Homework 1

(Due date: September 21st @ 5:30 pm)

Presentation and clarity are very important!

PROBLEM 1 (25 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)

- ✓ $F(X,Y,Z) = \prod(M_0, M_1, M_4, M_6)$
- ✓ $F = (A + \bar{C} + D)(\bar{A}C + \bar{D})$
- ✓ $F = \overline{X(\bar{Y} \oplus \bar{Z}) + \bar{Y}}$
- ✓ $F = \overline{(\overline{A+B})C + AB\bar{D}}$

b) Determine whether or not the following expression is valid, i.e., whether the left- and right-hand sides represent the same function. Suggestion: complete the truth tables for both sides: (5 pts)

$$\overline{x_1}\,\overline{x_3} + x_2x_3 + x_1\overline{x_2} = \overline{x_1}x_2 + x_1x_3 + \overline{x_2}\,\overline{x_3}$$

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

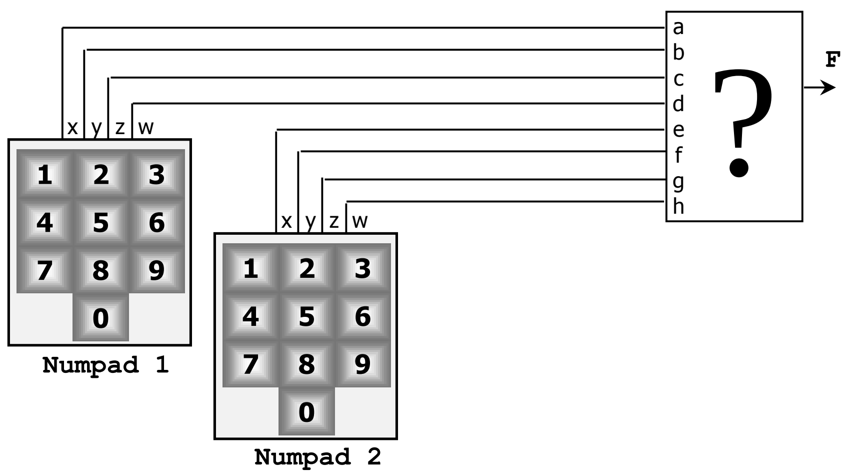
- Provide the Boolean functions using the Canonical Sum of Products (SOP), and Product of Sums (POS).
- Express the Boolean functions using the minterms and maxterms representations.
- Sketch the logic circuits as Canonical Sum of Products and Product of Sums.

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

PROBLEM 2 (15 PTS)

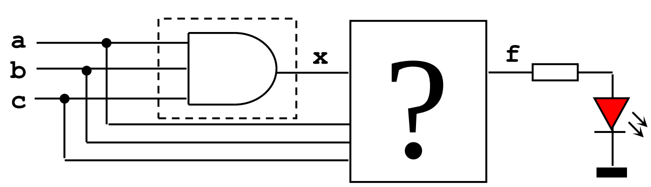
- Design a logic circuit (<u>simplify your circuit</u>) that opens a lock ($f = 1$) whenever the user presses the correct number on each numpad (numpad 1: **8**, numpad2: **3**). The numpad encodes each decimal number using BCD encoding (see figure). We expect that the 4-bit groups generated by each numpad be in the range from `0000` to `1001`. Note that the values form `1010` to `1111` are assumed not to occur.
  <u>Suggestion</u>: Create two circuits: one that verifies the first number (**8**), and another that verifies the second number (**3**). Then perform the AND operation on the two outputs. This avoids creating a truth table with 8 inputs.
- Sketch the resulting logic circuit using ONLY 2-input NAND gates. (5 pts)

BCD code x	y	z	w	Number pressed
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9



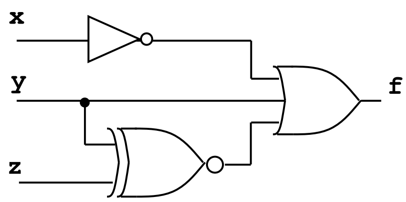
PROBLEM 3 (11 PTS)

- Design a circuit (<u>simplify your circuit</u>) that verifies the logical operation of a 3-input AND gate. f = '1' (LED ON) if the AND gate works properly. Assumption: when the AND gate is not working, it generates 1's instead of 0's and vice versa.



PROBLEM 4 (23 PTS)

a) Construct the truth table describing the output of the following circuit and write the simplified Boolean equation (5 pts).

