GIT Department of Computer Engineering
CSE 232 - Spring 2020
Homework 2

BARAN HASAN BOZDUMAN
171044036

1. Assume for a particular year that a particular size chip using state-of-the-art technology can contain 1 billion transistors. Assuming Mooreâs Law (doubling each 18 months) holds, how many transistors will the same size chip be able to contain in ten years?

For each 18 month it increases in multiplies of two so there will be
10 years=> 120 months 120/18=6.66666666667 times power of two
2^6.66666666667 =**101.593667326** in ten years

2. Evaluate the Boolean equation F = (a AND b) OR c OR d for the given values of variables a, b, c, and d:

a. a=1, b=1, c=1, d=0
(1 AND 1) OR 1 OR 0
1 OR 1 OR 0
1 OR 0 =1
b. a=0, b=1, c=1, d=0
(0 AND 1) OR 1 OR 0
0 OR 1 OR 0
1 OR 0 =1
c. a=1, b=1, c=0, d=0
(1 AND 1) OR 0 OR 0
1 OR 0 OR 0
1 OR 0 =1
d. a=1, b=0, c=1, d=1
(1 AND 0) OR 1 OR 1
0 OR 1 OR 1
1 OR 1= 1

3. For the function F = a + a’ b + acd + c’:
a. List all the variables.
{ a, b, c, d }
b. List all the literals.
{ a, a’, b, c, c’,d }
c. List all the product terms.
{ a, a’ b, acd, c’ }

4.
Convert the function F shown in the truth table in the table to an equation. Do not minimize the equation

a	b	c	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

F =a’b’c + a’bc’ + a’bc + ab’c + abc’ + abc

5. Use algebraic manipulation to minimize the equation obtained in Exercise 4.

F = a’b’c + a’bc’ + a’bc + ab’c + abc’ + abc
F = a’(b’c + bc’ + bc) + a(b’c + bc’ + bc)-→distributive rule
F = (a’ + a)(b’c + bc’ + bc)--------------------→ distributive rule
F = (a’ + a)(bc’ + c(b+ b’))----------------- → distributive rule
F = (1) (bc’ + c(1)) --------------------→ complement rule
F = bc’ + c--→ identitiy rule

above minimalized equation it ensures you for the for the all c = 1 equation just for one 7. equation on table except this rule but when we put the values in our minimalized equation it also provides the result and also it provides equations which results are false too.

6. Determine whether the Boolean functions F = (a + b)’ * a and G = a + b’ are equivalent, using:
(a) algebraic manipulation

F = (a + b)’ * a
F = (a’ * b’)* a ----------→ demorgan’s law
F = a* (a’ * b’)------------->commutative
F = (a * a’) * b’----------→associative
F = (0) * b’----------------→complement
F = 0-------------------------->null elements

G = a+ b’

F and G are not equal

(b) truth tables.
F function

inputs				outputs
a	b	a+b	(a+b)’	(a+b)’ * a
0	0	0	1	0
0	1	1	0	0
1	0	1	0	0
1	1	1	0	0

G function

a	b	b’	a + b’
0	0	1	1
0	1	0	0
1	0	1	1
1	1	0	1

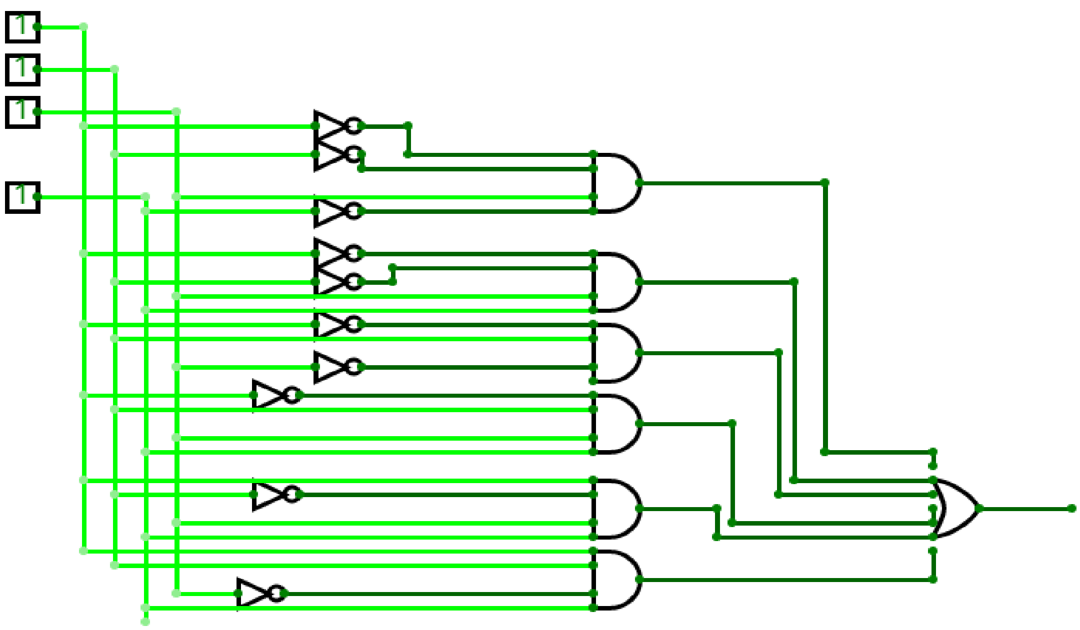
7. Using the combinational design process, create a 4-bit prime number detector. The circuit has four inputs, N3, N2, N1, and N0 that correspond to a 4-bit number (N3 is the most significant bit) and one output P that is 1 when the input is a prime number and that is 0 otherwise.

