Another hazard occurs when certain data is loaded from memory into register and immediately required in the next instruction:

addi $t1,$zero,123

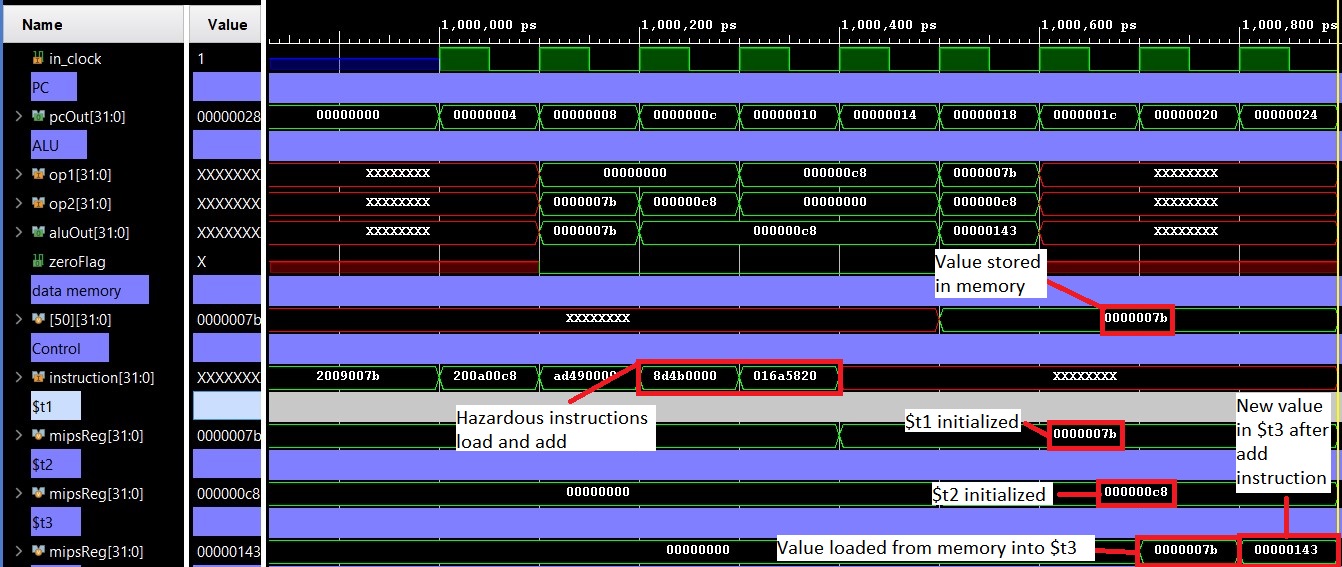
addi $t2,$zero,200

sw $t1,($t2)

lw $t3,($t2)

add $t3,$t3,$t2

The last two instructions pose a hazard. We have solved this by forwarding the output from the data memory from the memory stage, back to the input of the ALU. The input MUXs are appropriately controlled by forwarding unit.



A variation of the previous hazard is loading data from the memory and accessing the register one instruction after:

addi $t1,$zero,123

addi $t2,$zero,200

sw $t1,($t2)

lw $t3,($t2)

add $t9,$zero,$zero

add $t3,$t3,$t2

This hazard is solved by forwarding the output from data memory, taken from the writeback stage, into the inputs of the ALU. The input MUXs are appropriately controlled by forwarding units. (image below)

