**Initial State**

- **Registers:** R0=0, R3=10, R5=50, R6=25.
- **Data Memory:** Mem[0] = 77.
- **Program Counter (PC):** 0.

---

**Cycle 1 (Time: 15ns)**

- **IF (Instruction Fetch):** The processor fetches the first instruction from memory address 0.
    - **Instruction:** I1: LOAD R1, 0(R0)
    - The PC is updated to 4.
- **ID, EX, WB:** These stages are empty.

---

**Cycle 2 (Time: 25ns)**

- **IF:** Fetches the second instruction from memory address 4.
    - **Instruction:** I2: ADD R2, R1, R3
    - The PC is updated to 8.
- **ID (Instruction Decode):** Decodes the LOAD instruction. It reads the value from R0 (which is 0).
- **EX, WB:** Empty.

---

**Cycle 3 (Time: 35ns) - Hazard Detected**

- **IF:** Fetches the third instruction from memory address 8.
    - **Instruction:** I3: SUB R4, R5, R6
    - The PC is updated to 12.
- **ID:** Decodes the ADD instruction. It needs to read registers R1 and R3.
    - **HAZARD:** The Hazard Unit detects that this ADD instruction needs to read R1, but the LOAD instruction currently in the EX stage is the one that will produce the value for R1. The correct value is not available yet.
    - **Action:** The Hazard Unit asserts the **Stall** signal.
- **EX (Execute):** Executes the LOAD instruction. The ALU calculates the memory address: value(R0) + 0 which is 0.
- **WB (Write Back):** Empty.

---

**Cycle 4 (Time: 45ns) - Pipeline Stall**

- **IF:** This stage is **frozen** because of the stall. The PC is not updated (it stays at 8), and the SUB instruction is held.
- **ID:** This stage is also **frozen**. The ADD instruction is held and does not move forward.
- **EX:** A **bubble (NOP)** is inserted into this stage. This is the crucial delay that allows the LOAD to catch up.
- **WB:** The LOAD instruction moves to this stage. It takes the address calculated in the previous cycle (0), reads the value 77 from Data Memory, and prepares to write it.

---

**Cycle 5 (Time: 55ns) - Pipeline Resumes**

- **IF:** The stall is over. The stage fetches the instruction at PC address 8 again.
    - **Instruction:** I3: SUB R4, R5, R6
    - The PC is updated to 12.
- **ID:** The ADD instruction, which was held, now proceeds. It can safely read the registers.
- **EX:** The bubble (NOP) from the previous cycle moves here and does nothing.
- **WB:** The LOAD instruction completes. **The value 77 is written into register R1.** The hazard is now fully resolved.

---

**Cycle 6 (Time: 65ns)**

- **IF:** Fetches from address 12. Since there are no more instructions, it fetches a NOP (all zeros). PC becomes 16.
- **ID:** Decodes the SUB instruction. Reads values from R5 (50) and R6 (25).
- **EX:** Executes the ADD instruction. The ALU calculates value(R1) + value(R3) which is 77 + 10 = 87.
- **WB:** The bubble is in this stage. No register is written.

