[CSED311] Lab5

Pipelined CPU

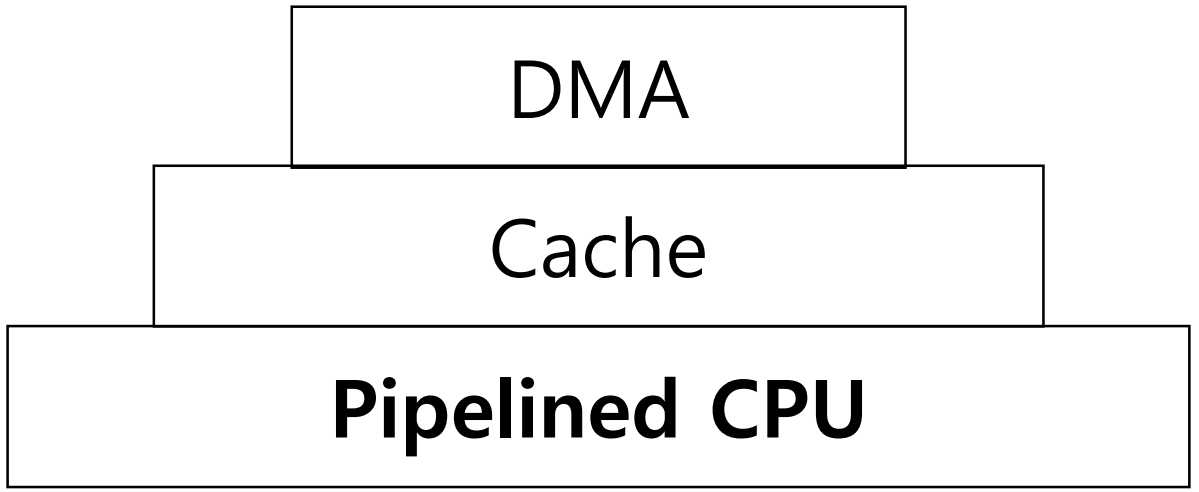
Sanghwan Jang
jsh710101@postech.ac.kr

POSTECH

Objectives

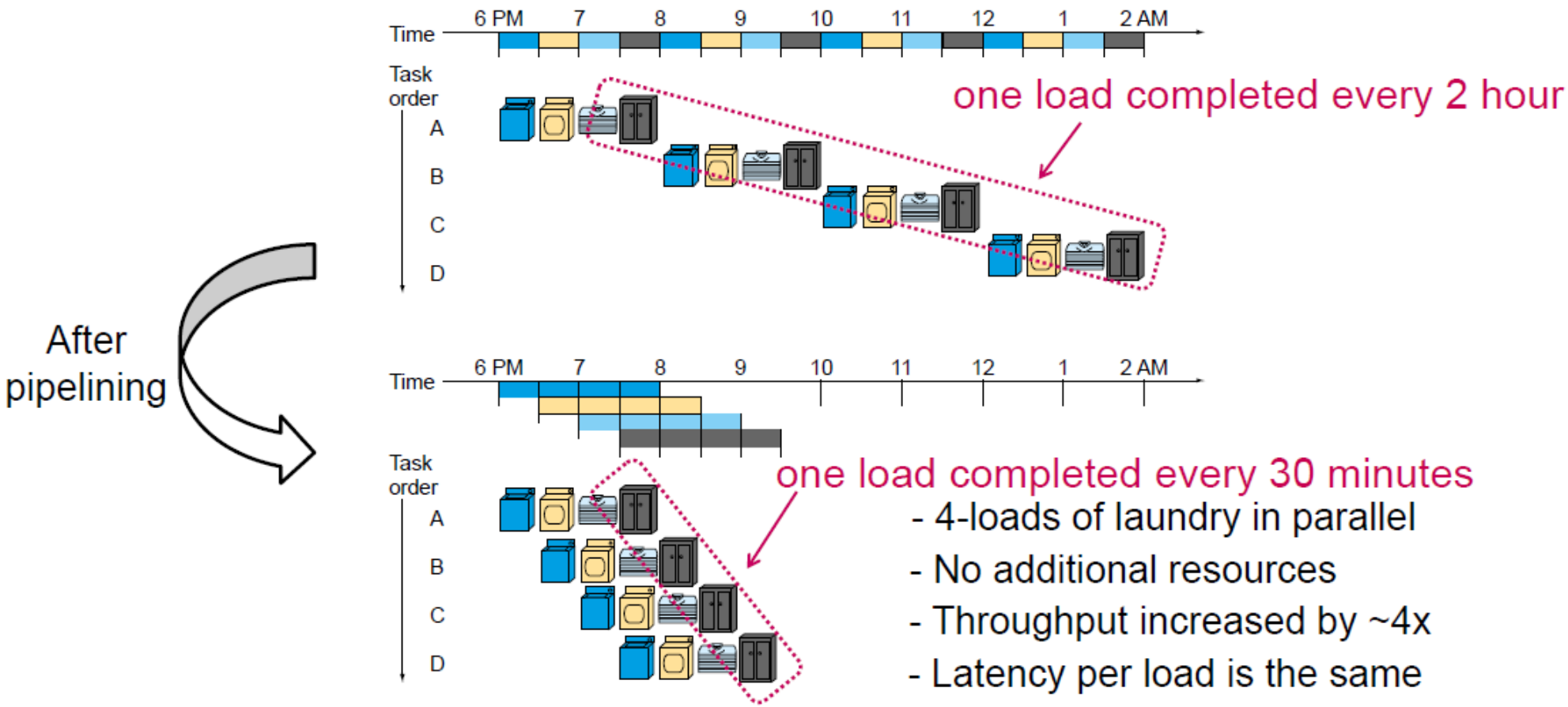
- Understand the reason why pipelined CPUs have better throughput
- Understand data & control hazards and how to solve it
- Design & Implement the pipelined CPU

