SonicBOOM – The Third Generation Berkeley Out-of-Order Machine

Jerry Zhao, Ben Korpan, Abe Gonzalez, Krste Asanovic

UC Berkeley

jzh@berkeley.edu

Berkeley Architecture Research

Goal of the BOOM project

72x 8-wide OOO “Skylake”

4x 10-wide OOO “Sunny Lake”

2x 7-wide OOO “Vortex”
4x 3-wide OOO “Tempest”

2x 3-wide OOO “Tempest”

2x 9-wide OOO “Typhoon”

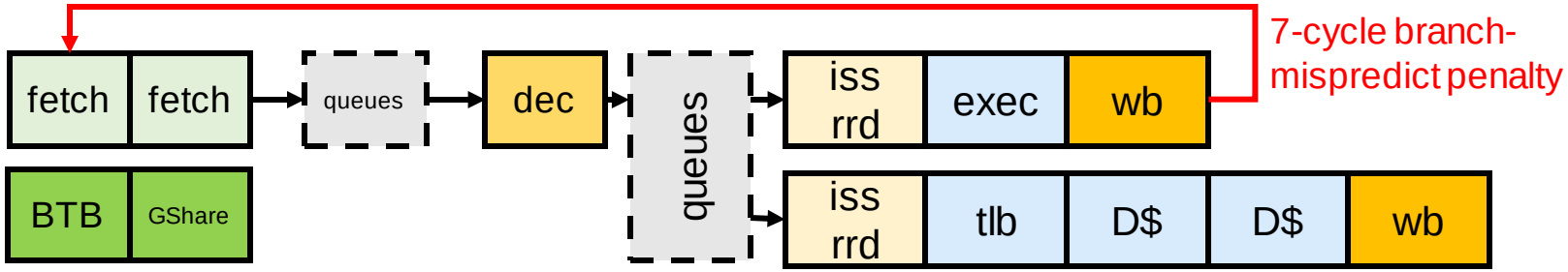
General-purpose performance is important across the entire computing ecosystem.

BOOM Goals:

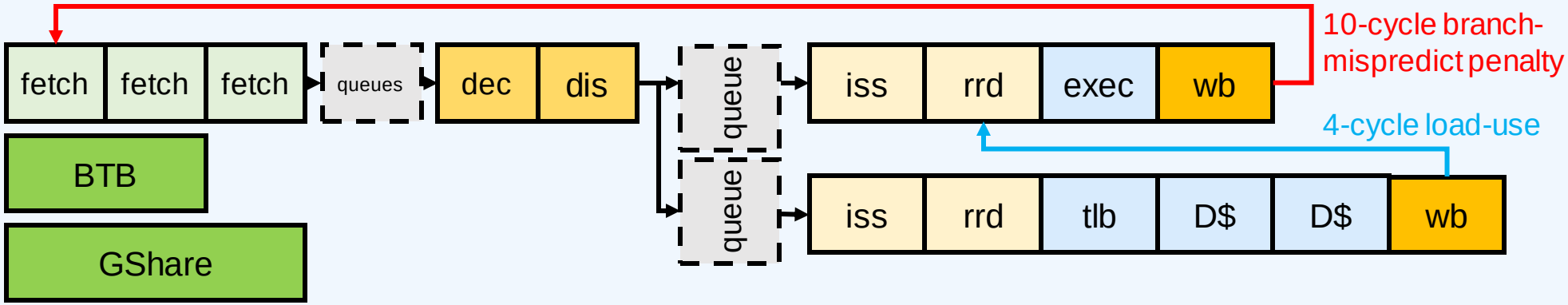
Build a high-performance open-source RISC-V out-of-order core

Support research in various aspects of high-performance SoC design (microarch, security, accelerators, etc.)