---

**Cycle 7 (Time: 75ns)**

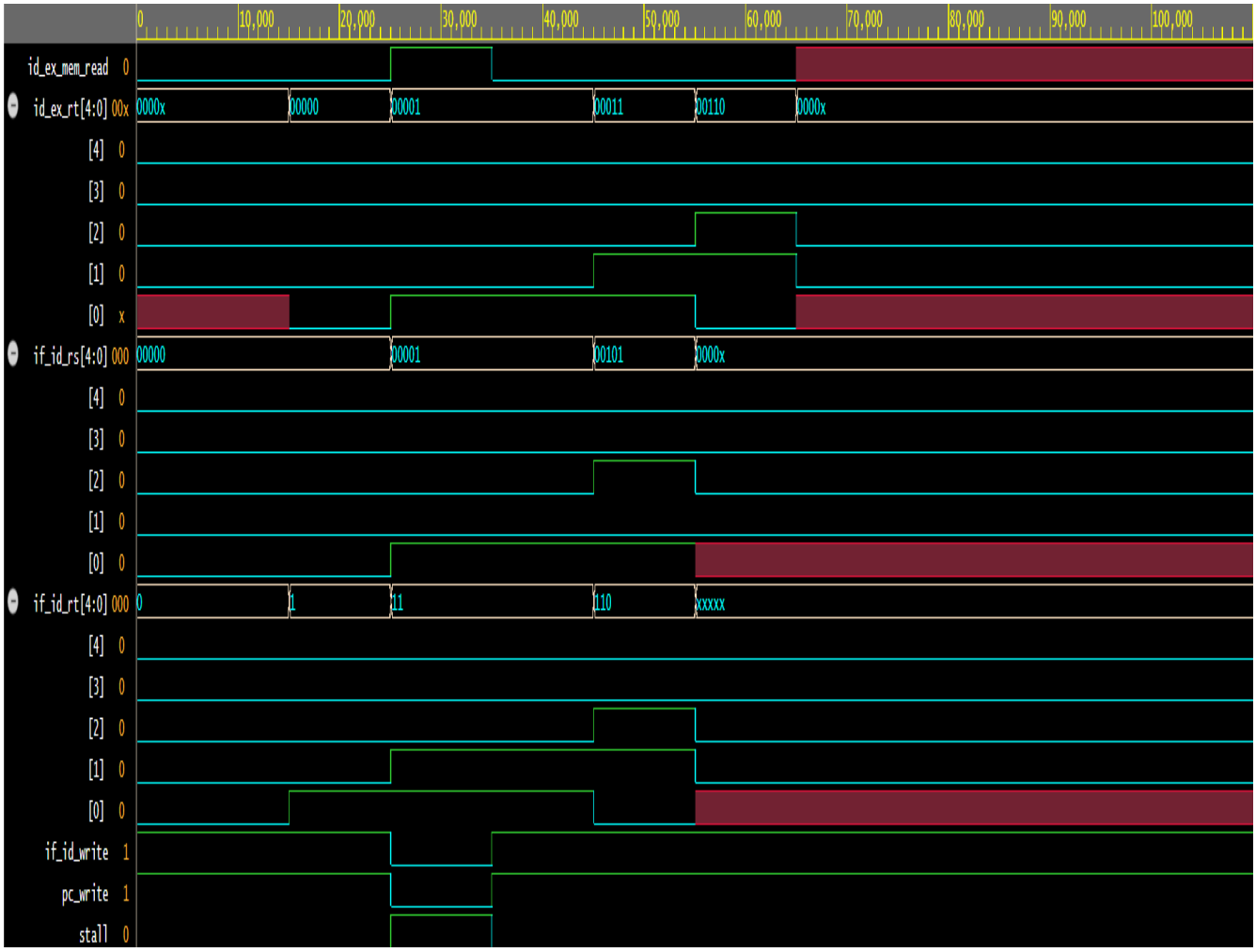
- **IF:** Fetches a NOP. PC becomes 20.
- **ID:** Decodes the NOP.
- **EX:** Executes the SUB instruction. The ALU calculates value(R5) - value(R6) which is 50 - 25 = 25.
- **WB:** The ADD instruction completes. **The value 87 is written into register R2.**

---

**Cycle 8 (Time: 85ns)**

- **IF:** Fetches a NOP. PC becomes 24.
- **ID:** Decodes the NOP.
- **EX:** The NOP moves here.
- **WB:** The SUB instruction completes. **The value 25 is written into register R4.**

The remaining cycles will just flush the NOPs out of the pipeline until it is empty.




#############log for Verilog code############################

Time=0, PC= 0, Stall=0, IF/ID_Inst=00000000, ID/EX_Inst=0, EX/WB_Inst=0, WB_Inst=Xxxxxxxxxx
Time=15000, PC= 4, Stall=0, IF/ID_Inst=8c010000, ID/EX_Inst=a, EX/WB_Inst=0, WB_Inst=Xxxxxxxxxx
Time=25000, PC= 8, Stall=1, IF/ID_Inst=00231020, ID/EX_Inst=c, EX/WB_Inst=2, WB_Inst=20xxxxxxxx
Time=35000, PC= 8, Stall=0, IF/ID_Inst=00231020, ID/EX_Inst=0, EX/WB_Inst=3, WB_Inst=20xxxxxxxx

Time=45000, PC= 12, Stall=0, IF/ID_Inst=00a62022, ID/EX_Inst=a, EX/WB_Inst=0, WB_Inst=00xxxxxxx
Time=55000, PC= 16, Stall=0, IF/ID_Inst=xxxxxxxx, ID/EX_Inst=a, EX/WB_Inst=2, WB_Inst=22xxxxxxx
Time=65000, PC= 20, Stall=0, IF/ID_Inst=xxxxxxxx, ID/EX_Inst=X, EX/WB_Inst=2, WB_Inst=24xxxxxxx
Time=75000, PC= 24, Stall=0, IF/ID_Inst=xxxxxxxx, ID/EX_Inst=X, EX/WB_Inst=x, WB_Inst=xxxxxxxxx
Time=85000, PC= 28, Stall=0, IF/ID_Inst=xxxxxxxx, ID/EX_Inst=X, EX/WB_Inst=x, WB_Inst=xxxxxxxxx
Time=95000, PC= 32, Stall=0, IF/ID_Inst=xxxxxxxx, ID/EX_Inst=X, EX/WB_Inst=x, WB_Inst=xxxxxxxxx
Time=105000, PC= 36, Stall=0, IF/ID_Inst=xxxxxxxx, ID/EX_Inst=X, EX/WB_Inst=x, WB_Inst=xxxxxxxxx