Problem 1 (10 pts): Boolean algebra

1. Which of the following Boolean expressions are equivalent to $(a+\bar{b})(bc+\bar{a}\bar{c})$? Refer to the Boolean algebra properties on the last page of the exam booklet if needed.

a) $(a+\bar{b})\overline{(\bar{b}+\bar{c})(\bar{a}+\bar{c})}$

b) $\overline{(\bar{a}+b)(\bar{b}\bar{c}+ac)}$

c) $\bar{a}b+\overline{bc+\bar{a}\bar{c}}$

✓ d) $\overline{\bar{a}b+(\bar{b}+\bar{c})(a+c)}$

e) $\overline{\bar{a}b}\left(\overline{(\bar{b}+\bar{c})}+\bar{a}\bar{c}\right)$

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

-2

2. Use proof by **perfect induction** to demonstrate that $M_1=\overline{m_1}$ for $f(x,y)$.

$m_1 = \bar{x}y$

$M_1 = \overline{m_1} = x\bar{y}$

x	y	f
0	0	m_0
0	1	m_1
1	0	m_2
1	1	m_3

-5

3. The function $f(w,x,y,z)=M_2+M_{10}$, where M_2 and M_{10} are maxterms. Which expression below is equivalent to $f(w,x,y,z)$?

a) M_{10} b) M_{12} c) $M_2 \cdot M_{10}$ d) $m_2 \cdot m_{10}$ e) 0 f) 1

-1

7

Problem 2 (12 pts): Canonical forms

A 3-variable function $g(x, y, z)$ is defined by the following truth table:

x	y	z	g
0	0	0	**0**
0	0	1	**0**
0	1	0	**1**
0	1	1	**1**
1	0	0	**0**
1	0	1	**0**
1	1	0	**0**
1	1	1	**1**

1. Using the table, write the **canonical SOP** representation for g using **literals**.

   Answer: $g(x,y,z) = \bar{x}y\bar{z} + \bar{x}yz + xyz$

2. Using the table, write the **canonical SOP** representation for g using **minterm** notation m_i.

   Answer: $g(x,y,z) = \Sigma\, m_2 m_3 m_7$

3. Using the table, write the **canonical POS** representation for g using **literals**.

   -2 Answer: $g(x,y,z) = (\bar{x}+\bar{y}+\bar{z})\cdot(\bar{x}+\bar{y}+z)\cdot(x+\bar{y}+\bar{z})\cdot(x+\bar{y}+z)\cdot(x+y+\bar{z})$

4. Using the table, write the **canonical POS** representation for g using **maxterm** notation M_i.

   -1 Answer: $g(x,y,z) = \Sigma\, M_0 M_1 M_4 M_5 M_6$

5. For function `f(a,b,c,d)=a'bd'+a'cd'`, write the corresponding **canonical SOP**.

   $a'b(c+c')d' + a'(b+b')cd'$

   -1 Answer: $f(a,b,c,d) = \bar{a}bc\bar{d} + \bar{a}b\bar{c}\bar{d} + \bar{a}bc\bar{d} + \bar{a}\bar{b}c\bar{d}$

6. For function `g(a,b,c,d)=(c'+d)(a+b'+c'+d)`, write the corresponding **canonical POS**.

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

   -1 Answer: $g(a,b,c,d) = (a+b+\bar{c}+d)(a+\bar{b}+\bar{c}+d)(\bar{a}+b+\bar{c}+d)(\bar{a}+\bar{b}+\bar{c}+d)(a+\bar{b}+\bar{c}+d)$

8

Problem 3 (14 pts): Function simplification

Consider a 4-variable Boolean function `f(w,x,y,z)` given by its K-map (drawn twice):

wx \ yz	00	01	11	10
00	X	1	1	1
01	1	X	0	1
11	0	0	X	1
10	1	1	0	X

wx \ yz	00	01	11	10
00	X	1	1	1
01	1	X	0	1
11	0	0	X	1
10	1	1	0	X

-3 **1.** List all **non-essential prime implicants.**

Answer: $\bar{x}\bar{z}$, wxy

2. Give a **minimal SOP** expression for `f(w,x,y,z)` and **show the corresponding loops** on the left map.

Answer: $f(w,x,y,z) = \bar{w}\bar{y} + \bar{w}\bar{x} + \bar{x}\bar{y} + y\bar{z}$

-2 **3.** Is your solution unique? ~~Yes~~ If no, give another minimal solution.

