中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

中国开放指令生态（RISC-V）联盟
China RISC-V Alliance

鹏城实验室

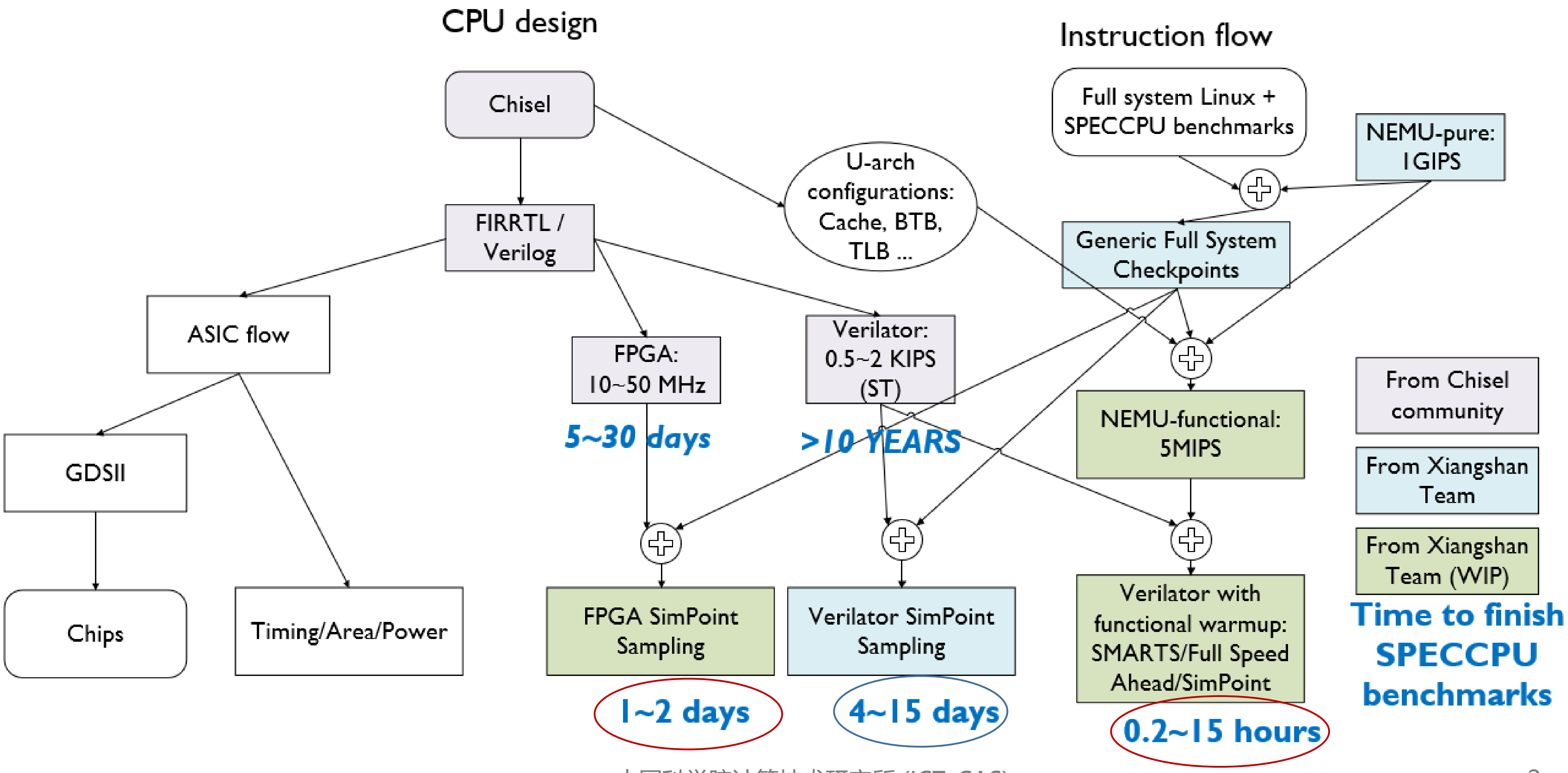
BetaPoint:

