

PALMED & GUS: Performance prediction and optimisation through microbenchmarking and abstract resources

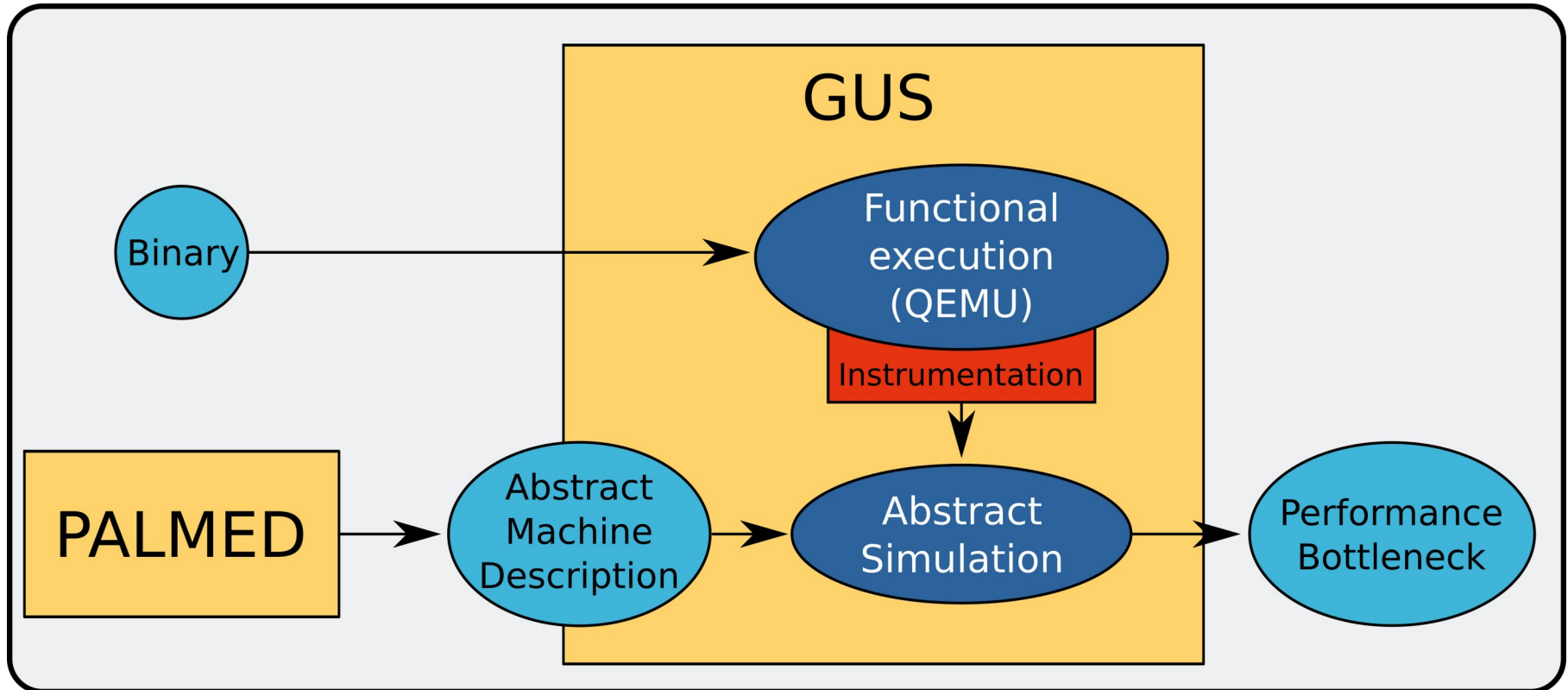
*Nicolas DERUMIGNY,
Université Grenoble Alpes*

