Computer Architecture Experiment

3rd lab

Topics

- 0、Basic Knowledge
- 1、Warm up
- 2、simple 5-stage of pipeline CPU Design
- 3、Pipelined CPU with stall
- 4、Pipelined CPU with forwarding
- 5、Pipelined CPU resolving control hazard and support execution 31 MIPS Instructions

Outline

- Experiment Purpose
- Experiment Task
- Basic Principle
- Operating Procedures
- Precaution

Experiment Purpose

- Understand the principles of Pipelined CPU Stall
- Understand the principles of Data hazard
- Master the method of Pipelined CPU Stalls Detection and Stall the Pipeline.
- master methods of program verification of Pipelined CPU with Stall

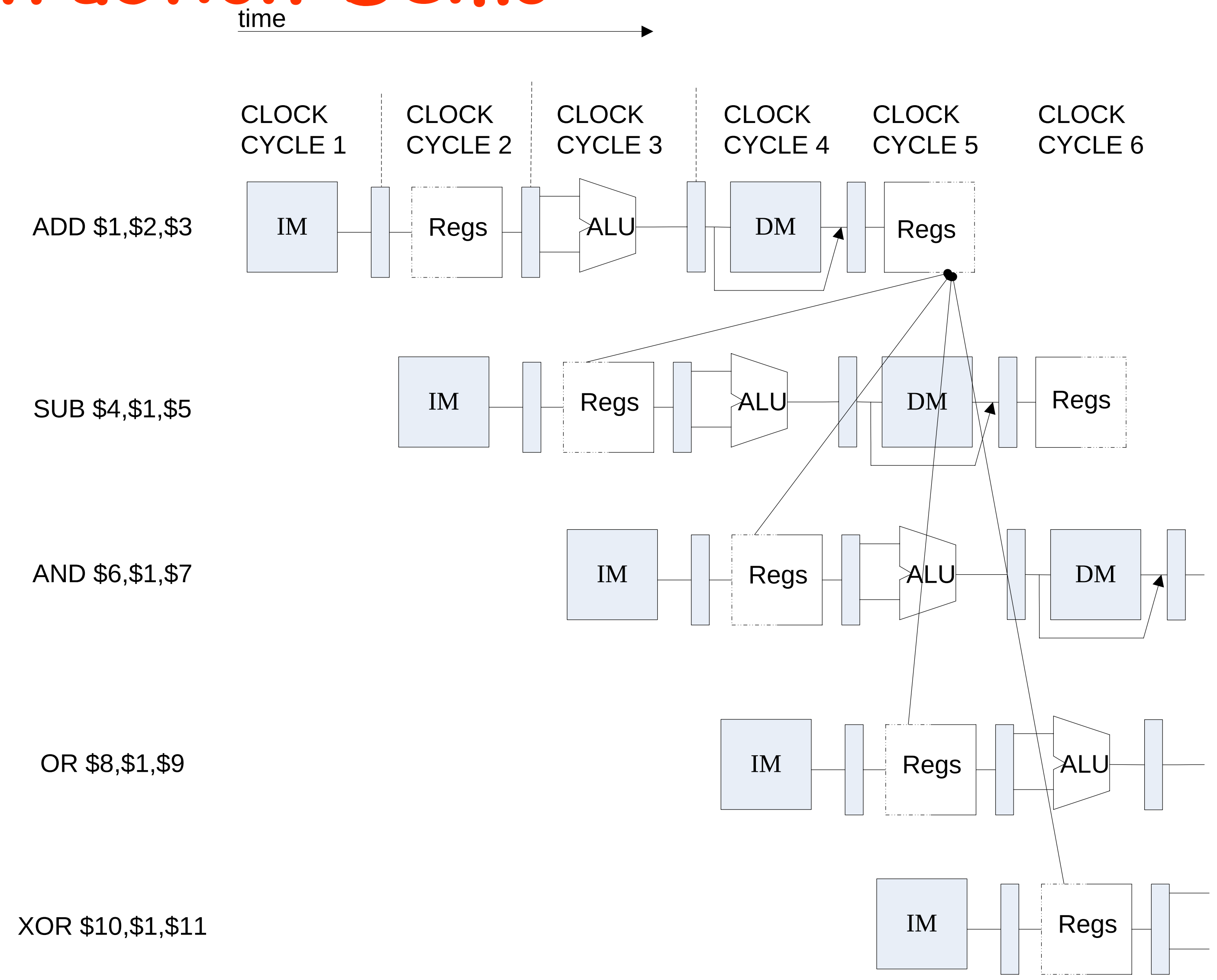
Experiment Task

- Design the Stall Part of Datapath of 5-stages Pipelined CPU
- Modify the CPU Controller, adding Condition Detection of Stall.
- Verify the Pp. CPU with program and observe the execution of program
- Write a testing code that including all the 15 instructions you design, and including data hazards and control hazards.
- Compare the executing result running on your CPUs in Lab2 and Lab3.