A pre-silicon performance evaluation framework

周耀阳	Zhou Yaoyang
中科院计算所	ICT, CAS

2021年6月19日	June 19, 2021

■ Agile performance evaluation roadmap



CPU design
Instruction flow
Chisel
FIRRTL / Verilog
U-arch configurations: Cache, BTB, TLB …
Full system Linux + SPECCPU benchmarks
NEMU-pure: 1GIPS
Generic Full System Checkpoints
ASIC flow
FPGA: 10~50 MHz
Verilator: 0.5~2 KIPS (ST)
NEMU-functional: 5MIPS
GDSII
5~30 days
>10 YEARS
Chips
Timing/Area/Power
FPGA SimPoint Sampling
Verilator SimPoint Sampling
Verilator with functional warmup: SMARTS/Full Speed Ahead/SimPoint
1~2 days
4~15 days
0.2~15 hours
From Chisel community
From Xiangshan Team
From Xiangshan Team (WIP)
Time to finish SPECCPU benchmarks

■ Agenda

- Background and motivation
  - Emerging RTL cores/SoCs
  - Lack of checkpoint and sampling support
- Current infrastructures
  - Cross-platform checkpoint format
- Ongoing works
  - Functional warmup

■ Opportunities and challenges

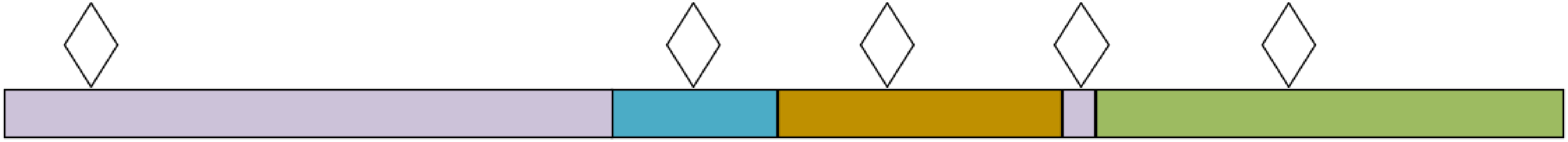
- Emerging open source cores/SoCs in RTL (Chisel, Verilog)
  - Enable agile prototyping
  - Enable researchers to produce solid results on performance, area, and timing
- RTL emulation software is slow: 7+ years to complete SPEC2017.*imagick*
- With FPGA, it is still slow: 5~30 days to finish the SPECCPU 2006/2017 benchmarks with one FPGA.
  - 5~30 days is still too long for performance iteration
  - Cloud FPGA cannot accommodate large cores (we use vu19p for single-core Xiangshan)
  - More FPGAs → more costs

→ Can we enable sampling?

Sampling methods

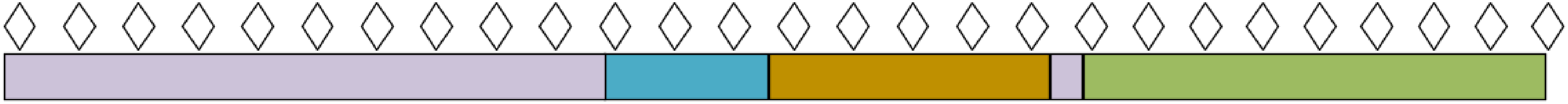
Checkpoint-based sampling (SimPoint):

- Selective weighted sampling
- Large simulation points (50M~200M)



