**NOTE: It is my policy to give a failing grade in the course to any student who either gives or receives aid on any exam or quiz.**

**INSTRUCTIONS: For short answer questions, circle the letter of the best choice. There is no penalty for guessing. Other questions tell where to put your answers. Unless otherwise indicated, all questions count equally.**

1. What is the *Zero* output of the ALU used for?
   A. To decide whether the *beq* instruction branches or not.
   B. To zero out the ALU result.
   C. To initialize register 0.
   D. To get the value of register 0.
   E. To increment the PC by zero.

2. What is the address of the next instruction that is executed after a *j* instruction?
   A. The address of the *j* instruction plus 4.
   B. The rightmost 16 bits of the *j* instruction, sign extended, and added to the address of the *j* instruction.
   C. The leftmost 4 bits of the PC contatenated with the rightmost 26 bits of the *j* instruction, concatenated with two zeros.
   D. The rightmost 16 bits of the PC concatenated with the leftmost 16 bits of the *j* instruction.
   E. The leftmost 16 bits of the *j* instruction, sign extended.

3. How many gates are needed to sign-extend the immediate field of an I-format instruction?
   A. 16
   B. 32
   C. It depends on whether the value is positive or negative.
   D. 1
   E. 0

4. What is the sign-extended representation of 0x8001?
   A. 0x00008001
   B. 0x80010000
   C. 0x00008001
   D. 0xFFFF8001
   E. 0x8001FFFF

5. What is a word address?
   A. A register that contains a word of data.
   B. Any 32-bit number.
   C. Any 26-bit number.
   D. A 32-bit number that ends with two binary zeros.
   E. A 26-bit number than ends with three binary zeros.

6. How is the *effective address* of a lw instruction computed?
   A. Sign extend the ALU control and add 4.
   B. Sign extend bits 0-15 of the instruction, and add the result to the contents of the register specified by the *rs* field of the instruction.
   C. Sign extend bits 0-15 of the instruction, and add the result to the contents of the register specified by the *rt* field of the instruction.
   D. Sign extend bits 0-15 of the instruction, and add the result to the contents of the register specified by the *rd* field of the instruction.
   E. Compare register 0 to register 31.

7. What does the instruction *jr $ra* do?
   A. It joins registers *r* and *a*, and puts the result in register *$*.
   B. It right-justifies the remainder of the line in register *ra*.
   C. It returns from a subroutine by jumping to the address in register *ra*.
   D. It calls a subroutine.
   E. It prevents a subroutine from being called.

8. If register *$a0* contains the address of an array of words in memory, what instruction could be used to get the contents of the first word of the array into register *$t0*?
   A. *lw $t0, $a0(0)*
   B. *lw $a0, 0($t0)*
   C. *sw $a0, $t0*
   D. *sw $t0, $a0*
   E. *lw $t0, 0($a0)*

9. Following Question 8, what instruction would be used to get to the second element in the array?
   A. *add $a0, $a0, $r4*
   B. *add $a0, $a0, $r1*
   C. *addi $r0,4($a0)*
   D. *addi $a0, $a0, 4*
   E. *sub $a0, $a0($a0)*

10). Translate *add $r20, $r20, $r16* to hexadecimal:
   A. 0xADD202016
   B. 0xADD141410
   C. 0x20141410
   D. 0x0290A020
   E. 0xBADF00D

11. What is wrong with this instruction: *sw $r0, 0xABCDE($r5).* (Read Question 12 before answering this one.)
   A. 0xABCDE is not a word address.
   B. $r5 is not a word address.
   C. $r5 plus 0xABCDE is an odd value.
   D. 0xABCDE will not fit into bits 0-15 of a *lw* instruction.
   E. 0xABCDE will not fit into memory.

12. What will the assembler do with the instruction in Question 11?
   A. It will turn it into a sequence of instructions, including a *lui* instruction, to calculate the effective address.
   B. It will print an error message.
   C. Both A and B.
   D. Neither A nor B.
   E. All of the above.

13. ALUOp = 00 forces the ALU to add. When is this value used?
   A. When executing an *add* or *addi* instruction.
   B. When executing a *sw* or *lw* instruction.
   C. When executing a *beq* or *bne* instruction.
   D. When executing a *j* instruction.
   E. When executing a *slt* or *slti* instruction.

14. ALUOp = 01 forces the ALU to subtract. When is this value used?
   A. When executing an *add* or *addi* instruction.
   B. When executing a *sw* or *lw* instruction.
   C. When executing a *beq* or *bne* instruction.
   D. When executing a *j* instruction.
   E. When executing a *slt* or *slti* instruction.

15. Why is the Mux controlled by *RegDst* necessary?
    A. So bits 20-16 of the instruction can be written into memory if necessary.
    B. In order to add either 4 or 8 to the PC, as needed.
    C. So *lw* instructions can use the *rt* field to specify the register being loaded into.
    D. So *add* instructions will not overflow.
    E. To compute the branch target address.

16. When would *RegWrite* be false?
    A. For *lw* and *sw* instructions.
    B. For *sw* and *beq* instructions.
    C. For *add* and *slt* instructions.
    D. For *addi* and *slti* instructions.
    E. For all instructions except *andi*.

17. Why are there two adders and an ALU in the diagram instead of just the ALU?
    A. So that branch, jump, and arithmetic instructions can be done at the same time.
    B. In case one breaks, the other two can take over its work and the computer doesn't have to be fixed.
    C. So their answers can be compared to make sure they all get the same result.
    D. Because this is a single-cycle design, so all three values have to be computed at the same time.
    E. Because this is a multiple-cycle design, and the outputs of one are the inputs to the next one.

18. Why is there no *InstrRead* (instruction read) control signal in the diagram?
    A. Because Instruction Memory is read-only memory, which means that a new instruction is automatically read out whenever a new address is loaded into the PC.
    B. Because the *MemRead* signal goes to both the Instruction Memory and the Data Memory.
    C. Because the instructions are fetched from Data Memory.
    D. Because the CPU does not read instructions, it writes them.
    E. It is a typographical error; there should be a *InstrRead* wire going from the *Sign extend* oval to the *ALU control* oval.

19. Which combination of control signals is true only for *lw* instructions?
    A. *Jump*, *Branch*, and *MemRead*.
    B. *Branch*, *MemRead*, and *MemtoReg*
    C. *MemRead* and *MemtoReg*
    D. *MemRead* and *MemWrite*
    E. *ALUSrc* and *Jump*

20. Assume the design in the figure is implemented using a 5 GHz clock. How long will it take to execute *lw* and *and* instructions?
    A. It will take *lw* five nsec, but *and* will take only one nsec.
    B. They will both take five nsec.
    C. They will both take one nsec.
    D. It will take *lw*200 psec, but it will take *and* only 40 psec.
    E. All instructions, include *lw* and *and*, will take 200psec.

21. Which elements in the figure contain sequential logic elements?
    A. The ALU and the two adders.
    B. The two shift and the sign extend logic sections.
    C. The Control and ALU.
    D. The Registers and the PC.
    E. The Registers, the PC, and the multiplexers.

22. What determines the maximum clock rate that can be used when implementing the single-cycle CPU design?
    A. The number of registers.
    B. The sizes of the instruction and data memories.
    C. The number of multiplexers.
    D. The number of adders and ALUs.
    E. The propagation delays needed to complete a *lw* instruction.