instructions from ICCM or I$
  - Dedicated aligner per thread
- Instruction Buffers (IB)
  - Each thread has 4 instruction buffers
  - Each cycle up to 2 threads are picked each cycle for decode
    - Typically one instruction from each thread is chosen if both threads are ready
    - Arbiter attempts to schedule the threads optimally
- Integer Register File (IRF)
  - Each thread has 31 integer registers

SweRV EH2 pipeline: SMT example



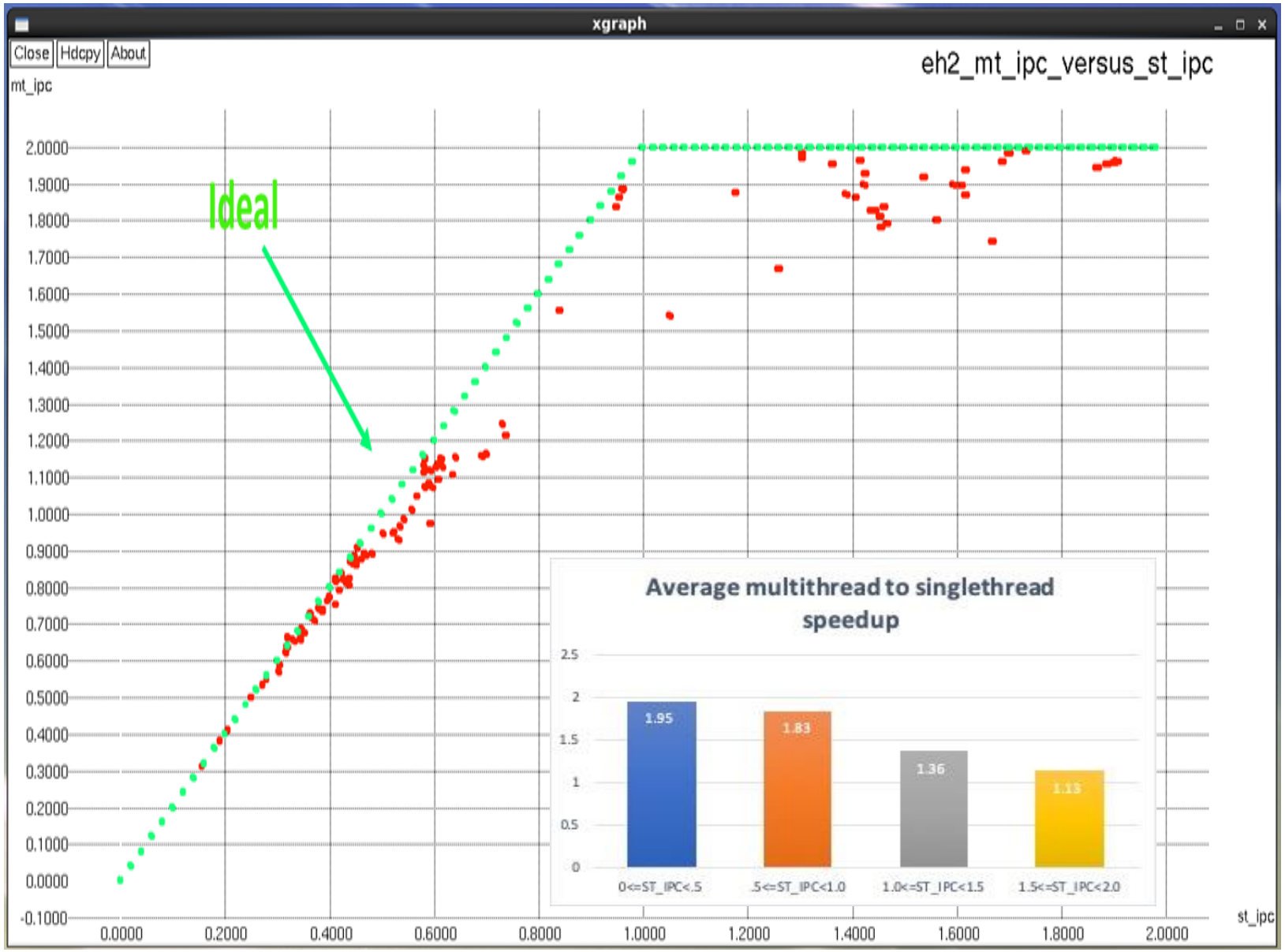
- Thread arbitration done at
  - Fetch
  - Decode
- Above aligner
  - Threading is “vertical”
    - Only 1 thread per pipe state
- Aligner and below
  - Threading is “simultaneous”
    - Multiple threads per pipeline stage
- Commit stage
  - Commit up to 2 different threads per cycle
  - Commit groups == decode groups