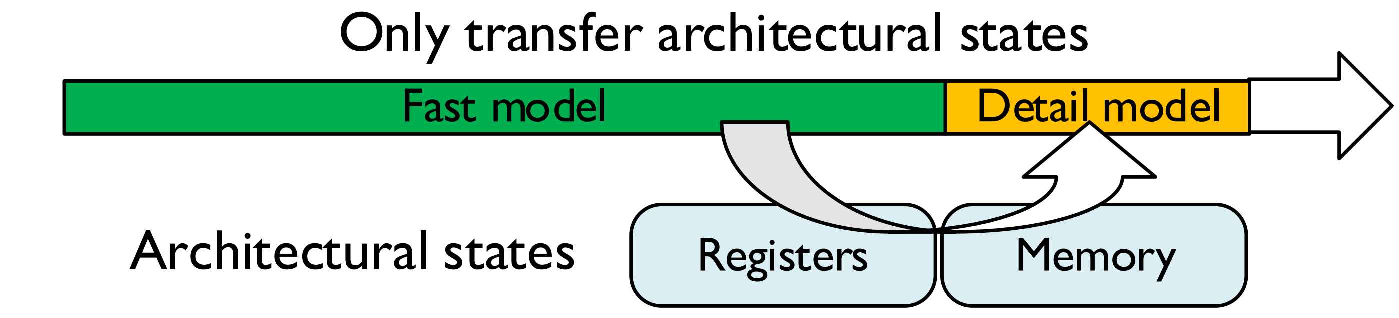
Fast-forwarding-based sampling (SMARTS):

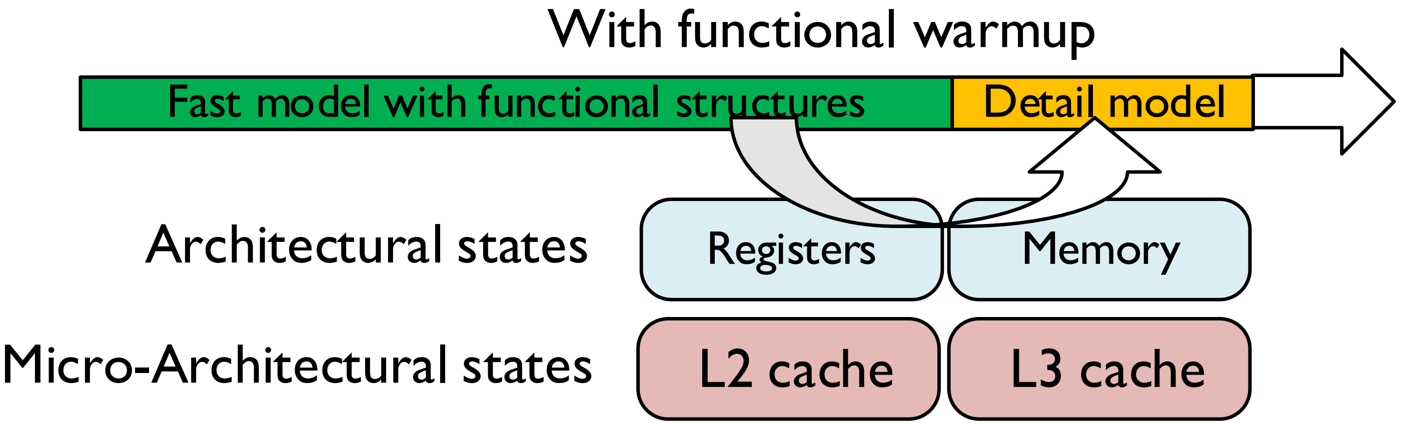
- Uniform sampling
- Smaller simulation points (5k~50k)



