



## EX2 DIGITAL ELECTRONICS

G\_\_\_\_\_

After completing the task and studying Units 1.5 and 1.7 students will be able to: (check all that apply):

- Use and explain the functionality of logic gates AND, NAND, OR, NOR, XOR, NXOR, NOT
- Analyze a logic circuit build using logic gates
- Explain and relate the following concepts for designing a logic circuit: truth table, Karnaugh map, Boolean<sup>1</sup> Algebra and logic functions, SoP (sum of products) and PoS (product of sums), canonical algebraic equations, minterms and maxterms
- Simplify or minimize logic function minimization up to 5 input variables by means of Karnaugh maps
- Find the datasheets of the Small Scale of Integration (SSI) integrated circuits
- Capture a diagram schematic in Proteus-VSM and run the simulation
- Simulate a digital circuit (inside a black box) using the virtual laboratory software Proteus-VSM<sup>2</sup>
- Produce a written solution for the exercise using the instructions from:  
[http://epsc.upc.edu/projectes/ed/unitats/unitat\\_1\\_1/Criteris\\_Correccio\\_Exercici.pdf](http://epsc.upc.edu/projectes/ed/unitats/unitat_1_1/Criteris_Correccio_Exercici.pdf)
- Work cooperatively in a team of 3 members using the method described in:  
[http://epsc.upc.edu/projectes/ed/problemes/metode\\_resolucio\\_cooperativa\\_recomanat.pdf](http://epsc.upc.edu/projectes/ed/problemes/metode_resolucio_cooperativa_recomanat.pdf)

Write down the most significant doubts or questions you have had while or after completing the task:

- 
- 
- 
- 
- 

### STATEMENT:

My signature below indicates that I have (1) made equitable contribution to **EX2** as a member of the group, (2) read and fully agree with the contents (i.e., results, conclusions, analyses, simulations) of this document, and (3) acknowledged by name anyone outside this group who assisted this learning team or any individual member in completing this document.

Today's date: \_\_\_\_\_

Active members

Roles: (reporter, simulator, etc.)

- (1) \_\_\_\_\_
- (2) \_\_\_\_\_
- (3) \_\_\_\_\_

- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_

Acknowledgement of individual(s) who assisted this group in completing this document:

- (1) \_\_\_\_\_
- (2) \_\_\_\_\_

Study time (in hours)	Group work	Sessions TGA, TGB		Sessions TGC	
		Individual			Student 1
				Student 2	
				Student 3	

<sup>1</sup> You may get more information about who was the mathematician George Boole in this link:  
<http://www-gap.dcs.st-and.ac.uk/%7Ehistory/Mathematicians/Boole.html>

<sup>2</sup> Proteus-VSM is the proprietary software in use for the whole course: <http://www.labcenter.com> for having effectively a "digital circuits' laboratory in the classroom"

## Analyzing a simple car's obstacle detection and emergency braking system

We want to build an electronic system for a car which depending on the car velocity and distance to the obstacle, has to make a normal or an emergency brake in order to avoid the clash.

- Sensors:
  - Speed sensor  $v$  : 5 bit, coded in binary: Range: 0 to 31 m/s) (0 ...112 km/h)
  - Distance sensor  $D$ : 7 bit coded in binary. Range: 1 to 127 m (if output is '0' there isn't obstacle detected)
- Actions to be taken:
  - Normal braking  $b$  at an deceleration of  $-4 \text{ m/s}^2$
  - Emergency braking  $e$  at  $-8 \text{ m/s}^2$

