Chapter 06: Instruction Pipelining and Parallel Processing

Lesson 04:

Control Hazards and Branches

Objective

- Understand control hazards during load and branching in instruction-pipeline
- Learn ways of overcoming these in the processor

Control Hazard due to load

Control Hazard During Load

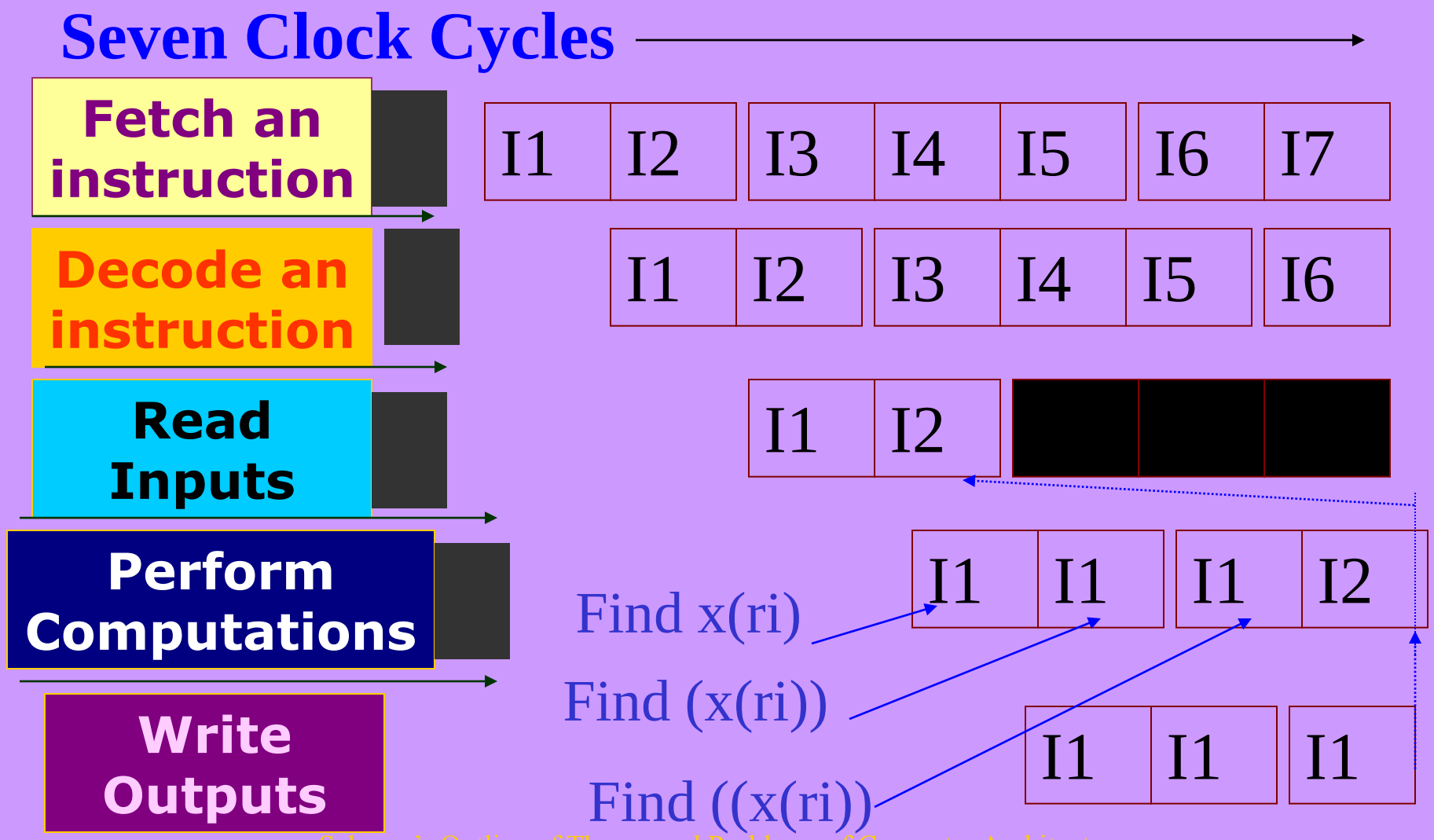
- During load instruction, the pipeline temporarily delayed due to need to fetch an operand from memory
- For example:

Consider LD rj, (x(ri)) and SUB rk, rj

Three cycle requirement in execution Stage for Load

- Instructions LD rj, ((x(ri)), and SUB rk, rj take three cycles at execution stage
- Three-cycles stall because SUB rk, rj needs to read rj, and LD also uses rj

