

Control Hazards (Slides T8)

VE370FA22 TA Yuxuan Tang

Branch Hazards

Branch outcome determined in MEM. There are three method to solve it.

- Stall on branch
- Always assume branch not taken
- Branch prediction (instead of assumption)

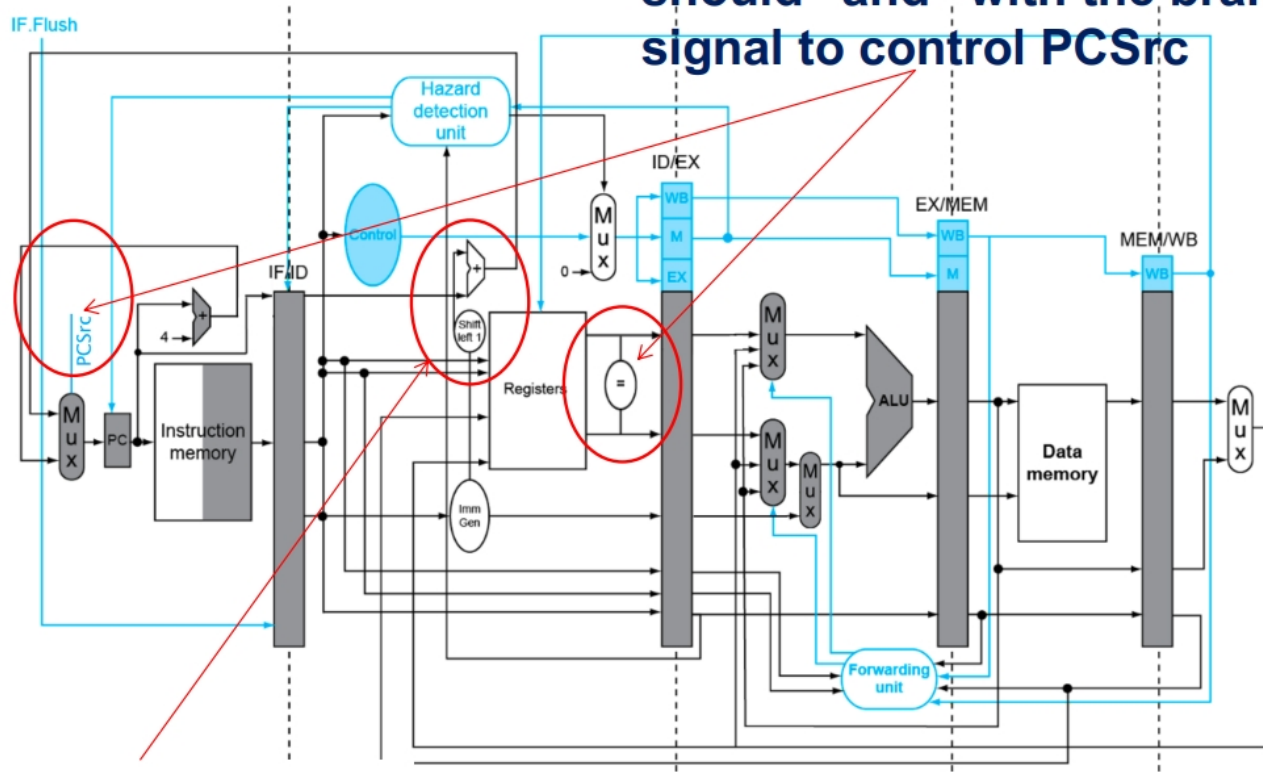
1. Stall on branch

- Wait until branch outcome is determined (In MEM) before fetching the next instruction – 3 nops
- By moving hardware for determining branch outcome to the ID stage – 1 nop

Move Branch Judgement Logic

- Target address adder
- Register comparator
- Branch logic (the and gate)

Register comparison, output should “and” with the branch signal to control PCSrc



2. Always assume branch not taken

Always assume branch not taken. If we are wrong, penalty will be to flush one (previously, three) instructions.

- Flush an Instruction
 - Clear IF/ID pipeline register, by a new control signal `IF.Flush`
 - In IF stage.

New Data Hazards

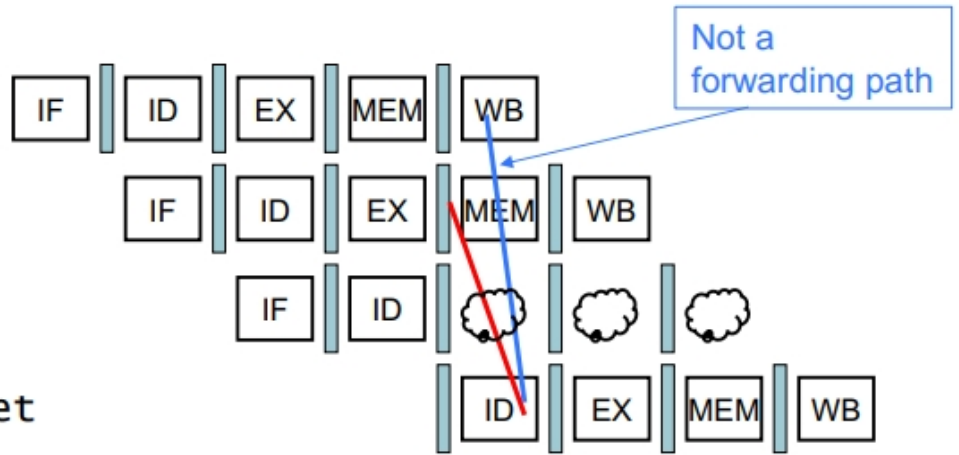
- Forwarding paths are created between stage pipeline register and comparator inputs.
- Need 1 stall cycle even with forwarding.
- Need 2 stall after load instruction.

lw x1, addr

add x4, x5, x6

beq stalled

beq x1, x4, target



3. Branch prediction

- Static prediction
- Dynamic prediction
 - 1-Bit Predictor:
 - 2-Bit Predictor:

