Sample Solutions

CprE 281: Digital Logic
Midterm 2: Friday Nov. 1, 2019

Name: ______________________________ **ID Number:**______________

Lab Section:	**Mon 12-3 (#13)**	**Tue 11-2 (#16)**	**Wed 8-11 (#8)**	**Thur 11-2 (#14)**	**Fri 11-2 (#7)**
(circle one)		**Tue 2-5 (#11)**	**Wed 11-2 (#18)**	**Thur 11-2 (#17)**	
		Tue 2-5 (#20)		**Thur 2-5 (#10)**	
				Thur 5-8 (#9)	

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

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

(b) Removing a NOT gate from a D Flip-Flop makes it positive-edge triggered. TRUE / [✓ FALSE]

(c) A 4-to-1 multiplexer can be built with 7 basic logic gates (AND, NOT, OR). [✓ TRUE] / FALSE

(d) An XNOR gate can be constructed using only one 2-to-1 multiplexer. [✓ TRUE] / FALSE

(e) The parallel input lines of an 8-bit register are one-hot encoded. TRUE / [✓ FALSE]

(f) Shannon mapped Boolean logic to switches in his ~~PhD~~ Master's thesis dissertation. TRUE / [✓ FALSE]

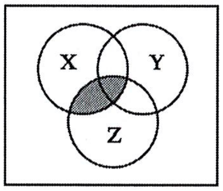
(g) Big Endian is used more often today than Little Endian. TRUE / [✓ FALSE]

(h) In 1's complement notation it is possible to have both +0 and -0. [✓ TRUE] / FALSE

(i) In 32-bit IEEE 754 format it is possible to have both +0.0 and -0.0. [✓ TRUE] / FALSE

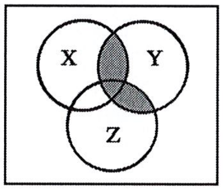
(j) Chewbacca's home planet is called Endor. TRUE / [✓ FALSE]

2. Venn Diagrams (5 x 1p each = 5p)
Write the Boolean expression that corresponds to each Venn diagram below each figure.



(A)

XZ

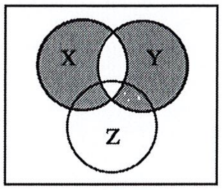


(B)

$XY + YZ$

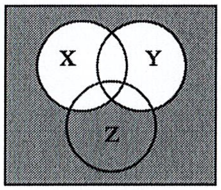
OR

$Y(X+Z)$



(C)

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

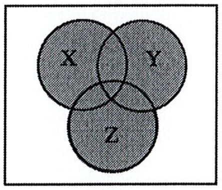


(D)

$\overline{X+Y}$

OR

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



(E)

$X+Y+Z$

3. Minimization with K-maps (3 x 5p each = 15p)

a) Draw the truth table for $f(a,b,c,d)=\Pi M(0, 2, 3, 7, 14) + D(8,15)$. (5p)

b) Use a K-map to derive the minimum-cost SOP expression for this function. (5p)

c) Use a K-map to derive the minimum-cost POS expression for this function. (5p)

(a)

a	b	c	d	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	d
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	d

(b)

cd \ ab	00	01	11	10
00	0	1	1	d
01	1	1	1	1
11	0	0	d	1
10	0	1	0	1

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

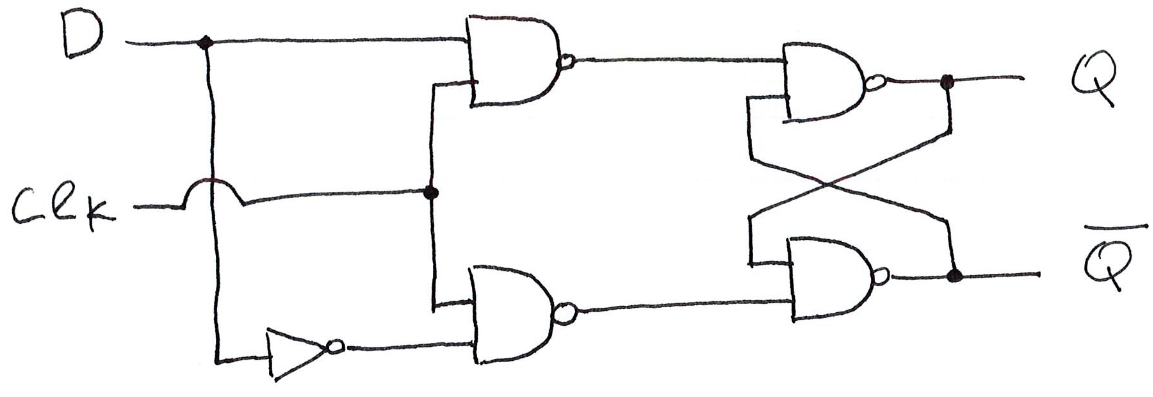
(c)

