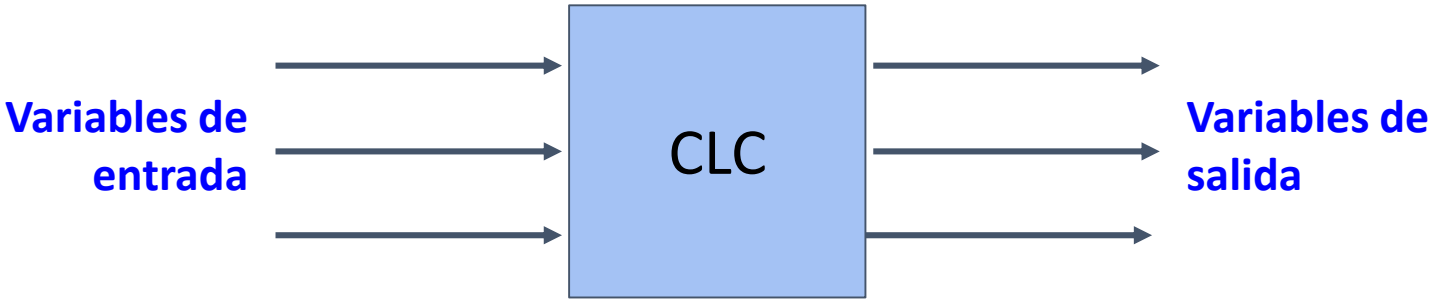
TEMA V. CIRCUITOS COMBINACIONALES

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.

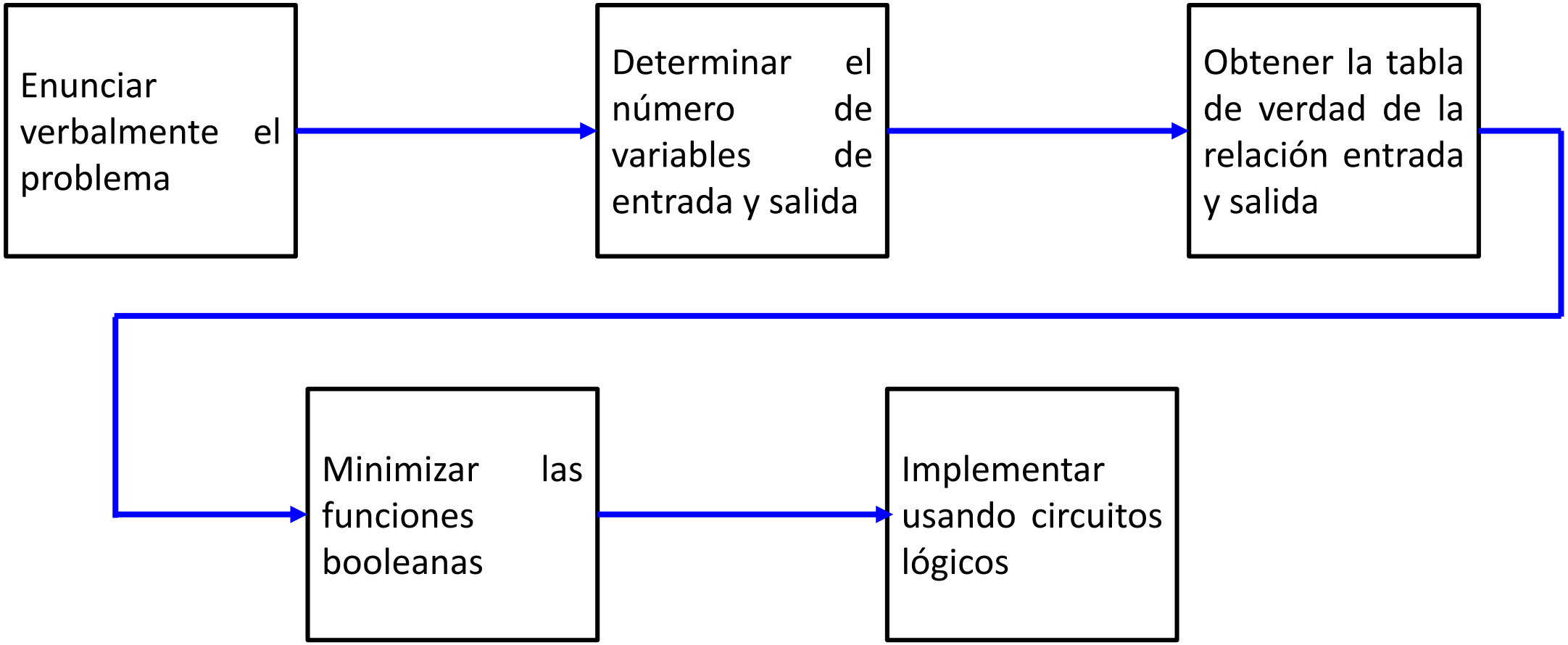


Circuito Lógico Combinacional

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

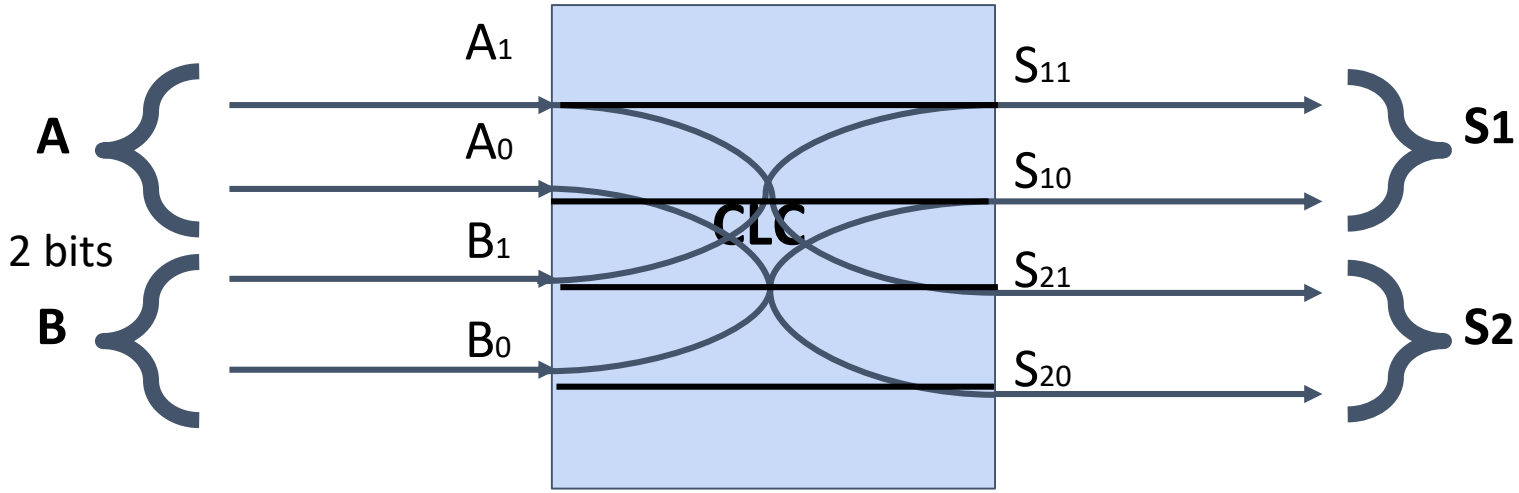
PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

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.

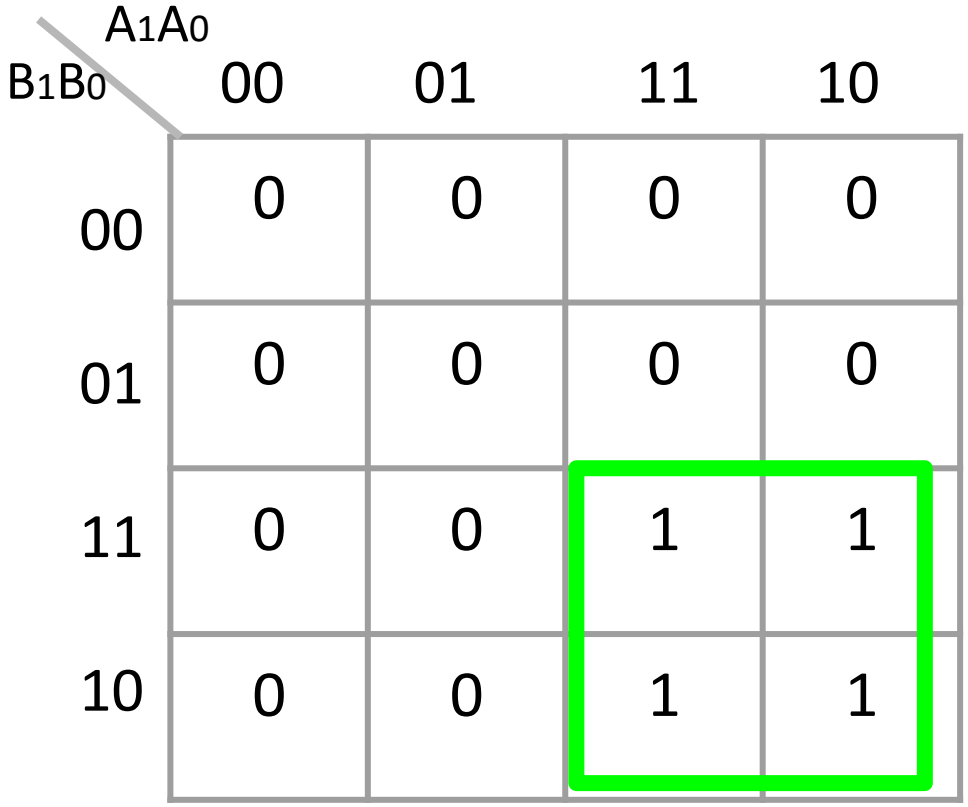


A1	A0	B1	B0	S11	S10	S21	S20
0	**0**	**0**	**0**	0	0	0	0
0	**0**	**0**	**1**	0	0	0	1
0	**0**	**1**	**0**	0	0	1	0
0	**0**	**1**	**1**	0	0	1	1
0	**1**	**0**	**0**	0	0	0	1
0	**1**	**0**	**1**	0	1	0	1
0	**1**	**1**	**0**	0	1	1	0
0	**1**	**1**	**1**	0	1	1	1
1	**0**	**0**	**0**	0	0	1	0
1	**0**	**0**	**1**	0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

PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

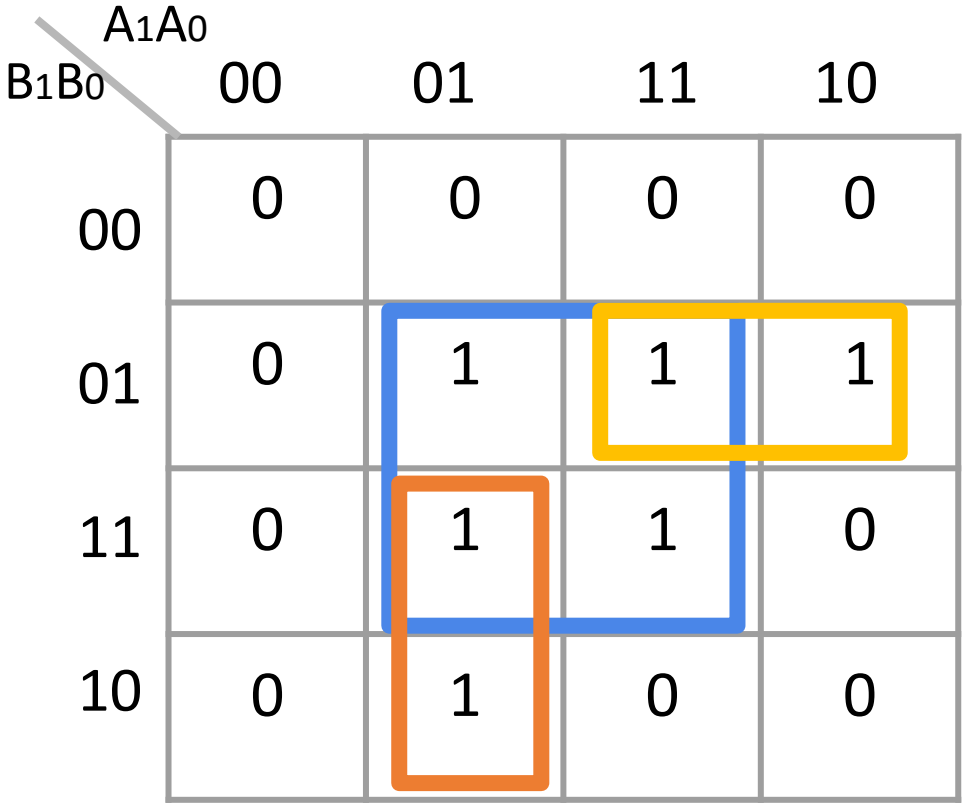
Minimizar cada salida.

