

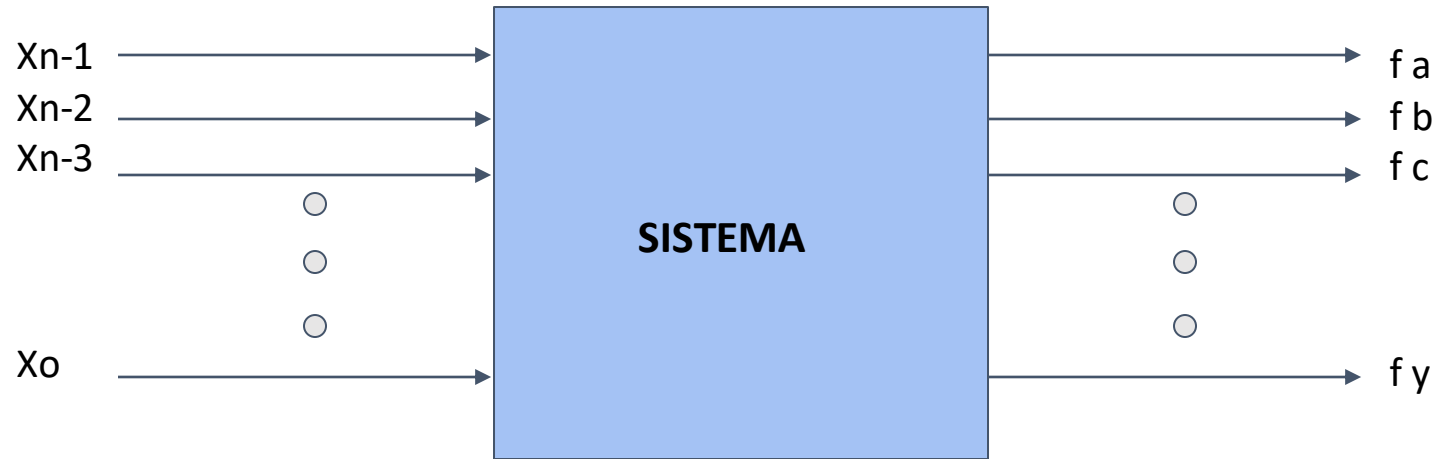
CLASE SABADO 2 DE MARZO 2024

DISEÑO DIGITAL

DISEÑO DIGITAL MODERNO

PROF: ING. ROBERTO MANDUJANO WILD

## II. Minimización de sistemas con múltiples funciones o salidas



**NOTA: ¿CUÁNTAS FUNCIONES BOOLEANAS PUEDE TENER EN UN ESPACIO BOOLEANO DE N VARIABLES?**

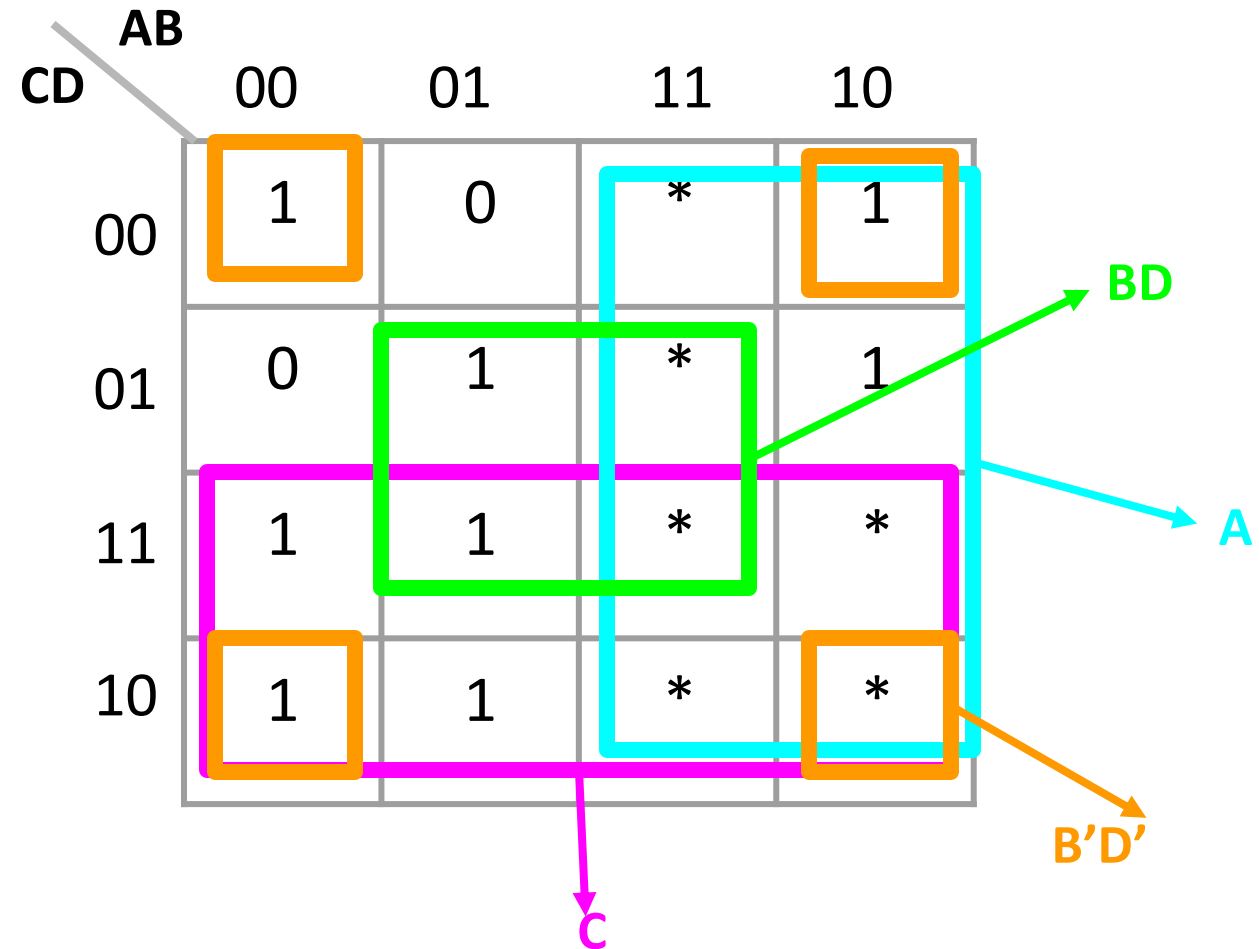
$$2^{2^n}$$

## Ejemplo

Minimizar las funciones booleanas del código de 7 segmentos.

A	B	C	D	f1	f2	f3	f4	f5	f6	f7
