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: Answer all questions in your examination booklet, *in sequence, please.* Unless otherwise indicated, all questions count equally.

1. (4 Points) Which statement describes the behavior of *beq* best?
   - A. Change the contents of the instruction register depending on the value in register 0.
   - B. Change the contents of register 0 depending on the value in the instruction register.
   - C. Compare the contents of two registers, and if they are different, change the contents of the PC.
   - D. Compare the contents of two registers, and if they are the same, change the contents of the PC.
   - E. Compute the branch target address based on the values of the carry, overflow, zero, and negative condition code bits.
2. (4 Points) Which instruction format does *beq* use?
   - A. B-Format
   - B. R-Format
   - C. I-Format
   - D. J-Format
   - E. M-Format
3. (4 Points) How is the address of the next instruction after a *j* (jump) instruction computed?
   - A. Add 4 to the PC
   - B. Sign extend bits 0:15 of the instruction, sign extend, and add to PC+4.
   - C. Concatenate bits 28:31 of PC+4 with bits 0:15 of the instruction, sign extended.
   - D. Concatenate bits 28:31 of PC+4 with bits 0:25 of the instruction shifted left by 2.
   - E. Multiply the PC by 4 and add to bits 0:25 of the instruction.
4. (4 Points) Which statement best describes the purpose of the *lui* instruction?
   - A. It is used for comparing two registers and changing the value in the PC based on the result of the comparison.
   - B. It is used as an intermediate step during multiplication.
   - C. It is used for converting R-Format instructions to J-Format.
   - D. It is used for converting J-Format instructions to R-Format.
   - E. It is used when a value more than 16 bits wide needs to be put into a register.
5. (4 Points) Why does *lw* use *rt* as the destination register instead of *rd*?
   - A. This question is bogus: *lw* does use *rd* as the destination register.
   - B. This question is bogus: *lw* doesn't have a destination register.
   - C. This question is bogus: there is no *lw* instruction.
   - D. Because *rd* is used for shifting in *lw* instructions.
   - E. Because *lw* is an I-Format instruction and there is no *rd* field in I-Format instructions.
6. (4 Points) What is the operation code for *sw* instructions?
   - A. 43
   - B. 6'b101011
   - C. 0x1B
   - D. All of the above.
   - E. None of the above.
7. (4 Points) What does sign extension do?
   - A. It changes the number of bits used to represent a two's complement number without changing the value.
   - B. It converts a two's complement number to a ones complement number.
   - C. It negates a two's complement number.
   - D. It removes the sign bit from any number.
   - E. It removes the sign bit from negative numbers.

8. (4 Points) How is the effective address of *lw* instructions determined?
   - A. By examining the *rt* and *rd* fields of the instruction to see if they are equal.
   - B. By adding the *rs* field to bits 0:15 of the instruction.
   - C. By adding the *rs* field to the sign extended value of bits 0:15 of the instruction.
   - D. By adding the contents of the register selected by the *rs* field of the instruction to the sign-extended value of bits 0:15 of the instruction.
   - E. By adding the contents of the register selected by the *rs* field of the instruction to the sign-extended value of bits 0:15 of the instruction shifted left two positions.

9. (4 Points) How many gates are required to sign-extend the immediate field of an instruction?
   - A. 0
   - B. 1
   - C. 16
   - D. 32
   - E. 2^{32}

10. (4 Points) What does the *slt* instruction do?
    - A. It loads the number 1 or 0 into register *rd* depending on whether the value in register *rs* is numerically less than the value in register *rt.*
    - B. It loads the difference between the *rs* and *rt* fields of the instruction into the *rd* field of the instruction.
    - C. It loads the difference between the *rs* and *rt* fields of the instruction into the *rd* register of the instruction.
    - D. It subtracts Register[*rt*] from Register[*rs*] and puts the difference into Register[*rd*].
    - E. It puts *true* or *false* into the PC depending on whether Register[0] says to branch or not.

11. (4 Points) What are the inputs to the *ALU Control* logic?
    - A. The ALU function code.
    - B. The *opcode* and the *funct* fields of the instruction.
    - C. The *opcode* and the *ALUOp* bits from the Controller.
    - D. The *ALUOp* bits and the *funct* field.
    - E. The *funct* field and the *ALU result*.

