5) Suppose you had one opcode left and you wanted to add one more instruction to the MIPS ISA.

a) [3 points] Which of the following 3 instructions could be encoded using one of the current MIPS instruction formats? (`address` and `immed` are 16-bit integers). Justify your choice by showing how the instruction is encoded. For the two you eliminate, say why they could not use any of the current MIPS formats.

```
addw    $rt, address($rs)          # rt = rt + Mem[rs+address]
add2w   $rd, $rt, address($rs)     # rd = rt + Mem[rs+address]
addiw   $rd, immed, address($rs) # rd =immed+Mem[rs+address]
```

b) [2 points] Assuming that you now include the instruction that can be encoded in one of the MIPS formats, can you still say that the MIPS ISA is a RISC machine? Justify your answer.

c) [8 points] Implement the new instruction using a multiple-cycle implementation, pictured below. Include all new sequential and combinational logic that you need, as well as any new control signals. How many cycles would it take to execute the new instruction? What happens in each cycle?