



جامعة المنارة الخاصة  
كلية الهندسة  
هندسة ميكاترونك

المعالجات الصغيرة ولغة التجميع  
المحاضرة الثامنة

مدرس المقرر  
د. بسام حسن

2025\_2026



## مفردات من المحاضرة الثامنة :

- MIPS Pipelining
- Pipeline Performance
- Hazards
- Structure Hazards



**1-cycle implementation** is not used .**Why?**

Because the **length of the clock cycle will always be determined** by the slowest operation (lw, sw) even if the data memory is not used.

1. دورة الساعة يجب أن يكون لها نفس الطول لكل تعليمة.

2. تتحدد هذه الدورة بالزمن الأطول الممكن في المعالج مثلاً تعليمة Load تستخدم خمس

وحدات وظيفية على الترتيب:

• ذاكرة التعليمات.

• ملف المسجلات.

• وحدة الحساب والمنطق.

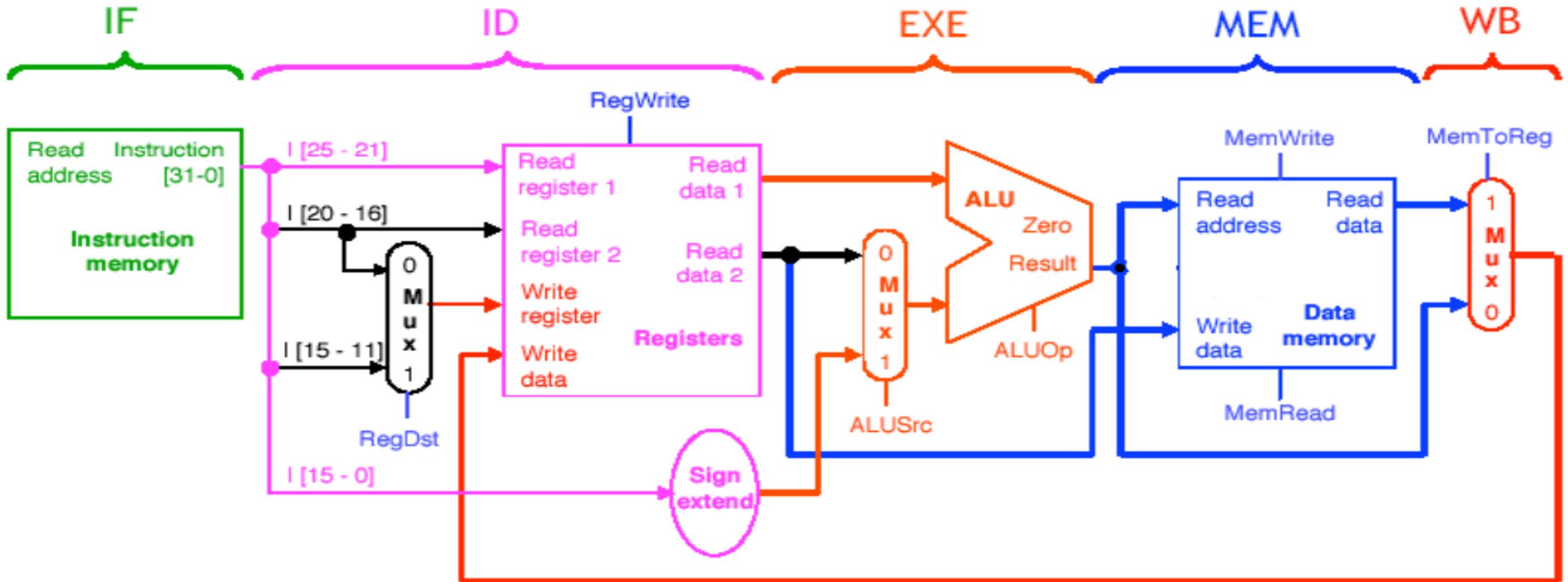
• ذاكرة المعطيات.

• ملف المسجلات.

3. باختصار: في تطبيقات الدورة الواحدة سيكون  $CPI=1$  (Clock Per Instruction)

وسيبدا الأداء العام لتطبيقات الدورة الوحيدة سيئاً لأن دورة الساعة ستكون طويلة جداً.