Para S_{11}



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

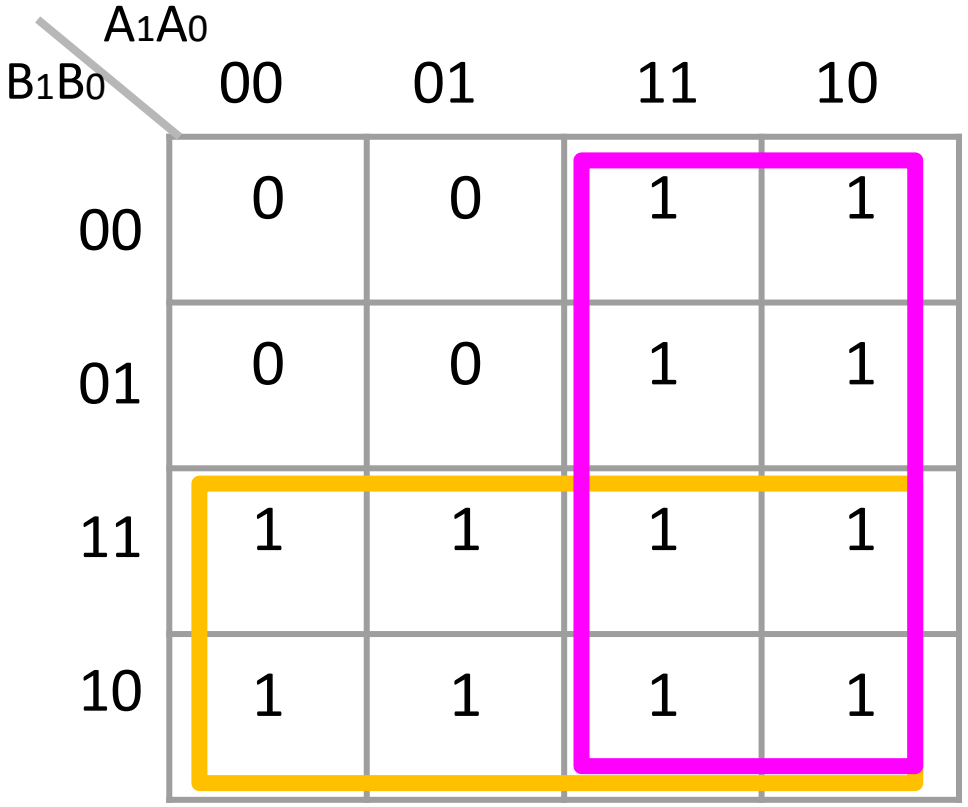
Para S_{10}



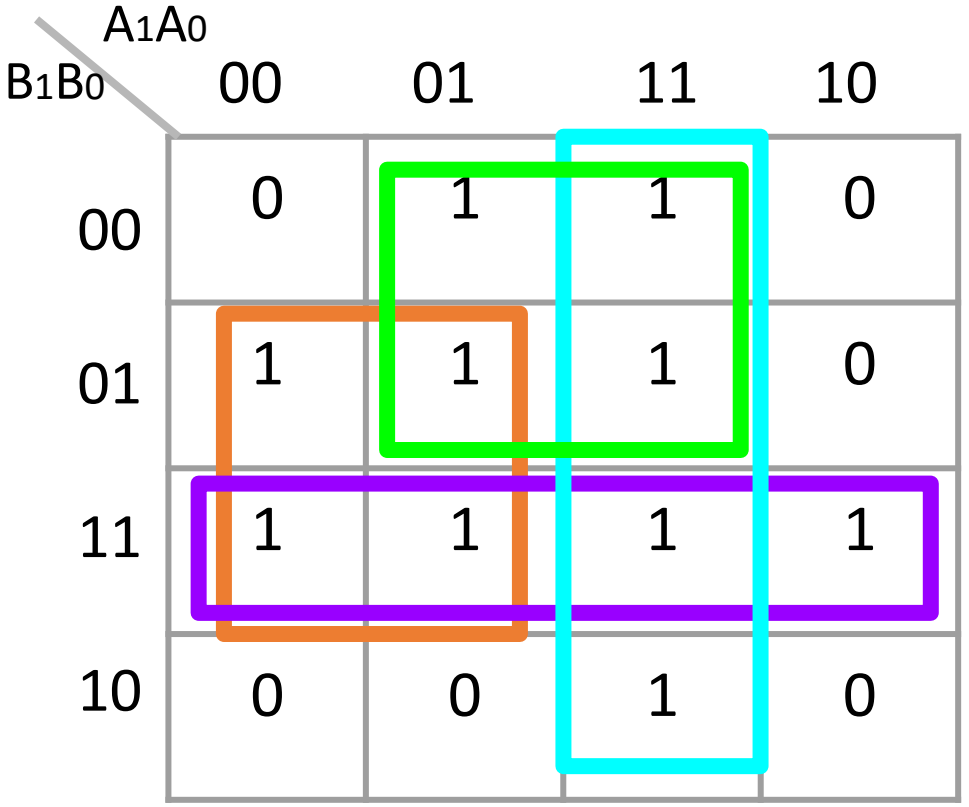
B_1B_0 \ A_1A_0	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	0	1	1	0
10	0	1	0	0

PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

Para S_{21}



Para S_{20}



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

$S_{11}=A_1B_1$

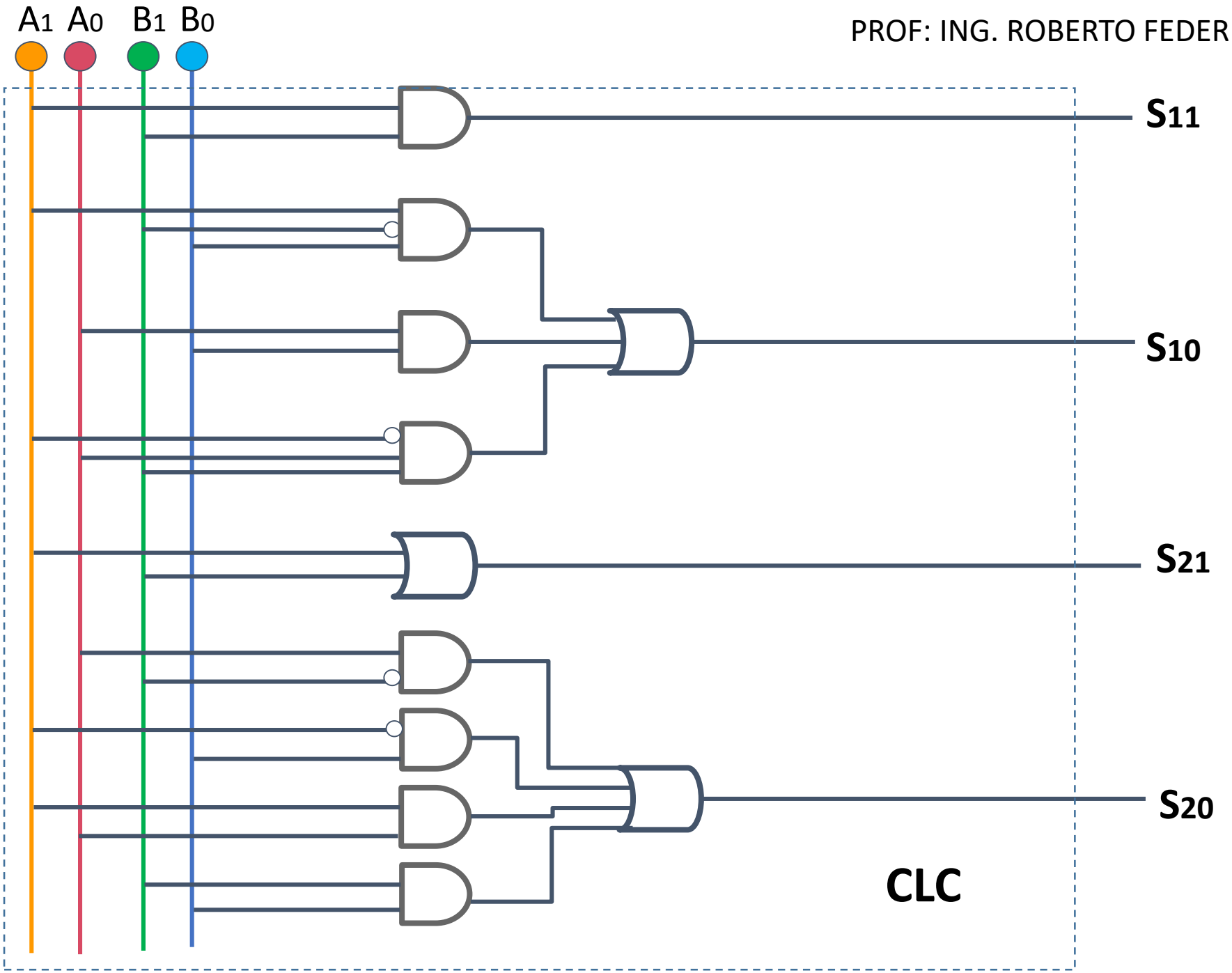
$S_{10}=A_0B_0+A_1B'_1B_0+A'_1A_0B_1$

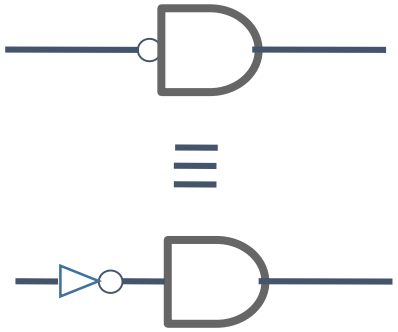
$S_{21}=A_1+B_1$

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

PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

PROF: ING. ROBERTO FEDERICO MANDUJANO WILD





$S_{11}=A_1B_1$

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

$S_{21}=A_1+B_1$

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

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.