■ Sampling methods in research community

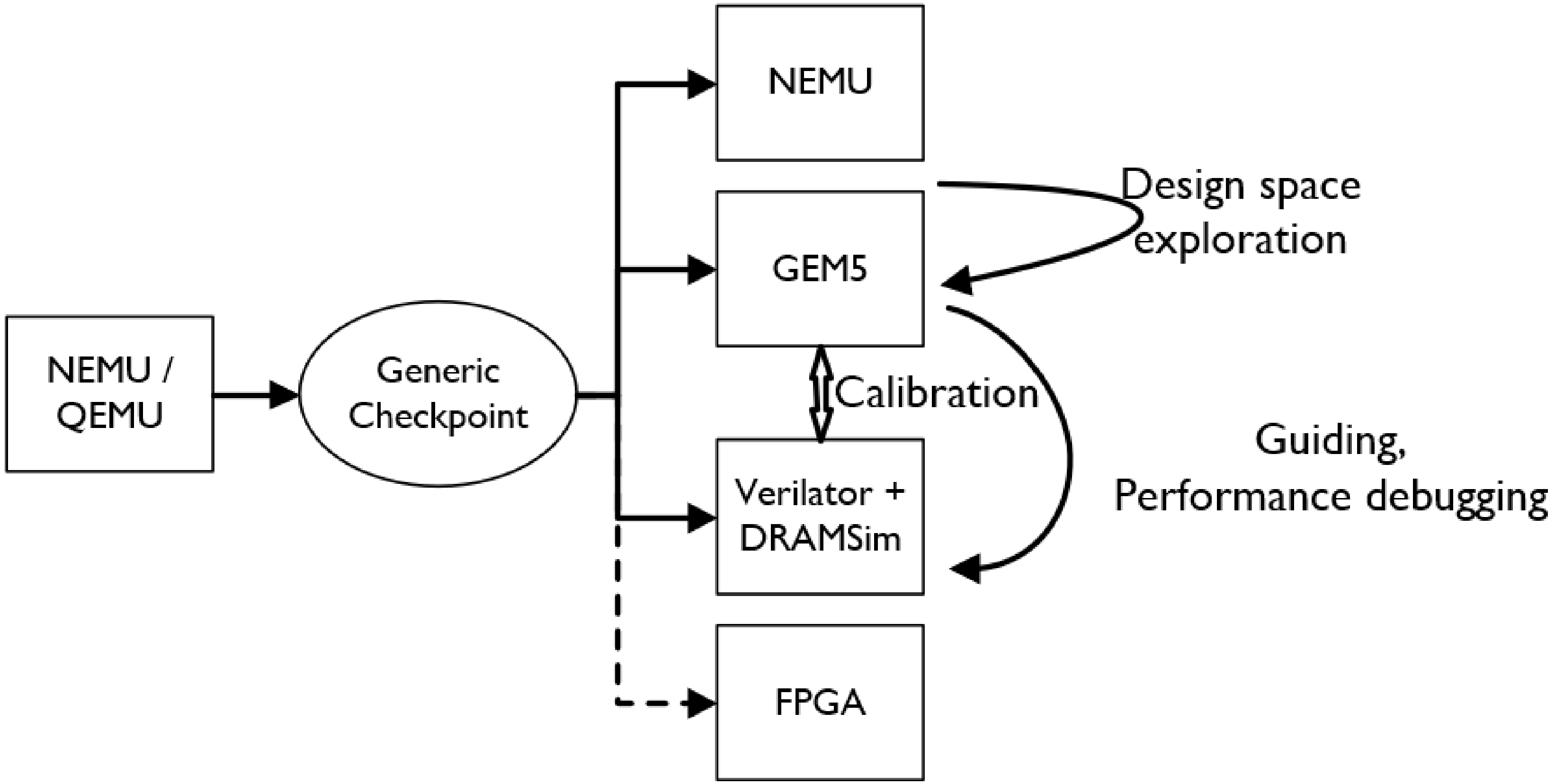
- Checkpoints: SimPoint
  - Requires **checkpoint support**
- Fast-forwarding: SMARTS
  - Requires **functional warmup** to speed up
- Virtualized fast-forwarding: CoolSim, DELOREAN
  - Faster than SMARTS
  - Requires (statistical) **functional warmup** to speed up