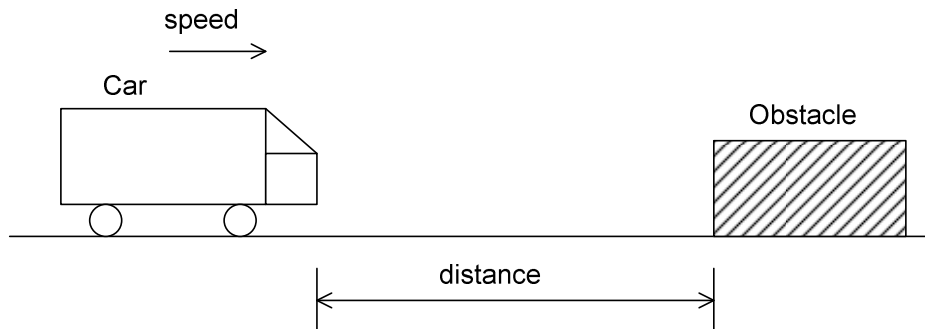


Fig. 1 The quantities involved in the system

Plan:

1. Theory review of mechanical quantities<sup>3</sup>: distance, speed, acceleration. The process of braking (negative acceleration) at an established distance from an obstacle. Find the design equation.
2. Propose a block diagram for the digital system to be implemented
3. Design every block and verify its operation by simulation
4. Simulate the final design and check its functionality using several numeric examples



Fig. 2 Accident-free driving may be a long-term dream, but automotive engineers have already started to develop **accident avoidance systems**<sup>4</sup>

<sup>3</sup> This web page is very helpful for reviewing the physical process  
<http://www.physicsclassroom.com/mmedia/kinema/pvna.html>

<sup>4</sup> Picture from <http://www.can-cia.org/applications/passengercars/invehicle.html>

1. The braking system control equation :

The car displacement **D** after a constant negative acceleration **a** will depends on the initial velocity squared as:

$$D = \frac{v_i^2}{2a}$$

Making it very simple<sup>5</sup>, a vehicle can have its own values of deceleration for what can be a normal brake **a<sub>b</sub>** and an emergency brake **a<sub>e</sub>**.

So, for a given car, the stopping distance to avoid a crash can be calculated in real time in function of its initial speed. For example:

$$D_b = \frac{v_i^2}{8} \quad (a_b = -4 \text{ m/s}^2) \quad D_e = \frac{v_i^2}{16} \quad (a_e = -8 \text{ m/s}^2)$$

Consequently, if both, the instantaneous distance to an obstacle **d** and the car speed **v** can be measured with sensors installed in the car, a couple of signals for activating normal **b** or emergency **e** braking systems can be implemented by an electronic circuit.

2. The architecture of the electronic braking system

The block diagram in Fig. 3 illustrates for academic purposes a very simple approximation of how to solve engineering problems by means of digital circuitry. A digital 5 X 5 combinational multiplier computes the squared velocity and the dividers obtain the safety distances which will be compared to the measured distance to the obstacle. If d = '0', represents no obstacle detected. In this example, a 5 bit vector for the car velocity (0 to 31 m/s) and a 7 bit vector for the distance (1 to 127 m) have been used.

