Esra Eryilmaz
171044046

- CSE232 -

Homework 2

06.04.2020

(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?

Total number of months in 10 years : $\frac{1 \text{ year} \times 12 \text{ month}}{10 \text{ year} \times ?}$

$\frac{}{120 \text{ month}}$

$120 \text{ month} / 18 \text{ month} \cong 6.67$

If it doubling each 18 months $\rightarrow$ $1 \text{ billion} \cdot 2^{6.67} \cong 101.82 \text{ billion}$ transistors

(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
b) a = 0 , b = 1 , c = 1 , d = 0
c) a = 1 , b = 1 , c = 0 , d = 0
d) a = 1 , b = 0 , c = 1 , d = 1

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

(3) For the function $F = a + a'b + acd + c'$:

a) List all the variables. → represents a value (0 or 1)

b) List all the literals. → Appearance of a variable, in true or complemented form

c) List all the product terms. → Product of literals.

a) a, b, c and d.

b) a, a', b, a, c, d and c'.

c) a, a'b, acd and c'.

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

inputs a	b	c	output F
0	0	0	0
0	0	1	1 a'b'c
0	1	0	1 a'bc'
0	1	1	1 a'bc
1	0	0	0
1	0	1	1 ab'c
1	1	0	1 abc'
1	1	1	1 abc

When converting the truth table to an equation, we should consider conditions where the output is 1.

$$F = a'b'c + a'bc' + a'bc + ab'c + abc' + abc$$

(5) Use algebraic manipulation to minimize the equation obtained in Exercize 4.

$$F = a'b'c + a'bc' + a'bc + ab'c + abc' + abc$$

(distributive)

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

(distributive)

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

(complement rule)

$$F = (b'c + bc' + bc) \cdot 1$$

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

(complement rule)

$$F = (b'c + b \cdot 1)$$

(identity rule)

$$F = b'c + b$$

(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) $F = (a+b)' \cdot a$ (DeMorgan's Law)

$F = (a'b') \cdot a$ (Associative Law)

$F = (a'a) \cdot b'$ (Complement Law)

$F = b' \cdot 0$ (Null elements)

$F = 0$

$G = a + b'$

$G = a + a'b'$

$\neq$

b)

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

$\neq$

a	b	G
0	0	1
0	1	0
1	0	1
1	1	1

F and G are not equivalent

(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.

Using the combinational design process:

Step 1
(Truth table)

N3	N2	N1	N0	P	
0	0	0	0	0	
0	0	0	1	0	
0	0	1	0	1	(2)
0	0	1	1	1	(3)
0	1	0	0	0	
0	1	0	1	1	(5)
0	1	1	0	0	
0	1	1	1	1	(7)
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	1	(11)
1	1	0	0	0	
1	1	0	1	1	(13)
1	1	1	0	0	
1	1	1	1	0	

- 2^4 = 16 rows in truth table
- Output is 1 when the input is a prime number

Step 2

2A (Create Equation)

$$P = N3'N2'N1N0' + N3'N2'N1N0 + N3'N2N1'N0 + N3'N2N1N0 + N3N2'N1N0 + N3N2N1'N0$$

Simplify the equation: (complement)

$$P = \underline{N3'N2'N1}\,\overbrace{(N0' + N0)}^{1} + N3'N2N0\,\overbrace{(N1' + N1)}^{1} + \underline{N3N2'N1N0} + N3N2N1'N0$$

(use distributive)

