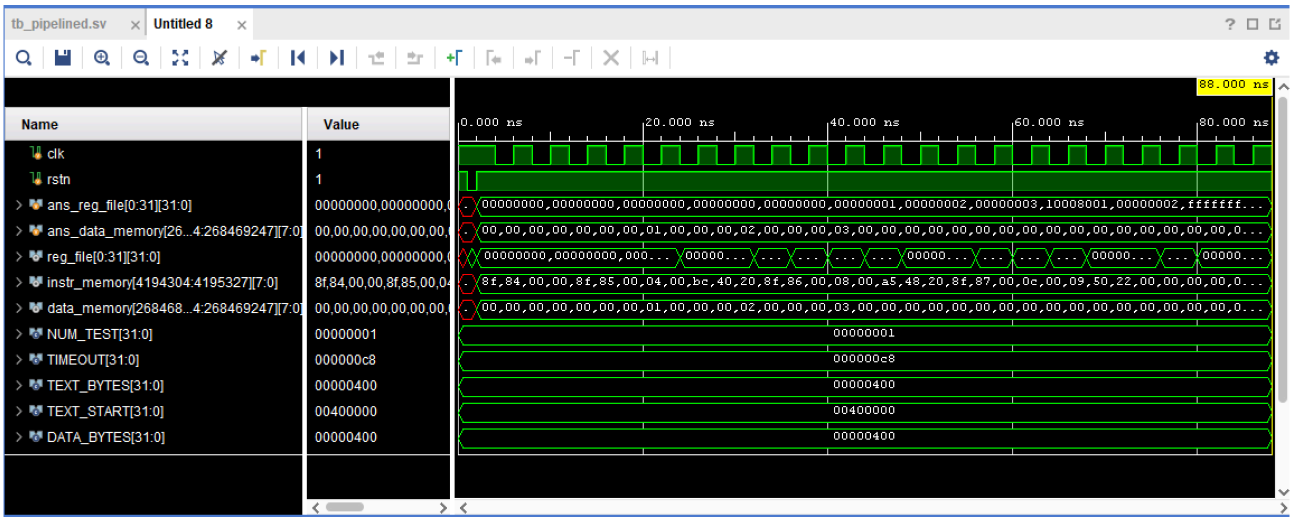
Report

Experimental Result

1. Show the waveform screen shot of the test we provided.



Name	Value
clk	1
rstn	1
ans_reg_file[0:31][31:0]	00000000,00000000,0
ans_data_memory[26...4:268469247][7:0]	00,00,00,00,00,00,00,0
reg_file[0:31][31:0]	00000000,00000000,0
instr_memory[4194304:4195327][7:0]	8f,84,00,00,8f,85,00,04
data_memory[268468...4:268469247][7:0]	00,00,00,00,00,00,00,0
NUM_TEST[31:0]	00000001
TIMEOUT[31:0]	000000c8
TEXT_BYTES[31:0]	00000400
TEXT_START[31:0]	00400000
DATA_BYTES[31:0]	00000400

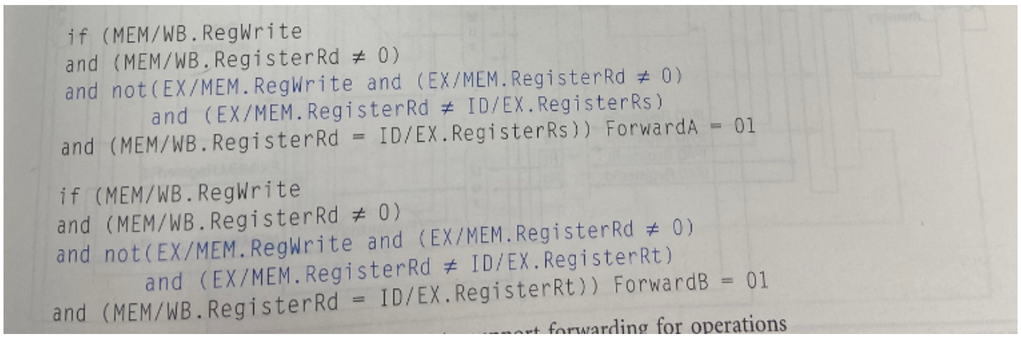
```
88.000 ns
0.000 ns            20.000 ns            40.000 ns            60.000 ns            80.000 ns
00000000,00000000,00000000,00000000,00000000,00000001,00000002,00000003,10008001,00000002,ffffff...
00,00,00,00,00,00,00,01,00,00,00,02,00,00,00,03,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0...
00000000,00000000,000...  00000...  ...  ...  ...  ...  00000...  ...  ...  ...  00000...  ...  00000...
8f,84,00,00,8f,85,00,04,00,bc,40,20,8f,86,00,08,00,a5,48,20,8f,87,00,0c,00,09,50,22,00,00,00,00,0...
00,00,00,00,00,00,00,01,00,00,00,02,00,00,00,03,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0...
00000001
000000c8
00000400
00400000
00000400
```

2. What other cases you've tested? Why you choose them?

I tested addi instructions, ex. addi $t0, $t1, 9 to ensure there is no error with control signals.
I also test various instructions, including lw, sw and R-format instructions, following by beq to test there is no error with my forwarding and hazard detection architecture.

Answer the following Questions

1. List out the equation to detect EX & MEM hazard in forwarding unit. Which part of the equation in textbook p.369 is wrong?



```
if (MEM/WB.RegWrite
and (MEM/WB.RegisterRd ≠ 0)
and not(EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
        and (EX/MEM.RegisterRd ≠ ID/EX.RegisterRs)
and (MEM/WB.RegisterRd = ID/EX.RegisterRs)) ForwardA = 01

if (MEM/WB.RegWrite
and (MEM/WB.RegisterRd ≠ 0)
and not(EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
        and (EX/MEM.RegisterRd ≠ ID/EX.RegisterRt)
and (MEM/WB.RegisterRd = ID/EX.RegisterRt)) ForwardB = 01
```

Ans: The equation to detect EX & MEM hazard in forwarding unit is EX/MEM.RegWrite and (EX/MEM.RegisterRd != 0) and (EX/MEM.RegisterRd == ID/EX.RegisterRs/*Rt in forward B) So the blue part of the text book should be revised as above.

2. In forwarding for beg , is forwarding from MEM/WB to ID needed? Why?

Ans: No, it is not needed. If branch read registers right after an ALU instruction writes it, we only need one stall and forward from EX/MEM to ID. If branch read registers right after a load instruction writes it, we need two stalls. However, in this case, since we write data in first half cycle and read data in second half cycle, the original data would already be written into register before we read it after two stalls. As a result, there is no need to forward from MEM/WB to ID.

3. Briefly explain how you insert 2 stalls when beq reads registers right after lw writes it.

Ans: There are two comparisons needed in my beq hazard detection architecture. First, I compare the stage between ID and IF. If ID stage instruction needs to write data in register and the register to write is same as the IF input register (rs or rt), then stall. Second, I compare the stage between EXE and IF. If EXE stage instruction needs to write data into register, need to read from memory, and the register to write is same as the IF input register (rs or rt), then stall. If one of the above condition is true, then stall. So it would automatically create two stalls when beq reads registers right after lw writes it.

4. sw right after lw is quite common since copy and paste a data from one address to another is used frequently. In textbook, a stall is followed by the lw in this case. Is it possible to remove this stall? How?

Ans: We can add a forward unit in MEM stage. The logic possibly would be: If write_register of previous lw instruction == rt of next sw instruction, then directly use the read_data from data memory of lw to be the write_data of sw.