Computer Architecture Experiment

Jiang Xiaohong

College of Computer Science & Engineerin

Zhejiang University

ZheJiang University

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 Bypass Unit
- Master the method of Pipelined Pipeline Forwarding Detection and Pipeline Forwards.
- Master the Condition In Which Pipeline Forwards.
- Master the Condition In Which Bypass Unit doesn’t Work and the Pipeline stalls.
- master methods of program verification of Pipelined CPU with forwarding

Experiment Task

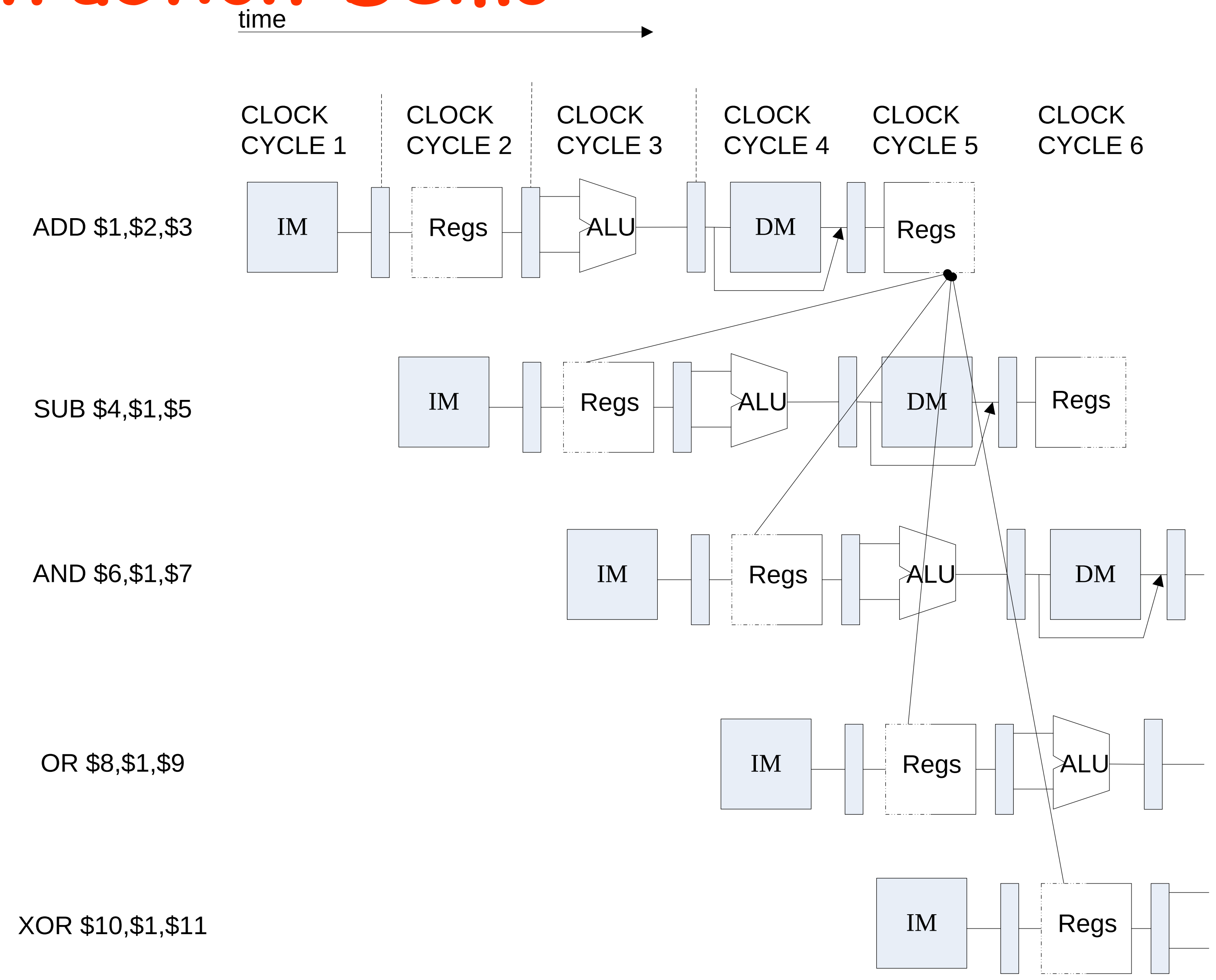
- Design the Bypass Unit of Datapath of 5-stages Pipelined CPU
- Modify the CPU Controller
  – Conditions in Which Pipeline Forwards.
  – Conditions in Which Pipeline Stalls.
- Verify the Pipeline CPU with program and observe the execution of program