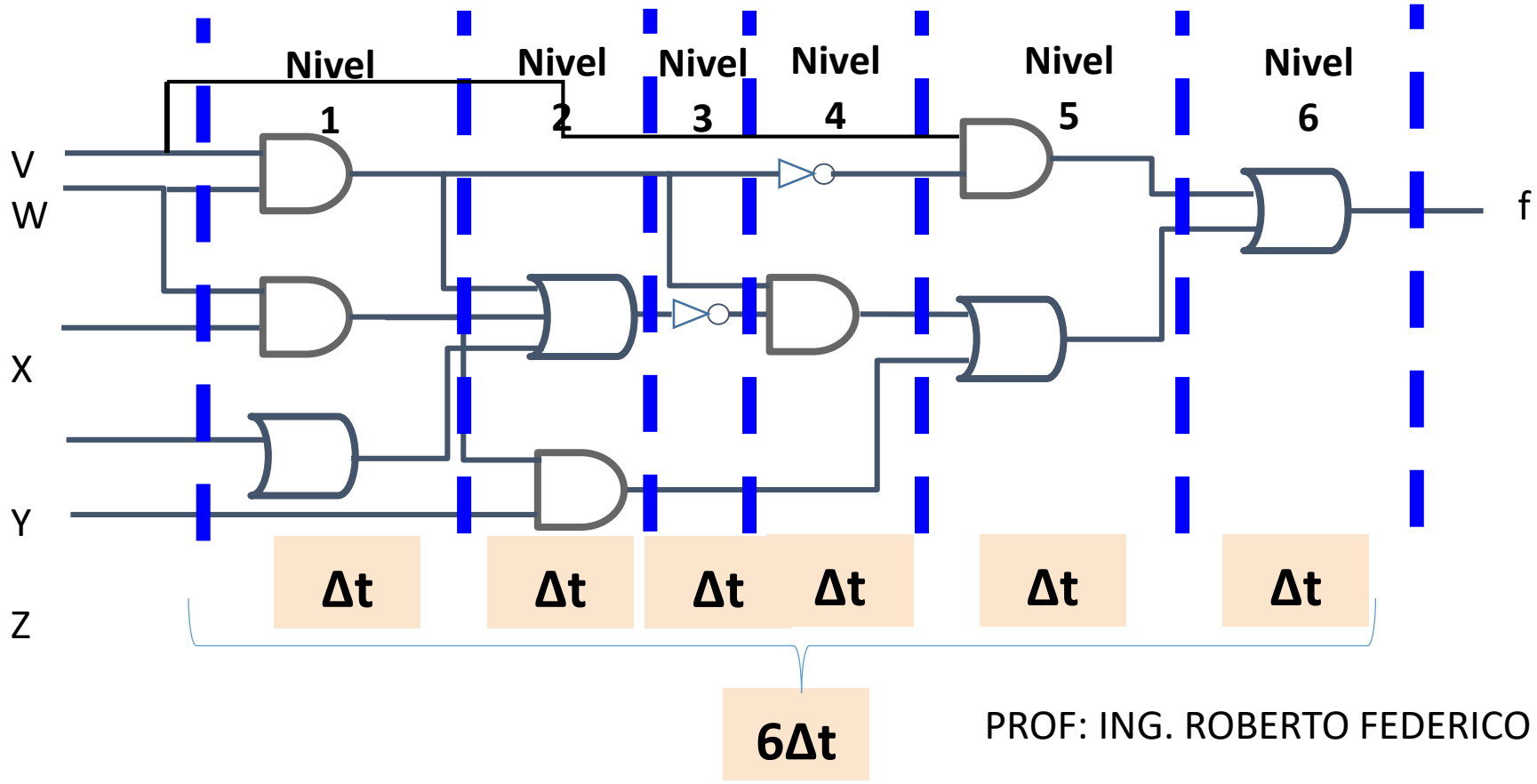
A	B	C	D	fredondeo
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	*
1	0	1	1	*
1	1	0	0	*
1	1	0	1	*
1	1	1	0	*
1	1	1	1	*

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:

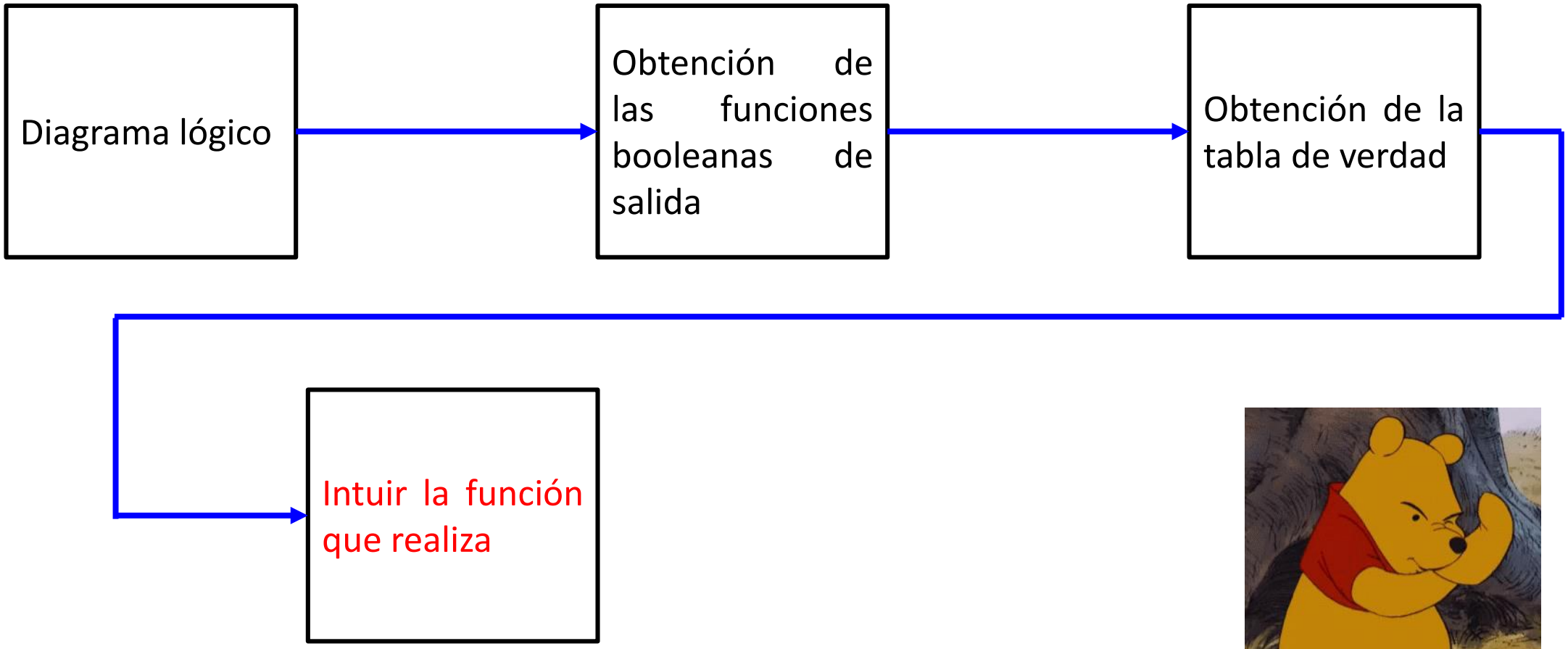
-Niveles de implementación.
-Tipo de compuertas lógicas.
-Tipo de familia con la cuál se está trabajando:



e). Analisis de circuitos combinacionales

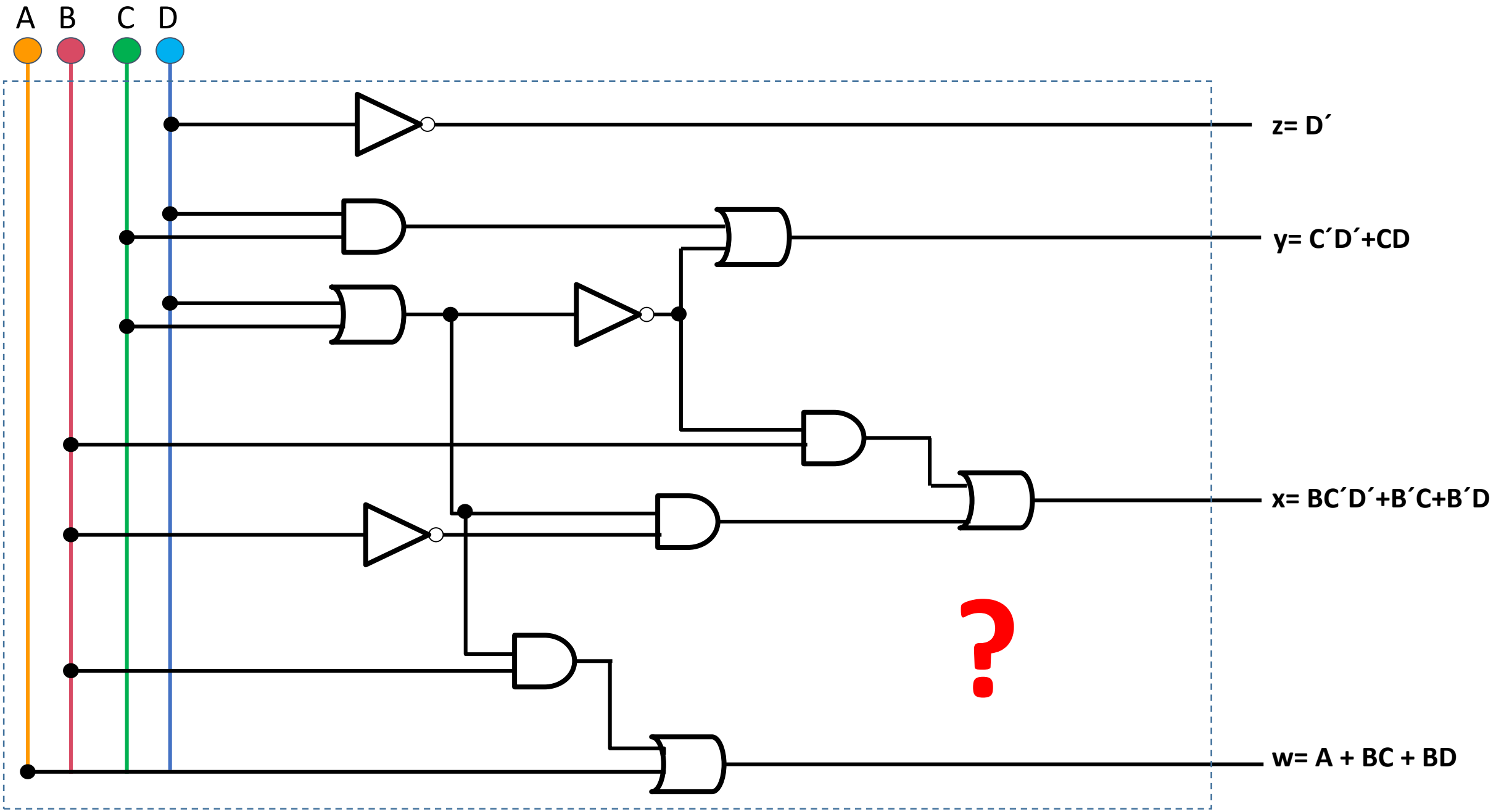
Consiste en dado un diagrama lógico identificar qué función está realizando dicho circuito.

Los pasos son:



PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

Ejemplo 1



Ejemplo 1

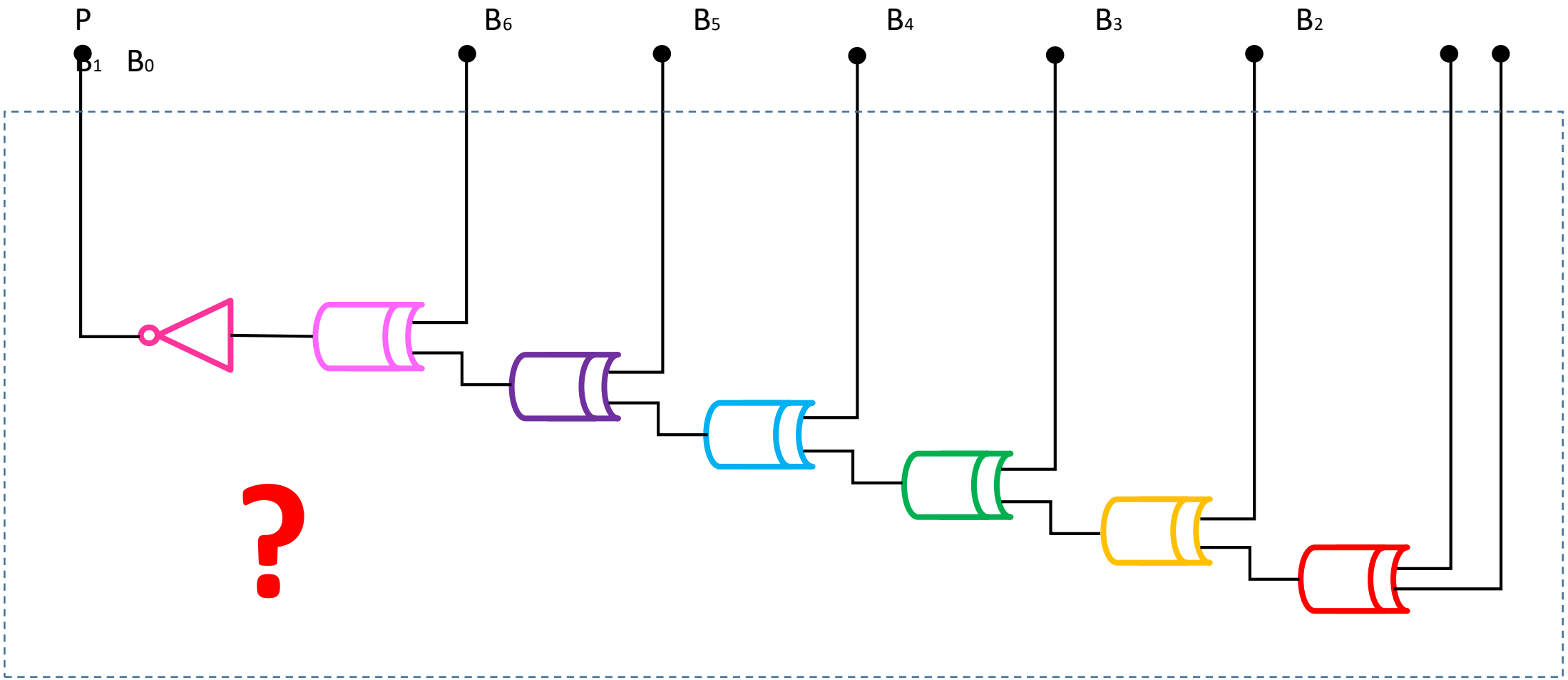
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	1	1	1	0
1	1	0	0	1	1	1	1
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	1
1	1	1	1	1	0	1	0

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 (*).

PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

EJEMPLO 2



PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

EJEMPLO 2

P= (B6⊕B5⊕B4⊕B3⊕B2⊕B1⊕B0)´

B6	B5	B4	B3	B2	B1	B0	P
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0
0	0	0	0	0	1	1	1
….	….	….	….	….	….	….	….
0	1	1	0	1	0	0	0
1	0	1	0	1	0	1	1
1	1	1	1	1	1	1	0

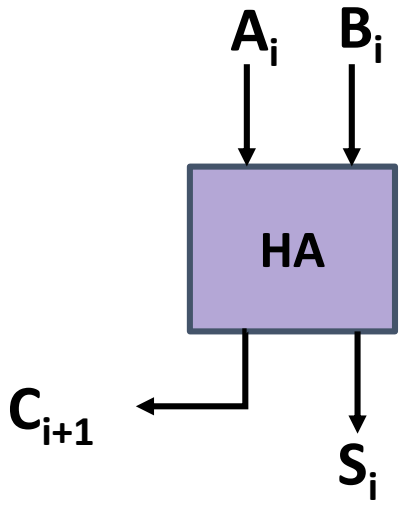
?

PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

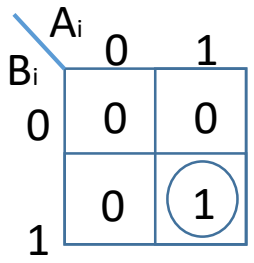
2. Diseño de sumadores

a) Medio sumador (HA)

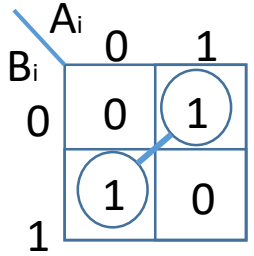
Es aquel que no considera el carry anterior para efectuar la suma de cada pareja de bits.



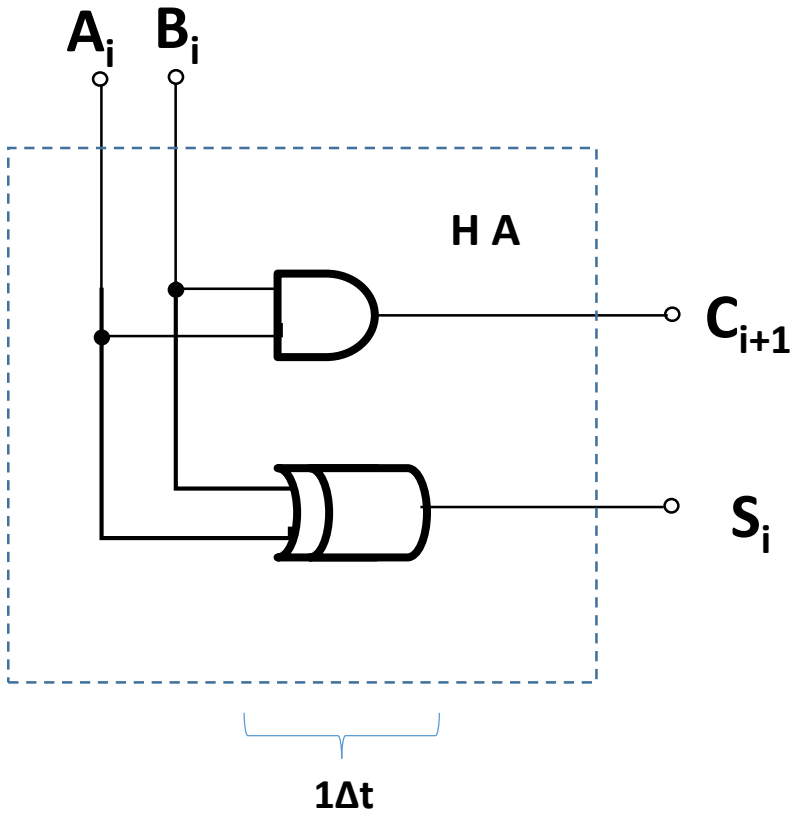
A_i	B_i	C_{i+1}	S_i
0	**0**	0	0
0	**1**	0	1
1	**0**	0	1
1	**1**	1	0



$C_{i+1} = A_iB_i$

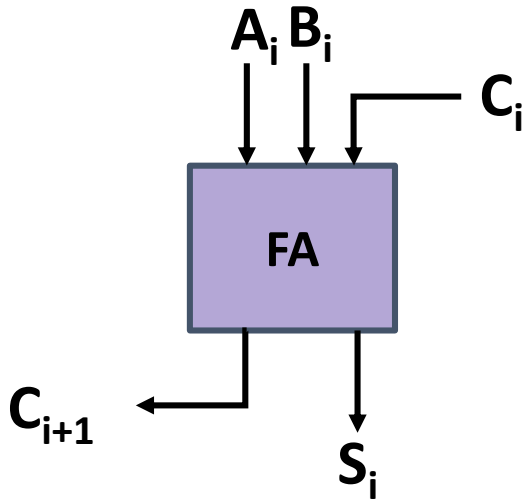


$S_i = A'_iB_i + A_iB'_i = A_i \oplus B_i$

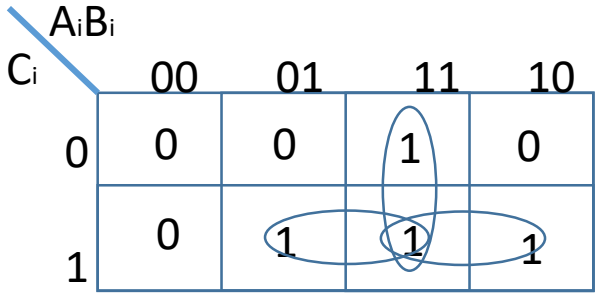


b) Sumador completo (FA)

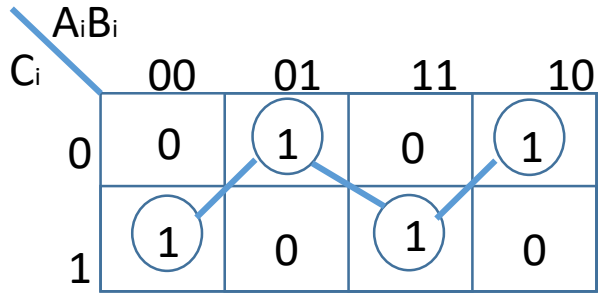
Es aquel que sí considera el carry anterior para efectuar la suma de cada pareja de bits.



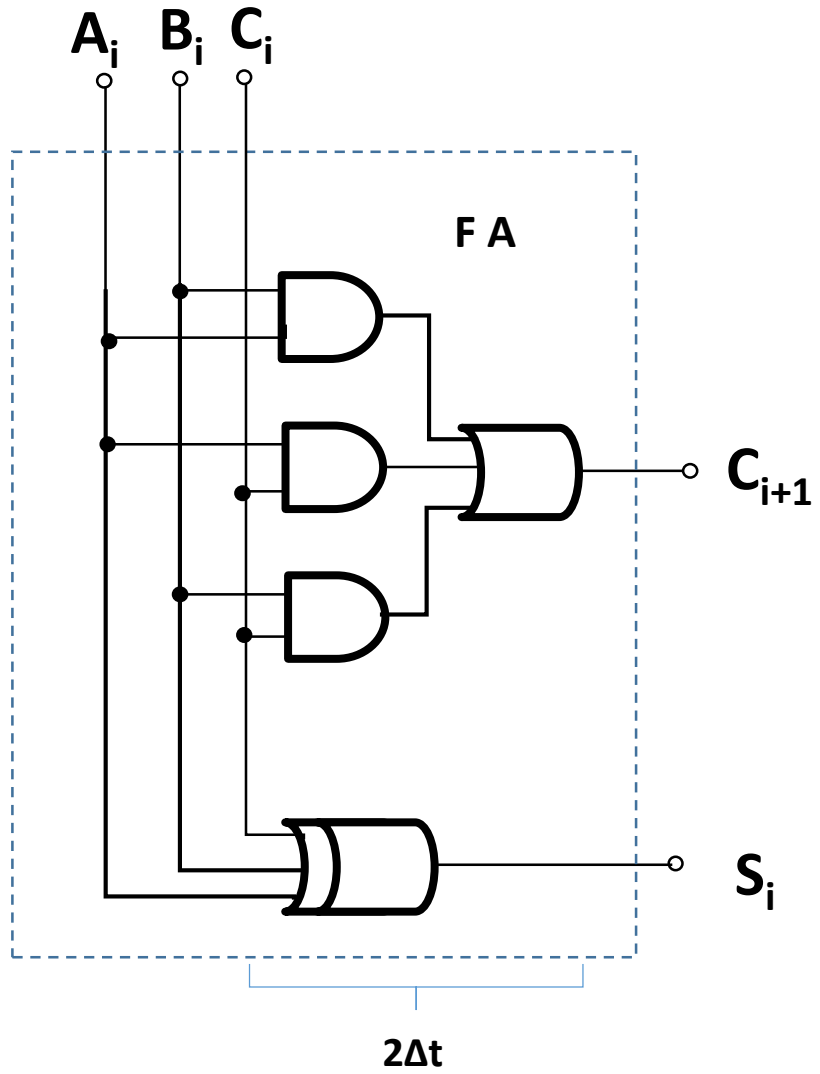
A_i	B_i	C_i	C_{i+1}	S_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



