CS 51 A Introduction to Digital Systems Design
UCLA Computer Science Department
Midterm Exam, July 23rd, 2018
Prof. Leon Alkalai
90 min closed-book exam
Show all work

Student Name [redacted]

Student ID: [redacted]

Problem	
1 (10 pts)	8
2 (30 pts)	30
3 (30 pts)	30
4 (30 pts)	22
Total	90

Absorbtion

5

Problem 1 (10 points)

Simplify the following Boolean expression by using postulates of Boolean algebra. For each step, indicate which Boolean algebra rule you are using.

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

$= (a'b'+c)(a+b)(b \cdot (a'c')')$ DeMorgan's Rule

$= (a'b'+c)(a+b)(b \cdot (a+c))$ DeMorgan's Rule

$= (a'b'+c)(a+b)(ba+bc)$ Distributive Prop

$= (a'b'+c)(aba+abc+bba+bbc)$ Distrib. Prop

$= (a'b'+c)(ab+abc+ba+bc)$ Self Identity $(a \cdot a = a)$

$= (a'b'+c)(ab+abc+bc)$ Self Identity $(a+a=a)$

$= (a'b'+c)(ab+bc)$ Absorbtion $(a+ab=a)$

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

$= (ba'b'+bc)(a+c)$ Distribution

$= (0+bc)(a+c)$ Identity $(b \cdot b' = 0)$

$= abc + ccb$ Distribution

$= abc + cb$ Identity $(c \cdot c = c)$, communtitive → $bc = cb$

$= \not{a}$ Absorbtion $(a+ab=a)$

−2

$(a+1)bc$

$= bc$

Problem 2 (30 points)

Show that the functions a) E (x,y) and b) E(x,y,z) shown in the truth tables below are universal functions. You can use constants 0 or 1 as your input to each function.

a) **(5 points)**

	x	y	E(x, y)
1	0	0	1
1	0	1	1
OR → 0	1	0	0
0	1	1	1

to show universal must be able to form another universal set.
function looks like not inputs, then OR gate.

$E(x,y) = (x' + y)$

NOT ⟶ $E(x,0) = (x' + 0) = x'$ ✓ "NOT GATE"

OR ⟶ simply use not gate derived above to complement x input, then use E(x,y)

$E(E(x,0), y))$ → "OR" function.

since {OR, NOT} is a universal set, so is E.

b) **(5 points)**

	x	y	z	E(x,y,z)
	0	0	0	0
	0	0	1	0
	0	1	0	0
{	0	1	1	1
	1	0	0	1
	1	0	1	0
{	1	1	0	1
	1	1	1	1

x \ yz	00	01	11	10
0	0	0	1	0
1	1	0	1	1

$(x \cdot z' + yz) = E(x,y,z)$

$E(1,0,x) = (1 \cdot x' + 0 \cdot x) = (x')$ → "NOT" Gate possible

$E(0,y,z) = (0 \cdot z' + y \cdot z) = yz$ → "AND" GATE possible

to create 3 input "and" gate simply...

$E(0, E(0,y,z), x)$

since {AND, NOT} universal, E is also universal.

$(xy')'$

Do

c) 20 points

Using only the first function E(x,y) as a gate, implement the second function E(x,y,z) using a **minimal** 2-level gate network. Clearly show all steps in the process.