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

Para  $f_1$



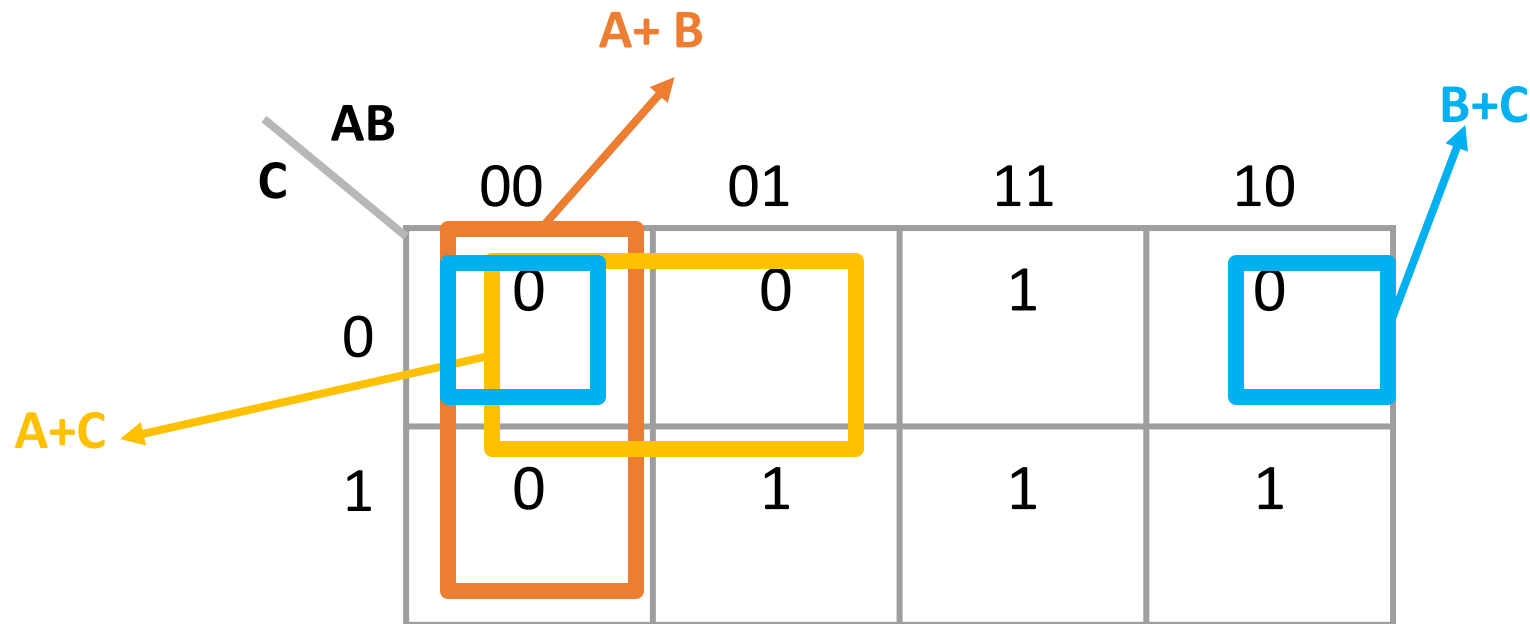
$$f_1 = C + B'D' + A + BD$$

### III. Minimización en producto de sumas utilizando mapas de Karnaugh

Para obtener la minimización en producto de sumas tenemos que agrupar los 0's de la misma forma que se agrupan los 1's en las funciones de suma de productos, solo que el dominio de cada grupo es la suma de las variables complementadas.

