Lab 06:
HW 5

Hunjun Lee
hunjunlee@hanyang.ac.kr

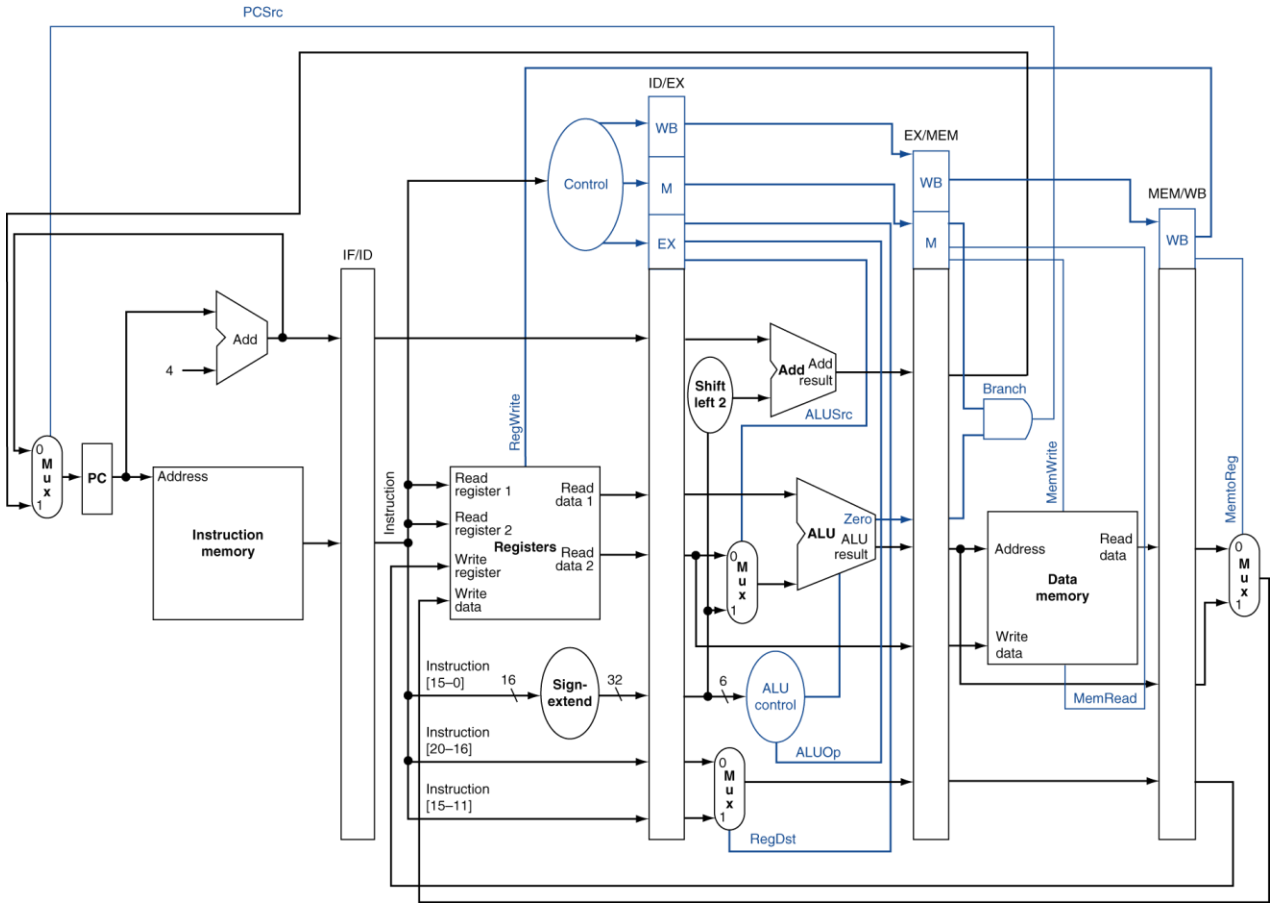
Overview

- You will upgrade your single cycle CPU into a pipelined CPU

- What to do?
  - Implement a five-stage pipelined CPU
  - We need a basic pipeline that does not support branch prediction and data forwarding
    - Predict not-taken and flush on misprediction
    - Stall upon detecting a data hazard

Target CPU Microarchitecture

- Implement a basic pipelined architecture following the CPU architecture you learned in the class
  - Add hazard detection module …



What to implement

- #1) Divide the computation into five stages and execute multiple instructions in parallel
- #2) Properly set the control signals over the five pipeline stages (whichever way you want)
- #3) Predict not-taken during instruction fetch
  - If the branch is taken, flush the mis-predicted instructions
  - You are allowed to perform early branch resolution @ ID stage if you want
- #4) Detect data hazard and stall
  - If the CPU detects a data hazard, stall the pipeline and re-execute the ID stage (do not support data forwarding for now)

Assignment

- **Files:**
  - **MEM.cpp/h + MEM.v (HW3)**
    - Use the MEM files from HW3 again (You cannot share resources between IF and MEM stage)
  - **Modify everything else if you need**
    - Use the files from previous HWs and modify them if you need
  - **You are allowed to add your own HW module and files if you want**
    - I personally added HAZARD.v/cpp/h for hazard detection and stall

Submission

- Submission (Zip all the files)
  - For a two-people team: lab5_student_id1_student_id2.zip
  - For a one-person team: lab5_student_id.zip
    - You must follow the format (-10% for wrong file format)
    - Example:
      - lab5_2020102030.zip
      - lab5_2020102030_2022103040.zip
  - The zip file should contain:
    - every cpp/h/v files
    - lab5_report.pdf

Submission

- You need to write a 4+-page report
  - Explain the overall structure and how you implemented each program
  - Draw the hardware modules if needed …
  - How did you implement the hazard detection unit, and stall logic …

- Due: Fri. May 30th
  - 1 week delay: -20%
  - 2 week delay: -50%
  - Further delay: 0 point

- You can submit HW4 by May 30th with only 20% deduction
  - I do not accept additional delay