Data Hazard Stalls

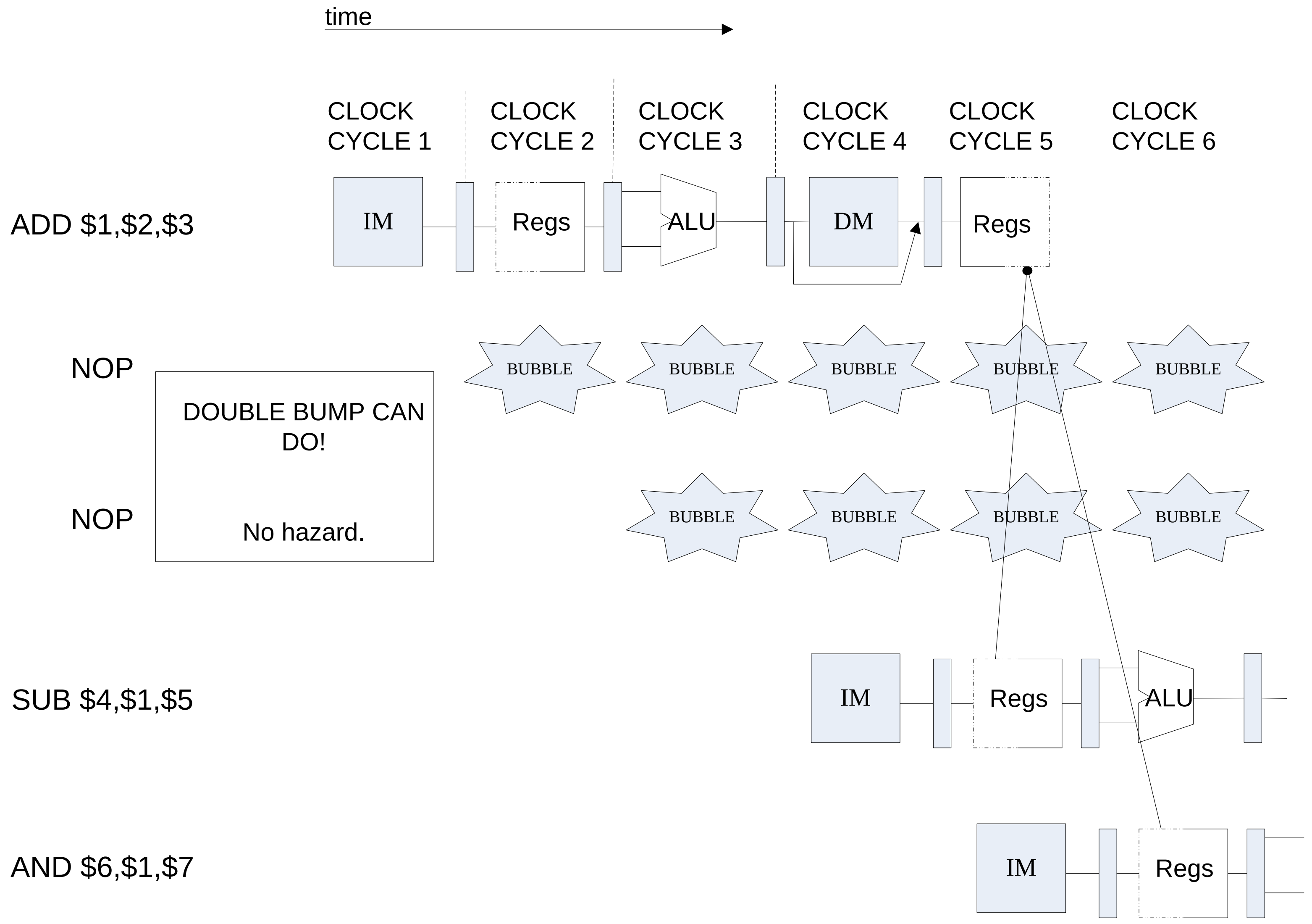
- Minimizing Data Hazard Stalls by Forwarding： In most cases, the problem can be resolved by forwarding, also called bypassing, short-circuiting.