cd \ ab	00	01	11	10
00	0	1	1	d
01	1	1	1	1
11	0	0	d	1
10	0	1	0	1

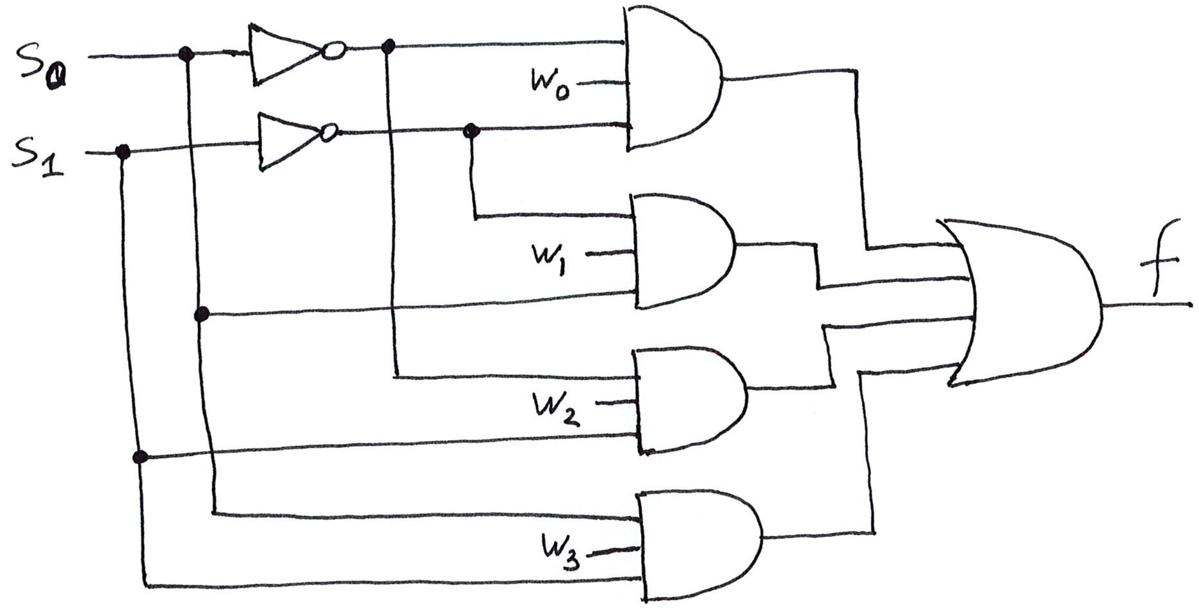
$$f = (a+b+d)\cdot(a+\bar{c}+\bar{d})\cdot(\bar{a}+\bar{b}+\bar{c})$$

4. Basic Circuits (3 x 5p each = 15p).
In all sub-problems, draw the complete wiring diagram using logic gates (no high-level graphical symbols are allowed in this problem). Clearly label all inputs and outputs.

(a) Gated D Latch (with NAND gates for the latch).

