STUDENT NAME and NUMBER: _____

(PLEASE PRINT CLEARLY)

Georgia Institute of Technology
Department of Electrical and Computer Engineering

Test #2

ECE2030: Introduction to Computer Engineering

Marks:		
	Q1	/20
	Q2	/20
	Q3	/20
	Q4	/20
	Q5	/20
Total:		/100

DATE: October 26, 2000 **TIME:** 4:35 p.m. - 5:55 p.m.

REMARKS: 1. All questions should be answered (in the space provided).
2. Books and notes may NOT be used, with the exception of one 8.5"x11" sheet of paper.

1. NUMBER SYSTEMS AND ARITHMETIC *(MAX MARK: 20)*

a) Represent the following numbers in binary **two's complement form** using 5-bits.

1_{10} = ___ ___ ___ ___ ___ -8_{10} = ___ ___ ___ ___ ___

-1_{10} = ___ ___ ___ ___ ___ -7_{10} = ___ ___ ___ ___ ___

b) Add the following 5-bit **unsigned** numbers. Indicate if a range overflow has occured.

	01011 + 00111	01011 + 10010	11100 + 11010
Overflow yes or no?			

c) Add the following 5-bit **signed two's complement** numbers. Indicate if a range overflow has occured.

	01111 + 00011	00011 + 11010	11110 + 11000
Overflow yes or no?			

d) Find the **diminished radix complement** of the following octal number using 3 digits. (Hint: You can approach this using octal math or by converting to binary, performing the appropriate complement, and then converting back to octal.)

326_8 => ___ ___ ___ 8

2. COMBINATIONAL BUILDING BLOCKS *(MAX MARK: 20)*

a) Someone has mixed up the input columns of the truth table for a **4-to-2 priority encoder**. List out the priority order for each of the inputs In_k from highest priority to lowest priority.

Inputs				Outputs	
In_a	In_b	In_c	In_d	Z_1	Z_0
0	0	1	0	0	0
1	0	X	0	0	1
X	1	X	0	1	0
X	X	X	1	1	1

Highest priority _____

_____

_____

Lowest priority _____

b) Implement the logic circuit for the following truth table using a **3-to-8 decoder** and a *minimal* set of any other simple gates that may be required (AND, OR, NAND, NOR, NOT).

Inputs			Outputs	
X_2	X_1	X_0	Z_1	Z_0
0	0	0	1	0
0	0	1	1	0
0	1	0	0	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	0

c) Implement the logic circuit for the following truth table using an **8-to-1 multiplexer** and a *minimal* set of other simple gates that may be required (AND, OR, NAND, NOR, NOT).

Inputs			Output
X_2	X_1	X_0	Z
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

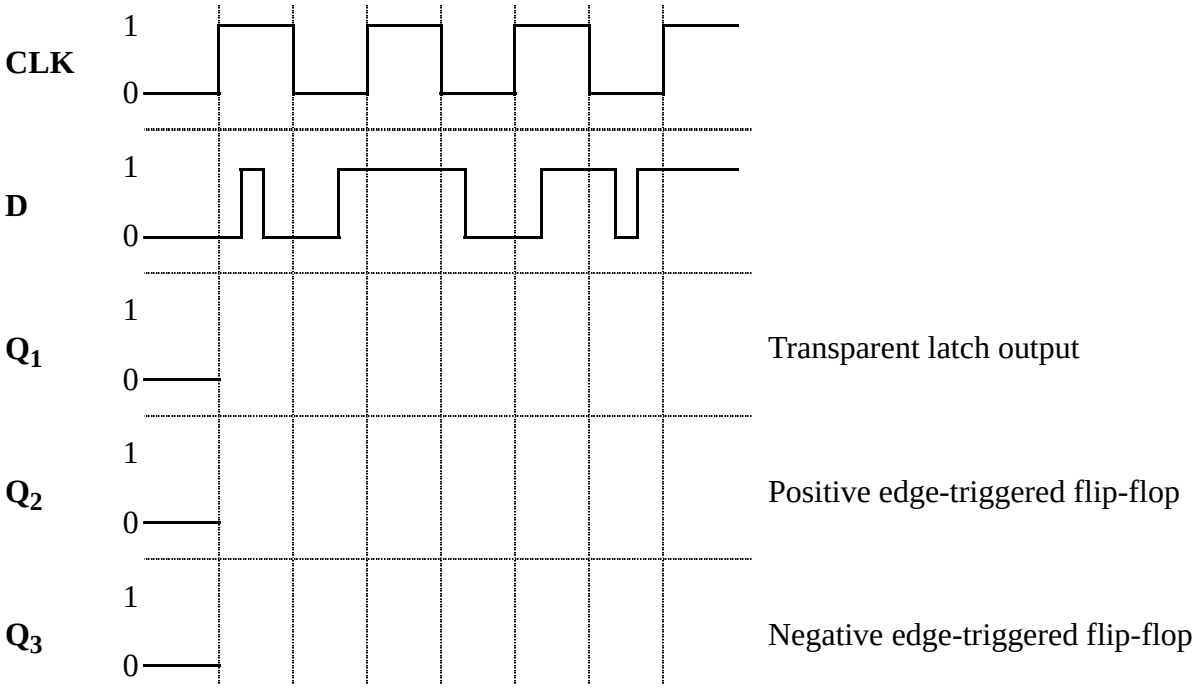
d) Slightly harder than the previous question, now implement the logic circuit for the same truth table using a **4-to-1 multiplexer** and a *minimal* set of other simple gates that may be required (AND, OR, NAND, NOR, NOT).

