CS 286-001 Computer Architecture & Organization
Fall 2018
Quiz #6 on October 1, 2018 (SOLUTIONS)

Your Last Three Digits: ________________
(please do NOT write all of your student ID or your name)

Grade: ______

(1) Assume that all the inputs for each instruction must be available by the beginning of the ID phase and the output from each instruction becomes available at the end of the WB phase, find which of the following four datapath architectures can data hazards for each of RAR, RAW, WAR, and WAW?

<table>
<thead>
<tr>
<th>2nd instruction</th>
<th>READ</th>
<th>1st instruction</th>
<th>WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RAR</td>
<td>RAW</td>
</tr>
<tr>
<td></td>
<td>READ</td>
<td>None (Ω)</td>
<td>Pipeline</td>
</tr>
<tr>
<td>WRITE</td>
<td>WAR</td>
<td>None (Ω)</td>
<td>Super-Scalar</td>
</tr>
<tr>
<td>WRITE</td>
<td>WAW</td>
<td></td>
<td>Super-Pipeline</td>
</tr>
</tbody>
</table>

(2) The following is a sketch of subroutine calls using jal and jr instructions. What is the problem in the program?

```
main:
    li $s0, 10
LOOP:
    jal xyz
    li $s0, 200
    add $s0, $s0, 5
    jne $s0, $zero, LOOP
    slt $s0, $3, $zero
    jr $r
xyz:
    li $s0, 10
    sub $s0, $s0, 1
    jne $s0, $zero, LOOP
    jr $31
```

The problem is that “$s0” register, which is sued as the loop counter in the loop, “LOOP” in module main is re-initialized in module xyz, causing an infinite loop for the loop in main.
(3) Describe what “jr $ra” instruction performs.

“jr $ra” instruction lets the processor jump to the memory address of the instruction, which is stored in “ra” register.

(4) What are the two tasks “jal” instruction performs?

1. Load the address of the next instruction to “$ra” register
2. Jump (set PC register) to the first instruction in the called subroutine

(5) What is the “terminating (exit) condition” that should be used in recursive structure? Why do we need one in most of the recursive structures we implement?

The terminating (exit) condition is a condition that is used (implemented as a conditional branch instruction) to prevent a processor from going deeper in a recursive structure.