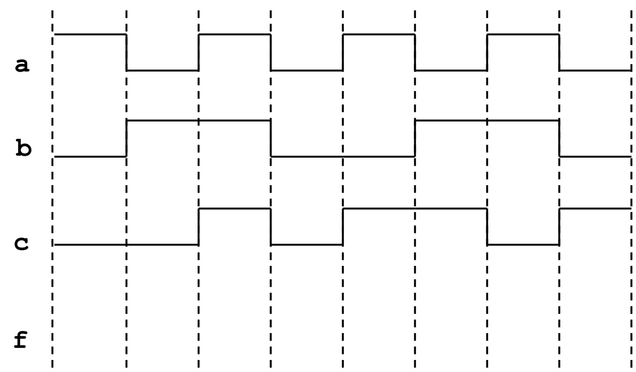


x	y	z	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

$f =$

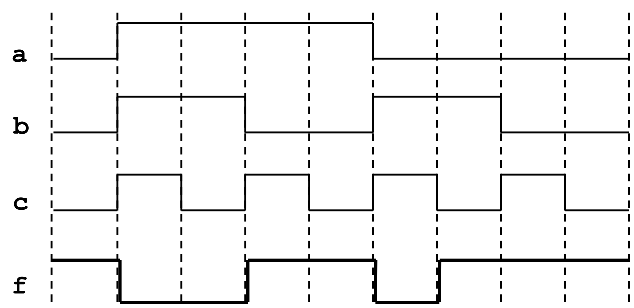
b) Complete the timing diagram of the logic circuit whose VHDL description is shown below: (5 pts)

```
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 struct of circ is
  signal x, y: std_logic;

begin
  x <= a xnor c;
  y <= x nand b;
  f <= y and (not b);
end struct;
```

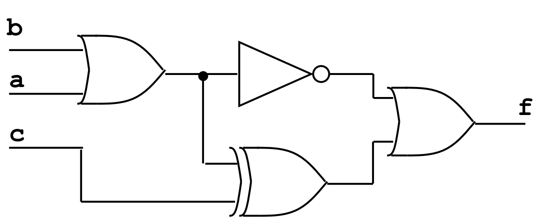


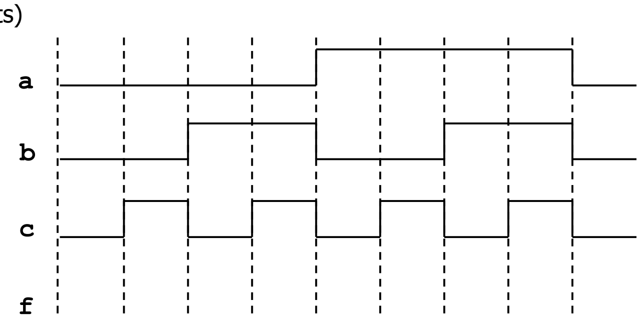
c) The following is the timing diagram of a logic circuit with 3 inputs. Sketch the logic circuit that generates this waveform. Then, complete the VHDL code. (8 pts)

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

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

architecture struct of wav is
-- ???
begin
   -- ???
end struct;
```



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

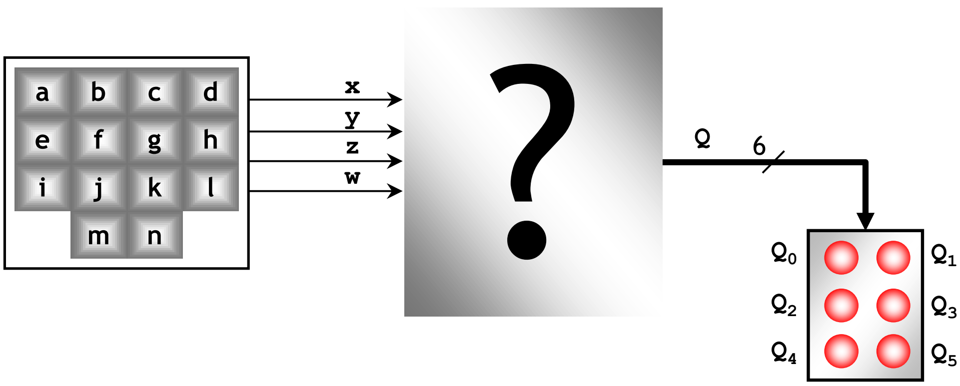


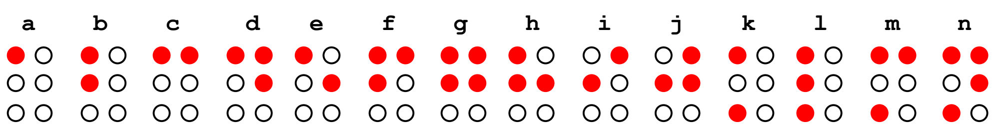


PROBLEM 5 (26 PTS)

- A 14-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 a 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_0-Q_5). Use Karnaugh maps for Q_5,Q_4,Q_1,Q_0 and the Quine-McCluskey algorithm for Q_3-Q_2. 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	