- Data Hazards Requiring Stalls: In some cases, data hazards can not be handled by bypassing.

Instruction Demo



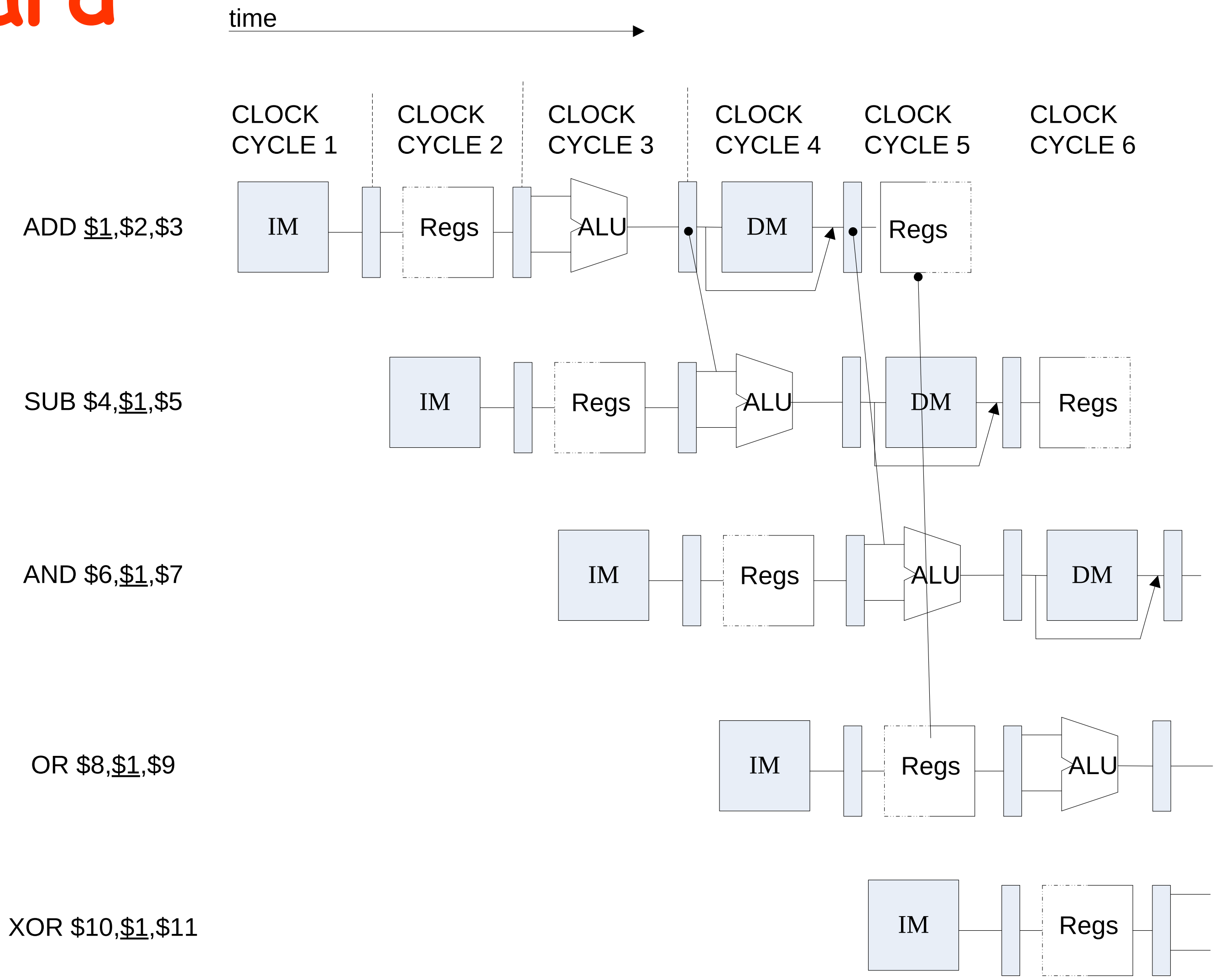
Data Hazard Causes Stalls



DOUBLE BUMP CAN DO!

No hazard.

