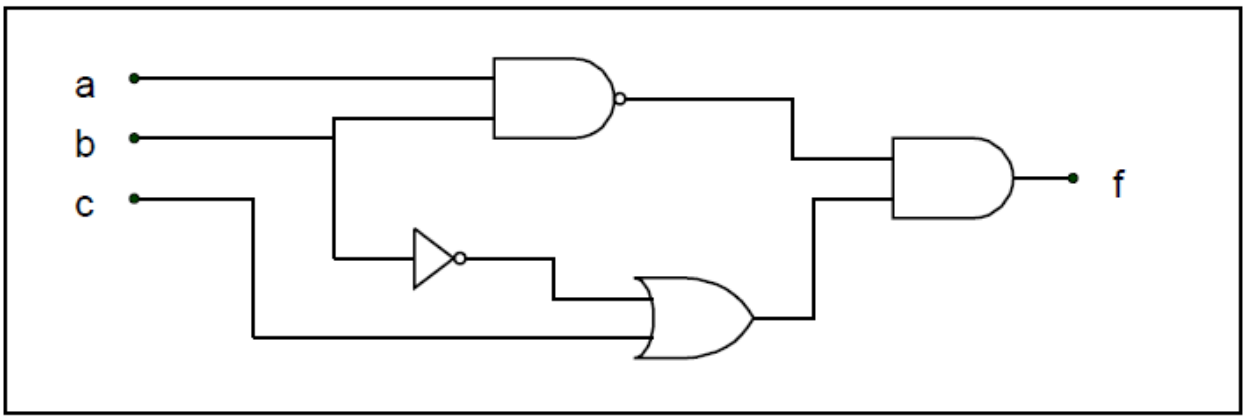
Ejercicio 1

Dado el diagrama de la figura:



Se pide:

a) Escribir su tabla de verdad.
b) Simplificar utilizando Álgebra de Boole.
c) Realizar una segunda implementación utilizando únicamente puertas de tipo NAND.

Ejercicio 2

Dada la función lógica de cuatro variables:

$$Z = AC\overline{D} + \overline{A}BC\overline{D} + \overline{A}\overline{B}CD + A\overline{B}D$$

Se pide:

a) Escribir su tabla de verdad.
b) Utilizando mapas de Karnaugh, reducir la función a una suma de productos mínima.
c) Utilizando mapas de Karnaugh, reducir la función a un producto de sumas mínimo.

Ejercicio 3

Queremos realizar un circuito que detecte cada vez que se establezcan en las entradas las combinaciones correspondientes a las 4 últimas cifras no repetidas del DNI del integrante más joven de cada grupo.

Además, la salida deberá también valer 1 para las combinaciones 12 y 14. Por ejemplo si el DNI a detectar fuera el 12.457.507 se deberá realizar un circuito cuya salida se ponga a 1 para los valores de entrada: 7, 0, 5, 4, 12, y 14. Se pide:

a) Tabla de verdad de la función de salida Z
b) Simplificar la función de salida utilizando mapas de Karnaugh
c) Realizar una segunda implementación utilizando únicamente puertas de tipo NAND

Ejemplo:

Nº	D	C	B	A	**Z**
0	0	0	0	0	**1**
1	0	0	0	1	**0**
2	0	0	1	0	**0**
3	0	0	1	1	**0**
4	0	1	0	0	**1**
5	0	1	0	1	**1**
6	0	1	1	0	**0**
7	0	1	1	1	**1**
8	1	0	0	0	**0**
9	1	0	0	1	**0**
10	1	0	1	0	**0**
11	1	0	1	1	**0**
12	1	1	0	0	**1**
13	1	1	0	1	**0**
14	1	1	1	0	**1**
15	1	1	1	1	**0**

Ejercicio 4

Diseñar un detector de números primos y números de Fibonacci de 4 bits (X3, X2, X1, X0). Interpretar los posibles valores de entrada como 0, 1, 2, ... , 15. Se valorará la utilización del menor número posible de puertas.

Números primos a detectar: 2, 3, 5, 7, 11, 13

Números de Fibonacci a detectar: 1, 2, 3, 5, 8, 13

a) Utilizando 1 decodificador 4-a-16 y las puertas básicas (OR de 2 entradas) necesarias

b) Utilizando 2 multiplexores 16-a-1

c) Utilizando 2 multiplexores 8-a-1 y las puertas básicas (NOT) necesarias

d) Utilizando 2 multiplexores 4-a-1 y las puertas básicas (AND, NOT) necesarias

Ejercicio 5

Diseñar un detector de números primos de 5 bits (X4, X3, X2, X1, X0), utilizando los siguientes circuitos combinacionales: 2 decodificadores 3-a-8, un multiplexor 8-a-1, dos multiplexores 2-a-1, y el menor número de puertas básicas (OR) que sean necesarias. Interpretar los posibles valores de entrada como 0, 1, 2, ... , 31.

Números primos a detectar: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31

Ejercicio 6

a) Diseñar un semisumador de 1 bit utilizando un decodificador 2-a-4 y las puertas básicas (AND, OR, etc.) necesarias. Se valorará la utilización del menor número posible de puertas.

b) Diseñar un sumador completo de 1 bit utilizando 2 multiplexores 4-a-1 y las puertas básicas (AND, OR, etc.) necesarias. Se valorará la utilización del menor número posible de puertas.

c) Utilizando los dos componentes diseñados anteriormente, diseñar un sumador de números de 2 bits A=(A1, A0) y B=(B1, B0), *sin acarreo de entrada*.

Ejercicio 7

Un codificador de prioridad es un circuito combinacional que tiene el mismo comportamiento de un codificador convencional con la particularidad de permitir que varias de sus entradas estén activas simultáneamente.

En este ejercicio, en caso de varias entradas activas, el codificador pone en su salida el índice codificado de la entrada activa de *menor* peso. Si ninguna entrada estuviera activa, el codificador se comportará de igual manera que si la entrada de menor peso estuviera activa.

La siguiente tabla de verdad muestra el comportamiento del codificador:

ENTRADAS				**SALIDAS**	
X3	X2	X1	X0	Y1	Y0
0	0	0	0	0	0
X	X	X	1	0	0
X	X	1	0	0	1
X	1	0	0	1	0
1	0	0	0	1	1

Se pide:

a) Simplificar las funciones de salida (Y1, Y0) utilizando mapas de Karnaugh.
b) Realizar el diseño del circuito simplificado.