#### Ejemplo

Dada la siguiente función booleana dada en un mapa de Karnaugh, minimizar en producto de sumas.



$$f_{ps} = (A+B) (B+C) (A+C)$$

## Ejercicio

De la función anterior, obtener:

- a) La función en mintérminos
- b) La función negada en mintérminos
- c) La función en maxtérminos
- d) La función negada en maxtérminos
- e) La función minimizada en suma de productos
- f) La función negada minimizada en suma de productos
- g) La función minimizada en producto de sumas
- h) La función negada minimizada en producto de sumas

$$f_m = \sum m(3,5,6,7)$$

$$f'_m = \sum m(0,1,2,4)$$

$$f_M = \prod M(0,1,2,4)$$

$$f'_M = \prod M(3,5,6,7)$$

$$f_{sp} = AB + BC + AC$$

$$f'_{sp} = A'B' + B'C' + A'C'$$

$$f_{ps} = (A+B)(B+C)(A+C)$$

$$f'_{ps} = (A'+B')(B'+C')(A'+C')$$

<b>AB</b> <b>C</b>					
		00	01	11	10
0	0	0	0	1	0
1	0	1	1	1	1

## Ejemplo 2

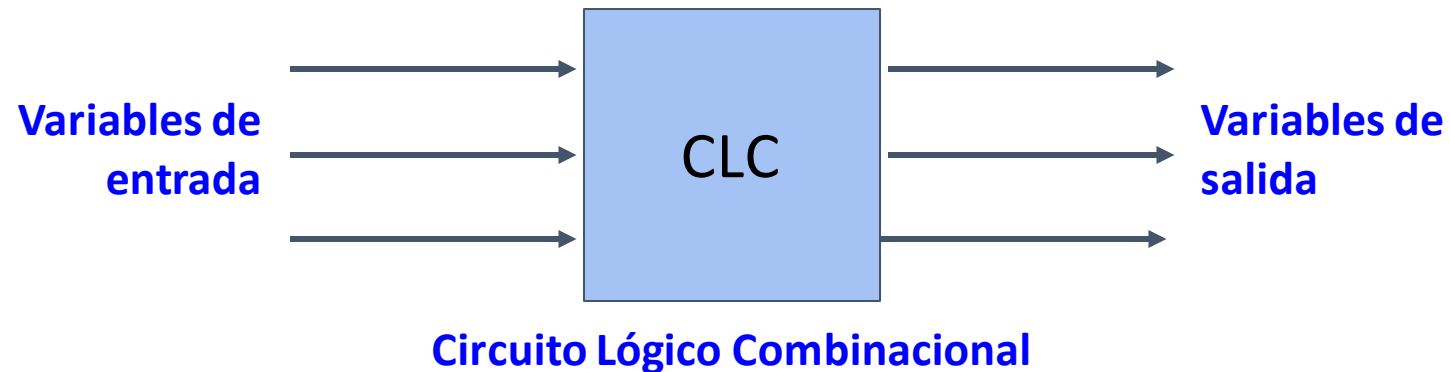
Minimizar la siguiente función booleana.