Lab Dependency

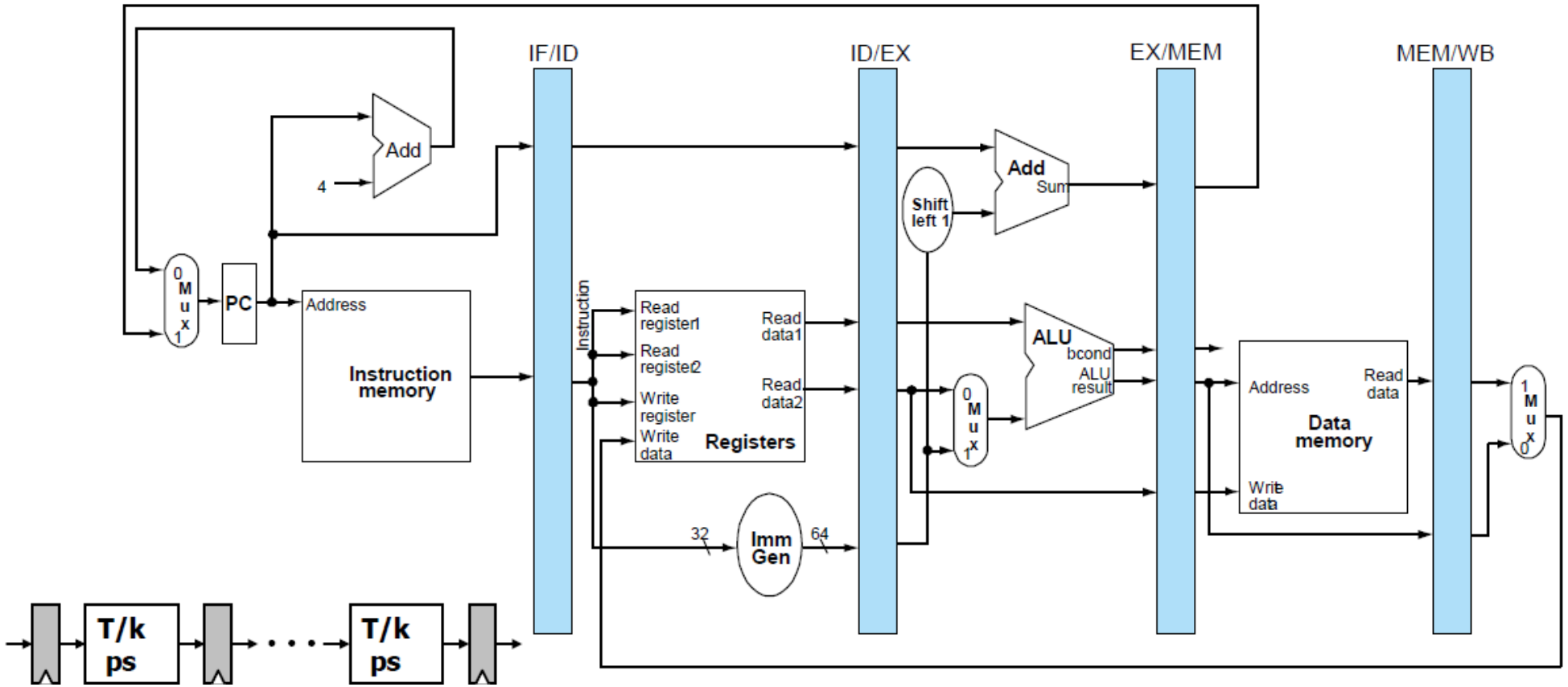


- From now on, **you have to complete your Lab to start the next one.**
