

Dependency and hazard

Presented By
Dr. Banchhanidhi Dash

School of Computer Engineering
KIIT University

Dependencies and Hazards

Incorrect execution due to...

... *dependencies* in program...

...and *hazards* in hardware (pipeline).

Incorrect execution above is the "fault" of the hardware...

...because the ISA does not forbid dependencies.

Dependency:

A relationship between two instructions ...

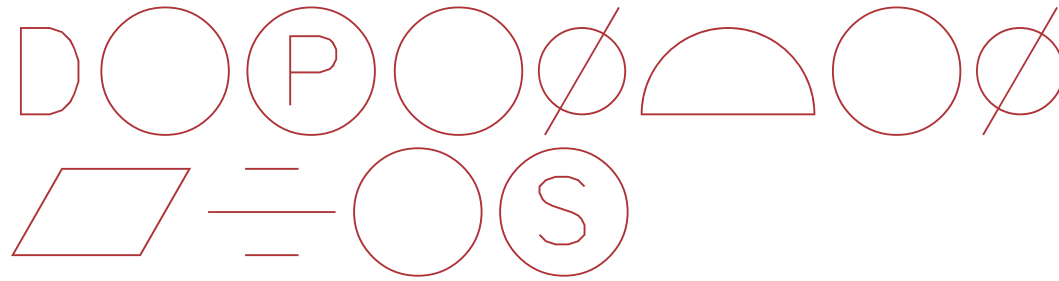
...indicating that their execution should be (or appear to be) in program order.

Hazard:

A potential execution problem in an implementation due to overlapping instruction execution.

There are several kinds of dependencies and hazards.

For each kind of dependence there is a corresponding kind of hazard.



Dependency:

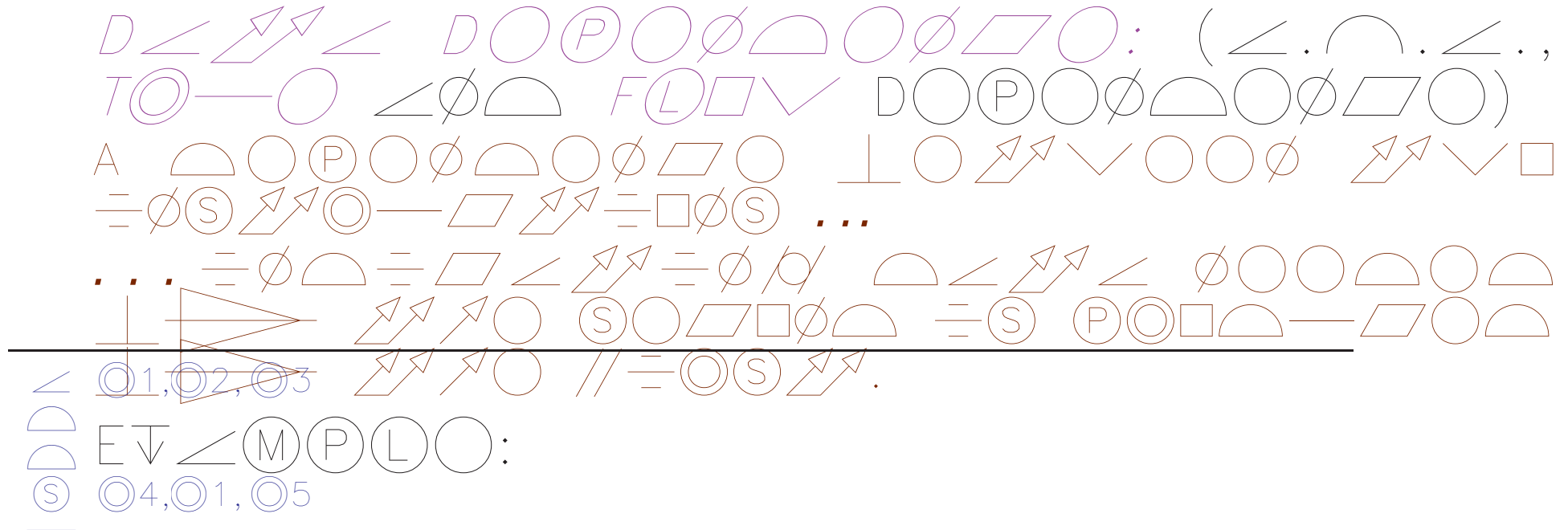
A relationship between two instructions ...

