TED ÜNİVERSİTESİ

CMPE361 Computer Organization

Department of Computer Engineering
TED University- Fall 2023

Pipelined MIPS processor

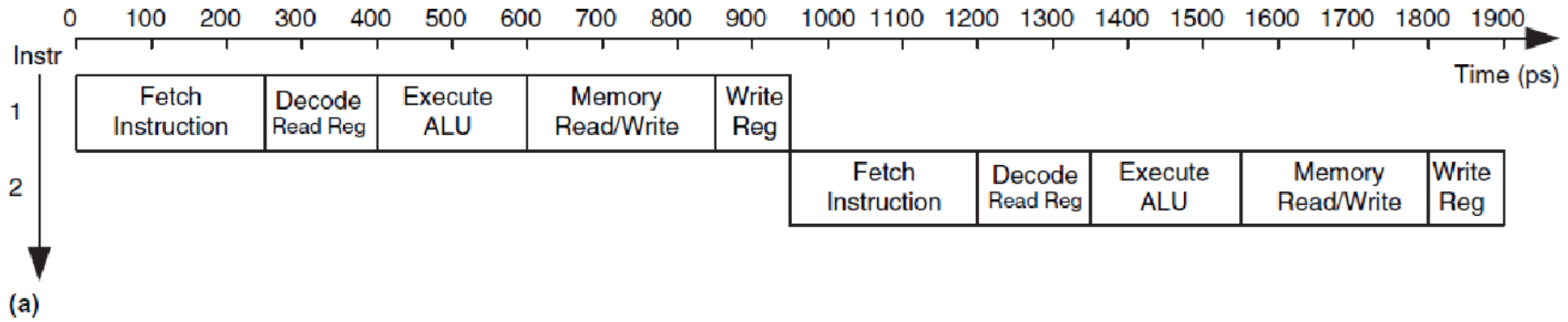
These Slides are mainly based on slides of the text book (downloadable from the book’s website).

Pipelined Processor (Includes add,sub, and or, lw and sw instructions)

Pipelined Datapath

- Pipelininig priovides parallelism in time
- Divide single-cycle processor into 5 stages:
  - Fetch
  - Decode
  - Execute
  - Memory
  - Writeback

