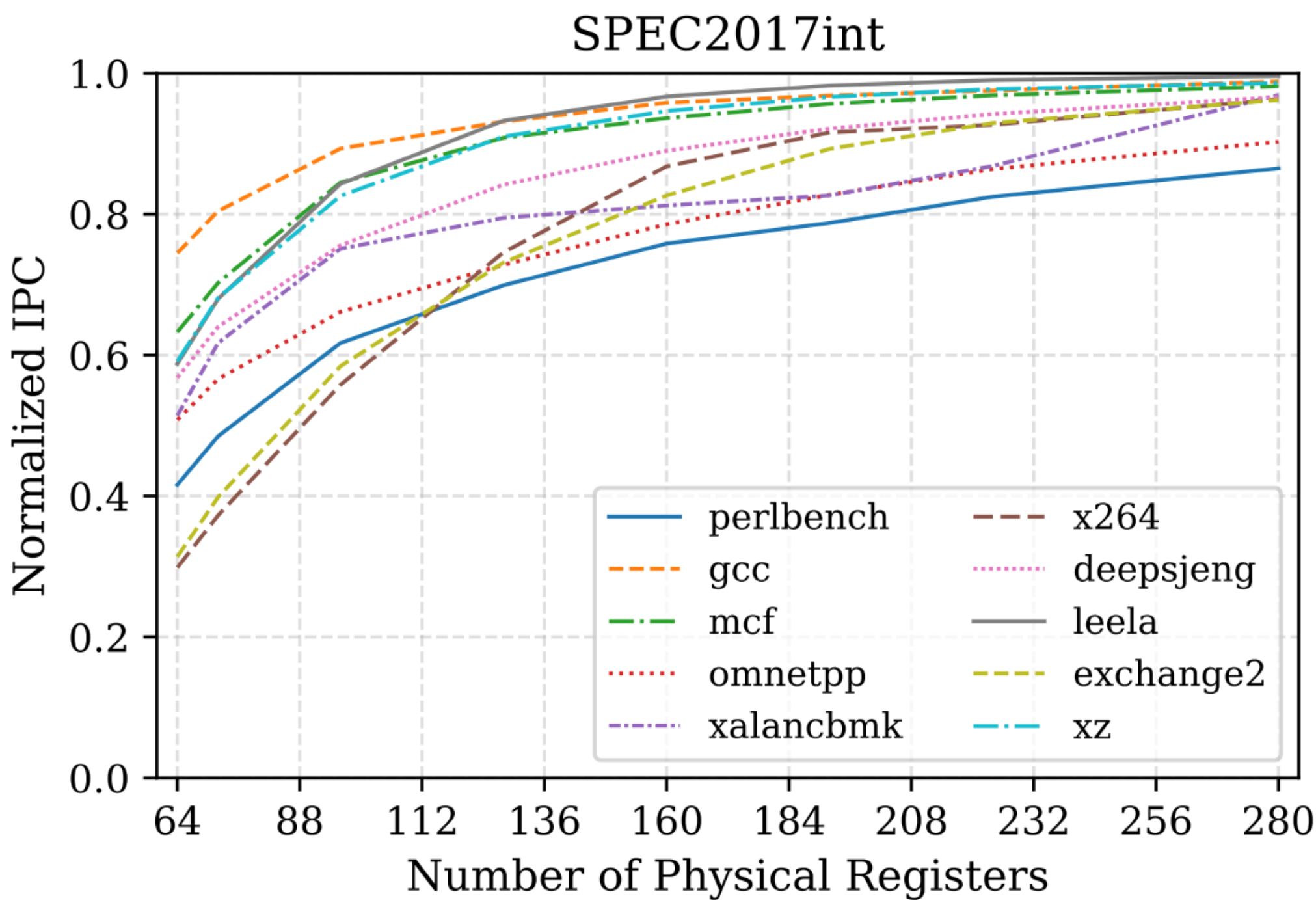


ATR: Out-of-Order Register Release Exploiting Atomic Regions

Yinyuan Zhao, Surim Oh, Mingsheng Xu, and Heiner Litz
University of California, Santa Cruz

Motivation

1. A Larger register file is crucial for supporting high numbers of in-flight instructions to sustain high IPC.
2. Scaling RF size introduces challenges in area, power and clock frequency.



