Branch Test Case Analysis:

* (Fixed Assembly file by adding nop’s of each incremental step is attached)
* Hazards in the testcase:
  1. JMP R1 - Control Hazard
  2. AND R1, R5, R5 - Control Hazard Because of the Interrupt
  3. JZ R2 - Control Hazard
  4. JZ R3 - Control Hazard
  5. NOT R5

INC R5 - Data Hazard

* 1. In R6

JZ R6 - Data and Control Hazard

* 1. RTI - Control Hazard
  2. Call R6 - Control Hazard because of the interrupt
  3. RET - Control Hazard

1. Without Forwarding and Hazard Detection Unit:
   * All Hazards were present:

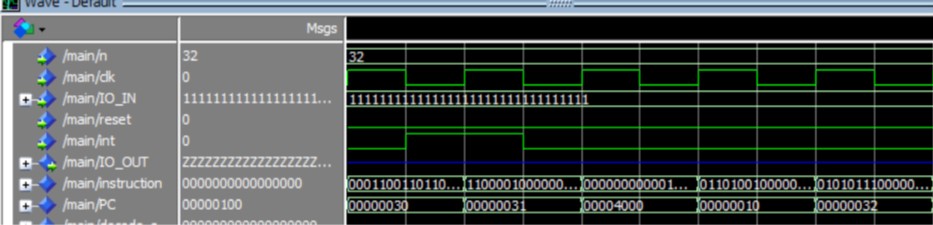


Figure 1: undefined behaviour on interrupt at 30 and instruction at 32 was fetched and excuted

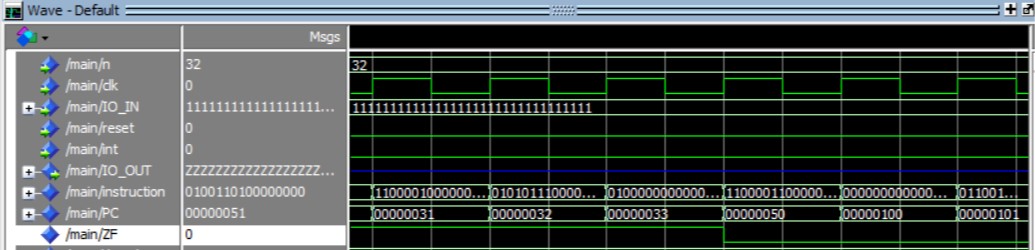


Figure 2: JZ at 31 was non-taken and the JZ at 50 was taken. and wrongly fetched instructions wasn't flushed

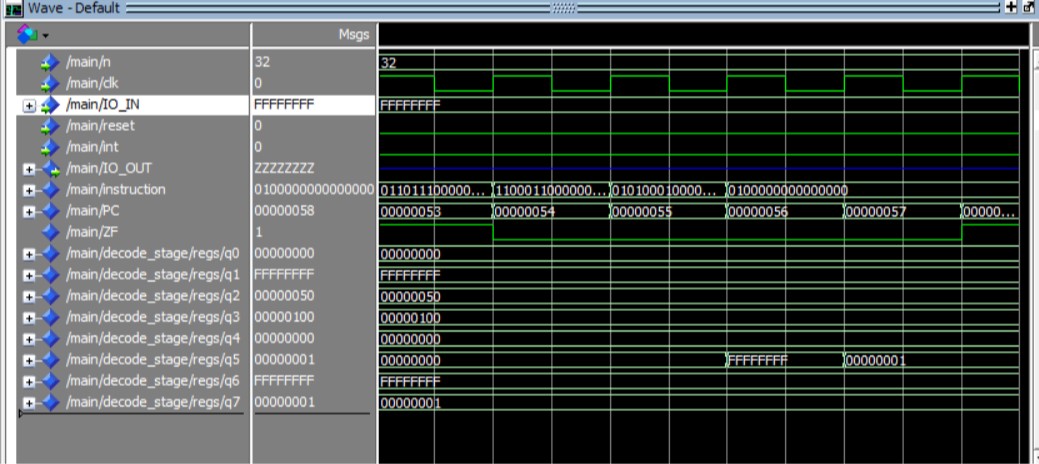


Figure 3: Data Hazard at R5 where an old value was used

* + To solve them using no operation:
    1. The JMP at 1, JZ at 3 and 4: add two nop after them to allow the new address calculation to take place without fetching wrong instructions
    2. Interrupt at 2 and 8: add 4 nop after them because the interrupts happen on 2 cycles and the second cycle is like a load-use case.
    3. The RTI at 7 and ret at 9: add 3 nop after them because it’s a load-use case where the PC is loaded from the memory
    4. The data Hazards at 5 and 6: add two nop between the two instructions
    5. The control hazard at 6: add two nop after the jump to allow the new address calculation to take place without fetching wrong instructions

