b10902052 lab3 Report

Module Explanations

- The following modules are simply copied from lab2. I will provide brief explanations for each:
  - `Adder.v` : Adds two values.
  - `ALU_Control.v` : Sends control signals to `ALU.v` .
  - `AndUnit.v` : Performs the AND operation on two bits.
  - `Control.v` : Determines control bits for all processes.
  - `EXMEM.v` , `MEMWB.v` : Pipeline registers.
  - `FwdUnit.v` : Forwarding control unit.
  - `HazardUnit.v` : Hazard control unit.
  - `ImmGen.v` : Generates immediate values.
  - `LeftShifter.v` : Shifts one bit left.
  - `MUX32.v` , `MUX32_4.v` : Multiplexing modules.
- The following modules are majorly copied from lab2 with some differences:
  - `IFID.v` : Adds a `pcnxt` register to store and pass the next PC value (PC+4) between the IF-stage and ID-stage for branch usage.
  - `IDEX.v` : Adds `Branch` (for checking if the current instruction is `beq`), `Decide` (indicates whether the branch is taken), `SEPC` (a register saving the PC value if an exception occurs to recover the PC value, updated only when `Branch` is true), and a `flush` bit for executing the flush operation.
  - `ALU.v` : Adds 1 input `Branch` to get the branch control bit. If true, the operation switches to `Data1-Data2` . Also adds an output bit `eq` for checking if the output is 0.
- **CPU.v**
  - Added the following datapath to complete this lab.
  - In the IF-stage, a mux is added to decide if an exception occurs.
  - At the ID-stage, `EqualUnit.v` is replaced by predictions from `PredictUnit.v` . Additionally, a mux is introduced to decide which value needs to be saved to the `SEPC` register.
  - At the EX-stage, the result of the branch operation is obtained using the `eq` value from `ALU.v` , the actual decision from the previous prediction , and the

branch control bit of the current instruction. All these values are then sent to `PredictUnit.v` for utilization.

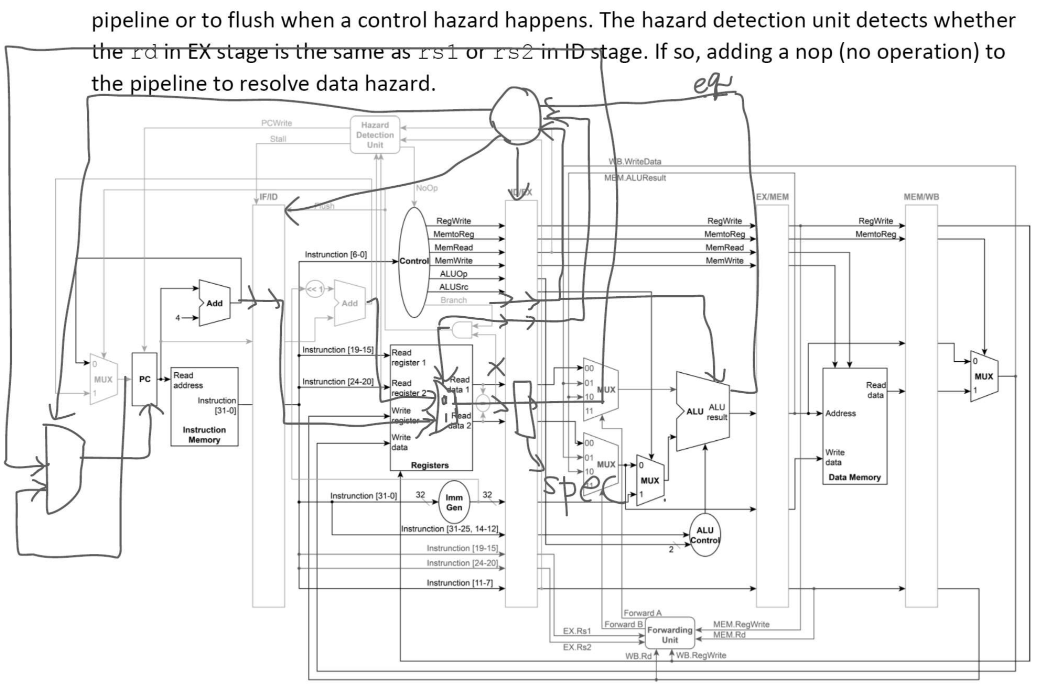
pipeline or to flush when a control hazard happens. The hazard detection unit detects whether the rd in EX stage is the same as rs1 or rs2 in ID stage. If so, adding a nop (no operation) to the pipeline to resolve data hazard.

Figure 5 Final Datapath after Adding Hazard Detection Unit (colored in orange)

- **PredictUnit**
  - This is a major part of this homework, so I will explain in detail.
  - Inputs:
    - `Branch_i` : Indicates whether the current instruction is a branch instruction.
    - `Decide_i` : Represents the actual branch operation.
    - `Eq_i` : The result of `ALU.v` is equal to zero.
  - Outputs:
    - `IFID_flush_o` : Indicates whether to flush the IFID pipeline stage.
    - `IDEX_flush_o` : Indicates whether to flush the IDEX pipeline stage.
    - `recover_o` : Control bit of the mux in the IF-stage.
    - `predict_o` : The predicted outcome of the branch.
  - Variables:
    - `status_value` : Stores a 2-bit value, indicating the current state.
  - Update the `status_value` when the clock is at posedge, referenced by the supplied automata.
  - Update others on any input change. The major idea is if the prediction is correct, continue to execute; if not, flush IFID and IDEX and reset PC.

Difficulties Encountered and Solutions in This Lab

- Difficulties

    1. I observed that the output of `ALU.v` is always positive in the sample `output.txt`, but the ALU copied from lab2 is signed.
    2. I noticed that the sample `output.txt` ends with `0x0D0A` (which represents `\r\n` in hexadecimal) while my output from the testbench doesn't (it's `0x0A`). As a result, the `diff` command consistently causes errors.
- Solutions
    1. To address the first issue, I opted to convert the output to unsigned.
    2. To resolve the second issue, I utilized the `diff -Z` command, which ignores whitespace differences.

Development Environment

- OS :
    - Distributor ID: Ubuntu
    - Description: Ubuntu 22.04.2 LTS
    - Release: 22.04
    - Codename: jammy
- compiler :
    - iverilog :
        - Icarus Verilog version 11.0 (stable)
- IDE :
    - vscode
        - 1.83.1
        - f1b07bd25dfad64b0167beb15359ae573aecd2cc
        - x64