Answer:

4. Give a **minimal POS** expression for `f(w,x,y,z)` and **show the corresponding loops** on the right map.

Answer: $f(w,x,y,z) = (\bar{w}+\bar{x}+y)(\bar{x}+\bar{z})(\bar{w}+\bar{y}+\bar{z})$

5. Do your answers to **Part 2 and 4** represent the same Boolean function? Justify your answer.

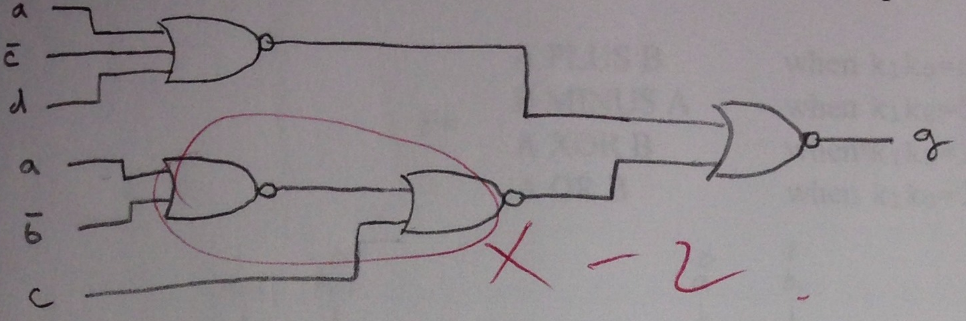
No, they are complements of each other?. The SOP expression is based on minterms whereas the POS expression is based on their complements (where minterm is zero; complemented literals

X -1

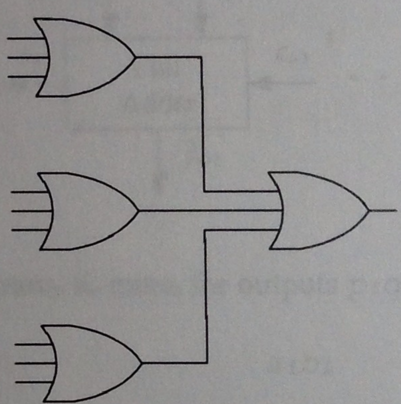
5.

Problem 4 (10 pts): 2-level circuits

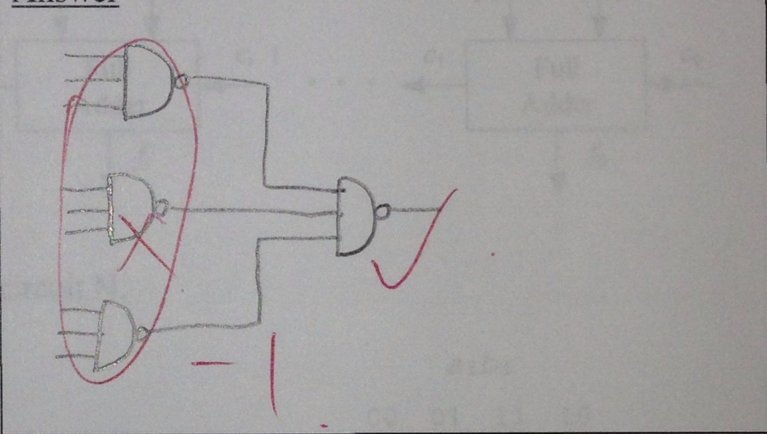
1. Implement the Boolean function `g(a,b,c,d)=(a+c'+d)(a+b')c` as a two-level network using **NOR** gates **only**. Assume that inverted inputs are available. Draw the circuit.