1. With Forwarding Unit but no Hazard Detection:
   * All the hazards are still present except: Data Hazard at the INC R5 at 5 is solved using forwarding:

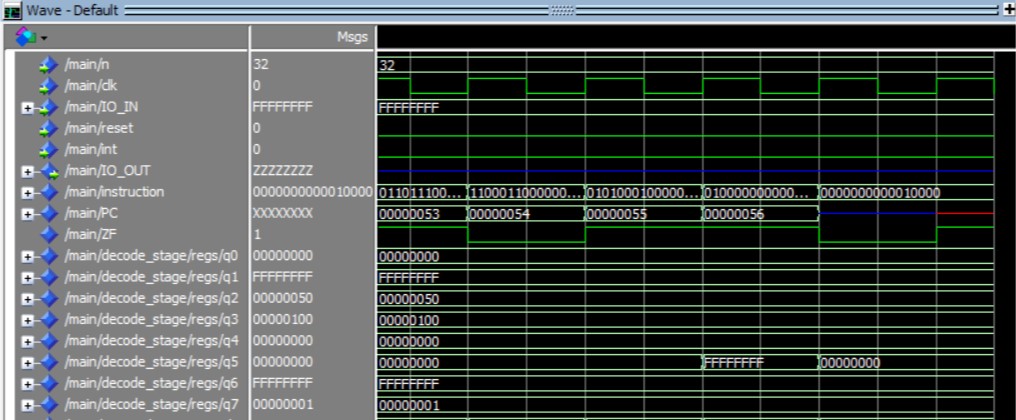


Figure 4: Data Hazard Solved using Forwarding unit

* + Data Hazard at 6 is not solved because branch unit is the one using the R6 value not the alu.

1. With Hazard Detection Unit but without the PC prediction:
   * The Data Hazard at 6 is handled by the stalling of the hazard unit.
   * The rest of the Hazards are still present.
2. With everything on:
   * All the Control Hazards are solved using branch prediction and flushing:

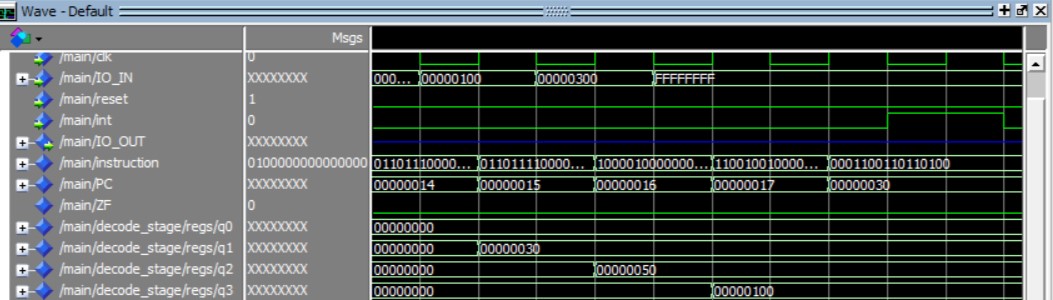


Figure 5: the control hazard of JMP at 17

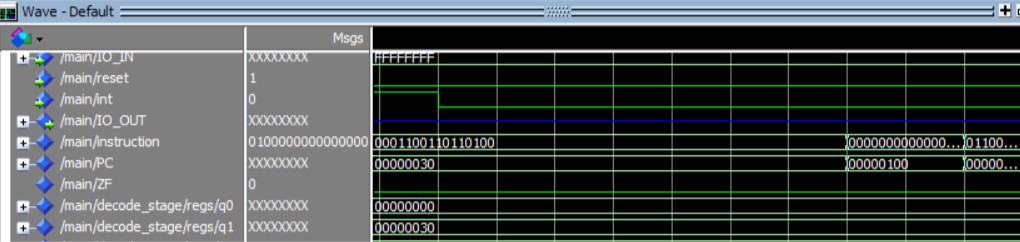


Figure 6: the control hazard of the interrupt handled by hardware stalling

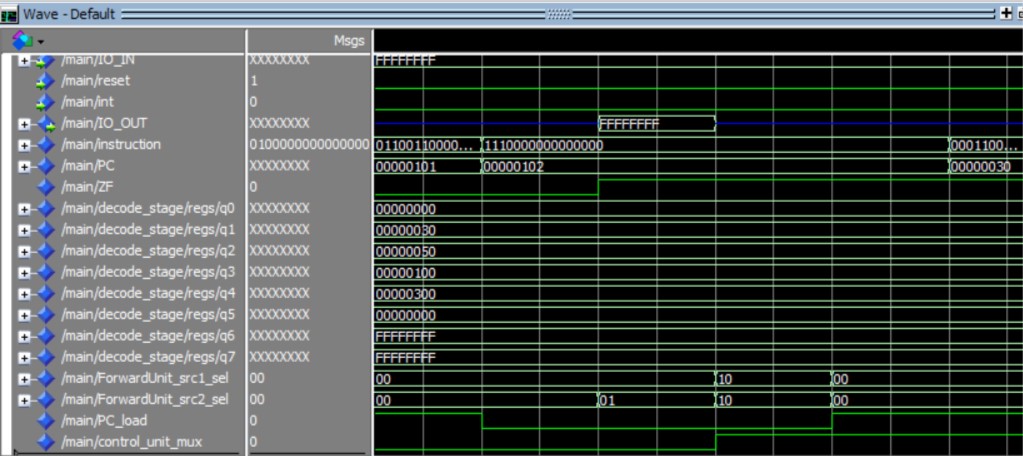


Figure 7: the control hazard at 102

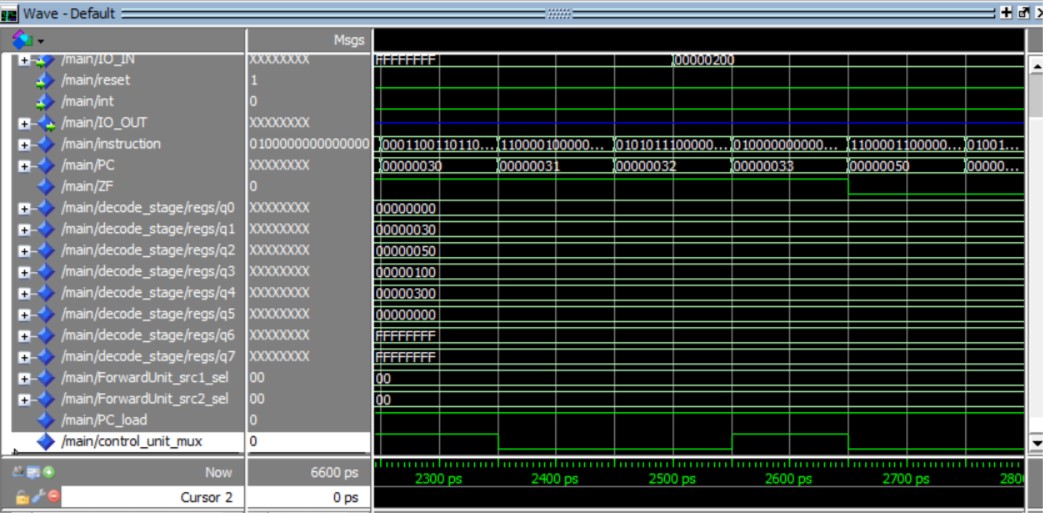


Figure 8: the instruction 32 flushed by "control\_unit\_mux" signal