... indicating that their execution should be, or appear to be, in program order.

If B is dependent on A then B should appear to execute after A . Dependency Types:

- *True, Data, or Flow Dependence* (Three different terms used for the same concept.)
- *Name Dependence*
- *Control Dependence*

Data Dependence



The **sub** is dependent on **add**(via **r1**).

The **and** is dependent on **sub**(via **r4**). The **and** is dependent **add**(via **sub**).

Execution may be incorrect if ...

... a program having a data dependence ...

...is run on a processor having an uncorrected RAW hazard.

Name Dependencies

$T \nearrow \bigcirc \bigcirc \bigcirc \bigcirc \quad \angle \bigcirc \bigcirc \quad \nearrow \nearrow \vee \square \quad \cap \equiv \emptyset \triangle \bigcirc S:$
 $\angle \emptyset \nearrow \nearrow \equiv \triangle \bigcirc P \bigcirc \emptyset \triangle \bigcirc \emptyset \square \bigcirc \quad \angle \emptyset \triangle$
 $\square - \nearrow \nearrow P - \nearrow \nearrow \triangle \bigcirc P \bigcirc \emptyset \triangle \bigcirc \emptyset \square \bigcirc.$

$A \emptyset \nearrow \nearrow \equiv \triangle \bigcirc P \bigcirc \emptyset \triangle \bigcirc \emptyset \square \bigcirc:$


$A \triangle \bigcirc P \bigcirc \emptyset \triangle \bigcirc \emptyset \square \bigcirc \quad \perp \bigcirc \nearrow \nearrow \vee \bigcirc \bigcirc \emptyset \quad \nearrow \nearrow \vee \square$
 $\equiv \emptyset S \nearrow \nearrow \bigcirc - \square \nearrow \nearrow \equiv \square \emptyset S \quad \dots$

$\dots \equiv \emptyset \triangle \equiv \square \angle \nearrow \nearrow \equiv \emptyset \emptyset \quad \angle \square \angle L - \bigcirc \quad \vee \bigcirc \equiv \nearrow \nearrow \nearrow \nearrow \bigcirc \emptyset$
 $\perp \triangle \nearrow \nearrow \nearrow \bigcirc \quad S \bigcirc \square \square \emptyset \triangle \quad \dots$

$\dots \nearrow \nearrow \angle \nearrow \nearrow \nearrow \nearrow \bigcirc \quad // \equiv \bigcirc S \nearrow \nearrow \equiv \emptyset S \nearrow \nearrow \bigcirc - \square \nearrow \nearrow \equiv \square \emptyset$
 $\bigcirc \bigcirc \angle \triangle \bigcirc S.$

$A \emptyset \nearrow \nearrow \equiv \triangle \bigcirc P \bigcirc \emptyset \triangle \bigcirc \emptyset \square \bigcirc \quad E \nabla \angle M P L \bigcirc$
 $\angle \triangle \triangle \triangle \quad \bigcirc 1, \quad \bigcirc 2, \quad \bigcirc 3$
 $S - \perp \quad \bigcirc 2, \quad \bigcirc 4, \quad \bigcirc 5$

$S - \perp \quad \equiv S \quad \angle \emptyset \nearrow \nearrow \equiv \triangle \bigcirc P \bigcirc \emptyset \triangle \bigcirc \emptyset \nearrow \nearrow \square \emptyset \nearrow \nearrow \nearrow \bigcirc$
 $\angle \triangle \triangle \triangle. \quad E \nabla \bigcirc \square - \nearrow \nearrow \equiv \square \emptyset \quad M \angle \triangle \triangle \triangle \quad \perp \bigcirc$
 $\equiv \emptyset \square \square \bigcirc \bigcirc \bigcirc \square \nearrow \nearrow \equiv // \quad \dots$
 $\dots \angle \quad P \bigcirc \square \emptyset \bigcirc \angle M \nearrow \angle \perp \equiv \emptyset \emptyset \quad \angle \emptyset$
 $\angle \emptyset \nearrow \nearrow \equiv \triangle \bigcirc P \bigcirc \emptyset \triangle \bigcirc \emptyset \square \bigcirc \quad \dots$