Single Cycle with stages

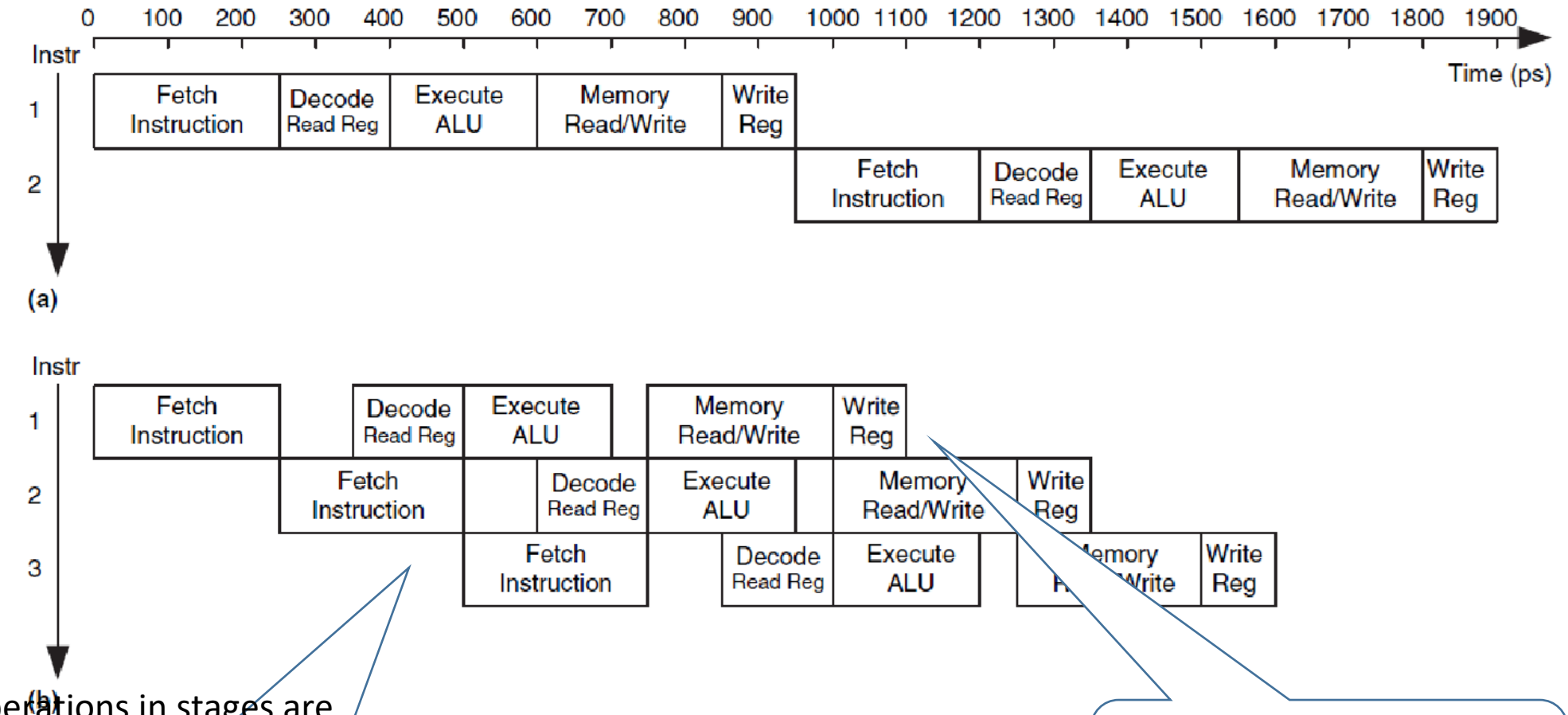


Stage Timing

- Accessing memory, register file and ALU are usually the longest in time.

Element	Parameter	Delay (ps)
Register clock-to-Q	t_{pcq_PC}	30
Register setup	t_{setup}	20
Multiplexer	t_{mux}	25
ALU	t_{ALU}	200
Memory read	t_{mem}	250