Pipeline Forward to Avoid the Data hazard



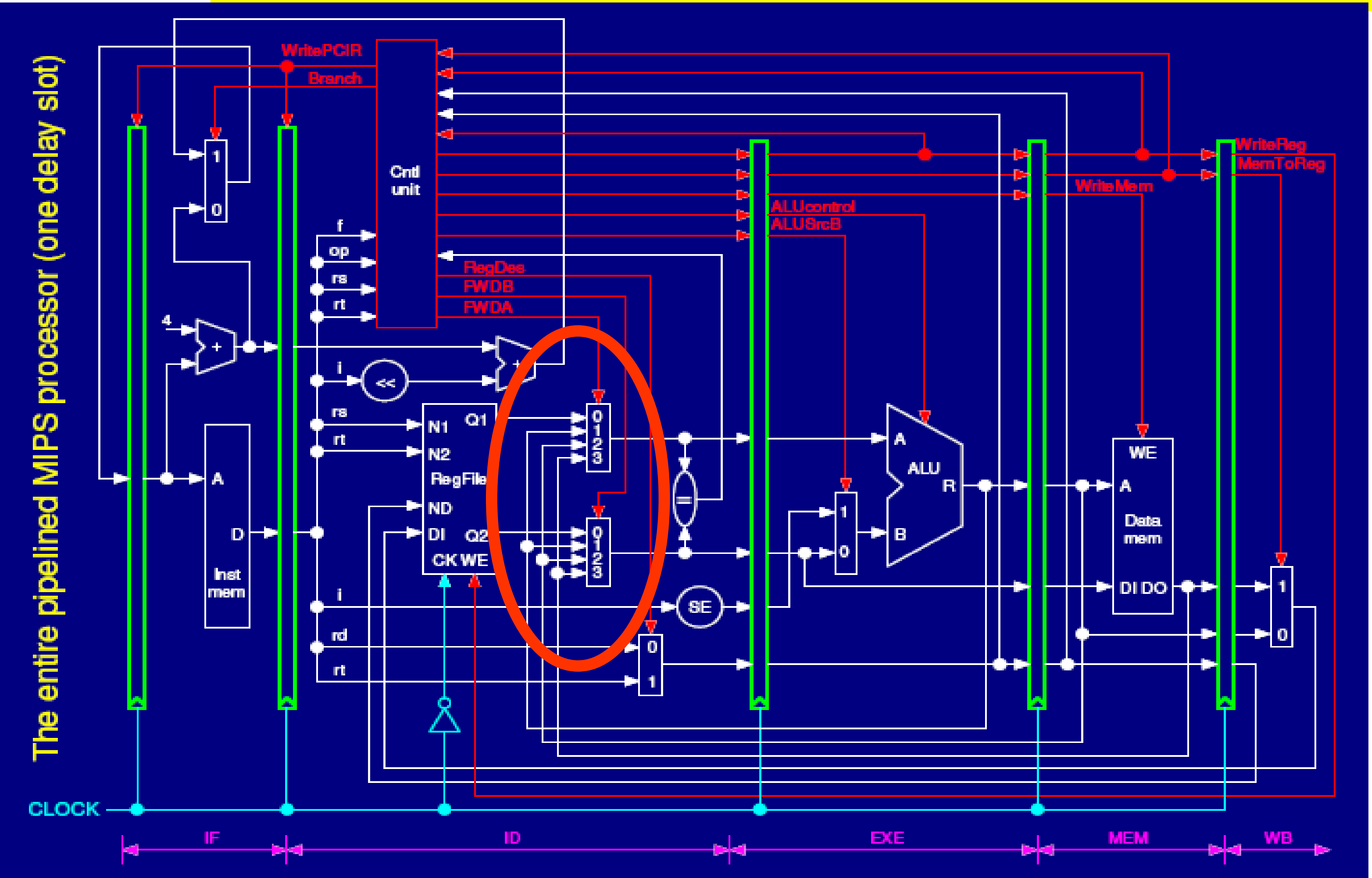
Setup forwarding path

- Adding forwarding path for your pipelined CPU implemented in Lab3.

