UPC. EETAC. Bachelor Degree. 2A. Digital Circuits and Systems (CSD). F. J. Robert, J. Jordana. Questions about the exam: Lecturers' office time. Grades will be available on November 14.

Exam 1 **November 7, 2018**

Problem 1.

Taking into account the truth table in Fig. 1 of a digital circuit named *Circuit_1*.

D	*C*	*B*	*A*	*Y*	*W*
0	0	0	0	1	0
0	0	0	1	1	1
0	0	1	0	1	1
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	1	1	1
0	1	1	0	1	1
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	1

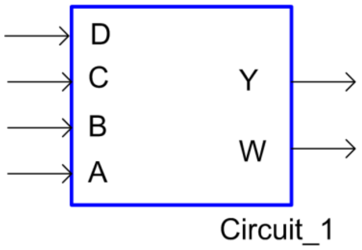

Fig. 1 Example of a combinational circuit symbol, truth table and minimisation results.

```
MINIMIZATION RESULT STATISTICS
==============================
FOUND 6 ESSENTIAL PRODUCT TERMS
MAXIMUM FANIN:                 4
TOTAL LITERAL COUNT:           18
MAXIMUM PRODUCT TERM SIZE:     3
MAXIMUM OUTPUT FUNCTION SIZE:  3
=========
DCBA    YW
=========
0---  | 1.
--1-  | 1.
---1  | 1.
0-01  | .1
0-10  | .1
111-  | .1
```

Equations

a) Express **Y** = *f*(D, C, B, A) as a product of maxterms. Express **W** = *f*(D, C, B, A) as a sum of minterms. Expand the **W** expression to calculate and verify its value when D = 1, C = 0, B = 1, A = 1.

b) The minimisation of **Y** and **W** by *minilog.exe* gives the result in Fig. 1, express **Y** and **W** as SoP and verify their values when D = 1, C = 0, B = 1, A = 1. Draw the logic circuit equivalent to these minimised equations.

Planning

c) Plan A, structural. Translate the equations **Y** and **W** obtained in b) into VHDL.

d) Plan C2, hierarchical. Implement **Y** and **W** using the method of multiplexers and using MUX_2.

e) Plan C2, hierarchical. Implement **Y** and **W** using the method of decoders. How many VHDL files are necessary in this project if the required decoder is build using *Dec_2_4* components?

f) Plan B, behavioural. Draw the truth table as a schematic or a flow chart and translate it into a VHDL architecture.

Problem 2.

This project aims to represent in 7-segment displays the number of occupied parking slots. Each slot has installed an ultrasonic presence sensor which gives a '1' when occupied. Thus, the first idea here for planning the entity *Parking_occupancy* in Fig. 2 is to consider components such as a *Ones_counter_32bit* where for example an input vector such as **D** = "1001 0011 1110 1111 1000 1111 1010 1110" will produce an output **K** = $(010101)_2 = (21)_{10}$; a *Converter_bin_BCD_6bit* where for example an input such as **K** = $(010101)_2$ will generate and output **T** = "0010", **U** = "0001" ; and a pair of *HEX_7seg_decoder* to drive the 7-segment displays.