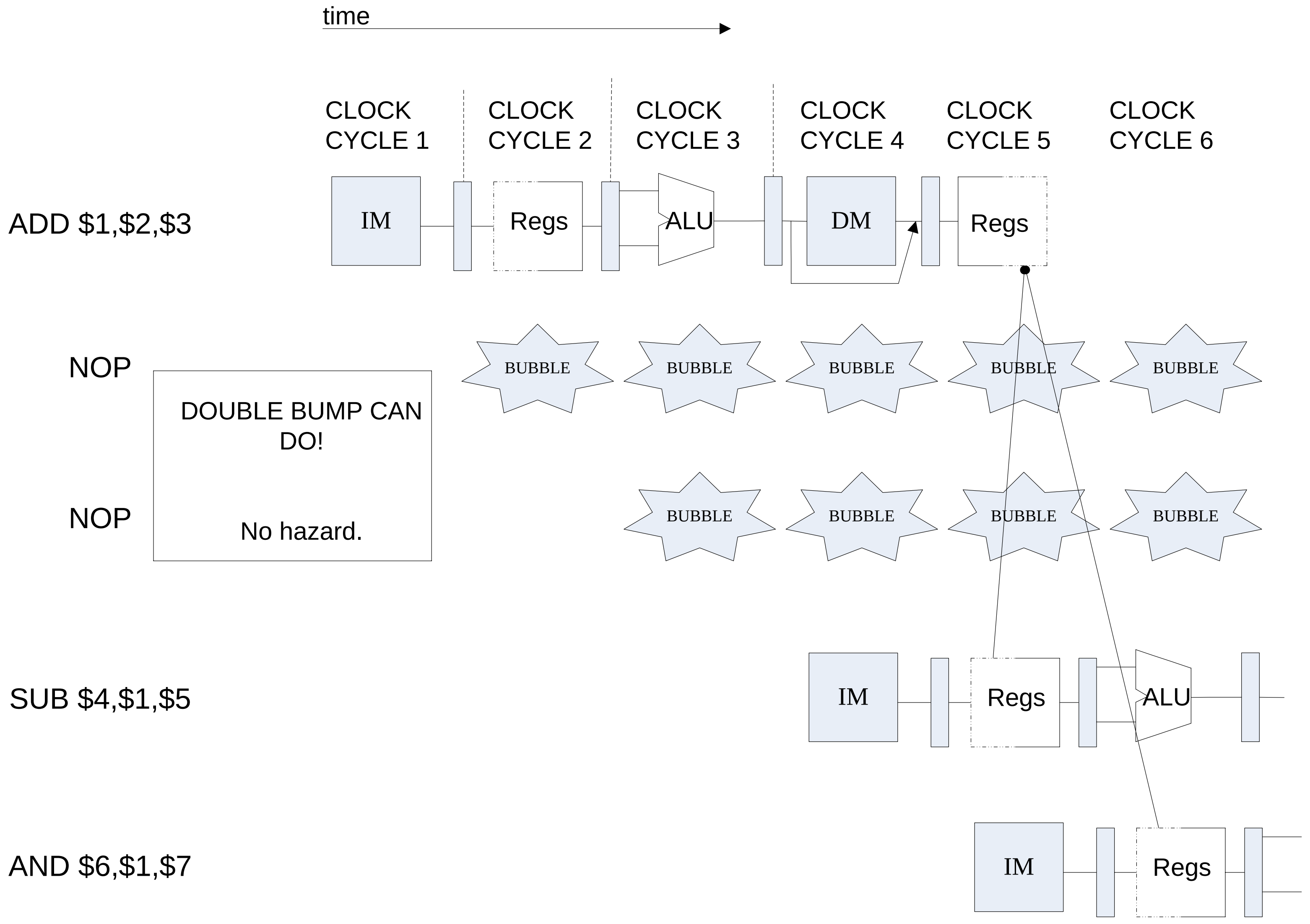
Data Hazard Definition

- Data Hazards arise when an instruction depends on the results of a previous instruction in a way that is exposed by the overlapping of instructions in the pipeline.
- When inst. i executes before instr. j, there are data hazards:
- RAW, i.e. instr. j read a source before instr. i writes it.
- WAW, i.e. instr. j write an operand before instr. i writes it.
- WAR, i.e. instr. j write a destination before instr. i read it.

