信息科学与技术学院
School of Information Science and Technology

CS 110
Computer Architecture
Multi-issue

Instructors: Siting Liu & Yuan Xiao
Course website: https://faculty.sist.shanghaitech.edu.cn/liust/courses/CS110.html
School of Information Science and Technology (SIST)
ShanghaiTech University

2025/4/21

Administratives

- Lab 7 released: project 1.1 check.
- HW 4 on-going, ddl April 21st! Start early!
  - The template is updated, please use the most up-to-date starter file!
- Project 1.2 on-going, DDL April 27th.
- Mid-term I Apr. 23rd 8:00am-10:00am
  - Location: Teaching center 301 (Siting's class);

    SIST1D107/108 (Yuan's class); Check on your egate;
  - Prepare your cheat sheet in advance!
  - Materials until Apr. 21st lecture.
- Discussion (SPST 4-122 18:00 - 19:40) schedule
  - Apr. 17th on mid-term I review by Chaofan Li;
  - Apr. 24th on datapath by Yuxuan Li.

Outline

- Overview on parallelism
- Multi-issue
  - Static multi-issue (VLIW)
  - Dynamic multi-issue (superscalar)
  - Design cases in modern computer systems

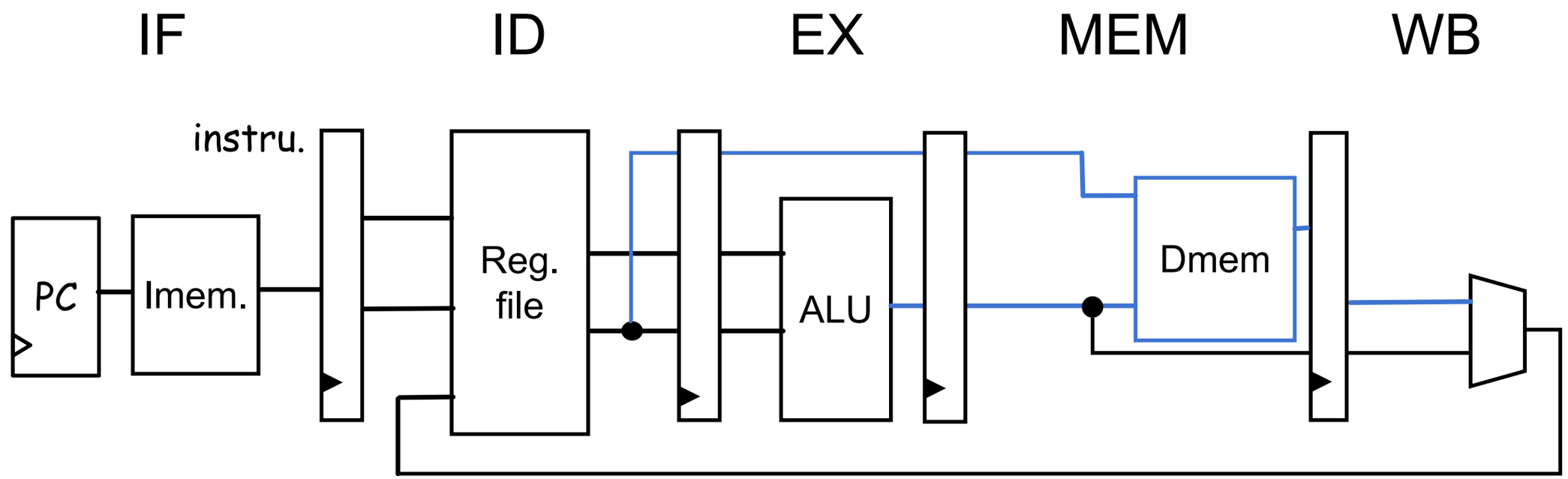
Review

- Overview on parallelism
  - **Instruction-level parallelism (ILP)**
    - Pipeline, deeper for faster clock, but potentially more hazards
    - Today's lecture (Multi-issue)
  - Data-level parallelism (DLP)
    - SIMD
- Thread-level parallelism (TLP)
  - Multi-threading/Hardware hyper-threading

$$\frac{Time}{Program} = \frac{Instructions}{Program} \cdot \frac{Cycles}{Instruction} \cdot \frac{Time}{Cycle}$$