- Your Implementation should be **functionally correct.**

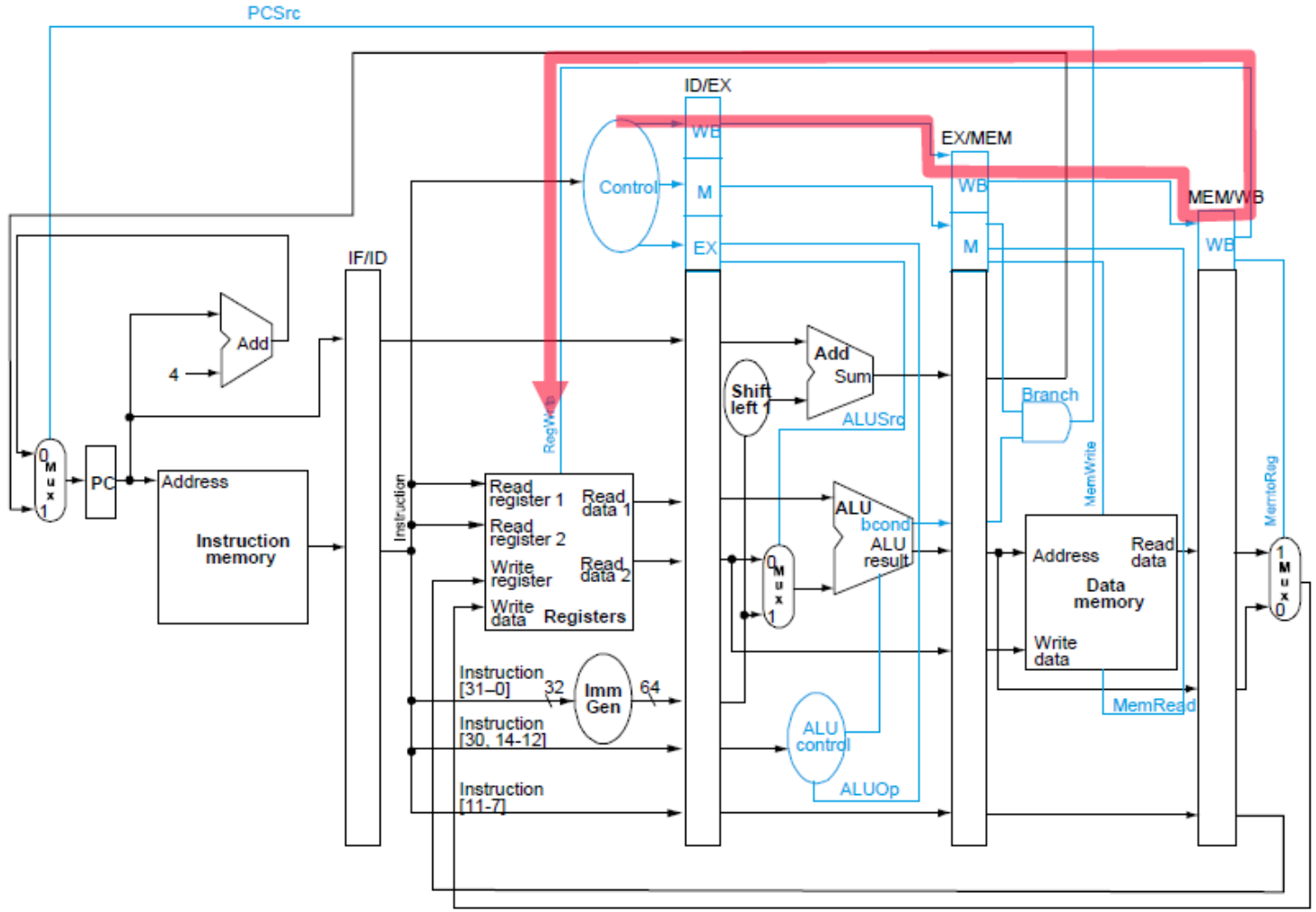
Why Pipelined CPU?



