Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Computer Organization (CS 345)**

**Final Exam**

Part 1: Short Answer (30 points)

a) Bits and bytes (2 points each).

1. How many bits are in a byte?

8

1. How many bytes are used in a 64-bit floating point register?

8

1. How many bytes are in an ASCII character?

1

b) Boolean algebra (2 points each): reduce each of the following Boolean expressions to a value or a sum of products equation. Assume A, B, C, and D are Boolean variables. ~ = NOT, \* = AND, and + = OR. 1 = true and 0 = false.

1. (A+1) + B\*1

1

1. B\*1 \* (B + 0)

B

1. (A + B \* C) \* D

A\*D + B\*C\*D

1. D + (A \* B \* C \* 1)

D + A\*B\*C

c) Assembly programming (4 points each)

1. Write assembly code to store 23, 14, and -3 in temporary integer registers $t0, $t2, and $t4.  
   li $t0, 23

li $t2, 14

li $t4, -3

1. Write an assembly instruction to multiply the value stored in $t2 by 4. Additionally, use the shift left logical (sll) instruction to multiply the value stored in $t2 by 4. Recall that shifting to the left by one bit multiplies by 2.

mul $t2, $t2, 4

sll $t2, $t2, 2

1. Write a for-loop that decrements $t4 by one, five times.

li $t1, 0

for:

subi $t4, $t4, 1

addi $t1, $t1, 1

ble $t1, 5, for

1. Write a function that takes two parameters and returns the product of the two numbers.

mul2:  
 mul $v0, $a0, $a1

jr $ra

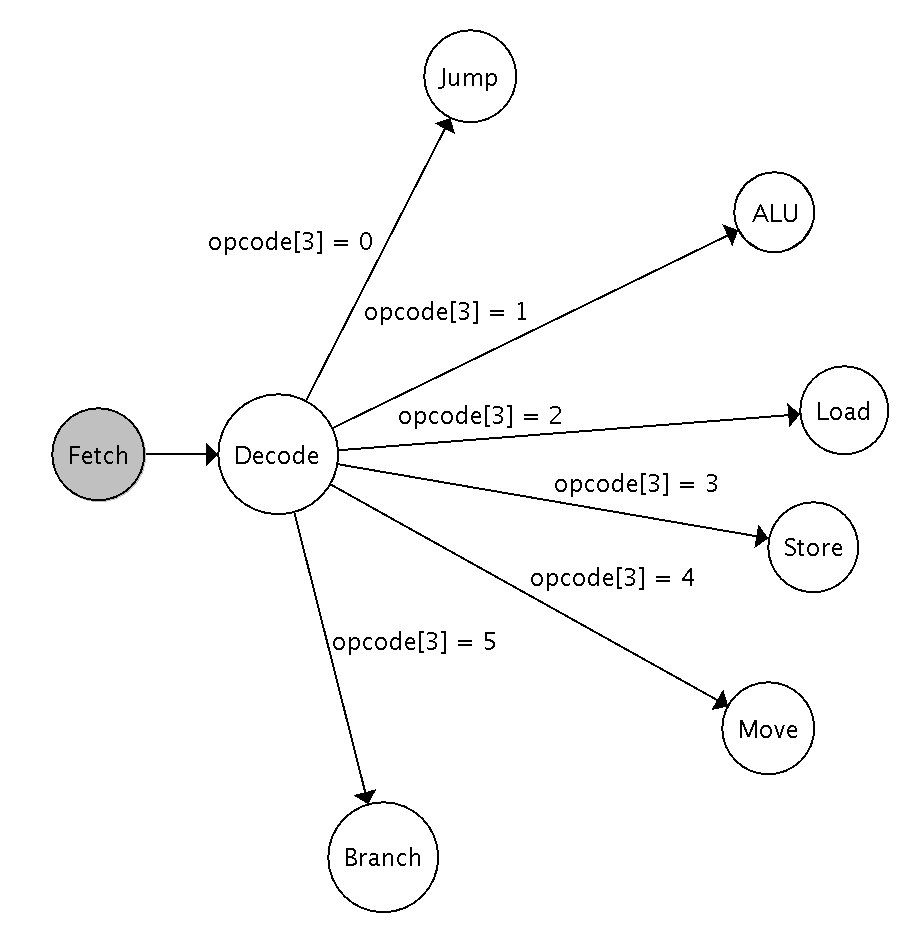
Part 2: Long Answer (70 points, 10 points each)

1. List the basic component that is common to all flip-flops, the storage capacity of this component, and the CPU memory component for which flip-flops are often used. Then, explain the difference between edge-triggered and level-triggered flip-flops and why both are useful.

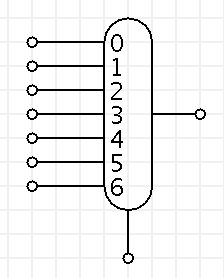
The RS-Latch is common to all flip-flops and its storage capacity is one bit. The CPU component for which flip-flops are often used is the register, which is used to store data temporarily.