- ❑ Executing a MIPS instruction can take up to five steps.

Step	Name	Description
Instruction Fetch	IF	Read an instruction from memory.
Instruction Decode	ID	Read source registers and generate control signals.
Execute	EX	Compute an R-type result or a branch outcome.
Memory	MEM	Read or write the data memory.
Writeback	WB	Store a result in the destination register.

## MIPS Pipeline

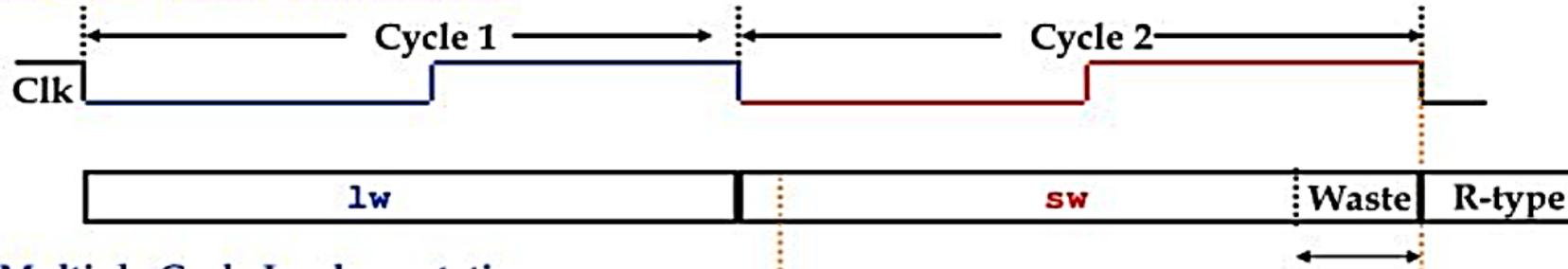
- ❑ However, as we saw, not all instructions need all five steps.

Instruction	Steps required				
beq	IF	ID	EX		
R-type	IF	ID	EX		WB
sw	IF	ID	EX	MEM	
lw	IF	ID	EX	MEM	WB