*Under the supervision of
Fabrice Rastello,
INRIA*



CORSE

GUS: a complete, dynamic simulator



- Access to runtime information

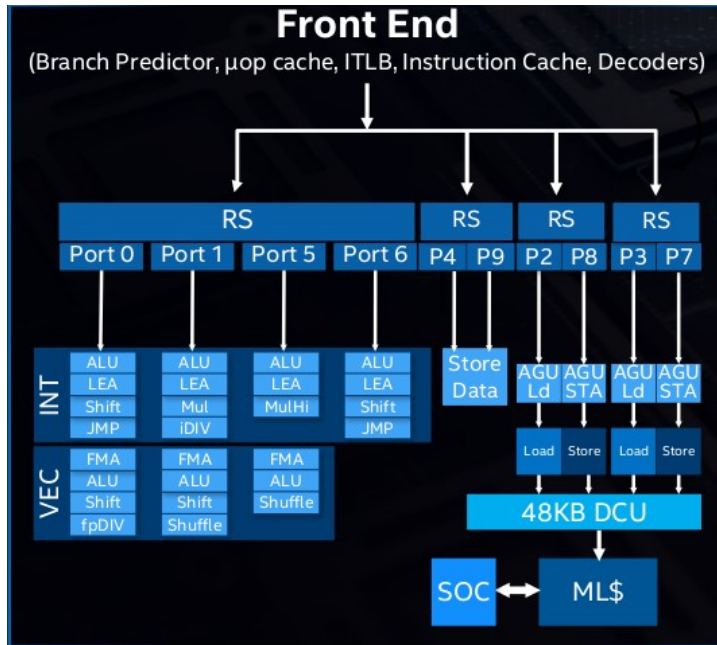
Similar (static) tools:

- IACA
- LLVM-MCA

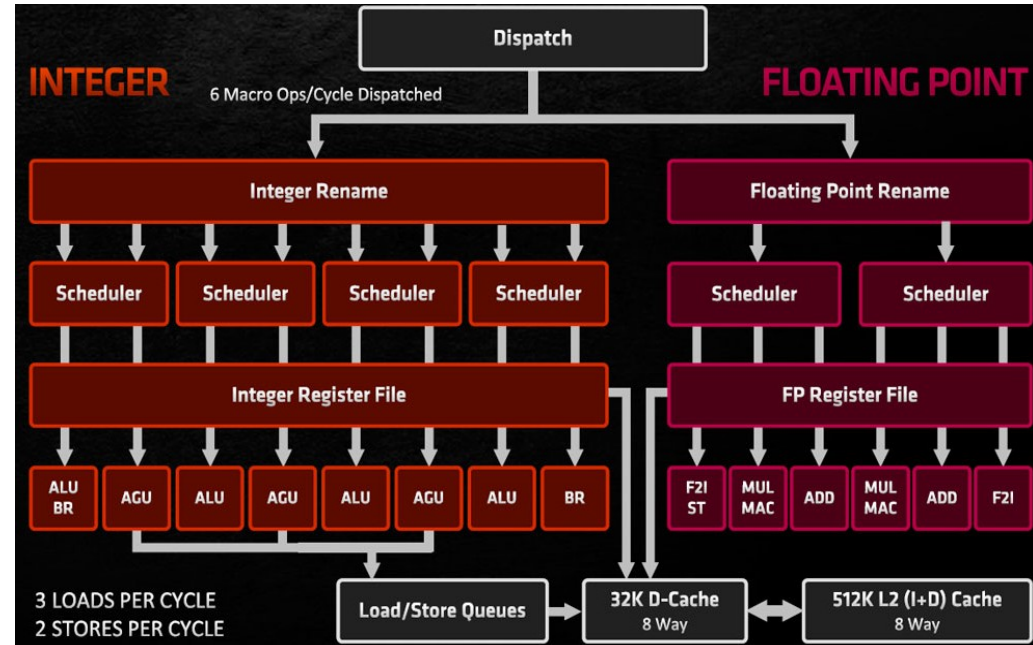


Demo!

