Homework 1

(Due date: September 20th @ 5:30 pm)
Presentation and clarity are very important!

Problem 1 (27 pts)

a) Simplify the following functions using ONLY Boolean Algebra Theorems. For each resulting simplified function, sketch the logic circuit using AND, OR, XOR, and NOT gates. (14 pts)

- ✓ $F = \overline{(A \oplus \bar{B})C + AB\bar{C}}$
- ✓ $F = \overline{\bar{X}\bar{Y} + Y(\bar{Z} + \bar{X})}$
- ✓ $F(A,B,C) = \prod(M_0, M_3, M_5, M_7)$
- ✓ $F = (A + \bar{C} + \bar{D})(\bar{B} + \bar{C} + D)(A + \bar{B} + \bar{C})$

b) Using ONLY Boolean Algebra Theorems, demonstrate: (5 pts)

$$X(Y \oplus Z) = (XY) \oplus (XZ)$$

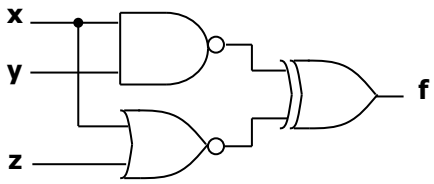
c) For the following Truth table with two outputs: (8 pts)

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

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

Problem 2 (26 pts)

a) Construct the truth table describing the output of the following circuit and write the simplified Boolean equation (7 pts).

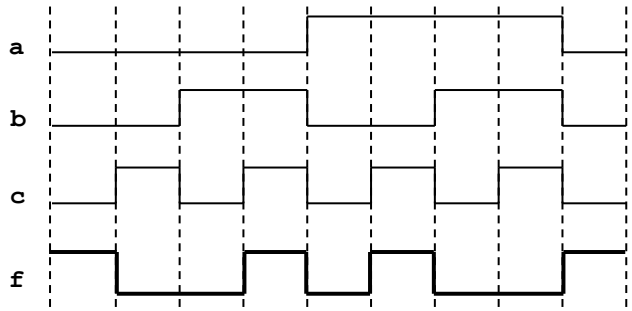


x	y	z	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

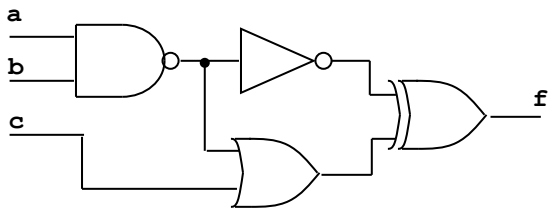
$f =$

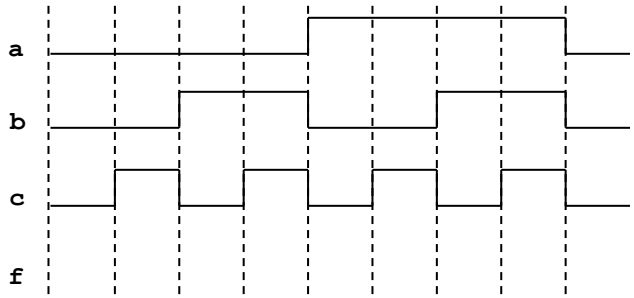
b) The following is the timing diagram of a logic circuit with 3 inputs. Sketch the logic circuit that generates this waveform. Then, complete the VHDL code. (8 pts)

```
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;
```



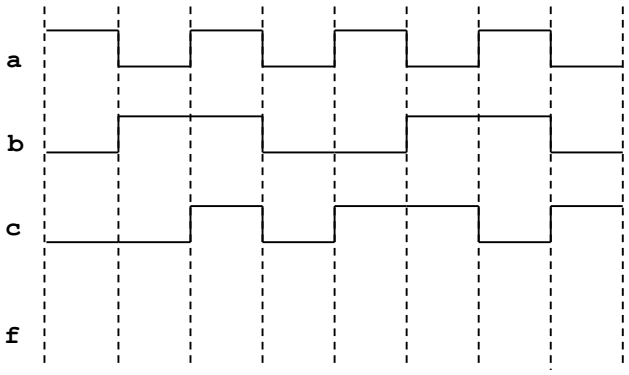
c) Complete the timing diagram of the following circuit: (5 pts)





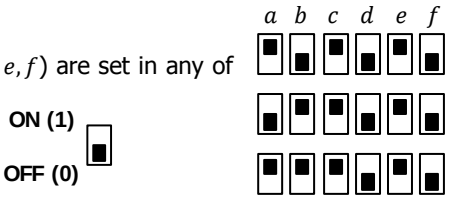
d) Complete the timing diagram of the logic circuit whose VHDL description is shown below: (6 pts)

```
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
  y <= x nand c;
  x <= a and b;
  f <= (not y) xor b;
end st;
```



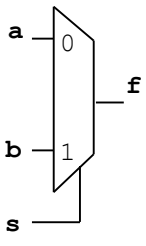
PROBLEM 3 (9 PTS)

- Security combinations: A lock only opens ($z = 1$) when the 6 switches (a, b, c, d, e, f) are set in any of the 3 configurations shown in the figure, otherwise the lock is closed ($z = 0$).
  - ✓ Provide the Boolean equation for the output z and sketch the logic circuit.



