Sample Solutions

CprE 281: Digital Logic
Midterm 2: Friday Oct 27, 2017

Student Name: ______________________ **Student ID Number:** ____________

Lab Section:	**Mon 9-12(N)**	**Tue 11-2(U)**	**Wed 8-11(J)**	**Thur 11-2(Q)**	**Fri 11-2(G)**
(circle one)	**Mon 12-3(P)**	**Tue 2-5(M)**	**Wed 11-2(W)**	**Thur 11-2(V)**	
		Tue 2-5(Z)	**Wed 6-9(T)**	**Thur 2-5(L)**	
				Thur 5-8(K)	

1. True/False Questions (10 x 1p each = 10p)

(a) I forgot to write down my name and student ID number and lab section. TRUE / [✓ FALSE]

(b) When K=$\overline{J}$ a JK flip-flop behaves just like a D flip-flop. [✓ TRUE] / FALSE

(c) Any Boolean function can be implemented using only 1-to-4 demultiplexers. [✓ TRUE] / FALSE

(d) The outputs of a demultiplexer are one-hot encoded. TRUE / [✓ FALSE]

(e) A D-Latch changes its value only when the clock is 1. [✓ TRUE] / FALSE

(f) XOR(XOR(A, B), B) = A. [✓ TRUE] / FALSE

(g) In 4-bit 2's complement: 1010 + 0111 results in an overflow. TRUE / [✓ FALSE]

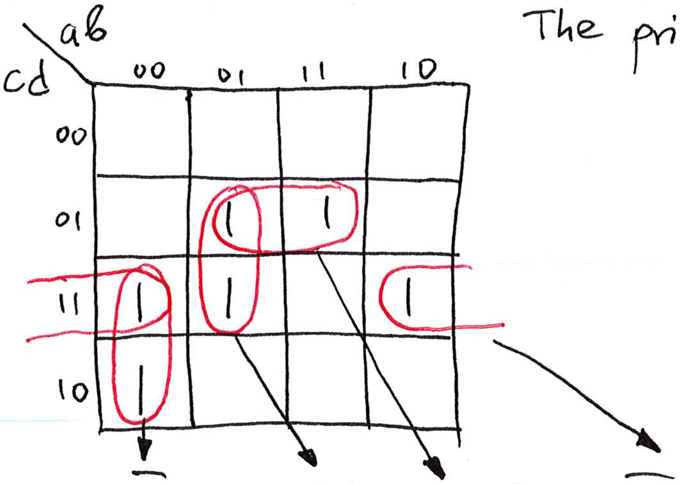
(h) In 16-bit 2's complement: $\text{FACE}_{16} < \text{CAFE}_{16}$ TRUE / [✓ FALSE]

(i) A D-flip flop and a NOT gate can be used to implement a T-flip flop. TRUE / [✓ FALSE]

(j) According to George Lucas, Princes Leia got her Ph.D. at age 19. [✓ TRUE] / FALSE

2. Prime Numbers (5p)
Let f(a, b, c, d) be a Boolean function that is equal to 1 if and only if the 4-bit number abcd is prime. Use a K-map to derive the minimum SOP expression for the function f.

The prime numbers from 0 to 15:
2, 3, 5, 7, 11, and 13



cd \ ab	00	01	11	10
00				
01		1	1	
11	1	1		1
10	1			

$$f = \bar{a}\bar{b}c + \bar{a}bd + b\bar{c}d + \bar{b}cd$$

3. **Binary Addition and Subtraction (4 x 3p each = 12p)**

Convert the following integers into binary numbers and perform the addition or subtraction using 2's complement if necessary. Write your answers and all intermediary steps to the right of each problem. Use 5-bit numbers for all problems and indicate if any bits need to be ignored.

$$\begin{array}{rr} & (+2) \\ + & (+6) \\ \hline & (+8) \end{array} \qquad \begin{array}{rr} & 00010 \\ + & 00110 \\ \hline & 01000 \end{array}$$

$$\begin{array}{rr} & (-5) \\ + & (-3) \\ \hline & (-8) \end{array} \qquad \begin{array}{rl} & (-00101) \Rightarrow \\ + & (-00011) \Rightarrow \\ \hline & \end{array} \qquad \begin{array}{rr} & 11011 \\ + & 11101 \\ \hline & \boxed{1}\,11000 \end{array}$$

