**Problem 1** (10 points)

Reduce the following expression using Boolean algebra postulates and theorems. The simplified expression should have the minimum number of gates. Show the intermediate steps.

$$f(a,b,c,d) = \overline{\overline{(acd)}(\bar a+\bar b+\bar d)(\overline{ad}+c)} + \overline{a\bar b}(\bar a+\bar b c+\bar b\bar c)$$

$$= \left(((acd)')' + (a'+b'+d')' + ((ad)'+c)' + (a'+(b')')\right)(a'+b'c+b'c')$$

$$= acd + abd + (ad)''c' + (a'+b)(a'+b'c+b'c')$$

$$= acd + abd + ac'd + (a'+a'b'c+a'b'c'+a'b+\overset{0}{\cancel{bb'c}}+\overset{0}{\cancel{bb'c'}})$$

$$= ad\underbrace{(c+c')}_{1} + abd + a'\underbrace{(1+b)}_{1} + a'b'\underbrace{(c+c')}_{1}$$

$$= ad + abd + a' + a'b'$$

$$= ad\underbrace{(1+b)}_{1} + a'\underbrace{(1+b')}_{1}$$

$$= ad + a'$$

6

**Problem 2** (15 points)

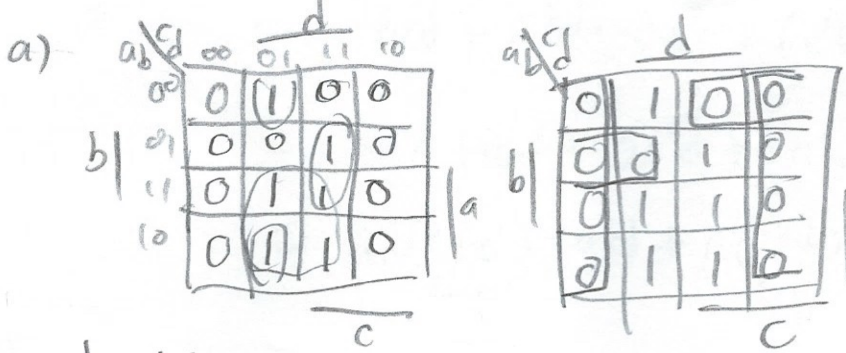
Consider the following function

$$f(a,b,c,d) = \Sigma m(1,7,9,11,13,15).$$

(a) (8 points) Use K-maps to minimize **both** the sum of products and products of sums forms. Write the Boolean expressions.

(b) (7 points) Implement the function using the minimal number of gates. You can use either NOR gates or NAND gates with maximum number of four inputs per gate.

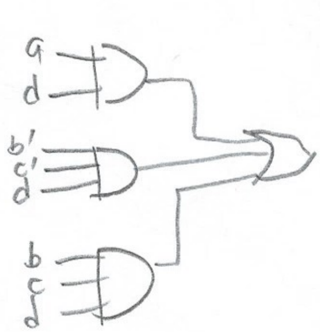
a)

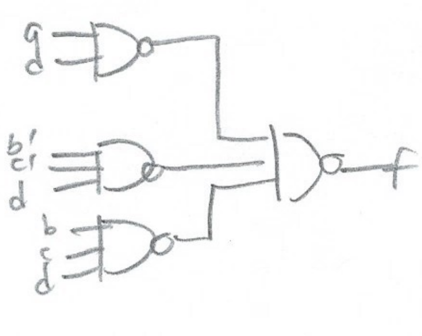



$ad + b'c'd + bcd$

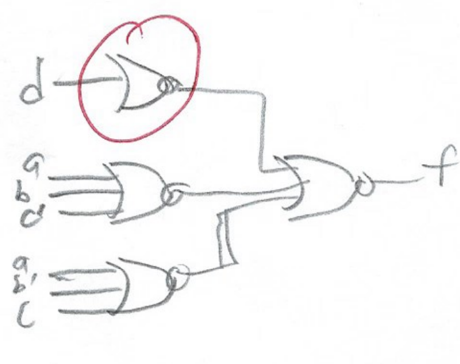
$(d)(a+b+c')(a+b'+c)$

8




or




6

19.5

**Problem 3** (20 points)
Consider the following function shown below.

(a) (3 points) Write the switching expression for the multiplexer circuit below. The multiplexers have binary select inputs. The expression need not be simplified.