12. (4 Points) Outside of the Register File, how many registers are there in the datapath?
    - A. One in the single-cycle design, and six in the multi-cycle design.
    - B. Six in the single-cycle design, and one in the multi-cycle design.
    - C. None in either design.
    - D. Two in the single-cycle design, and four in the multi-cycle design.
    - E. Four in the multi-cycle design, and two in the single cycle design.

13. (4 Points) What comes out of *Read data 1*?
    - A. The *rs* field of the instruction.
    - B. The *rt* field of the instruction.
    - C. The *rd* field of the instruction.
    - D. Register[*rs*]
    - E. Register[*rt*]

14. (4 Points) What is the purpose of *RegDst*?
    - A. It destroys the register.
    - B. It computes the distance between two registers.
    - C. It regulates the distance.
    - D. It selects whether to use *rt* or *rd.*
    - E. It selects whether to use *rs* or *rt.*

15. (4 Points) What is the purpose of *IorD*?
    - A. It tells the controller whether the Mux contains 0 or 1.
    - B. It tells the controller what address is in the PC.
    - C. It tells the PC what address to put in the Mux.
    - D. It tells the Mux what address to put in the PC.
    - E. It tells the Mux what address to send to the Memory.

16. (4 Points) What is *14{immediate[15]},immediate,2'b0*?
    - A. The concatenation of 14 copies of the sign bit, the immediate field, and two bits of zero.
    - B. SE((SL2(*immediate*))
    - C. The value that gets added to PC+4 when computing the Branch Target Address.
    - D. A 32-bit two's complement number giving the number of bytes between the address of the next instruction and the address of another instruction.
    - E. All of the above.
17. (4 Points) Why are there two adders in one datapath and not in the other one?
    - A. The single-cycle datapath doesn't do the calculations performed by those adders.
    - B. The multi-cycle datapath doesn't to the calcuclations performed by those adders.
    - C. The multi-cycle datapath uses registers to do those additions.
    - D. The multi-cycle datapath uses the ALU to do those additions.
    - E. They make the single-cycle datapath go faster.
18. (4 Points) What are the CPIs for the two datapaths?
    - A. One for the single-cycle design, and three for the multi-cycle design.
    - B. One for the single-cycle design and between three and five for the multi-cycle design.
    - C. Three for the single-cycle design and one for the multi-cycle design.
    - D. One for both designs.
    - E. It depends on which instruction is being executed.
19. (4 Points) Why might the multi-cycle datapath be faster than the single-cycle datapath?
    - A. It's faster because the register file is bigger.
    - B. It's faster because the ALU has fewer propagation delays.
    - C. It's faster because there are fewer clock cycles required.
    - D. It's faster because there are fewer instructions per program.
    - E. It's faster because there are fewer propagation delays within a cycle.
20. (4 Points) What is the purpose of *MemtoReg*?
    - A. It tells the controller whether the instruction is *lw* or *sw*.
    - B. The controller uses it to control whether the ALU result or data from memory is to be written to the register file.
    - C. The register file uses it to read from the two registers.
    - D. It determines whether Register 0 can be changed or not.
    - E. The instruction uses it to tell the ALU where to put the result.
21. (4 Points) In the single-cycle design, the Controller consists only of combinational logic.
    - A. True
    - B. False
22. (4 Points) In the multi-cycle design using a FSM to implement the Controller, it consists of a mix of combinational and sequential logic.
    - A. True
    - B. False
23. (4 Points) What happens if *RegWrite* gets stuck at 0?
    - A. Of the instructions implemented in Chapter 5, only *sw, beq,* and *j* will continue to work correctly.
    - B. The only instruction that won't work is *lui*.
    - C. Branch and jump instructions will stop working correctly.
    - D. Branch instructions will continue to work, but store instructions won't.
    - E. Store instructions will continue to work, but branch instructions won't.
24. (4 Points) What are the values that might be loaded into the PC?
    - A. The data words loaded by *lw* instructions.
    - B. The instruction words fetched by the controller.
    - C. The jump, branch target, or next instruction addresses.
    - D. The leftmost four bits of the PC concatenated with the rightmost 16 bits of the instruction.
    - E. The left half of the branch target address OR'd with the rightmost 16 bits of the instruction.
25. (4 Points) What happens during State 7 of the multi-cycle design?
    - A. Register[*rs*] gets combined with Register[*rt*] according to the *funct* field of the instruction.
    - B. The result of choice A gets written to Register[*rd*].
    - C. The result of choice A gets written to register ALUout.
    - D. The result of choice A gets written to memory.
    - E. The result of choice A gets written to Register[0].