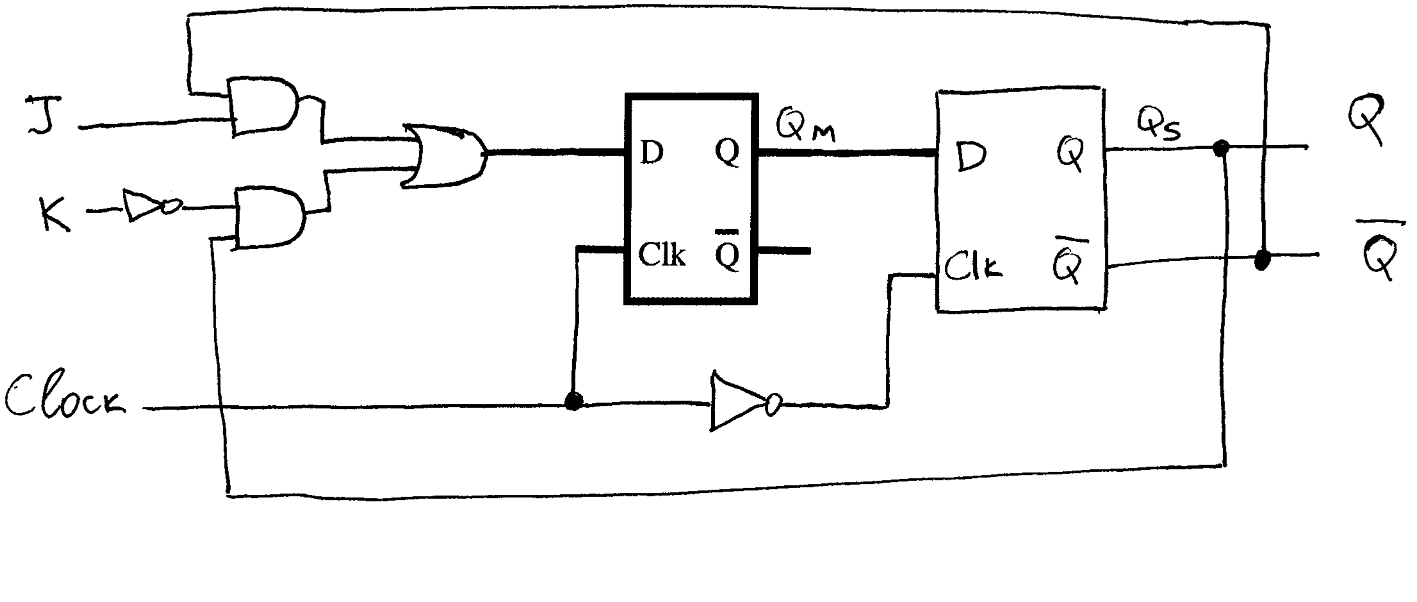
ignore

$$\begin{array}{rr} & (+7) \\ - & (+4) \\ \hline & (+3) \end{array} \qquad \begin{array}{rl} & 00111 \\ - & 00100 \Rightarrow \\ \hline & \end{array} \qquad \begin{array}{rr} & 00111 \\ + & 11100 \\ \hline & \boxed{1}\,00011 \end{array}$$

ignore

$$\begin{array}{rr} & (-6) \\ - & (-3) \\ \hline & (-3) \end{array} \qquad \begin{array}{rl} & (-00110) \Rightarrow \\ - & (-00011) \Rightarrow \\ \hline & \end{array} \qquad \begin{array}{rr} & 11010 \\ + & 00011 \\ \hline & 11101 \end{array}$$

4. **JK Flip-Flop (8pt)**
Complete the wiring diagram below to implement a negative-edge-triggered JK flip-flop. Clearly label all inputs and outputs.



5. **Code Converter (5p + 5p = 10p).**
Your task is to design a code converter with two inputs (X_1 and X_0) and four outputs (Y_3, Y_2, Y_1, and Y_0) that has the following truth table:

X_1	X_0	Y_3	Y_2	Y_1	Y_0
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

a) Derive the correct Boolean expressions for the four outputs.