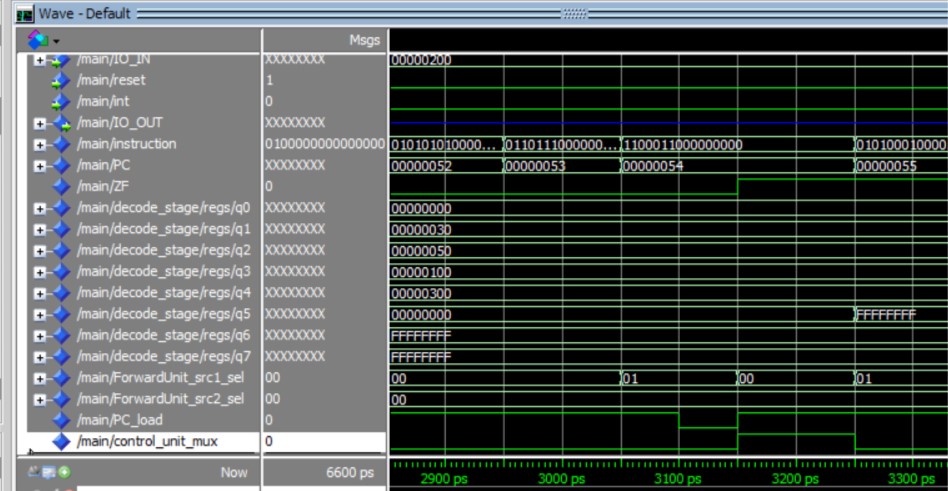


Figure 9: the control hazard at 54 JZ R6 by hardware stall

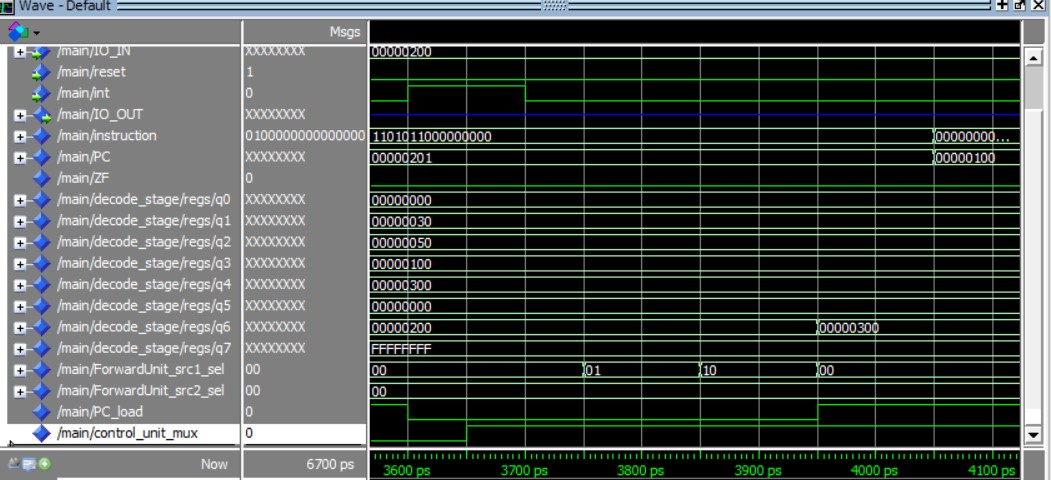


Figure 10: the interrupt signal at 201 handled using hardware stall

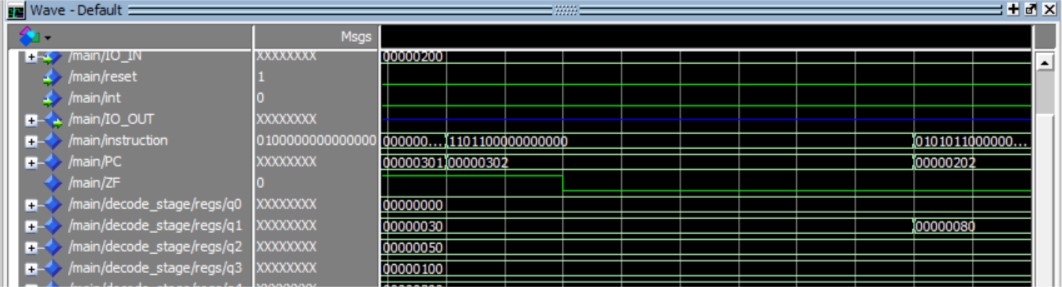


Figure 11: the control hazard at 302 handled using hardware stalls making sure the INC R7 is not executed.