Inputs			Output
X_2	X_1	X_0	Z
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

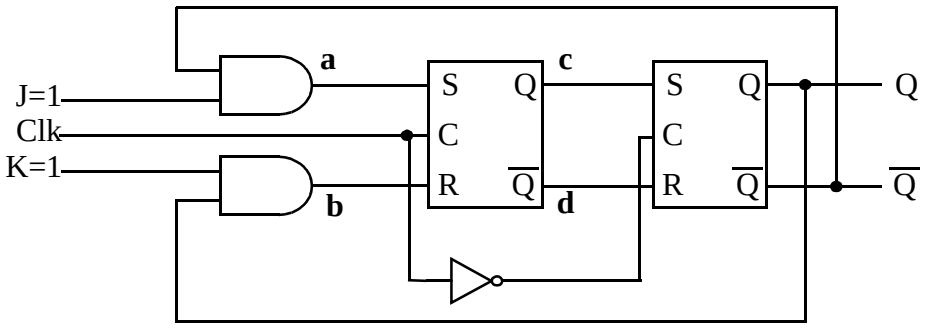
3. LATCHES AND REGISTERS *(MAX MARK: 20)*

a) Describe the differences between level-sensitive (or level-triggered), positive edge-triggered, and negative edge-triggered devices.

b) Complete the below timing diagram where $\mathbf{Q_1}$ is the output of a **transparent latch**, $\mathbf{Q_2}$ the output of a **positive edge-triggered flip-flop**, and $\mathbf{Q_3}$ the output of a **negative edge-triggered flip-flop**. These all have **D** as an input and **CLK** as the enable signal. Assume the latch and flip-flops are initially cleared.



c) Although we have not studied the JK flip-flop as illustrated below, it functions the same as the SR flip-flop except when the inputs are all high, which is normally indeterminate in an SR flip-flop. Fill in the two tables for the marked points in the circuit diagram for the given values of Q(*t*).



When Q(*t*) = 0

a	**b**	**c**	**d**	Q(*t*+1)

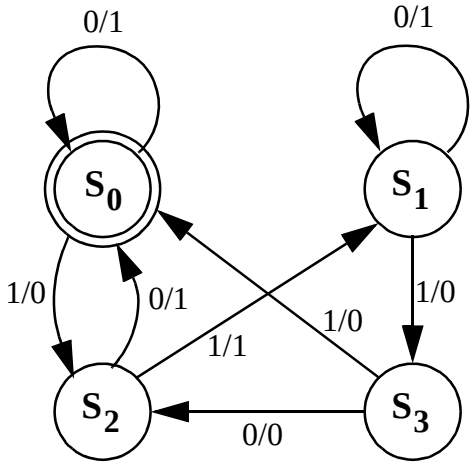
When Q(*t*) = 1

a	**b**	**c**	**d**	Q(*t*+1)

d) What is the operation being performed by the above JK flip-flop when the inputs are high? What type of cell is this similar too when the inputs are high?

4. FINITE STATE MACHINES *(MAX MARK: 20)*

a) Develop the **state table** for the following state diagram.