Our checkpoint format

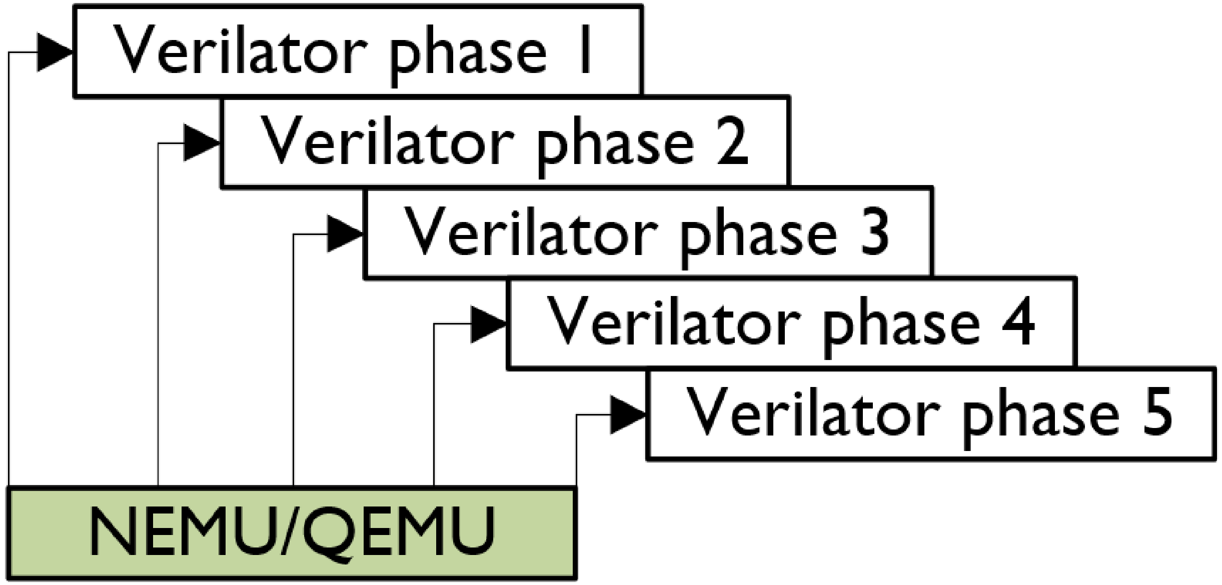
- A generic checkpoint format (GCPT) bridges fragmented parts



1. NEMU is a functional simulator developed by Yu Zihao, which runs at 300 MIPS~1GIPS.
2. GCPT is theoretically compatible with FPGA. But we have not test it on FPGA.

Current applications of NEMU+GCPT - Shotgun

- Shotgun method (The name is inspired by DNA sequencing)
  - Break the whole program into chunks
  - Generate checkpoints with fast models (NEMU)
  - Run detail models **in parallel**

