Final Exam

Name: ____

Write your responses on the extra paper provided. Hand in this exam paper along with your solutions, please place your name on the top of each page. Show all of your work.

1 Short Answer

1. (*10 Points*) Draw and label the diagram for the von Neumann model of a uniprocessor computer.
2. (*10 Points*) Convert the hexadecimal number $F32AD154$ to both binary and octal.
3. (*10 Points*) Convert 12345 into binary, octal, and hexadecimal.
4. (*10 Points*) Compute $FFC123 + 1ED56ABD$ and $1ED56ABD - FFC123$, give your answer in hexadecimal.
5. (*10 Points*) Find the 2's complement of the byte 11010110, the most significant bit is the sign bit. What is the decimal representation of 11010110, again considering that the most significant bit is the sign bit and our machine uses 2's complement form.
6. (*15 Points*) Construct the truth table for both the half adder and the full adder. Also, give their logical expressions.
7. (*15 Points*) Given the following truth table,

A	B	C	P
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

   (a) Write the logical expression for the table in both canonical SOP and POS forms.
   (b) Using the SOP form, construct the circuit for this table.
   (c) Construct the K-Map for this table.
   (d) Give the reduced logical expression.
   (e) Draw the minimized circuit.

8. (*10 Points*) Design a 3-bit parallel adder using only half-adders and OR gates.
9. (*10 Points*) Using a clocked SR flip-flop, design the JK, D, and T flip-flops.
10. (*15 Points*) Using clocked D flip-flops, construct the circuitry for a clocked 4-bit parallel register with clear and load bit inputs. The inputs for the D flip-flop are just D and the clock, nothing else.

2 NASM

11. (*20 Points*) Recall that we define the factorial of a non-negative integer as

$$n! = \begin{cases} 1 & \text{if } n = 0 \\ n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot 1 & \text{if } n > 0 \end{cases}$$

For this exercise you will write a 64 bit assembly program that will be called by a C++ program to calculate a user input factorial. The C++ program will handle all the input and output and the assembly program will calculate the factorial.

Specifically, the C++ program will ask the user for the input, it will call the assembly program to do the calculation and return the result to the C++ program, then the C++ program will display the result.

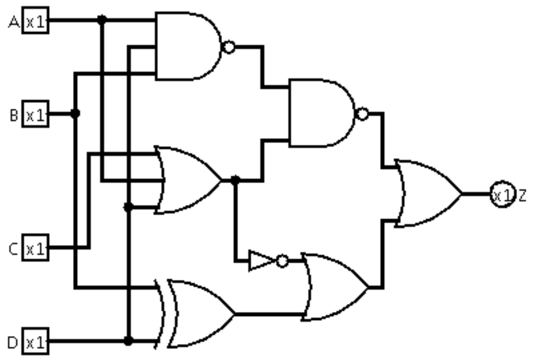
The assembly program will check to see if the input number (that came from the C++ program) is greater than or equal to 0. If the input is valid, the assembly program will do the factorial calculation and return the result. If the input was negative, then the assembly program will return -1. The C++ program is not to check the user's input, it will simply check the output of the assembly program, if the assembly program output is -1, the C++ program will display an error message and if not it will display the output of the assembly program.

Several runs of the program are below.

```
./prog
Input n: -3
Error: Input must be greater than or equal to 0.

./prog
Input n: 0
0! = 1

./prog
Input n: 12
12! = 479001600

./prog
Input n: 10
10! = 3628800

./prog
Input n: 20
20! = 2432902008176640000
```

3 Circuit Analysis

12. (*40 Points*) Do the following for the circuit below.



(a) Construct the truth table for the circuit.

(b) Write the circuit's logical function in canonical SOP form.

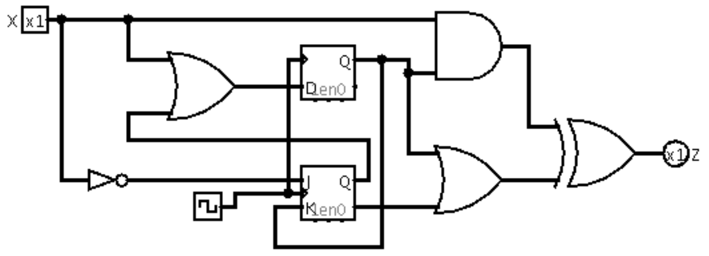
(c) Write the circuit in minterm form.

(d) Write the circuit in maxterm form.

(e) Write the K-Map for the circuit, show the groupings you would use, and then construct the minimized logical circuit function in SOP form.

(f) Using the K-Map work, write the circuit diagram of the minimized circuit.

13. (*40 Points*) For the following circuit, the top flip-flop is a D flip-flop and the bottom flip-flop is a JK flip-flop.



(a) Create the transition tables for the two flip-flops.

(b) Create the transition table.

(c) Create the next state table.

(d) Create the output table.

(e) Create the next state/output table.

(f) Create the state diagram.

Flip-Flop Characteristic Tables

$Q(t)$	SR	$Q(t+1)$
0	00	0
0	01	0
0	10	1
0	11	—
1	00	1
1	01	0
1	10	1
1	11	—

$Q(t)$	JK	$Q(t+1)$
0	00	0
0	01	0
0	10	1
0	11	1
1	00	1
1	01	0
1	10	1
1	11	0

$Q(t)$	D	$Q(t+1)$
0	0	0
0	1	1
1	0	0
1	1	1

$Q(t)$	T	$Q(t+1)$
0	0	0
0	1	1
1	0	1
1	1	0

Flip-Flop Excitation Tables

$Q(t)$	$Q(t+1)$	SR	D	JK	T
0	0	0d	0	0d	0
0	1	10	1	1d	1
1	0	01	0	d1	1
1	1	d0	1	d0	0