Computer Organization: Quiz 4 **Student ID:** **;Name:**

(TRUE or FALSE) (105 pts)

Note: the following problems are designed for realizing the R-type, lw/sw, and beq/j instructions on a MIPS CPU.

1. Single cycle pipeline diagram is good for showing the state of a pipeline in a given cycle. (o)
2. Multiple cycle pipeline diagram is good to analyze if forwarding scheme can help to resolve the data dependence between two instructions. (o)
3. ID/EX.RegisterRs implies the register number of register Rs of the instruction that is going to enter the ID stage. (x)
4. One forwarding conditions to detect the EX hazard is: (EX/MEM.RegWrite and (EX/MEM.RegisterRd $\neq$ 0) and (EX/MEM.RegisterRd = ID/EX.RegisterRs)) (o)
5. The double data hazard problem describes the situation that the target register of an instruction is the two input registers of its subsequent instruction. (x)
6. To stall the pipeline, we need to set as zero all the control values in the ID/EX register. That is all what we need to do for stalling the pipeline. (x)
7. If we want to reduce the penalty of branch instruction, we need to add a target address adder and a register comparator in the ID stage (o)
8. For a dynamic prediction scheme using one-bit register, the run of a loop always predict incorrectly two times. (x)
9. Exceptions are the unexpected events happening inside the CPU while interrupts are from an external IO controller. (o)
10. The behavior of exception is similar to that of incorrect prediction of branch, so they use much of the same hardware. (o)
11. Very long instruction word is a kind of scheme with static multiple issues. (o)
12. As we unroll the code sequence in a loop, the high parallelism among in the unrolled code sequence appears without any data dependency. (x)
13. For the superscalar-based CPU, the instructions are issued in the order, are executed out of order, and are committed in the order. (o)
14. Although all data dependences can be identified, not all data dependences can be eliminated. (x)
15. Modern compilers can precisely predict the outcome of a branch instruction. (x)