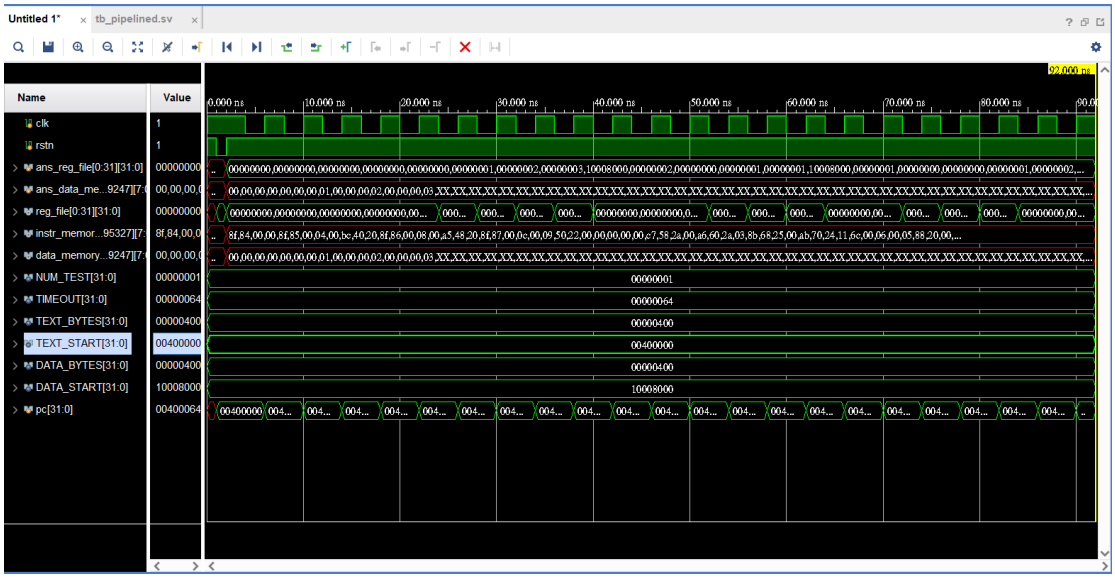
Lab3

111550088 張育維

1. Experimental Result

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



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

I test for the addi $5 24 and sw $8 4($28) because they are not in the testbench, and I want to check whether my code is correct.

2. Answer the following Questions

1. For each code sequence below, state whether it must stall, can avoid stalls using only forwarding, or can execute without stalling or forwarding.

Sequence 1	Sequence 2	Sequence 3

lw $t0,0($t0)
add $t1,$t0,$t0
	add $t1,$t0,$t0
	addi $t2,$t0,#5
	addi $t4,$t1,#5
		addi $t1,$t0,#1
		addi $t2,$t0,#2
		addi $t3,$t0,#2
		addi $t3,$t0,#4
		addi $t5,$t0,#5

Sequence 1: must stall.

Sequence 2: can avoid stalls using only forwarding.

Sequence 3:can execute without stalling or forwarding.

2. Explain the difference between throughput and latency.

Throughput refers to the number of tasks or instructions the CPU can execute within a specific time frame. A higher throughput indicates that the system is more efficient. Latency, on the other hand, refers to the amount of time the CPU takes to complete a task. Lower latency indicates that the system

can respond to requests more quickly.

3. A group of students were debating the efficiency of the five-stage pipeline when one student pointed out that not all instructions are active in every stage of the pipeline. After deciding to ignore the effects of hazards, they made the following five statements. Which ones are correct? Explain why or why not.

1. Allowing jumps, branches, and ALU instructions to take fewer stages than the five required by the load instruction will increase pipeline performance under all circumstances.

It’s not correct. If the instruction needs the result of the previous load instruction, it may still need to wait for it, which can’t increase pipeline performance.

2. Trying to allow some instructions to take fewer cycles does not help, since the throughput is determined by the clock cycle; the number of pipe stages per instruction affects latency, not throughput.

It’s correct. Because throughput depends on the clock cycle. the number of pipe stages doesn’t affect on it. It’s affects on latency.

3. You cannot make ALU instructions take fewer cycles because of the writeback of the result, but branches and jumps can take fewer cycles, so there is some opportunity for improvement.

It’s correct that ALU instructions can't be executed in fewer cycles, but branches and jumps can. This is because they don’t necessarily require waiting for the writeback stage to complete. For example, in branch instructions, if the branch condition can be resolved in the execution stage, the next instruction's execution path can be decided in subsequent stages without waiting for the writeback stage to complete.

4. Instead of trying to make instructions take fewer cycles, we should explore making the pipeline longer, so that instructions take more cycles, but the cycles are shorter. This could improve performance.

It's correct. By lengthening the pipeline and shortening the cycles, we increase concurrency. This allows for a greater number of different instructions to be in various stages simultaneously, leading to improved performance.