A 

... \equiv  \equiv  \angle  \equiv     \angle 

Diagrammatic equation for the associativity of the multiplication map. The left side shows a vertical line with a square box, followed by two parallel diagonal lines with arrows pointing up-right, and then a single diagonal line with an arrow pointing up-right. This is followed by an equals sign and a diagram consisting of a circle with a diagonal slash, a circle containing the letter 'S', two parallel diagonal lines with arrows pointing up-right, and a circle with two concentric circles. The right side of the equation shows a horizontal line with a parallelogram box, followed by two parallel diagonal lines with arrows pointing up-right, and then a square box followed by a circle with a diagonal slash and a circle containing the letter 'S'.

[illegible]

ET \angle M P L \bigcirc

Diagram illustrating the components of the construction: a triangle, two semicircles, and three circles labeled 1, 2, and 3.



The diagram illustrates three basic circuit connection types:

- T-junction:** A vertical line labeled 'T' meets a diagonal line with an arrow pointing towards a circle.
- Series connection:** A circle labeled 'S' is connected to a horizontal line, which then meets a vertical line.
- Parallel connection:** A circle labeled 'S' is connected to a horizontal line that branches into two parallel horizontal lines.

control Dependency

An instruction B has a control dependency on a preceding instruction A if the outcome of A determines whether B should be executed or not.

Example:

A: BEQ \$*s*1, \$*s*2,Label

B: add \$*s*3,\$*s*4,\$*s*5

C: Label: sub \$*t*0,\$*t*1,\$*t*2

Pipeline Hazards

What is a pipeline hazard?

A situation that prevents an instruction from executing during its designated clock cycles.

There are 3 classes of hazards:

Structural Hazards

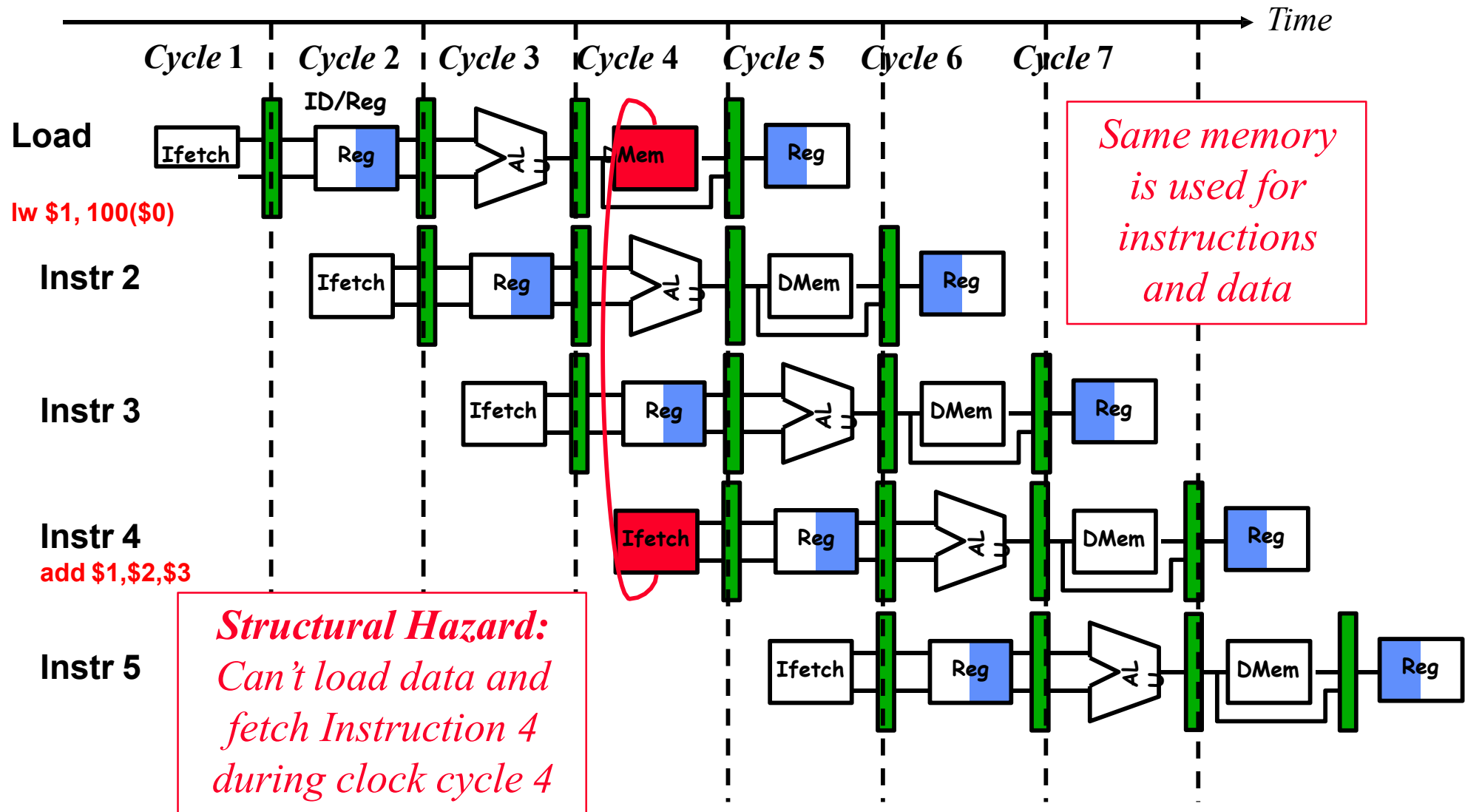
data hazard

Control Hazards

Pipeline Hazards

- Hazards can result in incorrect operations:
 - Structural hazards: Two instructions requiring the same hardware unit at same time.
 - Data hazards: Instruction depends on result of a prior instruction that is still in pipeline
 - Data dependency
 - Control hazards: Caused by delay in decisions about changes in control flow (branches and jumps).
 - Control dependency

Structural Hazard - Conflict due to Memory Access



Resolving structural hazards

❖ Problem

- ★ Attempt to use the same hardware resource (Memory) by ~~to~~ different instructions during the same cycle