CD \ AB				
	00	01	11	10
00	0	1	1	1
01	1	1	1	1
11	1	1	1	1
10	0	1	1	1

# TEMA V. CIRCUITOS COMBINACIONALES

## 1). Introducción

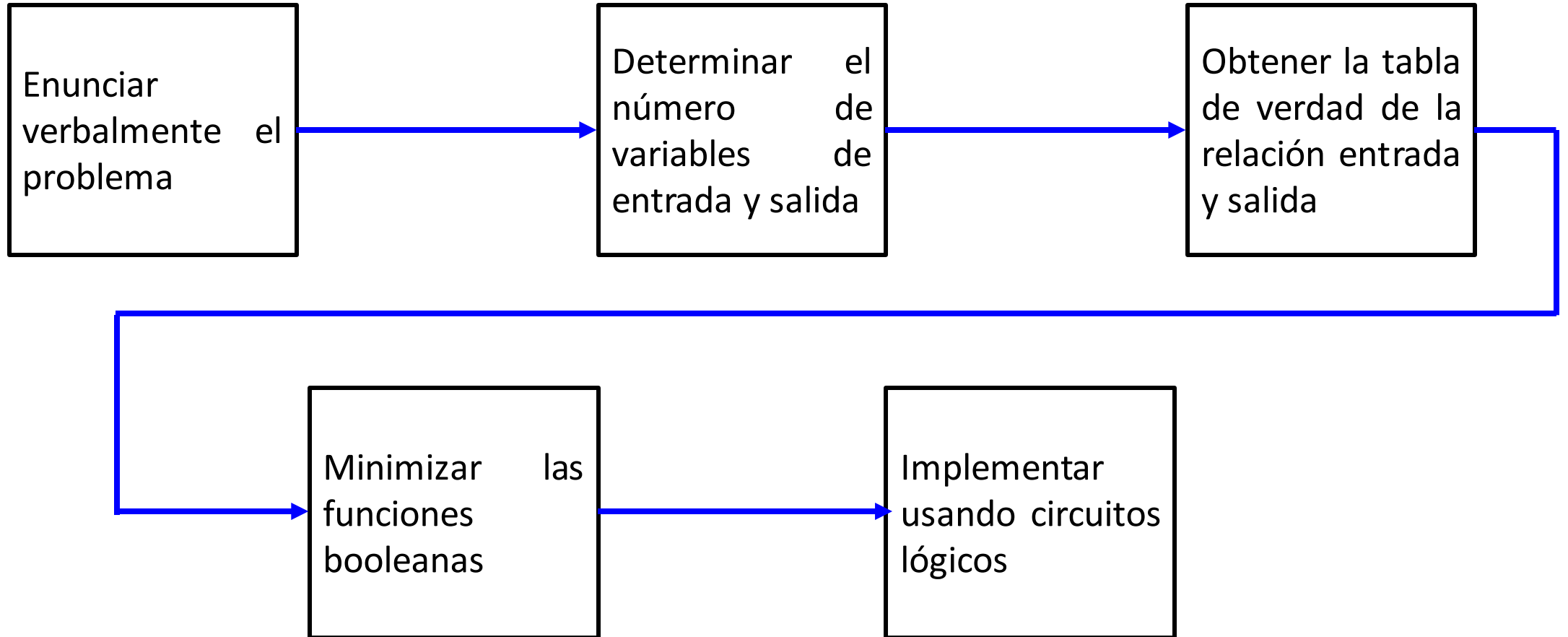
a) Definición: Un circuito combinacional es aquel que utiliza compuertas lógicas para su implementación y su salida o salidas están en función de las variables de entrada y del arreglo de las compuertas.



**OJO: Todo circuito lógico combinacional puede ser representado por su tabla de verdad**

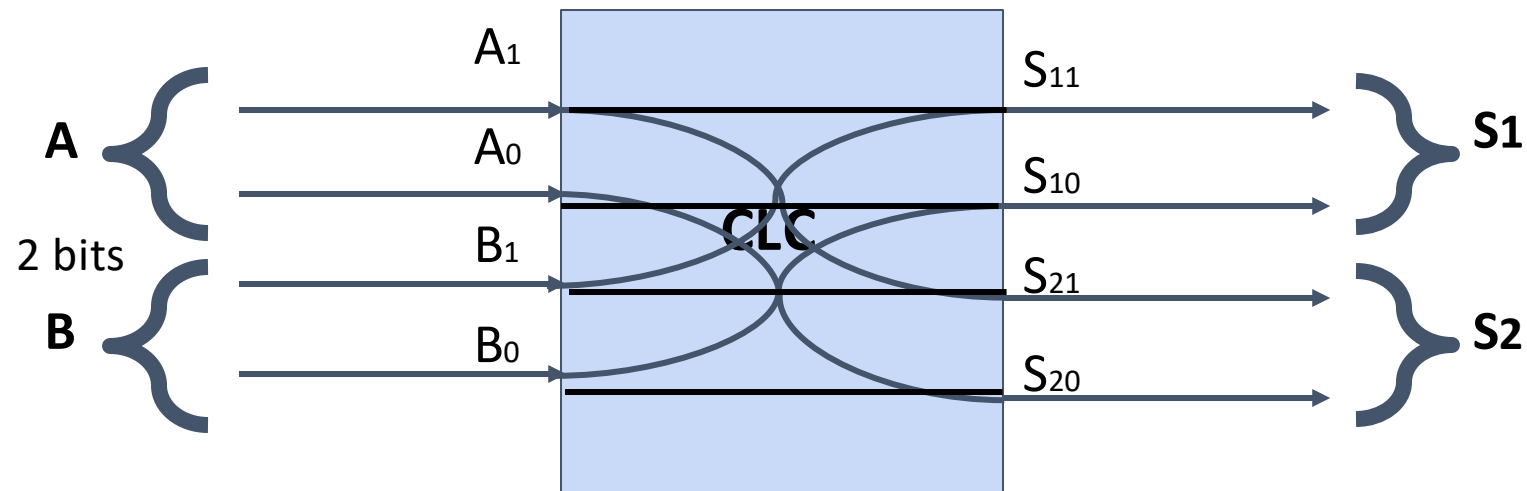


b). Pasos de diseño



## Ejemplo

Diseñar circuito combinacional (CLC) que compare dos señales A y B de 2 bits cada una y siempre que  $A > B$ , la información de A se obtendrá por la salida 2 y la de B en la salida 1, en caso contrario la información de A estará en salida 1 y la de B en salida 2.