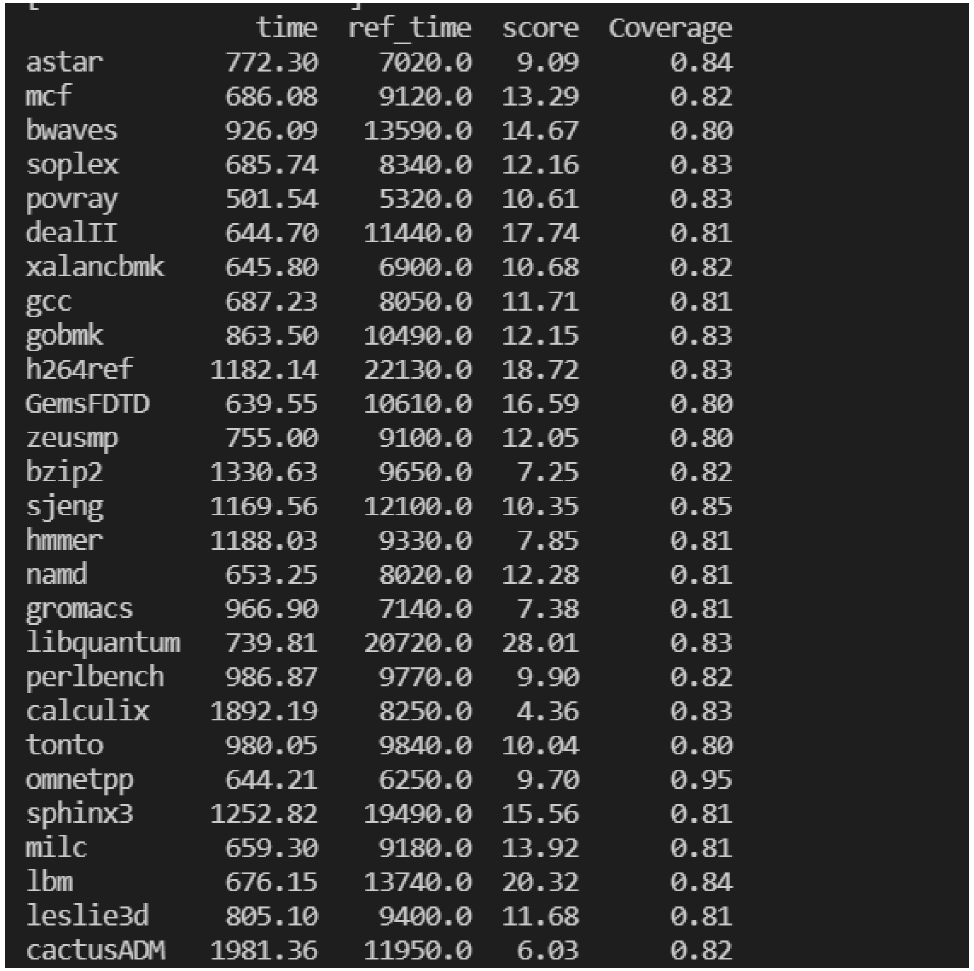


- Reduce the time to gather "ground truth performance"
  - Fake ground truth: limited warmup length

Current applications of NEMU+GCPT - SimPoint on Verilator

- Performance estimation using SimPoint for Xiangshan core

- 3~14 days to finish a 100M simulation point with 1 core
  - 14 days for *mcf* because very low IPC (0.1x)



```
                  time  ref_time  score  Coverage
astar           772.30    7020.0   9.09      0.84
mcf             686.08    9120.0  13.29      0.82
bwaves          926.09   13590.0  14.67      0.80
soplex          685.74    8340.0  12.16      0.83
povray          501.54    5320.0  10.61      0.83
dealII          644.70   11440.0  17.74      0.81
xalancbmk       645.80    6900.0  10.68      0.82
gcc             687.23    8050.0  11.71      0.81
gobmk           863.50   10490.0  12.15      0.83
h264ref        1182.14   22130.0  18.72      0.83
GemsFDTD        639.55   10610.0  16.59      0.80
zeusmp          755.00    9100.0  12.05      0.80
bzip2          1330.63    9650.0   7.25      0.82
sjeng          1169.56   12100.0  10.35      0.85
hmmer          1188.03    9330.0   7.85      0.81
namd            653.25    8020.0  12.28      0.81
gromacs         966.90    7140.0   7.38      0.81
libquantum      739.81   20720.0  28.01      0.83
perlbench       986.87    9770.0   9.90      0.82
calculix       1892.19    8250.0   4.36      0.83
tonto           980.05    9840.0  10.04      0.80
omnetpp         644.21    6250.0   9.70      0.95
sphinx3        1252.82   19490.0  15.56      0.81
milc            659.30    9180.0  13.92      0.81
lbm             676.15   13740.0  20.32      0.84
leslie3d        805.10    9400.0  11.68      0.81
cactusADM      1981.36   11950.0   6.03      0.82
```

■ Contradiction on warmup length

- Current warmup length (50M) is too short for accuracy
  - Some applications need more than 1G warmup
  - Discussed in BLRL (The Computer Journal, 2005); Elfies (CGO, 2021)

- Current warmup length is too long for speed
  - Two weeks to simulate 100 M instructions of mcf