Since the given number of bits is 4 we are able to create prime number detector until the 15 so thet means it will be true when the given numbers are 2, 3, 5, 7, 11, 13 so let start with truth table to get function of prime numbers(F)

N3	N2	N1	N0	RESULT	DECIMAL
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	2
0	0	1	1	1	3
0	1	0	0	0	4
0	1	0	1	1	5
0	1	1	0	0	6
0	1	1	1	1	7
1	0	0	0	0	8
1	0	0	1	0	9

1	0	1	0	0	10
1	0	1	1	1	11
1	1	0	0	0	12
1	1	0	1	1	13
1	1	1	0	0	14
1	1	1	1	0	15

F =N3'N2'N1N0' + N3'N2'N1N0 + N3'N2N1'N0 + N3'N2N1N0 + N3N2'N1N0 + N3N2N1'N0

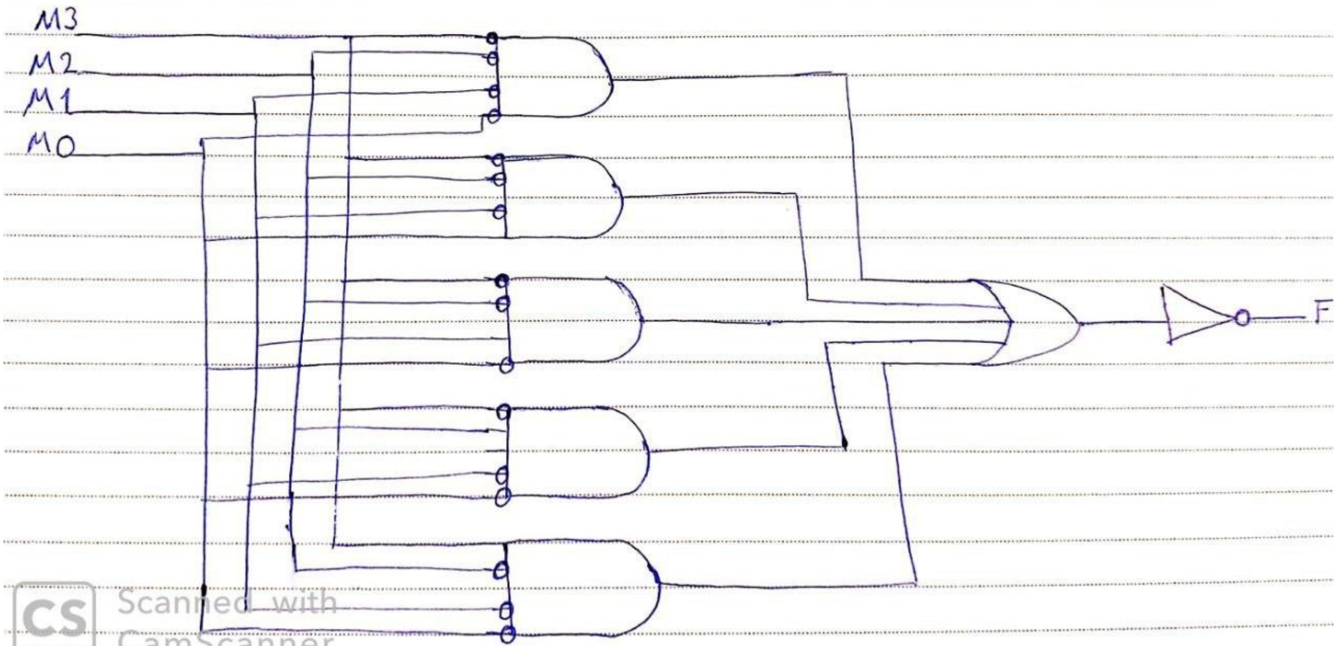


It starts from above N3 N2 N1 N0. And it gives F function
I can not find the proper gates I put there an inverter to express
inputs-------o|gate so I did the other question I draw it by my hand.

8. A network router connects multiple computers together and allows them to send messages to each other. If two or more computers send messages simultaneously, the messages âcollideâ and the messages must be resent. Using the combinational design process of Table 2.5, create a collision detection circuit for a router that connects 4 computers. The circuit has 4 inputs labeled M0 through M3 that are 1 when the corresponding computer is sending a message and 0 otherwise. The circuit has one output labeled C that is 1 when a collision is detected and 0 otherwise.

M3	M2	M1	M0	OUTPUT
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
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	1
1	1	1	1	1

For the F function we can handle it by using F ‘ function instead of F and after that we can add an inverter end of the circuit so we get the same result.
F ‘ = M3’M2’M1’M0’ + M3’M2’M1’M0 + M3’M2’M1M0’ + M3’M2M1’M0’ + M3M2’M1’M0’



M3
M2
M1
M0
F