10-wide u-Arch
• 20 read RF ports
• 10 write RF ports

512-entry RF
• 512:1 MUXs
• 64-bit flip-flops

2M transistors
 $1024 \times 64 \times 20$
 $+ 512 \times 64 \times 20$
1.3M + 650K

Background

Conventional Release

- I3 Committed

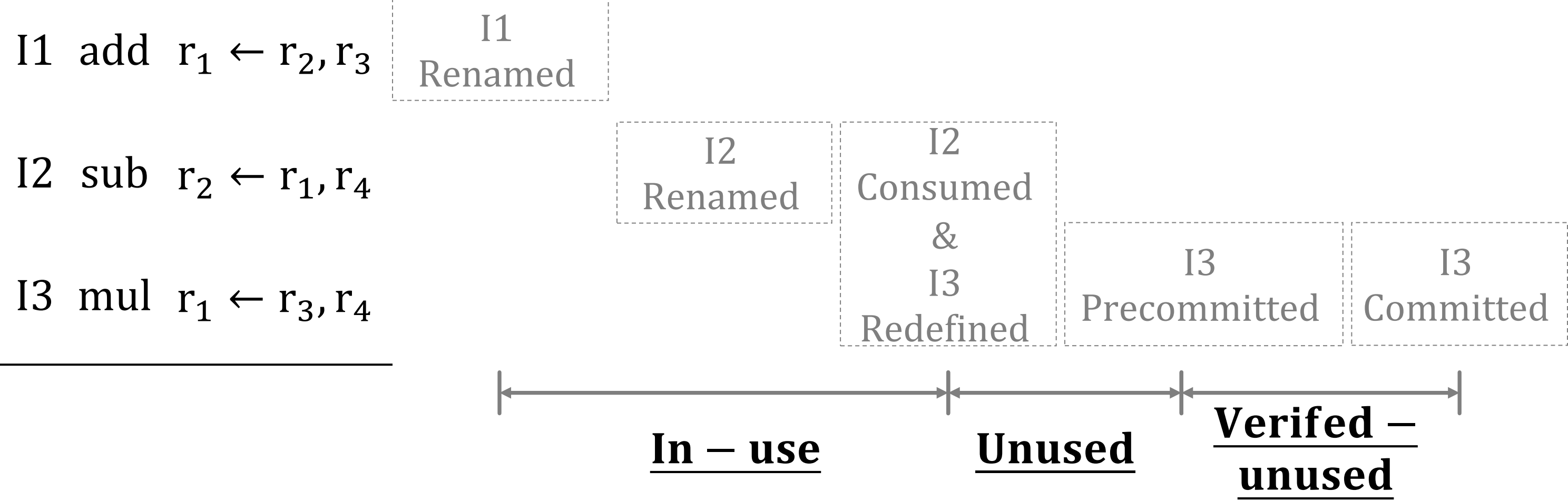
Speculative Early Release

- I2 Consumed + I3 Redefined

Non-speculative Early Release

- I2 Consumed + I3 Precommitted*

*BR/JMP resolved; ADDR generated; DIV completed.