Seven Clock Cycles
LD rj, (x(ri)) and SUB rk, rj



Pipeline execution delays

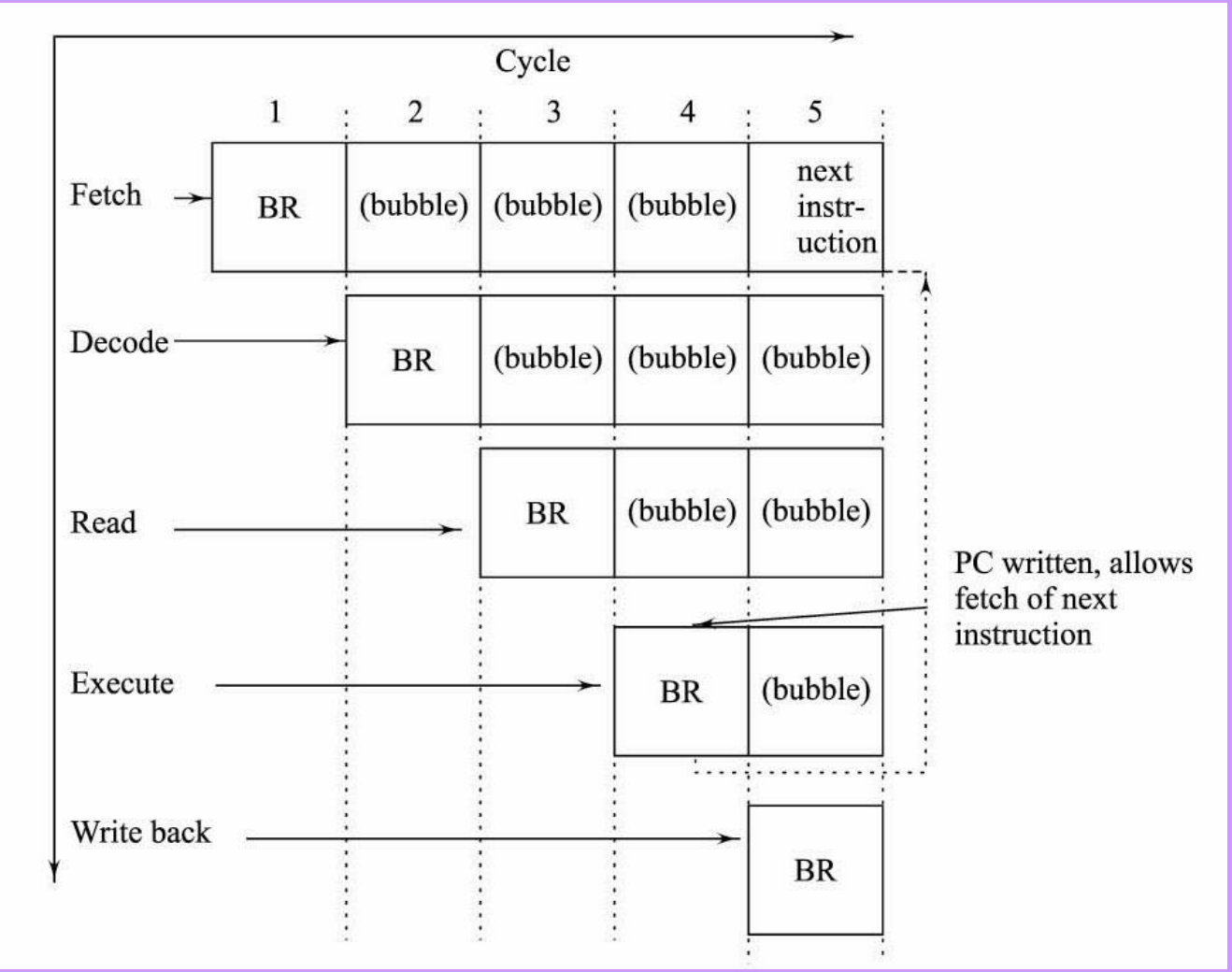
- Due to load from memory in cycle 6
- Due to dependency of SUB on memory accessed loaded operand of previous instruction

Control Hazard due to branch

Control Hazard During Branch

- Branch instructions can also cause delays in pipelined processors, because the processor cannot determine which instruction to fetch next until the branch executed at the execution stage (stage 4)
- The delay also known as *branch penalty*

Pipeline execution delay (Branch penalty)