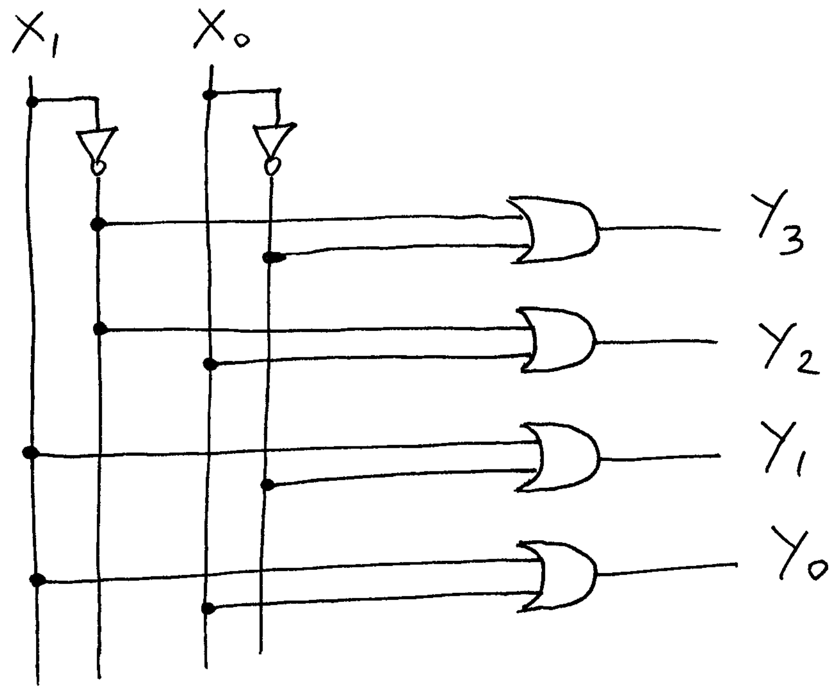
$Y_3 = \overline{X_1} + \overline{X_0}$

$Y_2 = \overline{X_1} + X_0$

$Y_1 = X_1 + \overline{X_0}$

$Y_0 = X_1 + X_0$

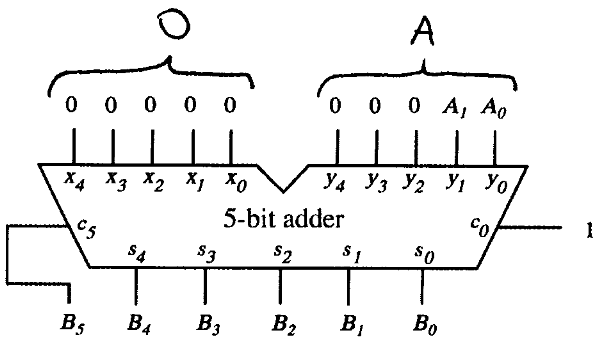
b) Draw the full wiring diagram of this code converter using basic logic gates. Label all inputs and outputs.



6. Computations with Adders (5 x 3p each = 15p)

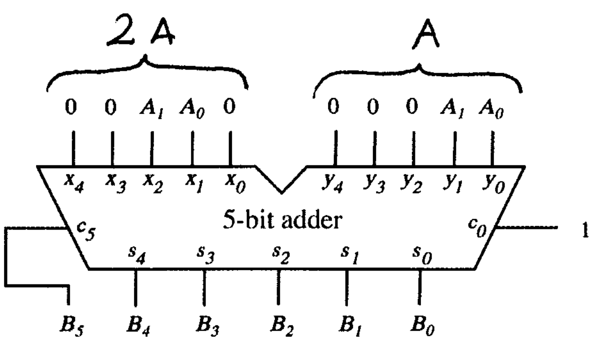
Let A = A_1 A_0 be a 2-bit <u>unsigned</u> binary number. Let B = B_5 B_4 B_3 B_2 B_1 B_0 be the result of the operation, interpreted as a number in 2's complement representation. For each of the following, write a formula for the value of B. The problem in a) is already solved.

a)


0
A
0 0 0 0 0
0 0 0 A_1 A_0
x_4 x_3 x_2 x_1 x_0
y_4 y_3 y_2 y_1 y_0
5-bit adder
c_5
c_0
1
s_4 s_3 s_2 s_1 s_0
B_5 B_4 B_3 B_2 B_1 B_0

$B = A + 1$

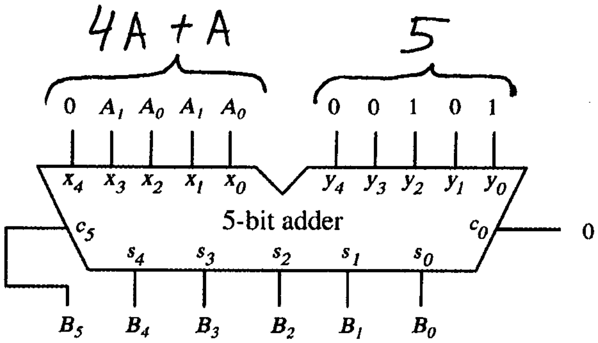
b)


2A
A
0 0 A_1 A_0 0
0 0 0 A_1 A_0
x_4 x_3 x_2 x_1 x_0
y_4 y_3 y_2 y_1 y_0
5-bit adder
c_5
c_0
1
s_4 s_3 s_2 s_1 s_0
B_5 B_4 B_3 B_2 B_1 B_0

$B = 2A + A + 1$

$= 3A + 1$

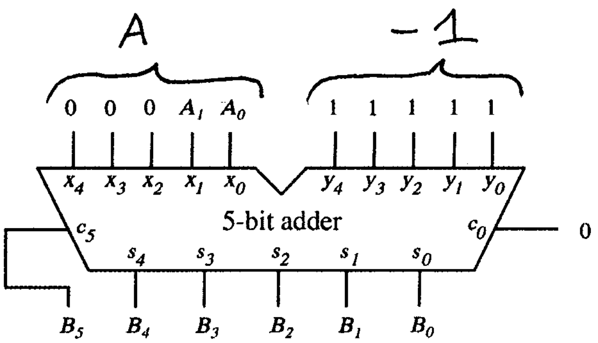
c)


4A + A
5
0 A_1 A_0 A_1 A_0
0 0 1 0 1
x_4 x_3 x_2 x_1 x_0
y_4 y_3 y_2 y_1 y_0
5-bit adder
c_5
c_0
0
s_4 s_3 s_2 s_1 s_0
B_5 B_4 B_3 B_2 B_1 B_0

$B = 4A + A + 5 + 0$

$= 5A + 5$

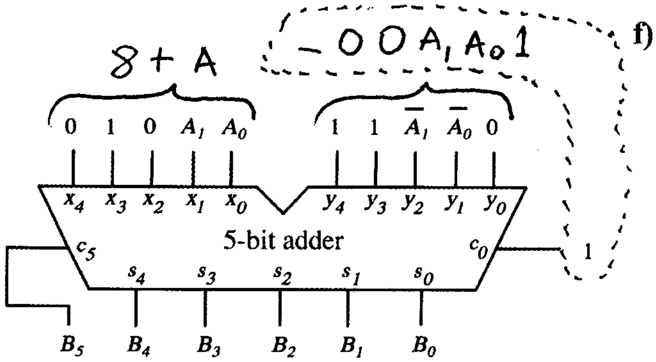
d)


