Lab 07:
HW 6

Hunjun Lee
hunjunlee@hanyang.ac.kr

Overview

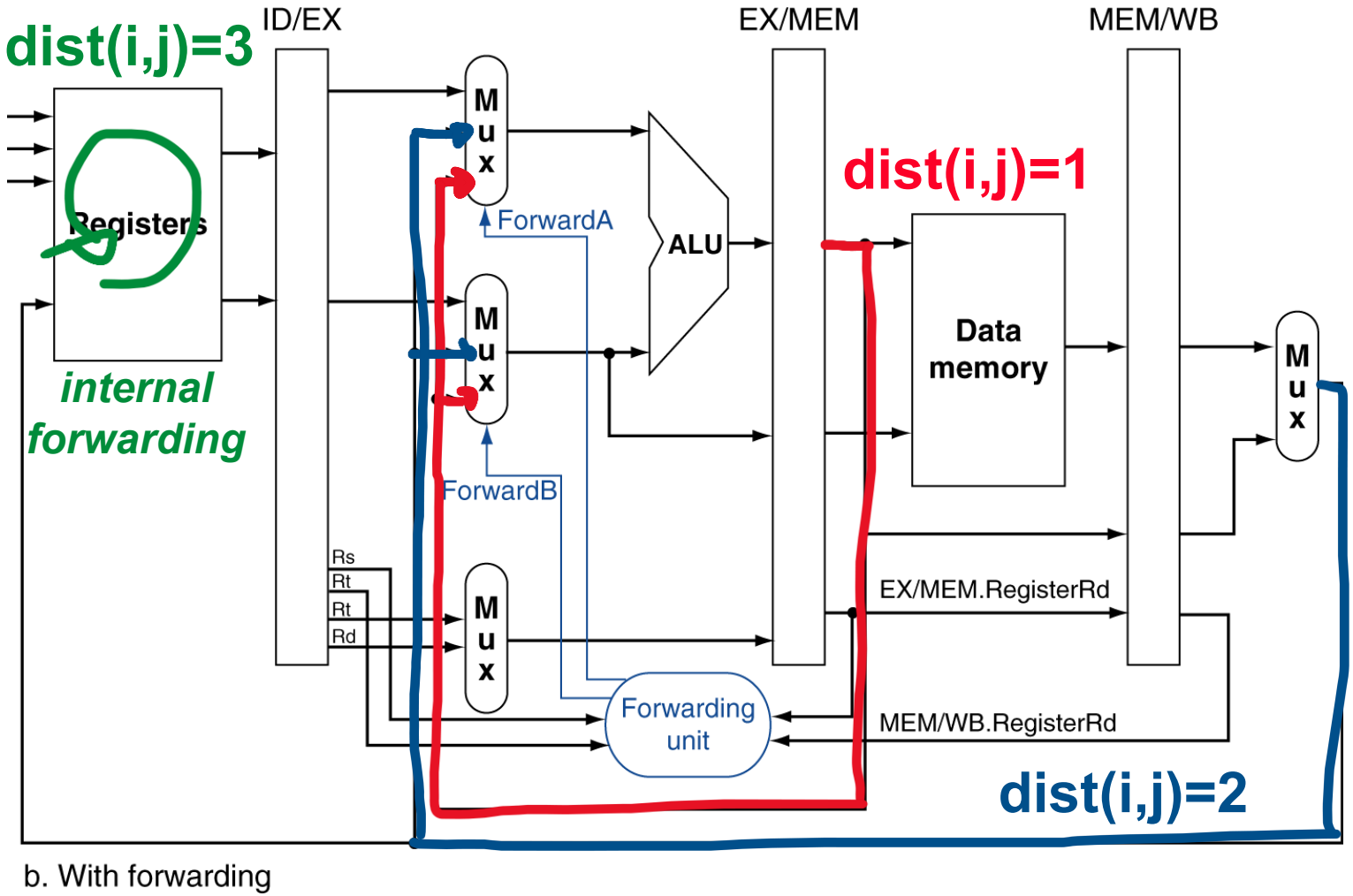
- You will implement additional features on your pipelined CPU

- What to do?
  - Implement a forwarding unit on your pipelined CPU
    - Enable various data forwarding to minimize pipeline stalls
      - ALU operands can be forwarded from prior ALU result
      - ALU operands can be forwarded from data memory
      - Register file supports internal data forwarding

