

Cache simulation comparison on RISC-V

QEMU vs gem5 caches

Joachim Le Fournis

2025-07-29

Internship supervisor: Julie Dumas

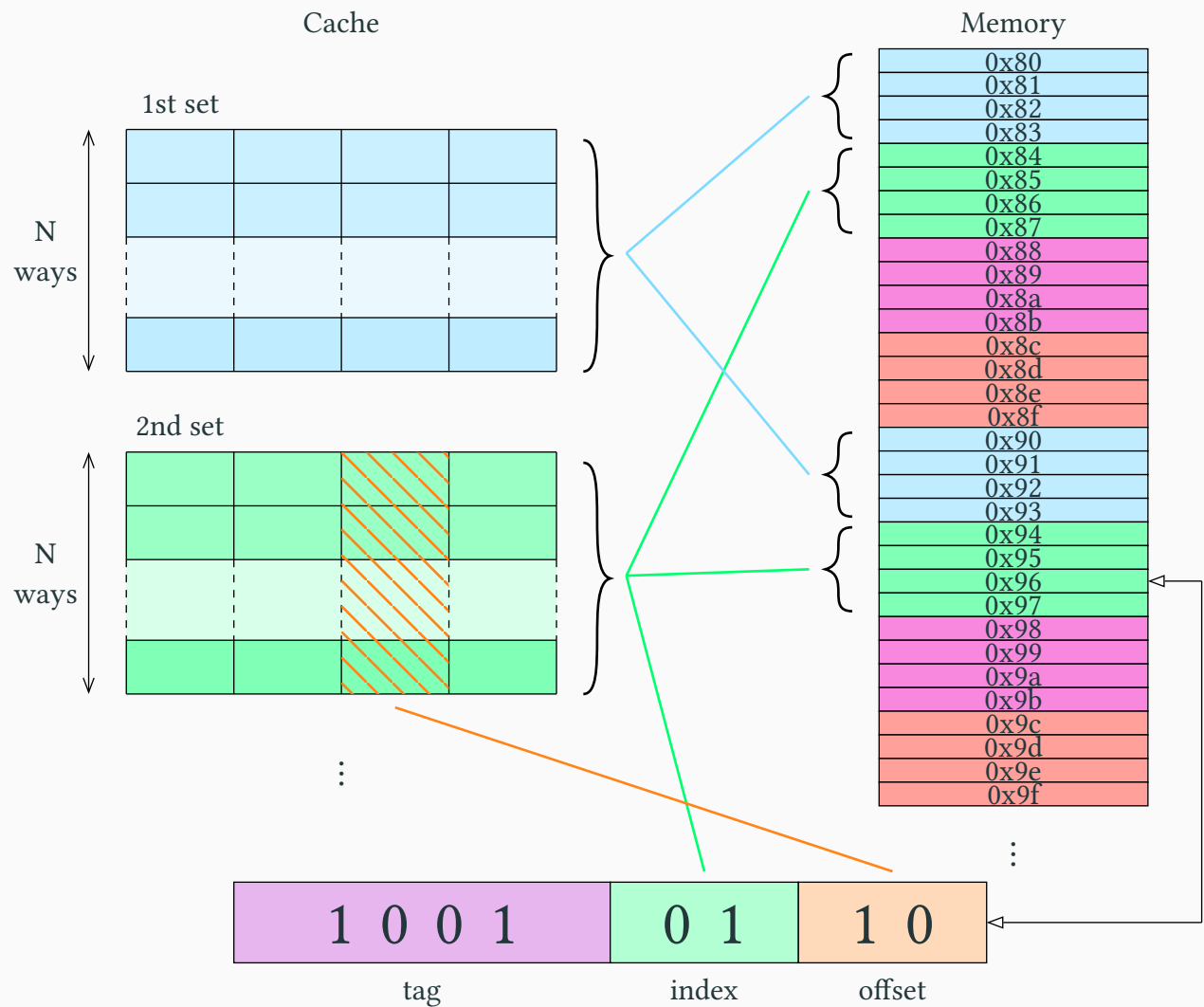


1. Context of internship

1. Context of internship

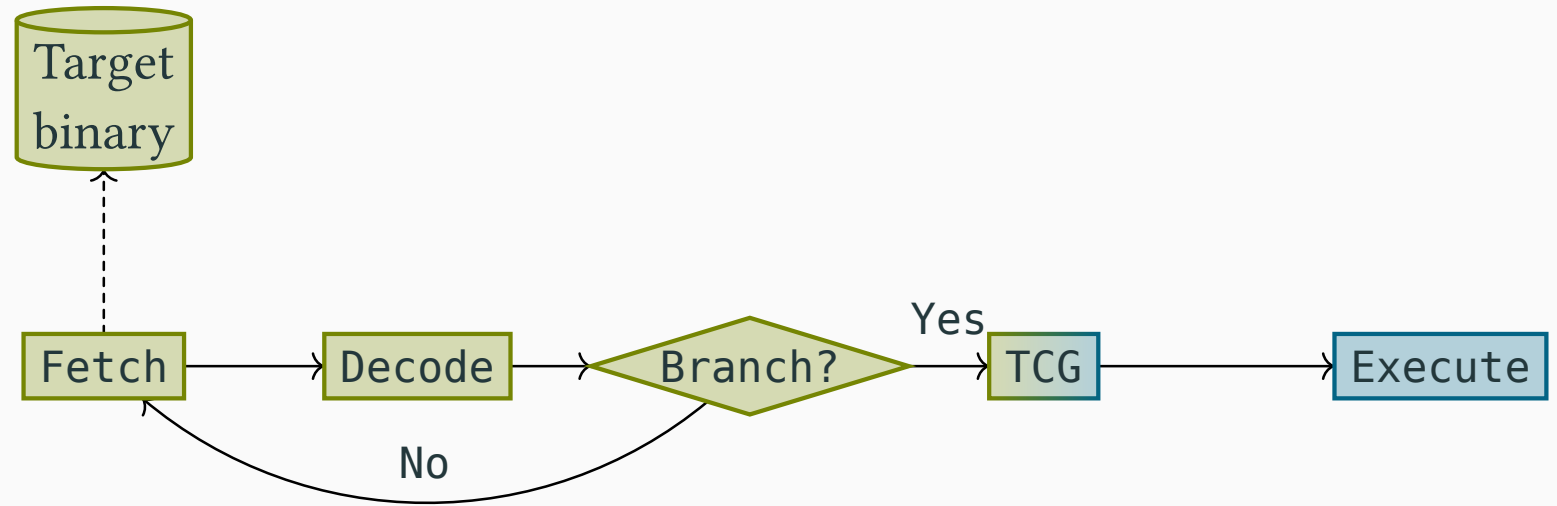
**gem5 is accurate but slow, while QEMU is fast
(10x on 1vCPU, even more with multicore systems),
but can it be really used by software engineers / researchers ?**

1.1 Cache 101 : N-way associative and direct mapped

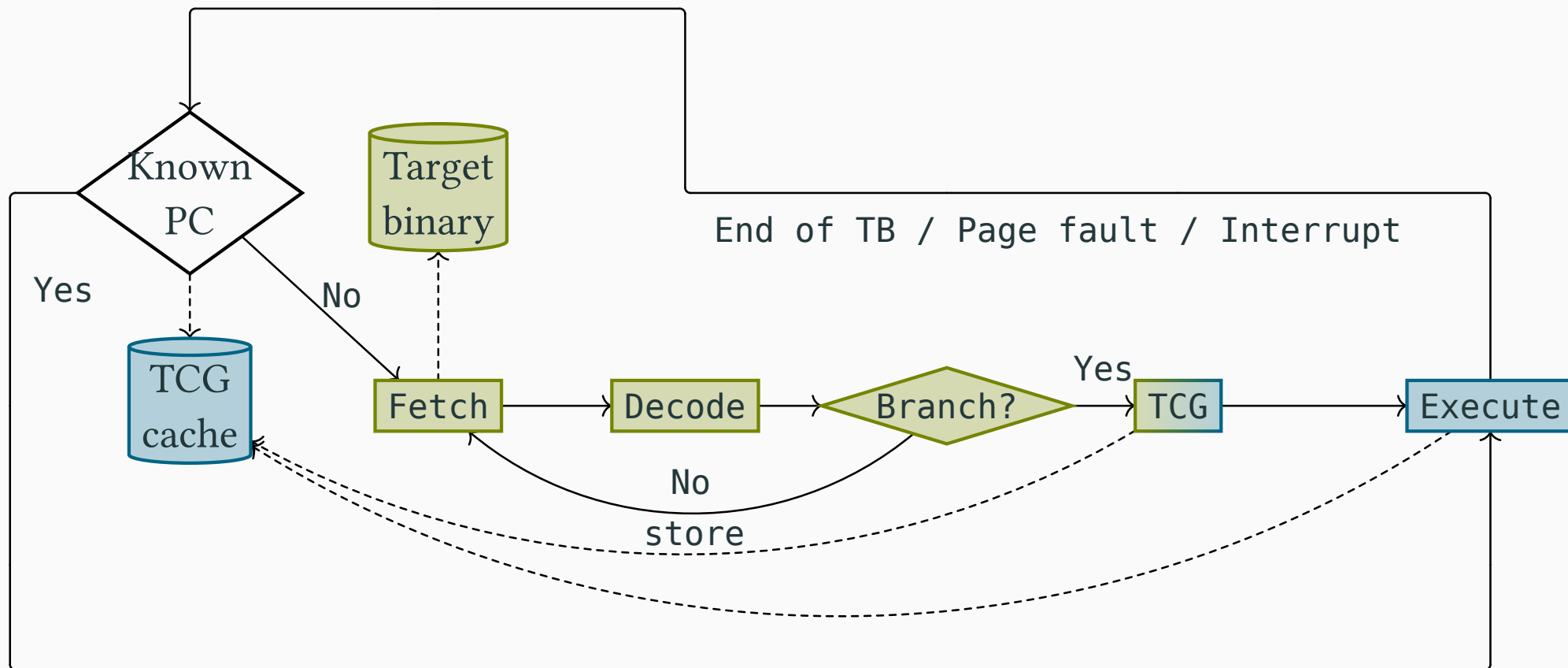


2. Our protagonists

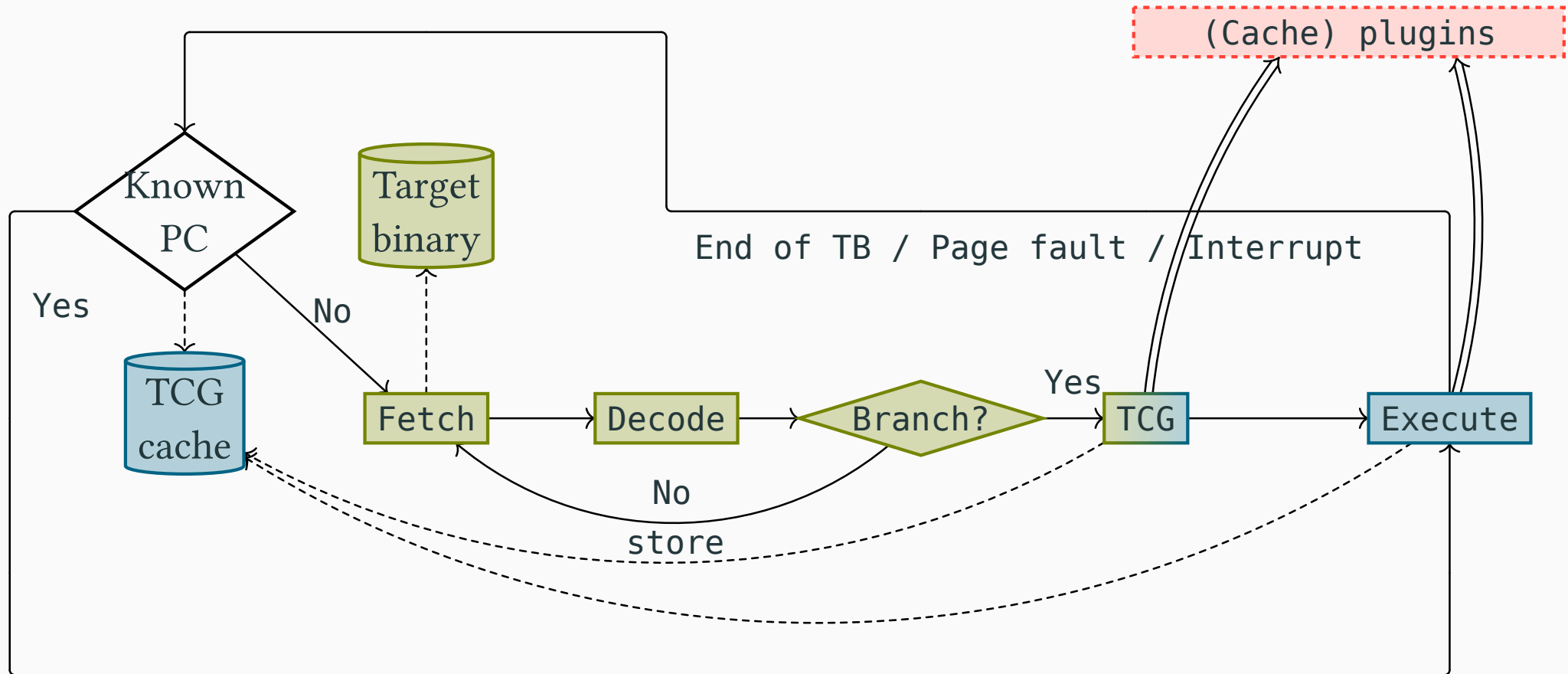
2.1 QEMU (Tiny Code Generator) + Cache plugin