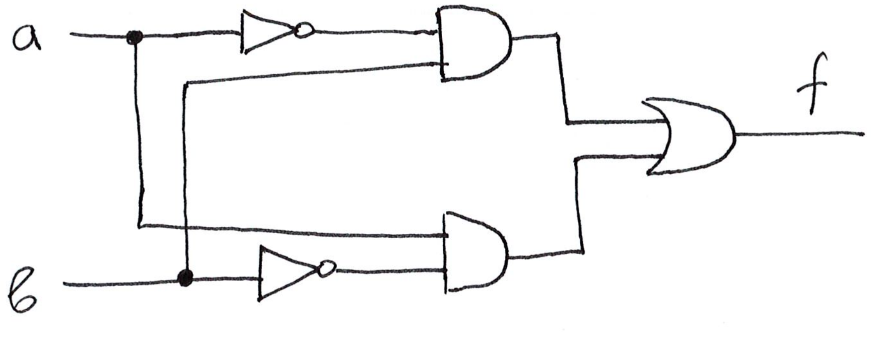


(b) 4-to-1 multiplexer.



(c) XOR gate, implemented using only AND, OR, and NOT gates.

$XOR(a, b) = \bar{a}b + a\bar{b}$



5. Number Conversions (3p + 4p + 4p + 4p = 15p)

(a) Convert 163_{10} to binary.

$163/2 = 81 \quad 1$

$81/2 = 40 \quad 1$

$40/2 = 20 \quad 0$

$20/2 = 10 \quad 0$

$10/2 = 5 \quad 0$

$5/2 = 2 \quad 1$

$2/2 = 1 \quad 0$

$1/2 = 0 \quad 1$

10100011_2

(b) Convert the following 32-bit float number (in IEEE 754 format) to decimal.

$2^{-1}\ 2^{-2}\ 2^{-3}\ 2^{-4}$

$\overset{\text{negative}}{1}\ \underbrace{1\,0\,0\,0\,0\,0\,1\,1}_{128+2+1=131}\ 0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0$

$$(-1)^1 \times 2^{131-127} \times (1+2^{-4}) = -2^4(1+2^{-4}) = -2^{+4} \underbrace{-2^{+4}\cdot 2^{-4}}_{-1} =$$

$$= -16 - 1 = -17$$

(c) Write down the 32-bit floating point representation (in IEEE 754 format) for 15.0

$15/8 = 1.875$

```
 15
-8
 70
-64
  60
 -56
   40
  -40
    0
```

$8 = 2^3 = 2^{130-127}$

$1.875 = 1 + 0.5 + 0.25 + 0.125$

$0 \vdots 10000010 \vdots 111\underbrace{00\ldots0}_{20 \text{ zeros}}$

(d) Convert -53_{10} to an 8-bit binary number in 2's complement representation.