A1	A0	B1	B0	S11	S10	S21	S20
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	0	0	1
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	0	0	1	0
1	0	0	1	0	1	1	0
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	1
1	1	0	0	0	0	1	1
1	1	0	1	0	1	1	1
1	1	1	0	1	0	1	1
1	1	1	1	1	1	1	1

Minimizar cada salida.

Para  $S_{11}$

$A_1A_0$ $B_1B_0$		00	01	11	10
00	0	0	0	0	0
01	0	0	0	0	0
11	0	0	1	1	
10	0	0	1	1	

Para  $S_{10}$

$A_1A_0$ $B_1B_0$		00	01	11	10
00	0	0	0	0	0
01	0	1	1	1	
11	0	1	1	0	
10	0	1	0	0	

Para  $S_{21}$

$A_1A_0$					
$B_1B_0$		00	01	11	10
00	0	0	1	1	
01	0	0	1	1	
11	1	1	1	1	
10	1	1	1	1	

Para  $S_{20}$

$A_1A_0$					
$B_1B_0$		00	01	11	10
00	0	1	1	1	0
01	1	1	1	1	0
11	1	1	1	1	1
10	0	0	1	1	0

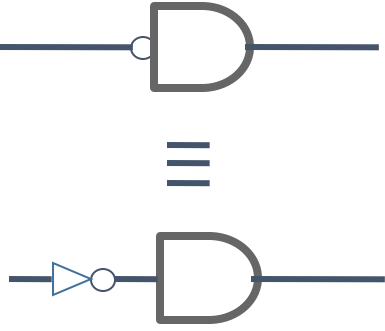
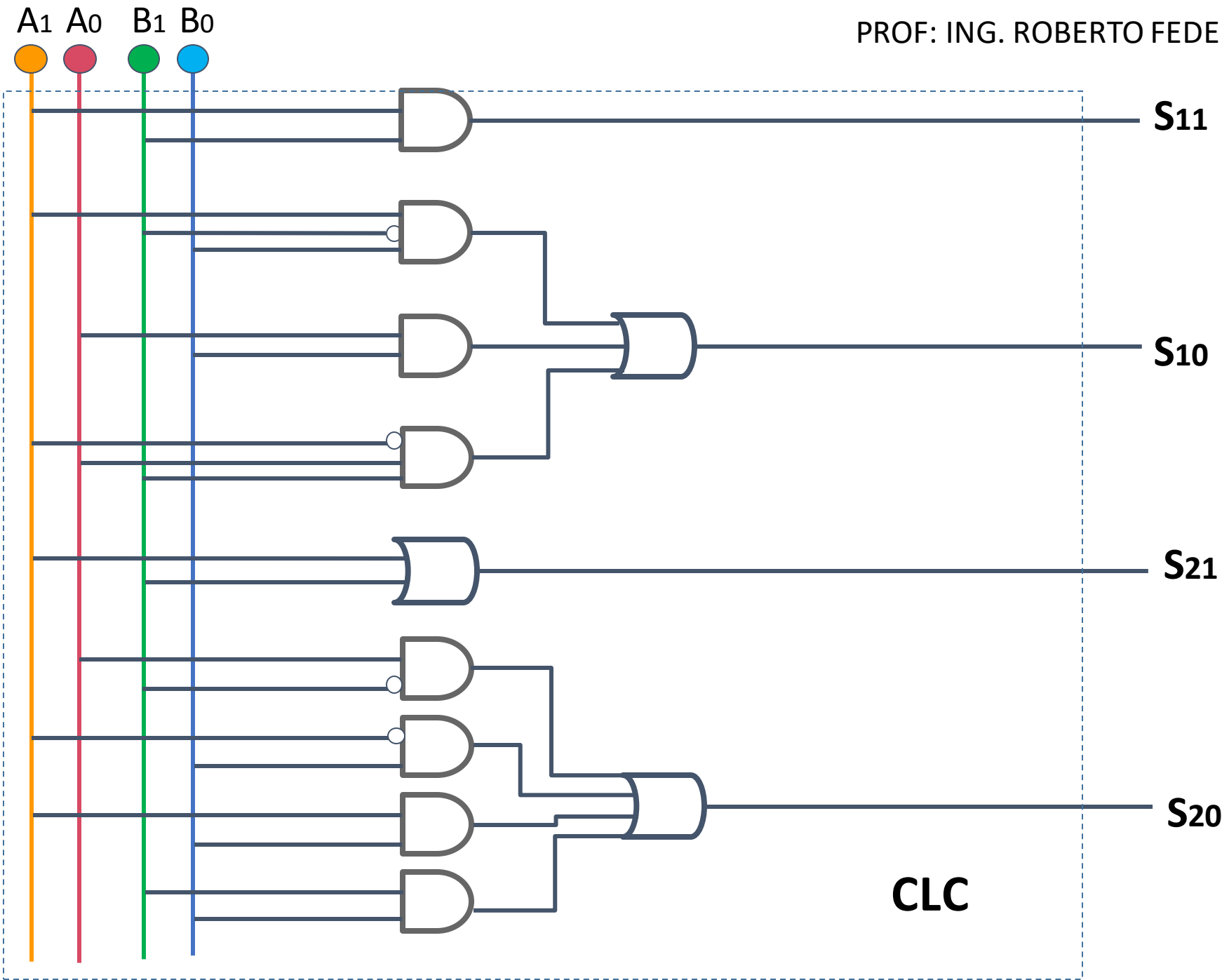
A1	A0	B1	B0	S11	S10	S21	S20
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	0	0	1
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	0	0	1	0
1	0	0	1	0	1	1	0
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	1
1	1	0	0	0	0	1	1
1	1	0	1	0	1	1	1
1	1	1	0	1	0	1	1
1	1	1	1	1	1	1	1