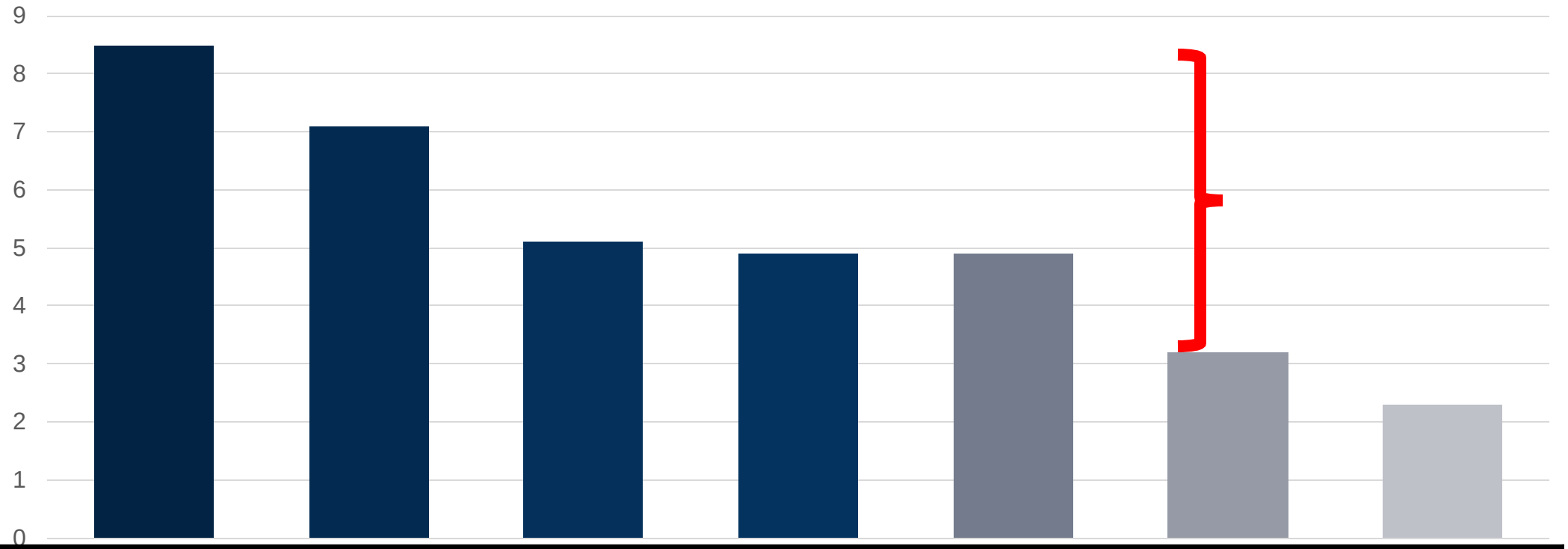
BOOMv1



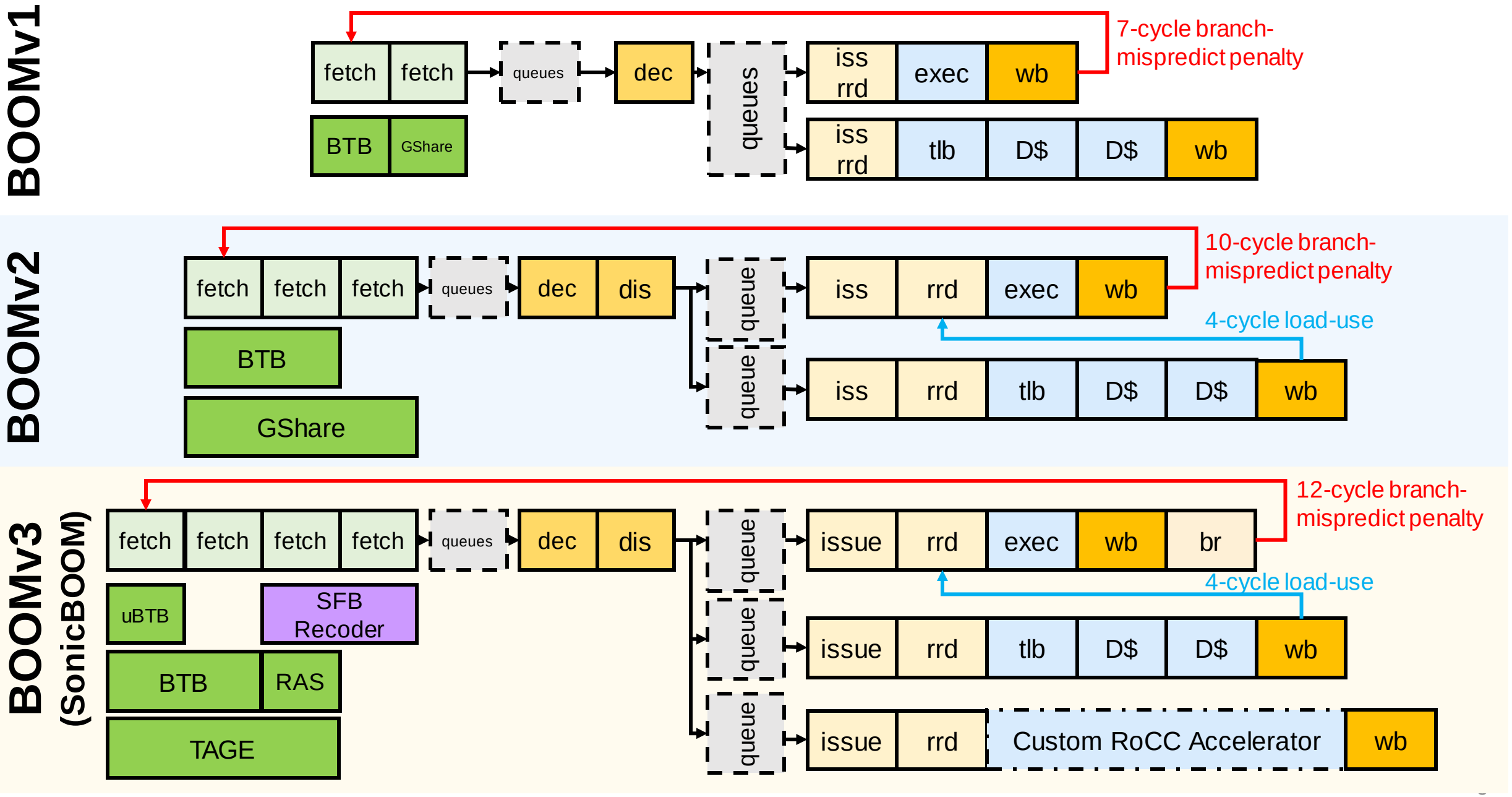
BOOMv2



Open-source Performance Gap



	Ivy Bridge	XuanTie 910	SiFive U74	WD SWERV	BOOMv1	BOOMv2	Rocket
Architecture	12+stage 4-w OOO	12-stage 3-w OOO	8-stage 2-w in-order	9-stage 2-w in-order	8-stage 4-w OOO	10-stage 4-w OOO	5-stage 1-w in-order
CoreMark/ MHz	8.5	7.1	5.1	4.9	4.9	3.2	2.3


