Homework 1

(Due date: September 12th @ 9:30 am)
Presentation and clarity are very important!

PROBLEM 1 (15 PTS)

a) Simplify the following functions using ONLY Boolean Algebra Postulates and Theorems. For each resulting simplified function, sketch the logic circuit using AND, OR, XOR, and NOT gates.

- ✓ $F = A(C + \bar{B}) + \bar{A}$
- ✓ $F(X, Y, Z) = \prod(M_0, M_1, M_4, M_5)$
- ✓ $F = (Y + Z)(\bar{Y} + X)$
- ✓ $F = \overline{(\bar{X} + \bar{Y})Z + X\bar{Y}Z}$

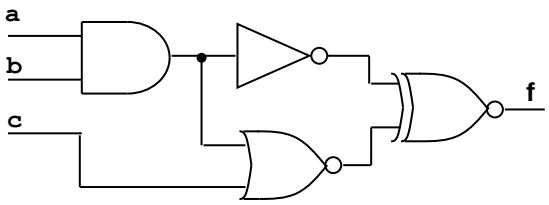
b) For the following Truth table with two outputs:

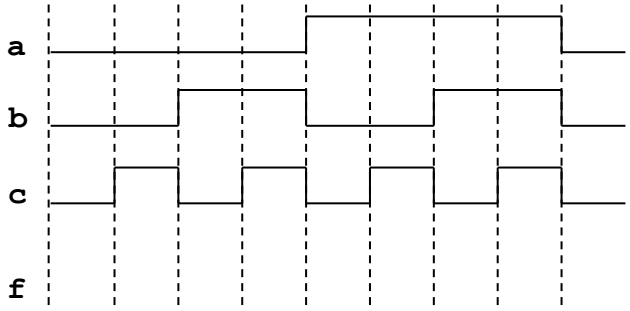
- Provide the Boolean functions using: Sum of Products (SOP), and Product of Sums (POS).
- Express the Boolean functions using the minterms and maxterms representations.
- Sketch the logic circuits.

x	y	z	f_1	f_2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	1	1
1	0	0	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

PROBLEM 2 (15 PTS)

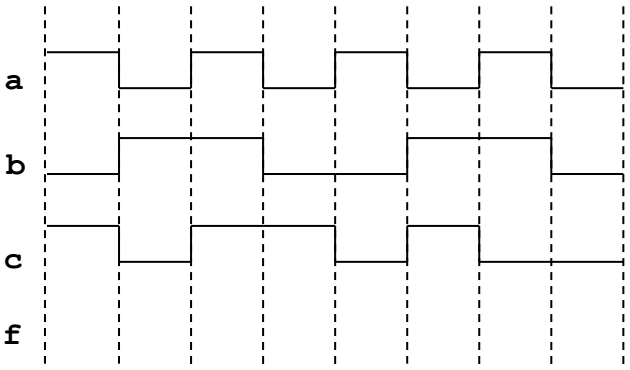
a) Complete the timing diagram of the following circuit:





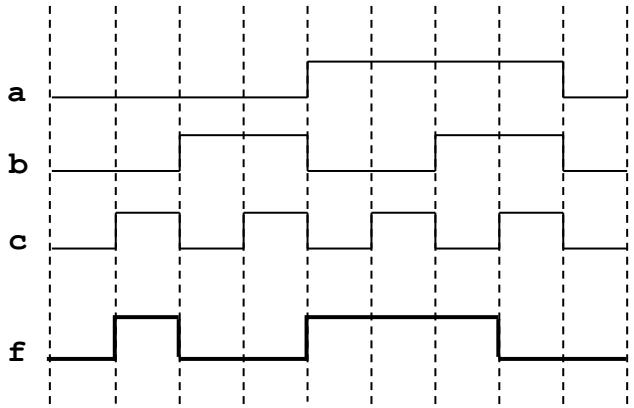
b) Complete the timing diagram of the logic circuit whose VHDL description is shown below:

```
library ieee;
use ieee.std_logic_1164.all;

entity circ is
  port ( a, b, c: in std_logic;
         f: out std_logic);
end circ;

architecture st of circ is
  signal x, y: std_logic;
begin
  x <= a and b;
  y <= x nand c;
  f <= y xor (not b);
end st;
```



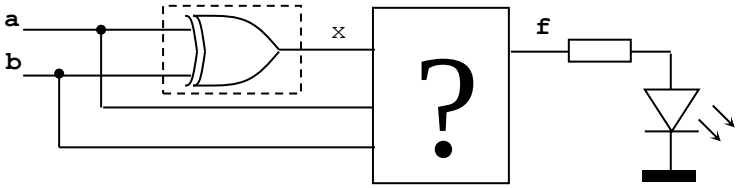
c) The following is the timing diagram of a logic circuit with 3 inputs. Sketch the logic circuit that generates this waveform. Then, complete the VHDL code.

```
library ieee;
use ieee.std_logic_1164.all;

entity circ is
 port ( a, b, c: in std_logic;
        f: out std_logic);
end circ;

architecture st of circ is
-- ???
begin
-- ???

end st;
```