$$S_{11}=A_1B_1$$

$$S_{10}=A_0B_0+A_1B'_1B_0+A'_1A_0B_1$$

$$S_{21}=A_1+B_1$$

$$S_{20}=A_0B'_1+A'_1B_0+A_1A_0+B_1B_0$$



$$S_{11} = A_1 B_1$$
$$S_{10} = A_0 B_0 + A_1 B'_1 B_0 + A'_1 A_0 B_1$$
$$S_{21} = A_1 + B_1$$
$$S_{20} = A_0 B'_1 + A'_1 B_0 + A_1 A_0 + B_1 B_0$$

**c). Circuitos “incompletamente” especificados**

Los circuitos “incompletamente” especificados son aquellos en los que las funciones booleanas que los describen no están especificados completamente, es decir, que para ciertas combinaciones de las variables de entrada, la salida no importa, puesto que estas combinaciones no son válidas o nunca se presentan.

**Ejemplo**

Determinar la tabla de verdad del dígito decimal de su calificación final para determinar si hay redondeo o no.



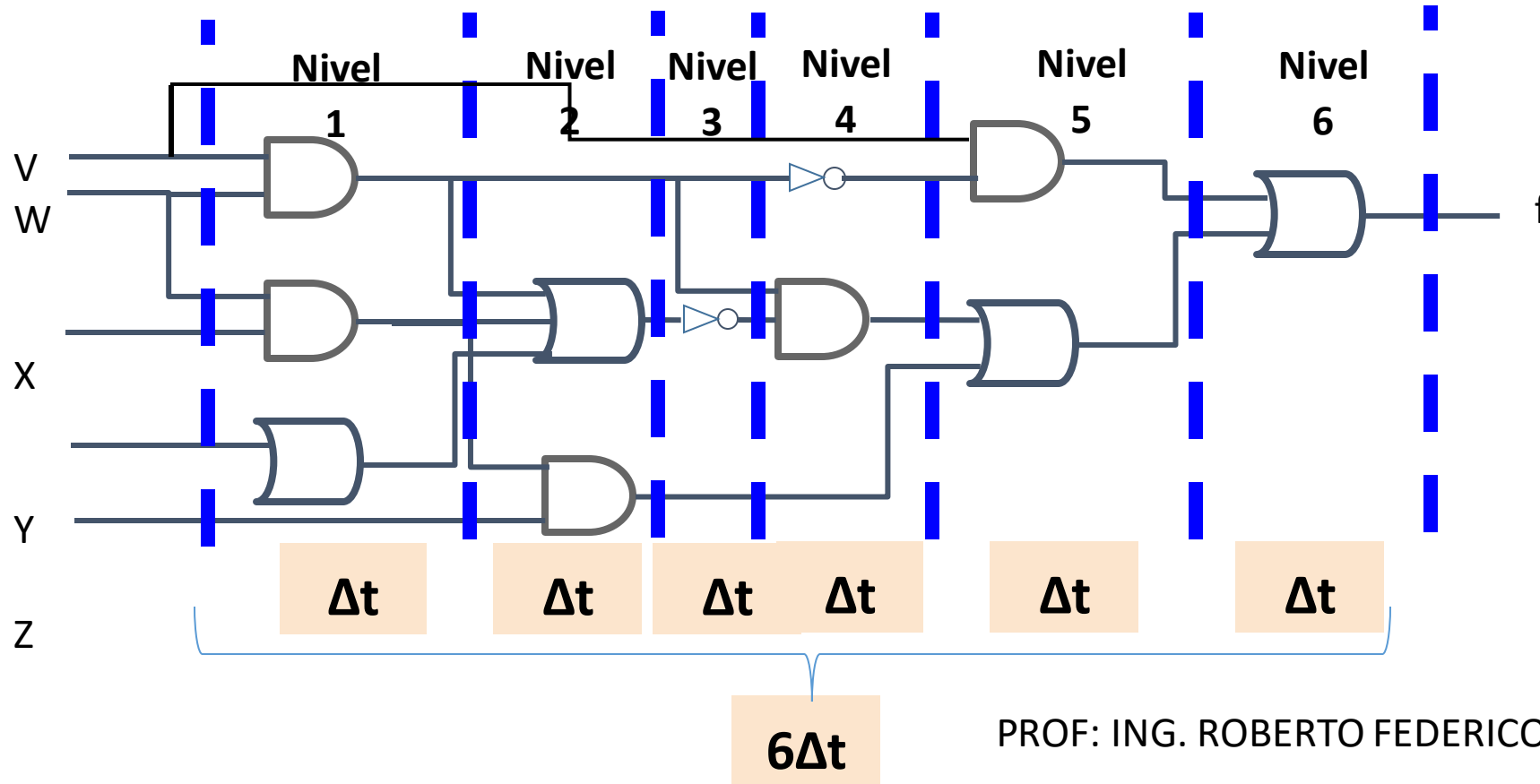
A	B	C	D	fredondeo
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	*
1	0	1	1	*
1	1	0	0	*
1	1	0	1	*
1	1	1	0	*
1	1	1	1	*