Register file read	t_{RFread}	150
Register file setup	$t_{RFsetup}$	20

Stages in Single Cycle versus Pipelined



Operations in stages are unbalanced. Yet length of each stage is equal to the longhest stage!

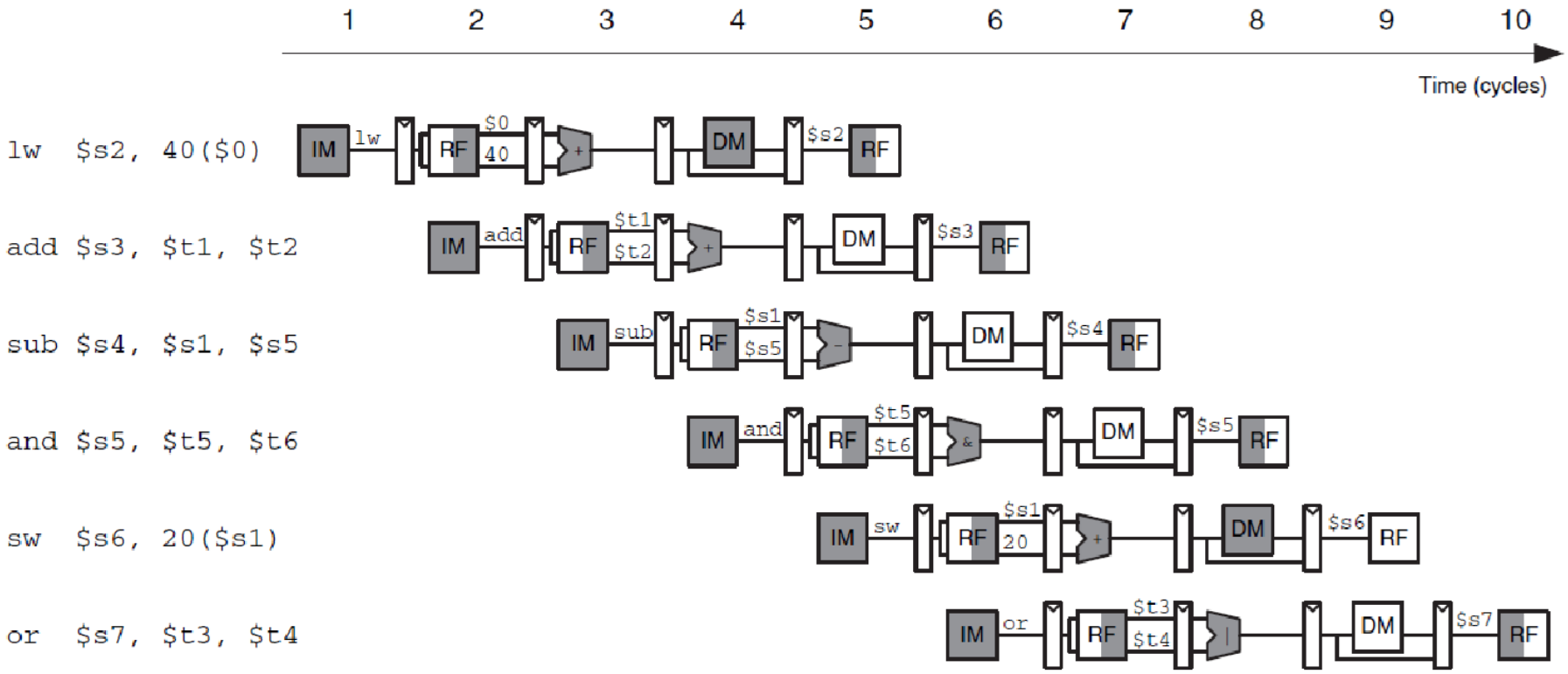
Time is wasted in Decode, Execute and WriteBack stages?

