NAME: VIVIAN KERUBO MOSOMI
REGISTRATION NUMBER: SCT212-0062/2021
UNIT: COMPUTER ARCHITECTURE(BCT 2408)

Lab 3
I) E1
Problem

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.

Solution
The types of data hazards include:
i) RAW(Read After Write) - When an instruction reads an operand before a prior instruction has written to it,causing a delay in the pipeline,
ii) WAW(Write After Write) - Occurs when an instruction writes to a register before a previous instruction writes to it.
iii) WAR(Write After Read) - When an instruction writes to a register before a previous instruction has read it

a.
```
LD R1, 0(R2)
DADD R3, R1, R2
```

Hazard: RAW
LD instruction loads data into register R1 from memory, and the DADD instruction reads the value from register R1 to perform an addition operation. This creates a data dependency since the DADD instruction is dependent on the value that is being loaded into R1 in the previous instruction. Thus a RAW hazard because DADD needs the value of R1 which is not available until the LD instruction completes.

b.
```
MULT R1, R2, R3
DADD R1, R2, R3
```

Hazard:WAW
The MULT instruction writes to R1 where the result of the multiplication of R2 and R3 is stored, and the DADD instruction also writes to R1. A WAW hazard occurs because the two instructions write to R1, and the second instruction could overwrite the first before it has been fully completed.

c.
```
MULT R1, R2, R3
MULT R4, R5, R6
```

There is no hazard since there's no data dependency as both instructions perform multiplication instruction on different registers. The first instruction performs the instruction on registers R1,R2 and R3 while the second performs on R4,R5 and R6.

d.
```
DADD R1, R2, R3
SD 2000(R0), R1
```

Hazard: RAW
This is because SD stores the value of R1 after a DADD operation.It may store the value before R1 is ready.

e.
```
DADD R1, R2, R3
SD 2000(R1), R4
```

Hazard: RAW
SD operation uses register R1 as an address but R1 is written by the DADD operation.Thus SD must first wait for the address to be computed

II) E2
Problem
a. 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.

Solution
The 2-bit saturating counter is a mechanism used by branch predictors to tell if a branch like an if or a loop will be Taken or Not Taken in future predictions.
The possible states are 00,01,10 and 11.

The predictions for the above states are:
00 - Not Taken
01 - Not Taken
10 - Taken
11 - Taken

The transition for the branch outcome depends on if:
i) If the branch is taken:
00 transitions to 01
01 transitions to 10
10 transitions to 11
11 still stays at 11.It does not transition.

ii) If the branch is not taken:
00 stays at 00 and does not transition.
01 transitions to 00
10 transitions to 01
11 transitions to 10

A table showing the current state,the prediction,outcome and the state the branch has transitioned to.

Current State	**Prediction**	**Branch Outcome**	**New State**
00	Not Taken	Not Taken	00
00	Not Taken	Taken	01
01	Not Taken	Not Taken	00
01	Not Taken	Taken	10
10	Taken	Not Taken	01
10	Taken	Taken	11
11	Taken	Not Taken	10
11	Taken	Taken	11

b. Consider the following code:

```
for (i=0; i<N; i++)
if (x[i] == 0)
y[i] = 0.0;
else
y[i] = y[i]/x[i];
```

Assume that the assembly code generated is then:

```
loop: L.D F1, 0(R2)
L.D F2, 0(R3)
BNEZ F1, else
ADD.D F2, F0, F0
BEZ R0, fall
else: DIV.D F2, F2, F1
fall: DADDI R2, R2, 8
DADDI R3, R3, 8
DSUBI R1, R1, 1
S.D -8(R3), F2
BNEZ R1, loop
```

where:

• the value of N is already stored in R1
• the base addresses for x and y are stored in R2 and R3, respectively
• register F0 contains the value 0
• register R0 (always) contains the value 0

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.

Solution

Initial Counter Value	**Prediction**	**Outcome if Taken**	**Outcome if Not Taken**
00	Not Taken	Transitions to 01	Stays at 00
01	Not Taken	Transitions to 10	Transitions to 00
10	Taken	Transitions to 11	Transitions to 01
11	Taken	Stays at 11	Transitions to 10