Single-issue

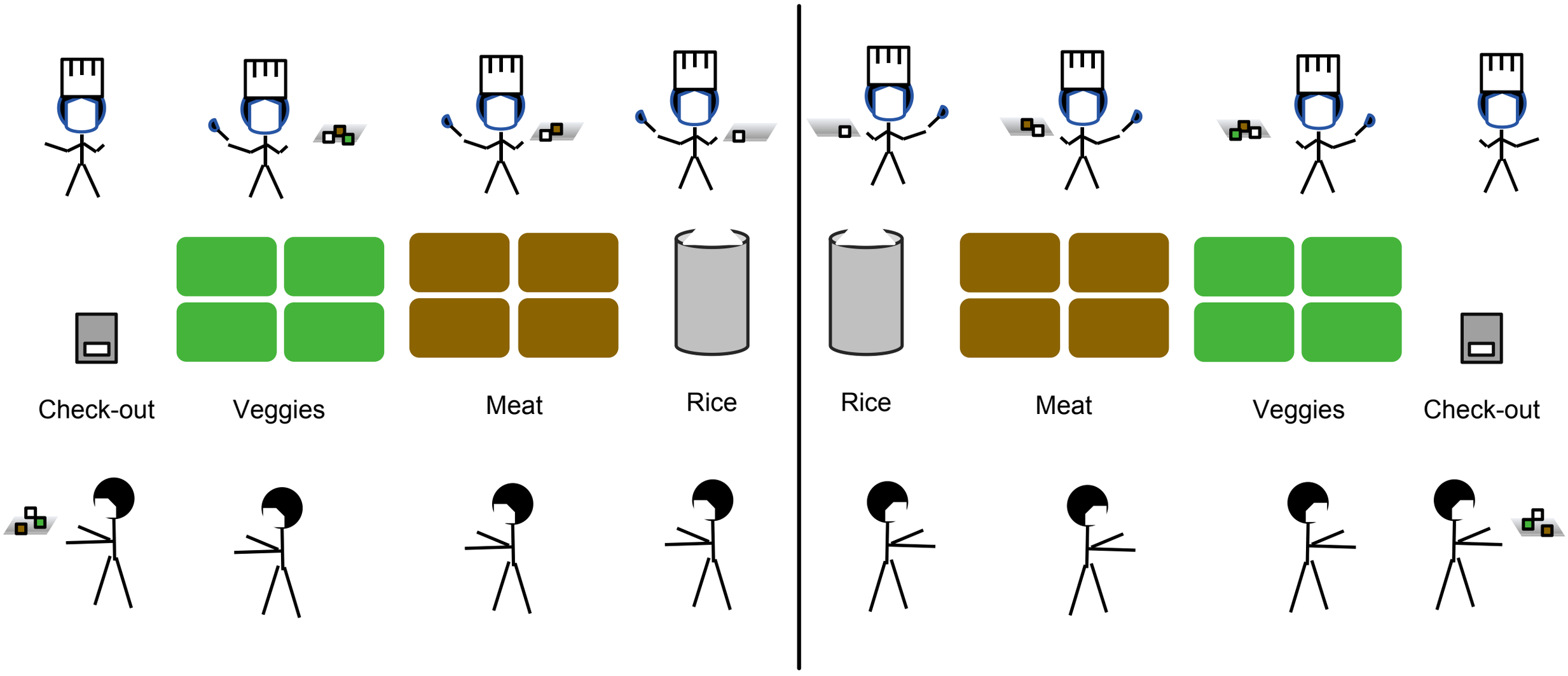
- Simplified 5-stage pipelined single-issue CPU datapath



- At most 1 instruction is “issued” to the datapath at 1 clock cycle

$$\text{average CPI} \geq 1$$

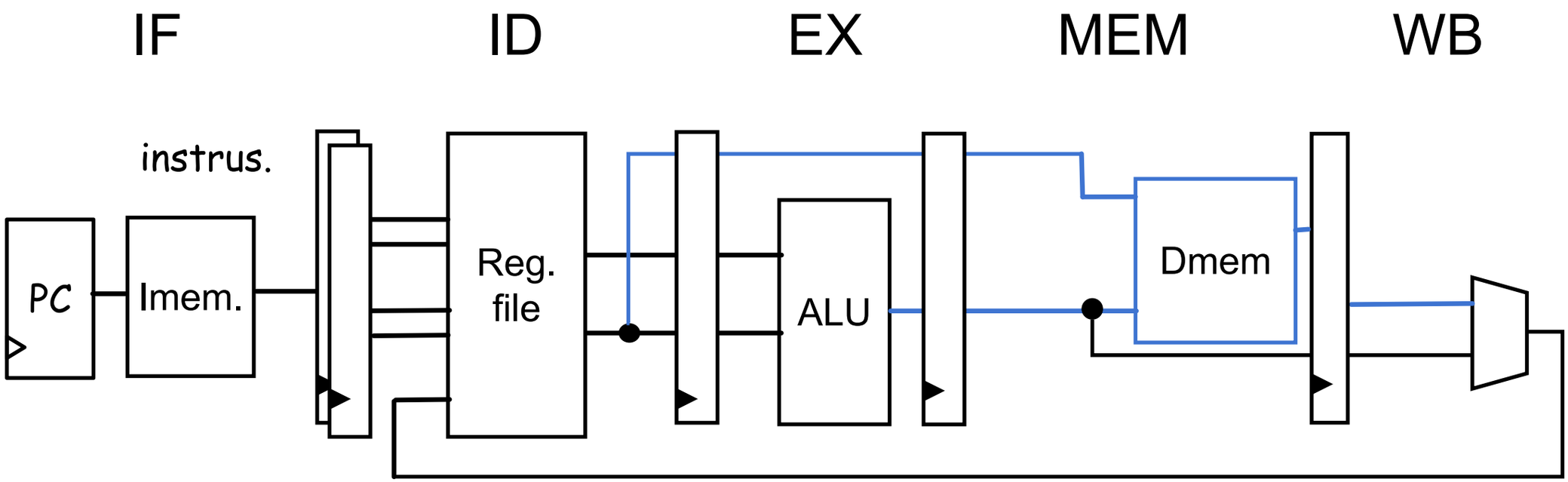
Multi-issue



Canteen analogy
(combined with pipelining)

Multi-issue (Hardware)

