Lógica Combinacional

OdC 2021

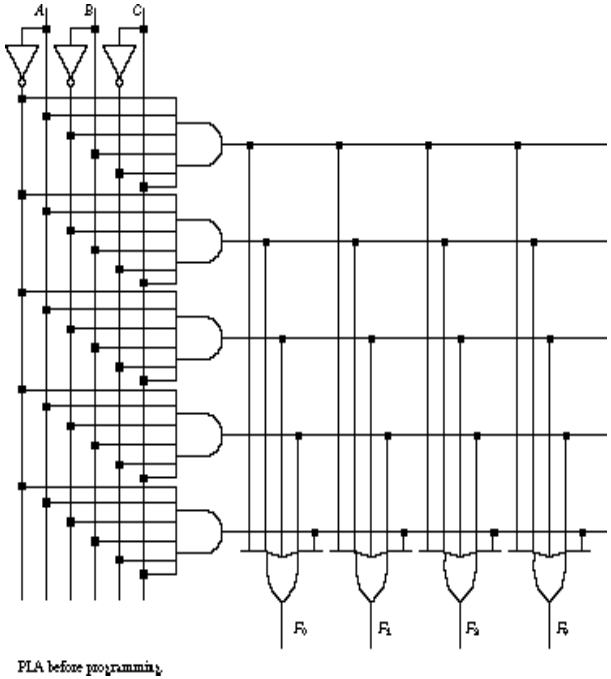
Minitérminos y maxitérminos

Minitérminos y maxitérminos para tres variables binarias

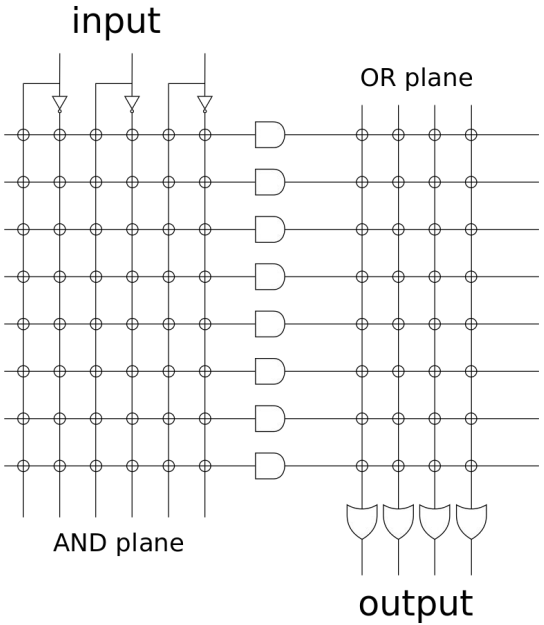
			Minitérminos		Maxitérminos	
x	y	z	Términos	Designación	Términos	Designación
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

Las funciones booleanas expresadas como suma de minitérminos o producto de maxitérminos están en **forma canónica**.

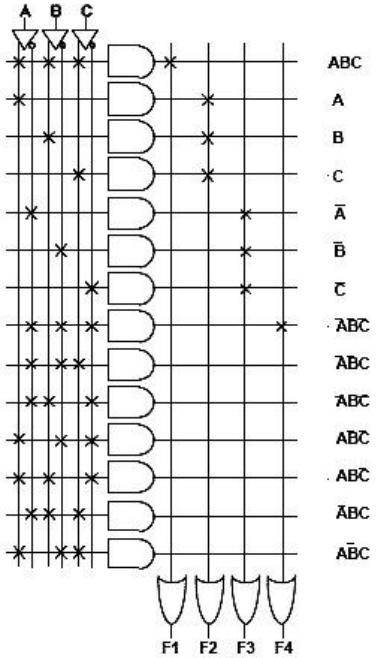
Programmable Logic Array (PLA)



Completa



Simplificada



Configurada

Ejercicio 1

Un detector de paridad impar de 4 entradas y una salida funciona de la siguiente manera: si la cantidad de entradas con valor ‘1’ es impar la salida se pone en ‘1’, en el resto de los casos la salida toma valor ‘0’.

a) Construir la tabla de verdad para dicho sistema.

b) Obtener la ecuación lógica como suma de minitérminos y producto de maxitérminos (funciones canónicas).

c) Implementar el sistema con compuertas NAND de la cantidad de entradas requeridas.

d) Implementar el sistema con una PLA.

Ejercicio 1 - a)

Un detector de paridad impar de 4 entradas y una salida funciona de la siguiente manera: si la cantidad de entradas con valor ‘1’ es impar la salida se pone en ‘1’, en el resto de los casos la salida toma valor ‘0’.