BOOMv1
fetch
fetch
queues
dec
queues
iss
rrd
exec
wb
7-cycle branch-
mispredict penalty
BTB
GShare
iss
rrd
tlb
D$
D$
wb
BOOMv2
fetch
fetch
fetch
queues
dec
dis
queue
queue
iss
rrd
exec
wb
10-cycle branch-
mispredict penalty
4-cycle load-use
BTB
GShare
iss
rrd
tlb
D$
D$
wb
BOOMv3
(SonicBOOM)
fetch
fetch
fetch
fetch
queues
dec
dis
queue
queue
queue
issue
rrd
exec
wb
br
12-cycle branch-
mispredict penalty
4-cycle load-use
uBTB
SFB
Recoder
BTB
RAS
TAGE
issue
rrd
tlb
D$
D$
wb
issue
rrd
Custom RoCC Accelerator
wb

SonicBOOM

Frontend:

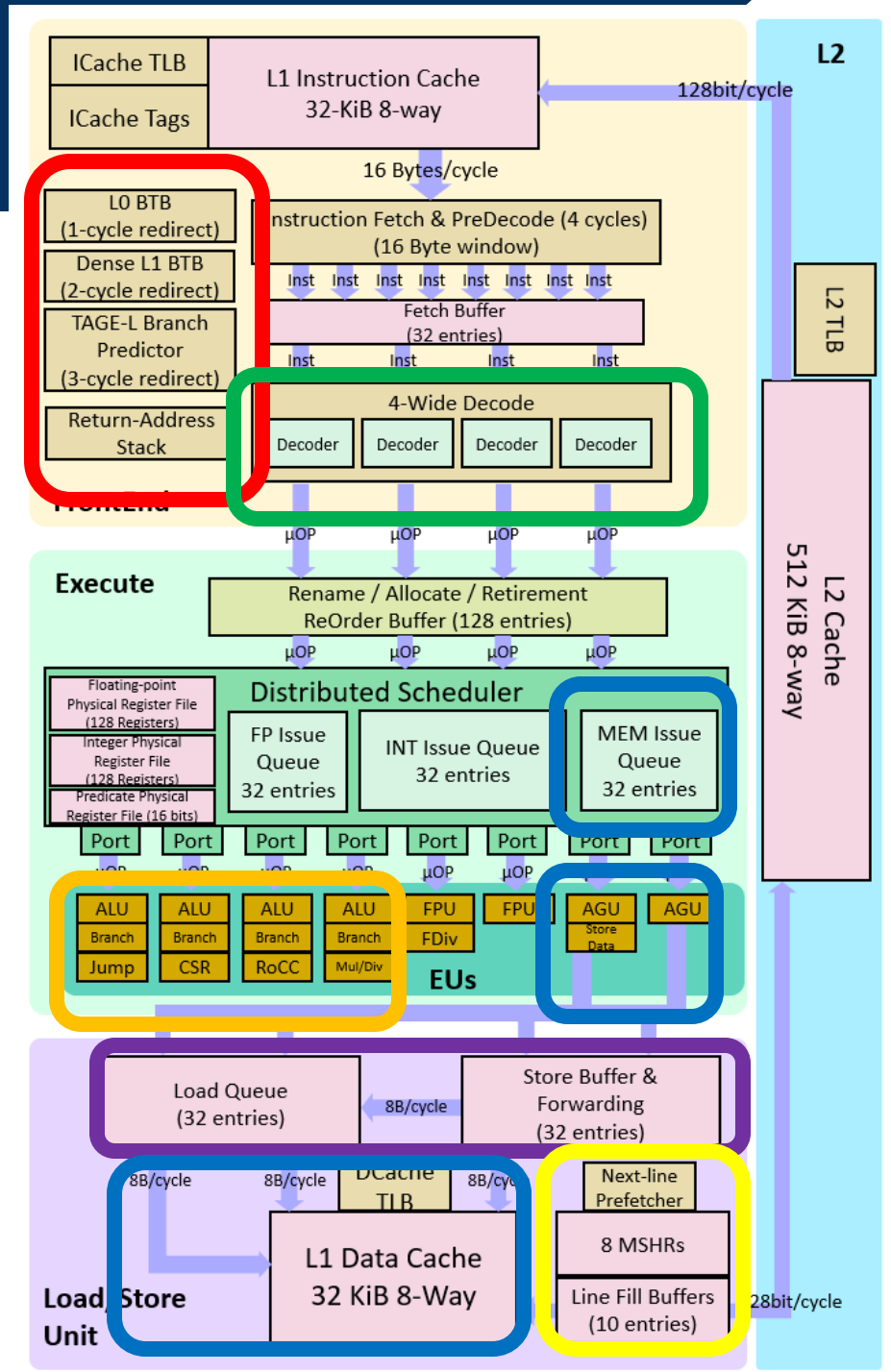
- New TAGE-L branch predictor
- New decoders for RISC-V compressed