❖ Solution 1: Wait

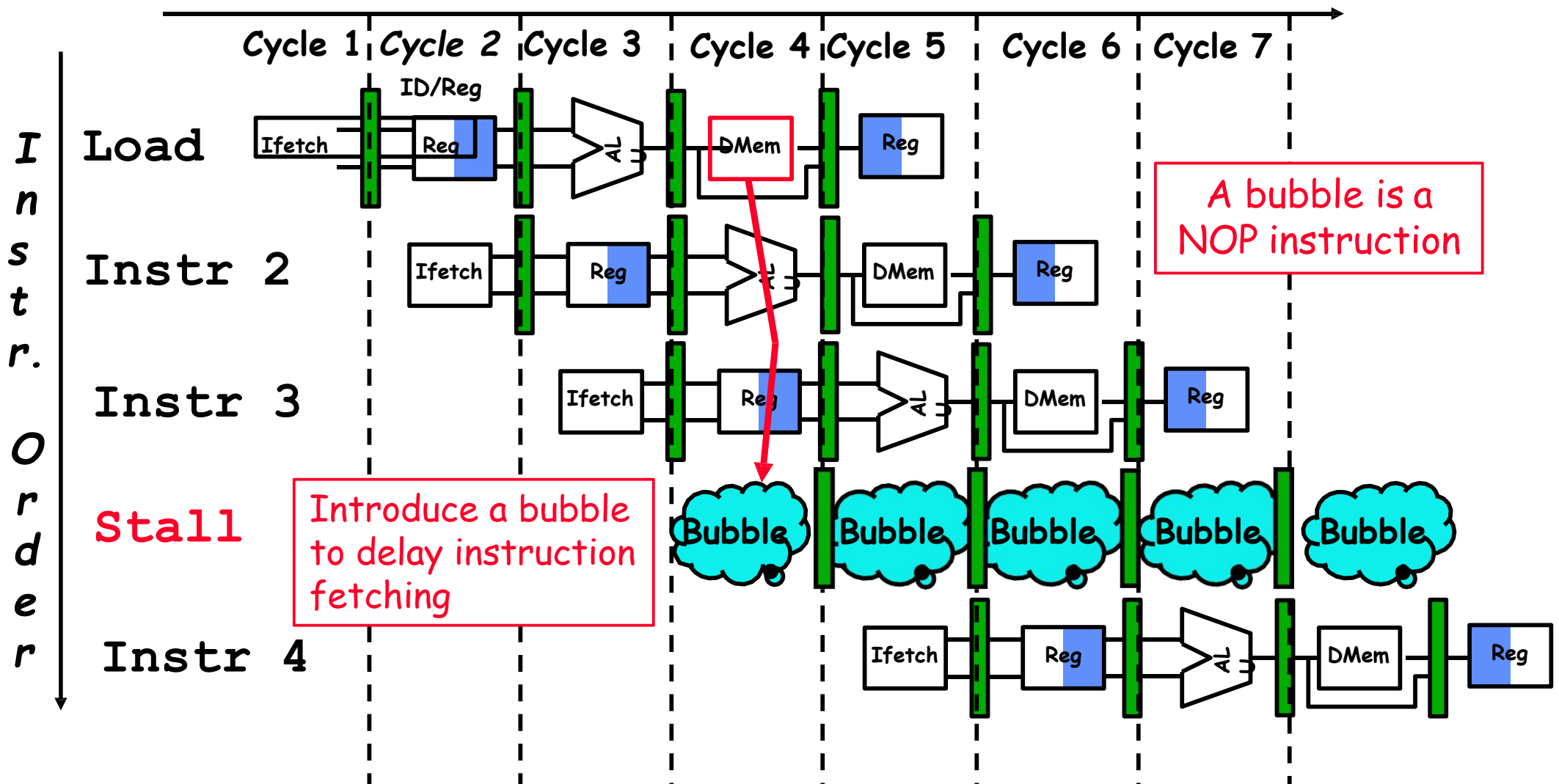
- ★ Must detect the hazard
- ★ Must have mechanism to delay (stall) instruction access to resource (Introduce bubble / NOP)
- ★ Serious: hazard cannot be ignored

❖ Solution 2: Redesign the pipeline

- ★ Add more hardware to eliminate the structural hazard
 - ★ In our example: use two memories with two memory ports
 - ✧ Instruction Memory
 - ✧ Data Memory
- } Can be implemented as caches

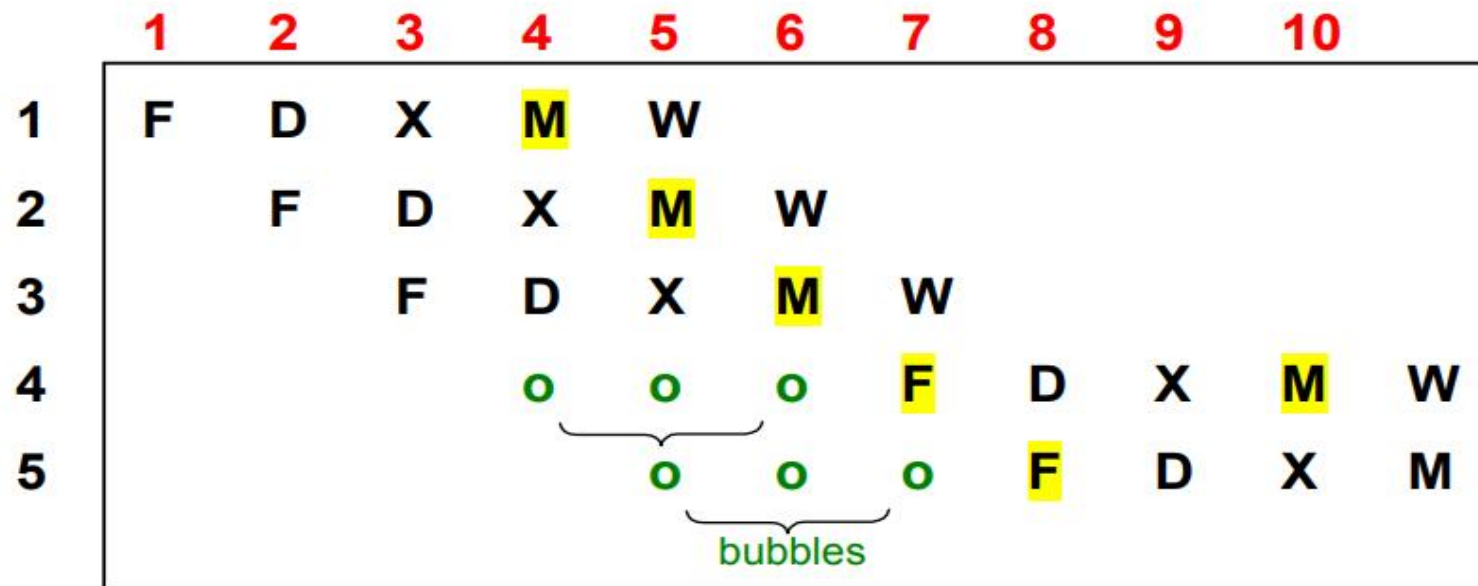
Solution 1 : Detect Structural Hazard and Delay