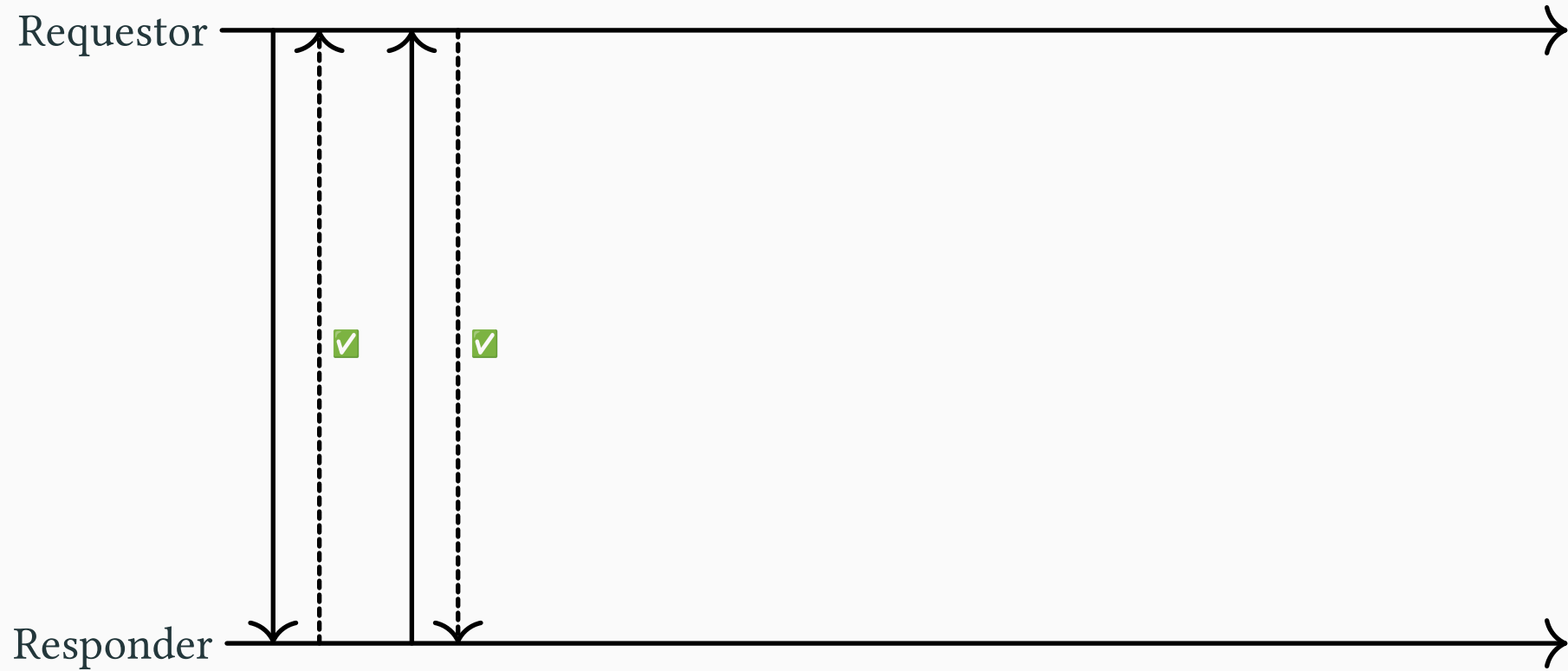
2.1 QEMU (Tiny Code Generator) + Cache plugin



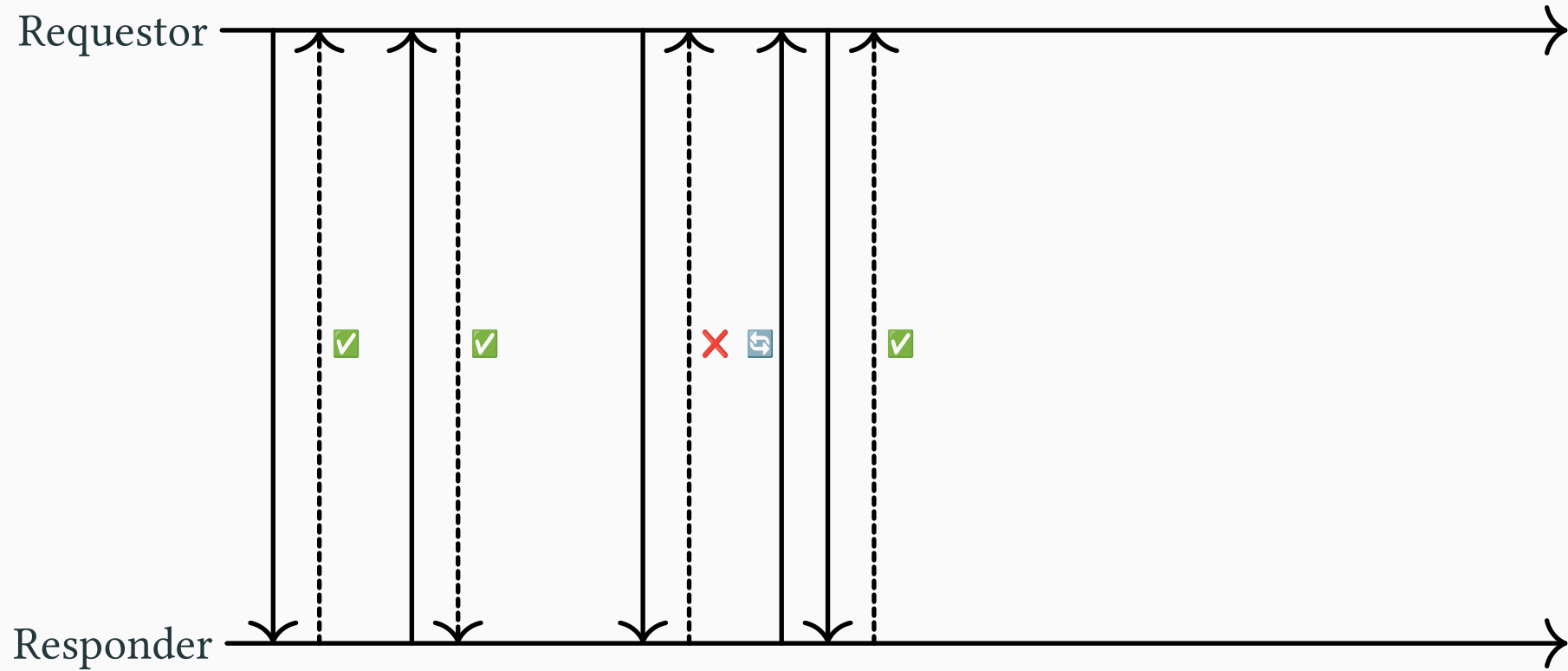
2.1 QEMU (Tiny Code Generator) + Cache plugin



