Speculative High-Level Synthesis of RISC-V Processors

Dylan Leothaud, Jean-Michel Gorius, Steven Derrien, Simon Rokicki

Univ Rennes, Inria, CNRS, IRISA

{dylan.leothaud, jean-michel.gorius, steven.derrien, simon.rokicki}@irisa.fr

Designing a RISC-V CPU *should* be as simple as writing an Instruction Set Simulator

Introduction

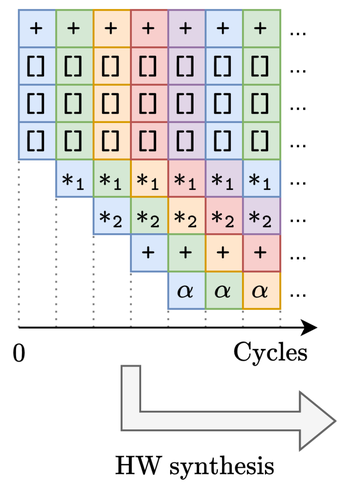
Context: Increasing need for customizable architectures for embedded applications.

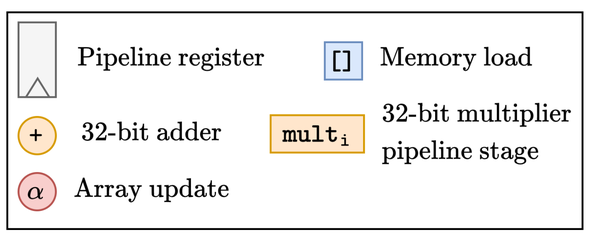
Problem: Micro-architectural design is tedious and error-prone, how do we make such customizations available to everyone?

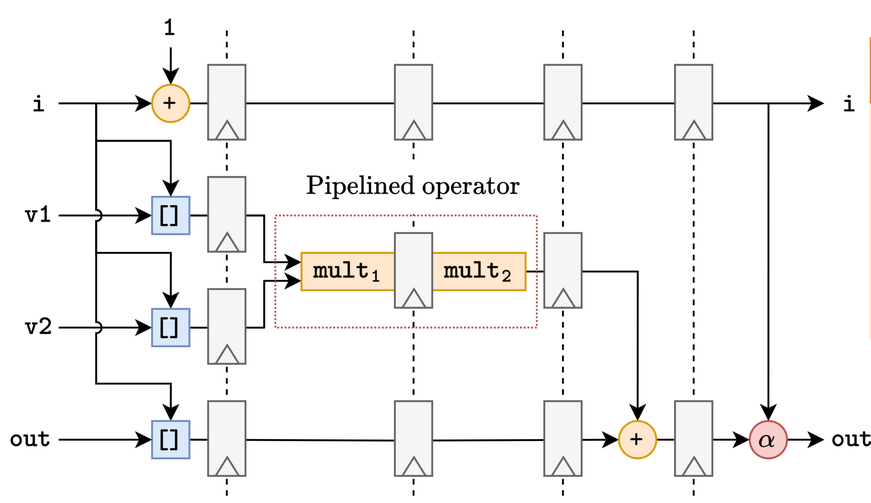
Our approach: Leverage High-Level Synthesis to **synthesize micro-architectures** from a single instruction set simulator in C.

```
void vec_mul(int *v1, int *v2, int *out,
             int n) {
  for (int i = 0; i < n; ++i)
    out[i] += v1[i] * v2[i];
}
```