- Multi-issue CPU datapath

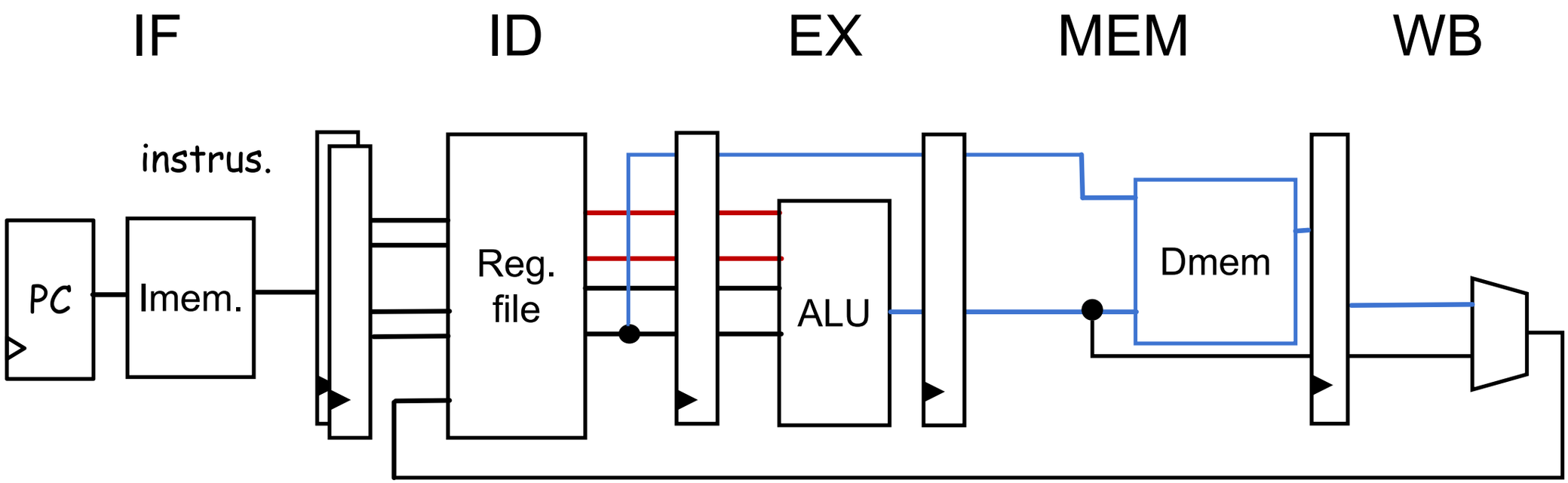


Hardware has to be adapted to avoid structural hazards

- Issue multiple instructions to the datapath in 1 clock cycle, average CPI may be smaller than 1.

Multi-issue (Hardware)

- Multi-issue CPU datapath

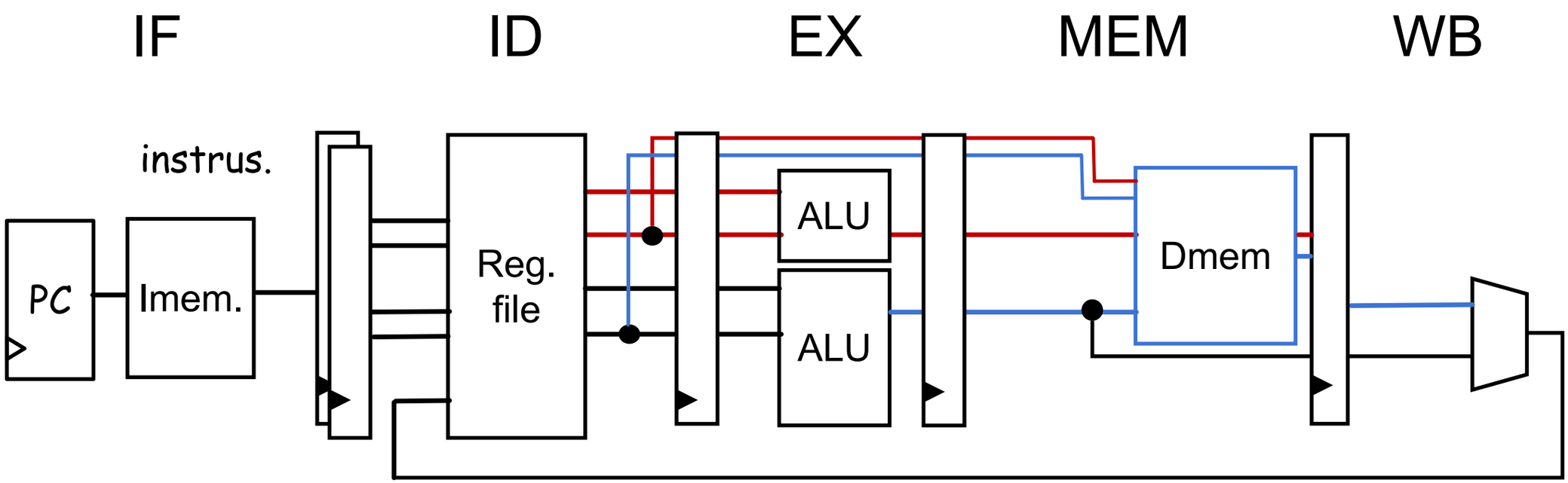


Hardware has to be adapted to avoid structural hazards

- Issue multiple instructions to the datapath in 1 clock cycle, average CPI may be smaller than 1.

Multi-issue (Hardware)

- Multi-issue CPU datapath

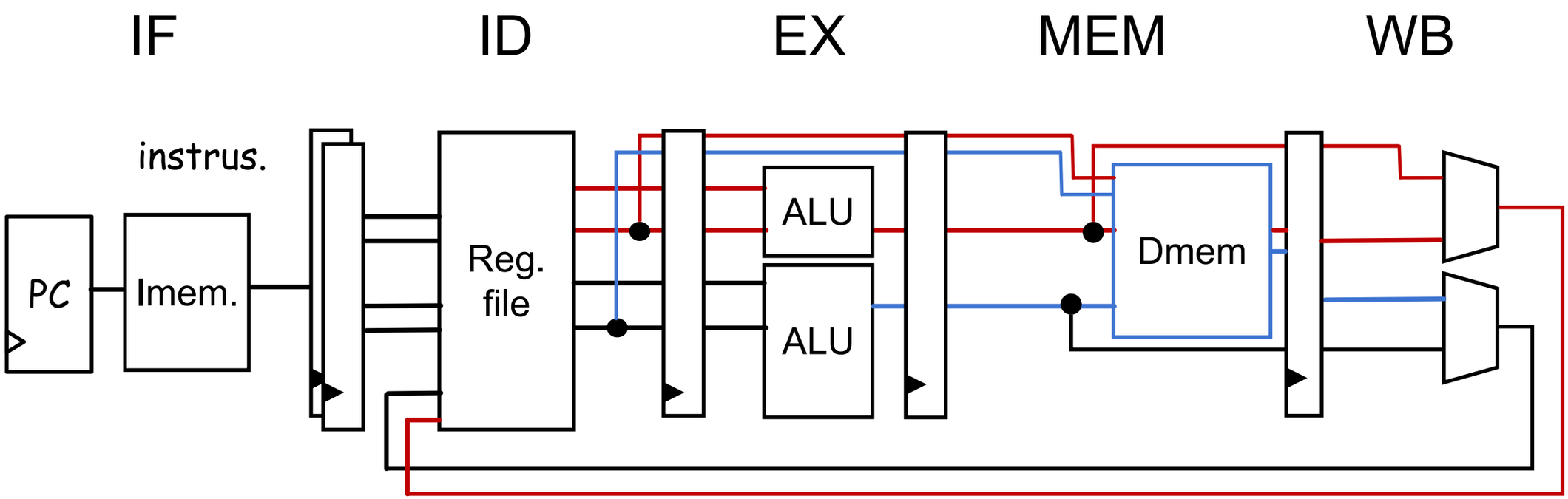


Hardware has to be adapted to avoid structural hazards

- Issue multiple instructions to the datapath in 1 clock cycle, average CPI may be smaller than 1.

Multi-issue (Hardware)

- Multi-issue CPU datapath

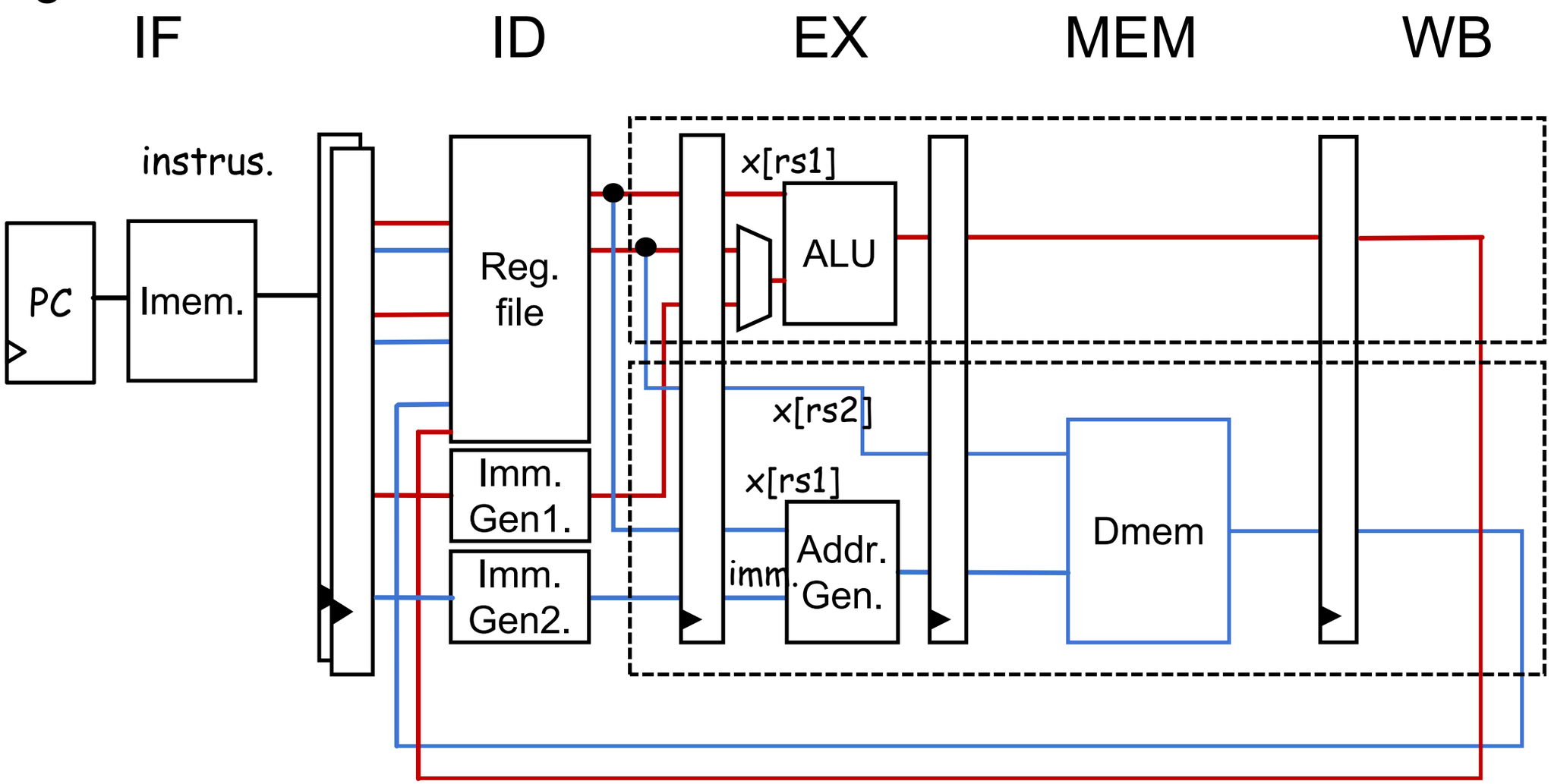


Hardware has to be adapted to avoid structural hazards

- Issue multiple instructions to the datapath in 1 clock cycle, average CPI may be smaller than 1.