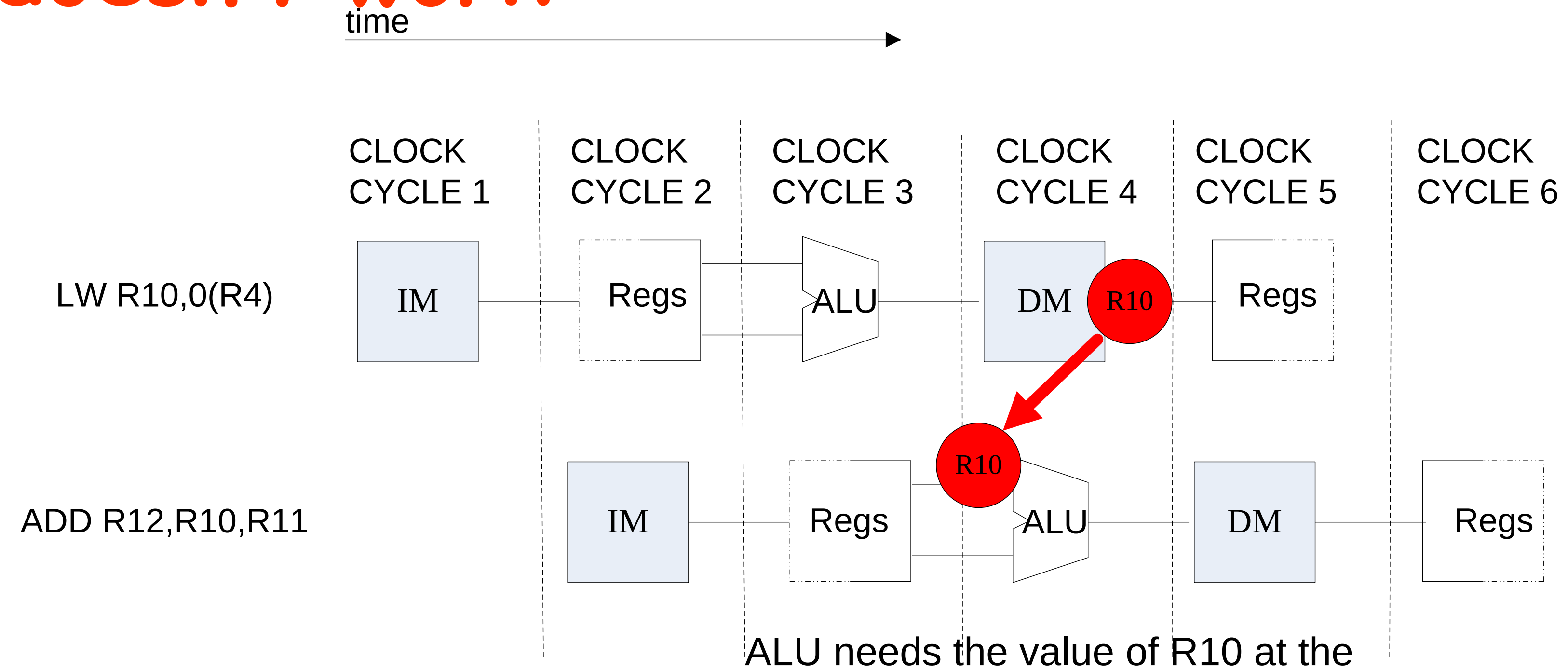
- Notes:
  - The graph on the following page is only for your reference. You can implement the forwarding paths in other way, say in the EXE stage. (Draw the graph in your lab report.)
  - You need to set up ALL the “forwarding paths” not only those go to “input ports” of ALU, but also go to “data input port” of Data Memory, or “branch equal judger unit” in your pipeline.
  - You still need the “stall” signal and the “interlock” unit to insert stalls in some cases.

Wh

- Move the Forwarding path to ID stage
- Move the forwarding control logic to ID stage