What happens in Pipelined Processor case

- Theoretically, a maximum of five instructions can execute simultaneously, one in each stage in different time slot.
- The latency of each instruction may be unchanged compared to single cycle
- Theoretically, throughput could go up as much as five times,
  - Although, in reality, this is not quite possible.

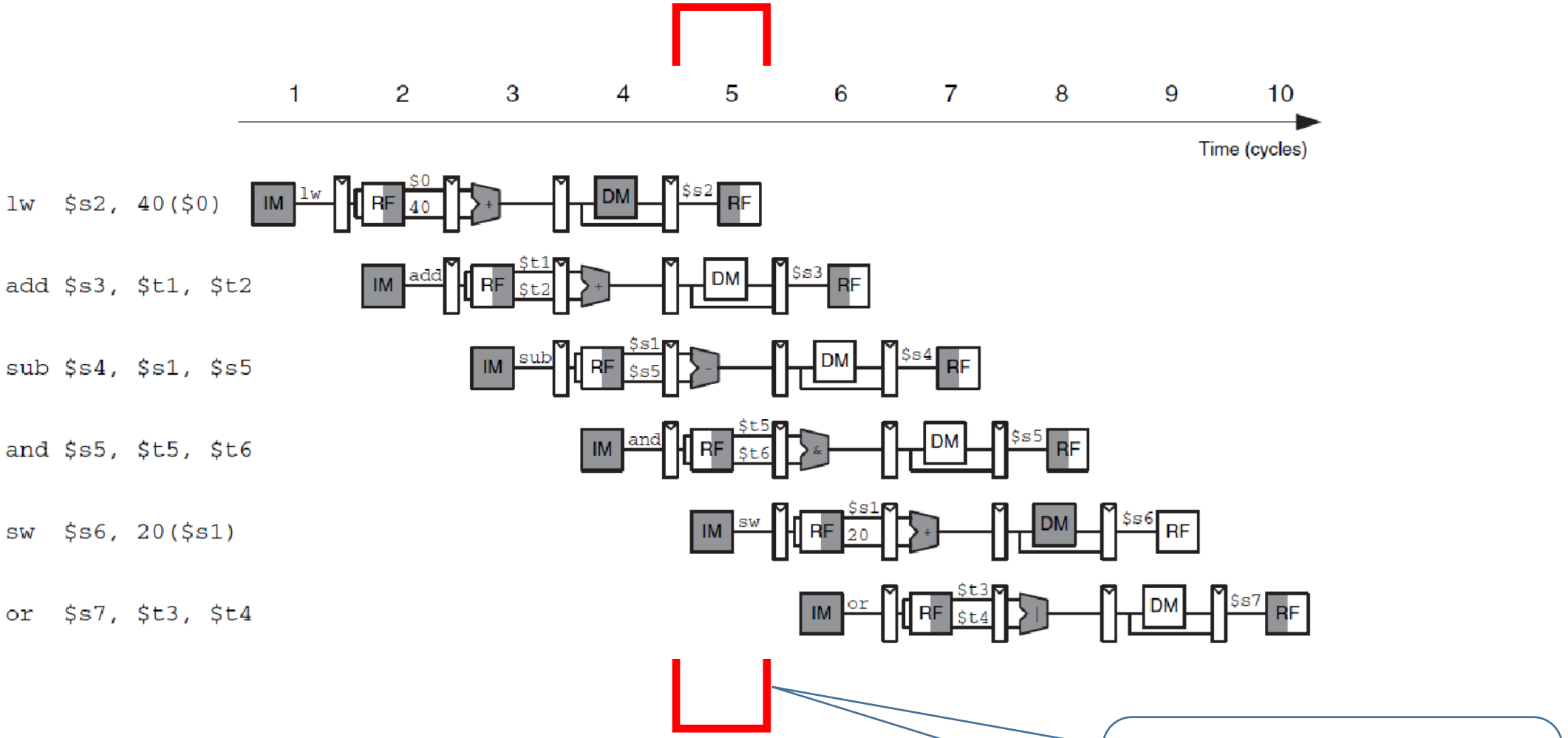
Pipelining Example 1

Reading across a row shows the clock cycles and the stage of an instruction .



Reading down a column shows stages different instructions are in, in a particular cycle.

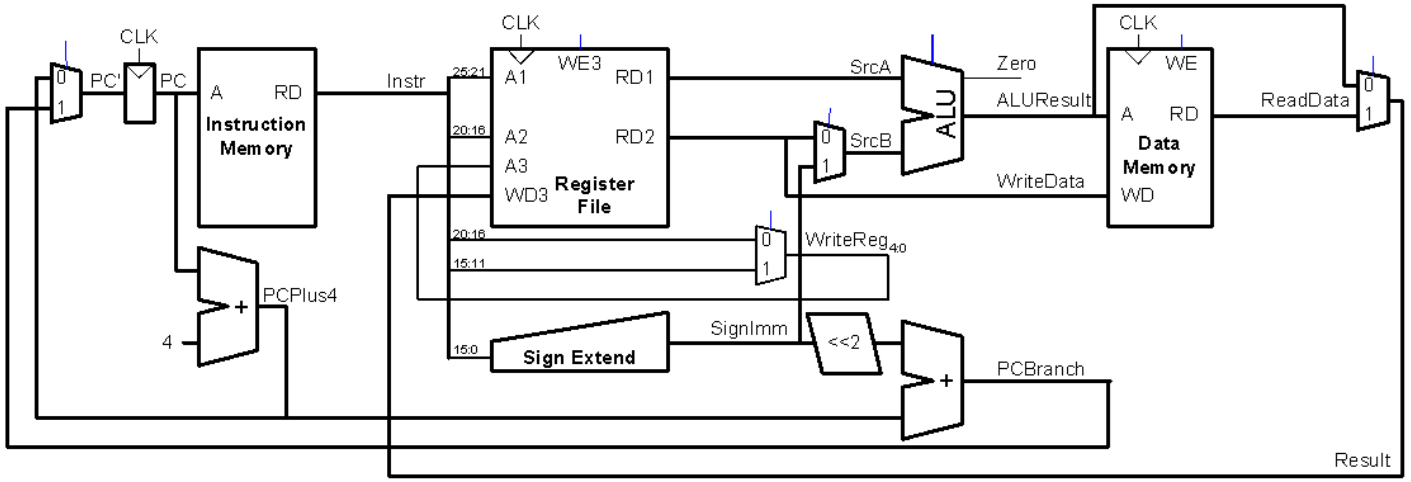
Pipelined Example 2



Signal Handling in Pipelined Datapath

- Between the stages extra registers are placed, allowing signals to be passed through the stages, synchronized with the data flow through the stages.
- Each instructions requires a different set of signals.

