

1) Rewrite the following delayed branch MIPS snippet to maximize performance, assuming it has forwarding.

```
Loop:    addi $v0, $v0, 1
         addi $t1, $a0, 4
         lw  $t0, 0($t1)
         add $a0, $t0, $a1
         addi $a0, $a0, 4
         bne $t0, $0, Loop
         nop
         jr  $ra
```

2) Now, assume for the delayed branch code from above exercise that our hardware can execute Static Dual Issue for any two instructions at once. Using reordering (with nops for padding), but no loop unrolling, schedule the instructions to make the loop take as few clock cycles as possible.

3) How many PCs would we need to implement hardware of Static Dual issue or any changes required in IF?

4) Would any changes be required in ID to support execution of two instructions at once?

5) Would any changes be required in EX to support execution of two instructions at once?

6) Would any changes be required in MEM to support execution of two instructions at once?

7) Would any changes be required in WB to support execution of two instructions at once?

8) Are there any new hazards in terms of type or their counts in this new processor? Please enumerate them and also how to resolve them (ie stalling or better options)