Homework 1

(Due date: January 20th @ 5:30 pm)

Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (28 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. (15 pts)

- ✓ $F(x,y,z) = \prod(M_0, M_1, M_4, M_6)$
- ✓ $F = \overline{x(y \oplus \bar{z}) + \bar{y}}$
- ✓ $F = (\bar{A}C + \bar{D})(A + \bar{C} + D)$

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)

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

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). (4 pts)
- Express the Boolean functions using the minterms and maxterms representations.
- Sketch the logic circuits as Canonical Sum of Products and Product of Sums. (3 pts)

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 (10 PTS)

- The following is a truth table for logic functions f_1 and f_2. Note that an 'X' on the input means that the logical value can be either '0' of '1'. So, if the input $xyzw$ is 01XX, it means that for the output f_1 to be 1, we only need $x = 0$ and $y = 1$.

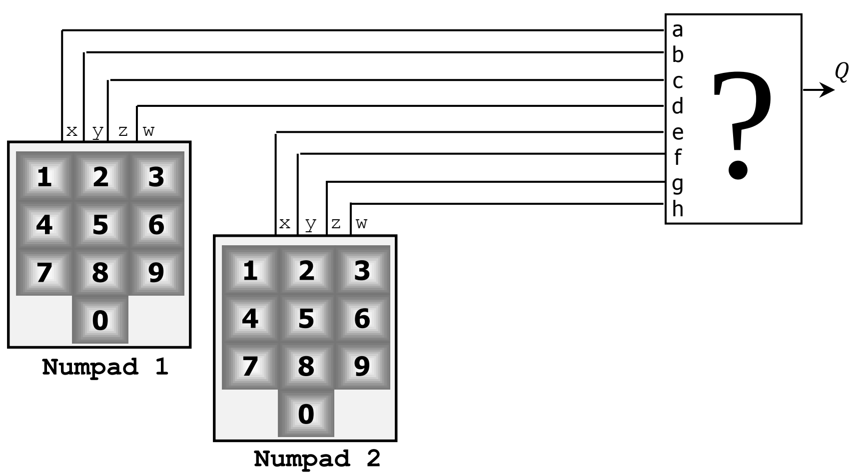
  ✓ Provide the simplified Boolean expressions for f_1 and f_2.

x	y	z	w	f_1	f_2
1	X	X	X	1	1
0	1	X	X	1	0
0	0	1	X	0	1
all others				0	0

PROBLEM 3 (11 PTS)

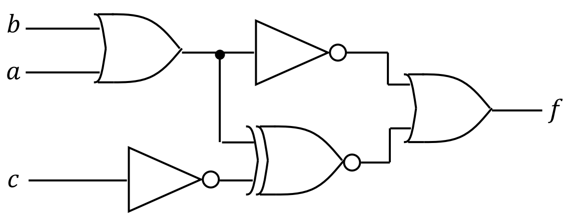
- We want to design a logic circuit that opens a lock ($Q = 1$) whenever the user presses the correct number on each numpad (numpad 1: **8**, numpad2: **3**). The numpad encodes each decimal number using BCD encoding (see figure). We expect that the 4-bit groups generated by each numpad be in the range from `0000` to `1001`. Note that the values from `1010` to `1111` are assumed not to occur.
- Provide the simplified expression for $Q(a,b,c,d,e,f,g,h)$ and sketch the logic circuit.
  Suggestion: 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.

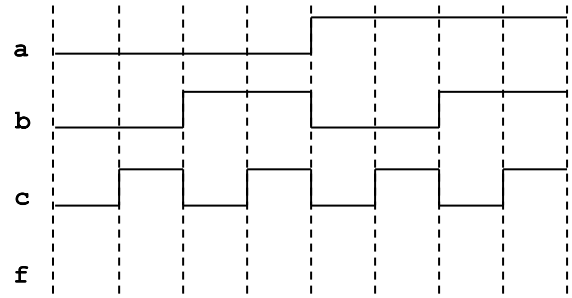
Number pressed	BCD code			
	x	y	z	w
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1



PROBLEM 4 (26 PTS)

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





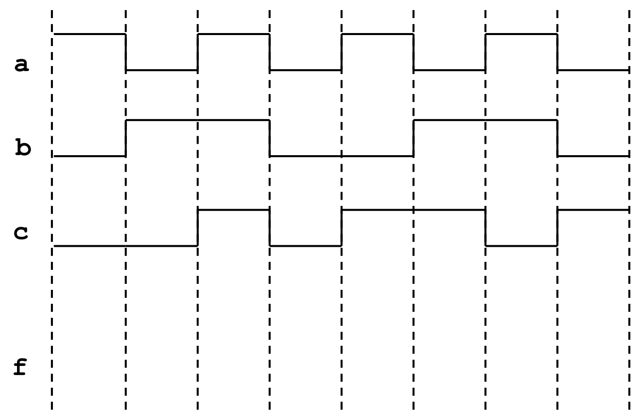
b) Complete the timing diagram of the logic circuit whose VHDL description is shown below: (7 pts)

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

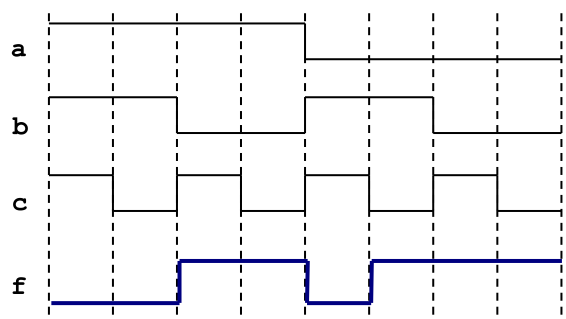


c) The following is the timing diagram of a logic circuit with 3 inputs. Sketch the simplified logic circuit that generates this waveform. Then, complete the VHDL code. (8 pts)

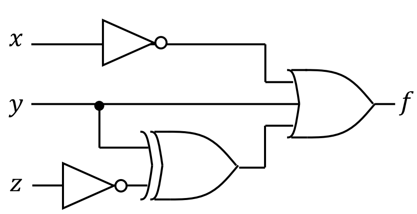
```
library ieee;
use ieee.std_logic_1164.all;

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) Construct the truth table describing the output of the following circuit and write the simplified Boolean equation (6 pts).



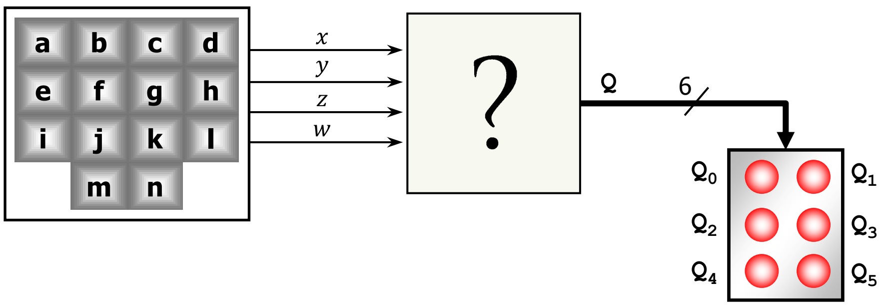
x	y	z	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

$f =$

PROBLEM 5 (25 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). (4 pts)
✓ 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