PROBLEM 3 (15 PTS)

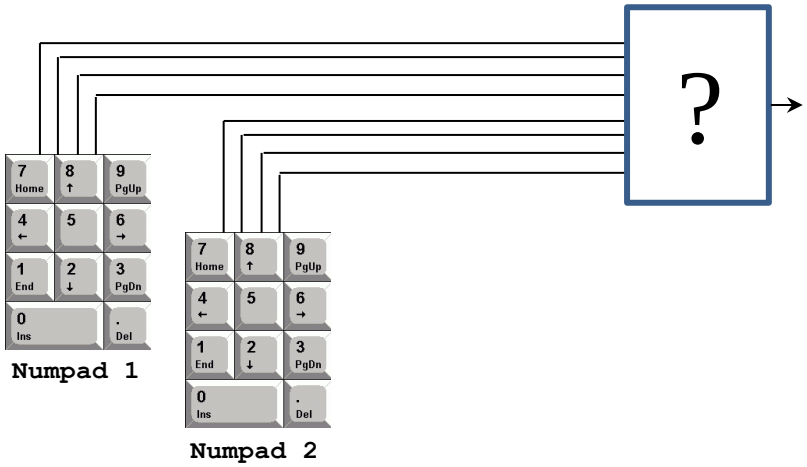
Design a circuit (simplify your circuit) that verifies the logical operation of a XOR gate. f = '1' (LED ON) if the XOR gate does NOT work properly. Assumption: when the XOR gate is not working, it generates 1's instead of 0's and vice versa.



PROBLEM 4 (10 PTS)

Design a logic circuit (simplify your circuit) that opens a lock (f = '1') whenever one presses the correct number on each numpad. We encode each decimal number on the numpad using BCD encoding. We expect that each group of 4 bits be in the range from 0000 to 1001, the values from 1010 to 1111 are assumed not to occur.

- Tip: create two circuits: one that verifies the first number (**9**), and the other that verifies the second number (**5**). Then perform the AND operation on the two outputs. This avoids creating a truth table with 8 inputs!

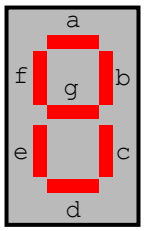


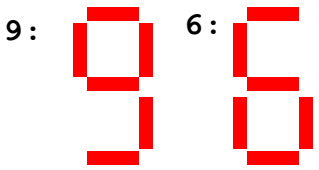
PROBLEM 5 (20 PTS)

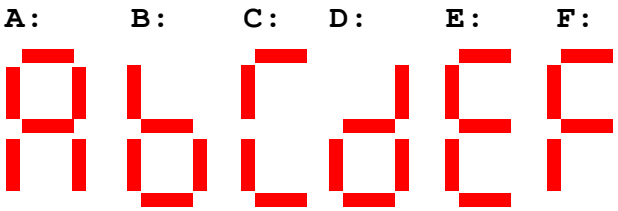
We want to display the hexadecimal value of a 4-bit number on a 7-segment display. The LEDs are lit with a logical '0' (negative logic or active low). The inputs are active high (or in positive logic).

- Complete the truth table for each output (a-g).
- Provide the simplified expression for each output (a-g)

b_3	b_2	b_1	b_0	a	b	c	d	e	f	g
0	0	0	0							
0	0	0	1							
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1	0	0	0	0	1	0	0
1	0	1	0							
1	0	1	1							
1	1	0	0							
1	1	0	1							
1	1	1	0	0	1	1	0	0	0	0
1	1	1	1							







PROBLEM 6 (25 PTS)

In these problems, you MUST show your conversion procedure.

a) Convert the following decimal numbers to i) binary, ii) octal, and iii) hexadecimal.
- 124, 200, 115, 128, 511.25, 64.625, 19.6875.

b) What is the minimum number of bits required to represent:
- 50,000 colors?
- 32679 symbols?
- Numbers between 25,000 and 29,095?
- 65536 memory addresses in a computer?

c) A microprocessor can handle addresses from 0x0000 to 0x7FFF. How many bits do we need to represent those addresses?

d) Complete the following table.

Decimal	BCD	Binary number	Reflective Gray Code
		1010100110	
			1011100010
128			
		10101011	
	01001001		
			10001001
		1110010	
			110011101
442			
	011000110001		