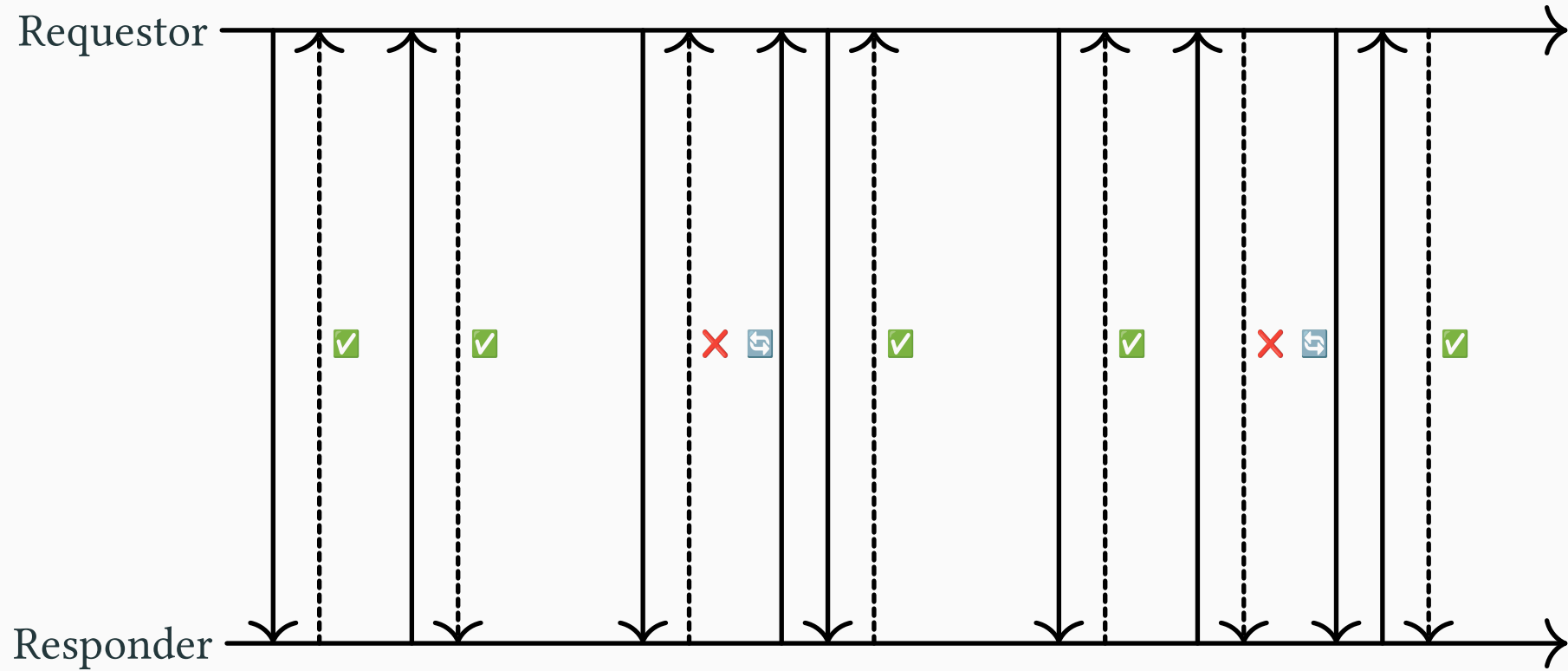
2.2 gem5



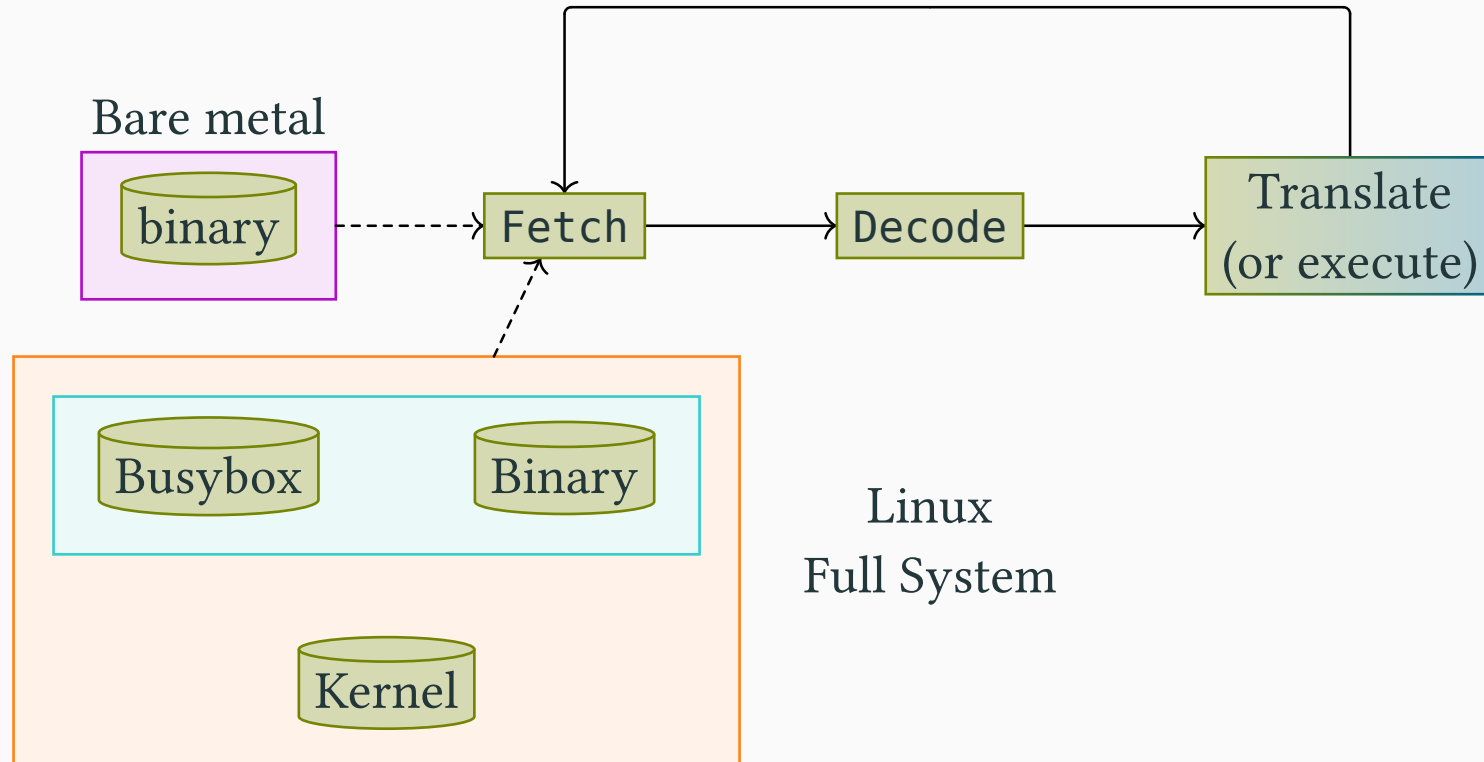
2.2 gem5



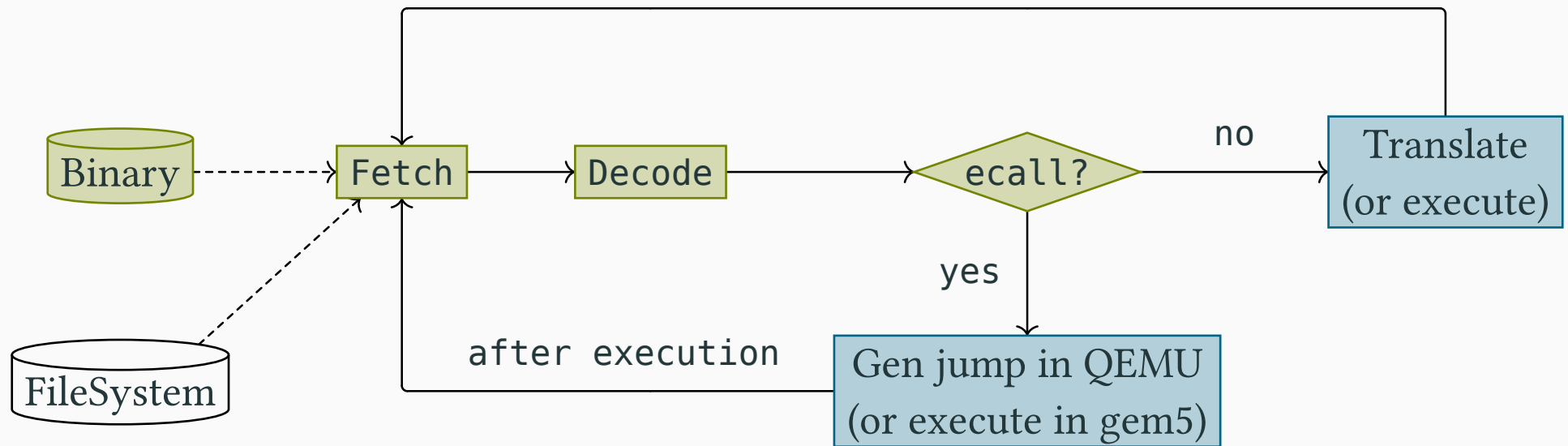
2.2 gem5



2.3 Simulation types



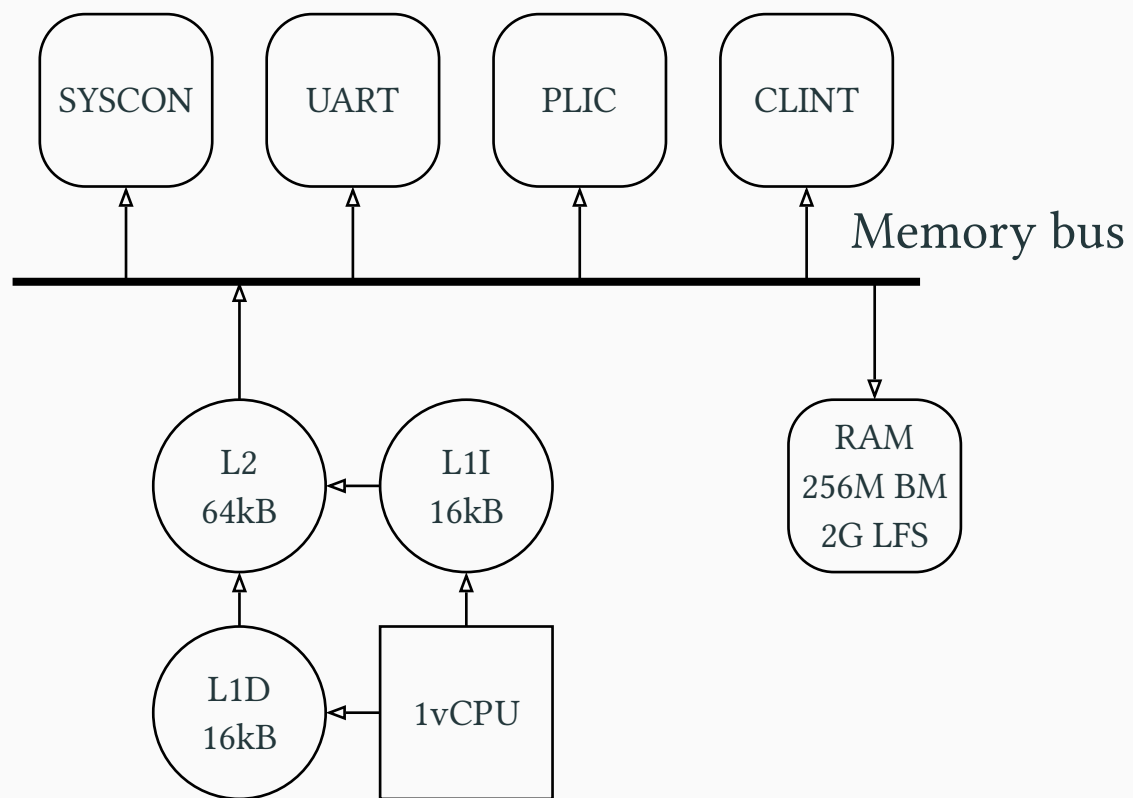
2.3 Simulation types



2.4 Our benchmarks

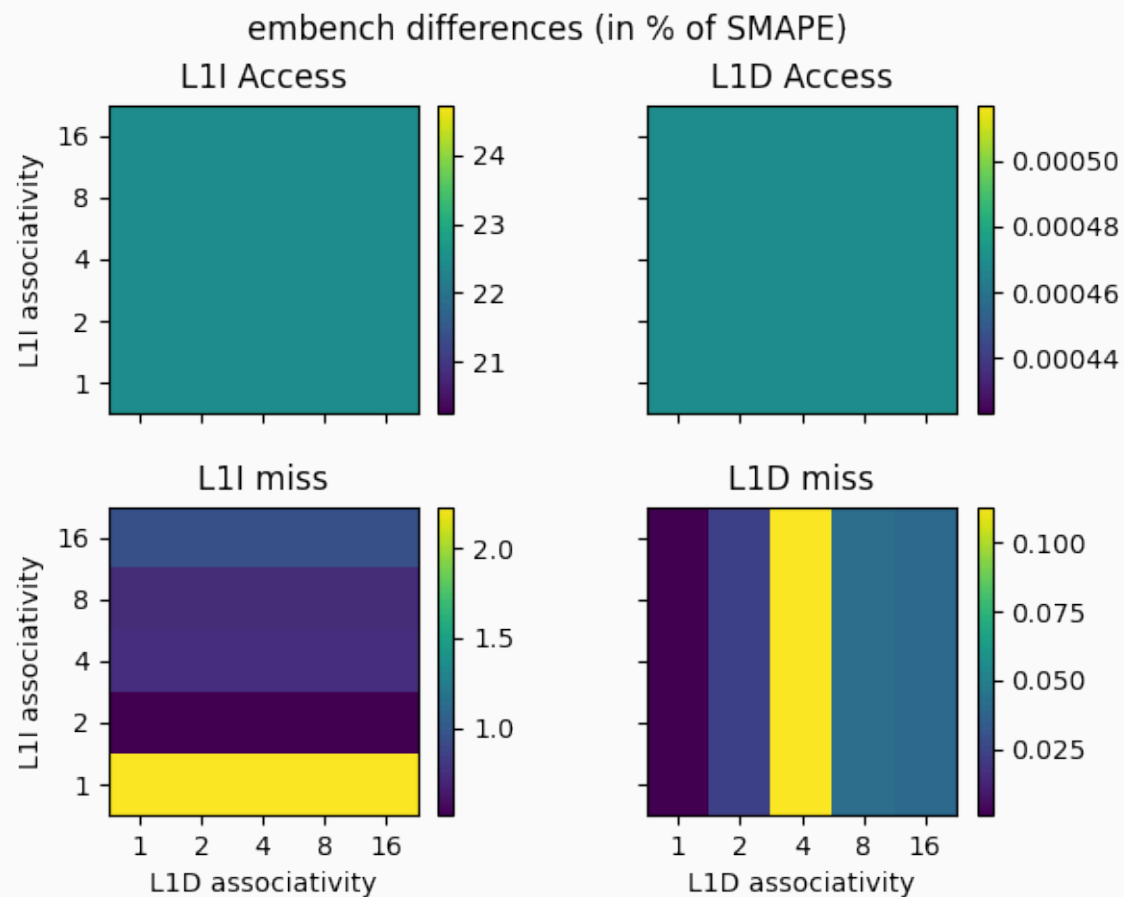
	PARSEC	Embench-iot
Description	Famous for share memory system	Adapted for MCU with minimal requirements (stdlib independant)
Workloads	cholesky, fft, lu_cb, radix	qrduino, cryptography, picojpeg, ...
Size	each one $\geq 45\text{kB}$ in text, $\geq 50\text{kB}$ in data/bss/rodata	Together $\approx 180\text{KB}$ in text, $\approx 60\text{Kb}$ in bss/data/rodata, but strong locality

