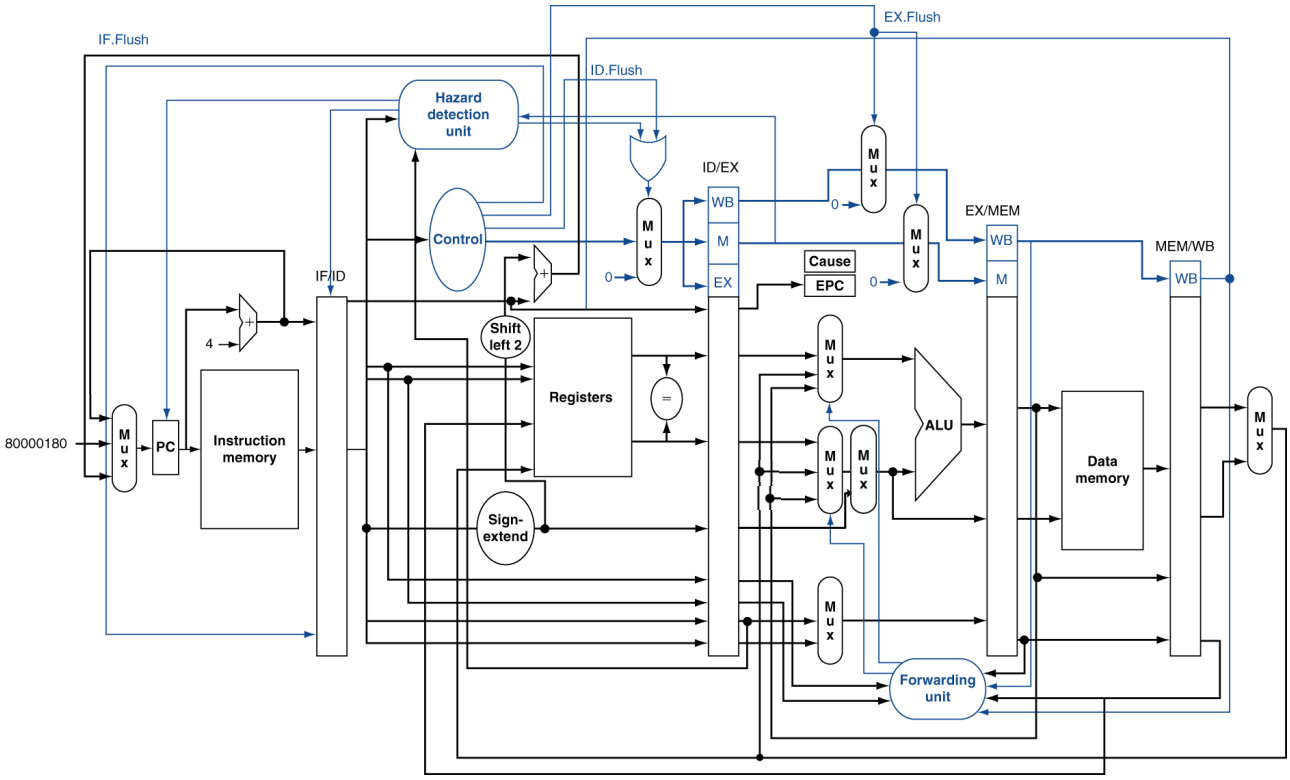
JOINT INSTITUTE
交大密西根学院

Ve370 Introduction to Computer Organization

Homework 5

1. By moving branch decision into the ID stage, the severity of control hazards can be reduced. This approach involves a dedicated comparator in the ID stage at the outputs of the register file, as shown in the following figure.



However, this approach potentially increase the latency of the ID stage, and requires additional forwarding logic and hazard detection. Given the following MIPS instructions :

```
        lw R2,0(R1)
Label1: beq R2,R0,Label2 ; Not taken once, then taken
        lw R3,0(R2)
        beq R3,R0,Label1 ; Taken
        add R1,R3,R1
Label2: sw R1,0(R2)
```

1) Using the first branch instruction in the given code as an example, describe the control hazard detection logic needed to support branch execution in the ID stage. (10 points)

2) For the given code, what is the speedup achieved by moving branch execution into the ID stage? Explain your answer. In your speedup calculation, assume that the additional comparison in the ID stage does not affect clock cycle time. (10 points)
3) Identify data hazards in the given code after moving branch decision into the ID stage. (10 points)
4) Using the first branch instruction in the given code as an example, describe the forwarding path that must be added to support branch execution in the ID stage as well as the conditions selecting the forwarding path. (20 points)

2. The importance of having a good branch predictor depends on how often conditional branches are executed. Together with branch predictor accuracy, this will determine how much time is spent stalling due to mis-predicted branches. Stall cycles due to mis-predicted branches increase the CPI. In this exercise, assume that the breakdown of dynamic instructions into various instruction categories is as follows:

R-Type	BEQ	JUMP	LW	SW
40%	30%	5%	15%	10%

Also, assume the following branch predictor accuracies:
Always taken: 45%; always-not-taken: 55%

1) What is the extra CPI due to mis-predicted branches with the always-taken predictor? Assume that branch outcomes are determined in the EX stage, that there are no data hazards, and that no delay slots are used. (15 points)
2) What is the extra CPI due to mis-predicted branches with the always-not-taken predictor as well as jump instructions? Make appropriate assumptions if necessary. (15 points)

3. Assume the following pattern of branch decisions for a branch instruction: T, NT, NT, T, T, T, T, NT. (T for Taken, NT for Not-Taken). What is the accuracy of a two-bit predictor if this pattern is repeated forever, assuming that the predictor starts off in the “strong” predict not taken state? (20 points)