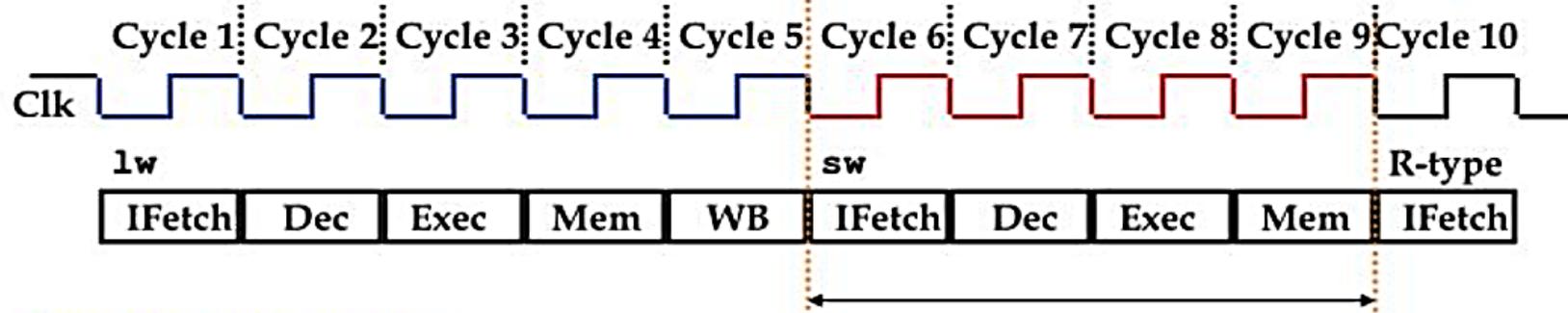


# MIPS Pipeline

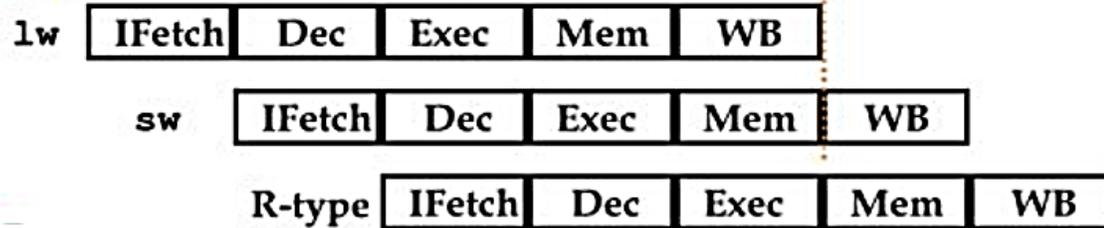
## Single Cycle Implementation:



## Multiple Cycle Implementation:



## Pipeline Implementation:



يتحدد زمن دورة الساعة في مسار التنفيذ Datapath (البسيط) وفقا لأطول مسار لازم للتعليمات  
يتحدد زمن دورة الساعة في مسار التنفيذ الأنبوبي بزمن المرحلة الأطول



## Pipeline Performance

Assume time for stages is

- 100ps for register read or write
- 200ps for other stages

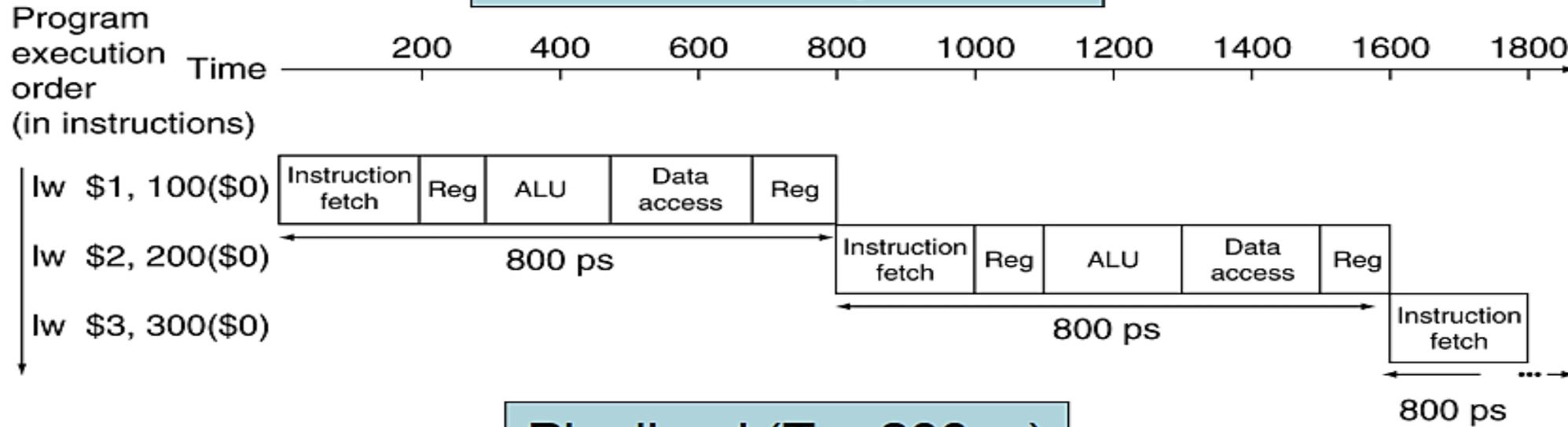
Compare pipelined datapath with single-cycle datapath

Instr	Instr fetch	Register read	ALU op	Memory access	Register write	Total time
lw	200ps	100 ps	200ps	200ps	100 ps	800ps
sw	200ps	100 ps	200ps	200ps		700ps
R-format	200ps	100 ps	200ps		100 ps	600ps
beq	200ps	100 ps	200ps			500ps