$53/2 = 26 \quad 1$

$26/2 = 13 \quad 0$

$13/2 = 6 \quad 1$

$6/2 = 3 \quad 0$

$3/2 = 1 \quad 1$

$1/2 = 0 \quad 1$

110101

pad to 8-bit

00110101

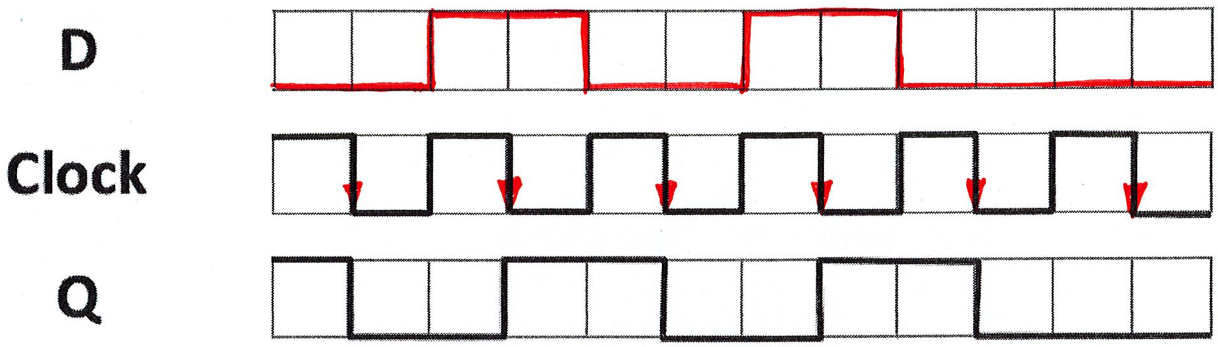
negate

11001011

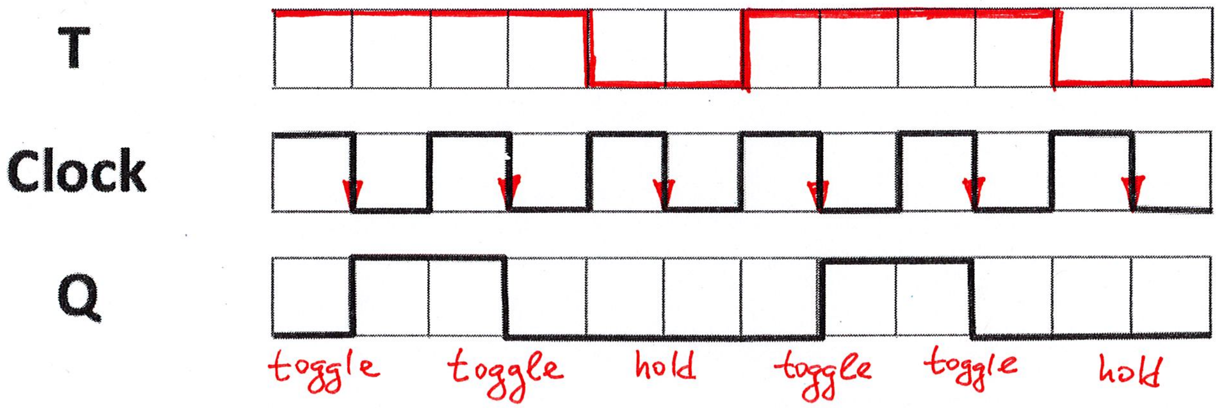
6. Flip-Flops and Timing Diagrams (3 x 5p = 15p)

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.

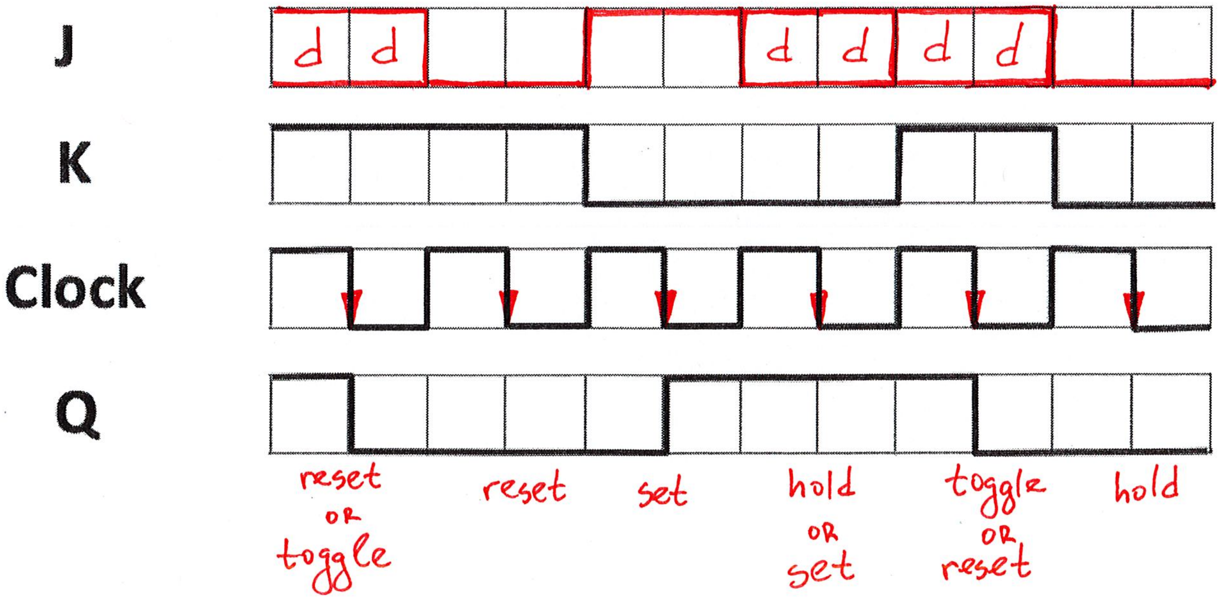
a) Complete the timing diagram for the D input to a negative-edge triggered D flip-flop.