EH2 msort example

```
@83000  0 #47886  8000039c blt x14, x13, . + 0x30 (NT)      0 #47887  800003a0 c.sw x13, 0x0(x15)
@83001  0 #47888  800003a2 c.addi x15, 0x4                  0 #47889  800003a4 c.addi x12, 0x1
@83002  0 #47890  800003a6 bne x15, x16, . - 0x2a (T)       - ------- ----------------------------
@83003  0 #47891  8000037c slli x14, x12, 0x2               - ------- ----------------------------
@83004  0 #47892  80000380 add x13, x20, x14                0 #47893  80000384 slli x14, x10, 0x2
@83005  0 #47894  80000388 c.add x14, x9                    0 #47895  8000038a bgeu x10, x19, . - 0x1a (NT)
@83006  0 #47896  8000038e slli x13, x12, 0x2               - ------- ----------------------------
@83007  0 #47897  80000392 c.add x13, x20                   0 #47898  80000394 c.lw x14, 0x0(x14) [0xc0680000]
@83008  0 #47899  80000396 bgeu x12, x21, . + 0x36 (NT)     0 #47900  8000039a c.lw x13, 0x0(x13) [0xc0680380]
@83009  1 #47883  8000039c blt x14, x13, . + 0x30 (NT)      - ------- ----------------------------
@83010  1 #47884  800003a0 c.sw x13, 0x0(x15)               - ------- ----------------------------
@83011  1 #47885  800003a2 c.addi x15, 0x4                  1 #47886  800003a4 c.addi x12, 0x1
@83012  1 #47887  800003a6 bne x15, x16, . - 0x2a (T)       1 #47888  8000037c slli x14, x12, 0x2
@83013  1 #47889  80000380 add x13, x20, x14                1 #47890  80000384 slli x14, x10, 0x
@83014  1 #47891  80000388 c.add x14, x9                    1 #47892  8000038a bgeu x10, x19, . - 0x1a (NT)
@83015  1 #47893  8000038e slli x13, x12, 0x2               - ------- ----------------------------
@83016  1 #47894  80000392 c.add x13, x20                   1 #47895  80000394 c.lw x14, 0x0(x14) [0xc0680400]
@83017  1 #47896  80000396 bgeu x12, x21, . + 0x36 (NT)     1 #47897  8000039a c.lw x13, 0x0(x13) [0xc0680780]
@83018  - ------- ----------------------------              - ------- ----------------------------
@83019  - ------- ----------------------------              - ------- ----------------------------
@83020  0 #47901  8000039c blt x14, x13, . + 0x30 (NT)      0 #47902  800003a0 c.sw x13, 0x0(x15)
@83021  0 #47903  800003a2 c.addi x15, 0x4                  0 #47904  800003a4 c.addi x12, 0x1
@83022  0 #47905  800003a6 bne x15, x16, . - 0x2a (T)       - ------- ----------------------------
@83023  0 #47906  8000037c slli x14, x12, 0x2               - ------- ----------------------------
@83024  0 #47907  80000380 add x13, x20, x14                0 #47908  80000384 slli x14, x10, 0x2
@83025  0 #47909  80000388 c.add x14, x9                    0 #47910  8000038a bgeu x10, x19, . - 0x1a (NT)
@83026  0 #47911  8000038e slli x13, x12, 0x2               - ------- ----------------------------
@83027  0 #47912  80000392 c.add x13, x20                   0 #47913  80000394 c.lw x14, 0x0(x14) [0xc0680000]
@83028  0 #47914  80000396 bgeu x12, x21, . + 0x36 (NT)     0 #47915  8000039a c.lw x13, 0x0(x13) [0xc0680384]
@83029  1 #47898  8000039c blt x14, x13, . + 0x30 (NT)      - ------- ----------------------------
@83030  1 #47899  800003a0 c.sw x13, 0x0(x15)               - ------- ----------------------------
@83031  1 #47900  800003a2 c.addi x15, 0x4                  1 #47901  800003a4 c.addi x12, 0x1
@83032  1 #47902  800003a6 bne x15, x16, . - 0x2a (T)       1 #47903  8000037c slli x14, x12, 0x2
@83033  1 #47904  80000380 add x13, x20, x14                1 #47905  80000384 slli x14, x10, 0x2
@83034  1 #47906  80000388 c.add x14, x9                    1 #47907  8000038a bgeu x10, x19, . - 0x1a (NT)
@83035  1 #47908  8000038e slli x13, x12, 0x2               - ------- ----------------------------
@83036  1 #47909  80000392 c.add x13, x20                   1 #47910  80000394 c.lw x14, 0x0(x14) [0xc0680400]
@83037  1 #47911  80000396 bgeu x12, x21, . + 0x36 (NT)     1 #47912  8000039a c.lw x13, 0x0(x13) [0xc0680784]
@83038  - ------- ----------------------------              - ------- ----------------------------
```