Time (clock cycles)



Pipeline Hazards

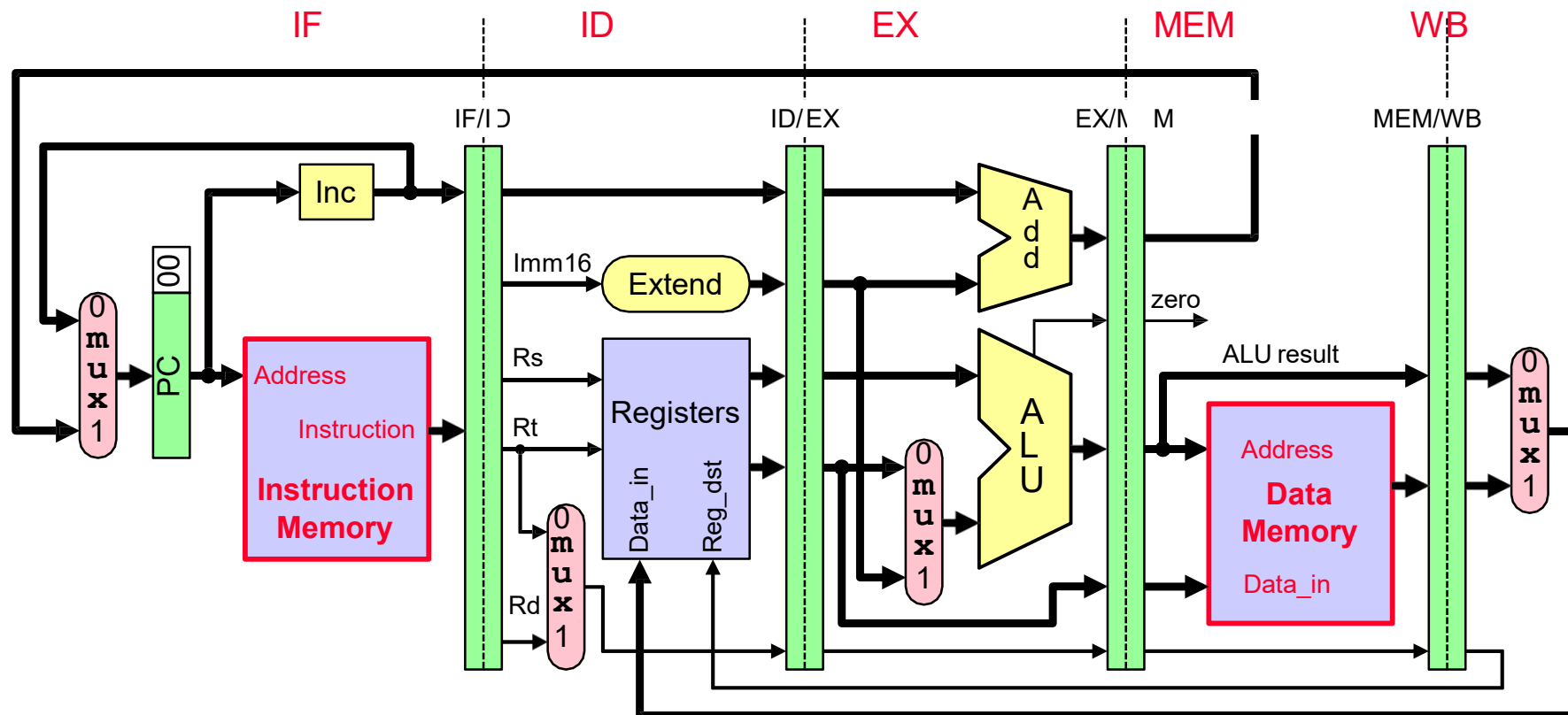
Can we fetch an instruction (F) and read data from the memory at the same time? For a single port memory, the answer is NO.