$C_{i+1} = A_iB_i + AiCi + BiCi$



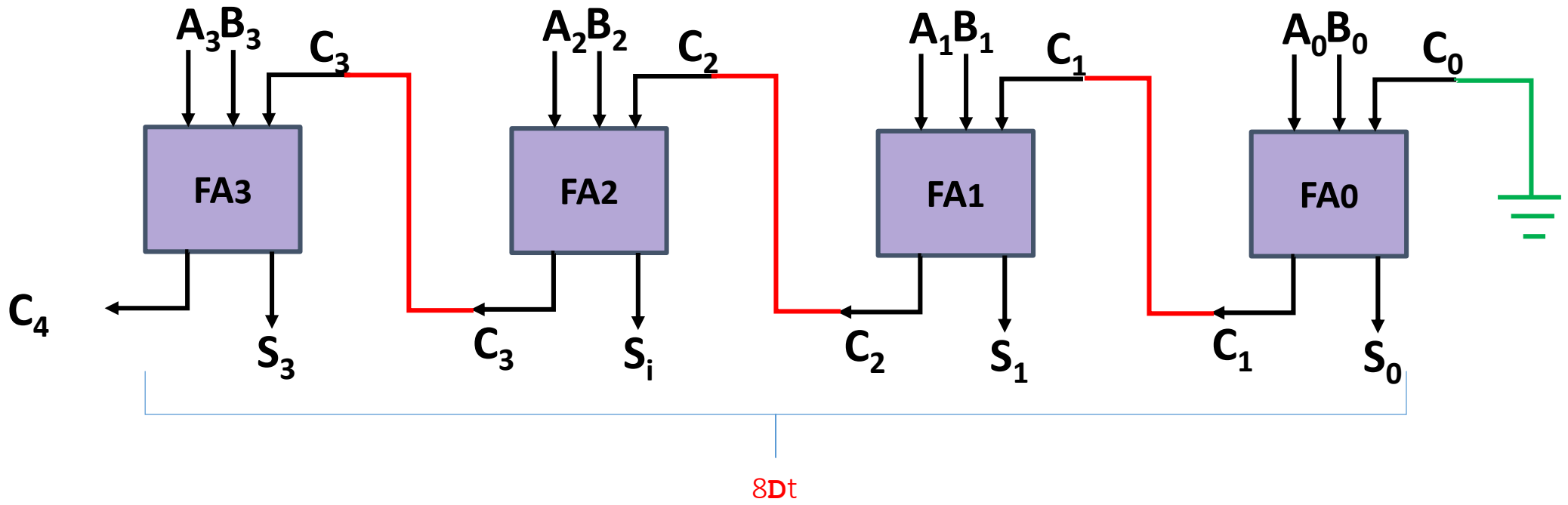
$S_i = A_i \oplus B_i \oplus C_i$



EJEMPLO 1 Diseñar un sumador de 2 palabras de 4 bits cada una utilizando FA

```
   C3C2C1
   A3A2A1A0
+  B3B2B1B0
 C4S3S2S1S0
```

EJEMPLO 1 Diseñar un sumador de 2 palabras de 4 bits cada una



OJO: Podemos observar que este circuito se tarda 8 Δt efectuar la suma, puesto que cada FA tiene que esperar el carry anterior para efectuar la suma de cada pareja de bits, de tal forma que si tenemos que sumar dos palabras de n bits cada una es sumador se tardaría:

2nΔt o bien (2n-1) Δt

c) Sumador con Carry Look Ahead (CLA)

Como mencionamos, el retraso que se produce al efectuar la suma es que cada etapa tiene que esperar el carry anterior para efectuar la suma de cada pareja de bits, si pudiéramos preveer cuanto vale el carry para cada etapa con anterioridad, podríamos realizar la suma de cada pareja de bits al mismo tiempo, para ello utilizaremos el Carry Look Ahead, veamos:

Desarrollo:

Ecs. (a)

del FA tenemos:

$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_{i+1} = A_iB_i + A_iC_i + B_iC_i = A_iB_i + (A_i \oplus B_i)\, C_i$$

llamaremos:

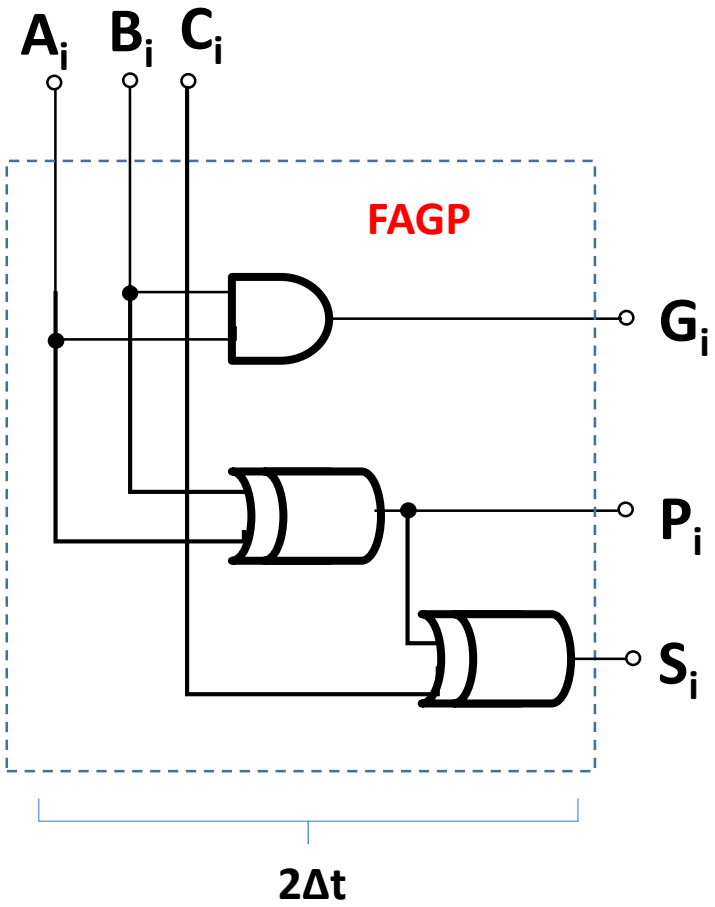
$$G_i = A_iB_i \qquad \text{(Generador)}$$

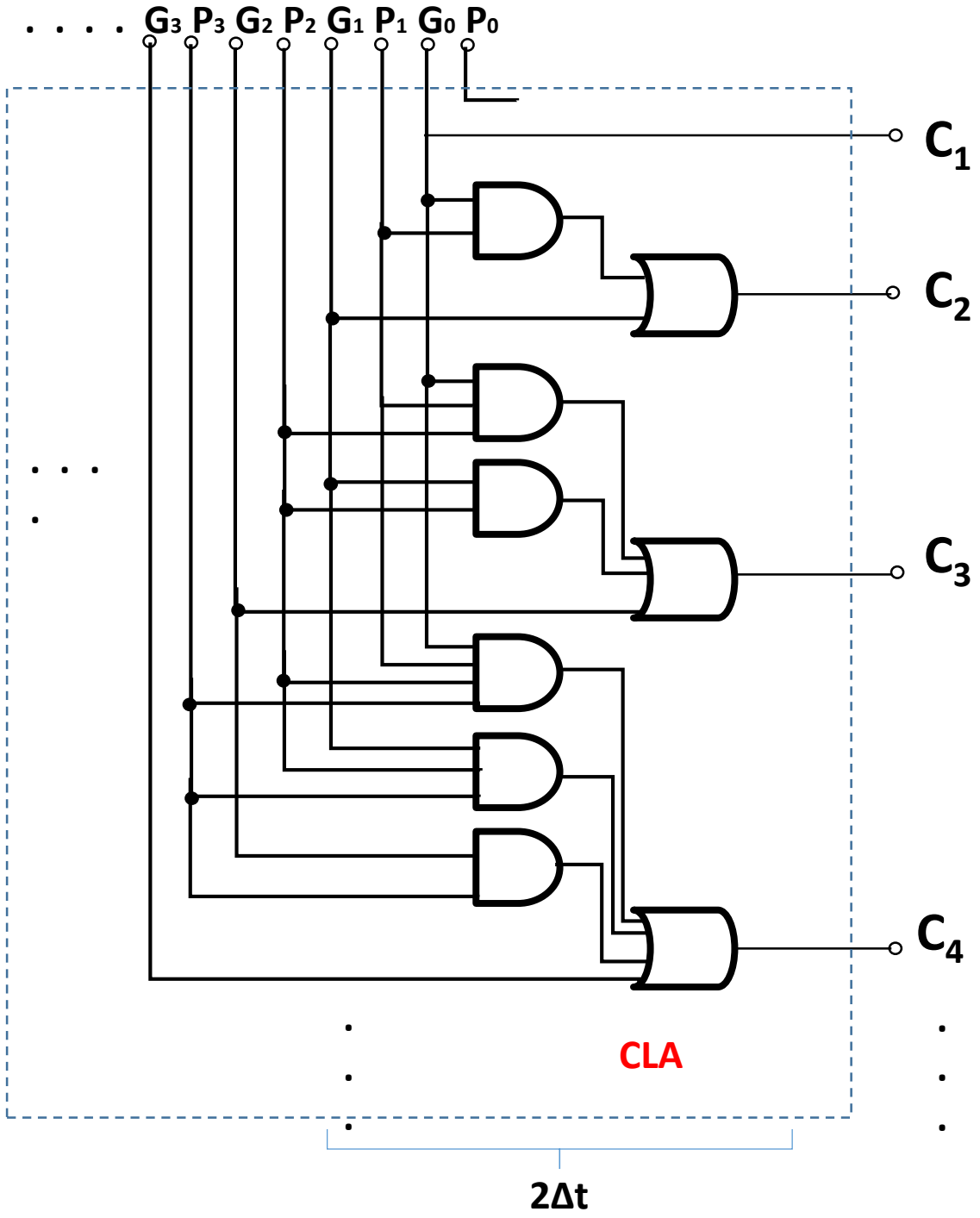
$$P_i = A_i \oplus B_i \qquad \text{(Propagador)}$$

de donde:

$$C_{i+1} = G_i + P_iC_i$$

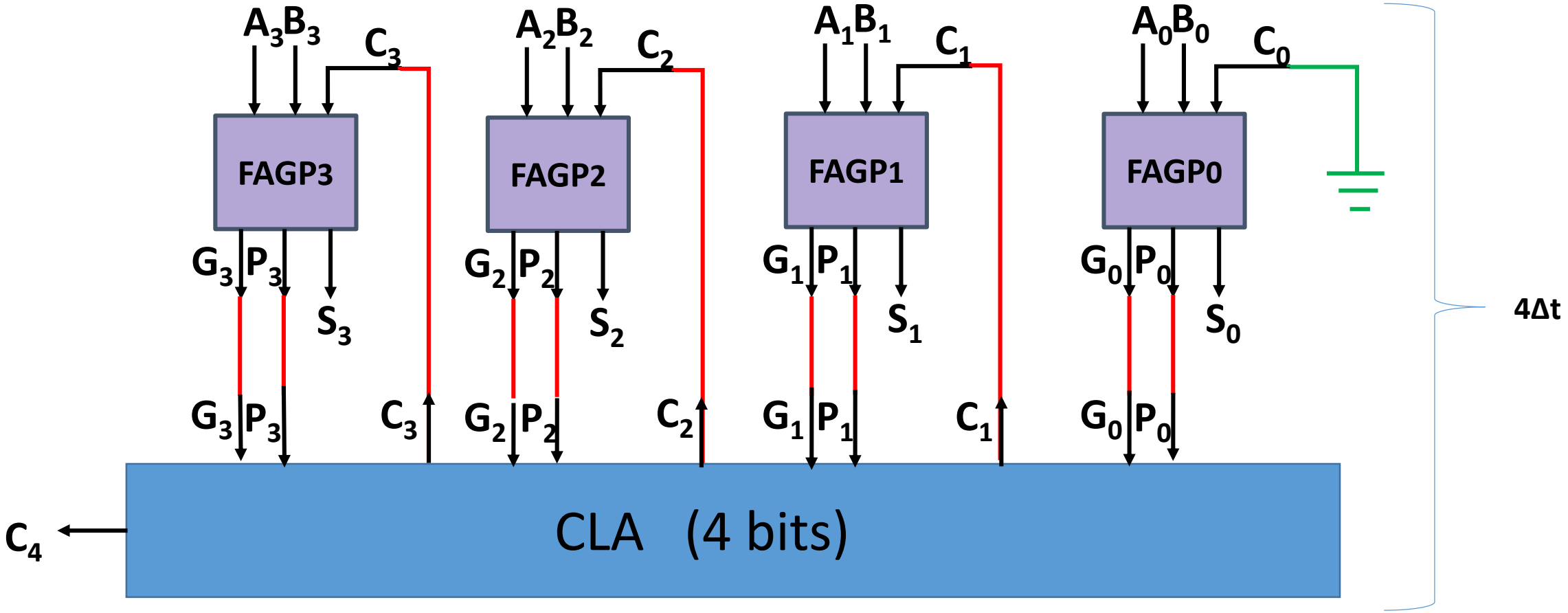
Nota: Generador, se genera un bit de acarro cuando A y B valen 1
Propagador, el carry se propaga si A o B valen “1”, pero solo uno de ellos





PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

EJEMPLO 1 BIS Diseñar un sumador de 2 palabras de 4 bits cada una con CLA



PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

OJO: PODEMOS OBSERVAR EN LA IMPLEMENTACIÓN ANTERIOR QUE EL SUMADOR, INDEPENDIENTEMENTE DEL TAMAÑO DE LA PALABRA (NÚMERO DE BITS), SIEMPRE SERÁ DE **4Δt**

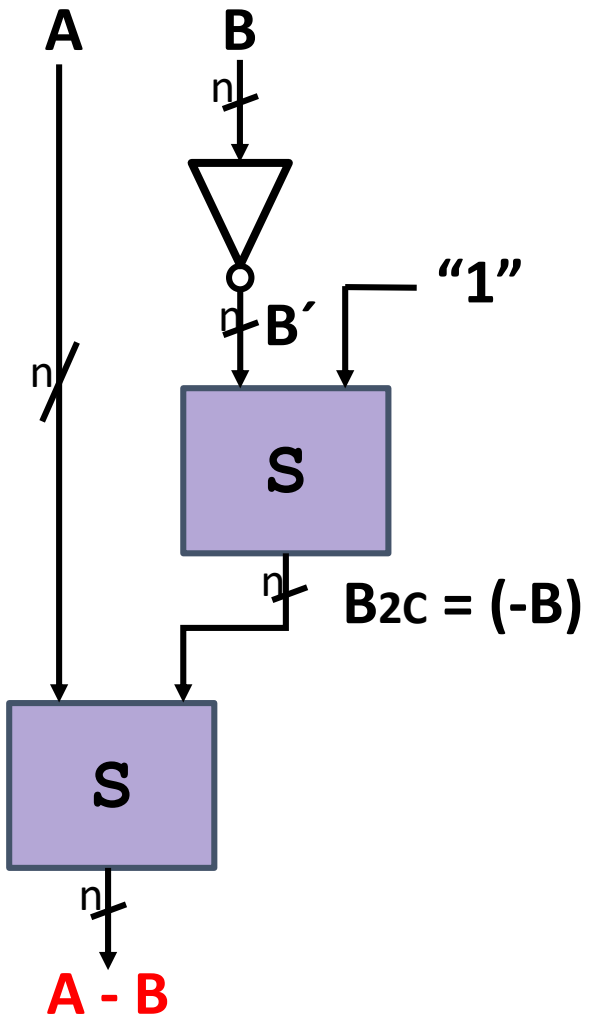
PROYECTO 3 DISEÑAR UN SUMADOR DE DOS PALABRAS DE 4 BITS UTILIZANDO CARRY LOOK AHEAD

PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

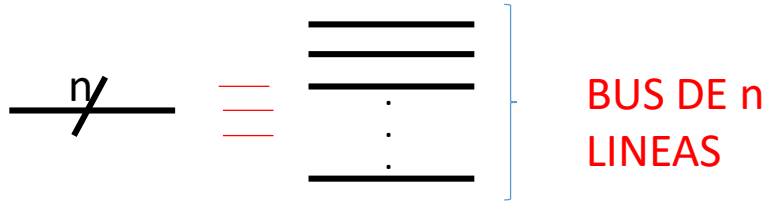
3. Diseño de restadores

OJO: QUEDAMOS QUE NO EXISTE LA RESTA, LO QUE EXISTE ES LA SUMA DE NUMEROS SIGNADOS.

EJEMPLO: Diseñar un restador de 2 palabras A y B de n bits cada una



NOTA:



4. Diseño de multiplicadores

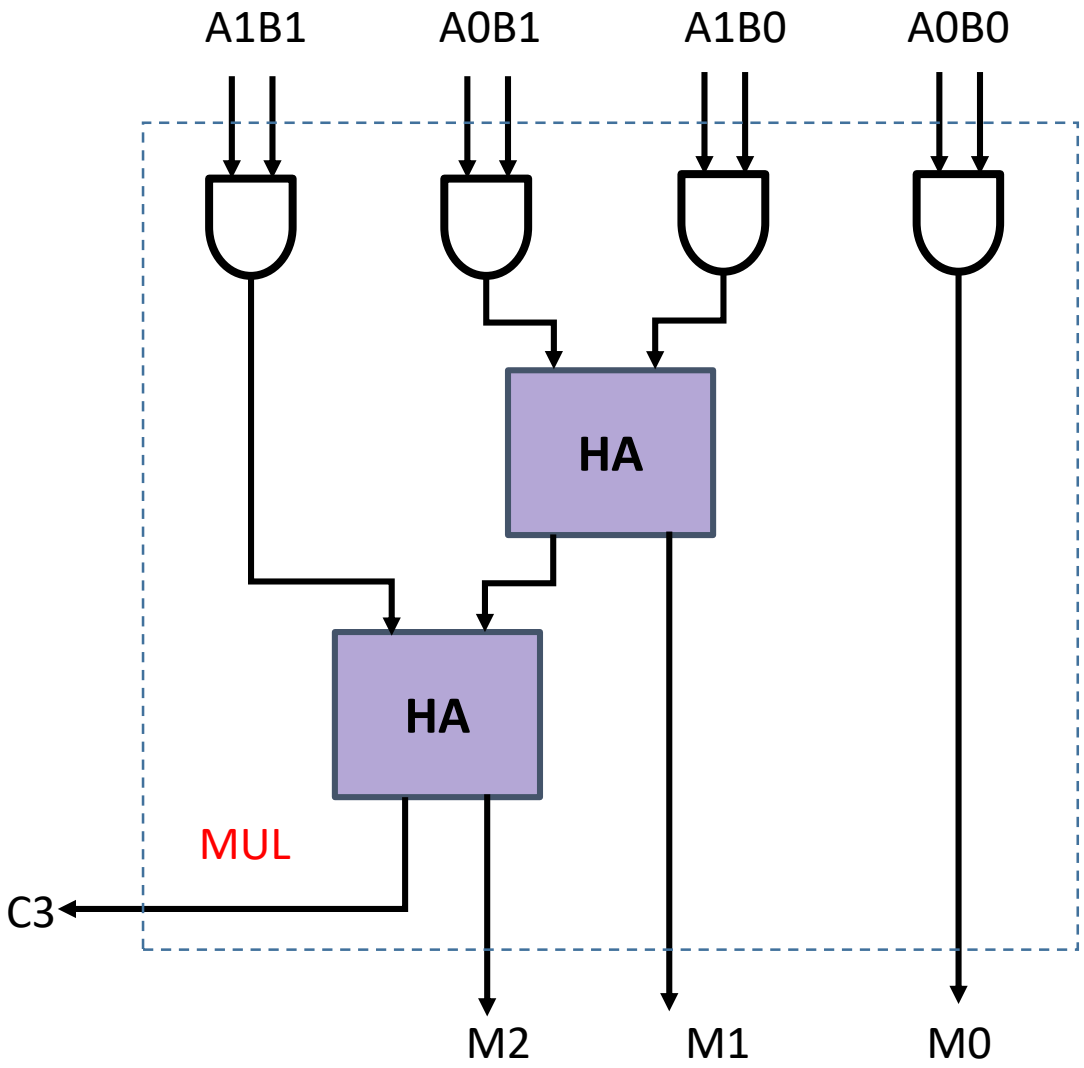
Existen 2 formas de diseñar multiplicadores con circuitos digitales, una es con Circuitos Combinacionales, entre ellos el método de Wallace Tree, y con Circuitos Secuenciales.

Veamos:

	2	3
x	1	5
	2x5	3x5
2x1	3x1	
M2	M1	M0

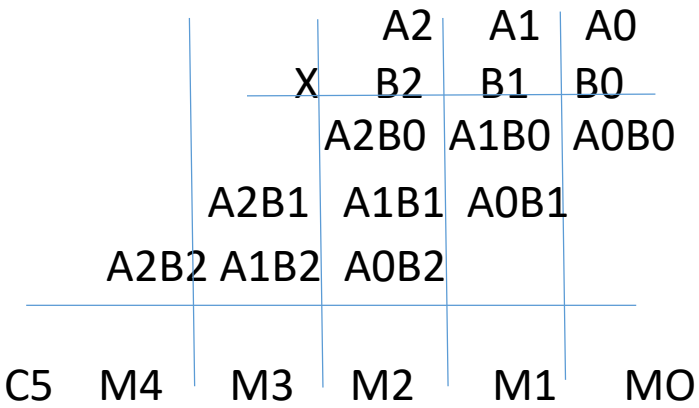
	A1	A0
x	B1	B0
	A1B0	A0B0
A1B1	A0B1	
M2	M1	M0

EJEMPLO 1: Diseñar un multiplicador de dos palabras A y B de dos bits cada una



PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

EJEMPLO 2: Diseñar un multiplicador de dos palabras A y B de tres bits cada una

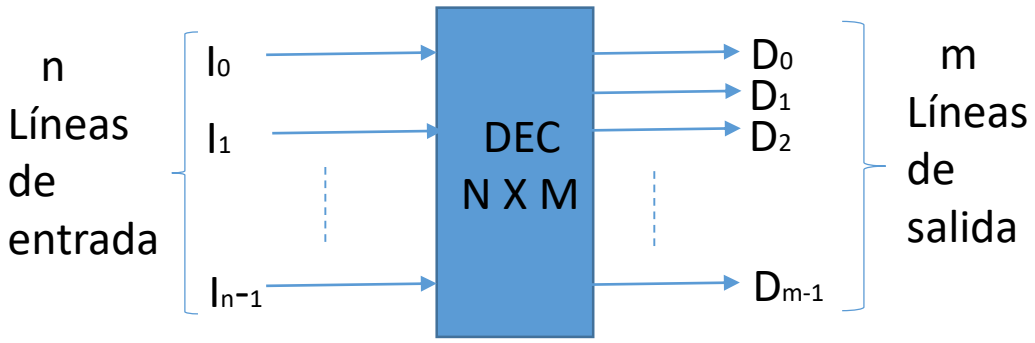


PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

5.-DISEÑO DE DECODIFICADORES

a)Conceptos

- Es un convertidor de código (cambia información binaria de un código a otro)
- Es un selector de dispositivos o de direcciones de memoria
- **SOLO UNA LÍNEA DE SALIDA SE VERIFICA PARA CADA UNA DE LAS COMBINACIONES DE LAS VARIABLES DE ENTRADA.**

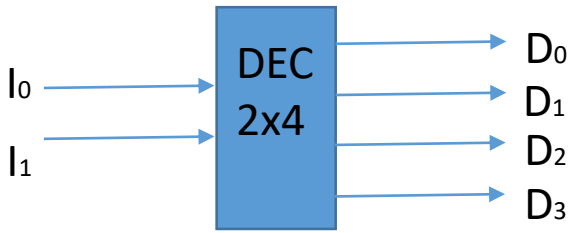


Donde $M = 2^n$

Los decodificadores están dados por el número de entradas por el número de salidas

Ejemplo: Diseñar un DEC 2X4

PRIMER PASO (Núm de var E/S)



SEGUNDO PASO (Tab. De verdad)

I_1	I_0	D_3	D_2	D_1	D_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

TERCER PASO (minimización)

$D_0 = I'_1 I'_0$

$D_1 = I'_1 I_0$

$D_2 = I_1 I'_0$

$D_3 = I_1 I_0$

TERCER PASO
(minimización)

$D_0 = I'_1 I'_0$

$D_1 = I'_1 I0$

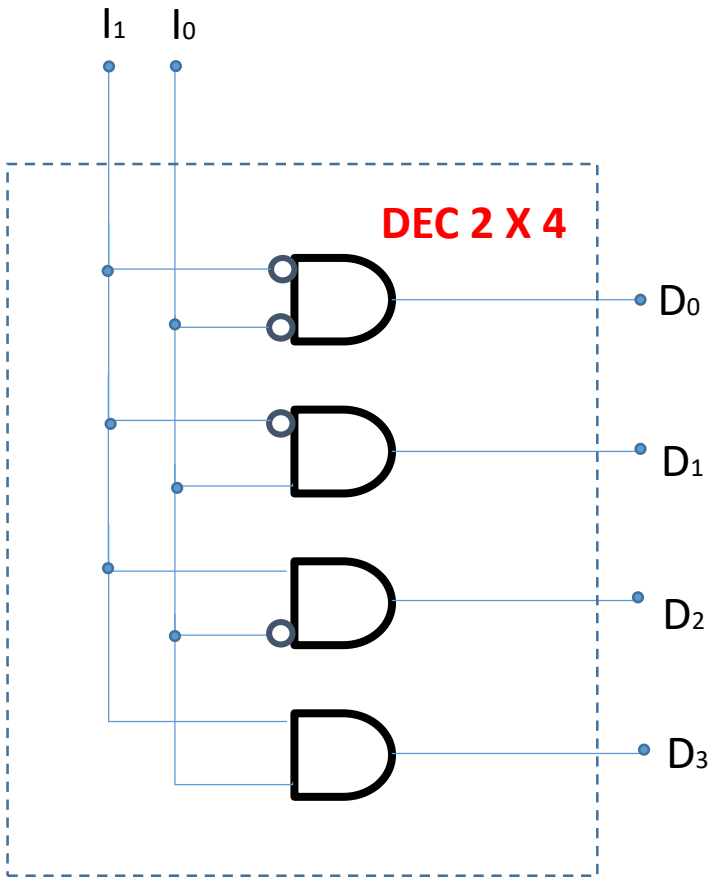
$D_2 = I_1 I'_0$

$D_3 = I_1 I_0$

mintérminos

OJO: Podemos observar que el decodificador genera mintérminos, es decir que para cada una de las combinaciones de las variables de entrada corresponden a un mintérmino.

