CS120 Fall 2016 Final Name:
Section A (noon) or B (9:40am). Circle one!

1) 4 points. Convert 8-bit (2-digit) 2's complement hex values:
47 -23

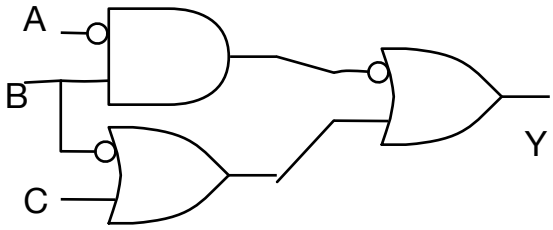
2) 4 points. Convert 8-bit 2's complement hex to decimal:
0x2A 0xE7

3) 4 points. You have 23 bits for a 2's complement number. Using your estimating abilities, what is roughly the largest positive number you can represent? I'm looking for an estimate, not the exact value….

4) 4 points. Sketch a 3-input NAND gate at the transistor level.

5) 4 points. We linked a number of full adder circuits to make a multi-bit adder in lab. A full adder has inputs, A, B, Cin, and outputs Cout and S. Fill out a truth table for this circuit.

6) 4 points. Create a truth table for the following circuit.



7) 4 points. For a sum-of products circuit with inputs A, B, C, what are min-terms m2, m3, m5, m7?

8) 8 points. Simplify the boolean equation Y = (ABC+A!BC)(B+C)(A)

9) 4 points. In CS120, we have class Monday, Wednesday, Friday, or on Thursday if it's the last day of classes. You have a circuit with inputs S (1 on Saturday and Sunday), T (1 on Tuesday and Thursday), and L (1 on the last day of classes). Design a circuit with Boolean gates that will be 1 if you should go to CS120.

10) 4 points. Draw a 2-input MUX (multiplexer)

11) 4 points. Sketch a finite state machine, with five states (you can number them 1 to 5). If an input is 1, the machine goes 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, …. If the input is 0, the machine moves to the next odd numbered state if it is currently on an even state, and then goes 5, 3, 1, 5, 3, 1, 5, 3, 1, ….

12) 8 points. Construct a Karnaugh map, and show the simplified circuit for the following truth table.

| A | B | C | Y |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | x |

13) 8 points. Write a MIPS assembly subroutine “swapArray”. It is passed two arguments: pointers to integer arrays (equal length, both terminated by zero). Your subroutine should swap the numbers in the arrays, so that A[i] is equal to the original B[i], and B[i] is equal to the original A[i] when you’re done.

14) 8 points. Prof. Madden isn’t very smart. He wrote a subroutine “addThree” that should add three integer arguments, and return the sum in register v0. His code has a bug; sometimes the number that is returned isn’t right. Can you fix the bug? Can you write the code differently, so that it’s correct, and faster?

Broken code…

```
addThree:
     add $v0, $v0, $a0
     add $v0, $v0, $a1
     add $v0, $v0, $a2
     jr $ra
```

15) 4 points. You have a pointer in register a0 to the C structure below. Write MIPS code for the line of C (the “obj” is your a0 pointer, and the -> is using the pointer to get to the individual fields within the structure).

```
struct
{
  int x;
  int y;
  int z;
} examObj;
```

```
obj->z = obj->x + obj->y;
```

16) 4 points. Write MIPS assembly to double a value (passed as a pointer). The C code looks like this. This is a subroutine.

```
times2(int *x)
{
  *x = *x + *x;
}
```

17) 8 points. Write MIPS assembly to multiply a value by 4 (passed as a pointer). The C code looks like this. This is also a subroutine.

```
times4(int *x)
{
  times2(x);
  times2(x);
}
```

18) 4 points. Name two types of hazards you might see on a pipelined microprocessor

19) 4 points. Modify the code below so that it could run faster on a pipelined microprocessor.

```
add $a0, $a3, $a4
add $a2, $a1, $a0
lui $a4, 4097
```

20) 1 point each. What are the abbreviations IPC, CPI, ALU, and PC?