Execute:

- Short-forwards-branch recoding
- Superscalar branch resolution
- Improved address-generation pipeline
- Custom RoCC accelerators

Memory:

- Superscalar address generation
- Superscalar load-store unit
- Optimized load/store scheduling
- L1 next-line-prefetcher w. line-fill-buffers



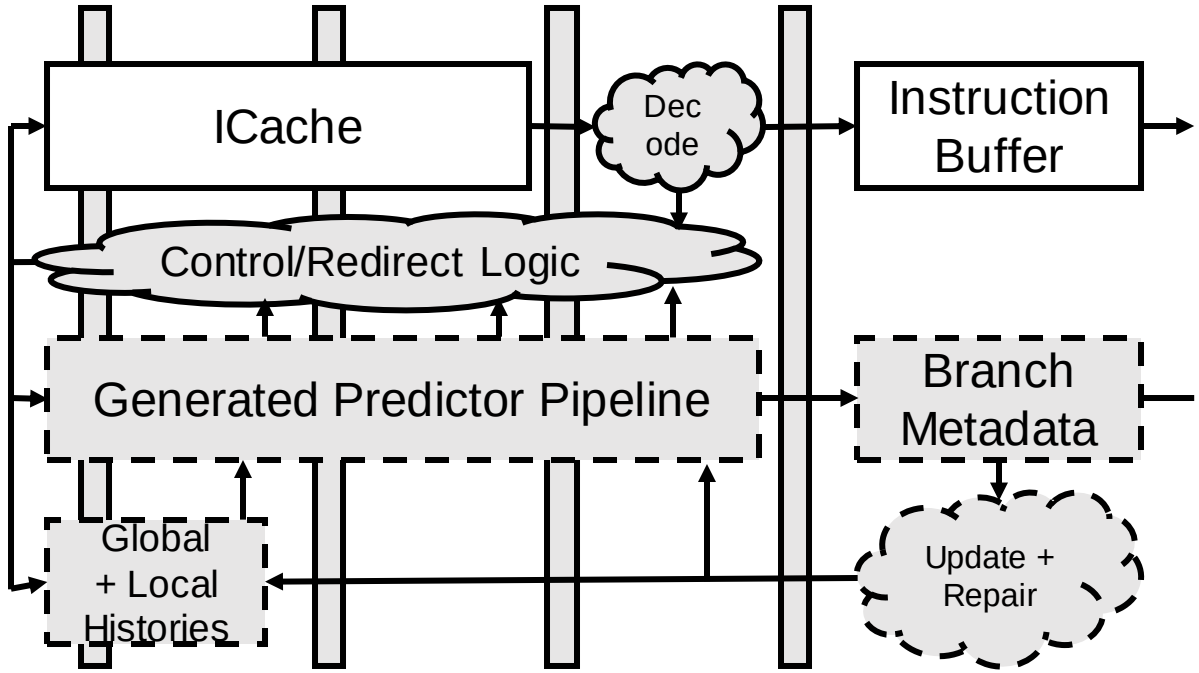
State-of-the-art Branch Prediction

Challenges:

- Superscalar fetch/predict
- Speculative updates
- Repair after misspeculation
- Predictor pipelining

SonicBOOM Instruction Fetch:

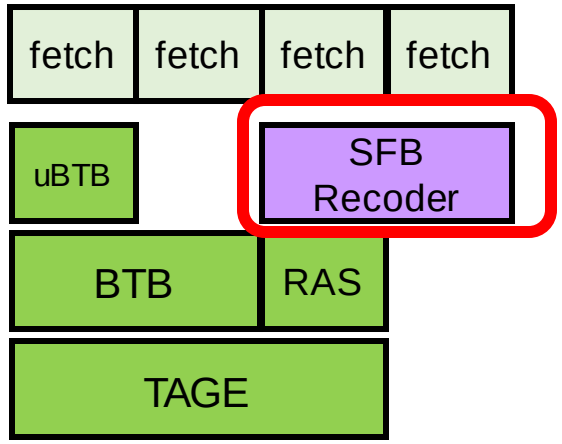
- Variable-width (RVC) decode
- L0/L1 BTBs
- Pipelined TAGE + Loop predictor
- Repaired return-address-stack



Improving Branch Performance

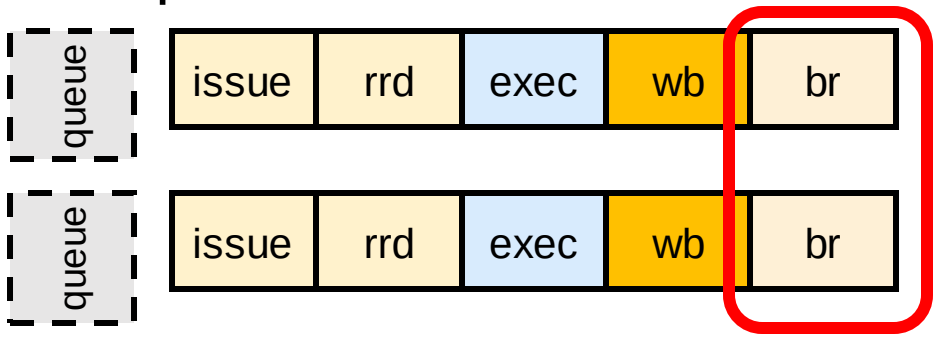
Dynamic Predication

- Recode short-forwards-branches into “predicated” micro-ops
- "POWER8"-style
- 5.1 CM/MHz -> 6.2 CM/MHz