Present State	Input	Next State	Output

b) With the following **input stream**, apply it to the previous state machine and determine the output as well as the **ending state** at the end of the input stream.

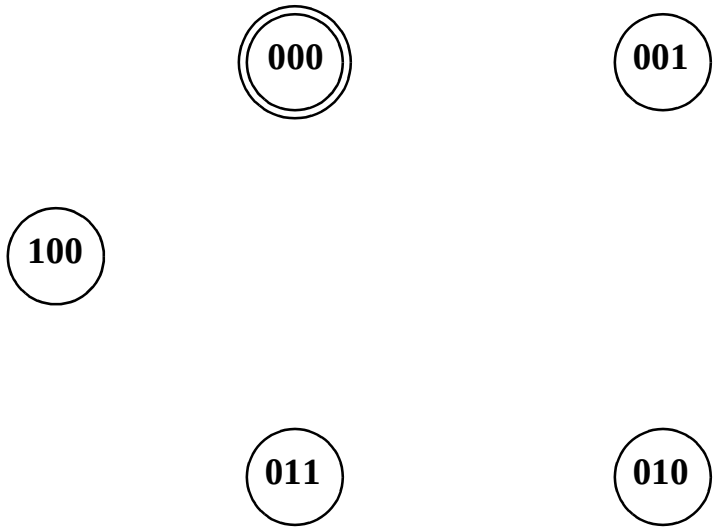
Input =	0	1	1	0	1	0	1	1	1	0	1
Output =											

Ending State = _____

c) Is the above state machine a **Mealy machine** or a **Moore machine**. Justify your answer!

d) A state machine is defined by the following state table. Draw the corresponding **state diagram** where $(S_2S_1S_0) = (000)$ is the initial state.

Present State S_2	S_1	S_0	Input	Next State N_2	N_1	N_0	Output
0	0	0	a	0	0	1	x
0	0	0	b	1	0	0	w
0	0	1	a	0	1	1	x
0	0	1	b	0	1	0	y
0	1	0	a	0	1	0	z
0	1	0	b	0	1	1	w
0	1	1	a	1	0	0	z
0	1	1	b	1	0	0	x
1	0	0	a	1	0	0	w
1	0	0	b	0	1	0	w



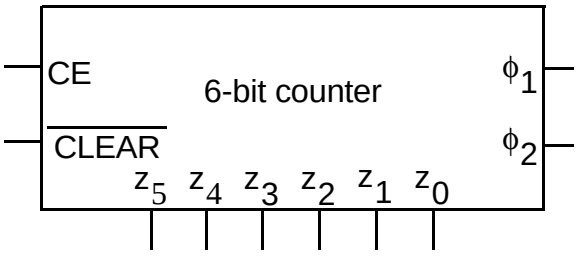
e) For the above state machine, are there any initial input combinations that would make it impossible to visit all of the states? If so, for what initial input combination(s), and which state(s) would be missed?

5. COUNTERS, SHIFT AND ROTATE REGISTERS *(MAX MARK: 20)*

a) Suppose you have a **4-bit bi-directional shift register** that can perform both **logical** and **two's complement arithmetic shifts**. The contents of the shift register is initially **0011**. When a serial shift-in value is required, it is set to **0**. The table below indicates what type of shift operation is performed at each clock pulse. Complete the table with how the contents of the shift register change at each clock pulse.