Edge-triggered flip-flops are those whose value may change only with the rising or falling edge of the enable or clock signal. Level-triggered flip-flops may have their value changed at any time during which the clock signal is on or off. Both are useful as they may be used to store data. More specifically, edge-triggered flip-flops are useful because they allow for the exact time at which a write may occur to be controlled. If such control is not necessary, a level-triggered flip-flop may be used.

1. Draw a state machine that transitions between the three basic CPU states. The final state actually consists of a group of states – one for each of six possible opcodes – ALU, load, store, move, branch, and jump.



1. Draw a multiplexor that takes seven data inputs. How many bits do you need to use for the selector? Why? What is provided as output from the multiplexor? Why?



3 bits are needed for the selector because 2^2 < 7 < 2^3. The output is the selected, enumerated input. For example, if 000 were provided as input to the selector, the bits from input 0 would be provided as output. If 101 were provided as input to the selector, the bits from input 5 would be provided as output.

1. Given the following instruction, provide the 32-bit binary representation of the instruction (this is one 32-bit binary number), and provide the hexadecimal representation of the instruction. The opcode for sub is 0 hex, the function code for sub is 22 hex, $t1 is represented as 9d, $t2 is represented as 10d, and $t3 is represented as 11d. Hint: the attached sheet shows how many bits are in each portion of the instruction.  
     
   sub $t1, $t2, $t3

Opcode rs rt rd shamt funct

000000 01010 01011 01001 00000 100010

0000 0001 0100 1011 0100 1000 0010 0010

0x014b4822

1. Create Boolean sum of products equations for the following truth table. You should have two equations in your solution, one for each output.  
     
   A B C Output1 Output2

--------------------------

0 0 0 0 0  
0 0 1 0 1

0 1 0 1 0

0 1 1 0 1

1 0 0 1 0

1 0 1 0 0

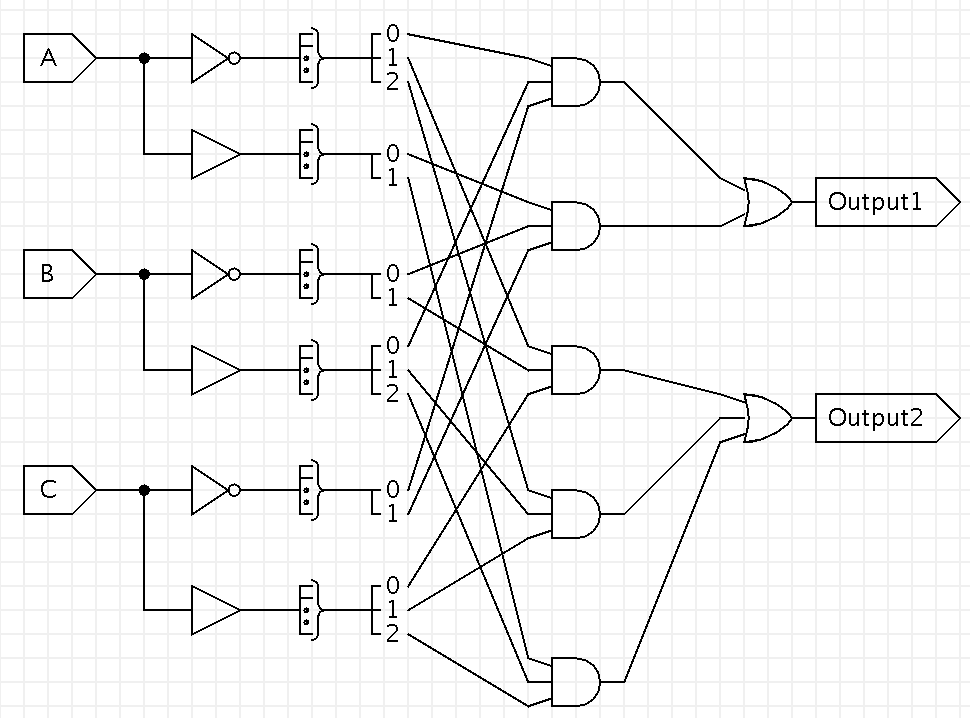
1 1 0 0 0

1 1 1 0 1

Output1 = ~AB~C + A~B~C

Output2 = ~A~BC + ~ABC + ABC

1. Draw complete logic gates for the equations from the previous problem. How long does it take to get a correct output? Assume propagation delays are 2ns for a NOT gate, 5ns for an OR gate, and 10 ns for an AND gate.



17ns to get a correct output.

1. Draw a component-based model for a 5-bit adder that adds two 5-bit values stored in registers and stores the result in a register. Be sure to indicate all possible inputs and outputs for each component.