Superscalar Branch Resolution

- BOOMv2: 1 branch/jump unit
- BOOMv3: Every ALU is a branch unit
  - Correct prediction is cheap, misprediction is expensive
  - Single JMP unit to handle AUIPC/JAL instructions
  - +1 branch latency to find oldest mispredicted branch



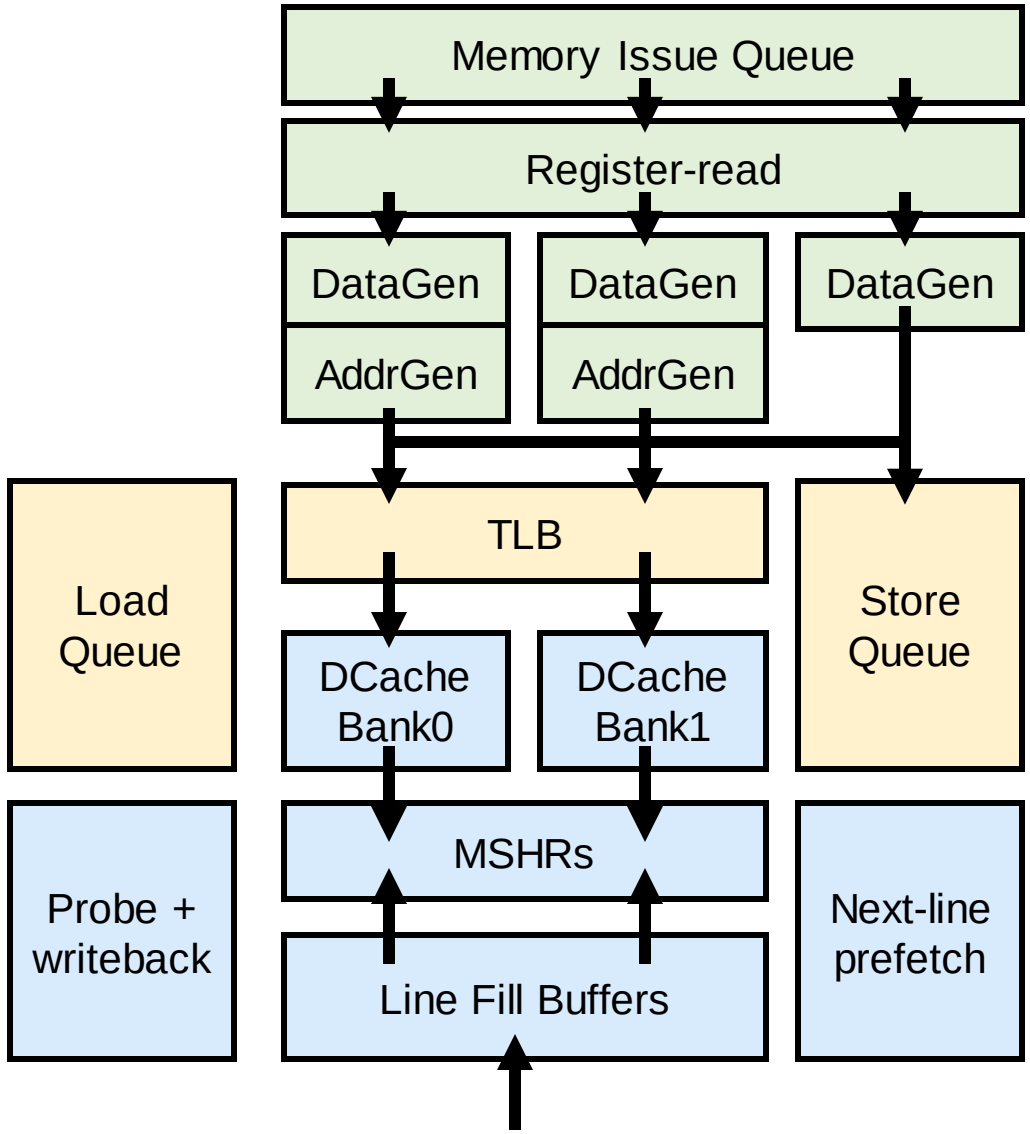
Advanced Load/Store Unit

Superscalar memory access:

- Addr-gen/translate/execute 2 loads per cycle
- Banked DCache data arrays

Improved L1 Data Cache:

- Fully non-blocking (refill in parallel with writeback)
- Line-fill-buffers with next-line-prefetcher
- Improved memory scheduler