These bubbles (stall cycles) will appear if we use a single-port memory to store both instructions and data.

Solution 2: Add More Hardware (Use Instruction and data memory)

- ❖ Eliminate structural hazard at design time
- ❖ Use **two separate memories** with **two memory ports**
 - ★ Instruction and data memories can be implemented as caches



Pipeline Hazards

Data Hazards

Occur when an instruction under execution depends

on: Data from an instruction ahead in pipeline.

Example:

$A=B+C$; $D=A+E$;

$A=B+C$;

IF	ID	EXE	MEM	WB
----	----	-----	-----	----

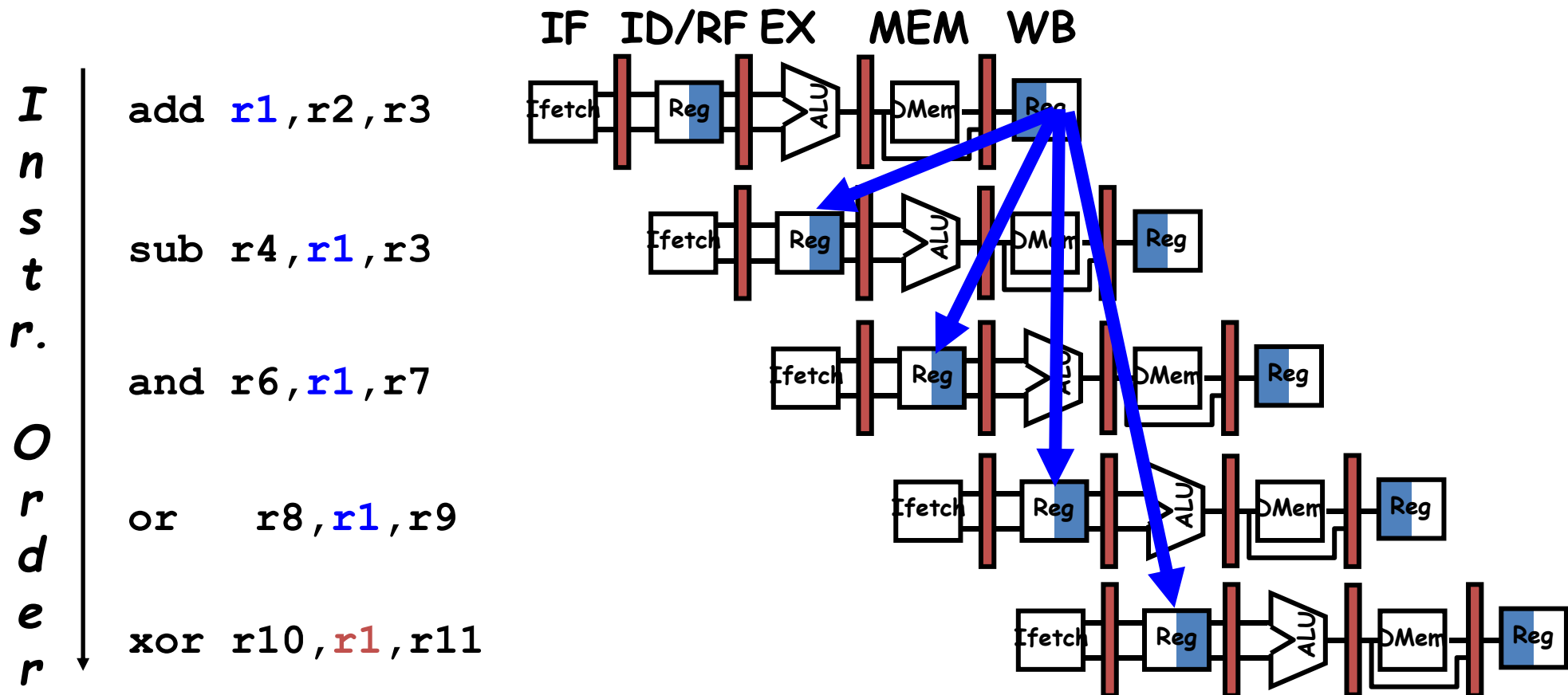
$D=A+E$;