→ For both accuracy and speed, we must speed up warmup

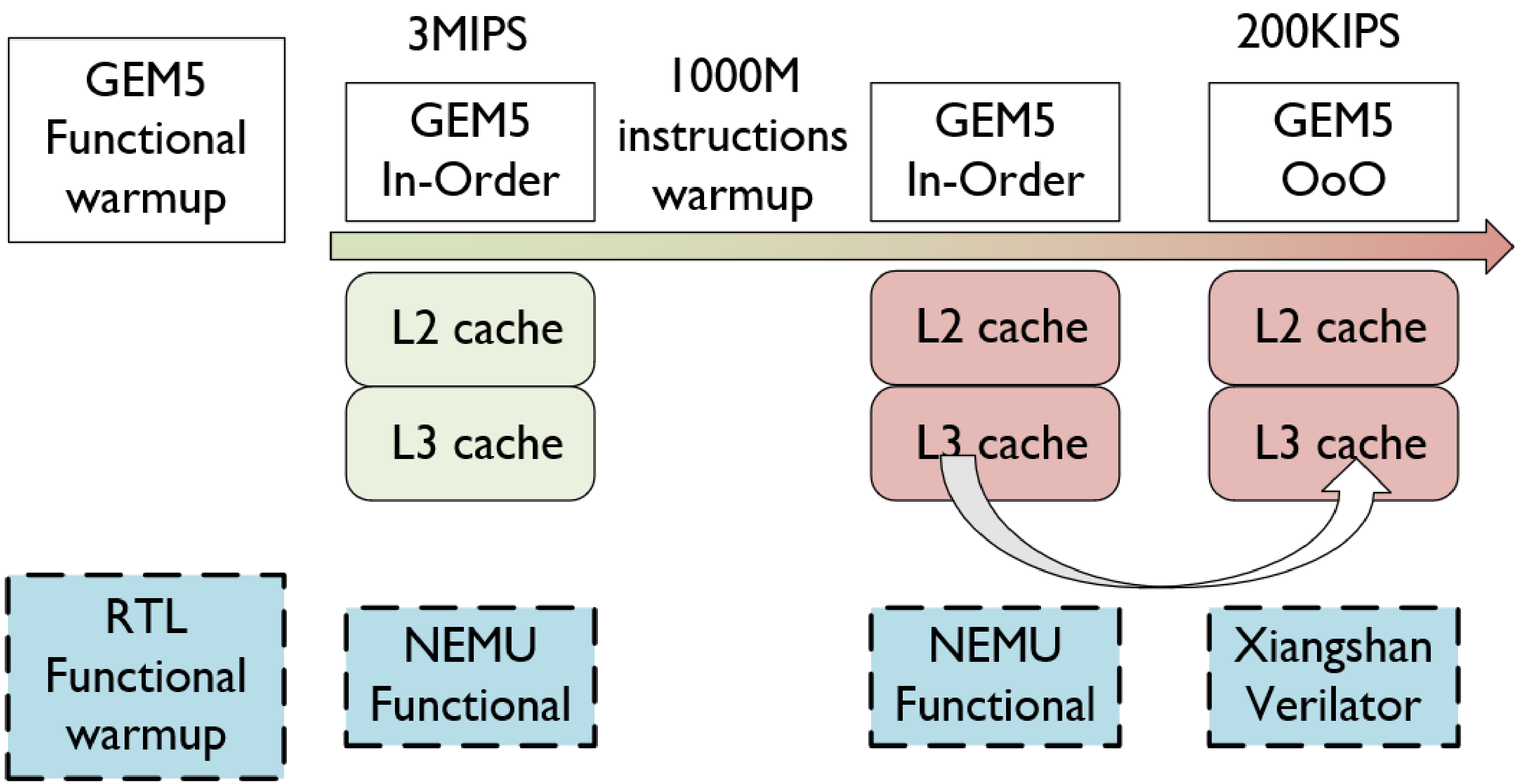
→ Functional warmup

Ongoing work:

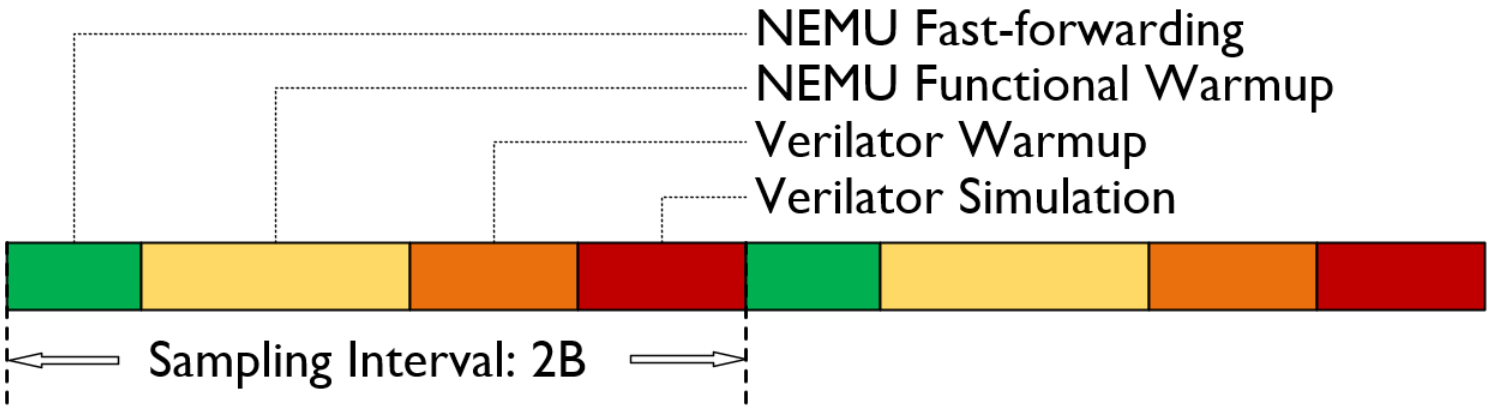
Functional warmup for RTL emulator

■ Functional warmup

- Warmup caches on fast model → Switch core to detailed model



Estimated speed of Verilator-FSA



	Assumed Speed	# instructions	Estimated Time
NEMU Fast-forwarding	1000 MIPS	1000M	1.0s
NEMU Functional Warmup	5.0 MIPS	1000M	200.0s
Verilator	0.5 KIPS	50K	100.0s
Total	6.64 MIPS	2000M + 50K	301.0s

If we achieve 6.64 MIPS, we can finish SPECCPU in 16.73h with 100 cores
1 day (**12K $**) VS. FPGA **5 days**@100MHz (**62K $**)

Current progress

Generating memory accesses with NEMU

  - Instruction fetch addresses
  - Load/Store addresses

- Functionally warm up caches @ ~16 MIPS
  - Arguably **the fastest functional warmup method** yet
  - Booting Linux in **17s**

Comparing against existing solutions:

- GEM5 Atomic with no caches and with simple memory: < 4MIPS
- Sniper: ~2 MIPS

■ Next challenge: bridging C++ simulator and RTL

How to transfer u-arch states between C++ and RTL Cleanly

- Chisel BlackBox / RAM initiation?
- Verilog DPI-C?

Open problem

■ Towards *Asset-Light* performance evaluation

Time to evaluate/estimate SPECCPU score:

	Without sampling		With sampling	
Devices	vu19p @ 100 MHz	Palladium Z1 @ 20 MHz	4x EPYC servers	**1x EPYC servers**
Methods	-	-	SimPoint + Checkpoint	Full Speed Ahead + Functional warmup
Pricing	~62K$	>1550K$	~50K$	~**12k$**
Time	5 days	25 days	14 days	< **1 day** (WIP)
Scalability	At most 2-core Xiangshan	Very large SoC	Very large SoC	Very large SoC

Thank you！

I am interested in

- Micro-architecture exploration
- Performance modeling and evaluation
- *Asset-Light* chip development

(expected to graduate in 2023)

My homepage:

https://archshine.xyz/