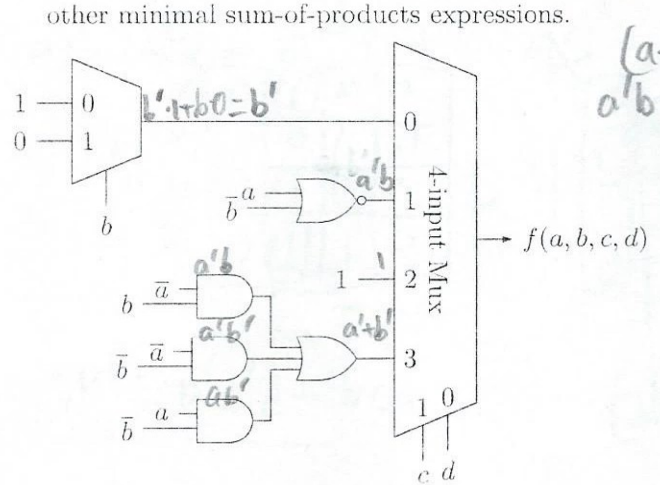
(b) (3 points) Determine the prime implicants and the essential prime implicants for this expression.

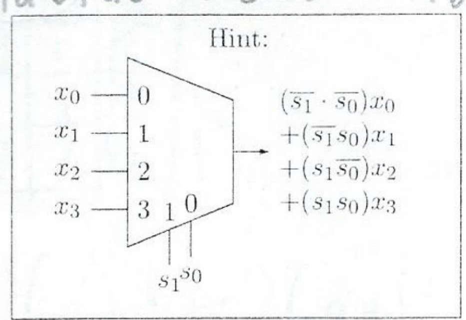
(c) (8 points) Find the minimal sum-of-product for the switching expression $f$.

(d) (6 points) Does this function have a unique minimal sum-of-products? If not, list any other minimal sum-of-products expressions.

$(a+b')' = a'b$

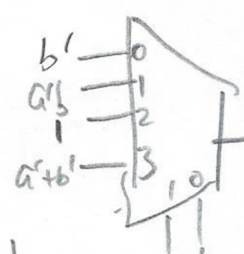
$a'b + a'b' + ab' = a'b + a'b' + ab' + a'b' = a' + b'$




a)

→




$f(a,b,c,d) = c'd'(b') + c'd(a'b) + cd'(1) + cd(a'+b')$

0 or 8 5

$= b'c'd' + a'bc'd + cd' + a'cd + b'cd$

b)


EPI: $a'bd$, $cd'$, $b'c$, $b'd'$

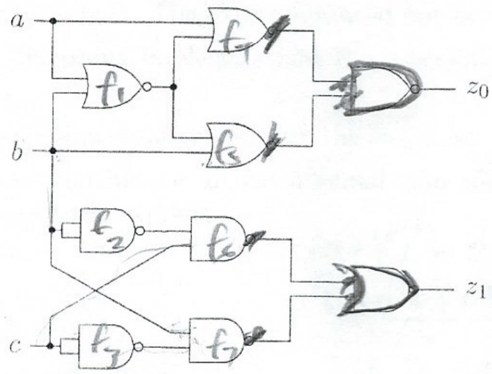
PI: $a'bd$, $cd'$, $b'c$, $b'd'$, $\bar{a}c$, -0.5

c) $f = a'bd + cd' + b'c + b'd'$

d) Yes, since all PIs are EPIs

10

**Problem 4** (10 points)
Analyze the NAND-NOR network shown in the figure below. Obtain switching expressions for the outputs $z_0$ and $z_1$.