A
−1
0 0 0 A_1 A_0
1 1 1 1 1
x_4 x_3 x_2 x_1 x_0
y_4 y_3 y_2 y_1 y_0
5-bit adder
c_5
c_0
0
s_4 s_3 s_2 s_1 s_0
B_5 B_4 B_3 B_2 B_1 B_0

$B = A + (-1) + 0$

$= A - 1$

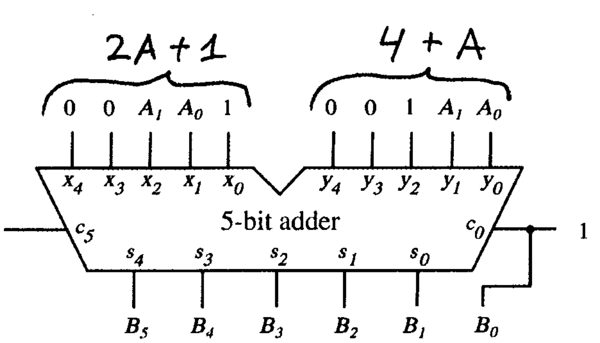
e)


8 + A
− 0 0 A_1 A_0 1
0 1 0 A_1 A_0
1 1 $\overline{A_1}$ $\overline{A_0}$ 0
x_4 x_3 x_2 x_1 x_0
y_4 y_3 y_2 y_1 y_0
5-bit adder
c_5
c_0
1
s_4 s_3 s_2 s_1 s_0
B_5 B_4 B_3 B_2 B_1 B_0

$B = 8 + A - (2A + 1)$

$= 7 - A$

f)


2A + 1
4 + A
0 0 A_1 A_0 1
0 0 1 A_1 A_0
x_4 x_3 x_2 x_1 x_0
y_4 y_3 y_2 y_1 y_0
5-bit adder
c_5
c_0
1
s_4 s_3 s_2 s_1 s_0
B_5 B_4 B_3 B_2 B_1 B_0

$B = (2A + 1 + 4 + A + 1)2 + 1$

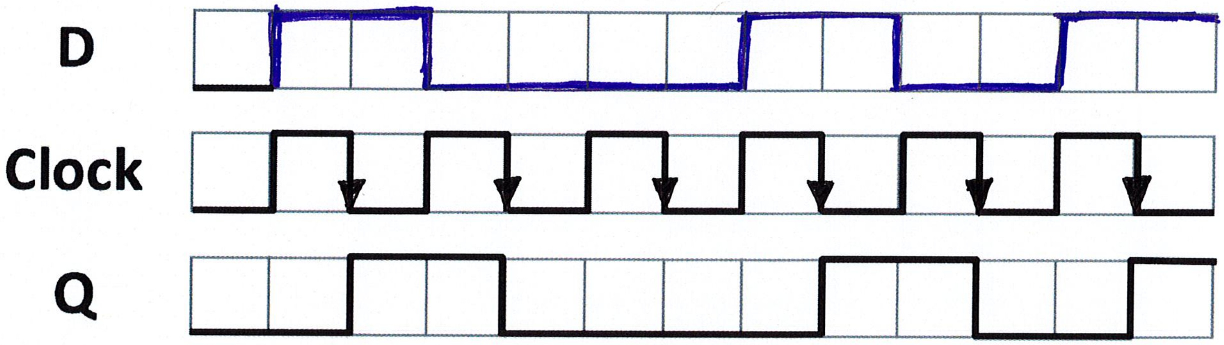
$= (3A + 6)2 + 1$

$= 6A + 13$

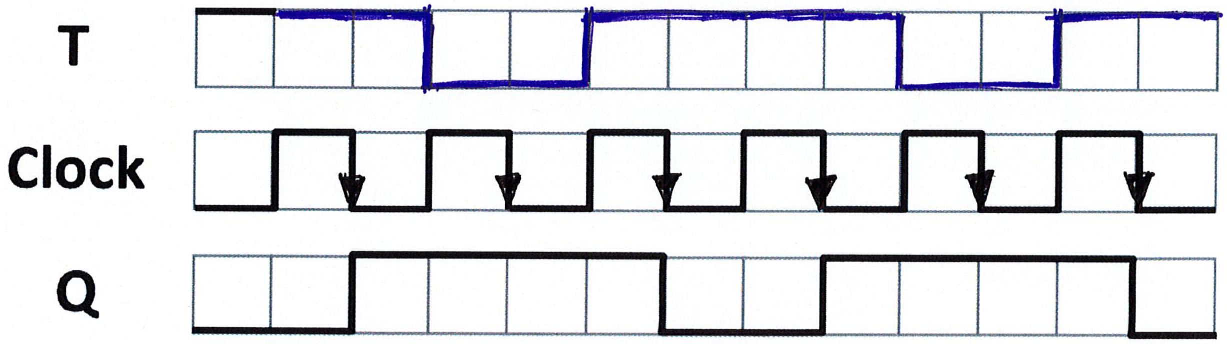
7. Flip-Flops and Timing Diagrams (3 x 5p = 15p)

Complete the timing diagram for the specified flip-flop such that the output Q will be as indicated. Assume that the input signal can change only on the vertical lines. Also, assume that the setup time t_{su} and the hold time t_h are each equal to the width of one square.