2.5 Our system

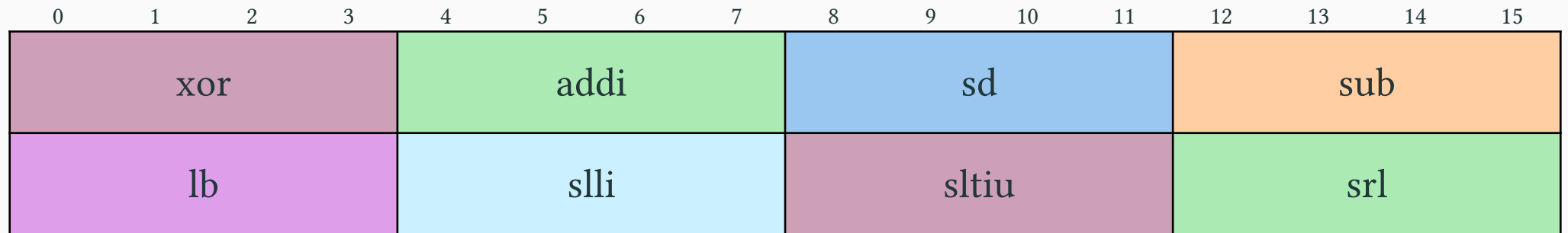


3. Results

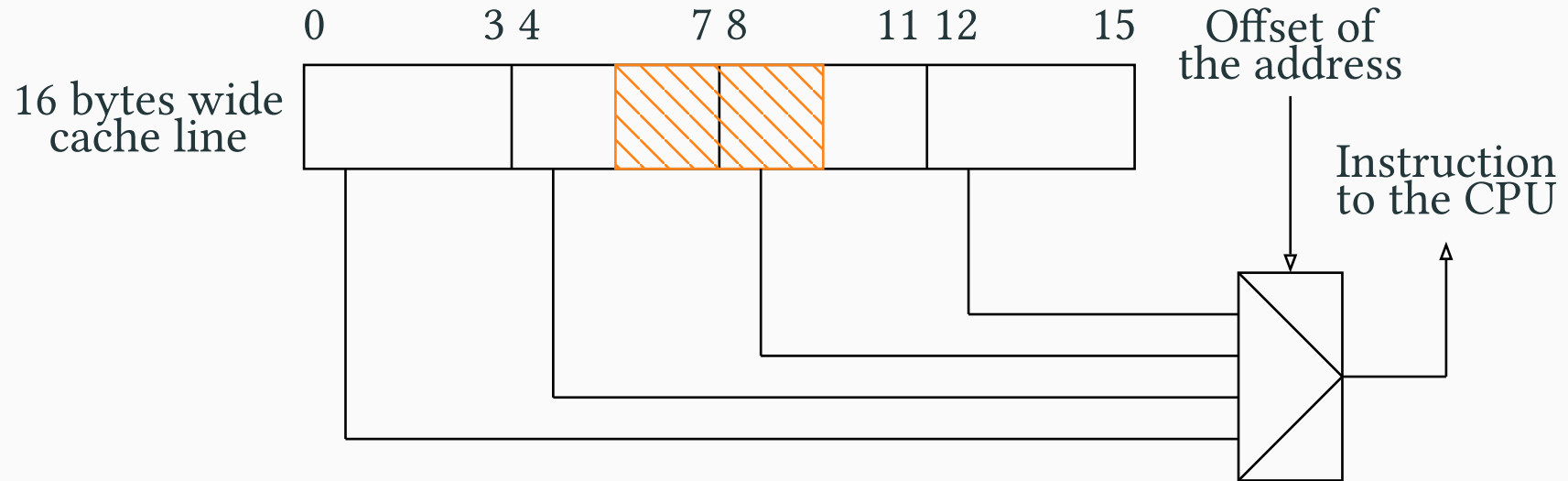
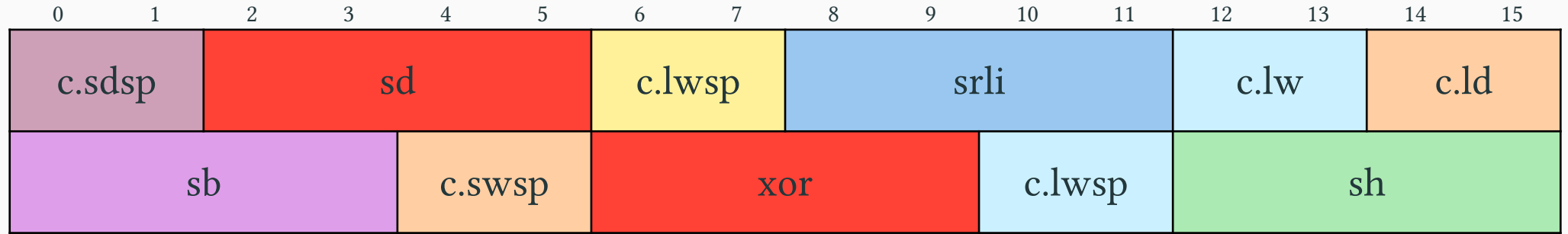
3.1 Bare metal embench