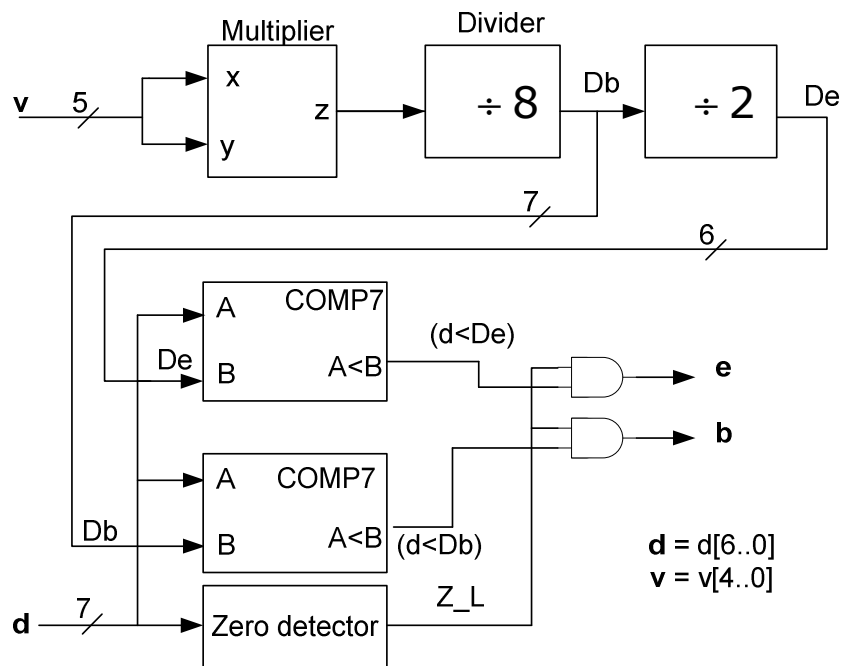


Fig. 3 The block diagram to calculate the braking signals

3. The implementation of the 5 × 5 bit combinational multiplier by means a network of 25 primitive 1-bit multiplier cells as the one shown in Fig. 4, can be found in page 304 of the book: Introduction to Digital Systems<sup>6</sup>, by M. D. Ercegovac, T. Lang and J. H. Moreno, John

<sup>5</sup> Indeed, it's far more complicated and a subject of never-ending research. You can gather more information about the subject of "accident avoidance systems" here: <http://www.can-cia.org/applications/passengercars/invehicle.html> ; and: <http://www.carpages.co.uk/lexus/lexus-ls460-29-08-06.asp>

<sup>6</sup> [http://www.cs.ucla.edu/Logic\\_Design/](http://www.cs.ucla.edu/Logic_Design/)

Wiley and Sons, 1999. Look at the implementation of the multiplier example (6×8) in Fig. 10.18: [http://www.cs.ucla.edu/Logic\\_Design/SLPDF/ch10.pdf](http://www.cs.ucla.edu/Logic_Design/SLPDF/ch10.pdf). As a result of the work carried out the last ED semester, you can investigate and simulate a 5 × 5 bit multiplier in: [http://epsc.upc.edu/projectes/ed/grups\\_classe/07-08-q1/1BT4/EX/multiplier.DSN](http://epsc.upc.edu/projectes/ed/grups_classe/07-08-q1/1BT4/EX/multiplier.DSN) (see Fig. 4a).