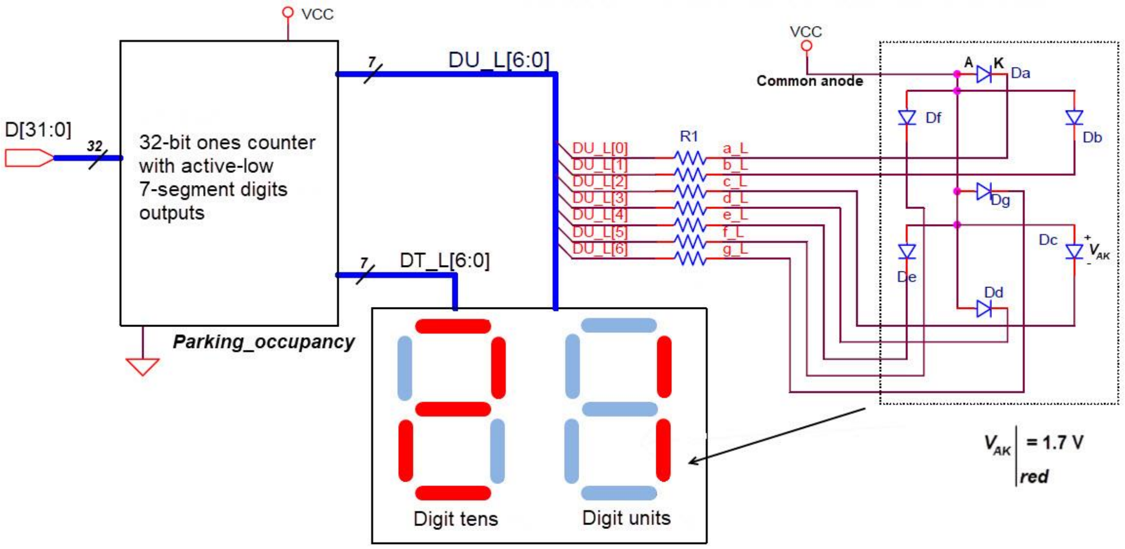

Fig. 2
Example of a parking occupancy monitor to calculate the number of occupied parking slots.

In this example it is represented the number 21, meaning this number of detected cars in any position in the parking.

V_{CC} = 5 V

a) Draw and explain the internal architecture of the parking occupancy circuit based on components and representing some examples of the components truth tables.

b) The *HEX_7seg_decoder* has active-low outputs to drive a common-anode display and its technology is LS-TTL with the characteristics represented in the table. Calculate the value of the limiting resistor **R1** in the worst case scenario if each segment must be biased with 15 mA when lighting.

Symbol	Parameter	Limits			Unit
		Min	Typ	Max	
V_{IH}	Input HIGH Voltage	2.0			V
V_{IL}	Input LOW Voltage			0.8	V
V_{OH}	Output HIGH Voltage	2.7	3.5		V
V_{OL}	Output LOW Voltage		0.25	0.4	V

Symbol	Parameter		Min	Max	Unit
t_{PLH}	Propagation Delay Time	LOW-to-HIGH Level Output	4	15	ns
t_{PHL}	Propagation Delay Time	HIGH-to-LOW Level Output			

c) The *Converter_bin_BCD_6bit* is used to translate 6-bit radix-2 numbers to 2 BCD digits. Assuming the circuit is based on equations PoS (plan A) and implemented in LS-TTL technology where each gate has propagation delays as indicated in the table, calculate the maximum speed of computing.

d) Invent the architecture of the *ones_counter_32bit* as a hierarchy of components (plan C2). For instance, Fig. 3 represents the schematic of a *ones_counter_8bit*. How many VHDL files will include this project? Check that your circuit works applying some input vectors.

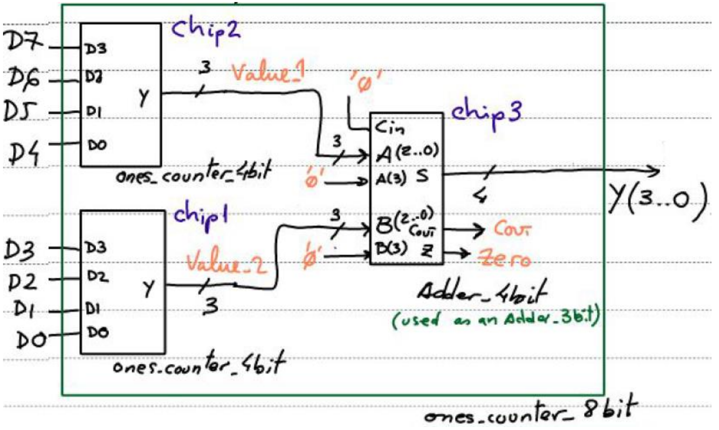

Fig. 3
Example of building a *ones_counter_8bit* using smaller components like *ones_counter_4bit* and *Adder_4bit*.

This is the truth table of a ones_*counter_4bit*

D(3..0)				Y(2..0)		
0	0	0	0	0	0	0
	⋮				⋮	
0	0	1	1	0	1	0
	⋮				⋮	
1	1	1	0	0	1	1
	⋮				⋮	
1	1	1	1	1	0	0