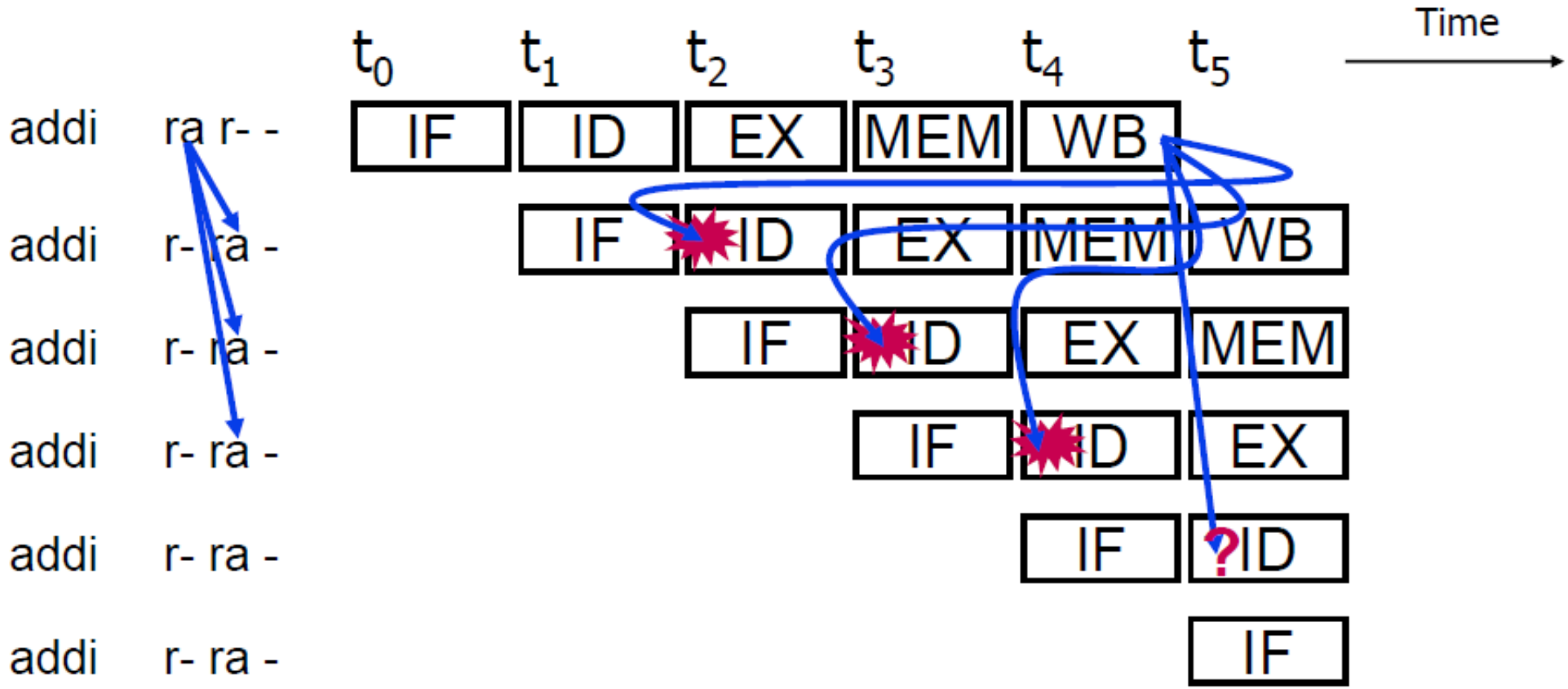
Pipelined CPU



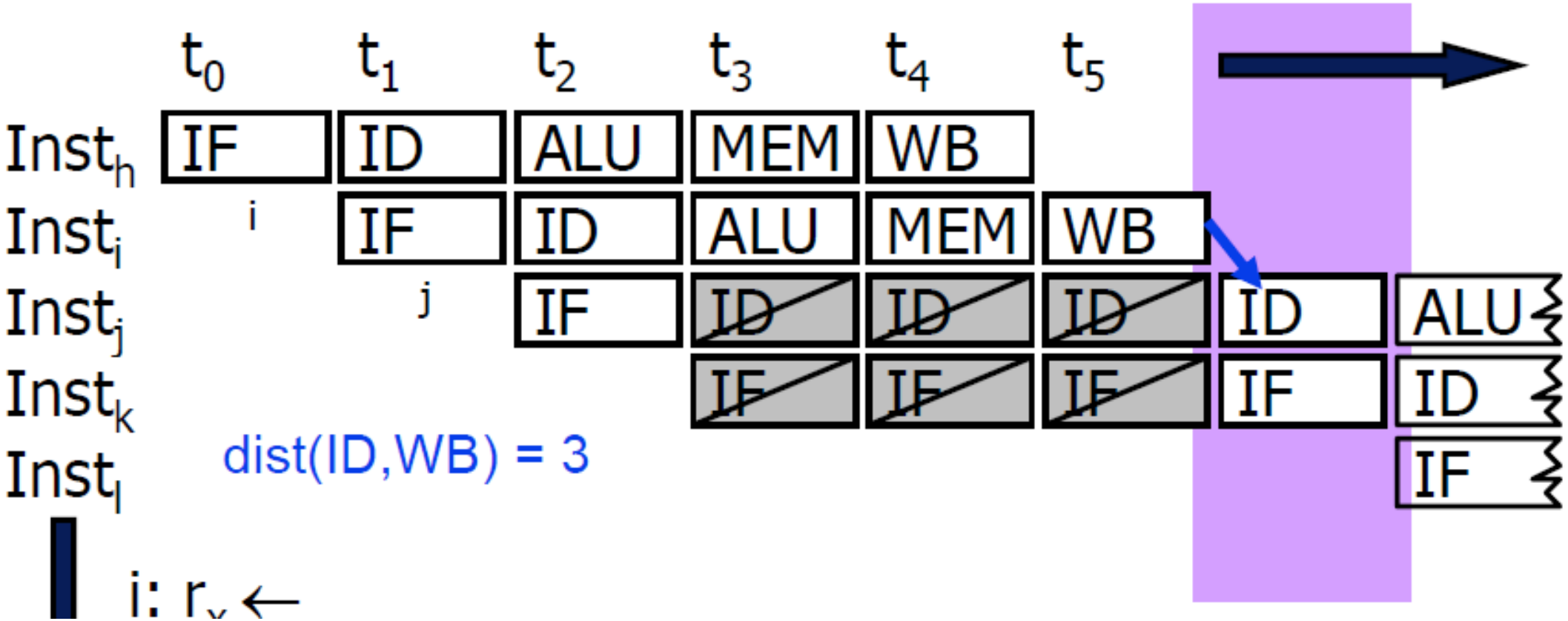
Pipelined CPU: Control



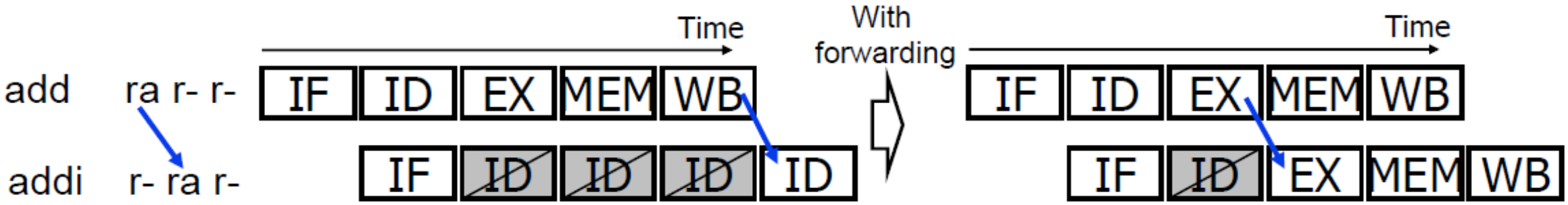
Data Hazard


Time
t0
t1
t2
t3
t4
t5
addi ra r- -
addi r- ra -
addi r- ra -
addi r- ra -
addi r- ra -
addi r- ra -
IF
ID
EX
MEM
WB
?ID