Analysis

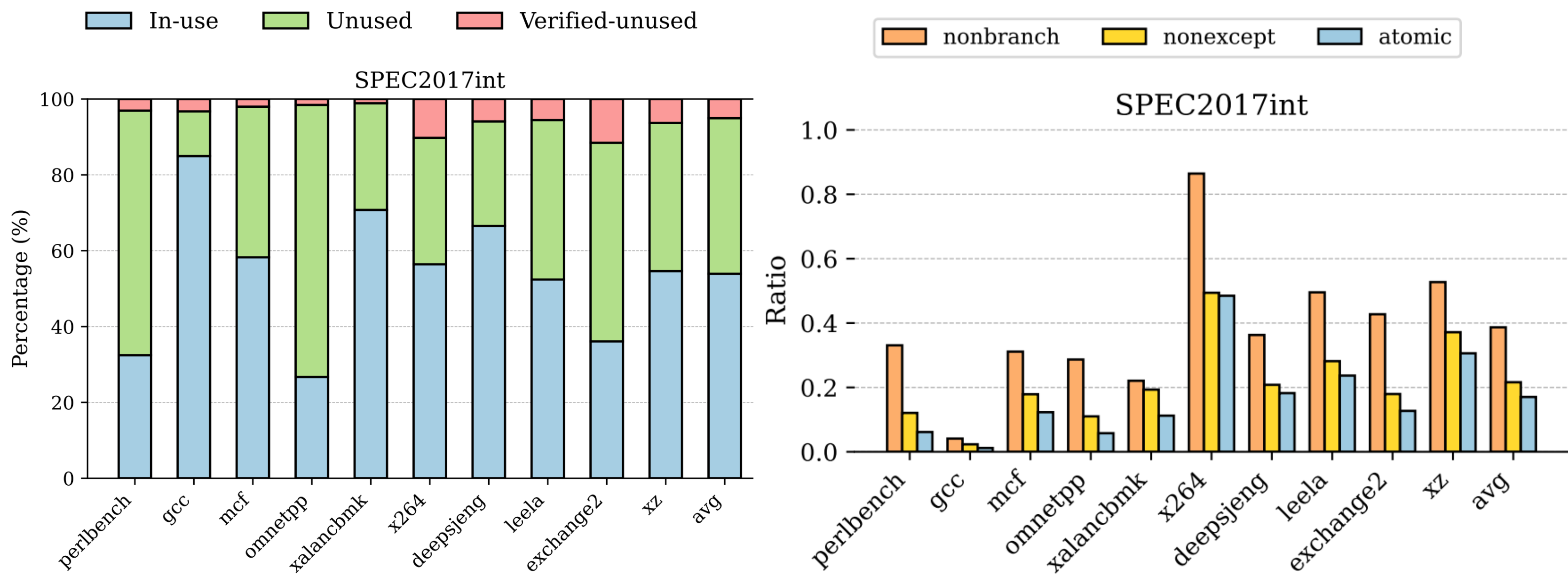
Conventional Release

- Registers are held much longer than needed.

Speculative Early Release

- A mis-predicted branch of I2 may flush I3.

		Re	Ex	Cm	Pr
I1 MOVE	RAX ← RAX	510	675	839	675
I2 TEST + JNZ	ZPS ← RAX	510	841	841	841
...	...				
I3 LEA	RAX ← RDI	709	716	716	841
...	...				
I4 LEA	RBX ← RAX	729	737	737	842
I5 SHR	RBX ← RBX, ZPS	729	738	738	842



