Rapid Prototyping of Complex Micro-architectures through High-Level Synthesis

Sara Hoseininasab, Caroline Collange, Steven Derrien
Inria, Univ Rennes, CNRS, IRISA, Rennes, France

✓ Synthesizing a Single-threaded CPU simulator through HLS:

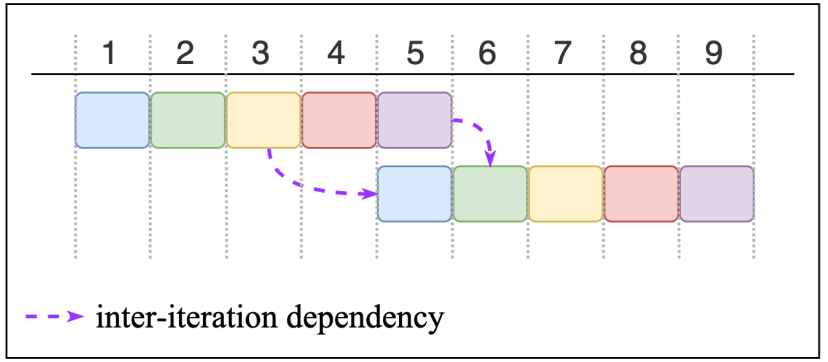
RAW on RF

RAW on pc

Input code

```
while(1) {
#prgma hls pipeline II = 1
    ir = I-Mem[pc];
    val1 = RF[ir.rs1];
    val2 = RF[ir.rs2];
    taken_br, result, ir.addr  = exec(val1, val2);
    if(ir == load)
      result = Memory(ir.addr);
    if(ir == store)
      Memory(ir.addr, val2);
    RF[ir.rd] = result;
    if(taken_br)
      pc = pc_branch;
    else
      pc = pc + 4;
}
```

Schedule by HLS



✓***II = 4*** **----->** *HLS* relies on static scheduling and considers worst-case scenarios

✓ How can achieve II = 1?

✓ To address this challenge, some works expose the pipeline stages in the simulator code and force HLS to follow the explicit scheduling [1]:

An explicitly pipelined simulator

```
while(1) {
#prgma hls pipeline II = 1
    writeback_tmp(memtowb);
    memtowb_tmp = memory(extomem);
    extomem_tmp = execute(dctoex);
    dctoex_tmp = decode(fetodc);
    fetodc_tmp = fetch();
  /* Handling stalls */
  if(!stall[fetch])
     fetodc = fetodc_temp;
  if(!stall[decode])
     dctoex = dxtoex_temp;
  ...
}
```

RTL in disguise!!

- ✓ Forces HLS to follow this schedule rather than relying on its automatic scheduling ---> not fully utilizing HLS capabilities
- ✓ Same level of complexity as RTL ---> not raising the level of abstraction
- ✓ Requires identification and handeling of pipeline stages and pipeline hazards
- ✓ Modifying the pipeline structure often requires a complete rewrite, making DSE difficult

Our approach: I will explain in my poster session how we exploit HLS features to synthesize complex micro-architectures **without** delving into RTL implementation.

[1] Simon Rokicki, Davide Pala, Joseph Paturel, Olivier Sentieys. What You Simulate Is What You Synthesize: Designing a Processor Core from C++ Specifications. ICCAD 2019 - 38th IEEE/ACM International Conference on Computer-Aided Design, Nov 2019, Westminster, CO, United States.