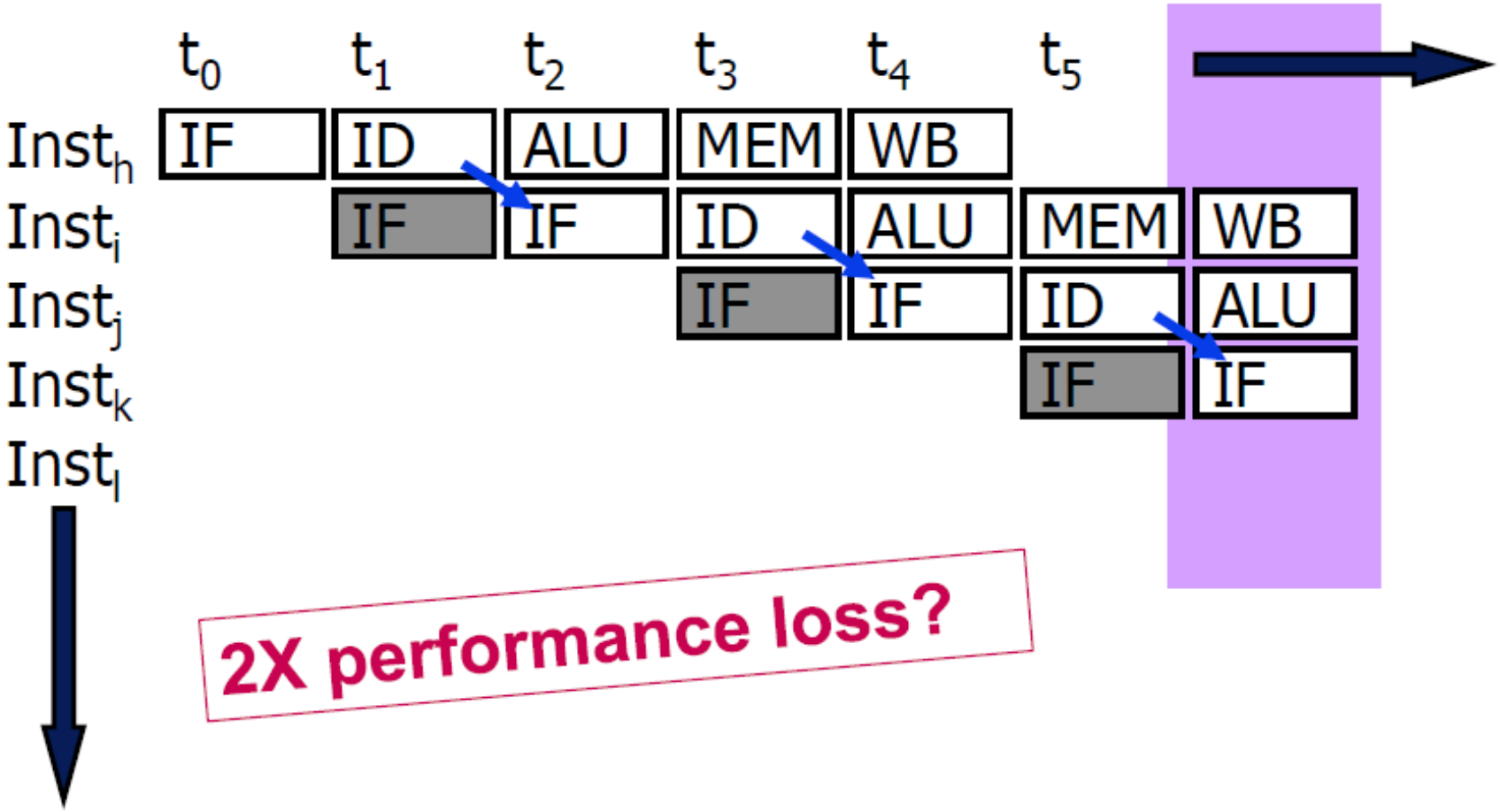
Data Hazard


t0
t1
t2
t3
t4
t5
Insth
Insti
Instj
Instk
Instl
IF
ID
ALU
MEM
WB
i
j
dist(ID,WB) = 3
i: rx ←