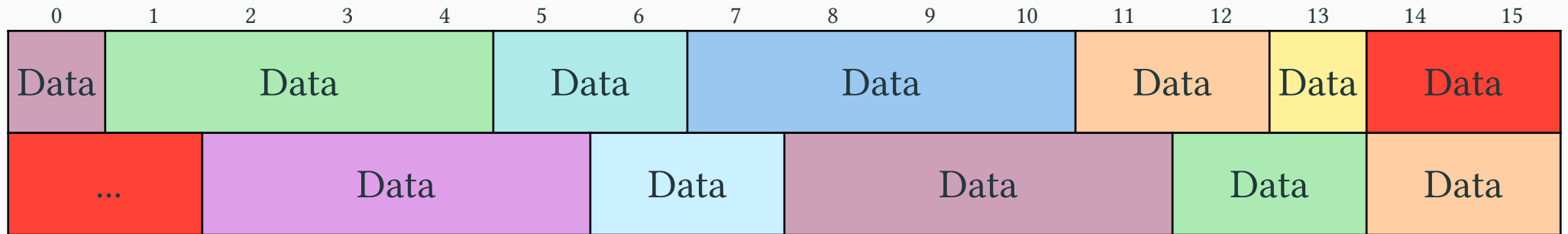
3.2 Multiple accesses to cache



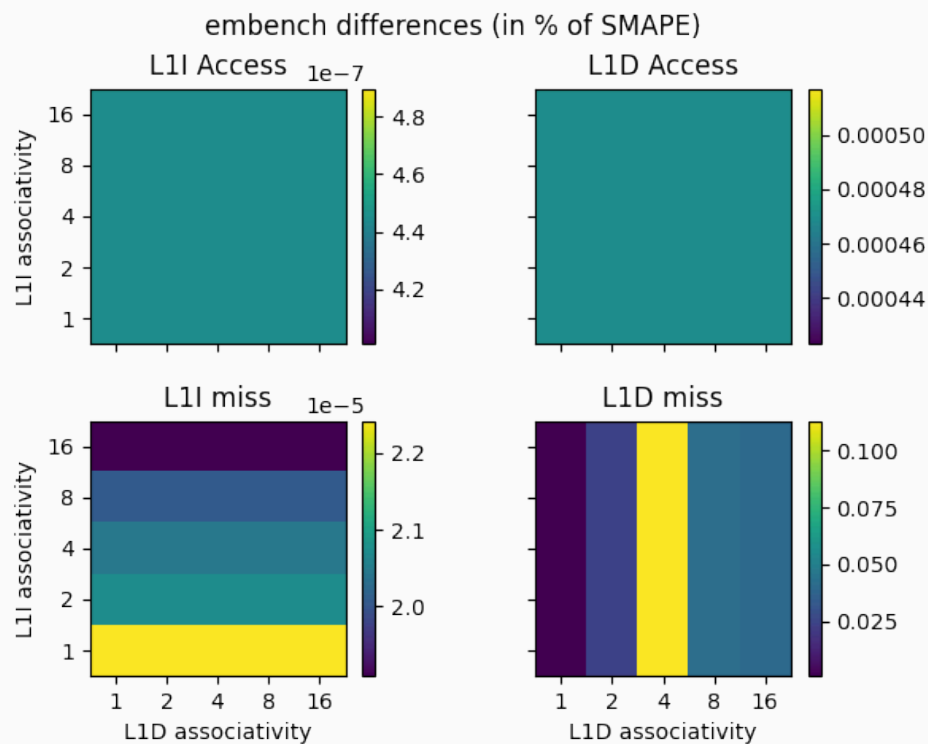
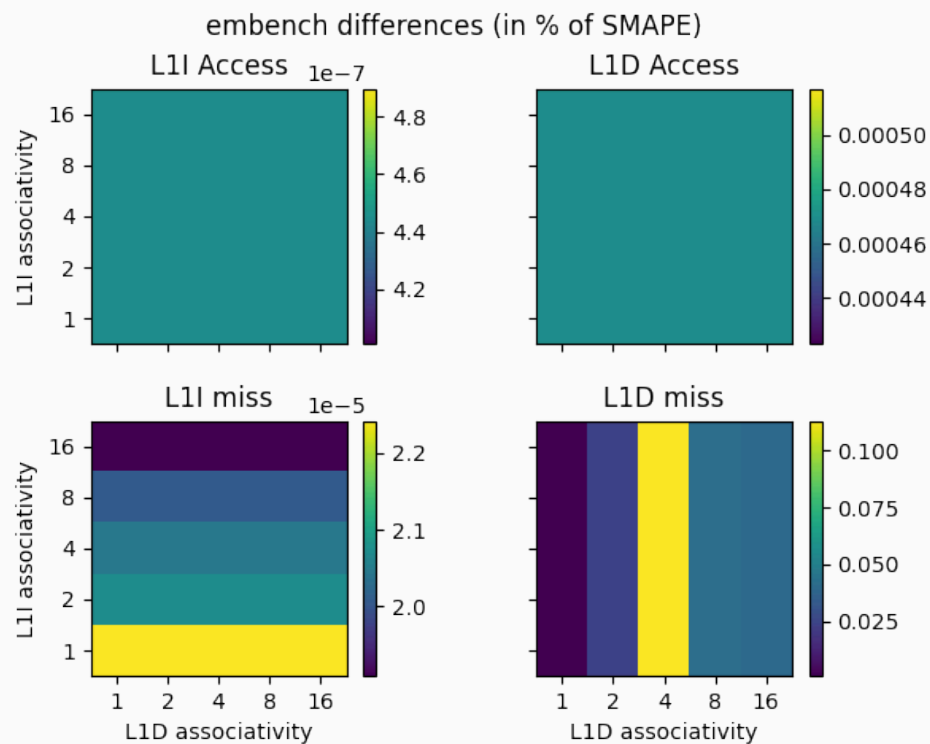
3.2 Multiple accesses to cache



3.2 Multiple accesses to cache



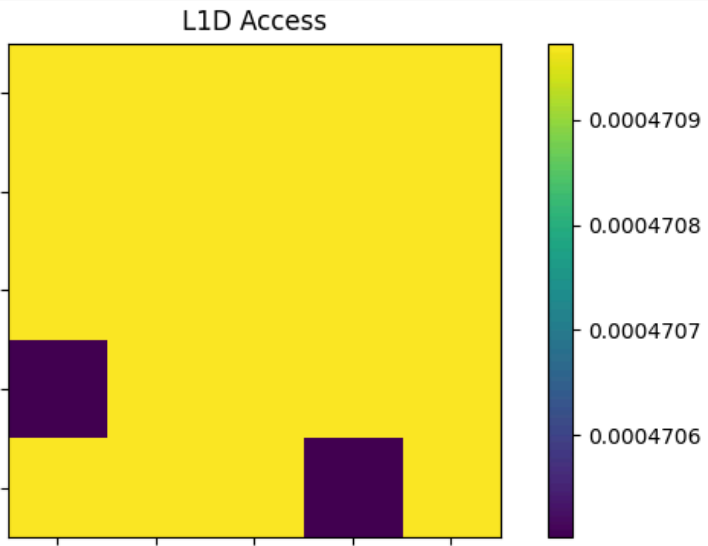
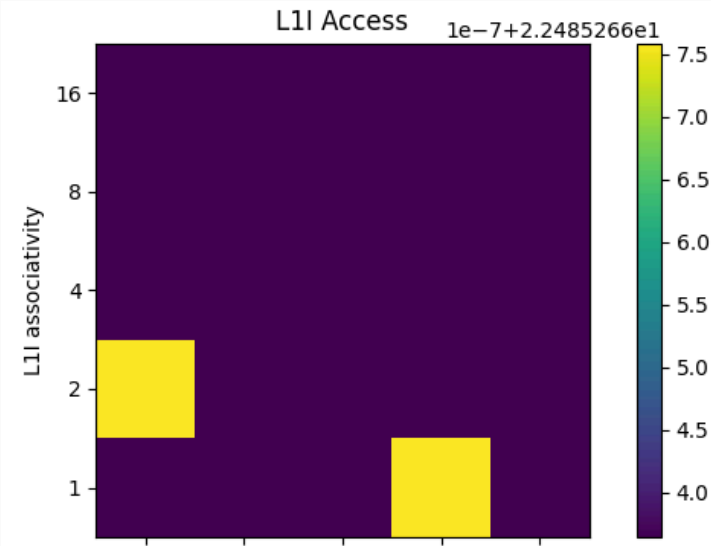
3.2 Multiple accesses to cache



3.3 The mystery of the holes



3.3 The mystery of the holes



$\lambda = 1254240576$	< 20 simulations	= 50 simulations	= 100 simulations
$\lambda + 0$	100%	82%	46%
$\lambda + 5$	0%	18%	42%
$\lambda + 10$	0%	0%	4%

3.3 The mystery of the holes

```
void poweroff(void) {  
    print("Poweroff requested\r\n");  
    *(uint32_t *)SYSCON_ADDR = 0x5555;  
}
```

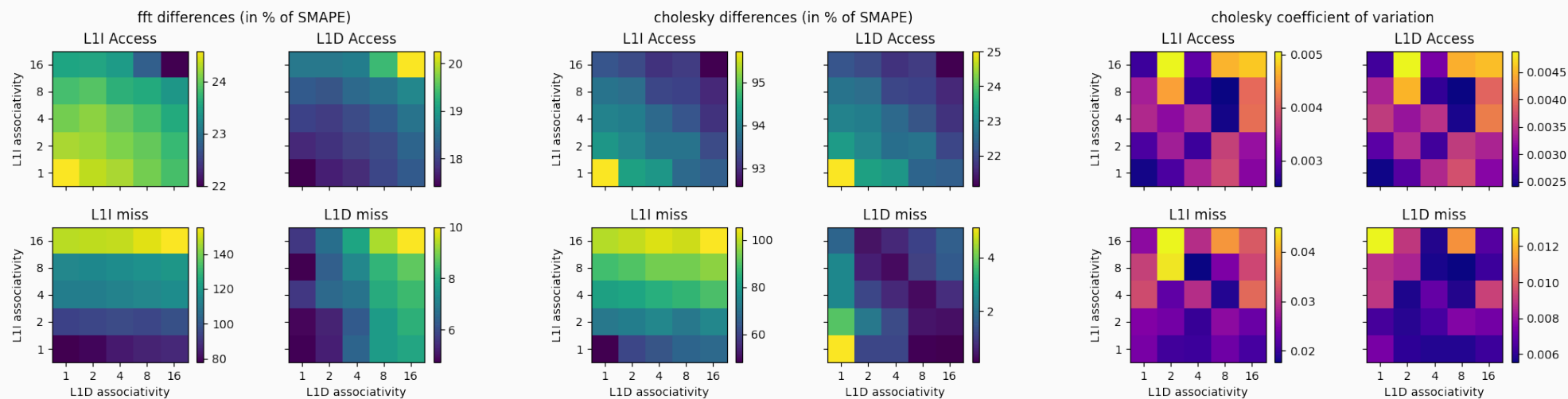
```
void poweroff(void) {  
    print("Poweroff requested\r\n");  
    *(uint32_t *)SYSCON_ADDR = 0x5555;  
    for(;;) { asm("wfi"); }  
}
```

4.

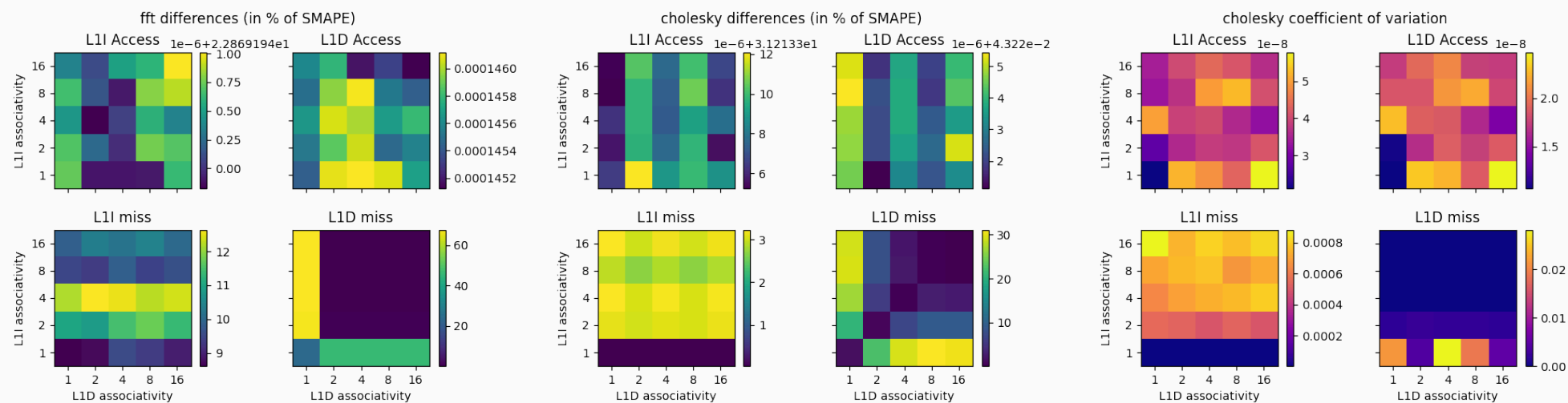


moment

4.1 FS linux + busybox + PARSEC



4.2 SE PARSEC



5. Conclusion

5. Conclusion

- gem5 isn't a always perfect simulator
- Most issues in QEMU don't seem to come from the cache plugin but rather inside QEMU
- A lot more to test (other cache parameters, QEMU situations) / to continue digging (timing in linux kernel on multi vCPU, ...)

Usage	Bare metal	Linux FS	SE
Software engineer comparing algorithms' usage of cache	✓ ✓ ✓	✓ ✓	✓
Not too accurate researcher work (eg testing different cache hierarchy)	✓ ✓ ✓	✓	-
Accurate experimental work of researcher (eg implementing new cache coherency protocols)	✓ ✓ ✓	-	✗

6. Opening

6. Opening

- Cache plugins :
 - Other cache properties (Prefetching, cache hierarchy, write policies, replacement policies, ...)
 - Other QEMU cache plugins (TBCache, MESI, ...)
- Differences in QEMU :
 - Other system properties (vCPU count)
 - Continue digging for differences in FS and SE

7. Thank you