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.



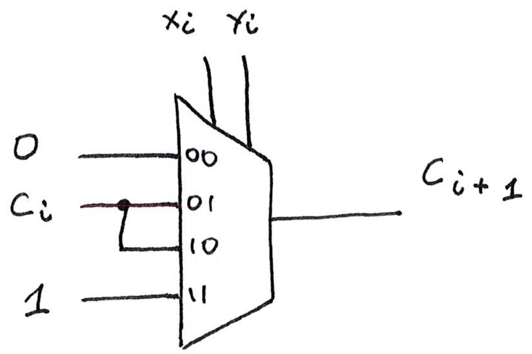
7. Full Adder (5p + 10p = 15p)

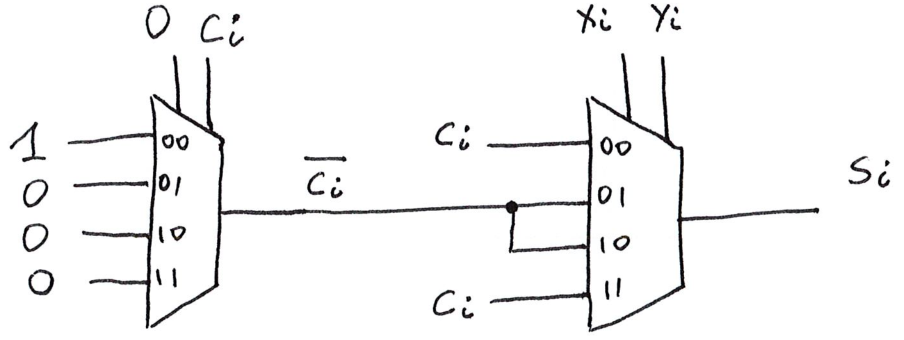
a) Draw the truth table for a full adder with inputs x_i, y_i, and c_i and outputs c_{i+1} and s_i. (5p)

X_i	Y_i	C_i	C_{i+1}	S_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

C_{i+1}	S_i
0	C_i
C_i	$\overline{C_i}$
C_i	$\overline{C_i}$
1	C_i

b) Implement a full adder with a minimal number of 4-to-1 multiplexers and no other logic gates. Assume that the input signals are available only in their non-inverted form, along with the constants 0 and 1. Clearly label all inputs, outputs, and pins of your circuit. (10p)





this implements a NOT gate

8. Alternative Implementation (10p)

Implement the logic expression in each sub-problem in three different ways using the circuits below. In this problem, you are <u>not allowed</u> to use any other logic gates. Assume that a and b are available in both their inverted and non-inverted form, along with the constants 0 and 1. Indicate with words if some implementations are not possible. <u>Label all inputs and outputs.</u>

a) Implement in three different ways: f = a + b .

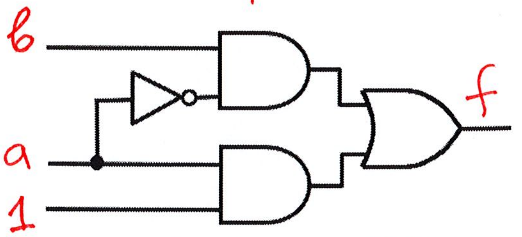
a	b	f	
0	0	0	} b
0	1	1	
1	0	1	} 1
1	1	1	

$\overline{\overline{a} \cdot 1} = a$

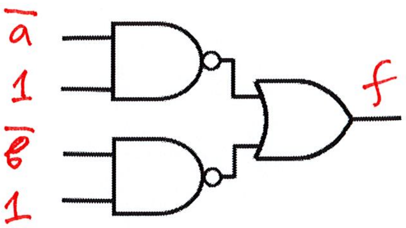
$\overline{\overline{b} \cdot 1} = b$

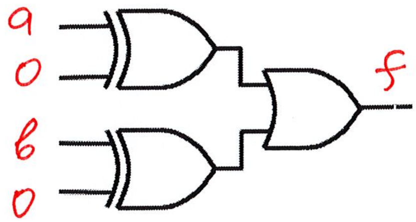
XOR$(a, 0) = a$

XOR$(b, 0) = b$



2-to-1 MUX





b) Implement in three different ways: $f = a + \overline{b}$.