IF	ID	EXE	MEM	WB
----	----	-----	-----	----

Dependent instruction uses old data: Results in wrong computations

Data Hazards

Time (clock cycles)



The use of the result of the ADD instruction in the next three instructions causes a hazard, since the register is not written until after those instructions read it.

Data hazards

Data hazards occur when instructions that exhibit data dependence modify data in different stages of a pipeline. There are three situations in which a data hazard can occur:

- read after write (RAW), a true dependency
- write after read (WAR), an anti-dependency
- write after write (WAW), an output dependency

Read after read (RAR) is not a hazard case.

Execution Order is:

Instr_i

Instr_j

Read After Write (RAW)

Instr_j tries to read operand before Instr_i writes it

 I: add **r1**, r2, r3
J: sub r4, **r1**, r3

Execution Order is:

Instr_i

Instr_j

Write After Read (WAR)

Instr_j tries to write operand before Instr_i reads i

- Gets wrong operand

```
I:  sub  r4, r1, r3
J:  add  r1, r2, r3
K:  mul  r6, r1, r7
```

- Called an “**anti-dependence**” by compiler writers. This results from reuse of the name “**r1**”.
- Can’t happen in MIPS 5 stage pipeline because:
 - All instructions take 5 stages, and
 - Reads are always in stage 2, and
 - Writes are always in stage 5

Execution Order is:

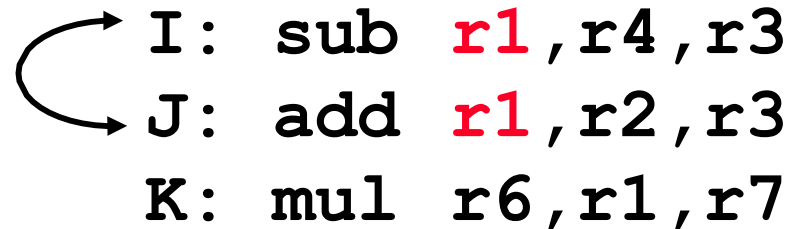
Instr_i

Instr_j

Write After Write (WAW)

Instr_j tries to write operand before Instr_i writes it

- Leaves wrong result (Instr_i not Instr_j)



I: sub **r1**, r4, r3
J: add **r1**, r2, r3
K: mul r6, r1, r7

- Called an “**output dependence**” by compiler writers
This also results from the reuse of name “**r1**”.
- Can't happen in MIPS 5 stage pipeline because:
 - All instructions take 5 stages, and
 - Writes are always in stage 5
-

Q.Consider the following MIPS assembly code:

LW R1, 10(R2)

ADD R7, R1, R5

SUB R8, R7, R6

MUL R6, R4, R8

Identify each dependency by type and list the two instructions involved.

Solution

LW& ADD
ADD & SUB
SUB & MUL
SUB & MUL

(R1- True),
(R7 - True),
(R8 - True),
(R6- Anti)

Q.Consider the following MIPS assembly code:

LW R4, 10(R2)

LW R5, 0(R4)

MUL R7, R5, R4

DIV R3, R7, R6

Identify each data hazard by type and list the two instructions involved in it.

RAW :- 1st LW - 2nd LW (due to R4)

RAW :- 1st LW - MUL (due to R4)

RAW :- 2nd LW - MUL (due to R5)

RAW :- MUL - DIV (due to R7)