Como la información está en BCD solo tengo hasta el 9.

d). **Retrasos en el tiempo**

En los circuitos combinacionales y en general en todos los circuitos lógicos hay retrasos en el tiempo entre la señal de entrada y salida, ese retraso depende de:

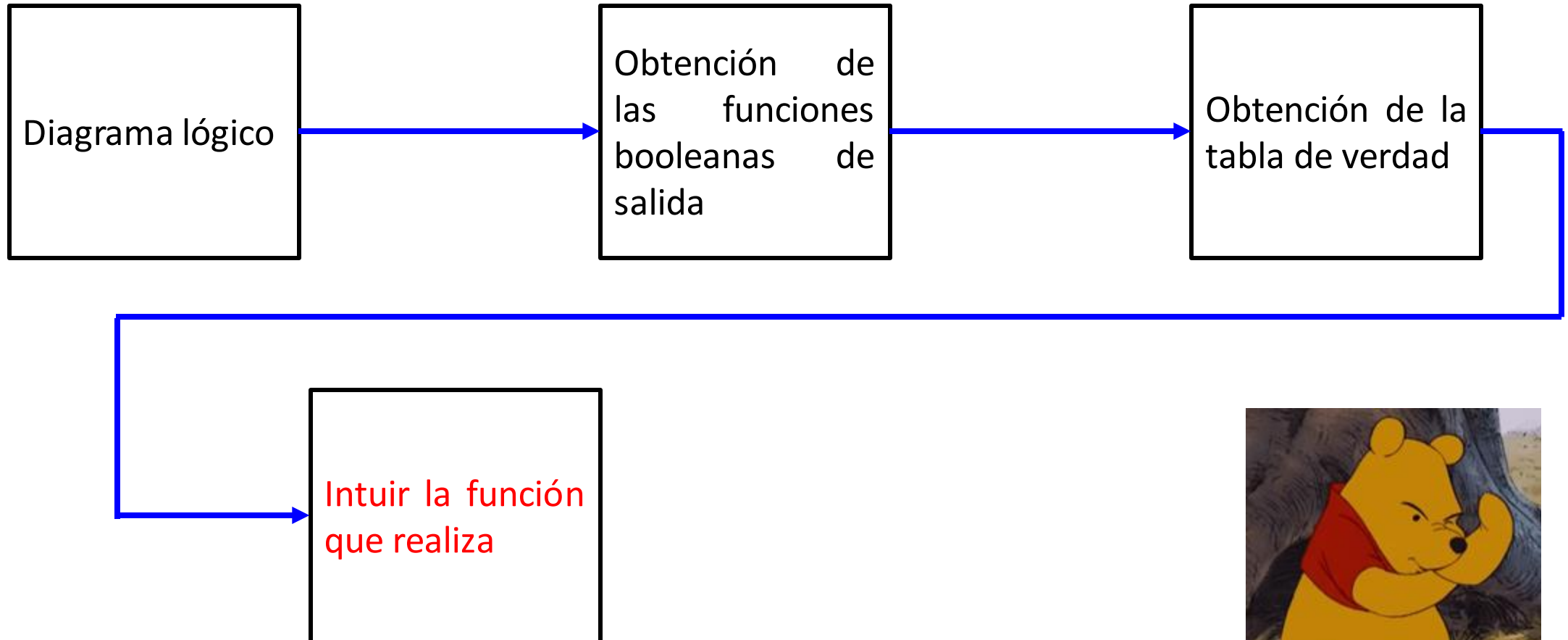
- Niveles de implementación.
- Tipo de compuertas lógicas.
- Tipo de familia con la cuál se está trabajando:



e). **Análisis de circuitos combinacionales**

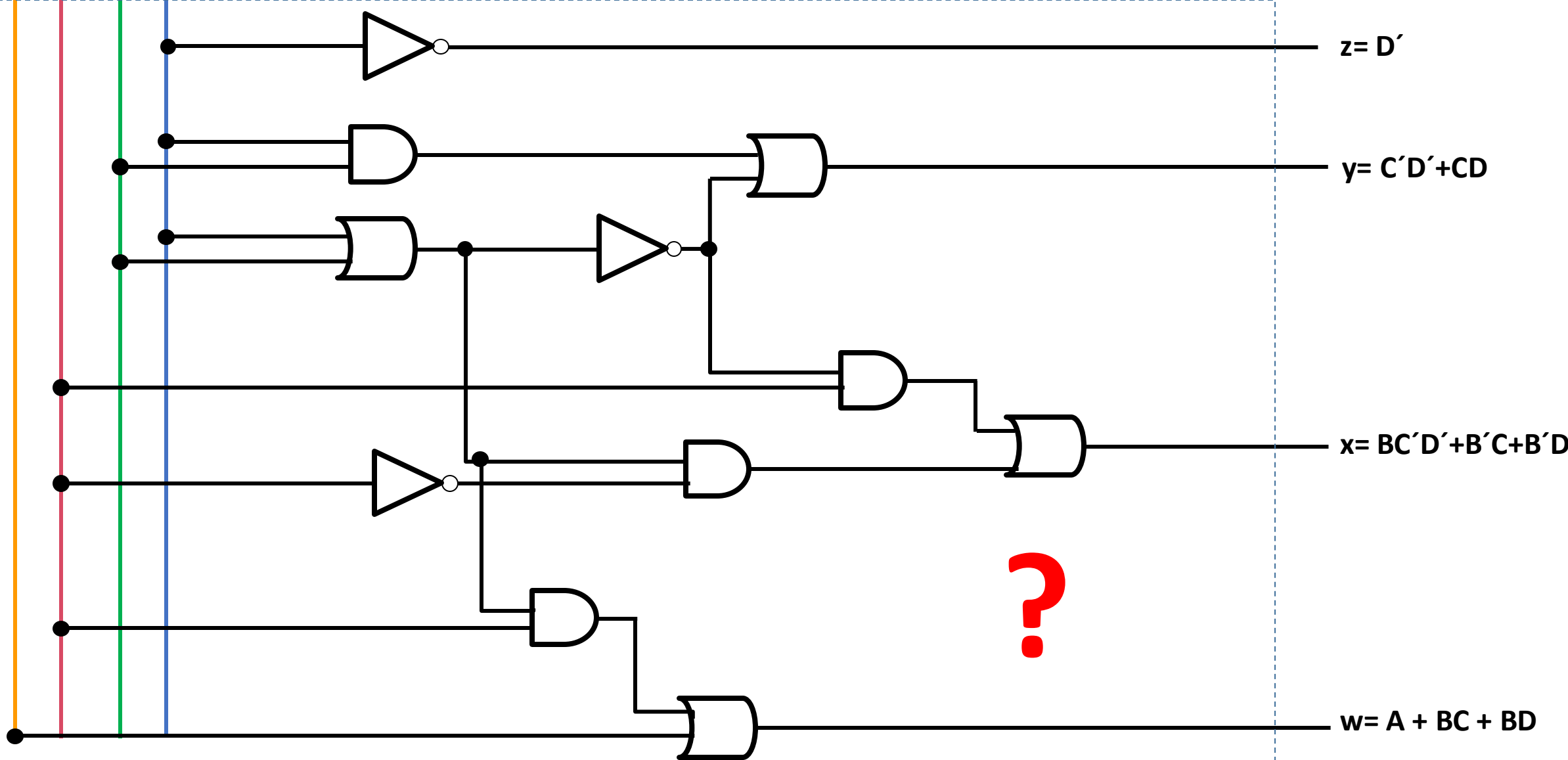
Consiste en dado un diagrama lógico identificar qué función está realizando dicho circuito.

Los pasos son:



Ejemplo 1

A B C D



## Ejemplo 1

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	1	1	1	0
1	1	0	0	1	1	1	1
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	1
1	1	1	1	1	0	1	0

**ES UN CONVERTIDOR  
DE CÓDIGO BCD A  
CÓDIGO EX-3**

**OJO: EL BCD ES DE 0 A 9, POR  
ESO ES CONVERTIDOR A  
EXCESO DE 3, AUNQUE  
DESPUÉS DEL 9 NO ME  
IMPORTA, SON CONDICIONES  
DON'T CARE (\*).**

