

## HW9 Q4 Documentation

### Control Hazard – control\_haz.y8:

In this program, we have a control hazard in which *jne* is called on line 11 causing the program to jump to "target", where the first command is *halt*. However, this jump should not be taken. Since it will be taken by default, the pipelined implementation will fetch and start two commands from target (one of which is *halt*), before realizing that it should not have jumped. The correct implementation will turn the two instructions loaded from "target" into bubbles, and therefore the program will not *halt* as per the instruction on line 26, but will continue through with the "done" instructions. The incorrect (broken) implementation of pipelining will actually execute the two instructions loaded from "target", thereby halting the program prematurely. This is a control hazard since the wrong instructions are executed. See ch.pdf for an annotated walkthrough of this process with screenshots of the two versions of psim (std and broken).

### Data Hazard – data\_haz.y8:

In this program, we have a data hazard in which the value of `%ebx` at the end of the program should be equal to 23, but a poorly-pipelined y86 processor will not have written the most recent values of the two operand registers (`%eax` and `%ebx`) to the registers by the time that the *addl* instruction is decoded. In the correct version, forwarding is used to make it so that the result of the *addl* instruction uses the correct (most recent) values of written to the operand registers. See dh.pdf for an annotated walkthrough of this process with screenshots of the two versions of psim (std and broken).