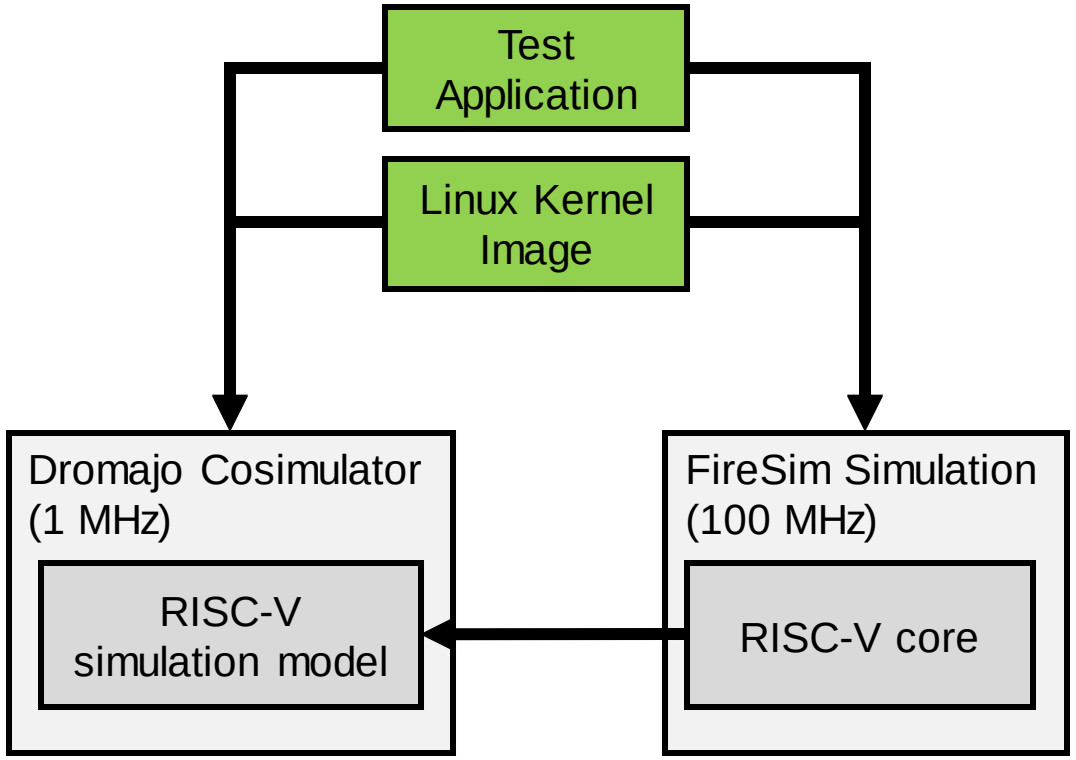


FPGA-accelerated Co-simulation

Dromajo: simulator developed by Esperanto, checks correctness of RISC-V trace

Fromajo: couple Dromajo to FireSim FPGA simulation of core

- Committed instruction stream pulled from core
- Committed instructions checked against Dromajo at 1 MHz
- Cycle-exact, reproducible divergences
- Works with other RISC-V cores (Ex: Ariane)



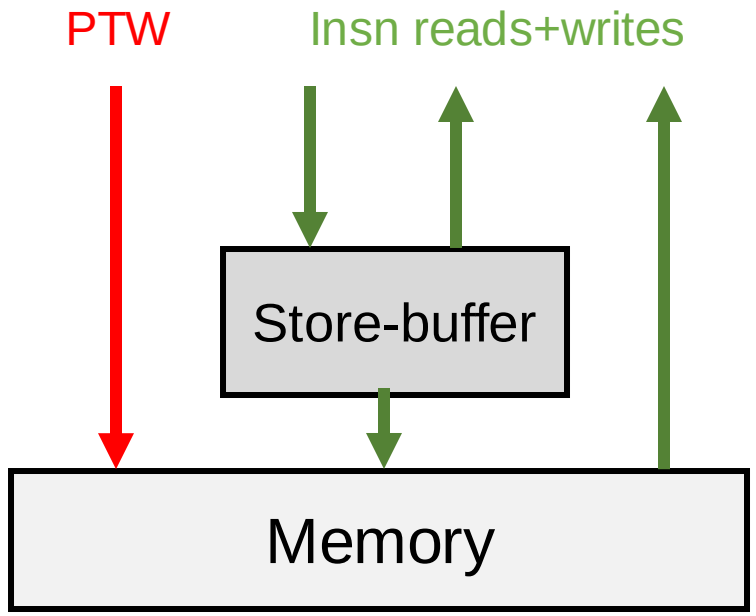
Finding a RISC-V Linux Bug

Background:

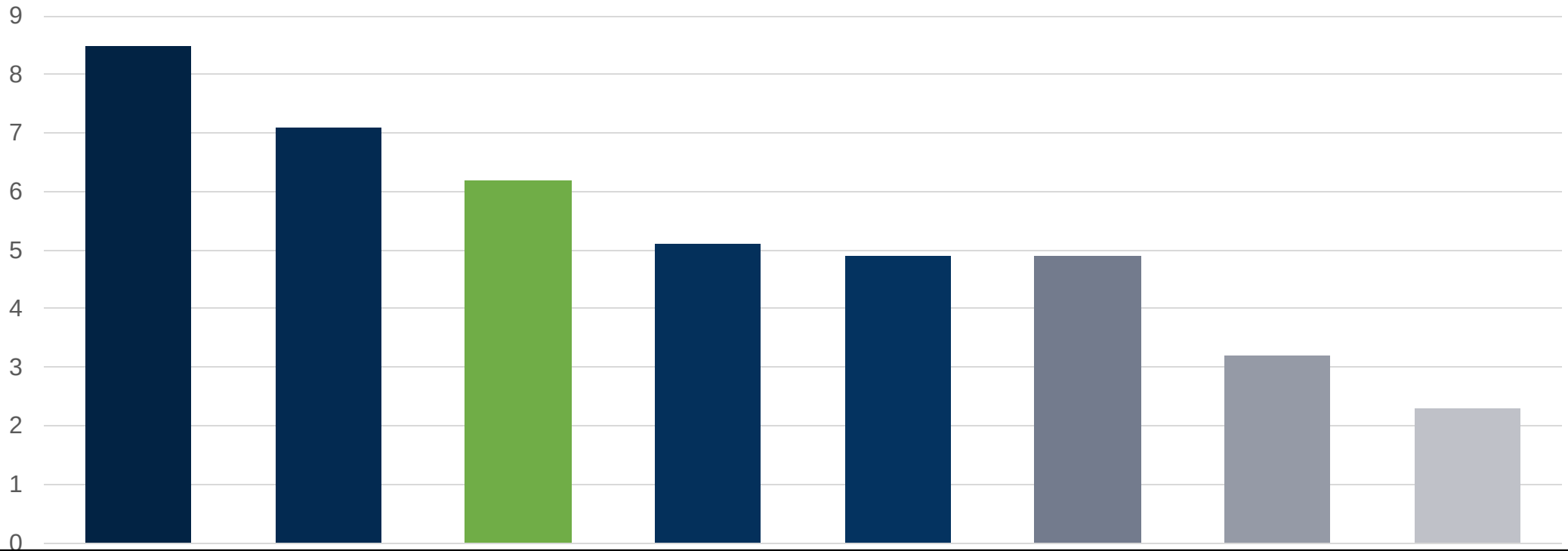
- PTWs are unordered w.r.t. loads/stores
- SFENCE.VMA orders page-table updates with accesses

Found Linux hang with SonicBOOM

- Kernel load launches a PTW to recently written PTE
- No SFENCE between PTE write and PTW
- Only materializes on a deeply speculating core
- Patch in-progress



CoreMark IPC



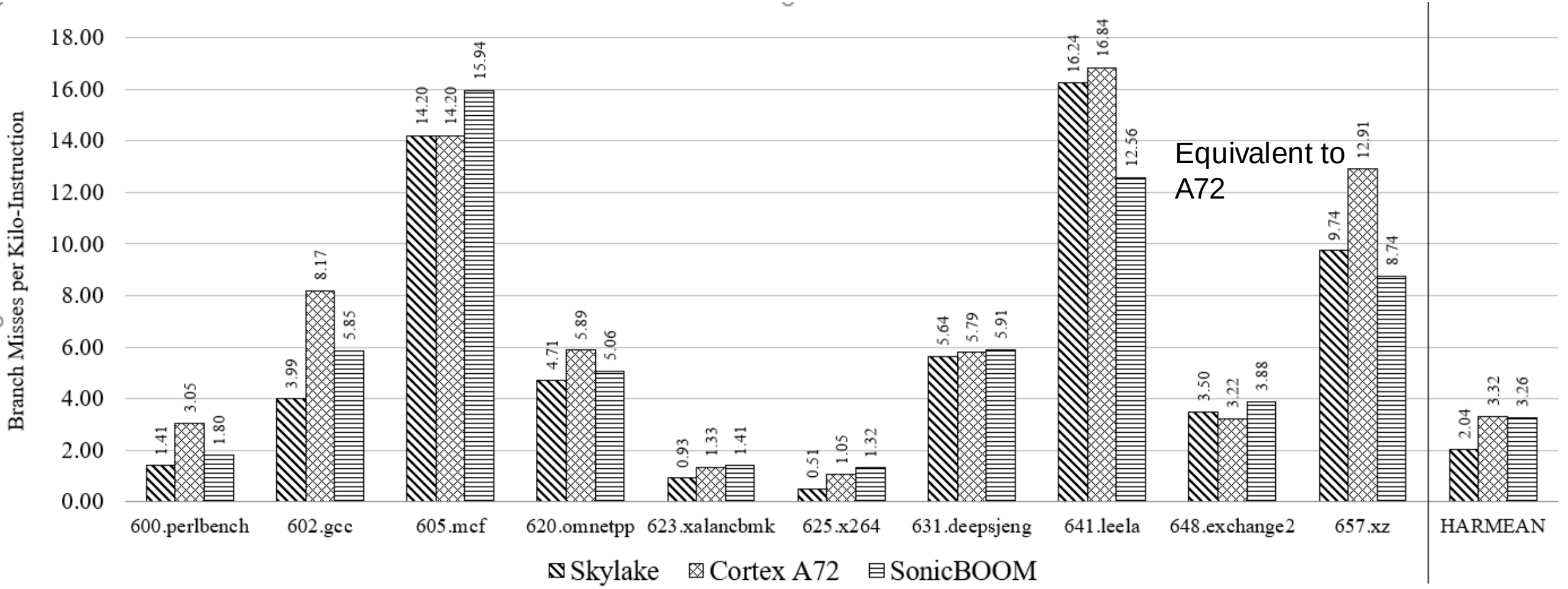
	Ivy Bridge	XuanTie 910	BOOMv3	SiFive U74	WD SWERV	BOOMv1	BOOMv2	Rocket
Architecture	12+stage 4-w OOO	12-stage 3-w OOO	12-stage 4-w OOO	8-stage 2-w in-order	9-stage 2-w in-order	8-stage 4-w OOO	10-stage 4-w OOO	5-stage 1-w in-order
CoreMark/ MHz	8.5	7.1	6.2	5.1	4.9	4.9	3.2	2.3