A	B	C	D	S
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Ejercicio 1 - b)

Ecuación lógica como suma de minitérminos (suma de productos):

S = A’B’C’D + A’B’CD’ + A’BC’D’ + A’BCD + AB’C’D’ + AB’CD+
ABC’D + ABCD’

Ecuación lógica como producto de maxitérminos (producto de sumas):

S = (A+B+C+D)*(A+B+C’+D’)*(A+B’+C+D’)*(A+B’+C’+D)*(A’+B+C+D’)
(A’+B+C’+D)(A’+B’+C+D)*(A’+B’+C’+D’)

A	B	C	D	S
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Ejercicio 1 - c)

A partir de la ecuación lógica como suma de minitérminos:

S = A’B’C’D + A’B’CD’ + A’BC’D’ + A’BCD + AB’C’D’ + AB’CD + ABC’D + ABCD’

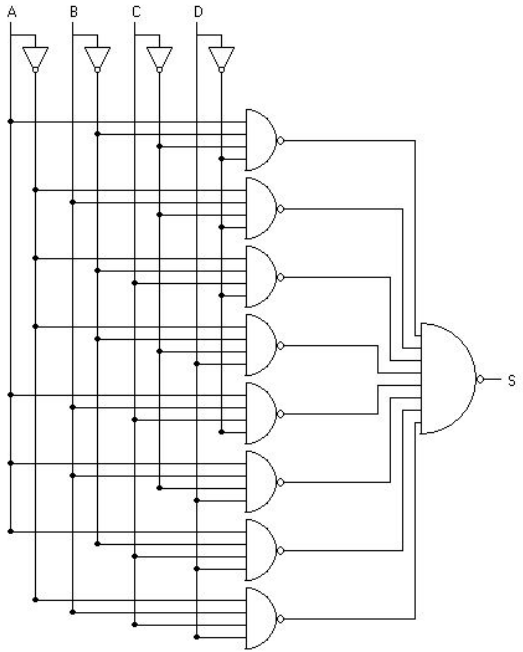
Negando 2 veces S (S’’ = S) se obtiene:

S’’ = (A’B’C’D + A’B’CD’ + A’BC’D’ + A’BCD + AB’C’D’ + AB’CD + ABC’D + ABCD’)’’

S = ((A’B’C’D)’*(A’B’CD’)’*(A’BC’D’)’*(A’BCD)’*(AB’C’D’)’*(AB’CD)’*(ABC’D)’*(ABCD’)’)’

Ejercicio 1 - c)

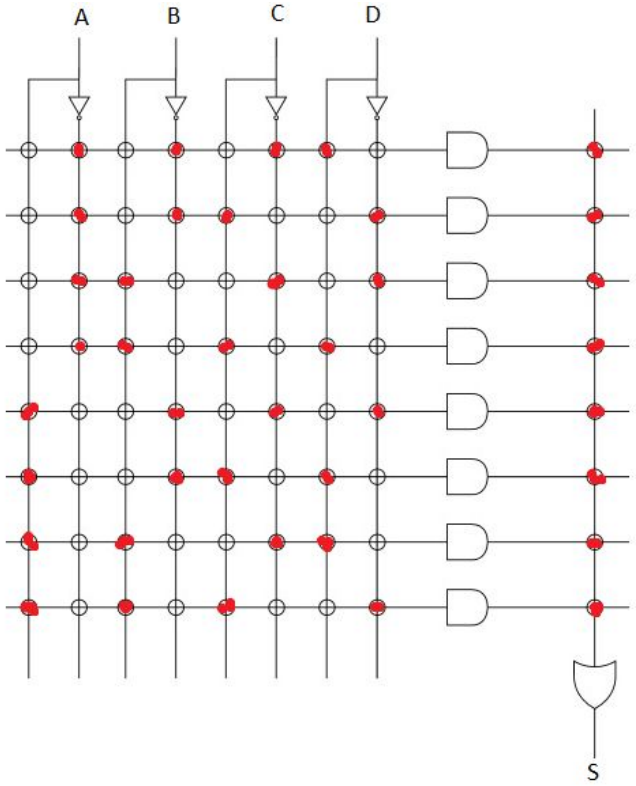
S = ((A’B’C’D)’*(A’B’CD’)’*(A’BC’D’)’*(A’BCD)’*
(AB’C’D’)’*(AB’CD)’*(ABC’D)’*(ABCD’)’)’



Ejercicio 1 - d)

Ecuación lógica como suma de minitérminos (suma de productos):

S = A’B’C’D + A’B’CD’ + A’BC’D’ + A’BCD + AB’C’D’ + AB’CD+ABC’D + ABCD’



Ejercicio 2

Un sistema digital recibe información en forma de palabras de 5 bits (**ABCDE**) en un código protegido contra errores, de tal forma que cualquier dato que se reciba debe contener 3 y sólo 3 bits en ‘1’. Diseñar un circuito con las entradas **ABCDE** y una salida **err** que se activa por bajo cuando se recibe un dato incorrecto.

a) Construir la tabla de verdad para dicho sistema.

b) Obtener la ecuación lógica como suma de minitérminos y producto de maxitérminos (funciones canónicas).

c) Implementar el sistema con una PLA.

Ejercicio 2 - a)

A	B	C	D	E	err
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	0
...					

A	B	C	D	E	err
...					
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	0	1	0
1	1	1	1	0	0
1	1	1	1	1	0