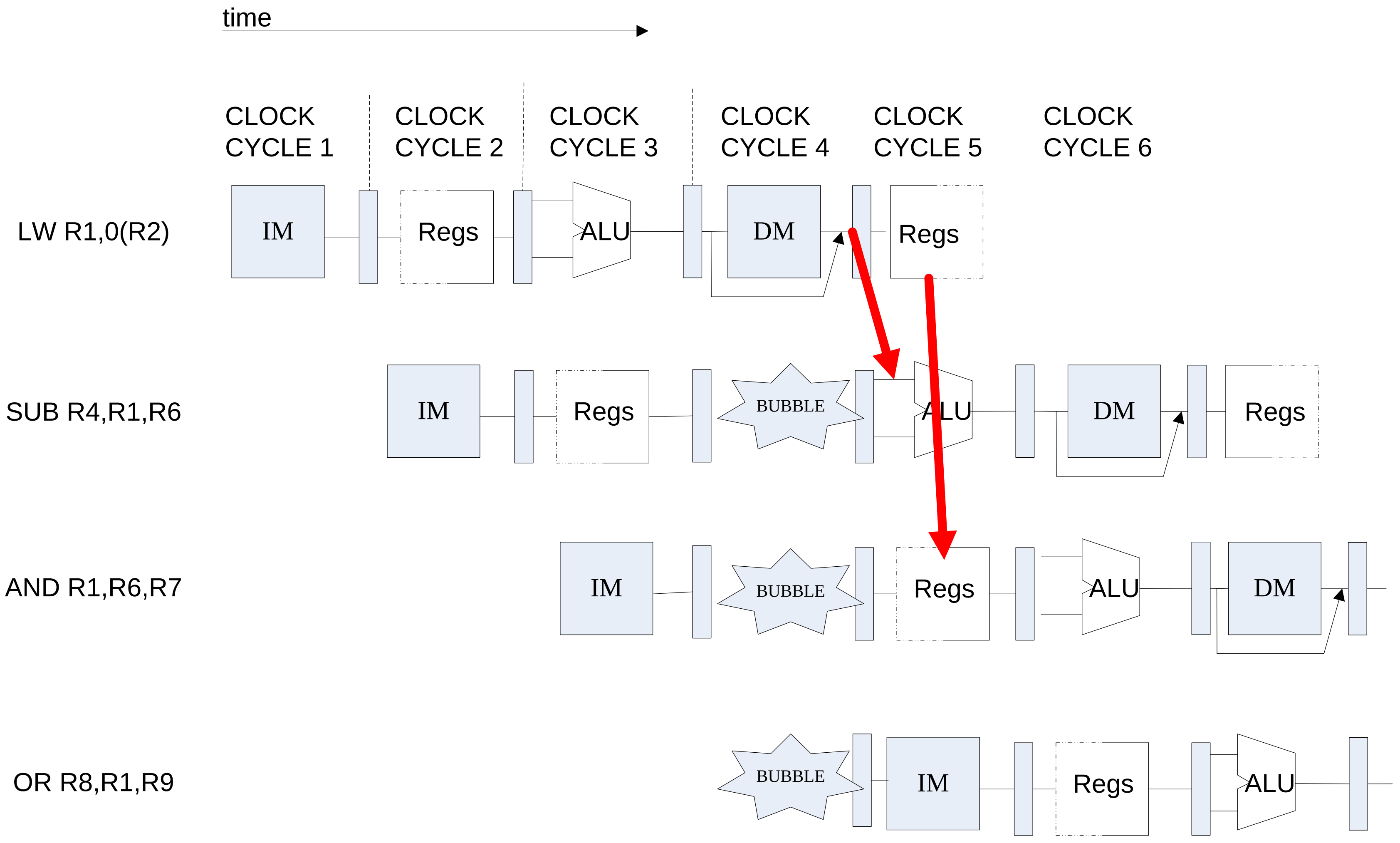


Condition in Which Bypass Unit doesn't work

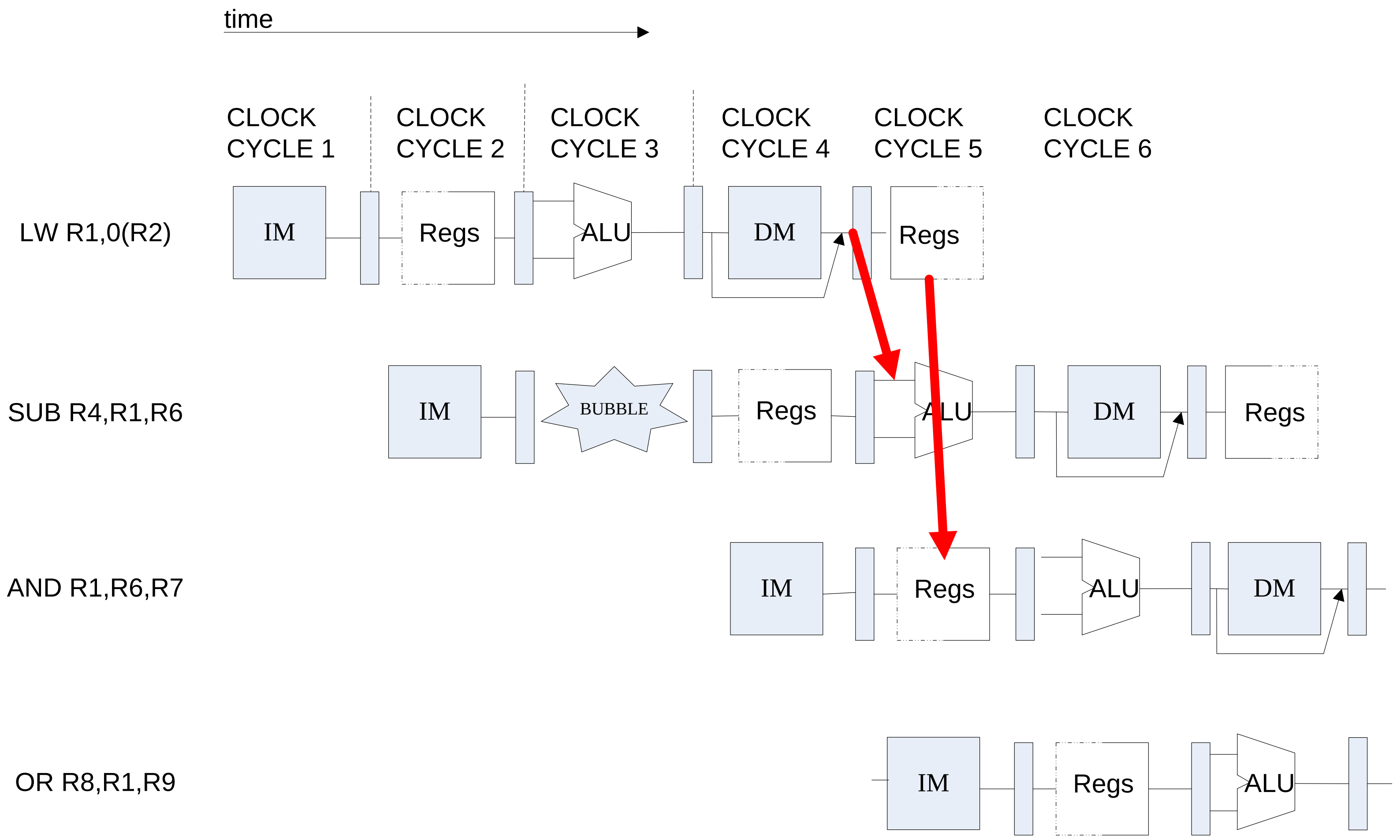


ALU needs the value of R10 at the beginning of Clock Cycle 4, but at the end of Clock Cycle 4 DM outputs the R10

Pipeline Stalls



Pipeline stalls at ID Stage



Pipelined CPU Top Module

- module top (input wire CCLK, BTN3, BTN2, input wire [3:0]SW, output wire LED, LCDE, LCDRS, LCDRW, output wire [3:0]LCDDAT);
- assign pc [31:0] = if_npc[31:0];

- if_stage x_if_stage(BTN3, rst, pc, mem_pc, mem_branch, id_wpcir, …
-    IF_ins_type, IF_ins_number,ID_ins_type,ID_ins_number);

- id_stage x_id_stage(BTN3, rst, if_inst, if_pc4, wb_destR,…,
-    EX_ins_type, EX_ins_number, id_FWA, id_FWB);
-
- ex_stage x_ex_stage(BTN3, id_imm, id_inA, id_inB, id_wreg, ..
- id_FWA, id_FWB,mem_aluR, wb_dest, … , MEM_ins_number);
-
- mem_stage x_mem_stage(BTN3, ex_destR, ex_inB, ex_aluR, …
- mem_aluR, ... , WB_ins_type, WB_ins_number);

- wb_stage x_wb_stage(BTN3, mem_destR, mem_aluR, …
- wb_dest, …, OUT_ins_type, OUT_ins_number);

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
  - stage name: 1-”f”, 2-”d”, 3-”e”, 4-”m”, 5-”w”

Test code

- You can use the same test code for Lab3.
- Test code can be downloaded in the material directory on coursewebsite.

- Thanks!