LAB EXERCISE 3

1. Consider the following MIPS code fragments, each containing two instructions. For each code fragment identify the type of hazard that exists between the two instructions and the registers involved.

a.

LD R1, 0(R2)

DADD R3, R1, R2

- Hazard: Data (RAW). `LD` writes `R1` (WB, cycle 5), `DADD` reads `R1` (ID, cycle 2).

- Registers: `R1`.

- Answer: Data hazard, `R1`.

b.

MULT R1, R2, R3

DADD R1, R2, R3

- Hazard: Data (WAW). Both write `R1` (WB), order matters.

- Registers: `R1`.

- Answer: Data hazard, `R1`.

c.

MULT R1, R2, R3

MULT R4, R5, R6

- Hazard: None. Different registers (`R1` vs. `R4`).

- Answer: No hazard.

d.

DADD R1, R2, R3

SD 2000(R0), R1

- Hazard: Data (RAW). `DADD` writes `R1` (WB, cycle 5), `SD` reads `R1` (MEM, cycle 4).

- Registers: `R1`.

- Answer: Data hazard, `R1`.

e.

DADD R1, R2, R3

SD 2000(R1), R4

- Hazard: Data (RAW). `DADD` writes `R1` (WB, cycle 5), `SD` reads `R1` (MEM, cycle 4).

- Registers: `R1`.

- Answer: Data hazard, `R1`.

2. Explain the behaviour of a 2-bit saturating counter branch predictor. Show the state of the predictor and the transition for each outcome of the branch

- Behavior:

  - States: 00 (strongly NT), 01 (weakly NT), 10 (weakly T), 11 (strongly T).

- Predictions: 00, 01 → NT; 10, 11 → T.

- Transitions:

  - Taken (T): 00→01, 01→10, 10→11, 11→11.

  - Not Taken (NT): 00→00, 01→00, 10→01, 11→10.

- Answer: States 00, 01 predict NT; 10, 11 predict T. T increments, NT decrements counter.

b.) Assuming that every other element of x has the value 0, starting with the first one, show the outcomes of predictions when a 2-bit saturating counter is used to predict the inner branch BNEZ F1, else. Assume that the initial value of the counter is 00

Code:

```
loop: L.D F1, 0(R2)    ; x[i]
      L.D F2, 0(R3)    ; y[i]
      BNEZ F1, else    ; if x[i] != 0
      ADD.D F2, F0, F0 ; y[i] = 0
      BEZ R0, fall     ; always taken
else: DIV.D F2, F2, F1 ; y[i] = y[i] / x[i]
fall: DADDI R2, R2, 8
      DADDI R3, R3, 8
      DSUBI R1, R1, 1
      S.D -8(R3), F2
```

BNEZ R1, loop

- Conditions: `x[0] = 0`, `x[1] != 0`, `x[2] = 0`, ... (alternating). Predict `BNEZ F1, else` with 2-bit counter, initial state 00.

Analysis:

- Outcomes: NT (x[i] = 0), T (x[i] != 0) → NT, T, NT, T, ...

- Counter: Starts at 00 (predict NT), alternates 00↔01.

- Predictions: Always NT (00 or 01).

- Correct: NT (correct), T (incorrect).

Prediction Table (first 4 iterations):

i	x[i]	Outcome	State	Prediction	Correct	New State
0	0	NT	00	NT	Yes	00
1	!= 0	T	00	NT	No	01
2	0	NT	01	NT	Yes	00
3	!= 0	T	00	NT	No	01

Answer: Predicts NT always, correct for NT (i = 0, 2, ...), incorrect for T (i = 1, 3, ...). States: 00→01→00→01.

Final Answers

E1:

- a. Data hazard, `R1`.

- b. Data hazard, `R1`.

- c. No hazard.

- d. Data hazard, `R1`.

- e. Data hazard, `R1`.

E2:

- a: 2-bit counter: 00, 01 (NT); 10, 11 (T). T: 00→01→10→11; NT: 11→10→01→00.

- b: `BNEZ F1`: Outcomes NT, T, NT, T, ...; predicts NT (00↔01); correct for NT, incorrect for T.