x \ yz	00	01	11	10
0	0	0	1	0
1	1	0	1	1

prime implicants

0s	1s
$(x + z)$	xz'
$(y + z')$	yz

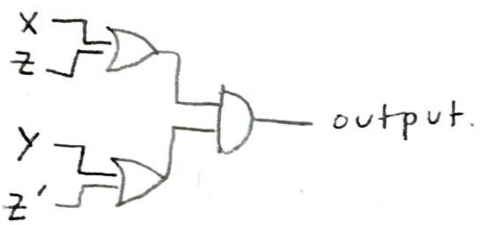
$E(x,y,z) = (x+z)\cdot(y+z')$

↳ 3 gates.
6 inputs

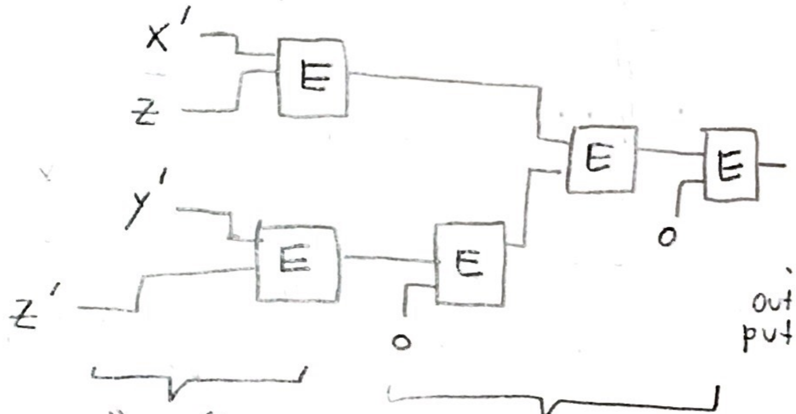
$E(x,y,z) = xz' + yz$

↳ 3 gates
6 inputs

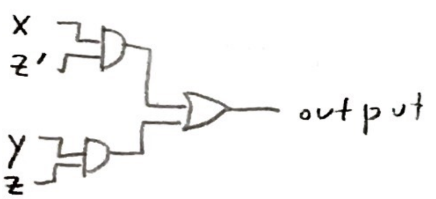
same cost!



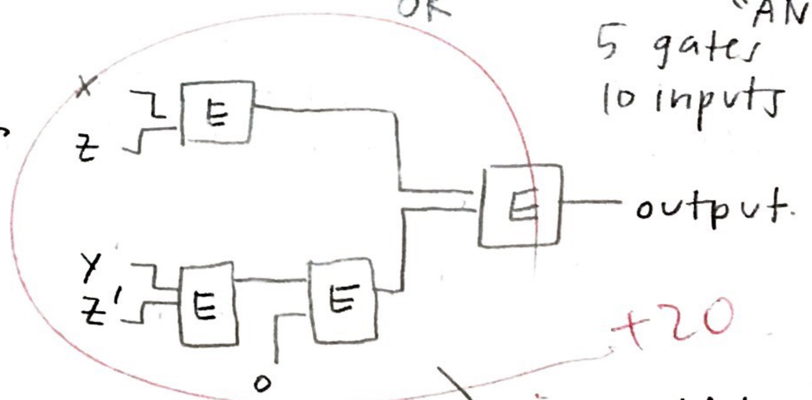
OR/AND
product of sums



5 gates
10 inputs



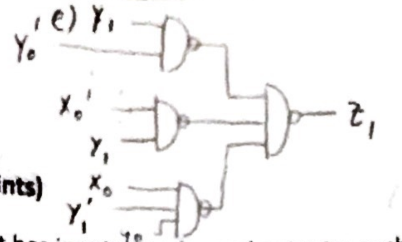
AND/OR.
sum of products.

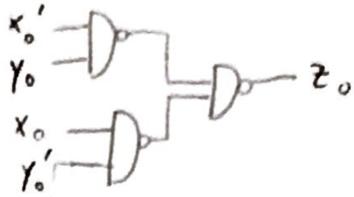


4 gates
8 inputs

+20

→ MINIMAL NETWORK

(e) 



Problem 3 (30 points)

F is a function that has inputs x and y, and output z, such that $x \in \{0, 1, 2\}$, $y \in \{0, 1, 2, 3\}$, and outputs $z = (x^2 + y)\ mod8$. Use the binary code to encode x, y, and z, that is, x as x_1x_0, y is as y_1y_0, and z as $z_2z_1z_0$.

(a) **(6 points)** Show the truth table for function z.

(b) **(6 points)** Based on the truth table in (a), draw the K-map for z_2, z_1, and z_0.

(c) **(6 points)** Use the K-maps in (b) to find all the prime implicants for z_2, z_1, and z_0 respectively.

(d) **(6 points)** Use the K-maps in (b) to find all the essential prime implicants for z_2, z_1, and z_0 respectively.

(e) **(6 points)** Implement z_2, z_1, and z_0 using minimal NAND-NAND networks. Note that each output has a separate gate network. You can directly use x_1, x_0, y_1, y_0 as inputs.

a)

		0	1	2	3 ← Y
X	X_1X_0	00	01	10	11 ← Y_1Y_0
0	0 0	000	001	010	011
1	0 1	001	010	011	100
2	1 0	100	101	110	111
3	11	—	—	—	—

→ $z_2 z_1 z_0$

↳ "don't care"

X \ Y	0	1	2	3
0	0	1	2	3
1	1	2	3	4
2	4	5	6	7
3	–	–	–	–

z

b.