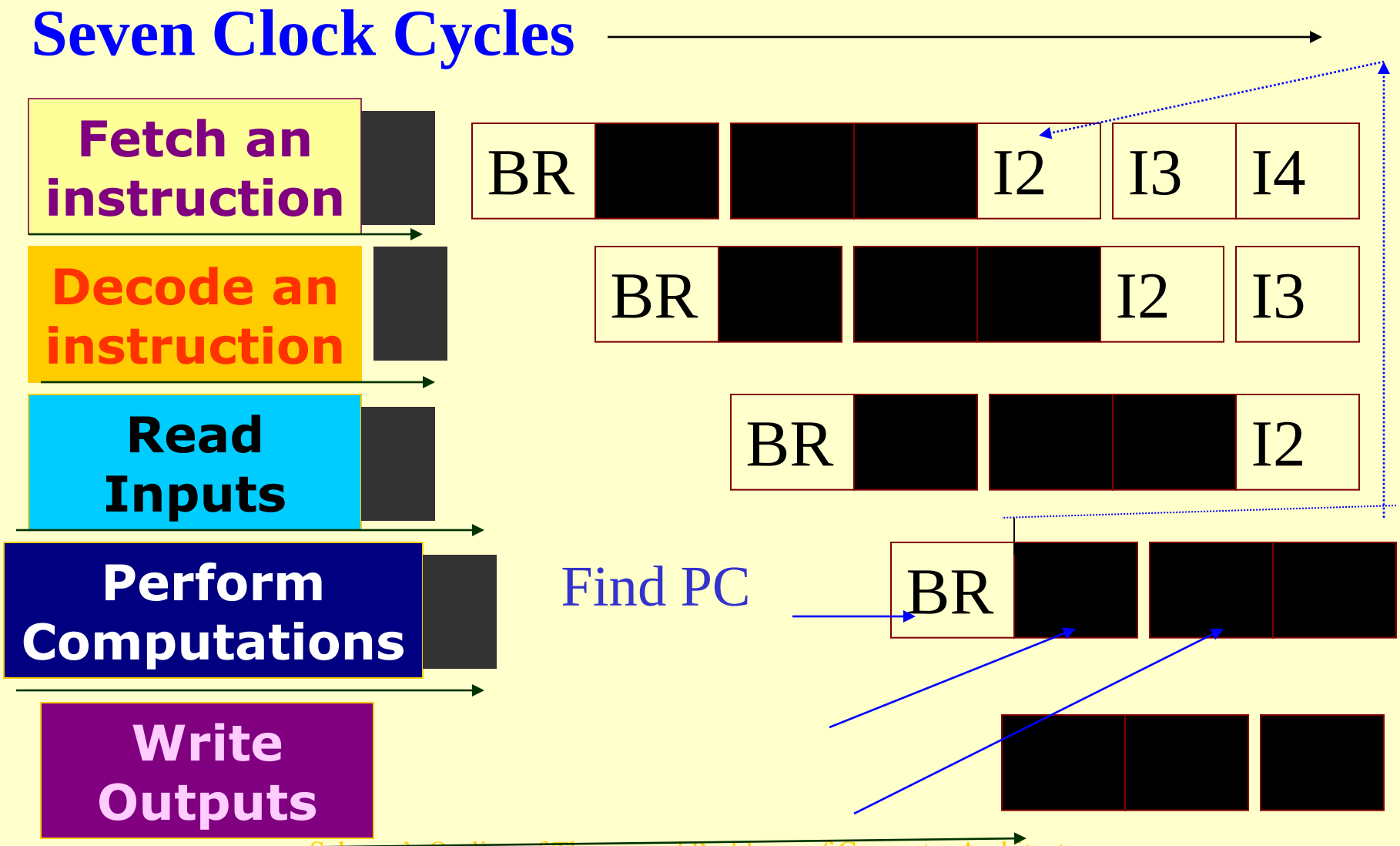
Conditional Branches

- Effectively, branch instructions, particularly conditional branches, create data dependencies between the branch instruction and the instruction fetch stage of the pipeline, since the branch instruction computes the address of the next instruction that the instruction fetch stage should fetch.

Inserting NOPs for the Delays

- For the periods needing delays, either the hardware or compiler inserts special no-operation (NOP) instructions (bubbles)
- Pipeline delay is often called the processor's *delay*

Four Cycle for New PC value and 3 bubbles



Branch Delays (control hazards)

- Delay is due to the control flow of the program
- Four Cycle delay reduces when forwarding of Program Counter value from Execution stage itself there will be three bubbles

Hardware for pre-calculation of Branch Address

- One technique— pre-calculation of branch using additional hardware to allow the result of a branch instruction to be computed earlier in the pipeline

Summary

We learnt

- Control Hazard due to load instruction
- Load Result forwarding after execution stage reduces the NOP in pipeline
- Control Hazard due to branch instruction
- Branch address pre-calculation reduces the NOP in pipeline
- Hardware can be used to predict branching earlier in a pipeline

End of Lesson 04 on

Control Hazards and Branches