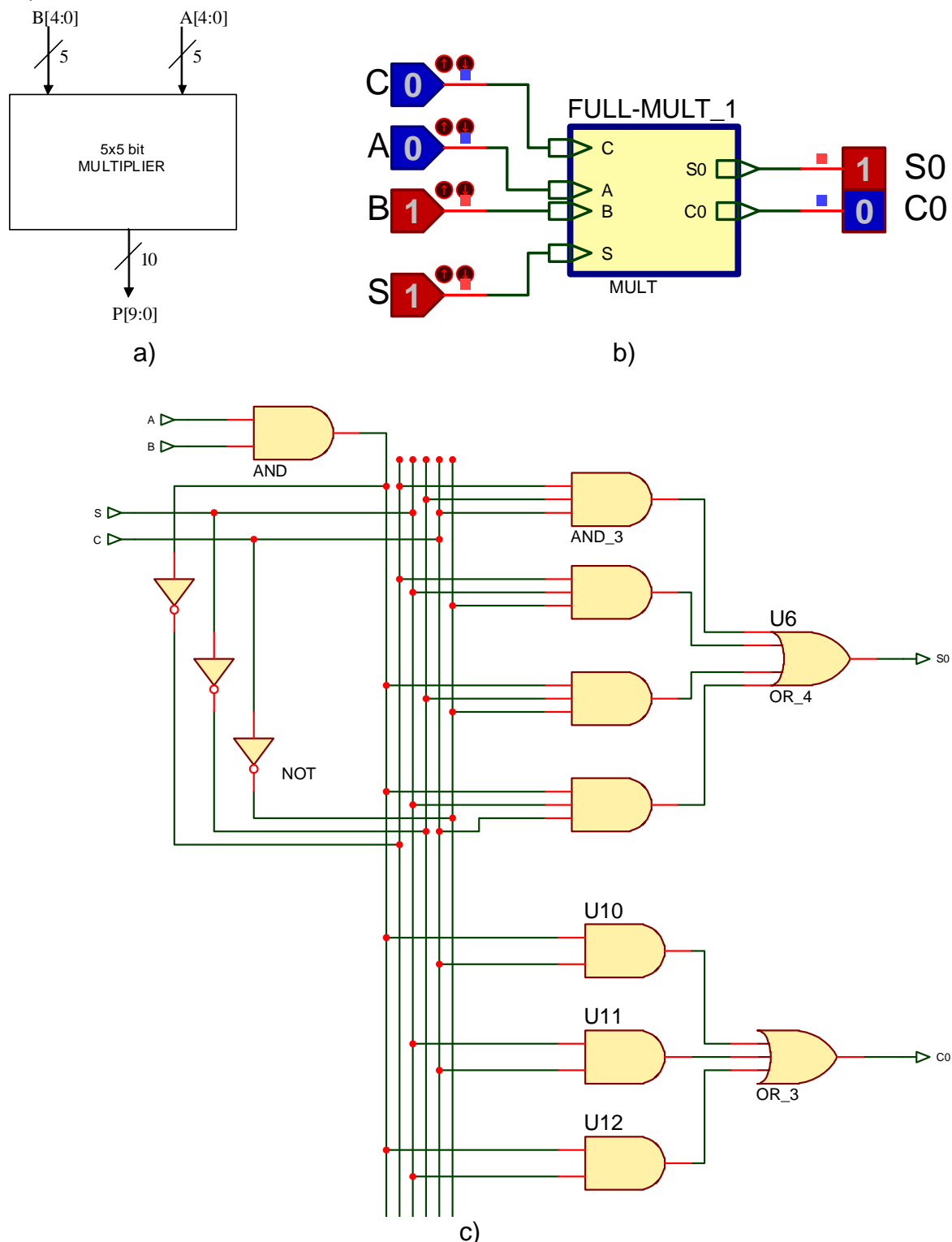


Fig. 4 a) The combinational 5x5 multiplier block; b) a primitive 1-bit multiplier cell; internal combinational block proposed for the 1-bit multiplier cell

a) If the 1-bit full multiplier cell shown in Fig. 4b has the internal circuit represented in Fig. 4c, analyze the network of logic gates and determine the output equations (that means applying the theorems of Boolean algebra to deduce the truth table of the output functions). You may draw a plan and follow the procedure:

1. Sketch using cycles and arrows as in here<sup>7</sup>, the general procedure to obtain and verify the solution
2. From the circuit operate to obtain  $S_0$  (and  $C_0$ ) as an algebraic equation  

$$S_0 = f(S, B, A, C); \quad C_0 = g(S, B, A, C)$$
3. Apply Boolean algebra to obtain a sum of products (SoP)
4. Add, if necessary, the missing input variables to obtain the canonical sum of minterms
5. Build the truth table
6. Verify your answer performing a Proteus-VSM simulation of the circuit in Fig. 4c
7. Capture in Proteus-VSM the circuit obtained in step 2, and simulate it to verify that it produces the same function that the circuit in Fig. 4c.

- b)** Write both output functions as a canonical expression using a product of maxterms
- c)** Simplify  $C_0$  by Karnaugh and the method of zeros and draw the schematics using only NOR
- d)** Simplify  $S_0$  by Karnaugh and the method of ones and draw the schematics using only NAND
- e)** Find the official circuit (or one of the most commons in literature) for a 1-bit multiplier and explain its structure in detail. Explain why XOR's are used.