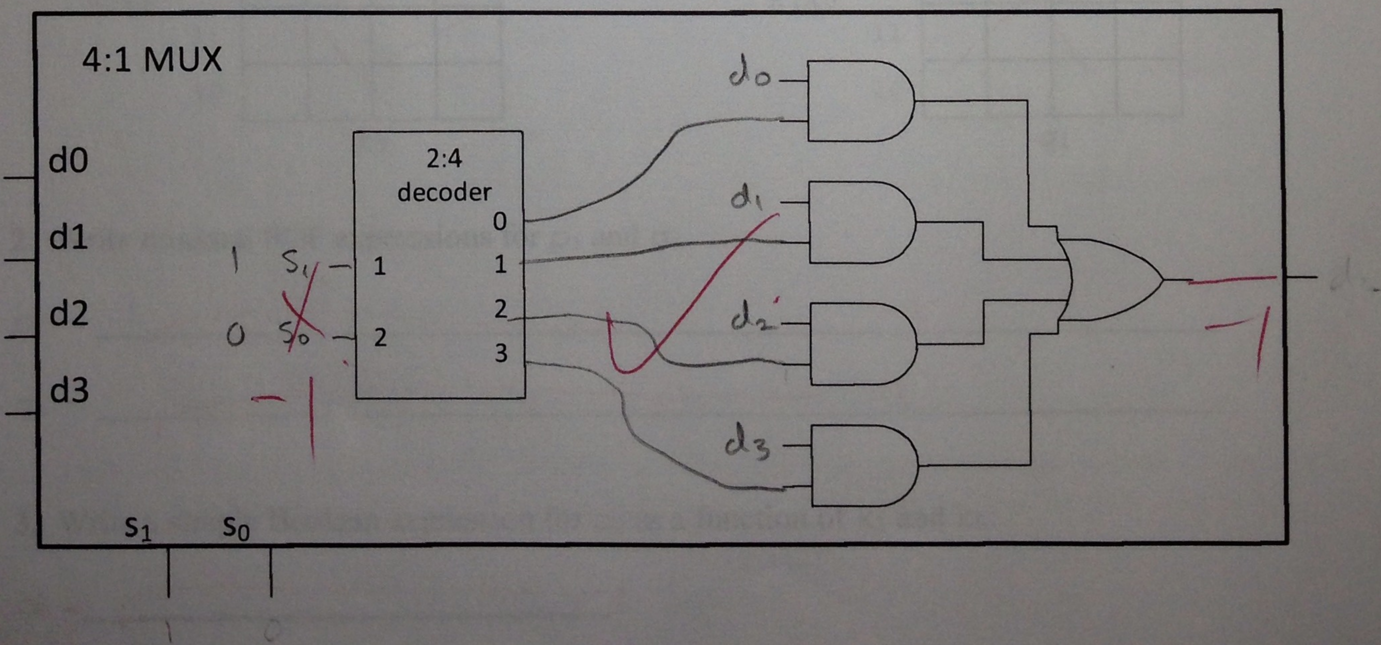
2. Shown below is an implementation of a 9-input OR function. Re-implement it using **NOR** and **NAND** gates only. (*Hint:* Use alternative representation for NAND gate.)



Answer



3. Implement a 4:1 MUX using a 2:4 decoder and a 2-level AND-to-OR circuit.



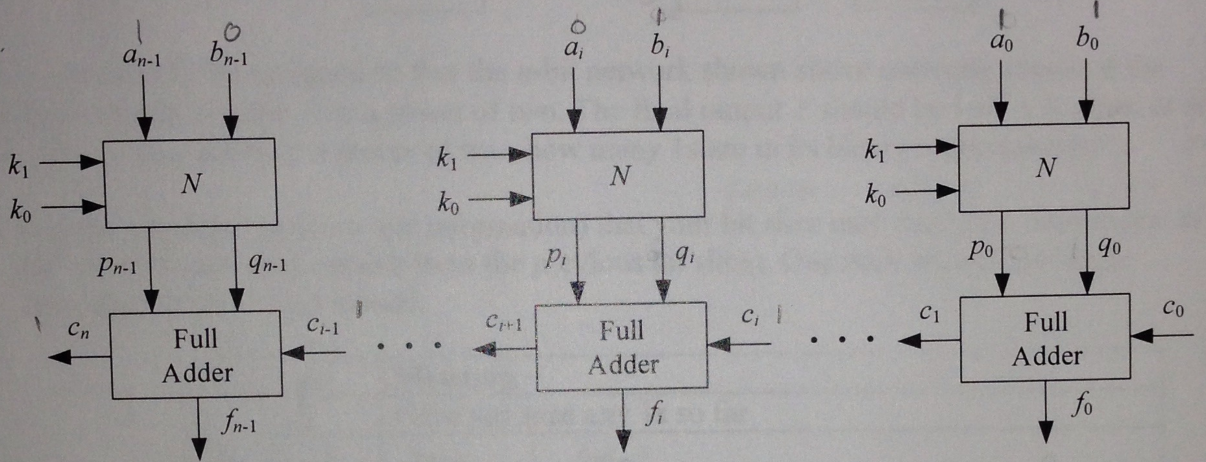
$\frac{0}{11}$

Problem 5 (11 pts): Combinational logic design

Design network N such that the n-bit arithmetic unit shown below can perform the following arithmetic **and logic** operations for inputs $A=a_{n-1}$ ... a_0 and $B=b_{n-1}$ ... b_0:

$$F=\begin{cases}\text{A PLUS B} & \text{when } k_1k_0=00\\ \text{B MINUS A} & \text{when } k_1k_0=01\\ \text{A XOR B} & \text{when } k_1k_0=10\\ \text{A OR B} & \text{when } k_1k_0=11\end{cases}$$

