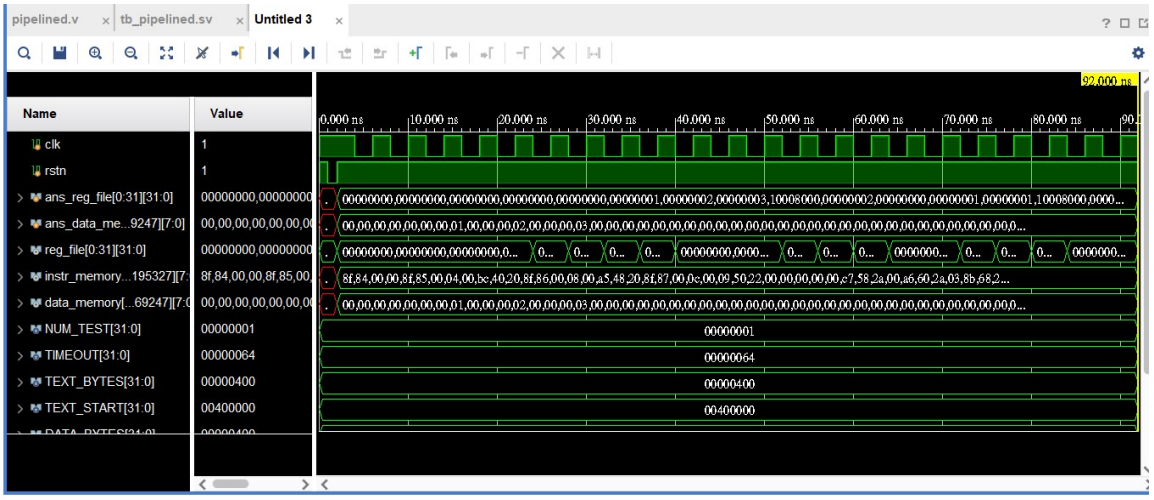
HW3 Report

分類	
Date	@May 12, 2024 10:11 PM
Review	☐

Experimental Result

- Waveform



- other case : addi $t9 $t9 256

Questions

1. Sequence Analysis:
    - Sequence 1:
      This code sequence can be executed without stalling or forwarding.
    - Sequence 2:
      This sequence must stall due to data hazards. The "add $t1, $t0, $t0" instruction needs to wait for the "addi $t2, $t0, #5" instruction to complete before using the updated value of $t0. Similarly, "addi $t4, $t1, #5" needs to wait for "addi $t2, $t0, #5" to complete before using the updated $t1 value.
    - Sequence 3:
      This sequence can avoid stalls using only forwarding. The results can be forwarded

to the next instruction without waiting for the previous instruction to complete.

2. Throughput and Latency:
    - Throughput refers to the number of instructions that can be completed in a given time period, often measured in instructions per cycle (IPC). It represents the overall performance of the pipeline.
    - Latency, on the other hand, is the time it takes for a single instruction to complete execution from start to finish. It measures the delay or time required to execute an individual instruction.
3. True or False:
    - False. Allowing jumps, branches, and ALU instructions to take fewer stages than the five required by the load instruction will not increase pipeline performance under all circumstances. While it may reduce the latency for those specific instructions, it could also introduce additional stalls and data hazards, potentially reducing overall performance.
    - True. The throughput of the pipeline is determined by the clock cycle, as instructions are issued at each clock cycle. The number of pipe stages per instruction affects the latency, not the throughput.
    - True. ALU instructions can’t take fewer cycles due to the writeback. However, branches and jumps can potentially take fewer cycles, since they do not require the writeback stage.
    - False. Higher branch misprediction penalty may comes with longer pipeline.However, by dividing the work of each instruction into more stages, the cycle time can be reduced, potentially increasing the clock frequency and overall throughput, even though each instruction takes more cycles.