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: Circle the letter of the *best* choice for each question.

1. The MIPS *register file* has three *ports.* Identify them.
   A. Upper, lower, bottom, and top.
   B. Left, Right, and middle.
   C. Two read ports and one write port.
   D. Two write ports and one read port.
   E. Text, binary, and undefined.

2. How many bits are associated with each port of the MIPS register file?
   A. 4
   B. 16
   C. 24
   D. 37
   E. 111

3. Why is the MIPS register file called a *file*?
   A. Because that's just conventional hardware terminology.
   B. Because the MIPS processor has no facility for keeping files on disk.
   C. Because the MIPS processor keeps the registers in a disk file.
   D. Because MIPS registers can be used to create iron shavings from pieces of metal.
   E. All of the above.

4. How many flip-flops are there in the MIPS register file?
   A. $2^5 \times 2^5$
   B. $2^5 \times 31$
   C. 31×31
   D. 16
   E. 32

5. What does this instruction do: *add $0, $1, $2*
   A. It replaces the contents of register zero with the sum of the contents of registers one and two.
   B. It adds the contents of register one to the contents of register two and discards the answer.
   C. It prints an error message because you can't compute the sum of registers one and two.
   D. It prints an error message because you can't store anything in register zero.
   E. It sets register zero to either 0x000000001or 0x00000000 depending on whether the value in register one is less than the value in register two or not.

6. What does a *lui* instruction do?
   A. It loads a word of memory into one of registers 16 through 31.
   B. It loads a word of memory into one of registers 0 through 15.
   C. It stores a word of memory into register zero.
   D. It loads the immediate field of an instruction into the left half of a register and zeros out the right half.
   E. It loads the immediate field of an instruction into the right half of a register and zeros out the left half.

7. What does a *beq* instruction do?
   A. It branches to a subroutine.
   B. It returns from a subroutine.
   C. It calls a subroutine recursively.
   D. It changes the PC if two registers contain the same value.
   E. It computes the sum of two registers.

8. MIPS uses *relative branching.* What does this mean?
   A. Branches are made relative to the result of the previous calculation.
   B. Branches are made relative to the previous branch instruction.
   C. The branch target address is calculated by adding a signed number to the PC.
   D. Child branches depend on parent branches.
   E. The root node is used for all branch nodes.

9. How is an *effective address* computed?
   A. R[rd] = R[rs] + R[rt]
   B. R[rd] = R[rs] – R[rt]
   C. It depends on the *funct* field.
   D. It depends on the *shamt* field.
   E. R[rs] + {16{immediate[15]},immediate}

10. What determines the maximum clock speed of the single-cycle processor?
   A. The propagation delays needed to fetch and execute an *lw* instruction.
   B. The memory access time.
   C. The size of the data memory.
   D. The time it takes to compute the branch target address.
   E. The number of states in the control unit.

11. In the single-cycle processor, when is an instruction fetched?
   A. When *MemRead* is true.
   B. When *MemWrite* is true.
   C. When *ALUop* is true.
   D. When a new value is loaded into the PC.
   E. When *IRWrite* is false.

12. Both processors have a logic block named *Sign extend*. What circuitry is inside this block?
   A. 16 flip-flops and 32 AND gates
   B. 16 AND gates and 32 flip-flops
   C. 32 flip-flops and no AND gates
   D. 32 AND gates and no flip-flops
   E. No flip-flops and no AND gates

13. How many input wires and output wires does the *ALU control* block have?
   A. 2 input wires and 1 output wire
   B. 3 input wires and 6 output wires
   C. 8 input wires and 4 output wires
   D. 32 input wires and 32 output wires
   E. No input wires and no output wires.

14. What is "random" about a *random access memory*?
   A. If you give it a random address, it always returns the same data.
   B. If you give it any address, it returns random data.
   C. If you give it random data, it randomly decides whether to read it or to write it.
   D. The access time is constant even if the sequence of addresses is random.
   E. The access time is a random function of the data being written.

15. Why is the *RegDst* control signal needed?
   A. To decide which instruction goes to the right register.
   B. Because the immediate field uses bits 15-11, so the *rt* field has to be used for the destination register number for I-format instructions.
   C. To avoid buffer overflows.
   D. To select what data gets written to the register file.
   E. So the multiplexer can tell the control unit whether the Zero bit is true or not.

16. Why are there two adders and an ALU in one datapath, but only an ALU and no adders in the other one?
    A. Because the ALU can do only one operation per clock cycle, so separate adders are needed for computing the branch target address and PC+4 in the single-cycle design.
    B. Because the ALU can be used to calculate the branch target address, PC+4, the effective address, and the result of arithmetic or logic operations during different clock cycles in the multi-cycle design.
    C. Both A and B
    D. Neither A nor B
    E. All of the above.

17. What do the lines connecting the circles in the diagram represent?
    A. The D inputs to the ALU.
    B. The Q inputs to the ALU.
    C. The Q outputs from the ALU.
    D. Transitions caused by clock pulses.
    E. The effective address.

18. How many clock cycles does it take to fetch and execute an *add* instruction?
    A. 1 in the single-cycle design; 3 in the multi-cycle design.
    B. 1 in the single-cycle design; 4 in the multi-cycle design.
    C. 1 in the single-cycle design; 5 in the multi-cycle design.
    D. 1 in both designs.
    E. 5 in both designs.

19. What is the purpose of the *IorD* control signal?
    A. It tells the control unit whether to fetch an instruction or data.
    B. It selects whether the PC or the ALU supplies the address for a memory operation.
    C. It determines whether the Instruction Register or the Memory Data Register receives the result of a memory write operation.
    D. It helps differentiate between the instruction and data memories in the single-cycle design.
    E. It selects whether the multiplexer will output 16 or 32 bits.

20. Why is ALUOp equal to 01 in state 8?
    A. To force the ALU to subtract.
    B. To make sure the Zero bit really is zero.
    C. To make sure the Zero bit is actually one.
    D. To make the value of the Zero bit toggle on the next clock pulse.
    E. To force a memory read operation.