  - Add branch prediction logic to facilitate complicated branch prediction
    - Predict the branch target + branch taken or not-taken

Data forwarding

- You must implement three different data forwarding methods



b. With forwarding

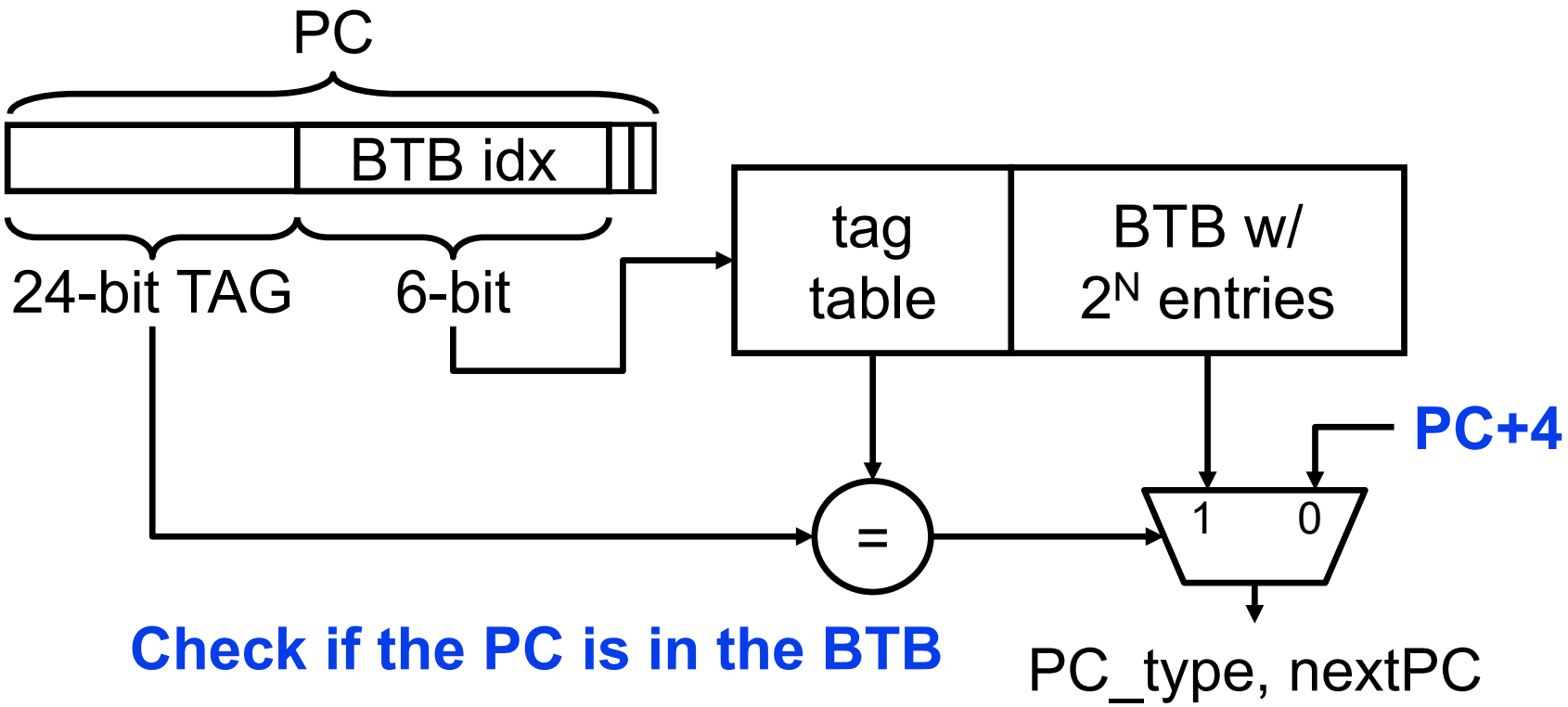
You still need stalls

	R/I-Type	LW	SW	Br	J	Jr
IF						
ID						use
EX	use produce	use	use	use		
MEM		produce				
WB						