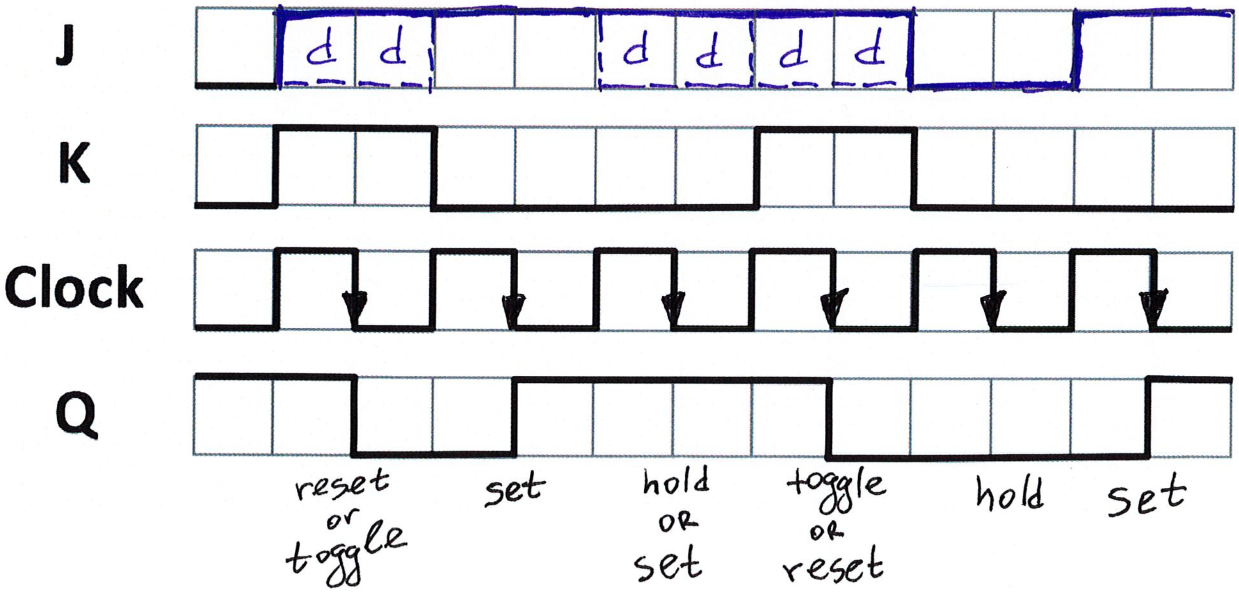
a) Complete the timing diagram for the D input to a negative-edge triggered D flip-flop.



b) Complete the timing diagram for the T input to a negative-edge triggered T flip-flop.



c) Complete the timing diagram for the J input to a negative-edge triggered JK flip-flop.



8. Multiplexers (3 x 5p each = 15p)

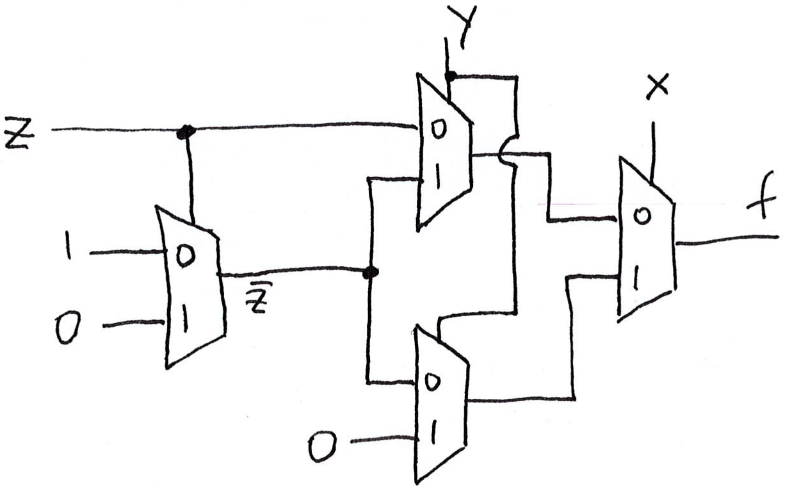
a) Draw the truth table for the function $f(x, y, z) = \overline{x}\,\overline{y}\,z + \overline{x}\,y\,\overline{z} + x\,\overline{y}\,\overline{z}$.

X	Y	Z	$\overline{X}\,\overline{Y}\,Z$	$\overline{X}\,Y\,\overline{Z}$	$X\,\overline{Y}\,\overline{Z}$	f	
0	0	0	0	0	0	0	XOR (Y, Z)
0	0	1	1	0	0	1	
0	1	0	0	1	0	1	
0	1	1	0	0	0	0	
1	0	0	0	0	1	1	NOR (Y, Z)
1	0	1	0	0	0	0	
1	1	0	0	0	0	0	
1	1	1	0	0	0	0	