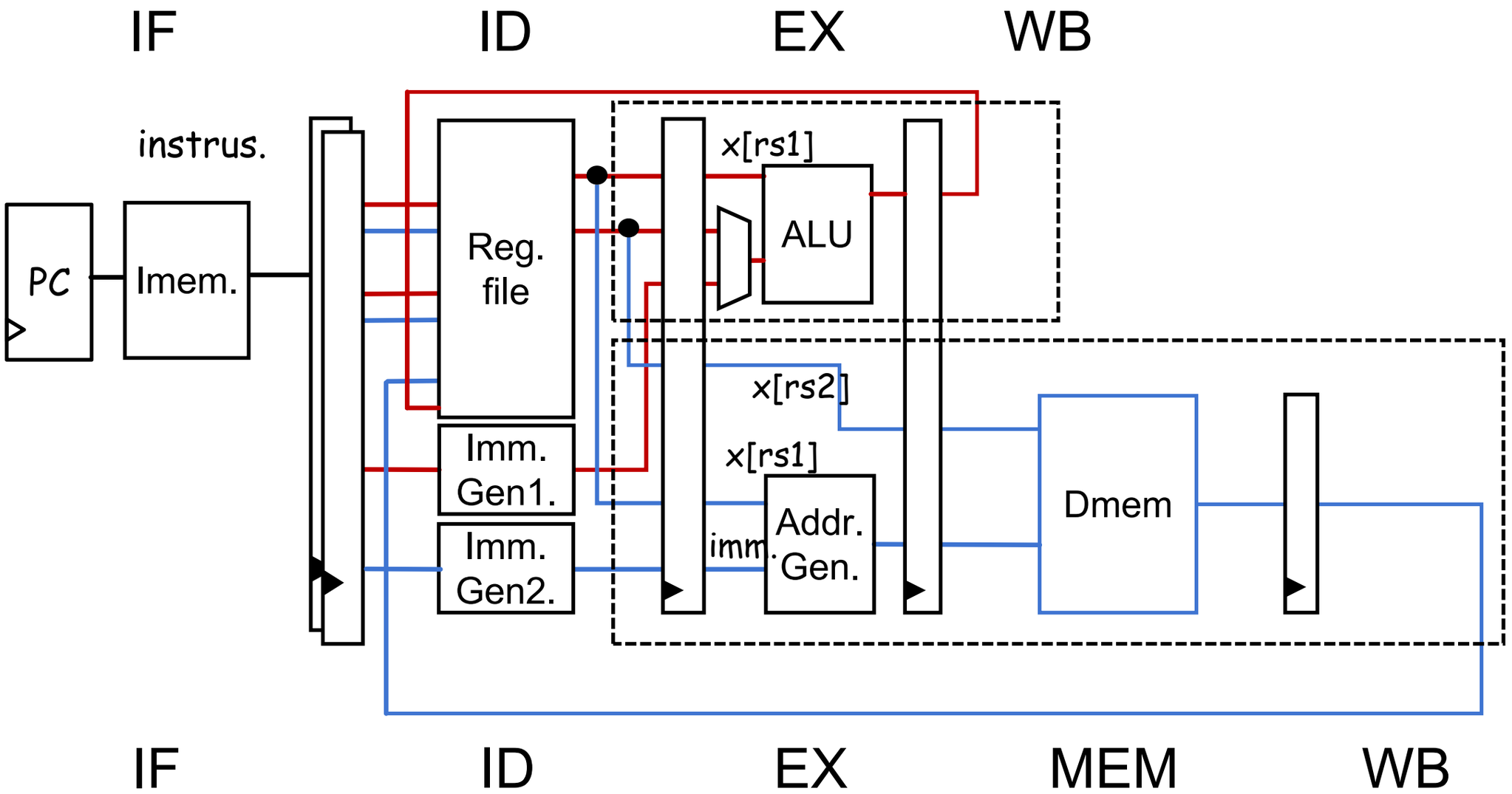
Multi-issue (Hardware)

- In practice, we build different datapaths for different types of instructions
- E.g.



Multi-issue (Hardware)

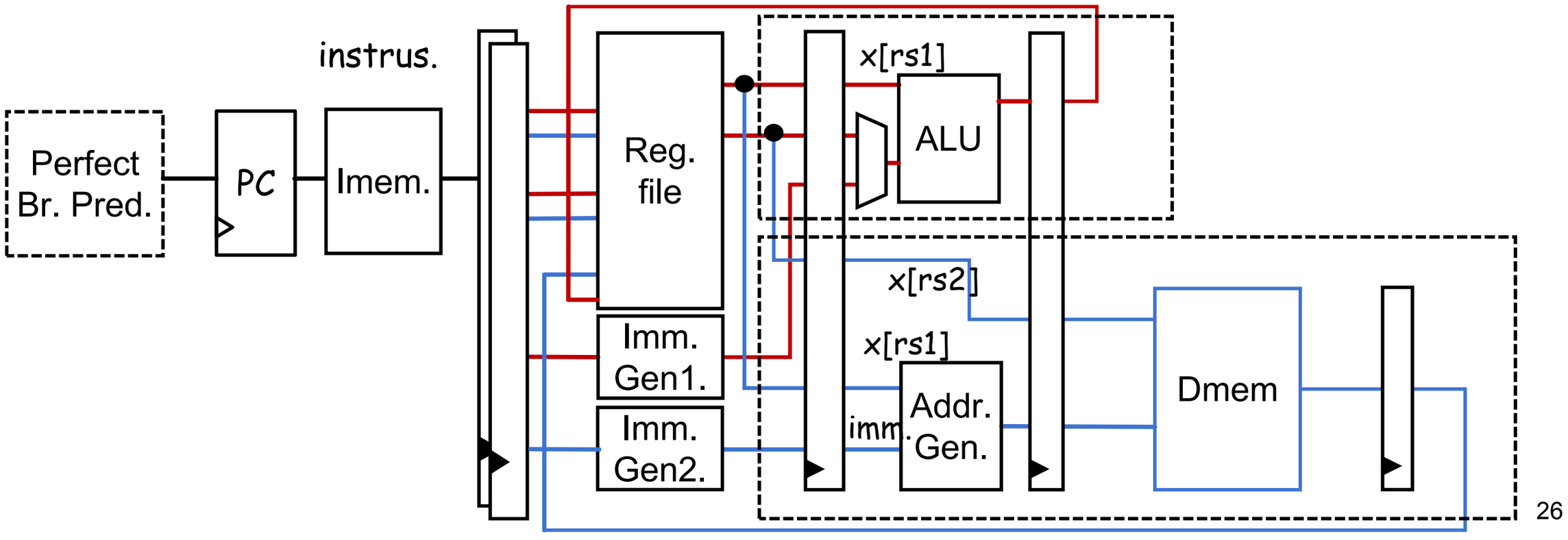
- In practice, we build different datapaths for different types of instructions
- E.g.