- **You know the rules!**

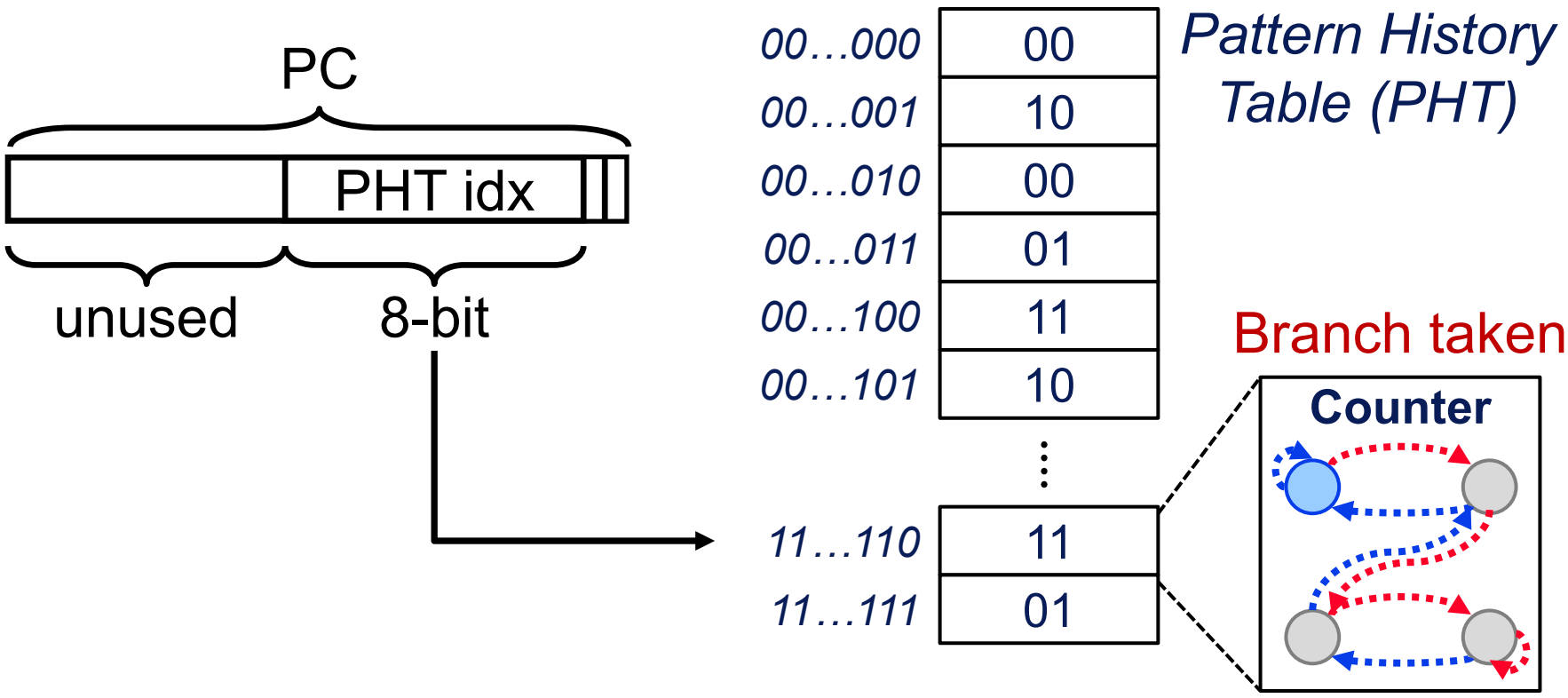
Branch Prediction - BTB

- You need to implement a BTB w/ 24-bit TAG + 6-bit idx (direct-mapped BTB)
- BTB stores two bit to indicate whether the BTB entry is empty (not initialized, jump, or branch)
- You don't need to support BP for JR-type instructions



Branch Prediction - Pattern

- Implement a 2-bit single-level branch predictor **(using saturation counters)**
- The PHT index should be 8-bit (allow multiple branches to share a PHT entry)



Updating BTB + PHT

- Update the branch predictor after the branch resolution
  - @ ID stage or EX stage depending on your implementation

Assignment

- **Files:**
  - **Add custom hardware modules to enable BP + Forwarding!**
  - **Just make sure you modify Makefile if you add additional files**

Submission

- Submission (Zip all the files)
  - For a two-people team: lab6_student_id1_student_id2.zip
  - For a one-person team: lab6_student_id.zip
    - You must follow the format (-10% for wrong file format)
    - Example:
      - lab6_2020102030.zip
      - lab6_2020102030_2022103040.zip
  - The zip file should contain:
    - every cpp/h/v files
    - lab6_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: Mon. June 16th
  - No delay allowed (I need time to grade your project)

- You can submit HW5 by June 16th with only 20% deduction
  - I do not accept additional delay