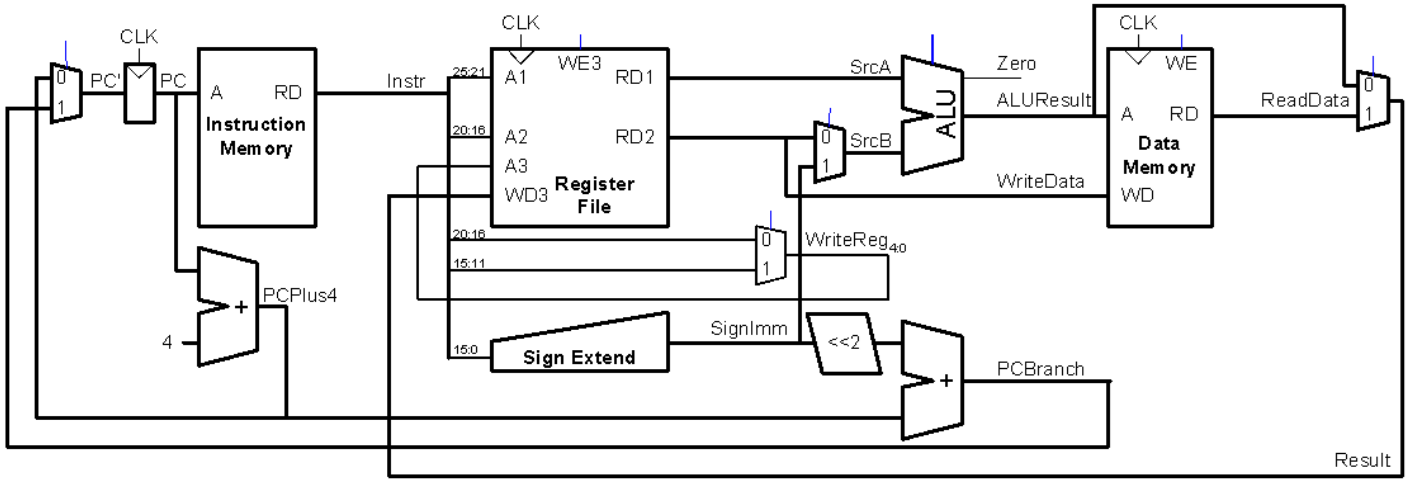
Single-Cycle processor vs pipelined



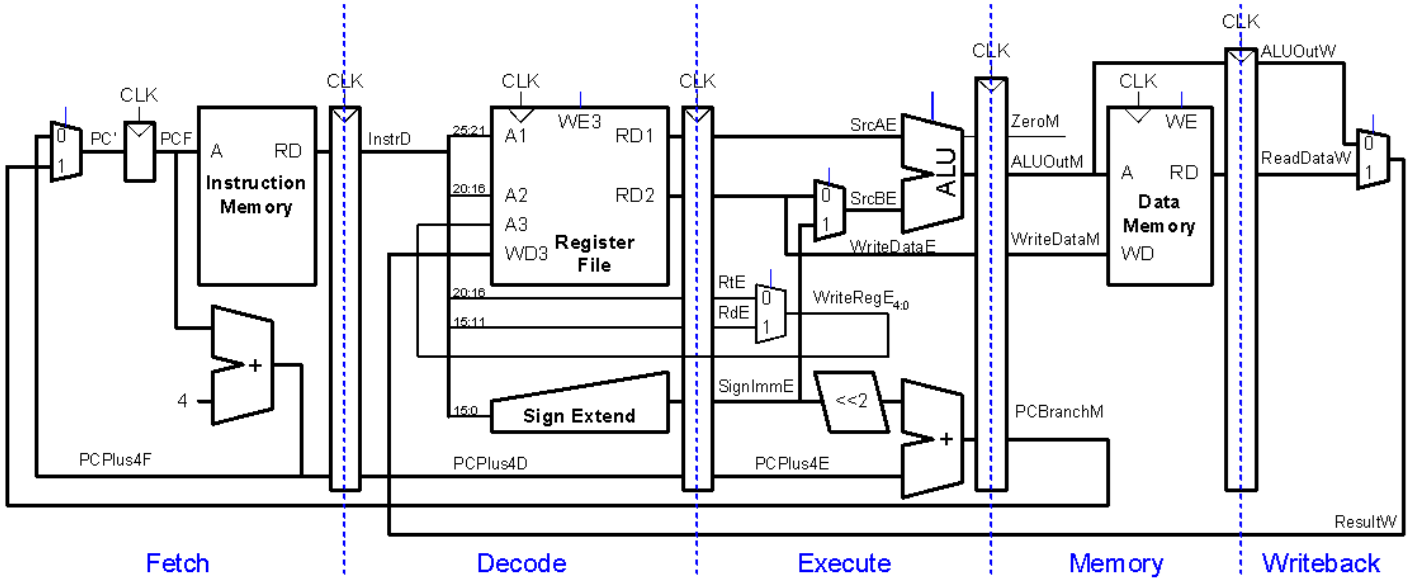
SCP

Single-Cycle vs Pipelined Datapath

SCP



PP



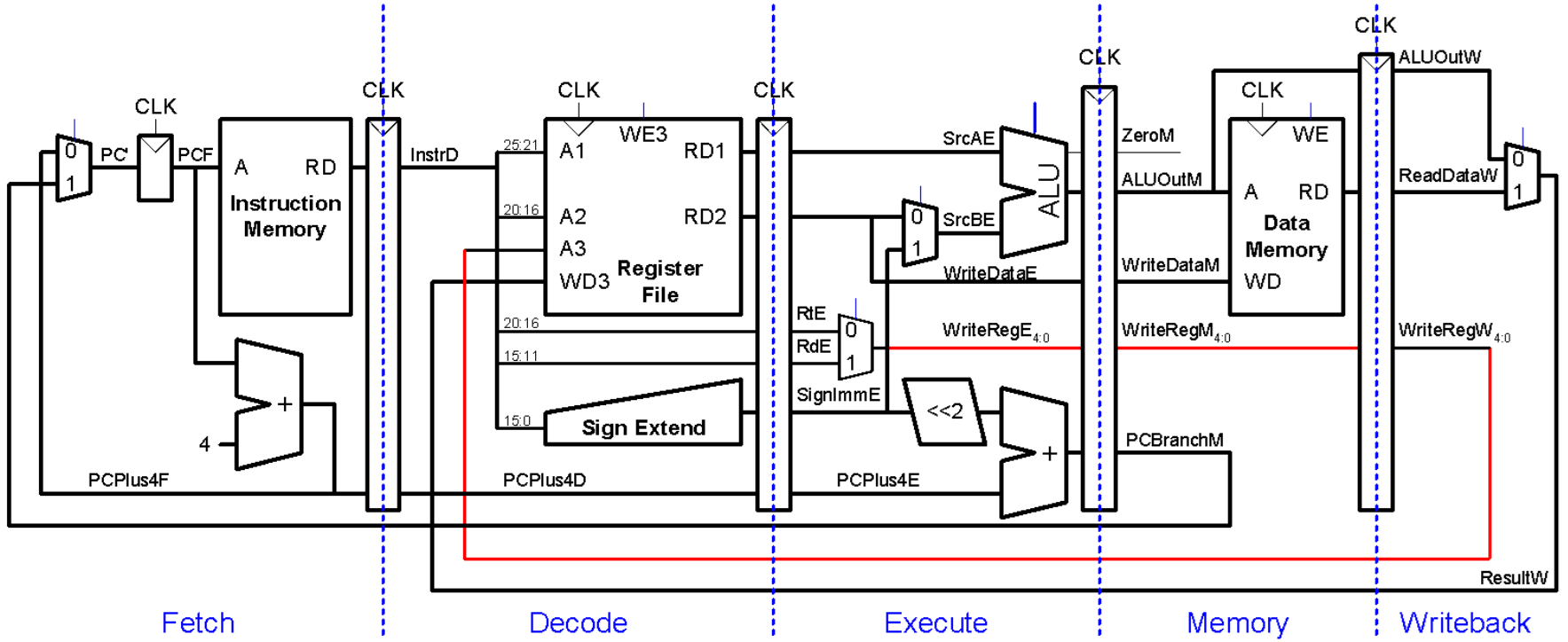
An inst in W stage can cause signal conflict with an inst in D stage, ?

Hint: All Signals must move across pipeline registers in synchronous way

Avoid incorrect register writes

- The critical issues in pipelining: The signals associated with a particular instruction must advance through the pipeline.
- Otherwise,
  - For example, the result of the lw instruction would be incorrectly written to $s4 rather than $s2, in cycle 5, as the current register comes from sub instruction; register write from lw is lost…