Multi-issue

- Ideally, issue two instructions with different type to ALU/mem datapaths

Instruciton type	cc1	cc2	cc3	cc4	cc5	cc6	cc7
ALU or branch	IF	ID	EX	WB			
Load or store	IF	ID	EX	MEM	WB		
ALU or branch		IF	ID	EX	WB		
Load or store		IF	ID	EX	MEM	WB	
ALU or branch			IF	ID	EX	WB	
Load or store			IF	ID	EX	MEM	WB



Multi-issue

```
for (int i=1000; i>0; i=i-1)
     x[i] = x[i] + s;
```

```
1.      addi    s0,x0,s         #initialize s0
2.Loop:lw       t3,0(t1)        #load array element
3.      add     t3,t3,s0        #add s to $t3
4.      sw      t3,0(t1)        #store result
5.      addi    t1,t1,-4        #t1 =t1-4
6.      bne     t1,t2,Loop      #repeat loop if t1!=t2
```

Instruciton	cc1	cc2	cc3	cc4	cc5	cc6	cc7	cc8	cc9
1.addi	IF	ID	EX	WB					
2.lw	IF	ID	EX	MEM	WB				
3.add		IF	ID	EX	WB				
4.sw		IF	ID	EX	MEM	WB			
5.addi			IF	ID	EX	WB			
nop			nop	nop	nop	nop	nop		
6.bne			IF	ID	EX	WB			
2.lw			IF	ID	EX	MEM	WB		
3.add				IF	ID	EX	WB		
4.sw				IF	ID	EX	MEM	WB	
5.addi					IF	ID	EX	WB	
nop					nop	nop	nop	nop	nop

Hazards!

Multi-issue

```
for (int i=1000; i>0; i=i-1)
     x[i] = x[i] + s;
```

```
1.      addi    s0,x0,s         #initialize s0
2.Loop:lw       t3,0(t1)        #load array element
3.      add     t3,t3,s0        #add s to $t3
4.      sw      t3,0(t1)        #store result
5.      addi    t1,t1,-4        #t1 =t1-4
6.      bne     t1,t2,Loop      #repeat loop if t1!=t2
```

Instruciton	cc1	cc2	cc3	cc4	cc5	cc6	cc7	cc8	cc9
1.addi	IF	ID	EX	WB					
2.lw	IF	ID	EX	MEM	WB				
nop		nop	nop	nop	nop				
nop		nop	nop	nop	nop	nop			
3.add			IF	ID	EX	WB			
4.sw			IF	ID	EX	MEM	WB		
5.addi				IF	ID	EX	WB		
nop				nop	nop	nop	nop	nop	
6.bne					IF	ID	EX	WB	
2.lw					IF	ID	EX	MEM	WB
… …									
… …									

Forwarding

Forwarding

Forwarding

Multi-issue

- Loop unrolling & register renaming to optimize (also will be used in SIMD)

```
1.  addi    s0,x0,s
2.L:lw      t3,0(t1)
3.  lw      t4,-4(t1)
4.  lw      t5,-8(t1)
5.  lw      t6,-12(t1)
6.  add     t3,t3,s0
7.  add     t4,t4,s0
8.  add     t5,t5,s0
9.  add     t6,t6,s0
10. sw      t3,0(t1)
11. sw      t4,-4(t1)
12. sw      t5,-8(t1)
13. sw      t6,-12(t1)
14. addi    t1,t1,-16
15. bne     t1,t2,L
```