$f_1 = (a+b)' = a'b'$

$f_2 = (bb)' = b'$

$f_3 = (cc)' = c'$

$f_4 = a + f_1 = a + a'b'$

$f_5 = b + f_1 = b + a'b'$

$f_6 = f_2 c = b'c$

$f_7 = b f_3 = bc'$

$z_0 = f_4 f_5 = (a + a'b')(b + a'b')$

$= ab + 0 + 0 + a'b'$

$= ab + a'b'$ (XNOR)

$= (a \oplus b)'$

$z_1 = f_6 + f_7 = bc' + b'c$ (XOR)

$(b \oplus c)$

11

**Problem 5** (15 points)

Design a combinational circuit that converts a 3-bit sign-and-magnitude number, $a$, into a 3-bit one's complement number, $b$. You are allowed to use any combination of the following blocks:

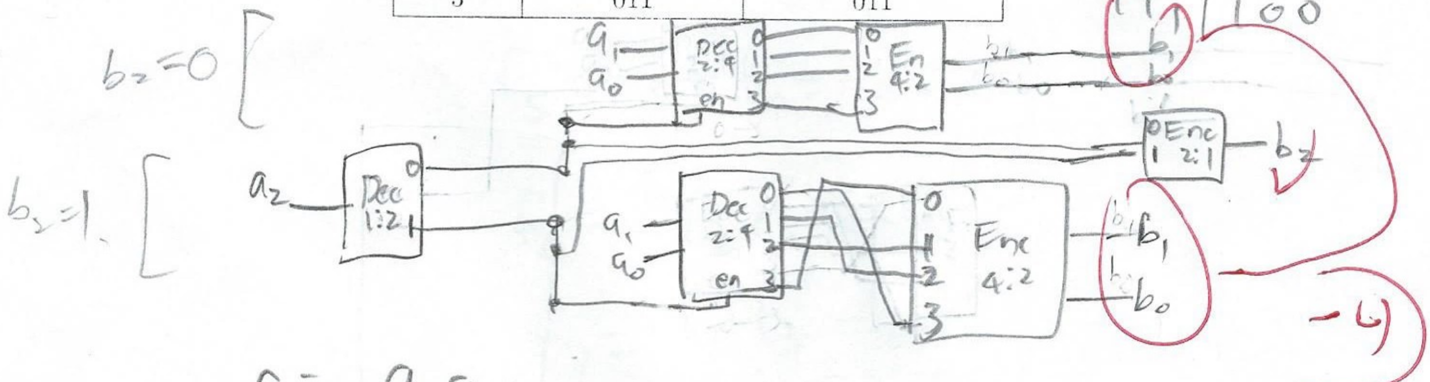
- Decoders: $1 \to 2$ and $2 \to 4$ decoders
- Encoders: $2 \to 1$ and $4 \to 2$ encoders
- Logic gates: Use either OR gates or AND gates, **but not both**

2->1 enc

Every block or wire must be clearly labeled.

| Decimal | Sign-Magnitude ($a_2a_1a_0$) | One's Complement ($b_2b_1b_0$) |
|---|---|---|
| -3 | 111 | 100 |
| -2 | 110 | 101 |
| -1 | 101 | 110 |
| -0 | 100 | 111 |
| 0 | 000 | 000 |
| 1 | 001 | 001 |
| 2 | 010 | 010 |
| 3 | 011 | 011 |

| $a_2\ a_1\ a_0$ | $b_2\ b_1\ b_0$ |
|---|---|
| 0 0 0 | 0 0 0 |
| 0 0 1 | 0 0 1 |
| 0 1 0 | 0 1 0 |
| 0 1 1 | 0 1 1 |
| 1 0 0 | 1 1 1 |
| 1 0 1 | 1 1 0 |
| 1 1 0 | 1 0 1 |
| 1 1 1 | 1 0 0 |




$a = a_2a_1a_0$

$b = b_2b_1b_0$

0000 0 0 11
1111 1100 +1
11111101



**Problem 6** (15 points)

15

Compute $z = (a - b) + (c - d)$ in 2's complement.

(a) (4 points) Fill up the table given below.

(b) (2 points) How many bits should $z$ have to represent the correct result?