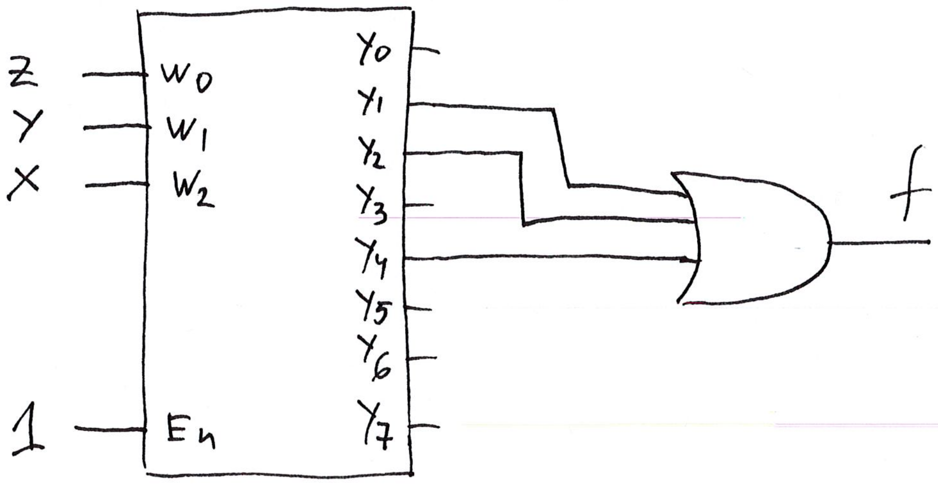
b) Implement this function using **only** 2-to-1 multiplexers and **no other logic gates**. Assume that the signals x, y, and z are available **only** in their non-inverted form as well as the constants 0 and 1. Clearly label all inputs, outputs, and pins.

Y	Z	XOR	
0	0	0	Z
0	1	1	
1	0	1	$\overline{Z}$
1	1	0	

Y	Z	NOR	
0	0	1	$\overline{Z}$
0	1	0	
1	0	0	0
1	1	0	



c) Implement this function using a 3-to-8 decoder (with enable) and one OR gate. Clearly label all inputs, outputs, and pins of the decoder.



9. Alternative Implementation (10p)

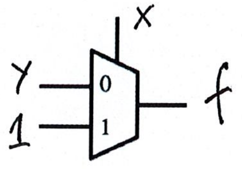
Implement the logic gates OR, XOR, and NAND in three different ways: 1) using a 2-to-1 multiplexer; 2) using AND-OR logic; and 3) using OR-AND logic. In this problem, you are not allowed to use any other logic gates. Assume that both x and y are available in their inverted and non-inverted form, along with the constants 0 and 1. If some implementation is not possible, then indicate that with words. Label all inputs and outputs.

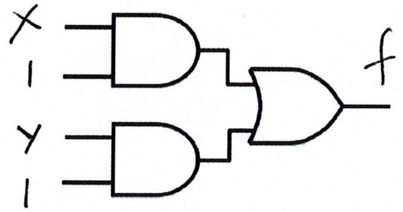
a) Implement in three different ways: f = OR(x, y).

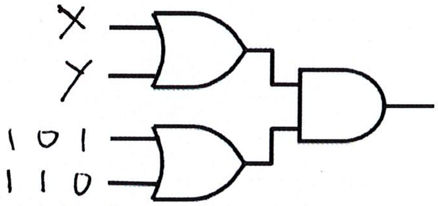
X	Y	OR	
0	0	0	} Y
0	1	1	
1	0	1	} 1
1	1	1	

$Y \backslash X$		
	0	1
	1	1

X
Y





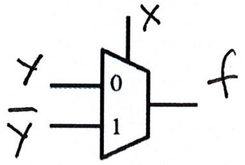


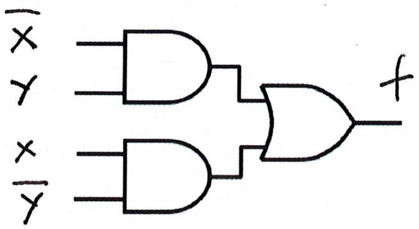
one of three options

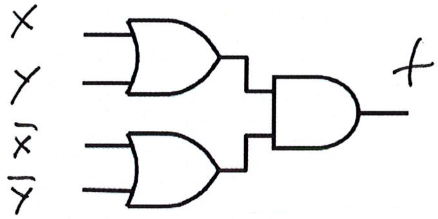
b) Implement in three different ways: f = XOR(x, y).

X	Y	XOR	
0	0	0	} Y
0	1	1	
1	0	1	} $\overline{Y}$
1	1	0	

$Y \backslash X$		
	0	1
	1	0





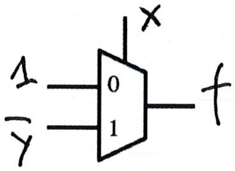


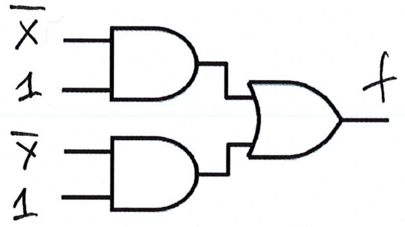
c) Implement in three different ways: f = NAND(x, y).