Corrected Pipelined Datapath for lw

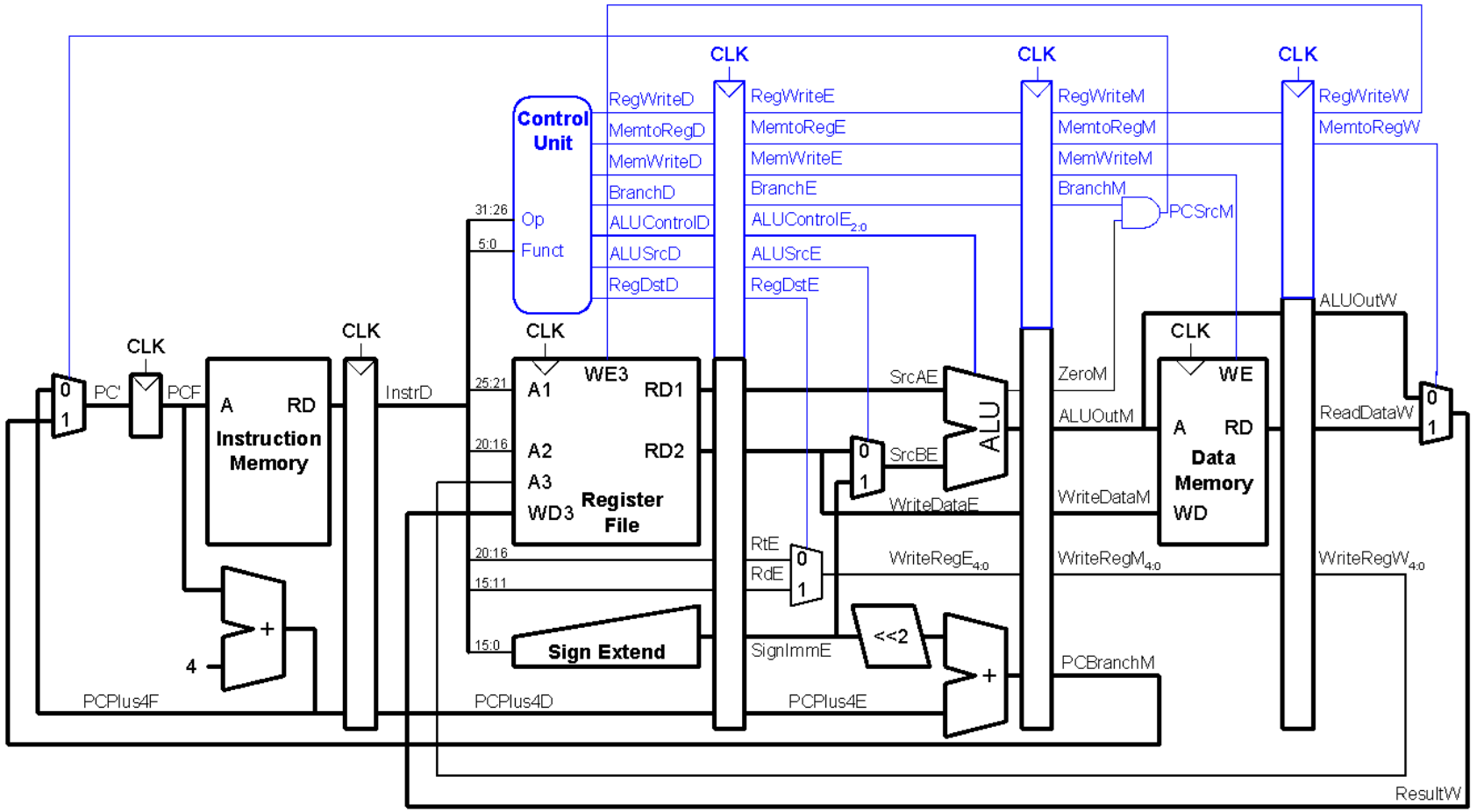


WriteReg* must arrive at same time as *Result

Correct register writes

- WriteReg signal is now pipelined along through the Memory to the Writeback stages, thus, it remains in sync with the rest of the instruction.
- WriteRegW and ResultW are fed back together to the register file in the Writeback stage.

Pipelined Processor with Control



- **Same control unit as single-cycle processor**
- **Control delayed to proper pipeline stage**

PC’ problem

- PC′ logic can also cause problem.
  - ✓ it might be updated with a Fetch or a Memory stage signal (PCPlus4F or PCBranchM) of two different instructions...
  - ✓ Proper synchronization of signals and the corresponding data is critical
  - ✓ This is known as control hazard, which will be addressed in upcoming lectures…