Designing other blocks for the car's problem:

- f)** Deduce and explain the way the binary dividers in Fig. 3 can be rapidly implemented (concept of shifter). Implement a **div2** and a **div8** block using gates
- g)** Deduce and build the zero detector in Fig. 3
- h)** Verify your circuits in Proteus-VSM

Additional questions: (Unit 1.11: the comparator)

- i)** Deduce and explain how a 7-bit comparator COMP7 can be produced expanding 1-bit comparators COM1 (see Fig. 3). Implement the circuits using logic gates.
- j)** Verify your circuits in Proteus-VSM

<sup>7</sup> [http://epsc.upc.edu/projectes/ed/grups\\_classe/03-04-q2/1bt6/Exemple%20de%20solucio%201a.pdf](http://epsc.upc.edu/projectes/ed/grups_classe/03-04-q2/1bt6/Exemple%20de%20solucio%201a.pdf)

Another design exercise: **a digital wind direction meter**

We want to design a digital wind direction meter as seen in Fig. 5 based on an optoelectronic rotary encoder of 32 positions (5 bits). The sensor disk, as shown in Fig. 6, is coded in Gray<sup>8</sup>, which was originally used instead of binary code to prevent spurious output from electromechanical switches.

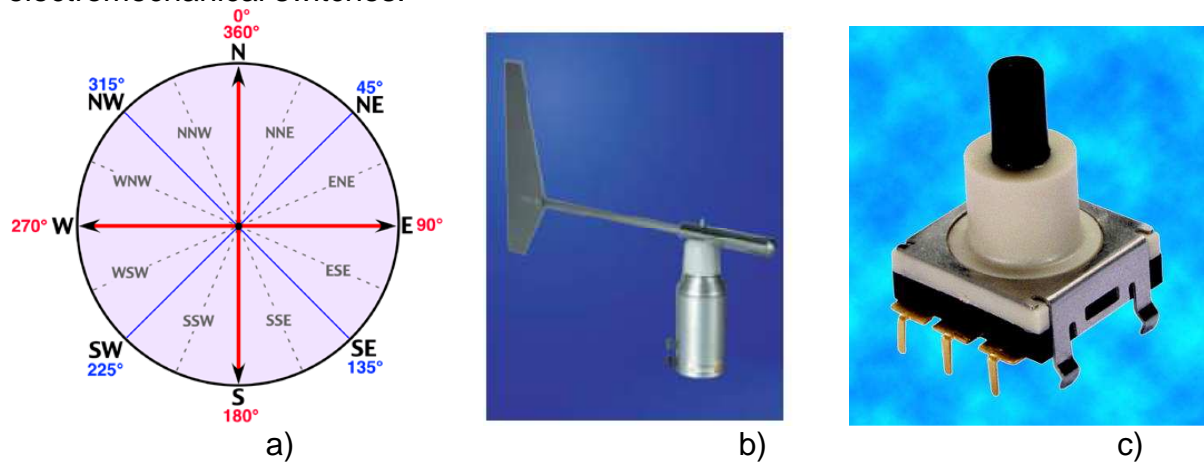


Fig. 5a) Wind compass describing the sixteen principal bearings used to measure wind, b) the wind wave and c) the rotary sensor to be designed