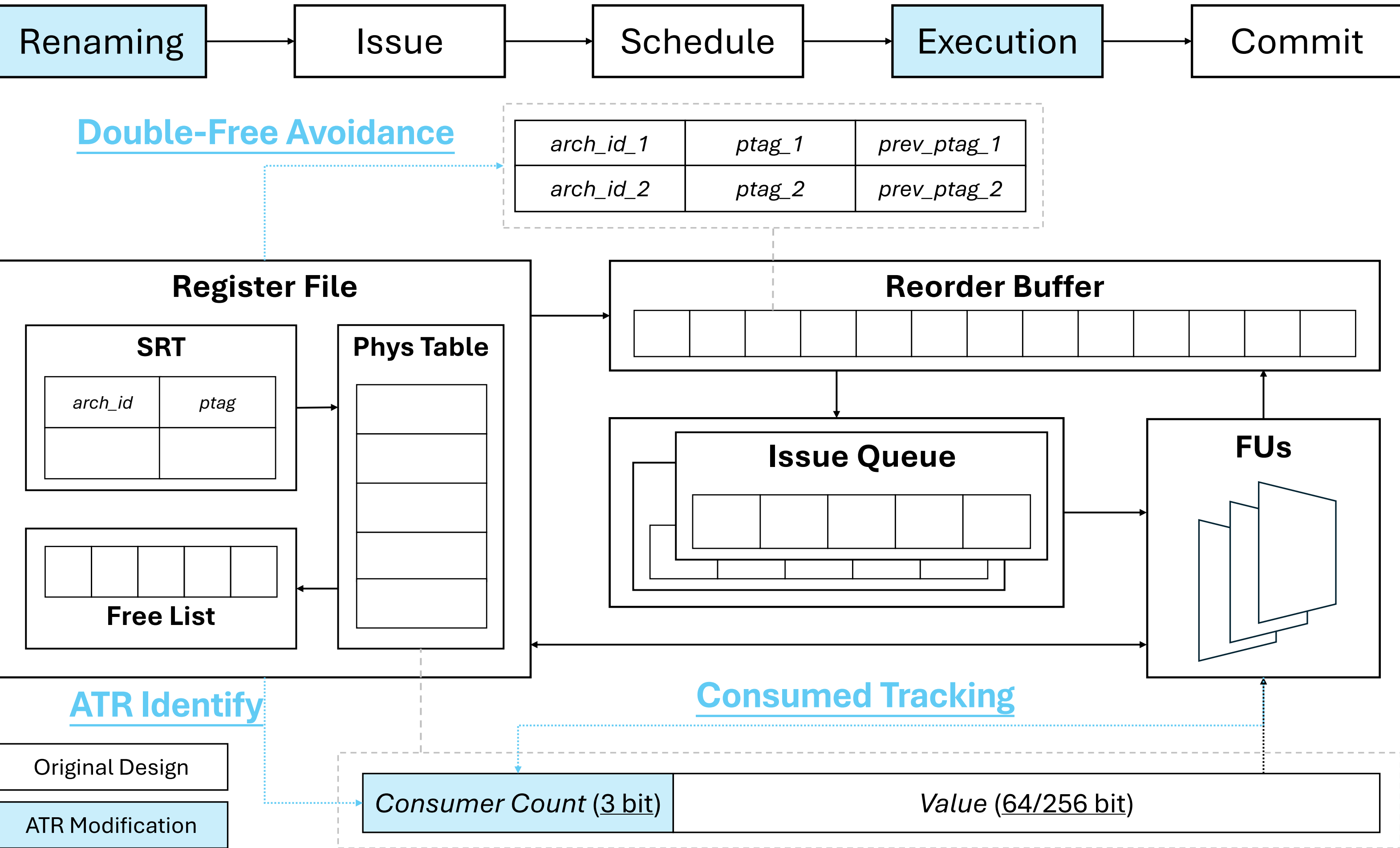
Non-speculative Early Release

- Big gap between unused and verified-unused.
- Lots of cycles are wasted in waiting for precommitting.

Atomic Release

- Freeing the physical register of RBX is safe, even if I2 mis-predicts and flushes all younger instructions.
- Over 17% of all registers are located with atomic regions.

Design



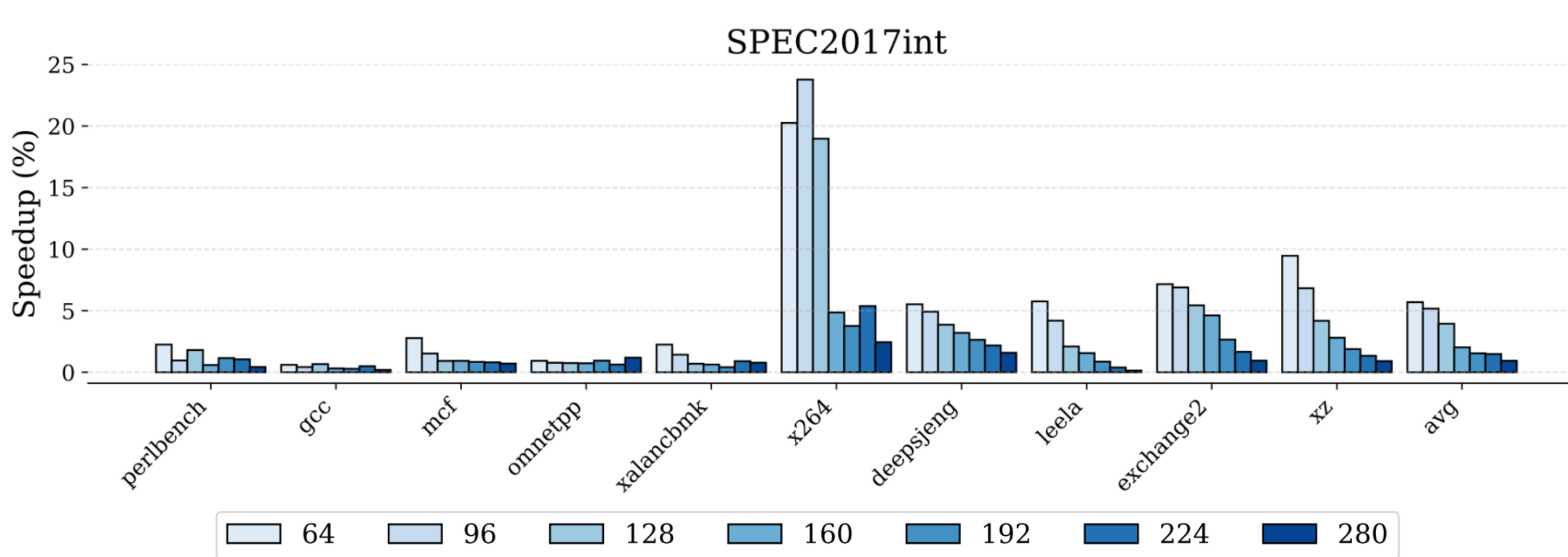
Atomic Region Detection

- Each ptag maintains a 3-bit count.
- Set as NO-EARLY-RELEASE when a br/jmp/ld/st/div is renamed.

Atomic Early Release

- The consumer count is zero. → Atomic + Consumed
- The ptag has been redefined.

Evaluation



vs Conv@64: IPC +5.7%

vs Conv@280: Power -5.5%

Conclusion

- Atomic regions: instruction sequences without br/jmp/ld/st/div.
- Registers allocated within atomic regions can be safely released out of precommit-order (either commit or be flushed together).
- Require no stacks, queues, extra memory, or shadow cells but maintain precise exceptions.