a_i^*	b_i^*	C_0
a_i	b_i	0
$\overline{a_i}$	b_i	1
a_i	b_i	0
a_i	b_i	0



1. Draws K-maps for outputs p_iq_i from circuit N.

p_i (k_1k_0 rows, a_ib_i columns):

k_1k_0 \ a_ib_i	00	01	11	10
00	0	0	1	0
01	0	1	0	0
11				
10				

q_i (k_1k_0 rows, a_ib_i columns):

k_1k_0 \ a_ib_i	00	01	11	10
00				
01				
11				
10				

2. Write minimal SOP expressions for p_i and q_i.

p_i = ________________ −1

q_i = ________________ −1

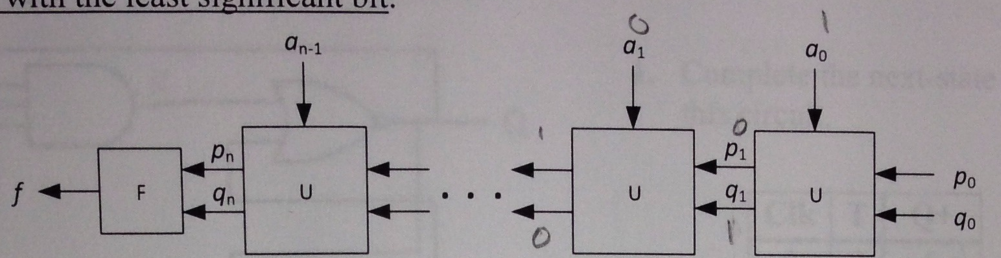
3. Write a simple Boolean expression for c_0 as a function of k_1 and k_0.

c_0 = ________________ −1

Problem 6 (11 pts): Combinational logic design

10/11

Design a *bit-slice circuit* that checks whether an unsigned integer $A=a_{n-1}a_{n-2}..a_1a_0$ is a power of 2, starting with the least significant bit.



The U cell should be designed so that the *n*-bit network shown above correctly checks if the unsigned integer number A is a power of two. The final output f should be 1 iff A is a power of two. (*Hint*: If a number is a power of two, how many 1s are in its binary representation?) 1 one

10000 0010 0100 1000

1. List the possible 'answers' (or information) that your bit slice may need to communicate to the next bit slice (and receive from the previous bit slice). One such answer is already provided for you to get started.

k1 k0

p_i	q_i	Meaning
0	0	Have not seen any 1s so far
0	1	one 1 seen
1	0	Have not seen another 1, but saw 1 previously
1	1	two ones seen (not a Power of 2)

2. What are initial values for p_0 and q_0? p_0 = 0, q_0 = 0

3. Draw K-maps for p_{i+1} and q_{i+1}.

p_{i+1}:

a_i \ p_iq_i	00	01	11	10
0	0	1	1	1
1	0	1	X	1

q_{i+1}:

a_i \ p_iq_i	00	01	11	10
0	0	0	1	0
1	1	1	X	1

x -1

4. Give minimal POS expressions for p_{i+1} and q_{i+1}.

p_{i+1} = $Q_i + P_i$ q_{i+1} = $a_i + P_iQ_i$

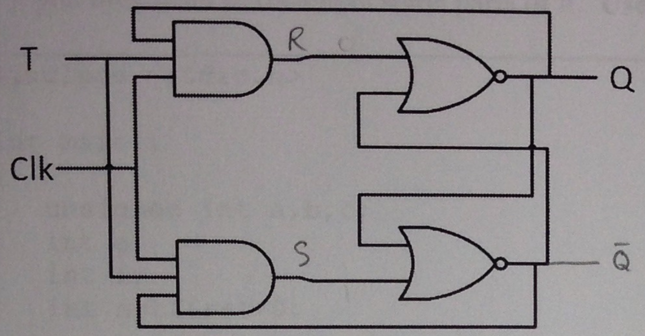
5. Write a Boolean expression for the decision circuit F that maps from the outputs of the most-significant-bit bit-slice p_nq_n to the answer $f=1$ *if A is a power of two, or* $f=0$ *otherwise.*

$f(p_n, q_n)$ = $P_n \oplus Q_n$

7

Problem 7 (16 pts): Sequential logic components

Part A. (10 pts) Shown below is the logic diagram of a gated T latch. It consists of 2 NOR gates and two AND gates and has 2 inputs: T and Clk.