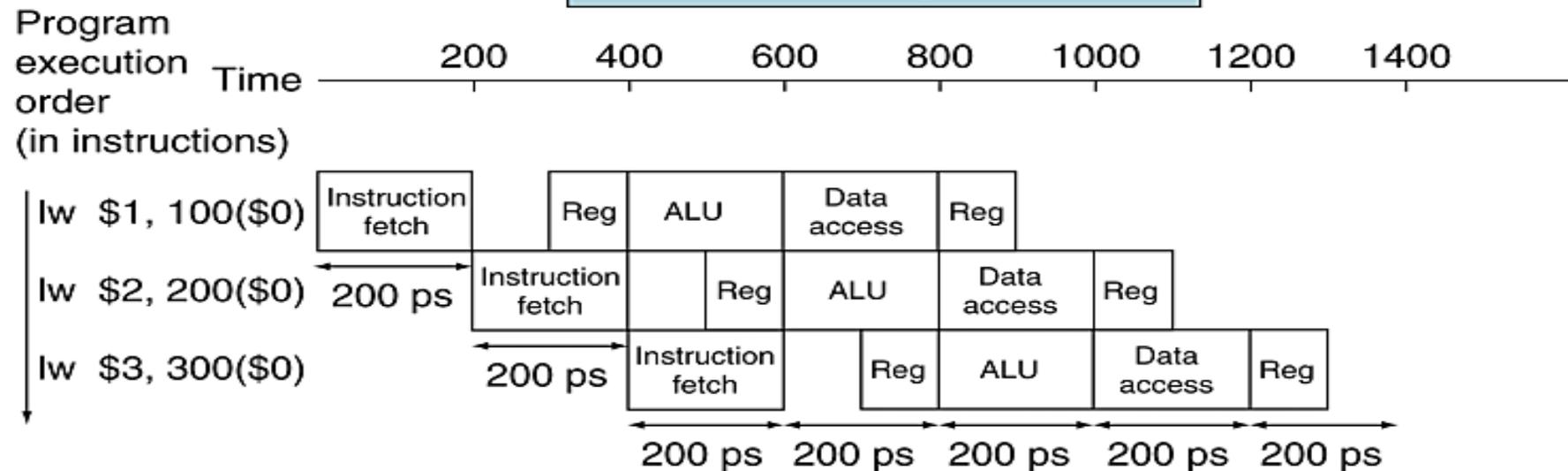




## Single-cycle ( $T_c = 800\text{ps}$ )



## Pipelined ( $T_c = 200\text{ps}$ )



### Speed-up

- Theoretical speed-up: 4 times
- Actual speed-up in example: 1.71 times
  - sequential:  $800\text{ps} + 800\text{ps} + 800\text{ps} = 2400\text{ps}$
  - pipelined:  $1000\text{ps} + 200\text{ps} + 200\text{ps} = 1400\text{ps}$
- Many tasks  $\rightarrow$  speed-up approaches theoretical limit



## Pipeline Performance

- **Example:** A program consisting of 500 instructions is executed on a 5-stage processor. How many cycles would be required to complete the program, (i) without pipelining, (ii) with pipelining? Assume *ideal* overlap in case of pipelining.

- **Solution:**

Without pipelining: Each instruction will require 5 cycles. There will be no overlap amongst successive instructions.

$$\text{Number of cycles} = 500 * 5 = 2500$$

With pipelining: Each pipeline stage will process a different instruction every cycle. First instruction will complete in 5 cycles, then one instruction will complete in every cycle, due to ideal overlap.

$$\text{Number of cycles} = 5 + ((500-1)*1) = 504$$

- **Speedup for ideal pipelining =  $2500/504 = 4.96$  (or approx. 5)**



## Hazards

- **Data hazards**
  - **Instruction depends on data** that is not ready yet  
(Data dependencies through the register file)
- **Control hazards**
  - The choice of **next instruction depends on results** that aren't ready yet  
(We'll see this next)
- **Structural hazards**
  - **Hardware** cannot support a **combination of instructions**  
(We saw this with reading and writing the register file at the same time)



# Structure Hazards



- Conflict for use of a resource
- In MIPS pipeline with a single memory
  - Load/store requires data access
  - Instruction fetch would have to *stall* for that cycle
    - Would cause a pipeline “bubble”
- Hence, pipelined datapaths require separate instruction/data memories
  - Or separate instruction/data caches

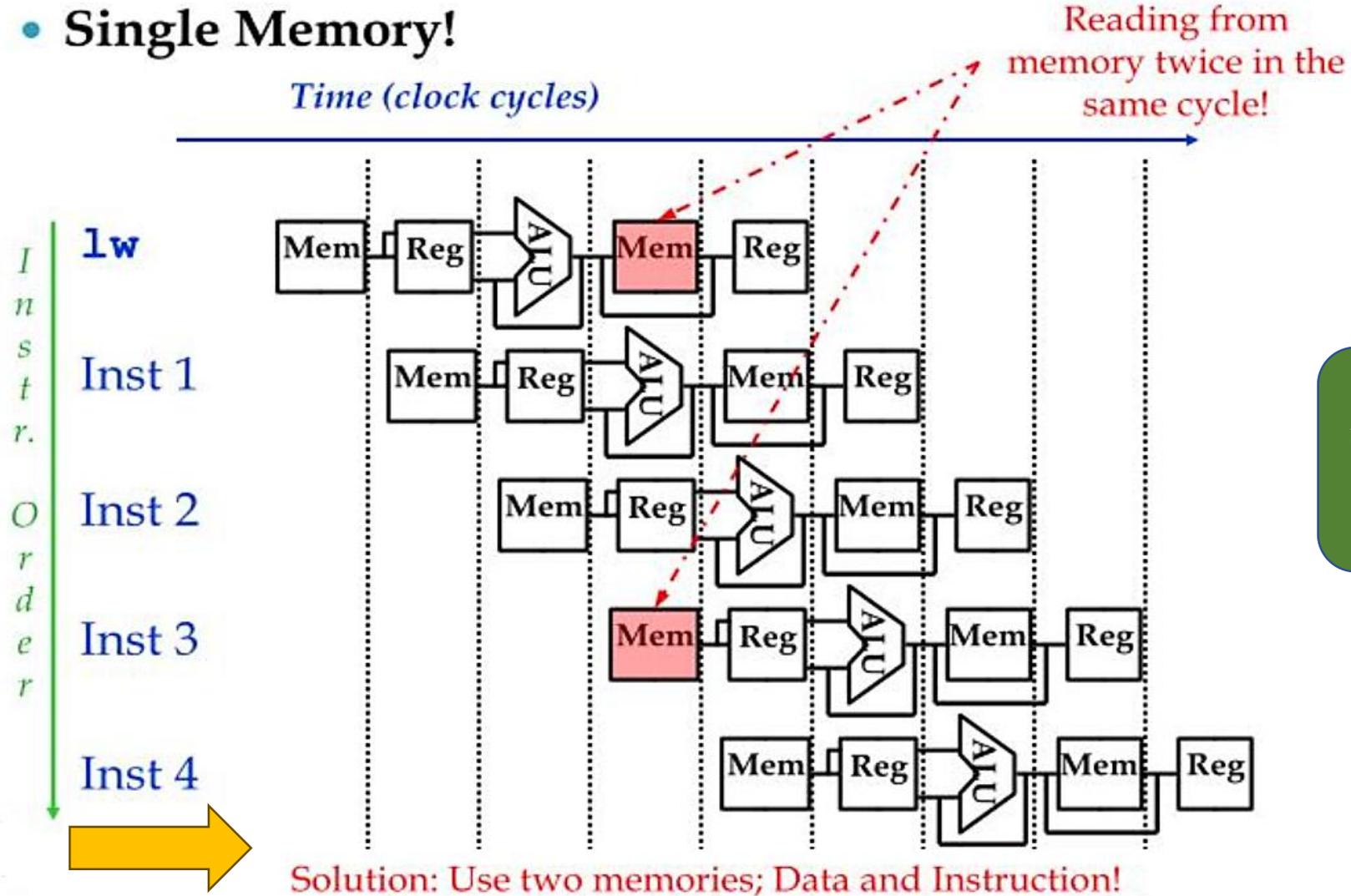




# Structure Hazards

1

- **Single Memory!**



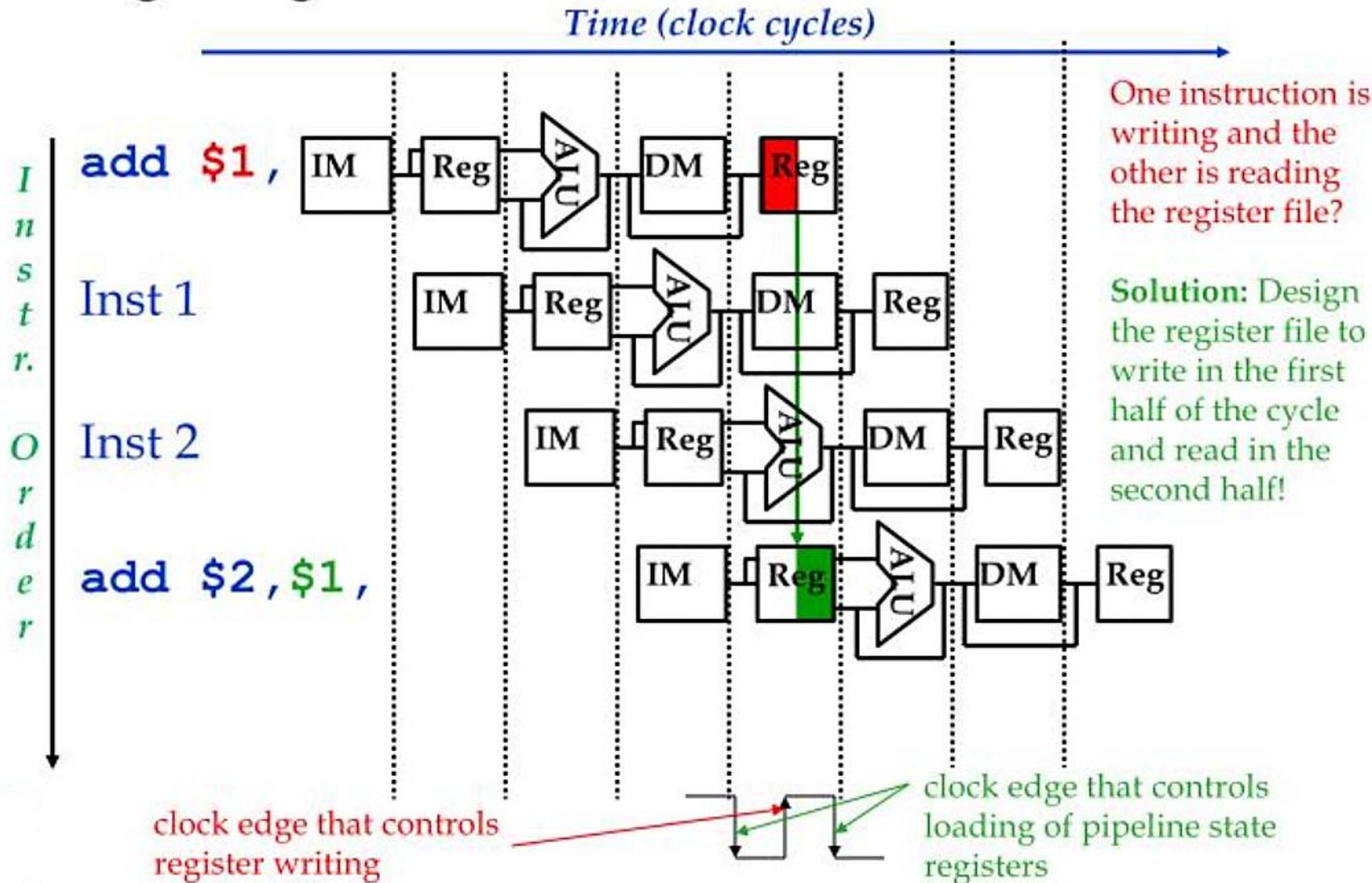
معالجنا حل هذه المشكلة بوجود ذاكرة منفصلة كل من المعطيات والتعليمات



# Structure Hazards

2

- Single Register File!



## نهاية المحاضرة الثامنة

