Lab7 Chen-Yu Chang

In the hazardunit module, I applied a mux that would store the IF process when the instruction in EX stage is lw and its Rt equal to the Rt or Rs of instruction in ID stage. For forwarding, in the forwardmux module I output the mux select forwardA and forwardB, which will output the corresponding select when forwarding hazard in MEM or WB stage occurs. I then applied these two forwarding signals as mux into two 3to1 mux, that output the inputs of ALU module.

A hazard occurs in instruction 4, 5, 6:

addi R4, 146

addi R5, 5

add R5, R1, R4

The whole module successfully added R4 with the correct result 423 to R1, 423, and saved the result 569 to R5 without causing stalls.

Another hazard occurs in instruction 6, 7:

add R5, R1, R4

slt R6, R3, R5

It is clear that the module managed to compare R5 in 569 instead of incorrect 5 to R3 in 13, and saved the correct result 1 to R6

The next hazard occurs in instruction 7,8,9:

slt R6, R3, R5

lw R4, 4(R0)

sub R7, R4, R6

The module successfully stalled under lw condition and applied R4 and R6 in latest result 4 and 1, and returned the correct answer 3 to R7.

The next hazard occurs in instruction 9,10:

sub R7, R4, R6

sw R7, 2(R0)

It is obvious that the module managed to save the correct result 3 instead of 0 into mem[0].

The final hazard occurs in instruction 9,10,11:

sub R7, R4, R6

sw R7, 2(R0)

add R8, R7, R2

The module managed to add the new R7 in 3 with R2 in 92 and return the result 95 to R8.