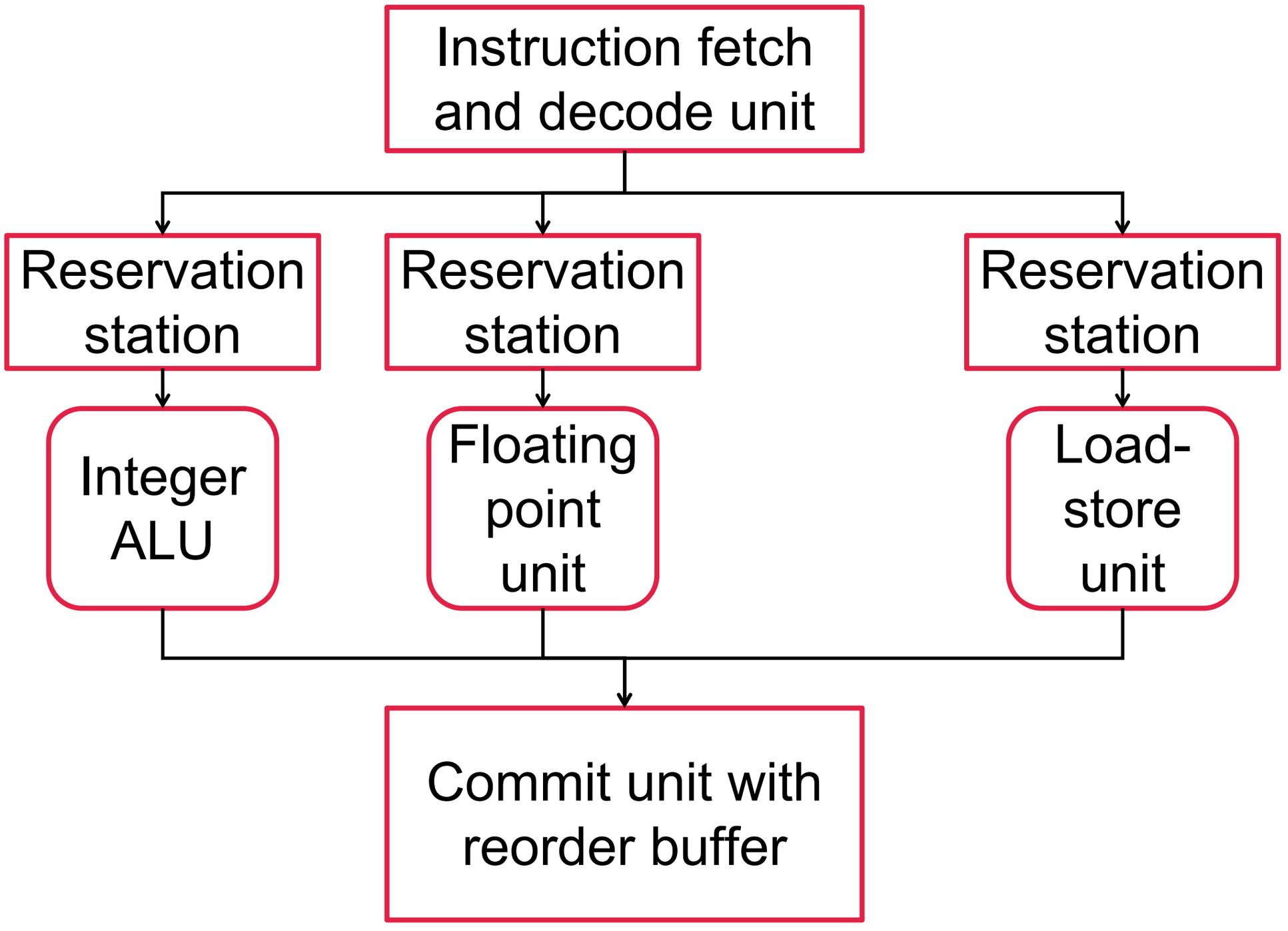
Instruciton	cc1	cc2	cc3	cc4	cc5	cc6	cc7	cc8	cc9
1.addi	IF	ID	EX	WB					
2.lw t3	IF	ID	EX	MEM	WB				
nop		nop	nop	nop	nop				
3.lw t4		IF	ID	EX	MEM	WB			
6.add t3			IF	ID	EX	WB			
4.lw t5		IF	ID	EX	MEM	WB			
7.add t4			IF	ID	EX	WB			
5.lw t6			IF	ID	EX	MEM	WB		
8.add t5				IF	ID	EX	WB		
10.sw t3				IF	ID	EX	MEM	WB	
9.add t6					IF	ID	EX	WB	
11.sw t4					IF	ID	EX	MEM	
14.addi						IF	ID	EX	
12.sw t5						IF	ID	EX	
15.bne							IF	ID	
13.sw t6							IF	ID	

Static vs. Dynamic multi-issue

- Static multi-issue
  - Package instructions into issue slots and detect hazards statically (at compile time mostly)
  - Hardware may also detect/resolve hazards
  - Also called VLIW (very long instruction word)
- Dynamic multi-issue
  - Package instructions into issue slots and detect hazards dynamically (during execution by hardware mostly)
  - Compiler may also help avoiding hazards
  - Also called superscalar

Instruciton type	cc1	cc2	cc3	cc4	cc5	cc6	cc7
ALU or branch	IF	ID	EX	WB			
Load or store	IF	ID	EX	MEM	WB		
ALU or branch		IF	ID	EX	WB		
Load or store		IF	ID	EX	MEM	WB	
ALU or branch			IF	ID	EX	WB	
Load or store			IF	ID	EX	MEM	WB

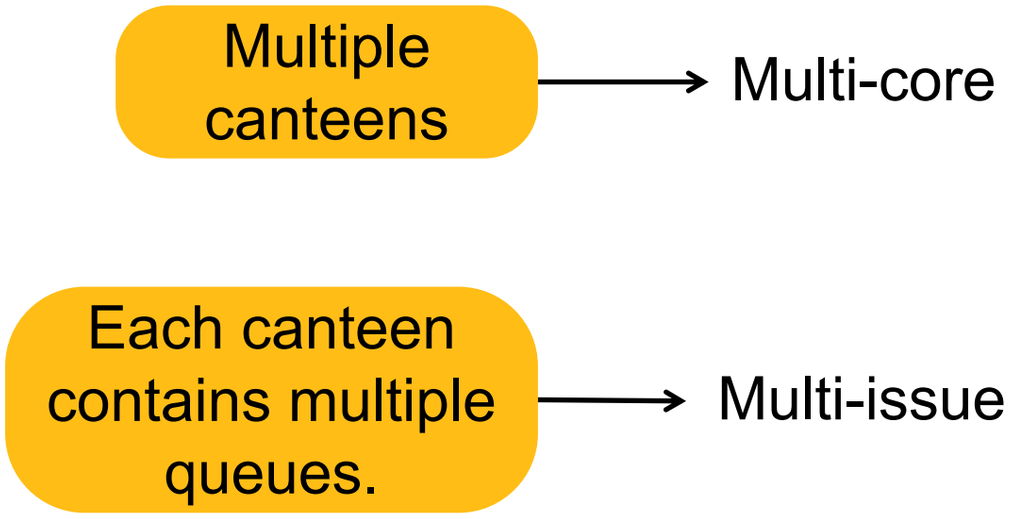
Hardware implementation of superscalar



More in CS211 CA II

Multi-issue Pitfalls

- Multi-issue is not multi-core;
- Multi-issue is not SIMD;
- Multi-issue can be combined with pipelining, SIMD, multi/hyper-threading, etc. to improve the performance of the processor;