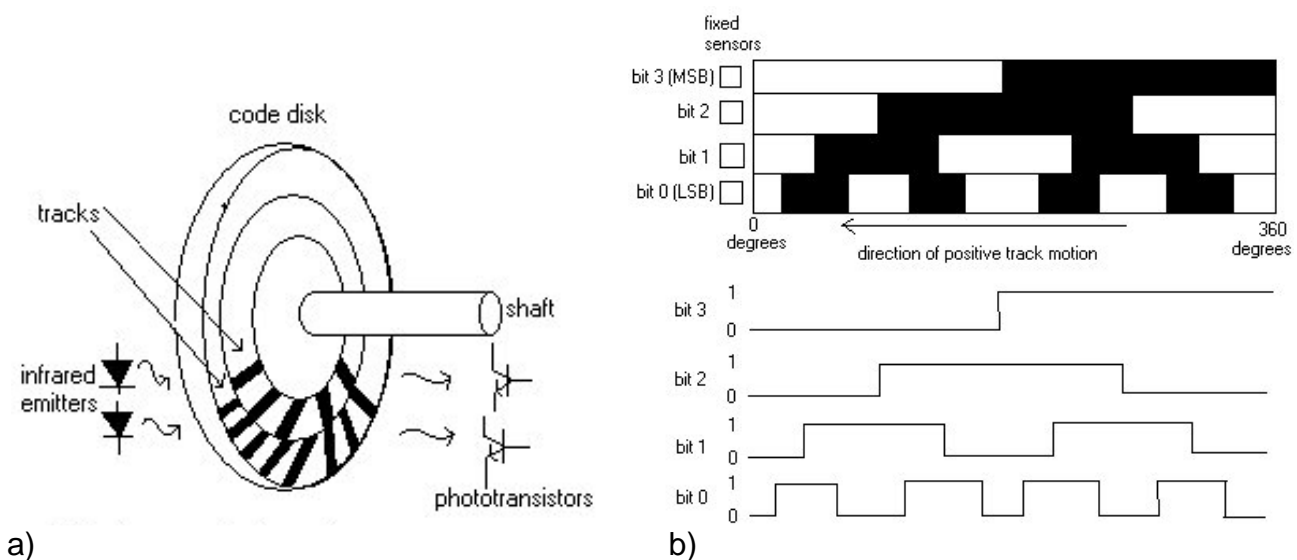


Fig. 6 a) A rotary optical encoder. b) 4-bit Gray code absolute disk track patterns

1. Design a block diagram for the wind direction meter if we want to have both, a 2-digit BCD output and a 1-bit coded output to light a circle of 32 LED to display wind direction.
2. Write the truth table for each block and express each output in terms of a sum of minterms or a product of maxterms
3. Design the code converters minimizing the output logic functions, for example using 5-variables Karnaugh maps for the Gray-to-BCD converter, and implementing the logic circuits using only NAND or only NOR gates
4. Capture your schematics in Proteus-VSM<sup>9</sup> and verify your solution.

<sup>8</sup> [http://mechatronics.mech.northwestern.edu/design\\_ref/sensors/encoders.html](http://mechatronics.mech.northwestern.edu/design_ref/sensors/encoders.html), and <http://www.novalynx.com/>

<sup>9</sup> Use the initial schematic available in the web produced by Helena Vilà in 06-07-Q1 – 1BT4 – MI-4



# DIGITAL ELECTRONICS

G\_\_\_\_\_

## Working plan<sup>10</sup> for solving the exercise EX\_\_\_\_\_

Explain succinctly how the cooperative group has organized the realization of the exercise: i.e., which has been your working plan; in which way has you divided the task fairly so that more or less all of you are doing a similar amount of work; how have you learned each other's materials; what has been worked out in class time (sessions A and B) and what has been resolved in sessions C; and so on... write down also your impressions or opinions about the subject and how your group work is going<sup>11</sup> ...

-----  
Active members' signatures

<sup>10</sup> This document, filled before delivering the exercise, will be included in the group learning portfolio  
<sup>11</sup> Check similar documents in [http://epsc.upc.edu/projectes/ed/unitats/ED\\_05-06\\_Q1\\_Autoavaluacio\\_Grup\\_Base.pdf](http://epsc.upc.edu/projectes/ed/unitats/ED_05-06_Q1_Autoavaluacio_Grup_Base.pdf),  
and [http://epsc.upc.edu/projectes/ed/unitats/que\\_va\\_malament\\_al\\_grup.pdf](http://epsc.upc.edu/projectes/ed/unitats/que_va_malament_al_grup.pdf)