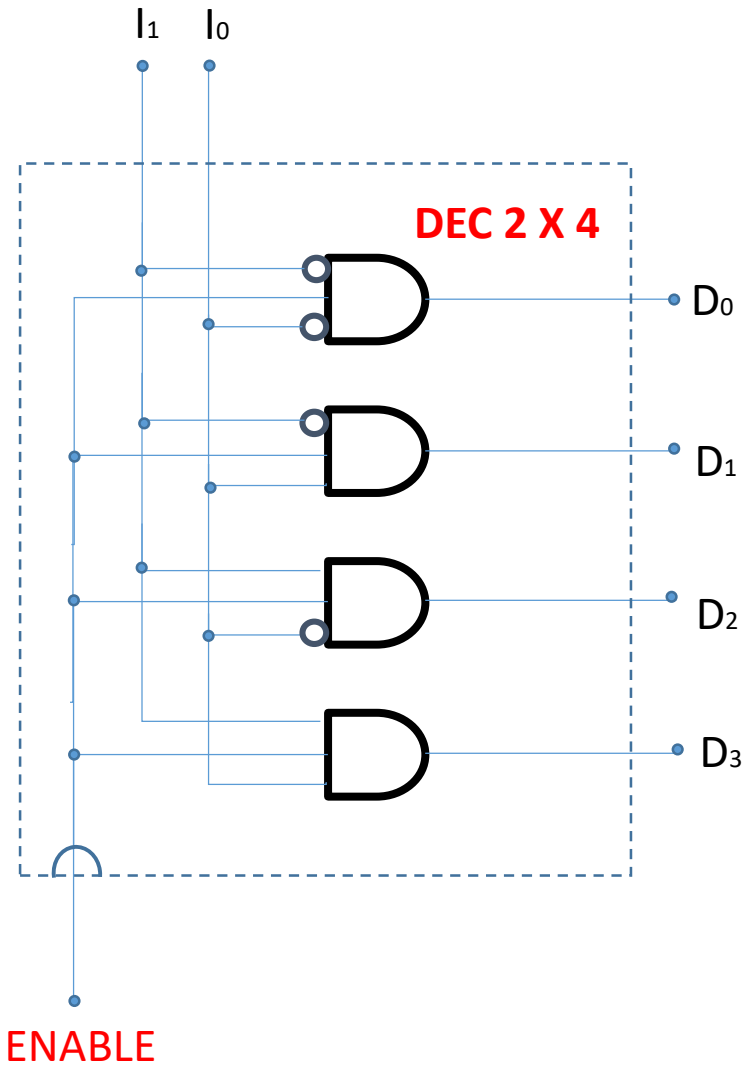
CUARTO PASO
(implementación)



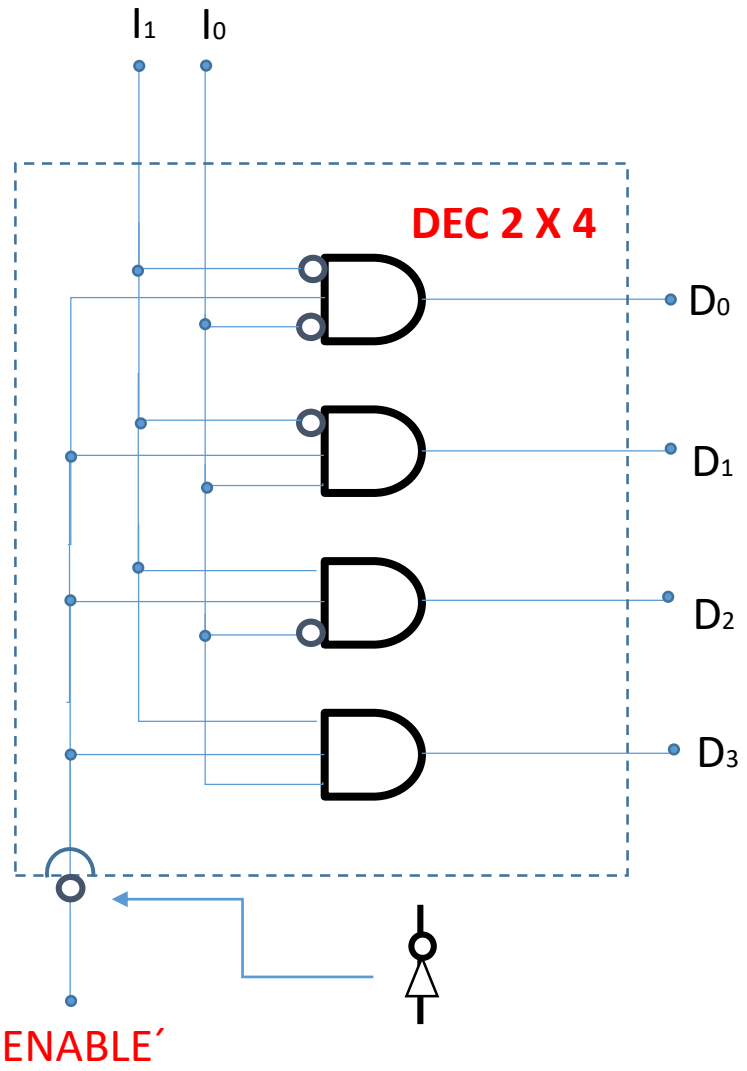
PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

EJEMPLO 3: DISEÑAR UN DEC 2 X 4 CON SEÑAL DE ENABLE

NOTA: La señal de ENABLE es una entrada adicional a los circuitos lógicos que habilitan el circuito para que funcione o no.



O bien



b) Circuitos MSI y LSI

Los circuitos MSI son circuitos de Mediana Escala de Integración y los circuitos LSI son circuitos de Gran Escala de Integración.

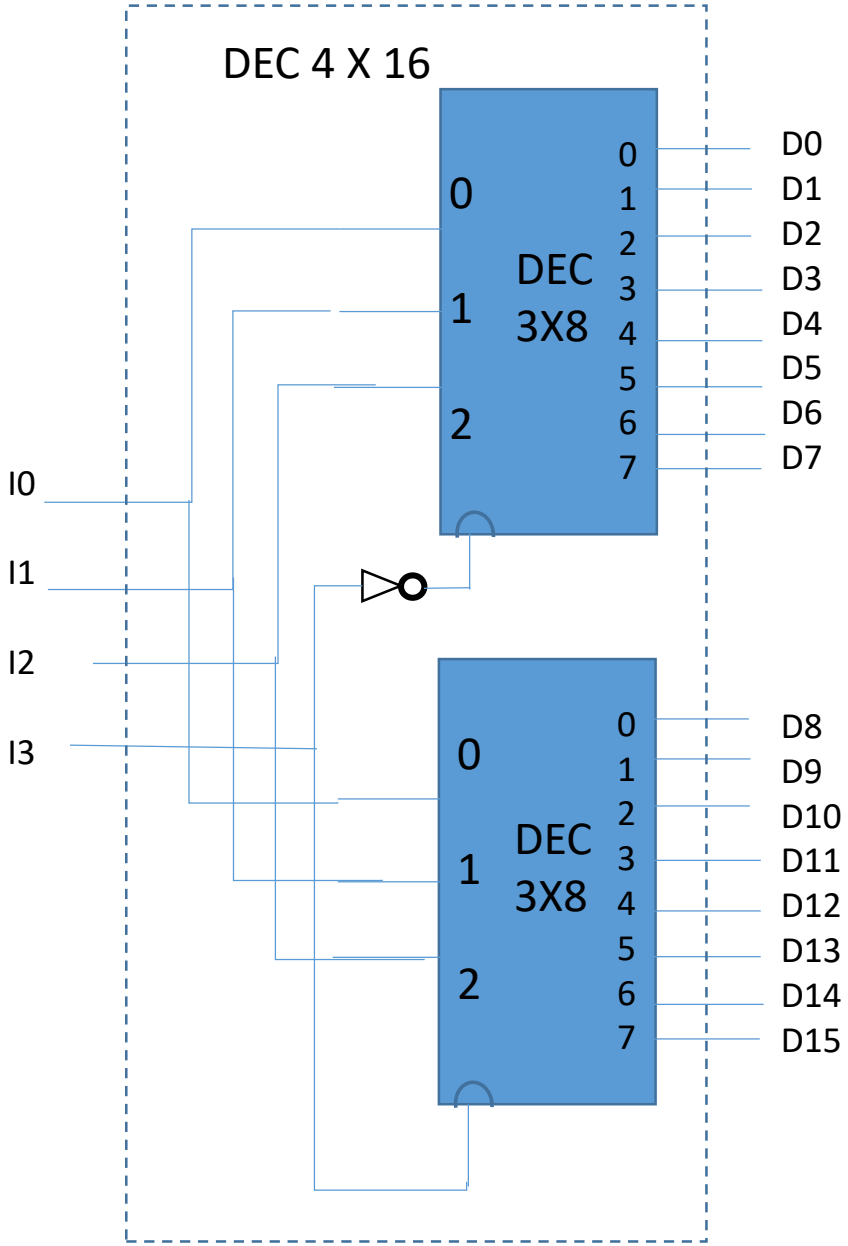
Otra forma de medir el nivel de integración de una componente es mediante la llamada DENSIDAD que indica la cantidad de transistores que representa una componente (compuertas, circuitos MSI, LSI, VLSI,…., Microprocesador).

Las compuertas lógicas son de muy baja densidad y son circuitos SSI, es decir, de Pequeña Escala de Integración.

Los Decodificadores son circuitos MSI y los podemos ya encontrar integrados en un sol CHIP (componente), y por lo tanto podemos diseñas CON ellos.

c) Diseño CON decodificadores.