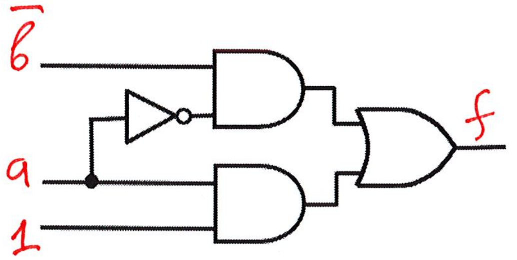
a	b	f	
0	0	1	} $\overline{b}$
0	1	0	
1	0	1	} 1
1	1	1	

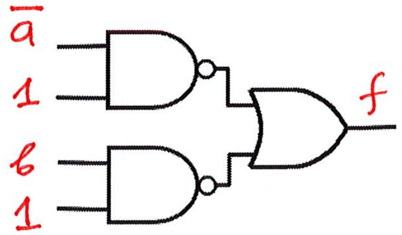
$\overline{\overline{a} \cdot 1} = a$

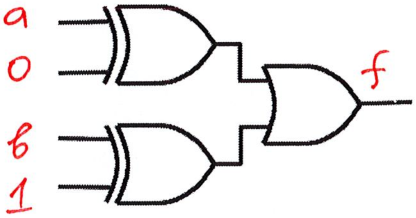
$\overline{b \cdot 1} = \overline{b}$

XOR$(a, 0) = a$

XOR$(b, 1) = \overline{b}$







c) Implement in three different ways: $f = \overline{a\,b}$.

a	b	f	
0	0	1	} 1
0	1	1	
1	0	1	} $\overline{b}$
1	1	0	

$\overline{ab} = \overline{a} + \overline{b}$

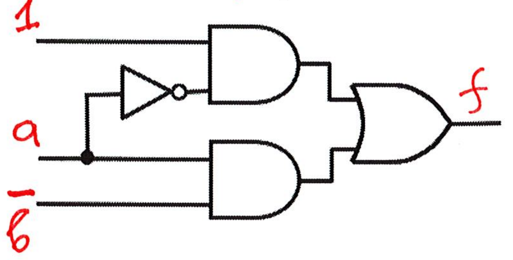
$\overline{a \cdot 1} = \overline{a}$

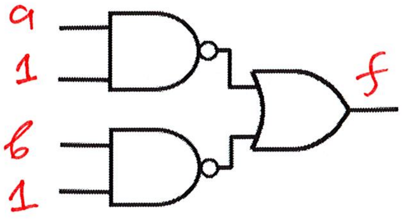
$\overline{b \cdot 1} = \overline{b}$

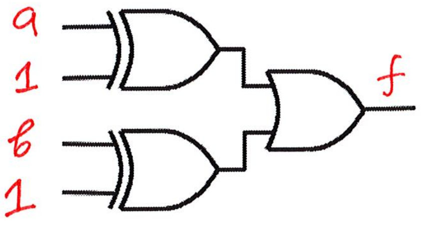
$\overline{ab} = \overline{a} + \overline{b}$