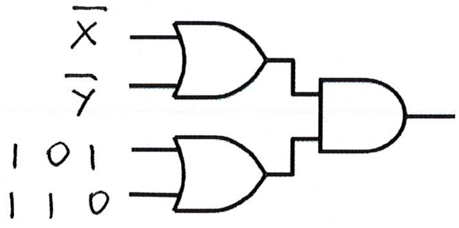
X	Y	NAND	
0	0	1	} 1
0	1	1	
1	0	1	} $\overline{Y}$
1	1	0	

$Y \backslash X$		
	1	1
	1	0

$\overline{X}$
$\overline{Y}$







one of three options

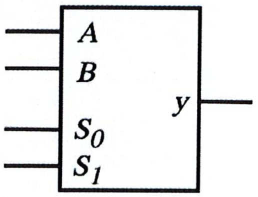
10. Mystery Circuit (5p + 10p = 15p)

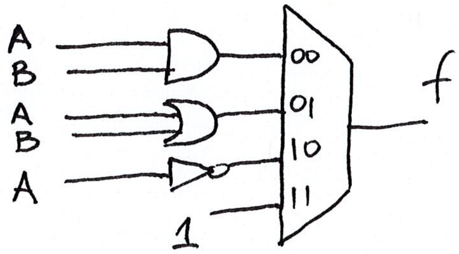
(a) The truth table and the graphical symbol for a mystery circuit are shown below. Implement this circuit using a 4-to-1 multiplexer and any number of basic logic gates (AND, OR, NOT). Assume that the input variables are available only in their regular (i.e., non-inverted form) as well as the constants 0 and 1. Label all inputs, outputs, and pins.

Truth table

S_1	S_0	A	B	y	
0	0	0	0	0	AND(A,B)
0	0	0	1	0	
0	0	1	0	0	
0	0	1	1	1	
0	1	0	0	0	OR(A,B)
0	1	0	1	1	
0	1	1	0	1	
0	1	1	1	1	
1	0	0	0	1	NOT(A)
1	0	0	1	1	
1	0	1	0	0	
1	0	1	1	0	
1	1	0	0	1	1
1	1	0	1	1	
1	1	1	0	1	
1	1	1	1	1	

Graphical symbol

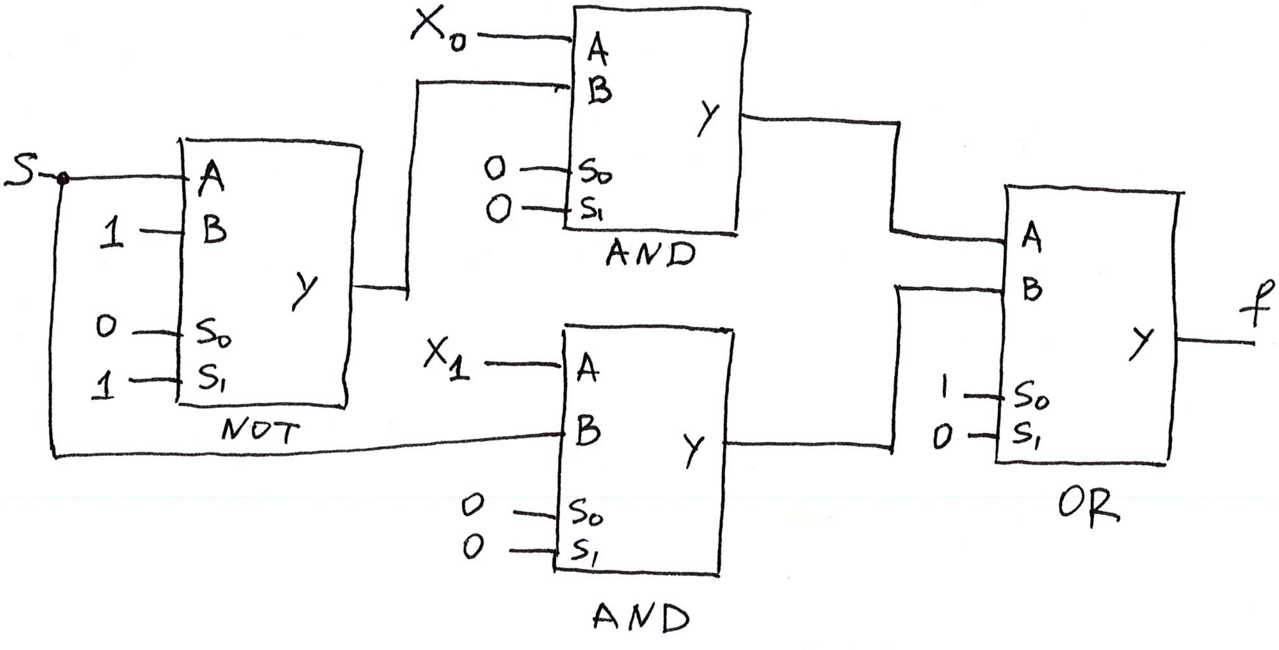




(b) Implement a 2-to-1 multiplexer using only copies of the mystery circuit and no other logic gates. Please use the graphical symbol of the mystery circuit to denote each copy. You can assume that the input variables are available only in their regular (i.e., non-inverted form) as well as the constants 0 and 1. Label all inputs, outputs, and pins of your circuit.

$f = \bar{S} X_0 + S X_1$

Need 2 ANDs, 1 OR, 1 NOT. The mystery circuit can do all of these.