HLS







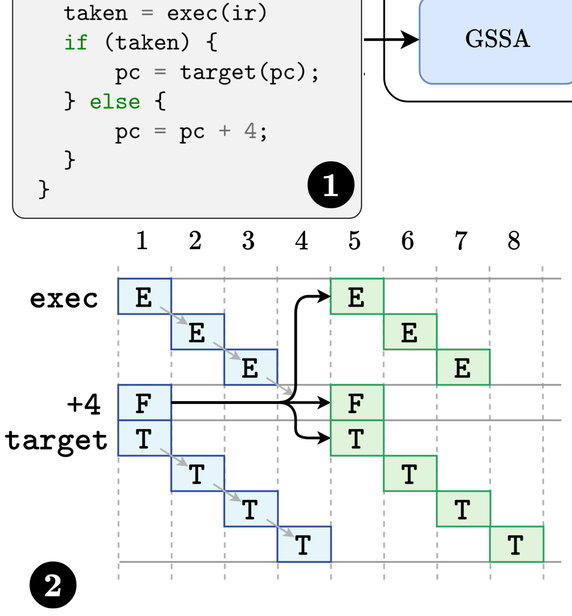
Key takeaway

High-Level Synthesis (HLS) tools allow automatic generation of hardware from C programs

From ISS to RISC-V Processors

```
while(1) {
  ir =  mem[pc];
  taken = exec(ir)
  if (taken) {
      pc = target(pc);
  } else {
      pc = pc + 4;
  }
}
```

❶

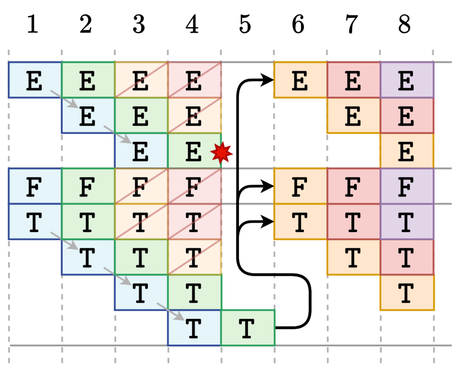


❷ Static pipelining

Speculative HLS: GSSA → SLP → Codegen

```
#pragma hls distance mis_pc=4
#pragma hls distance ctrl=3
do {
#pragma hls pipeline II=1
  ctrl[t] = branch(s_pc[t-3]);
  mis_pc[t] = target(s_pc[t-1]);
  cs = nextstate(cs, ctrl[t-3]);
  s_pc[t] = cs.selSlow ?
                  mis_pc[t-4] :
                  s_pc[t-1] + 4;
  if(cs.rollback)
    s_pc[t] = s_pc[t-4];
  if(cs.commit)
    pc = s_pc[t-4];
  t += 1;
} while(1);
```

❸



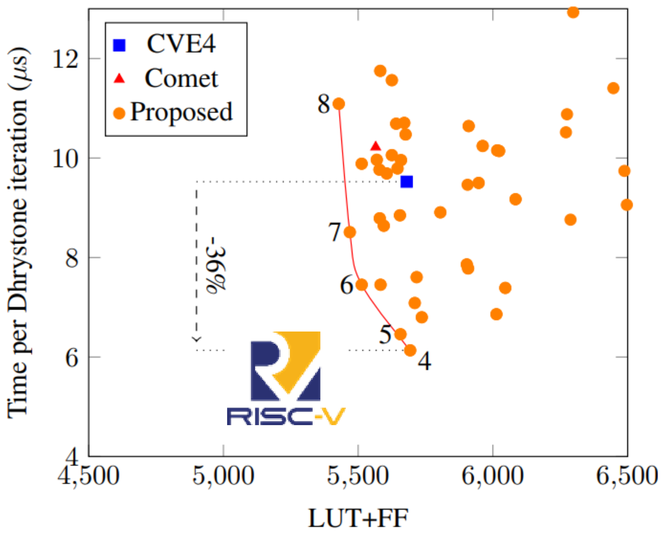
❹ Speculative pipelining

SpecHLS [1]

- Speculative HLS toolchain
- Backend-agnostic automatic transformation

GeCoS Generic Compiler Suite

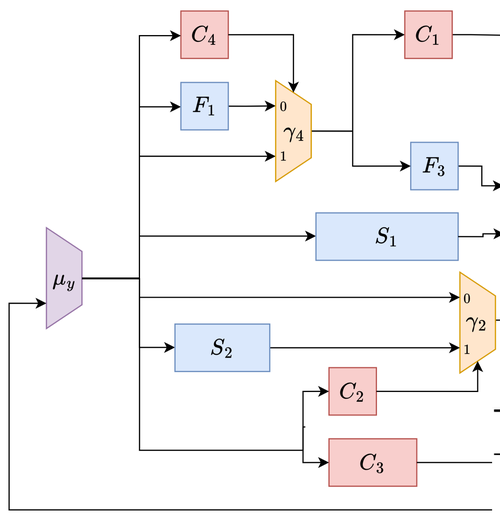
AMD Vitis



Key takeaway

Synthesizing RISC-V CPUs is possible [2], but time-consuming!