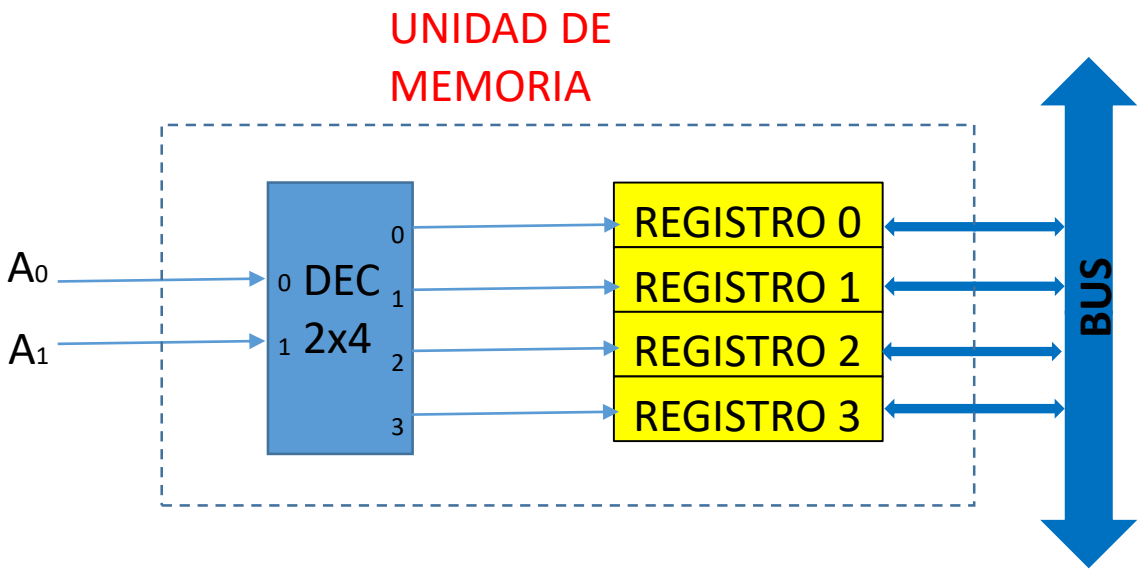
EJEMPLO 1: Diseñar un decodificador 4X16 con dos decodificadores 3X8 con señal de enable.



TAREA. IMPLEMENTAR UN DEC 5X32 CON 4 DEC 3X8 CON ENABLE Y UN DEC 2X4

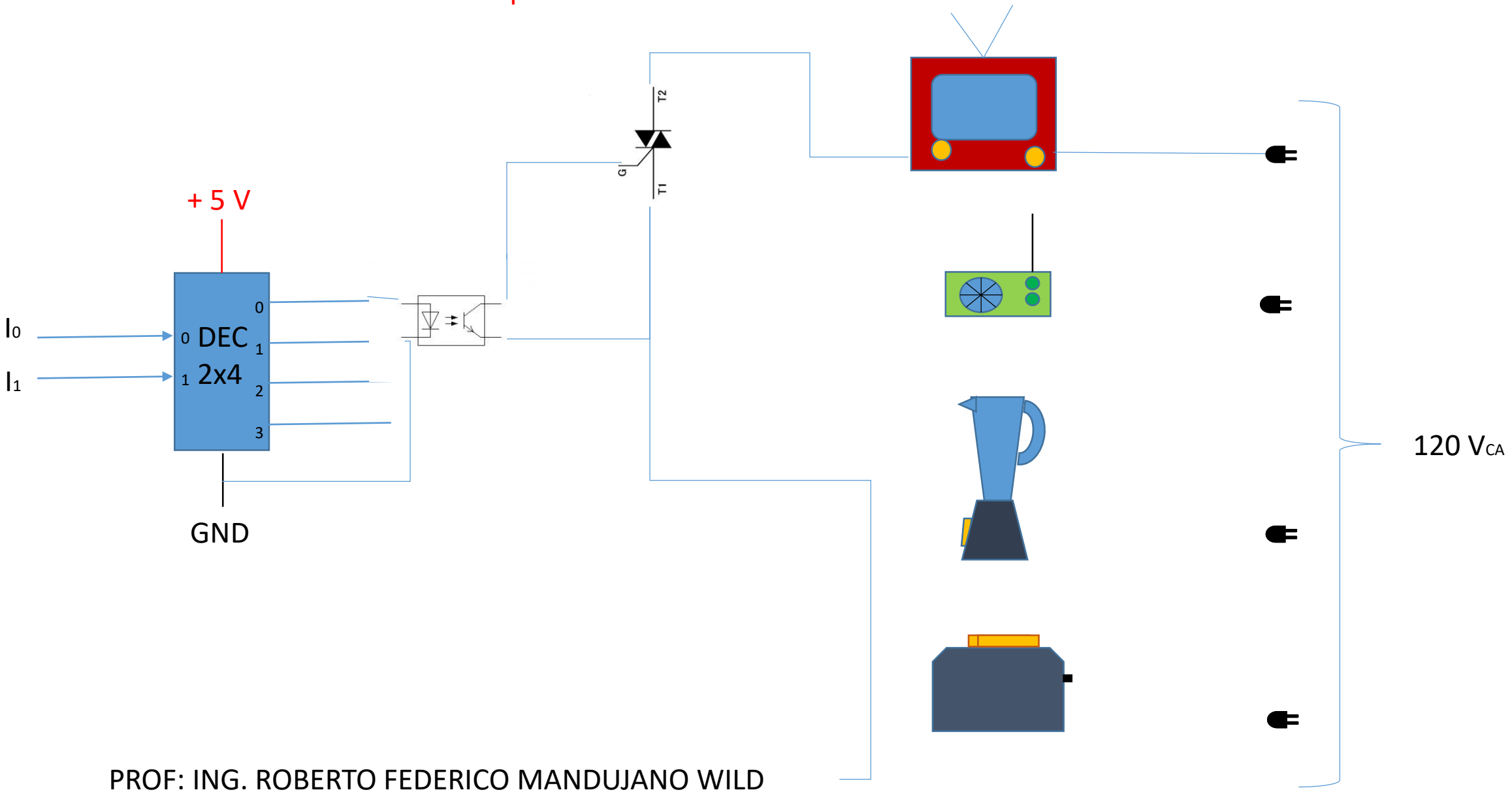
PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

EJEMPLO 2: Diseñar un selector de registros de memoria que están conectados a un BUS común, la Unidad de Memoria es de 4 registros.



PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

EJEMPLO 3: Diseñar un selector de 4 dispositivos de CORRIENTE ALTERNA

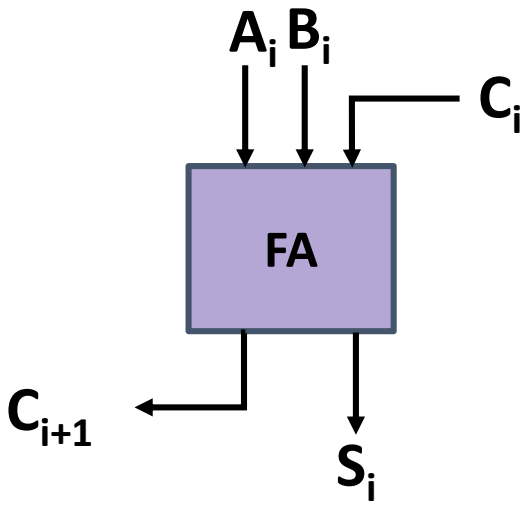


PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

EJEMPLO 4: Implementar un sumador completo FA utilizando un DEC 3X8

SI, Los de DECODIFICADORES nos sirven para implementar funciones boolenas, puesto que a la salida de un decodificador obtenemos mintérminos, y cualquier función boolena se puede representar en forma canónica: suma de productos (mintérminos) o producto de sumas (Maxtérminos). Al implementar con un decodificador no es necesario minimizar las funciones booleanas y solo es necesario "sumar" los mintérminos de las funciones utilizando compuertas OR. **LOS DECODIFICADORES ME SIRVEN PARA IMPLEMENTAR VARIAS FUNCIONES BOOLEANAS.**

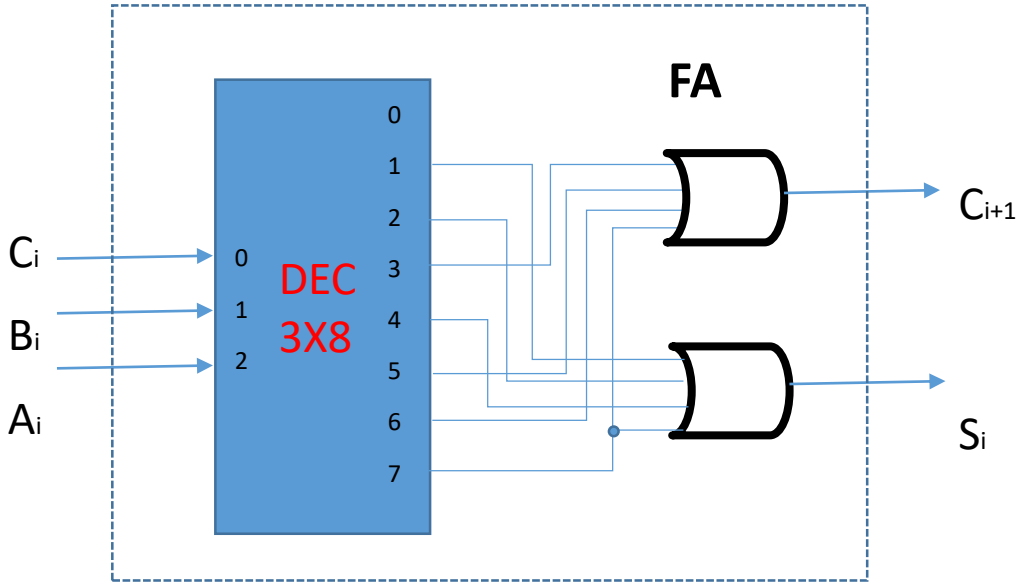
Para el ejemplo, tenemos:



Ai	Bi	Ci	C_{i+1}	S_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$C_{i+1} = \Sigma m\ (3,5,6,7)$

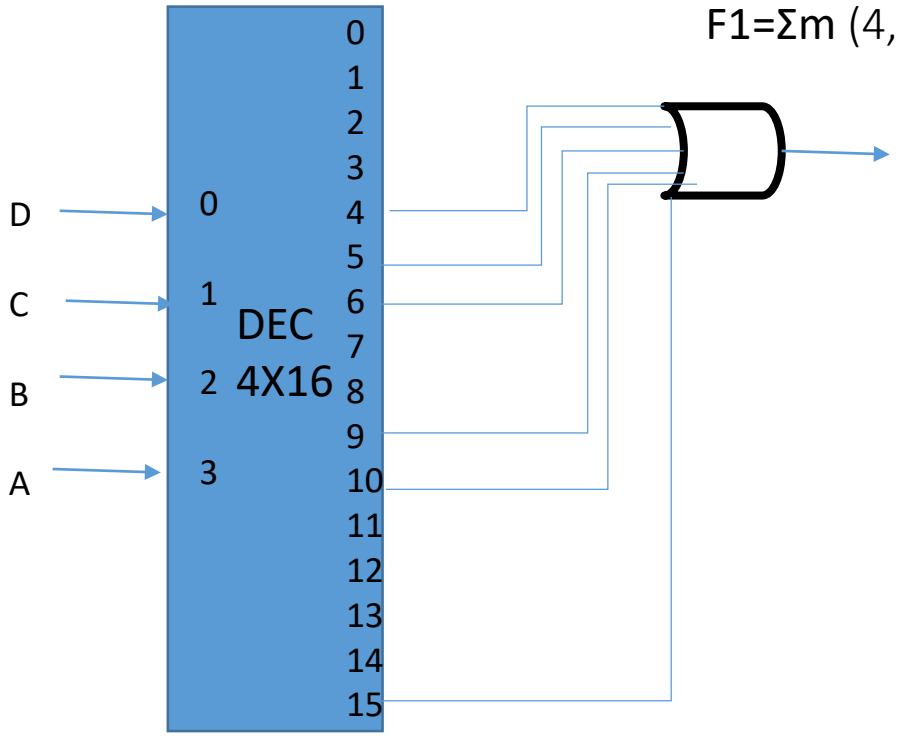
$S_i = \Sigma m\ (1,2,4,7)$



PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

EJEMPLO 5: Implementar las siguientes funciones boolenas utilizando un DEC

A	B	C	D	F1	F2	F3	F4
0	0	0	0	0	1	0	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	0
0	0	1	1	0	0	1	0
0	1	0	0	1	0	0	0
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	1	1
1	0	0	1	1	0	0	1
1	0	1	0	1	0	1	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	0	1
1	1	1	0	0	1	1	1
1	1	1	1	1	0	0	0



F1=Σm (4,5,6,9,10,15)

PROF: ING. ROBERTO FEDERICO MANDUJANO WILD