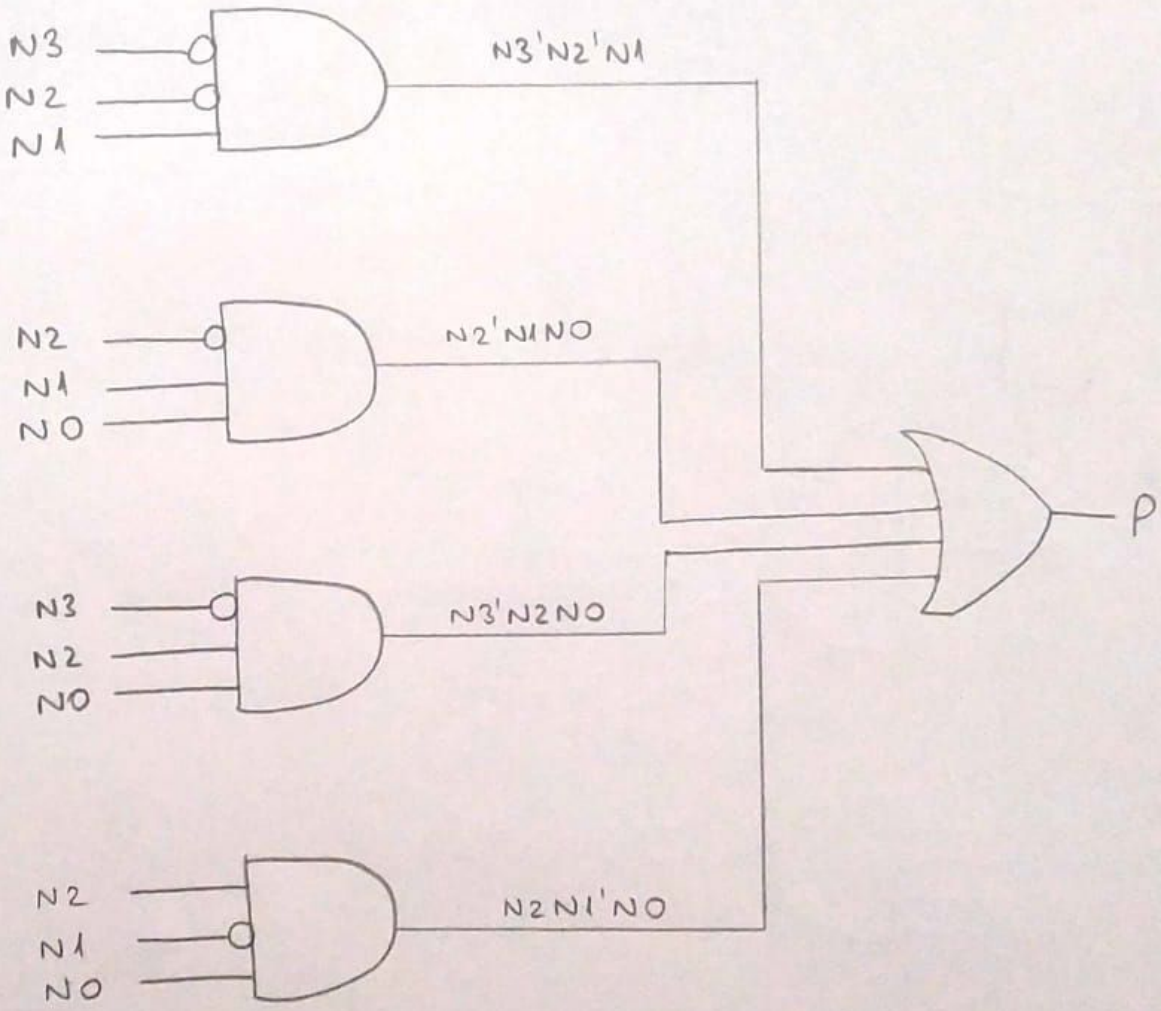
$$= N2'N1\,\underline{(N3' + N3N0)} + N2N0\,\underline{(N3' + N3N1')}$$

from $a + a'b = a + b$

$$= N2'N1\,(N3' + N0) + N2N0\,(N3' + N1')$$

$$P = N3'N2'N1 + N2'N1N0 + N3'N2N0 + N2N1'N0$$

2B (Implement as a gate-based circuit)



(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.

Using the combinational design process:

Step 1 (Truth table)

[If two or more input (computer) are 1 then I write output 1]

m3	m2	m1	m0	C	
0	0	0	0	0	m3'm2'm1'm0'
0	0	0	1	0	m3'm2'm1m0
0	0	1	0	0	m3'm2'm1m0'
0	0	1	1	1	
0	1	0	0	0	m3'm2m1'm0'
0	1	0	1	1	
0	1	1	0	1	
0	1	1	1	1	
1	0	0	0	0	m3m2'm1'm0'
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	

Step 2 (Create Equation)

2A

Since the ones in the output are less than the zeros. So I take the zeros while creating the equation and than I take the inverse of the function.

So I got a simpler equation

$$C' = m3'm2'm1'm0' + m3'm2'm1'm0 + m3'm2'm1m0' + m3'm2m1'm0' + m3m2'm1'm0'$$

Simplify the equation:

$$C' = m3'm2'm1'\underbrace{(m0' + m0)}_{1\ (\text{complement})} + m3'm2'm1m0' + m3'm2m1'm0' + m3m2'm1'm0'$$

$$C = (m3'm2'm1' + m3'm2'm1m0' + m3'm2m1'm0' + m3m2'm1'm0')'$$

2B (implement as a gate-based circuit)