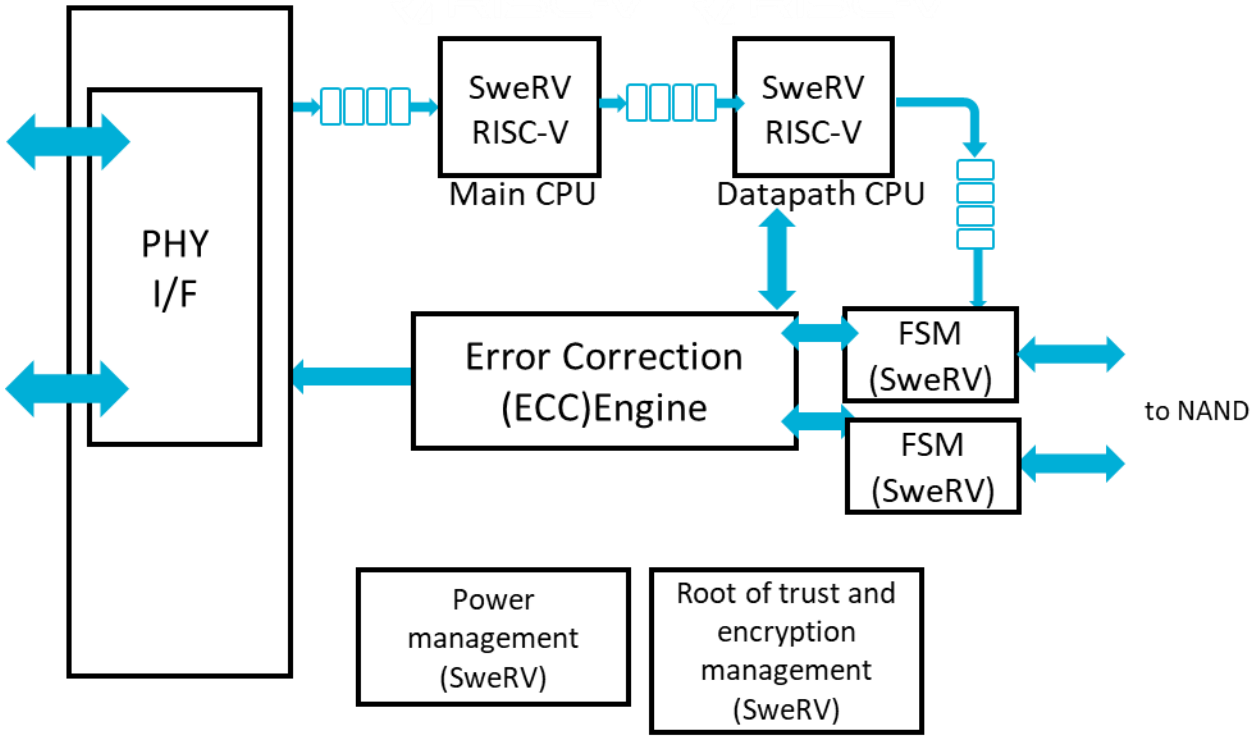
- Homogeneous
  - Msort running on both threads
- Thread colors
  - Thread 0
  - Thread 1
- Measured at Commit Stage
  - Commit up to 2 instructions per cycle
- IPC of .79 for single thread:
  - Load-to-use of 11 cycles from external bus
- IPC of 1.50 for multi-thread
  - Multithread speedup of 1.9X over single-thread on same workload
  - Load-to-use of 11 cycles from external bus

SweRV EL2 and EH2 Performance



- SweRV EL2 Performance
  - For applications that hit to I$/ICCM and DCCM
    - IPC will be close to 1.0
      - Coremark ~.95 IPC
      - Dhrystone ~.98 IPC
    - Microarchitecture designed for 1 IPC
- SweRV EH2 Performance
  - ST performance
    - Essentially same as EH1
  - MT performance
    - Measured on wide range of ~120 mixed and non-mixed workloads
    - For ST IPC < 1.0
      - Average speedup of 1.9X

Flash Controller Applications



- SweRV RISC-V customized cores are finding new applications in flash controllers:
  - Main CPU processing host attachment I/F
  - Datapath CPU
  - Finite State Machine sequencers controlling individual NAND channels
  - Power management
  - Root of trust and encryption management

Conclusions

- Western Digital continues its commitment to RISC-V:
  - Continuing plans to deliver 1 billion cores through application in storage controllers
- Western Digital is releasing the 2nd generation of SweRV cores:
  - Powerful dual-threaded, dual-issue SweRV EH2
  - Ultrasmall low-power SweRV EL2
- RTL of the cores will be shared with the open source community via CHIPS Alliance:
  - https://github.com/chipsalliance/Cores-SweRV

Western Digital®