XOR$(a, 1) = \overline{a}$

XOR$(b, 1) = \overline{b}$

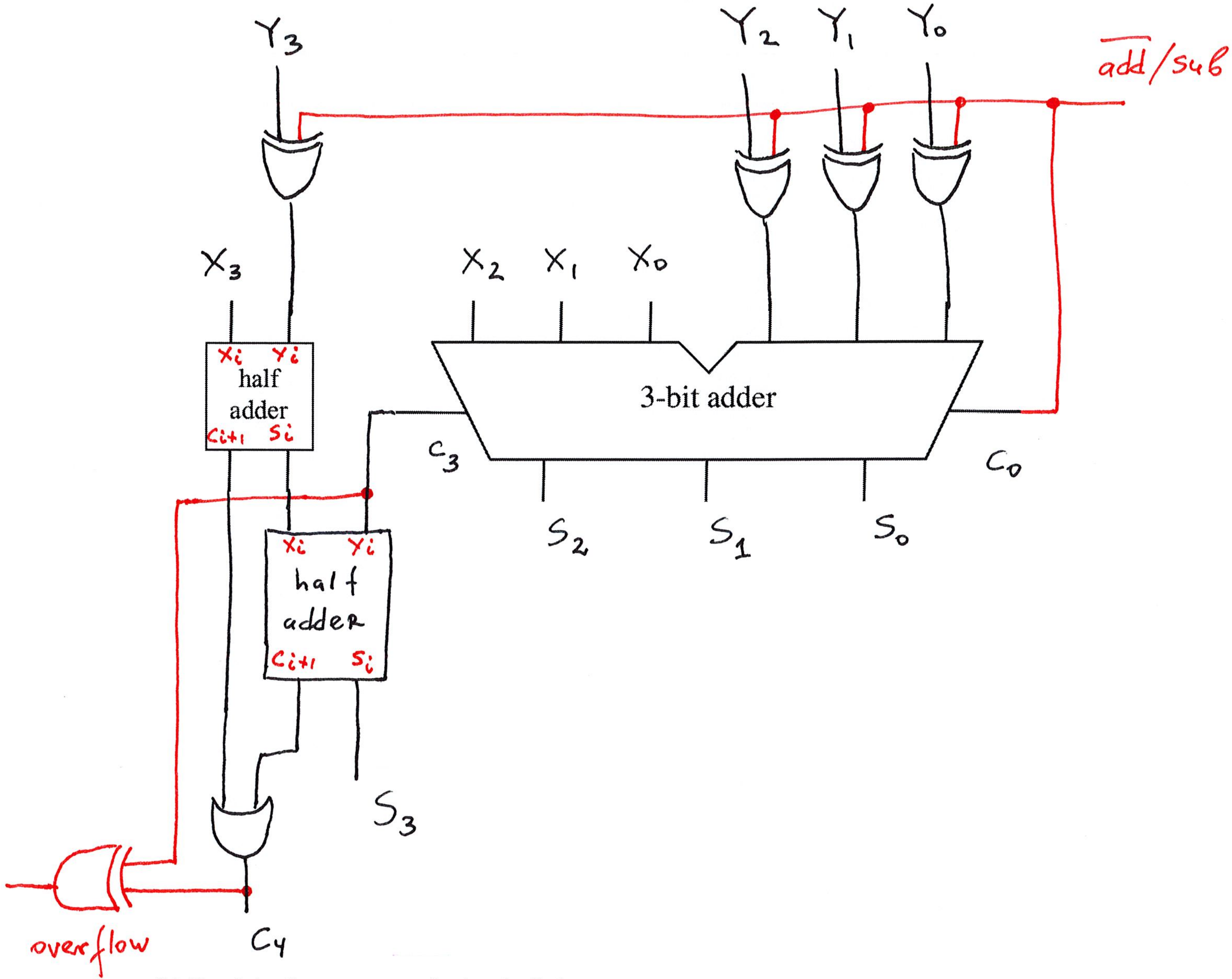






9. Adder/Subtractor with Overflow Detection (10p + 5p = 15p)

(a) You are given a 3-bit adder, but it is too small for what you need, and it also does not compute an overflow flag. Complete the wiring diagram below to implement a functional 4-bit adder/subtractor unit that can also detect when an overflow has occurred. Clearly label all inputs, outputs, and pins of your circuit.