Instruction Demo



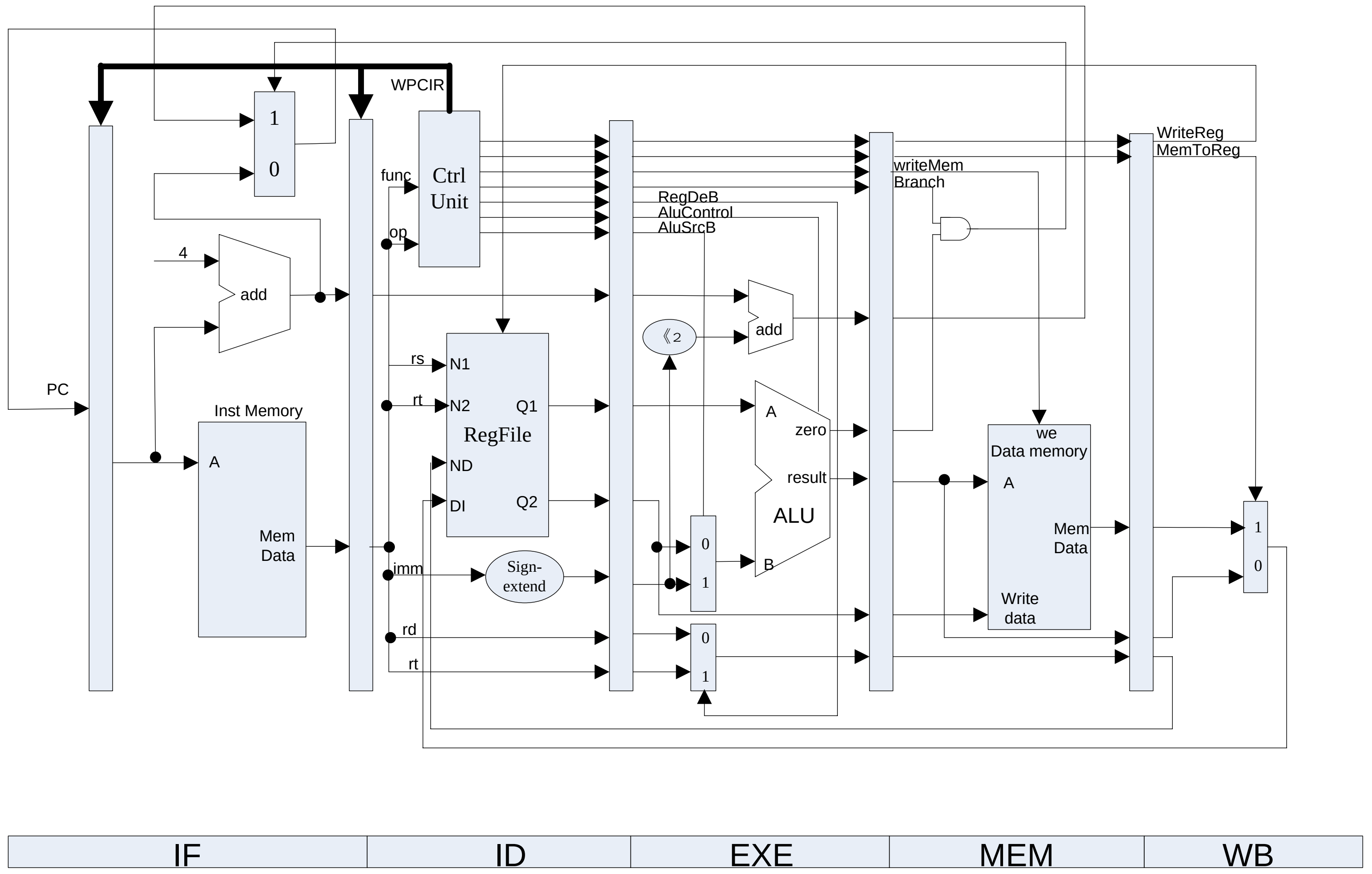
Data Hazard Causes Stalls



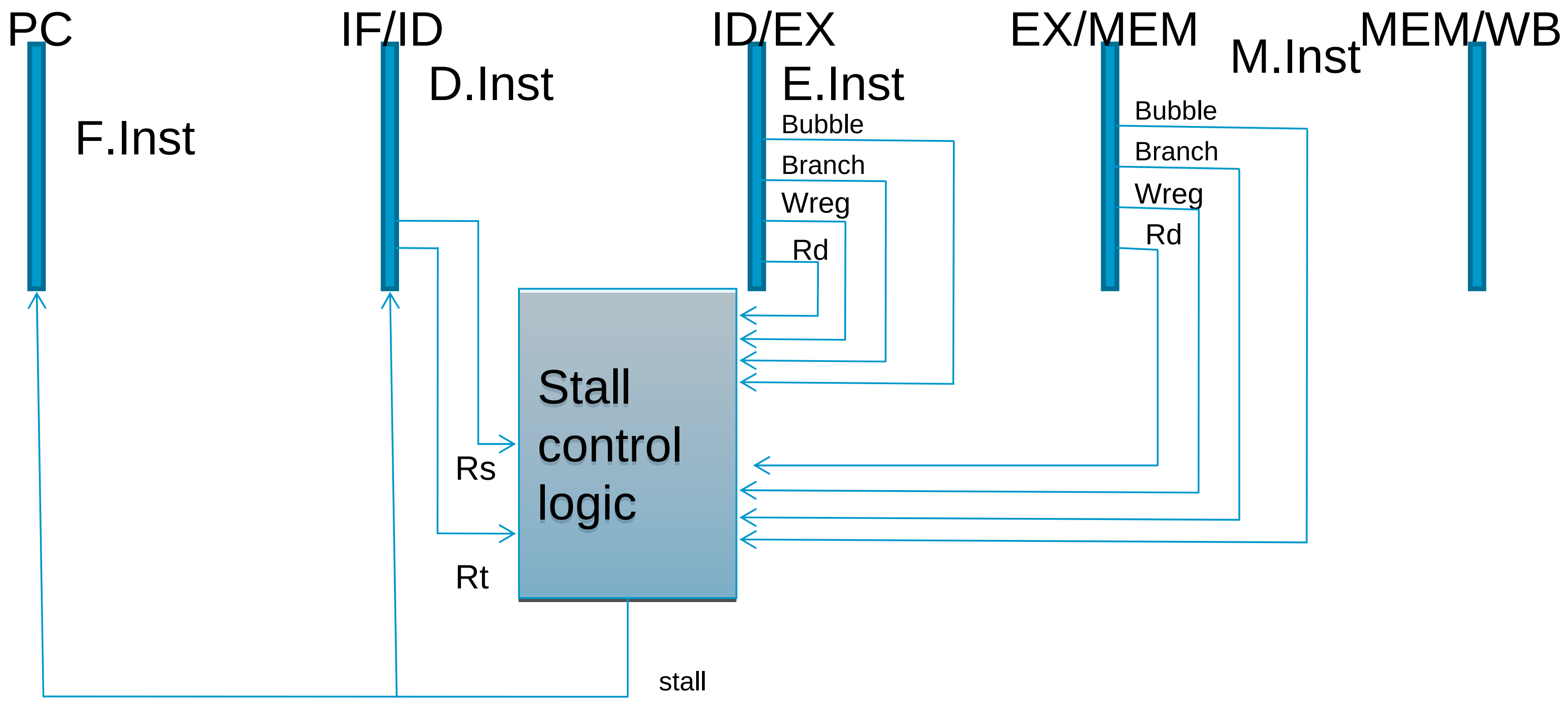
DOUBLE BUMP CAN DO!

No hazard.

How to Stall the Pipeline



In Which Conditions it Causes Stall



1) To produce a stall signal. If stall ==1 then
2) Use stall to disable writing PC write and IF/ID latch.
3) Push a bubble into next stage.
   - ✓ Bubble = 1 and disable control signal Wreg and Wmem.

Controller Module

- if_instr: if stage instruction
- Instr: id stage instruction
- ex_instr: ex stage instruction
- mem_instr: mem stage instruction
- wb_instr: wb stage instruction

- assign cu_wpcir = stall;

Observation Info

- Input
  - West Button: Step execute
  - South Button: Reset
  - 4 Slide Button: Register Index
- Output
  - 0-7 Character of First line: Instruction Code
  - 8 of First line : Space
  - 9-10 of First line : Clock Count
  - 11 of First line : Space
  - 12-15 of First line : Register Content
  - Second line : “stage name”/number/type
  - stage name: 1-”f”, 2-”d”, 3-”e”, 4-”m”, 5-”w”

Test code

lw r1, $21(r0)	8c010015
add r2,r1,r1	00211020
sub r3,r1,r2	00221822
beq r1,r1,2	10210002
andi r4,r2,0	30440000
addi r5,r4,1	20850001
ori r6,r3,0	34660000
bne r6,r3,2	14c30002
lw r7,$20(r0)	8c070014
sw r7,$21(r0)	Ac070015
addi r8,r7,1	20e80001
j 0	08000000
andi r9,r8,1	31090001

- Thanks!