(c) (9 points) Perform calculations on bit-vectors representing $a$, $b$, $c$ and $d$ (all in 2's complement) and show every step of your work. $z$ should be given in **both** decimal representation and 2's complement. **If you want**, you can extend the table below.

| | Decimal | Sign-Magnitude | 2's Complement |
|---|---|---|---|
| $a$ | -13 | 11101 | 10011 |
| $b$ | -10 | 1 1010 | 10110 |
| $c$ | -6 | 1110 | 1010 |
| $d$ | 82 | 0 1010010 | 01010010 |

82/2 = 41 r 0
41/2 = 20 r 1
20/2 = 10 r 0
10/2 = 5 r 0
5/2 = 2 r 1
2/2 = 1 r 0
1/2 = 0 r 1

16 − 10 = 6

13/2 = 6 r 1
6/2 = 3 r 0
3/2 = 1 r 1
1/2 = 0 r 1

$1\times2^3 + 0 + 1\times2^1 + 0 = 8+2 = 10$

~~16 − 10 = (6)₁₀ = (110)₂~~

32 − 10 = (22)₁₀ = 1

10 → 01010 → 10101 → 1.0110

16 − |−13| = 3 = (0011)₂

b) 8 bits

c)

| | Decimal | 2's Comp. |
|---|---|---|
| a | −13 | 11110011 |
| b | −10 | 11110110 |
| −b | 10 | 00001010 |
| a−b | −3 | 11111101 |
| c | −6 | 11111010 |
| d | 82 | 01010010 |
| −d | −82 | 10101110 |
| c−d | −88 | 10101000 |
| (a−b)+(c−d) | −91 | 10100101 |

Use range extension

−10 → 10

00001001+1 ⇒ 00001010

(a−b):
  11110011
+00001010
  11111101

−82 → 82

(c−d): 10101101+1 ⇒ 10101110
  11111010
+10101110
1 10101000

3

(a−b)+(c−d):
  11111101
+10101000
1 10100101

128 − 91 = 37

1

**Problem 7** (15 points)

Design a combinational network that has 4-bit inputs $a$ and $b$, and a four-bit output $z$. All inputs and outputs are given in two's complement representation. The function of the system is

$$z = \max\{a + b, 0\} \mod 8$$

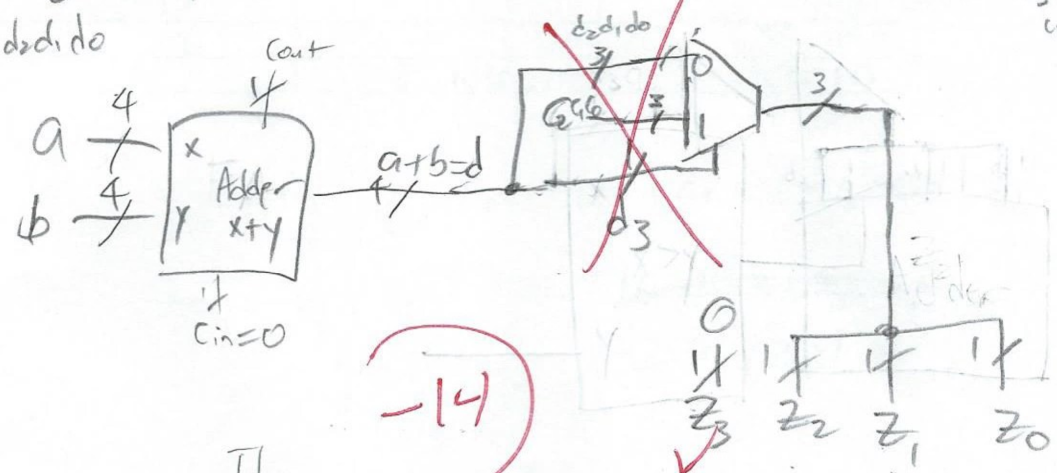
For example, if $a = 3$ and $b = 7$, then $z = 2$. If $a = 3$ and $b = -7$, then $z = 0$.

You are allowed to use only ~~magnitude comparators,~~ adders, and/or multiplexers. Every block and wire must be clearly labeled.

$a = \overbrace{a_3}^{\text{sgn}} a_2 a_1 a_0$ $b = \overbrace{b_3}^{\text{sign}} b_2 b_1 b_0$

Let $c = c_3 c_2 c_1 c_0$ where $c_3 = 0$, $c_2 = 0$, $c_1 = 0$, $c_0 = 0$

Let $a+b = d = d_3 d_2 d_1 d_0$




-14

This is assuming no overflow

but there could be overflow!

If $a, b > 0$