SPEC17 Comparison

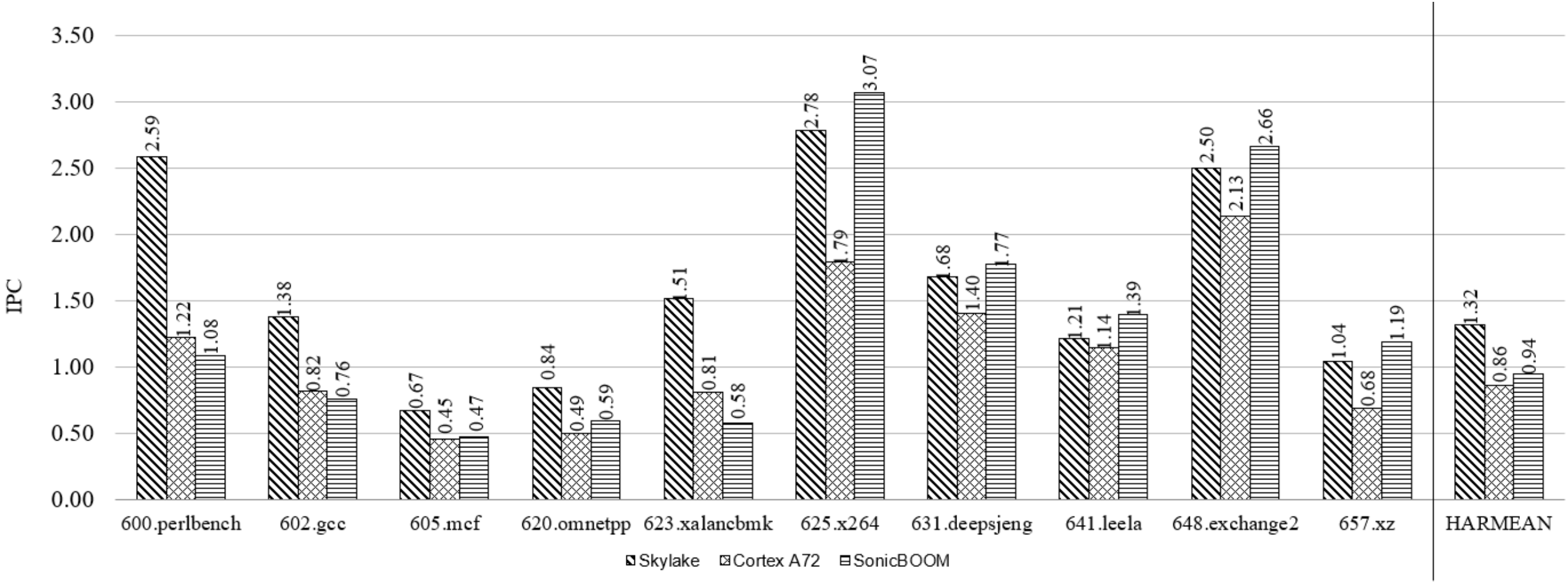
- Evaluate SPEC17 intspeed, single-core performance
- Target comparable branch-prediction accuracy and IPC

	Intel Xeon	AWS Graviton	SonicBOOM
Microarchitecture	Skylake Server	Cortex A72	BOOMv3
Branch Predictor	Undisclosed	Undisclosed	TAGE-L
L1 Cache Sizes (I/D)	64/64 KB	48/32 KB	32/32 KB
L2 Cache Size	1 MB	2 MB	512 KB
L3 Cache Size	24 MB	0 MB	4 MB
Compiler	gcc	gcc	gcc
OS	Ubuntu 18.04 Server	Ubuntu 18.04	Buildroot Linux
Platform	AWS EC2 bare-metal	AWS EC2 bare-metal	FireSim simulation

SPEC17 Branch Prediction Accuracy



SPEC17 IPC


IPC
3.50
3.00
2.50
2.00
1.50
1.00
0.50
0.00
2.59
1.22
1.08
1.38
0.82
0.76
0.67
0.45
0.47
0.84
0.49
0.59
1.51
0.81
0.58
2.78
1.79
3.07
1.68
1.40
1.77
1.21
1.14
1.39
2.50
2.13
2.66
1.04
0.68
1.19
1.32
0.86
0.94
600.perlbench
602.gcc
605.mcf
620.omnetpp
623.xalancbmk
625.x264
631.deepsjeng
641.leela
648.exchange2
657.xz
HARMEAN
Skylake
Cortex A72
SonicBOOM

Next steps

Physical Implementation:

- > 1 GHz possible according to preliminary results
- Critical path in issue-units (issue-select/compaction)
- Current SRAMs limit us to 1.4 GHz

Improving performance:

- Larger prefetchers between L2/LLC to hide L2 miss penalty
- Instruction prefetcher
- V-Extension support