X_1X_0 \ Y_1Y_0	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	—	—	—	—
10	1	1	1	1

Z_2

X_1X_0 \ Y_1Y_0	00	01	11	10
00	0	0	1	1
01	0	1	0	1
11	—	—	—	—
10	0	0	1	1

Z_1

X_1X_0 \ Y_1Y_0	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	—	—	—	—
10	0	1	1	0

Z_0

c. • X_1 • $X_0Y_1Y_0$ } Z_2

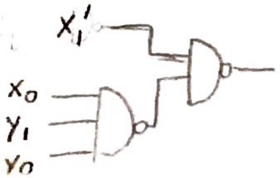
• Y_1Y_0' • $X_0Y_1'Y_0$ • X_1Y_1 • $X_0'Y_1$ } Z_1

• X_1Y_0 • $X_0'Y_0$ • X_0Y_0' } Z_0

d. • X_1 • $X_0Y_1Y_0$ } Z_2

• Y_1Y_0' • $X_0Y_1'Y_0$ • $X_0'Y_1$ } Z_1

• $X_0'Y_0$ • X_0Y_0' } Z_0

e. 

, see top of page for others.

Problem 4 (30 points)

F is a function that inputs a single binary coded decimal digit (BCD) x and outputs the value z which is equal to (x+1) mod4 if x<4, and (x-1) mod4 of x>3.

a) Show the mathematical representation of the high level function z.
b) Show the binary representation of x, ~~X~~, and z and the truth table for z.
c) Using K-maps, derive the minimal 2-level gate network, and clearly calculate the cost of each And/Or and Or/And gate networks, as discussed in class.
d) Graphically depict the 2 gate networks and show the lowest cost implementation.

a. $z = \begin{cases} (x+1) \bmod 4 & \text{if } x \leq 3 \\ (x-1) \bmod 4 & \text{if } x > 3 \end{cases}$ ✓

b.

x	$x_3 x_2 x_1 x_0$	z_1 z_0	z
0	0000	0 1	1
1	0001	1 0	2
2	0010	1 1	3
3	0011	0 0	0
4	0100	1 1	3
5	0101	0 0	0
6	0110	0 1	1
7	0111	1 0	2
8	1000	1 1	3
9	1001	0 0	0

c.

z_1:

x_3x_2 \ x_1x_0	00	01	11	10
00	0 (0)	1 (1)	0 (3)	1 (2)
01	1 (4)	0 (5)	1 (7)	0 (6)
11	— (12)	— (13)	— (15)	— (14)
10	1 (8)	0 (9)	~~0~~ (11)	~~0~~ (10)

z_0:

x_3x_2 \ x_1x_0	00	01	11	10
00	1 (0)	0 (1)	0 (3)	1 (2)
01	1 (4)	0 (5)	0 (7)	1 (6)
11	— (12)	— (13)	— (15)	— (14)
10	1 (8)	0 (9)	~~0~~ (11)	~~0~~ (10)

dc −2

$z_1 = (x_3 x_1' x_0' + x_2 x_1' x_0' + x_3' x_2' x_1' x_0 + x_3' x_2' x_1 x_0' + x_2 x_1 x_0)$

1s ↓

6 gates
22 inputs
AND/OR — minimum!

$z_1 = ((x_2 + x_1' + x_0') \cdot (x_3 + x_2 + x_1 + x_0) \cdot (x_2' + x_1 + x_0') \cdot (x_2' + x_1' + x_0) \cdot (x_3' + x_1') \cdot (x_3' + x_0'))$

0s ↓

7 gates
23 inputs.
OR/AND

−2

z_0: 1s: $x_3' x_0'$, $x_1' x_0'$ | 0s: x_0', $(x_3' + x_1')$

1s −2 0s

$z_0 = x_3' x_0' + x_1' x_0'$
3 gates
6 inputs
AND/OR

$z_0 = x_0' + (x_3' + x_1')$
2 gates
4 inputs.
OR/AND — minimum!

d. −2