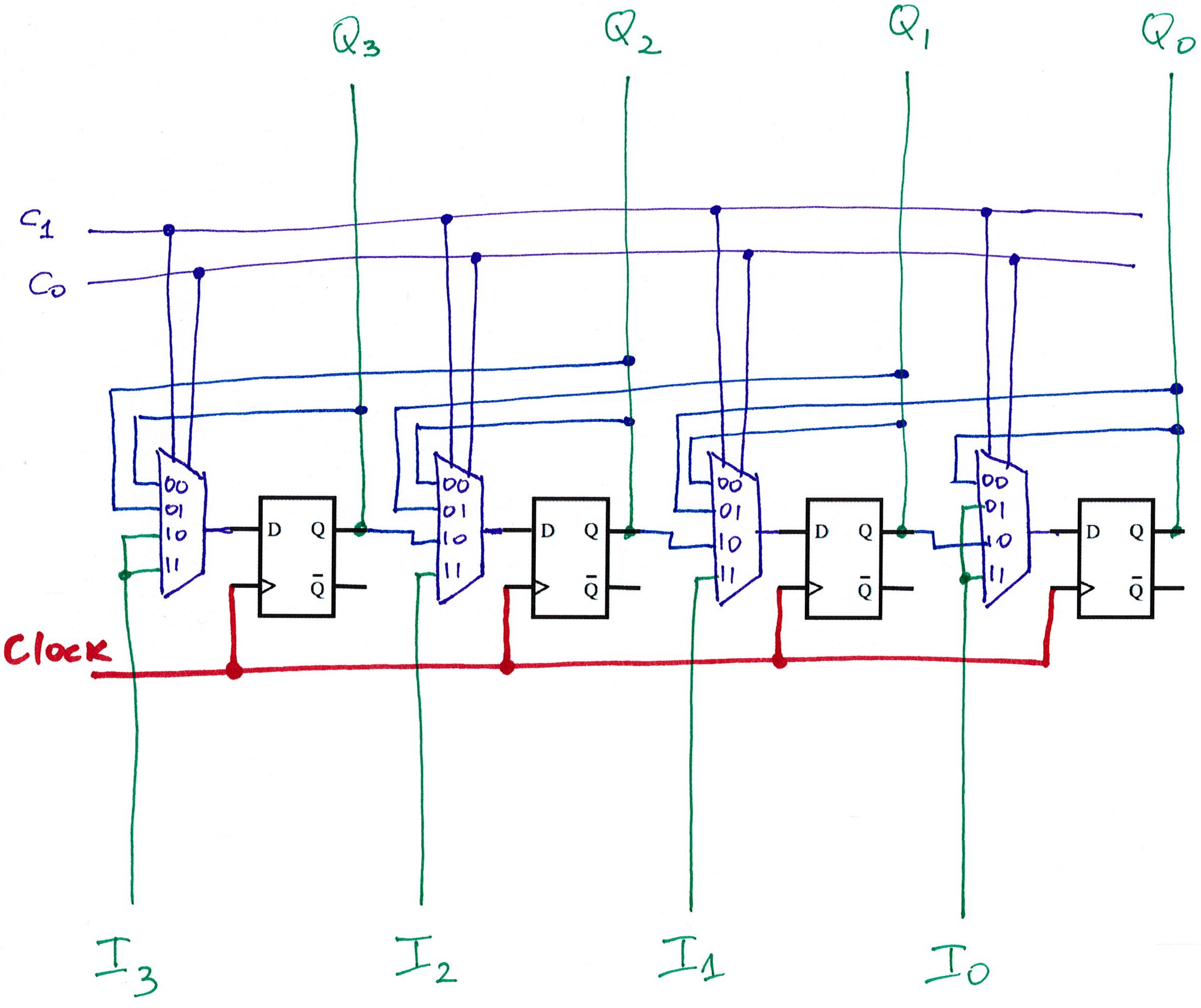


11. Parallel Load and Shift Left/Right Register (15p)

Complete the following circuit diagram to implement a 4-bit register that has both parallel load and shift left/right functionality. The register has two control inputs (C_1 and C_0), four parallel input lines (I_3, I_2, I_1, and I_0), and four output lines (Q_3, Q_2, Q_1, and Q_0). Depending on the values of C_1 and C_0, the register performs one of the following four operations:

C_1	C_0	Operation
0	0	Hold the current value (i.e., Q_3 Q_2 Q_1 Q_0 are not changed)
0	1	Shift left (i.e., new Q_3=Q_2, new Q_2=Q_1, new Q_1=Q_0, new Q_0=I_0)
1	0	Shift right (i.e., new Q_3=I_3, new Q_2=Q_3, new Q_1=Q_2, new Q_0=Q_1)
1	1	Load new data (i.e., new Q_3=I_3, new Q_2=I_2, new Q_1=I_1, new Q_0=I_0)

Clearly label all inputs, outputs, and pins.



Question	**Max**	**Score**
1. True/False	10	
2. Prime Numbers	5	
3. Addition/Subtraction	12	
4. JK Flip-Flop	8	
5. Code Converter	10	
6. Computations with Adders	15	
7. Flip-Flops	15	
8. Multiplexers	15	
9. Alternative Implementation	10	
10. Mystery Circuit	15	
11. Register	15	
TOTAL:	130	