Efficient Design Space Exploration



Key takeaway

Speculation comes at the cost of design space explosion

4×10^{28} configurations for a simple RISC-V CPU!

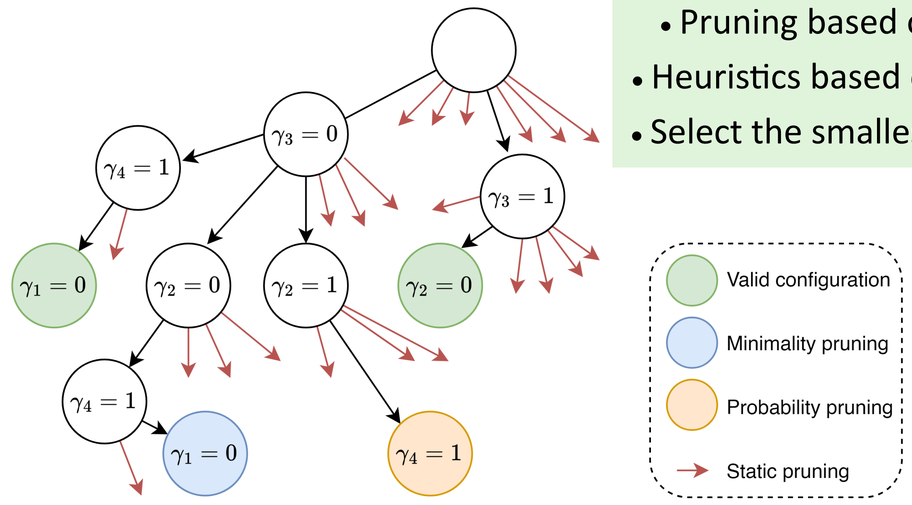
Scenario	γ_1	γ_2	γ_3	γ_4	CP (ns)	Prob.
configuration 1	1	0	1	0	2 ns	0.34
configuration 2	0	0	1	1	2 ns	0.21
configuration 3	0	0	0	1	8 ns	0.17
configuration 4	1	1	0	0	27 ns	0.10
configuration 5	0	1	0	0	13 ns	0.08
configuration 6	1	0	0	1	27 ns	0.04
static	-	-	-	-	27 ns	1
speculative 1	0	-	0	1	8 ns	0.17
speculative 2	-	0	1	-	2 ns	0.58

Exploring speculation opportunities

- Proposed algorithm based on Branch-and-Bound approach [3]
- Design space exploration leveraging *intertwined speculations* to obtain the best designs

Pruning configurations

- Pruning based on probability
- Heuristics based on achievable II
- Select the smallest configuration



Conclusion

- We need speculation to synthesize processor cores [2, 4].
- Processor design from an ISS using **speculative pipelining** enables **fast iteration times** and **intuitive** design exploration.

References

[1] Gorius, J.-M., Rokicki, S., and Derrien, S. (2022). *SpecHLS: Speculative Accelerator Design using High-Level Synthesis*. IEEE MICRO.

[2] Gorius, J.-M., Rokicki, S., and Derrien, S. (2022). *Design Exploration of RISC-V Soft-Cores through Speculative High-Level Synthesis*. ICFPT'22.

[3] Leothaud, D., Gorius, J.-M., Rokicki, S., and Derrien, S. (2024). *Efficient Design Space Exploration for Dynamic and Speculative High-Level Synthesis*. FPL'24.

[4] Derrien, S., Marty, T., Rokicki, S., and Yuki, T. (2020). *Toward speculative loop pipelining for high-level synthesis*. TCAD.