1. Complete the next-state table for this circuit.

$R = QTClk$
$S = \bar{Q}TClk$

Clk	T	Q+
0	0	1 (set) X-1
0	1	Q ✓
1	0	Q ✓
1	1	0 (reset) X-3

S	R	NOR
0	0	1
0	1	0
1	0	0
1	1	0

2. Express next state Q+ as a function of Clk, T, and Q.

$Q+ = \overline{(\bar{Q}+R)} + \overline{(Q+S)}$

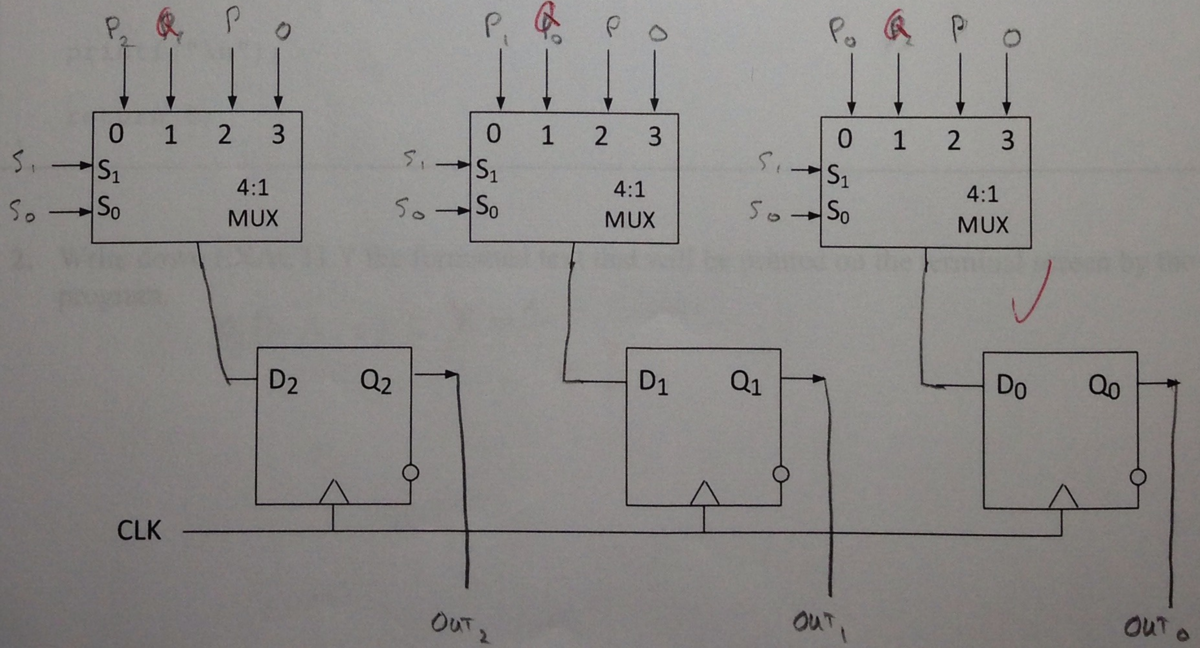
Answer: $Q+ = \overline{(\bar{Q}+(QTClk))} + \overline{(Q+(\bar{Q}TClk))}$ X-3

3. Describe the behavior of the T latch using common everyday English.

Answer: Ability to store a bit

Part B. (6 pts) Complete the design of a 3-bit register that performs the operations listed in the table to the right. Parallel load inputs are labeled and indexed as P_i. Serial input is labeled as S_{in}. **You may use inputs without drawing the wires, just write the appropriate labels at the MUX inputs.**

S_1	S_0	Operation
0	0	Parallel load ✓
0	1	Circular shift left X-1
1	0	Hold current value X-1
1	1	Clear ✓



Problem 8 (16 pts): Boolean algebra in C

1. Implement a program in C that prints canonical SOP representation for function `g(a,b,c)=a' (b⊕c)+ac` using minterm notation. The program is partially implemented, you only need to complete some parts of it. **Use bit-wise operators only.**

```
#include <stdio.h>

int main()
{
    unsigned int a,b,c;
    int g;
    int i;
    int notfirst=0;

    printf("g(a,b,c)=");
    for ( a = 0; a <= 1; __a = a + 1__ )
    {
        for ( b = 0; __b <= 1__ -1 ; b = b + 1 )
        {
            for ( __c = 0__ ; c <= 1; c = c + 1 )
            {
                g = __(~a &(b^c)|(a&c))__ ; /* function g */
                if (g == __1__ )
                {
                    if (notfirst) printf("+");
                    i = __a+b+c__ -3 ; /* minterm index */
                    printf("m%d", i);
                    notfirst = 1;
                }
            }
        }
    }

    printf("\n");

    return 0;
}
```

2. Write down EXACTLY the formatted text that will be printed on the terminal screen by the program.

g(a,b,c) = ~~+~~m1

-3