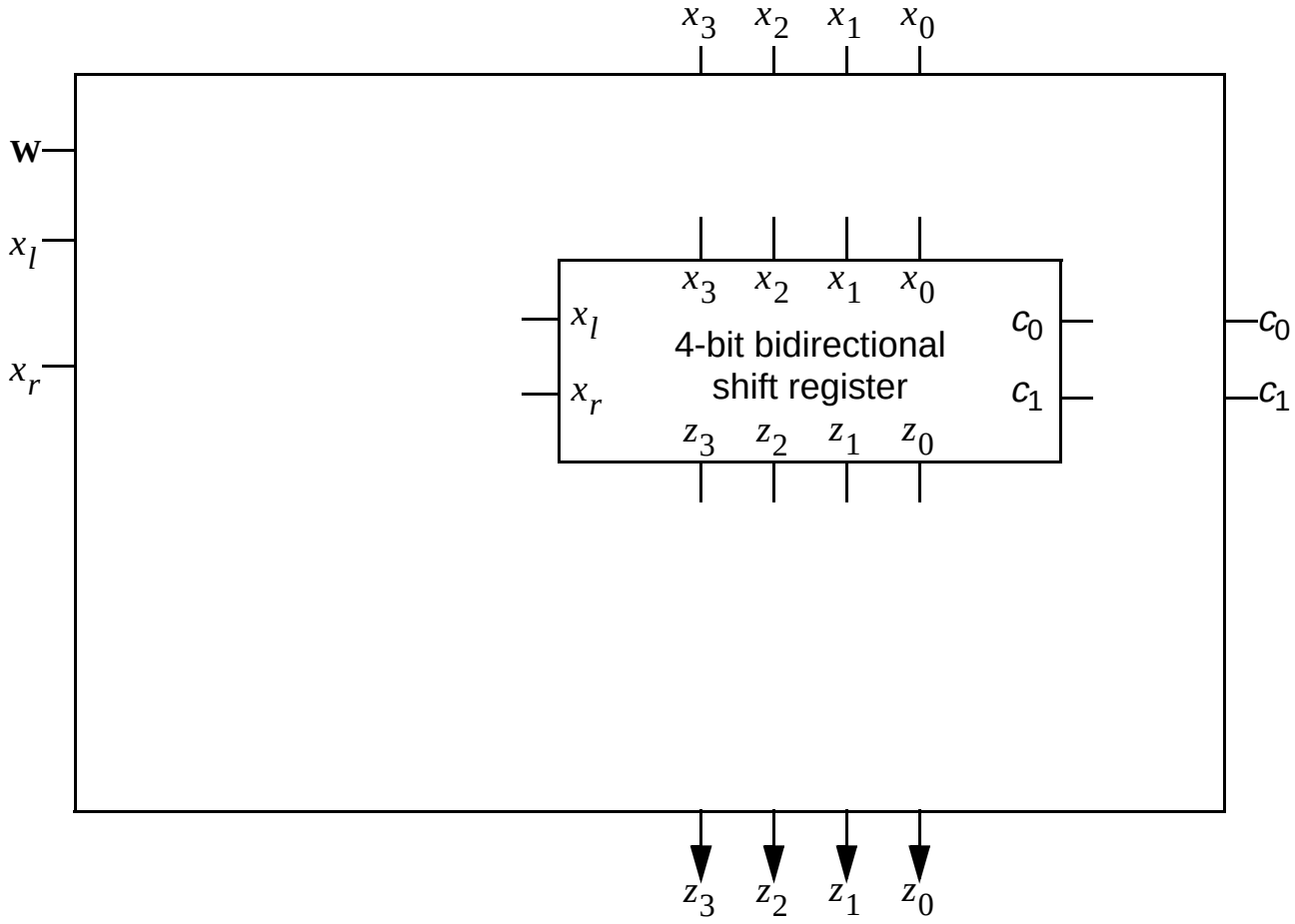
Clock Pulse #	Shift Operation	Shift Register Contents
0	—	0011
1	ASL	
2	LSL	
3	LSL	
4	ASR	
5	ASR	
6	LSR	
7	ASR	
8	ASL	

b) Given the following 6-bit counter, develop a **modulo-40 counter** (divide by 40 counter) with *minimal* external logic, but, that still includes the CE, $\overline{\text{CLEAR}}$, ϕ_1, and ϕ_2 inputs and the z outputs.



c) Below is a 4-bit bi-directional shift register where (x_3, x_2, x_1, x_0) is a parallel input, (z_3, z_2, z_1, z_0) is the contents of the register, x_l and x_r are serial shift inputs for the left and right shifting, respectively, and $(\mathbf{c}_1, \mathbf{c}_0)$ controls the shifting properties of the register.

Given this 4-bit bi-directional shift register that can perform *logical shifts*, design a **new 4-bit bi-directional shift register** that can perform both **logic shifts** and **two's complement arithmetic shifts** using *minimal* external logic. Include one extra input, **W**, that indicates whether to perform a logical shift (**W** = **0**) or an arithmetic shift (**W** = **1**).



d) Given an **8-bit ripple counter**, how many flip-flop values will be complemented to get the next count value if the following are the current counter contents (MSB on the left, LSB on the right).

Current counter contents	# of complements to next value
01101111	
00011111	
00011010	