Multi-issue Advancements-GPU

- Multi-issue can be combined with SIMD;
- In Fermi and later NVIDIA GPU architectures, the scheduler issues
  - 2 INT instructions or 2 single-point FP instructions or
  - 1 mixed INT or FPU or load or store or SFU instructions

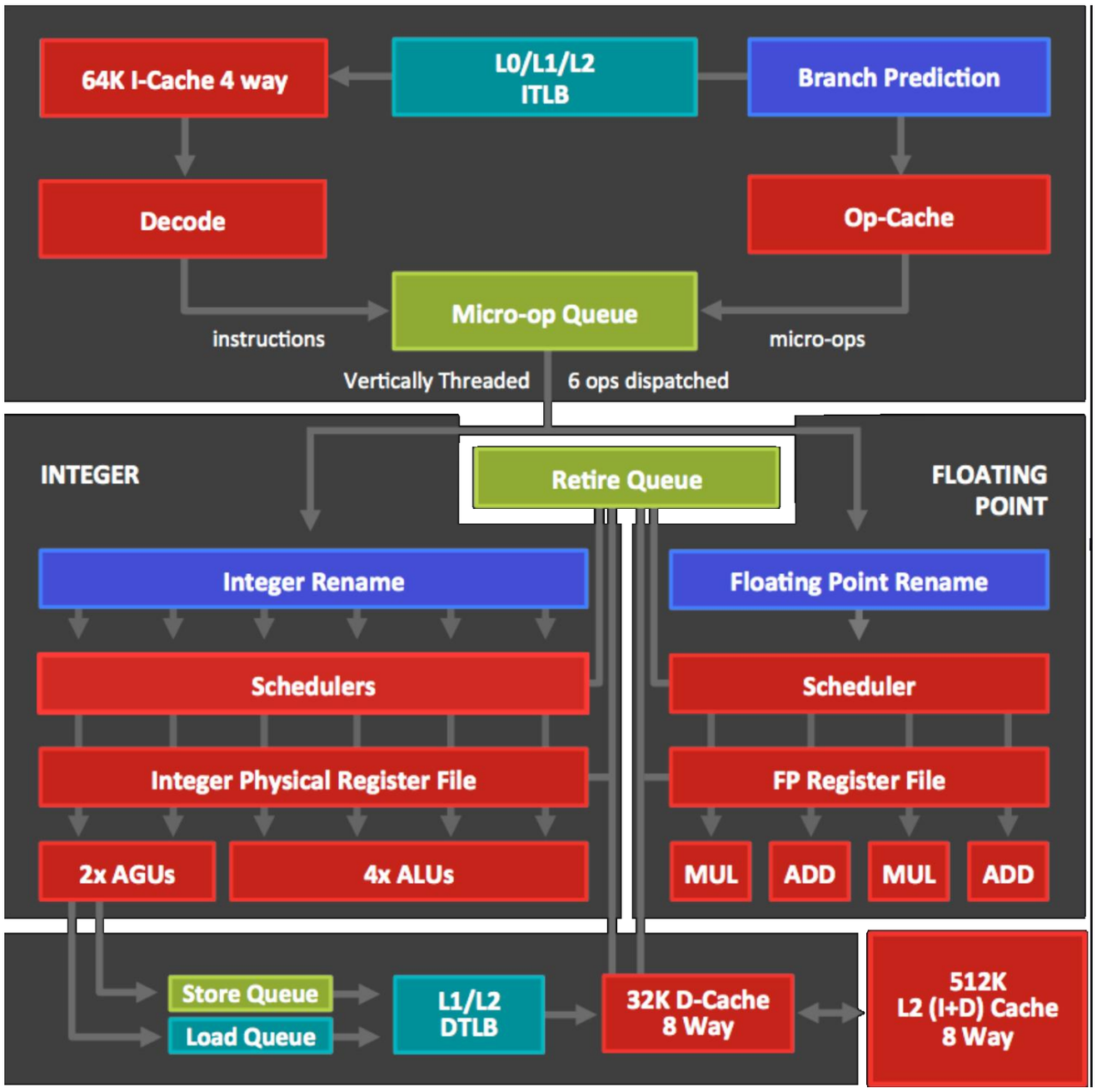
One streaming multi-processor inside an H100 GPU

Credit to Nvidia

More in EE219 AI computing systems

Multi-issue Advancements-CPU

- Multi-issue can be combined with multi/hyper-threading;
- Thread-level parallelism (TLP)
  - To tolerate latency (e.g., cache miss)
  - To further improve throughput
  - To reduce context switch penalty
- Types of Multithreading
  - Fine-grained: threads scheduled cycle by cycle
  - Corase-grained: threads scheduled on events (e.g., cache misses)/time quantum
  - SMT: simultaneous multi-threading



Credit to AMD

AMD CPU (Zen architecture) that supports simultaneous multithreading

More in CS211 CA II

Summary

- **Instruction-level parallelism**
  - Pipeline
    - Insert pipeline registers to execute the instructions stage by stage;
    - Multiple instructions co-exist in the pipeline to realize parallelism;
    - Induce (structural/data/control) hazards;
    - Strategies to deal with the hazards (insertion of bubbles, forwarding, hardware re-design, code scheduling, branch prediction, etc.);
  - Multi-issue
    - Multiple datapaths to execute multiple instructions in parallel;
    - Need to consider hazards as well;
    - Static vs. Dynamic multi-issue