**Computer Architecture Lab 3**

**Name: Eddy Kimani**

**Reg No.: SCT212-0596/2021**

**E1: MIPS**

1. LD R1, 0(R2)

DADD R3, R1, R2

* Hazard: RAW (Read After Write)
* Registers Involved: R1
* Explanation: LD writes to R1, and DADD immediately reads R1. This creates a RAW hazard if forwarding isn't used.

1. MULT R1, R2, R3

DADD R1, R2, R3

* Hazard: WAW (Write After Write)
* Registers Involved: R1
* Explanation: Both MULT and DADD write to R1. If MULT takes more cycles than DADD and finishes *after* DADD, it can overwrite the newer result.

1. MULT R1, R2, R3

MULT R4, R5, R6

* Hazard: Structural and possibly control, depending on the hardware
* Registers Involved: None directly (registers are different)
* Explanation: If the hardware has only one multiplier, issuing two MULT instructions back to back creates a structural hazard — the second cannot start until the first finishes.

1. DADD R1, R2, R3

SD 2000(R0), R1

* Hazard: RAW
* Registers Involved: R1
* Explanation: DADD writes to R1, and SD reads from R1 to store its value in memory. This is a RAW hazard.

1. DADD R1, R2, R3

SD 2000(R1), R4

* Hazard Type: RAW
* Registers Involved: R1
* Explanation: Here, R1 is used as a base address in the store. DADD writes to R1, but SD uses R1 to compute the effective memory address. This is also a RAW hazard.

**E2: Counter branch predictor**

1. 2-bit saturating counter branch predictor

|  |  |  |  |
| --- | --- | --- | --- |
| Current value | Prediction | Outcome | New value |
| 00 | NT | NT | 00 |
| 00 | NT | T | 01 |
| 01 | NT | NT | 00 |
| 01 | NT | T | 10 |
| 10 | T | NT | 01 |
| 10 | T | T | 11 |
| 11 | T | NT | 10 |
| 11 | T | T | 11 |

1. 2-bit counter prediction rate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteration | Current value | Prediction | Outcome | New value |
| 1 | 00 | NT | NT | 00, hit |
| 2 | 00 | NT | T | 01, miss |
| 3 | 01 | NT | NT | 00, hit |
| 4 | 00 | NT | T | 01, miss |