The Port Model: an industry standard



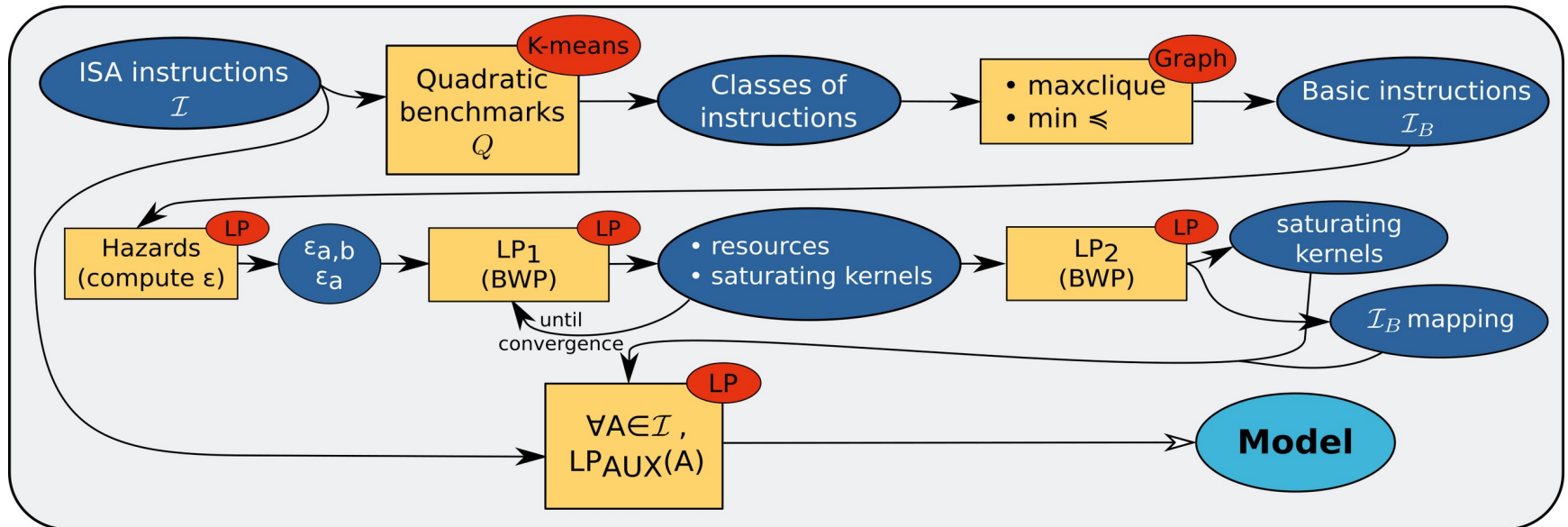
Intel's Sunny Cove back-end
(Intel Architecture Day, 2018)



AMD Zen 3 back-end
(AMD official presentation, 2020)

➔ Used in nearly every high-performance processors: x86/ARM

PALMED: Automated reverse-engineering...



Similar tools:

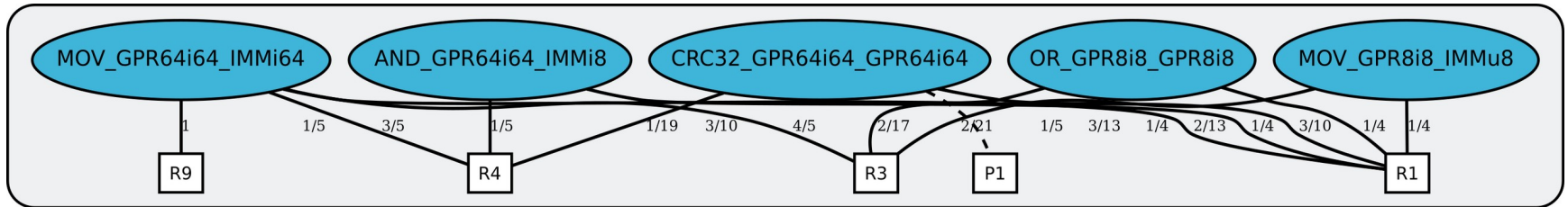
- Exegesis
- Uops.info

BUT

PALMED has:

- No need for performance counters

... and disjunctive *abstract* resources



- Simpler computation of the throughput (maximum of a sum)
- No need for a schedule
- No need for performance counters
- Can model every existing (conjunctive) port mapping

Open access to PALMED's results

Basic block input

```
# Try your own basic blocks here!
push %r12
mov %rdx, %r12
push %rbp
mov %rsi, %rbp
push %rbx
mov (%rdi), %rbx
movzx (%rbx), %eax
mov %rax, %rdx
mov 0xdac080(,%rax,4), %eax
cmp 0x8d, %dx
jz 0xaed780
```

Compute

The following instructions are not supported and have been ignored for this evaluation: ⓘ

- JZ_BRDISP32
- PUSH_GPR64i64

2.00 cycles ⓘ

3.50 IPC ⓘ

R6

2.00

MOV_GPR64i64_MEM64i64	0.50
MOVZX_GPR32i32_MEM64u8	0.50
MOV_GPR32i32_MEM64i32	0.50
CMP_GPR16i16_MEM64i16	0.50

R7

1.50

MOV_GPR64i64_GPR64i64	0.25
MOV_GPR64i64_GPR64i64	0.25
MOV_GPR64i64_MEM64i64	0.19
MOVZX_GPR32i32_MEM64u8	0.19
MOV_GPR64i64_GPR64i64	0.25
MOV_GPR32i32_MEM64i32	0.19
CMP_GPR16i16_MEM64i16	0.19

[What is going on here?](#)



Try it by yourself:

<https://palmed.corse.inria.fr/demo>