CSE232-HW2

Nevzat Seferoglu

171044024

CSE 232 SPRING 2020

HOMEWORK 2

Due Date April 6, Monday

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

According to law;

10 years is equivalent to 10*12 months.
From the 10*12/18, there are 6 number of doubling and remains 12 months.

Number of transistors will be (**1 billion * $2^{12*10/18}$ = 101,593 billion**) transistors can be embedded in same size chip.

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

`Given general equation can written as;`
`F (a, b, c, d) = (a*b) + c + d`

a. `a=1, b=1, c=1, d=0`
`F (1, 1 ,1 ,0) = (1*1) +1+0 = 1`
b. `a=0, b=1, c=1, d=0`
`F (1, 1 ,1 ,0) = (0*1) +1+0 = 1`
c. `a=1, b=1, c=0, d=0`
`F (1, 1 ,1 ,0) = (1*1) +0+0 = 1`
d. `a=1, b=0, c=1, d=1`
`F (1, 1 ,1 ,0) = (1*0) +1+0 = 1`

3. For the function `F = a + a'b + acd + c'`:
a. List all the variables.
Variables = {a, b, c, d}
b. List all the literals.
Literals = {a, a`, b, a, c, d, c`}
c. List all the product terms.
Product Terms = {a, a`b, acd, c`}

4. Convert the function `F` shown in the truth table in the table to an equation. Don't 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

- We need to use combinational equation design process.
- Truth-table has been already given so we do not need to capture the behavior of process.
- Step2A will be useful for us, it says create an equation for each output (we have only one type of it) , by ORing the all minters for that output which is 1.

Equation can be written as below ;

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

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

Calculated equation was ;

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

F = a`b(c+c`) + b`c(a`+a) + ab(c+c`)	Distributive	Rule
= a`b + b`c + ab	Complement	Rule
= b(a+a`) + b`c	Distributive	Rule
= b + b`c	Complement	Rule
= (b+b`) * (b+c)	Distributive	Rule
= 1 * (b+c)	Complement	Rule
= b+c	Identity	Rule

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

a	b	b`	a+b	(a+b)`	(a+b)`*a	a+b`
0	0	1	1	0	0	1
0	1	0	0	1	0	0
1	0	1	1	0	0	1
1	1	0	1	0	0	1
					F != G	

F = (a + b)` * a

= (a` * b`) * a	DE Morgan Law
= (a` * a) * b`	Associative Rule
= 0 * b`	Complement Rule
= 0	Null Element Rule

0 != a + b`

F != G

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.

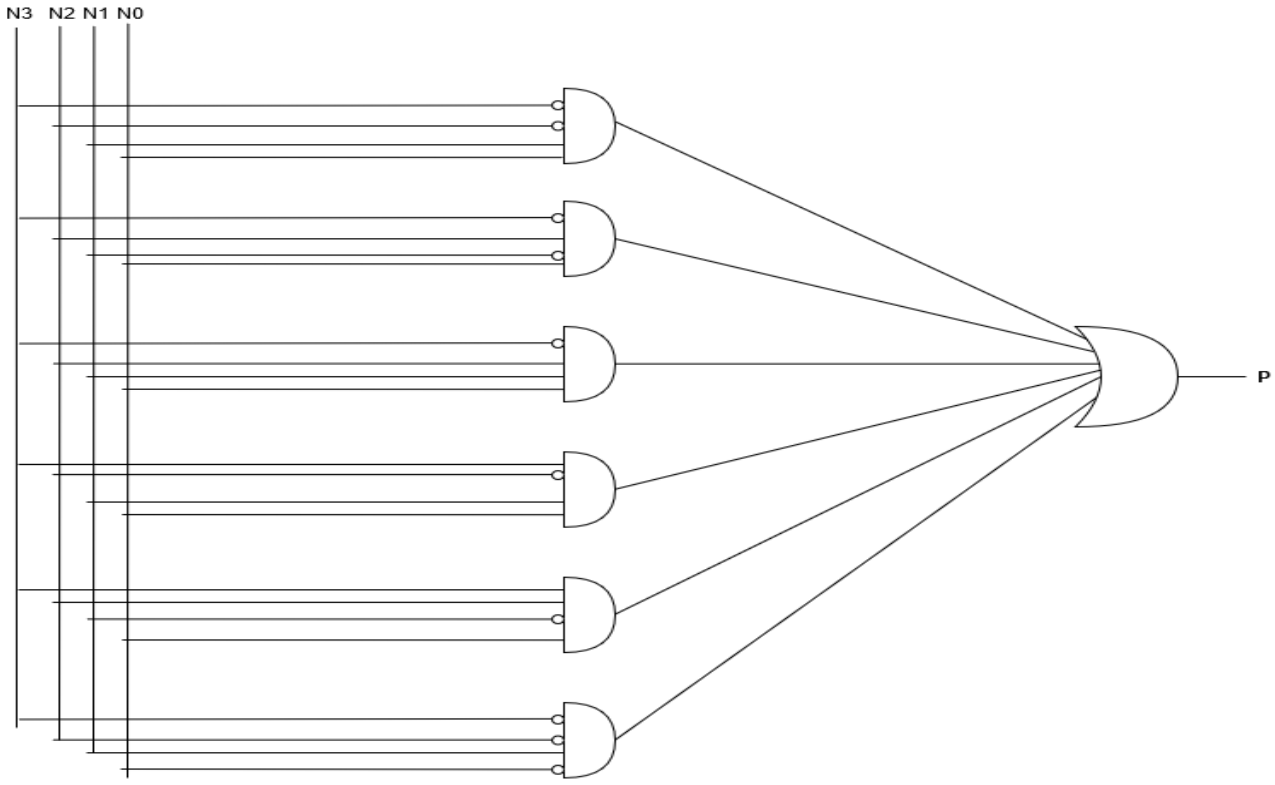
According to combinational design process , firstly I need to create a table of problem and capture the behavior of it. Normally there are 4 different input and there should be $2^4 = 16$ different rows. But I want to get rid of the final bits which is N0 , because all prime numbers are odd numbers except 2 which is 0010 according given number of bits. It will be a special case for me, and our new row amount will be $2^3 = 8$ and N0 will be 1(one) for each of them.

Special case which is two will be added later with ORing : N3`N2`N1`N0`

N3	N2	N1	N0	Decimal Equivalent	P
0	0	0	1	1	0
0	0	1	1	3	1
0	1	0	1	5	1
0	1	1	1	7	1
1	0	0	1	9	0
1	0	1	1	11	1
1	1	0	1	13	1
1	1	1	1	15	0

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

Special case which is 2.



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 "collide" 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.

Number of inputs = 4 , there must be 2^4 = 16 rows.

M0	M1	M2	M3	Collide
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

C = M0`M1`M2M3 + M0`M1M2`M3 + M0`M1M2M3` + M0`M1M2M3 + M0M1`M2`M3 + M0M1`M2M3` + M0M1`M2M3 + M0M1M2`M3` + M0M1M2`M3 + M0M1M2M3` + M0M1M2M3