**Lab 8 Computer Architecture**

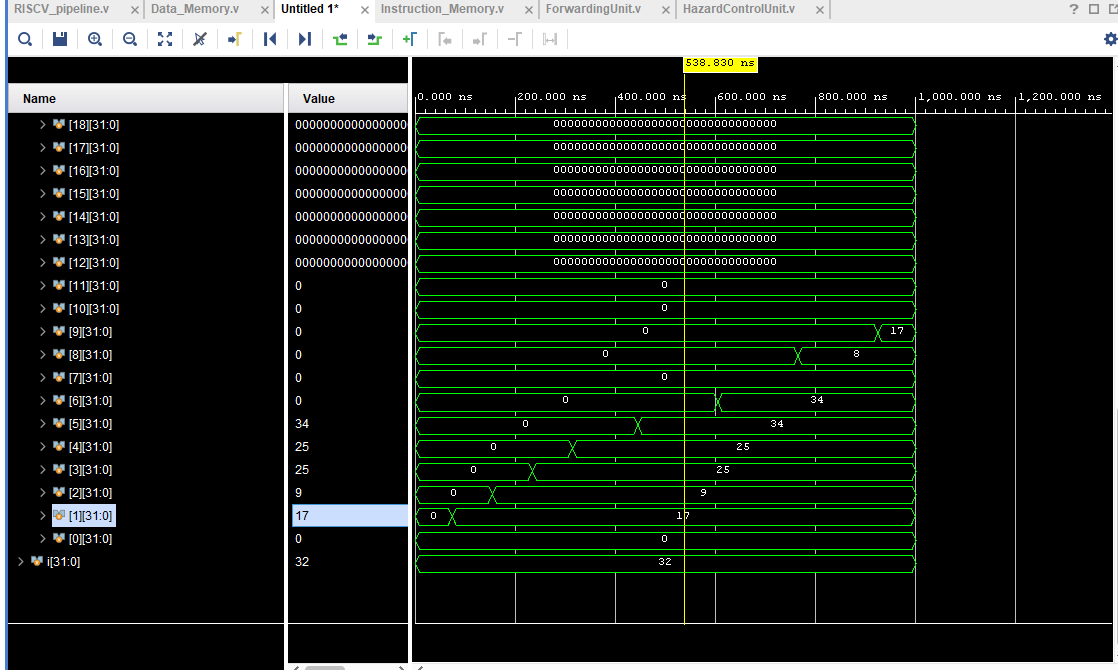
Team:

**Koussay jaballah 900234021**

**Ismaiel Sabet**

3)

Exp1 screenshot:



Exp2 screenshot:

**5)**

The **Execution Stage (EX)** will forward the data before the Memory Stage (MEM) because:

**Forwarding from the Execution Stage (EX):** When an instruction is in the EX stage, the result of its operation can be forwarded directly to the next instruction's EX stage, bypassing the need to wait for the write-back (WB) stage. This is because the execution result is available in the EX stage, and forwarding the data directly to the next instruction in the EX stage allows it to proceed without delay.

**Forwarding from the Memory Stage (MEM):** Forwarding data from the MEM stage is generally less common and less efficient because the memory operation may take longer, and the result is typically not available as soon as the EX stage is done. The hazard from the MEM stage can often be resolved by ensuring that the data is written back to the register file and then accessed by the next instruction.

**6)**

this condition is necessary in RISC-V because:

1. Register x0 is hardwired to zero:

register **x0** is always **zero**. It is a special register that is not writable. Any instruction that attempts to write to **x0** will effectively discard the data because **x0** will always read as **0**.

1. Preventing unnecessary forwarding:

If an instruction tries to forward data to **x0**, it would be pointless because **x0** is always zero, and the forwarded data would be discarded. The forwarding logic checks whether the destination register is **x0**, and if it is, it avoids forwarding data to it.

1. Correct behavior in the pipeline:

The condition **(EX/MEM.RegisterRd ≠ 0)** ensures that the forwarding unit only forwards data to registers that are not **x0** (register 0). This optimizes the pipeline and prevents unnecessary operations, which could potentially waste resources or introduce unnecessary complexity.