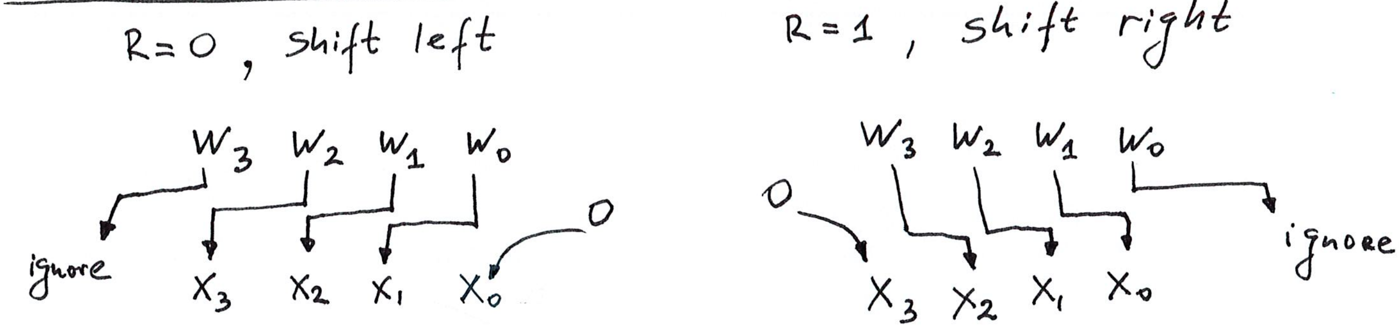
(b) Explain the correct solution in 3-4 sentences.

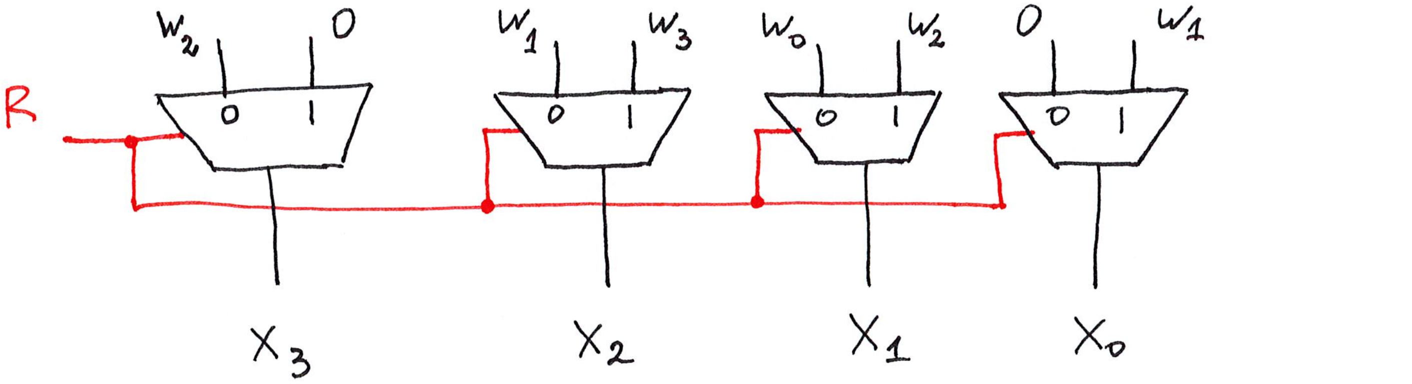
We need another half-adder in order to implement a full adder, plus an OR gate. To get a subtractor unit, we also need additional XOR gates, including one on the new input Y3. The overflow flag is computed as an XOR of C3 and C4.

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

(a) Implement a shifter circuit that takes in a 4-bit unsigned binary number W and shifts all of its bits either to the left or to the right by 1 position. The direction of the shift is determined by one of the inputs to this circuit that is called R. If R=0, then the circuit shifts to the left. If R=1, then the circuit shifts to the right. In both cases the bit that is shifted out is ignored and the empty space that is created is padded with a zero. Label all inputs and outputs.

Let the input be $W = W_3 W_2 W_1 W_0$ and the output be $X = X_3 X_2 X_1 X_0$. The control input is R.





(b) Explain the correct solution in 3-4 sentences.

The shifting is done with four 2-to-1 multiplexors. The input R acts as the select line for all of them. The bits of $W = W_3 W_2 W_1 W_0$ are arranged on the 0 and 1 inputs to map to left/right shift. The padding is done with two zeros.

Question	Max	Score
1. True/False	10	
2. Venn Diagrams	5	
3. Minimization with K-maps	15	
4. Basic Circuits	15	
5. Number Conversions	15	
6. Flip-Flops	15	
7. Full Adder	15	
8. Alternative Implementation	10	
9. Adder/Subtractor	15	
10. Shifter Circuit	15	
TOTAL:	130	