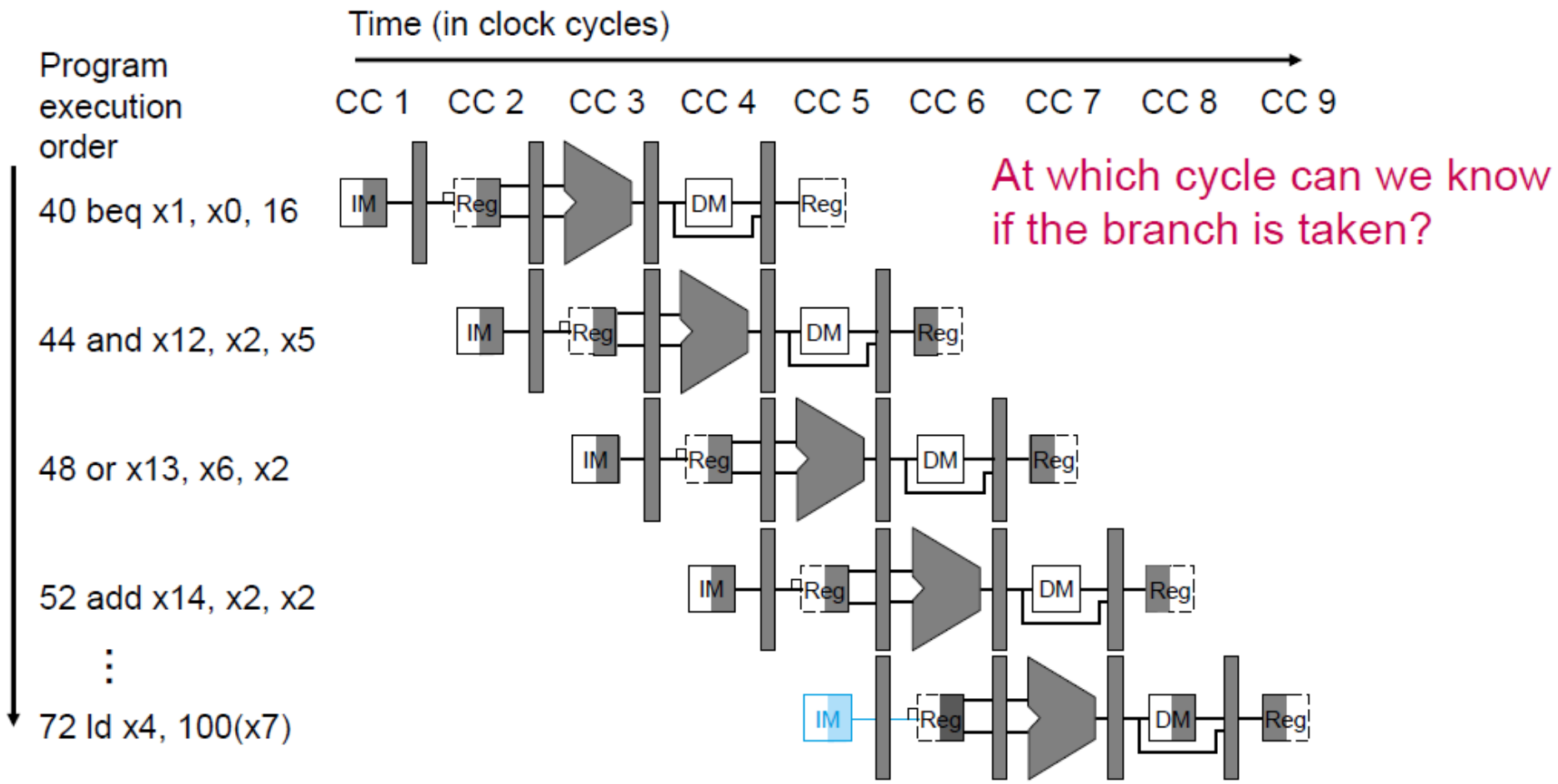
Data Hazard: Forwarding



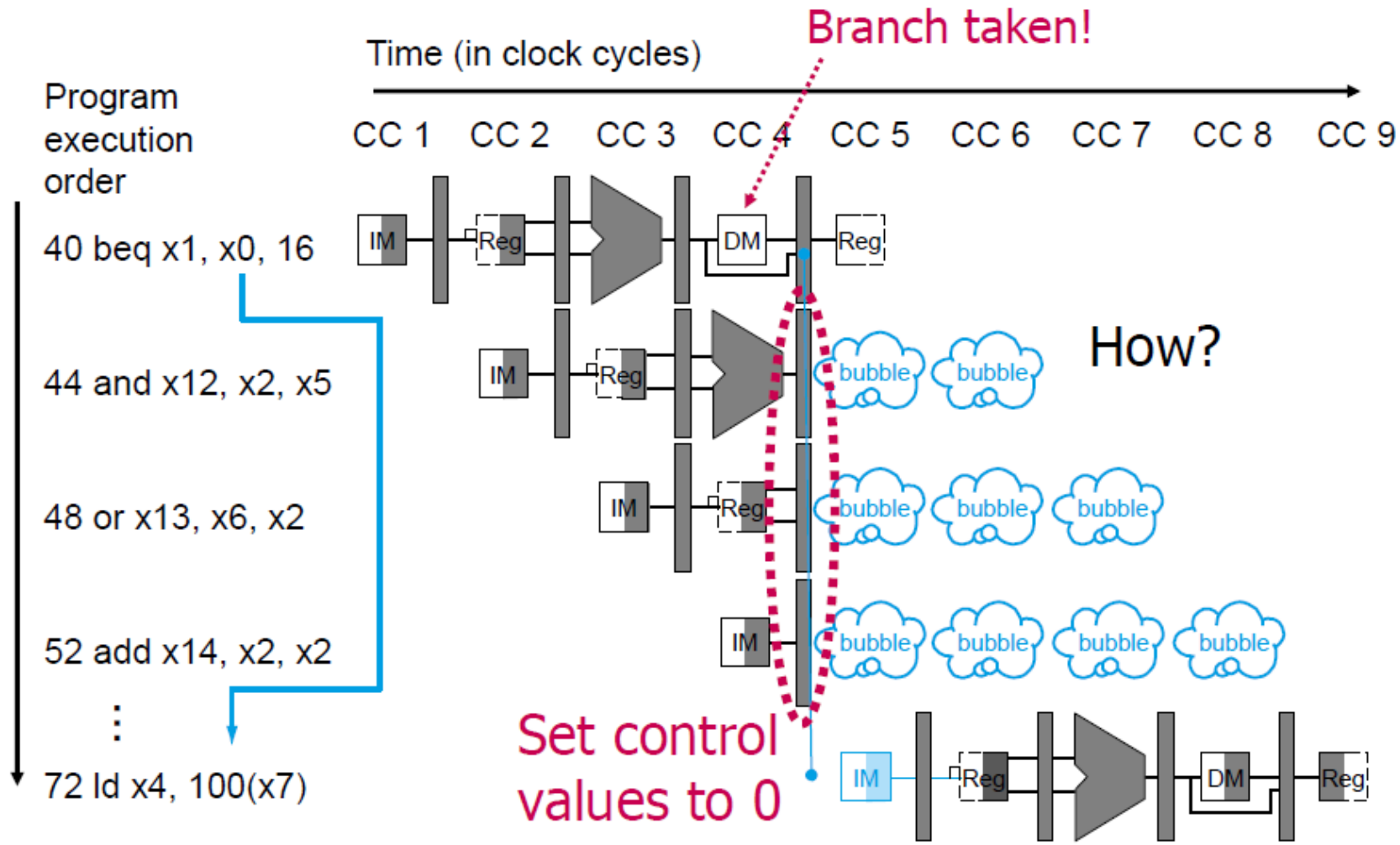
Control Hazard



Control Hazard: Branch Prediction



Control Hazard: Branch Prediction



Requirements

- Design & Implement the pipelined CPU
  - Updating register file at clock negative edge is allowed.

- Compare the performance of multi-cycle CPU and pipelined CPU
  - Compare the number of clock cycles to execute all test cases
  - Write it in the discussion section of the report.

- **EXTRA CREDIT**
  - Forwarding
  - Branch Prediction (+Flush)
    - Always Not-Taken, Always Taken, etc.