EC 413

Pipelined CPU - Forwarding Lab 9 Report

Ruben Carbajal
Claire Cropper

Hazard Detection Unit Modification

In order to modify our hazard detection unit from Lab 8, we modified it to compare the different stages of the CPU and only detect load use hazards.

Load-Use Hazard Logic

In the three variables being set, IFIDWrite, PCWrite are intended to output 1 if no hazards occur and HazardMux will output 0. Although using the logic (IDEX_rt == IFID_rt || IDEX_rt == IFID_rs) && memread we are able to change the respective outputs that detect the error. This logic is saying, if we are reading from memory and if the rt register in the IDEX stage match with the Rt or Rs register in the IFID stage then change the output for hazard

Forwarding Mux Unit Modification

1 Ahead Forwarding for A and B

For 1 ahead forwarding, we checked if (EX_MEM_RegWrite && (EX_MEM_Rd == ID_EX_Rs)) was true then forwardA returns 2'b10. A similar case with fowardB where (EX_MEM_RegWrite && (EX_MEM_Rd == ID_EX_Rs)) was met then we also return 2'b10. A and B have the same logic because they are both for the same thing just from a different register.

2 Ahead Forwarding for A and B

For 2 ahead forwarding, we checked if (MEM_WB_RegWrite && (MEM_WB_Rd == ID_EX_Rs)) was true then forwardA returns 2'b10. A similar case with fowardB where (MEM_WB_RegWrite && (MEM_WB_Rd == ID_EX_Rs)) was met then we also return 2'b10. A and B have the same logic because they are both for the same thing just from a different register.

CPU Modification

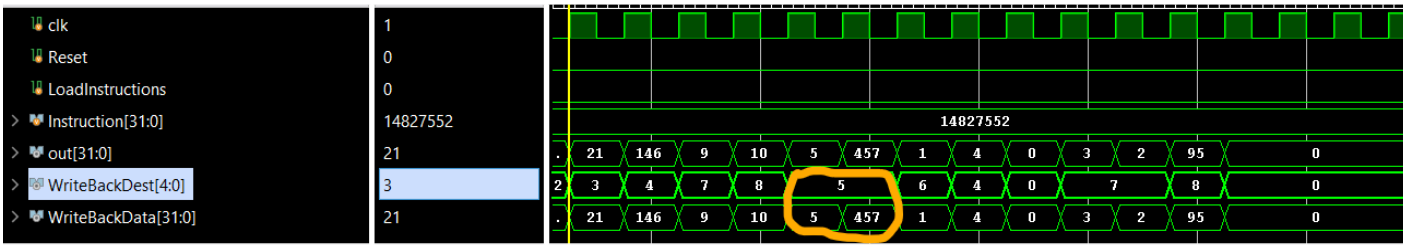
The main edits we did to the CPU was the change of Hazard Unit and the addition of the forwarding mux. Also we created forwarding wires that source into the ALU that come from the forwarding unit. Also implementing the values of ALU_a and ALU_b into parts of the execution and memory stages.

Waveforms

1 ahead Hazard Waveform

EX 1:

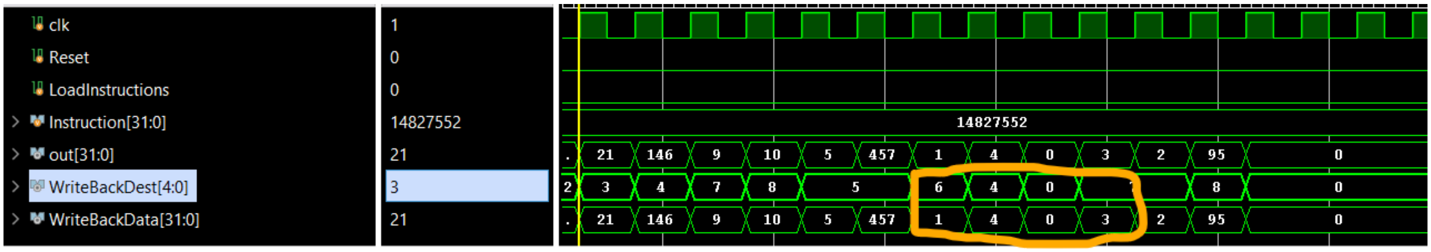
addi r5, 5
add r5, r1, r2 (457 = 365 + 92)



We can see register 5 immediately updates the next cycle showing the forwarding if working with 1 ahead

EX 2:

lw r4, 4(r0)
sub r7, r4, r6 (4 - 1 = 3)

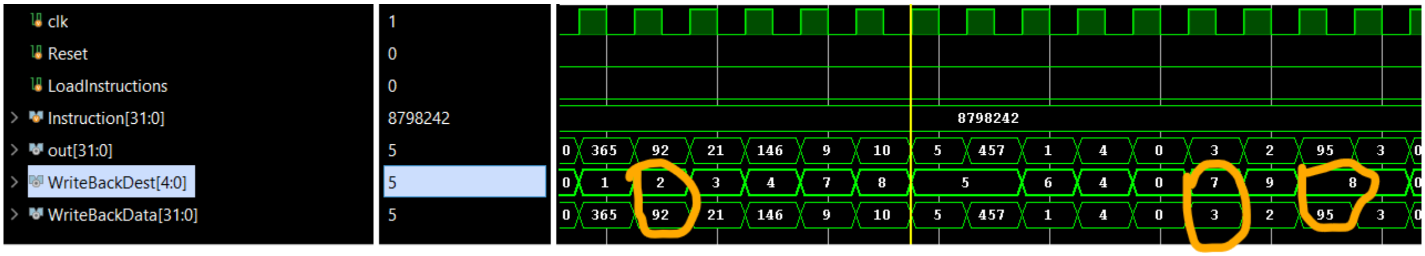


We see that even though Register 4 was just updated in the register before, it still operates under the correct instructions. It does the stall because it is a LW instruction but still forwards it all correctly.

2 ahead Hazard Waveform

EX 1:

sub r7, r4, r6
sw r9, 2(r0)
add r8, r7, r2 (95 = 92 + 3)



This is an example of a 2 ahead error since the value of the new Register 7 is correctly doing the addition from R7 and R2.