PROBLEM 4 (13 PTS)

a) The following circuit has the following logic function: $f = \bar{s}a + sb$.
   - ✓ Complete the truth table of the circuit, and sketch the logic circuit (3 pts)



s	a	b	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

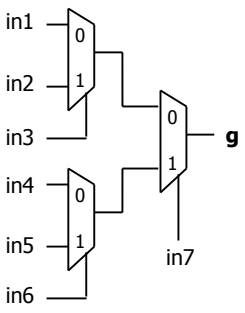
b) We can use several instances of the previous circuit to implement different functions. (10 pts)

- For example, the following selection of inputs produce the function: $g = x_2 + x_1\overline{x_3}$. Demonstrate that this is the case.

in1	in2	in3	in4	in5	in6	in7
x_1	0	x_3	x_2	1	x_3	x_2

- Given the following inputs, provide the resulting function g (minimize the function).

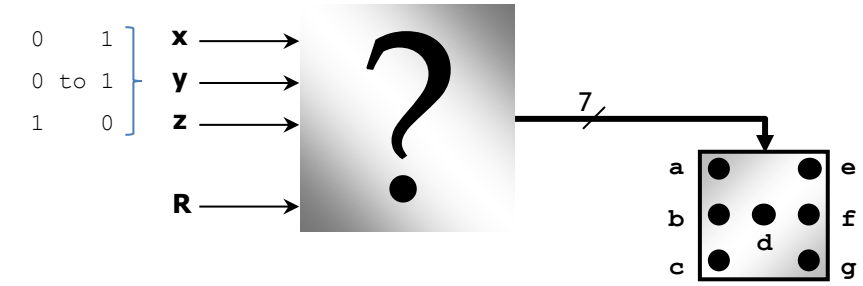
in1	in2	in3	in4	in5	in6	in7
x_3	1	x_1	x_1	0	x_2	x_3

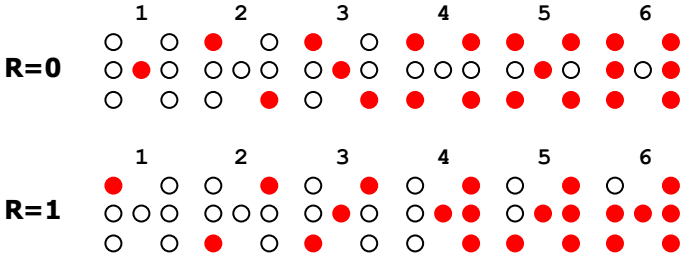


PROBLEM 5 (25 PTS)

- An array of seven LEDs is used to display the results of a roll of a die. Numeric data (1-6) is produced as a 3-bit value. We want to design a logic circuit that converts that 3-bit value to the corresponding 7-bit LED pattern in a die. For example, the code `101` is displayed such that it represents the number '5' in a die side.
- In addition, we have an input `R`. When `R=0`, values are displayed as in a normal die. When `R=1`, values are displayed a little bit different. See figure for details.
- Note: The LEDs are lit with a logical '1' (positive logic). The inputs are active high (or in positive logic).

  ✓ Complete the truth table for each output (a, b, c, d, e, f, g). Note that it is safe to assume that the inputs x, y, z will not produce the values `000` and `111`.
  ✓ Provide the simplified expression for each output (a, b, c, d, e, f, g). Use Karnaugh maps for a, b, c, d, e and the Quine-McCluskey algorithm for f, g.





Value	R	x	y	z	a	b	c	d	e	f	g
	0	0	0	0							
1	0	0	0	1							
2	0	0	1	0	1	0	0	0	0	0	1
3	0	0	1	1							
4	0	1	0	0							
5	0	1	0	1							
6	0	1	1	0							
	0	1	1	1							
	1	0	0	0							
1	1	0	0	1							
2	1	0	1	0							
3	1	0	1	1							
4	1	1	0	0							
5	1	1	0	1	0	0	1	1	1	1	1
6	1	1	1	0							
	1	1	1	1							