

Homework 1

(Due date: January 22nd @ 5:30 pm)
 Presentation and clarity are very important!

PROBLEM 1 (20 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. (12 pts)

✓ $F = A(B + \bar{C}) + \bar{A}$

✓ $F = (Z + \bar{Y})(\bar{Z} + X)(\bar{Y} + X)$

✓ $F(X, Y, Z) = \prod(M_2, M_4, M_6, M_7)$

✓ $F = \overline{(X + \bar{Y})Z} + \bar{X}YZ$

b) 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	0	0
0	0	1	1	0
0	1	0	1	1
0	1	1	1	1
1	0	0	1	0
1	0	1	0	1
1	1	0	1	1
1	1	1	0	1

PROBLEM 2 (20 PTS)

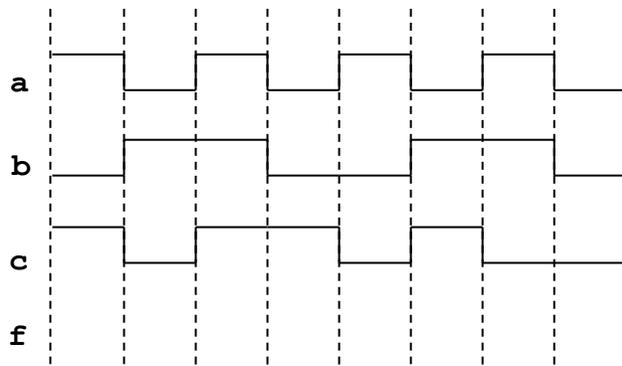
a) Complete the timing diagram of the logic circuit whose VHDL description is shown below: (5 pts)

```

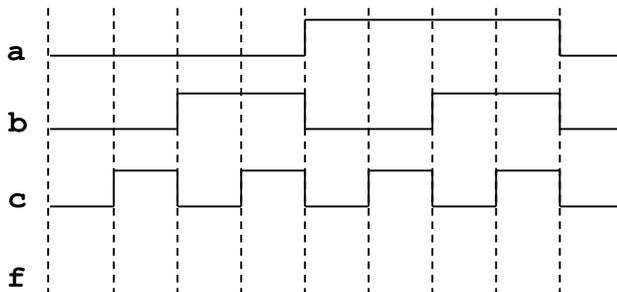
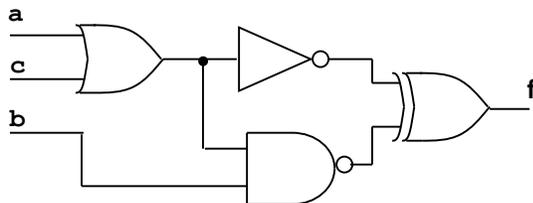
library ieee;
use ieee.std_logic_1164.all;

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
    x <= a nor b;
    y <= x xor c;
    f <= y nand (not a);
end st;
    
```



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



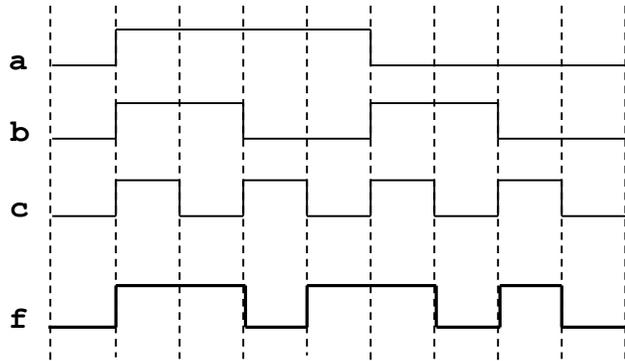
c) The following is the timing diagram of a logic circuit with 3 inputs. Sketch the logic circuit that generates this waveform. Then, complete the VHDL code. (10 pts)

```

library ieee;
use ieee.std_logic_1164.all;

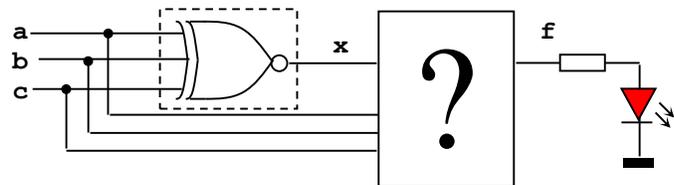
entity circ is
    port ( a, b, c: in std_logic;
          f: out std_logic);
end circ;

architecture st of circ is
    -- ???
begin
    -- ???
end st;
    
```



PROBLEM 3 (15 PTS)

Design a circuit (simplify your circuit) that verifies the logical operation of a 3-input XNOR gate. $f = '1'$ (LED ON) if the XNOR gate does NOT work properly. Assumption: when the XNOR gate is not working, it generates 1's instead of 0's and vice versa.



PROBLEM 4 (30 PTS)

A 16-letter keypad produces a 4-bit code as shown in the table. We want to design a logic circuit that converts those 4-bit codes to Braille code, where the 6 dots are represented by LEDs. A raised (or embossed) dot is represented by an LED ON (logic value of '1'). A missing dot is represented by an LED off (logic value of '0').

- ✓ Complete the truth table for each output (Q_0 - Q_5).
- ✓ Provide the simplified expression for each output (Q_5 - Q_2 and the Quine-McCluskey algorithm for Q_1 - Q_0). Note it is safe to assume that the codes 1110 and 1111 will not be produced by the keypad.

x	y	z	w	Letter
0	0	0	0	a
0	0	0	1	b
0	0	1	0	c
0	0	1	1	d
0	1	0	0	e
0	1	0	1	f
0	1	1	0	g
0	1	1	1	h
1	0	0	0	i
1	0	0	1	j
1	0	1	0	k
1	0	1	1	l
1	1	0	0	m
1	1	0	1	n
1	1	1	0	
1	1	1	1	

a	b	c	d	e	f	g	h	i	j	k	l	m	n
● ○	● ○	● ●	● ●	● ○	● ●	● ●	● ○	○ ●	○ ●	● ○	● ○	● ●	● ●
○ ○	● ○	○ ○	○ ●	○ ●	○ ○	● ●	● ●	● ○	● ●	○ ○	● ○	○ ○	○ ●
○ ○	○ ○	○ ○	○ ○	○ ○	○ ○	○ ○	○ ○	○ ○	○ ○	● ○	● ○	● ○	● ○

PROBLEM 5 (15 PTS)

The following die has a sensor on each side. Whenever a side rests on a surface, the sensor on that side outputs a logic '1' (which is transmitted wirelessly to a controller); otherwise, it outputs a '0'. The sensors are named s_i , where i is the number printed on the respective side.

We want to design a circuit that reads the state of the 6 sensors and outputs the value of the upper surface. That decimal value is represented using the binary number system (see table below). A LED ON is represented by the logic value of '0', and a LED OFF is represented by a logic value of '1'.

For example, in the figure below, the resting side has six dots while the upper surface has one. The resulting decimal number is then '1', which is represented by 001 on the LEDs. The state of the sensors is $s_6=1, s_5=0, s_4=0, s_3=0, s_2=0, s_1=0$.

Under normal circumstances, we expect only one sensor activated at a time. However, due to a variety of problems, we might have the following cases:

- Two or more sensors produce a '1' at the same time: In this case, the state of the LEDs must be 000.
 - No sensor produces a '1': In this case, the state of the LEDs must be 000.
- ✓ Using the state of the sensors as inputs, provide the Boolean expression for each LED: L_2, L_1, L_0 . Use the canonical Sum of Products for each function.

