

Actividades operaciones lógicas

Sistemas Informáticos

Francisco Javier Arruabarrena Sabroso

Índice

Ejercicio 1.....	2
Ejercicio 2.....	3
Ejercicio 3.....	3
Ejercicio 4.....	4
Ejercicio 5.....	5
Ejercicio 6.....	6
Ejercicio 7.....	7

Ejercicio 1

Realiza las siguientes operaciones aritméticas en binario.

$$\begin{array}{r} 110100_2 \\ + 101100_2 \\ \hline \end{array}$$

1100000

$$\begin{array}{r} 111000_2 \\ + 110000_2 \\ \hline \end{array}$$

1101000

$$\begin{array}{r} 100111_2 \\ + 111111_2 \\ \hline \end{array}$$

1100110

$$\begin{array}{r} 1001110_2 \\ - 100100_2 \\ \hline \end{array}$$

0101010

$$\begin{array}{r} 1010101_2 \\ - 110100_2 \\ \hline \end{array}$$

0100001

$$\begin{array}{r} 1001111_2 \\ - 100110_2 \\ \hline \end{array}$$

0101001

$$\begin{array}{r} 110101_2 \\ + 110001_2 \\ \hline \end{array}$$

1100110

$$\begin{array}{r} 110110_2 \\ + 100100_2 \\ \hline \end{array}$$

1011010

$$\begin{array}{r} 100110_2 \\ + 111110_2 \\ \hline \end{array}$$

1100100

$$\begin{array}{r} 1100001_2 \\ - 111100_2 \\ \hline \end{array}$$

100101

$$\begin{array}{r} 1011111_2 \\ - 111001_2 \\ \hline \end{array}$$

100110

$$\begin{array}{r} 1011010_2 \\ - 100111_2 \\ \hline \end{array}$$

110011

Ejercicio 2

Realizar las siguientes operaciones y expresar el resultado en binario y en hexadecimal, aplicando para las operaciones necesarias Ca2:

$$\begin{aligned}
 6FC2_{(16)} - 754_{(10)} &= 110111111000010 - 000001011110010 \\
 &\quad 110111111000010 - 111110100001101 \\
 &\quad 110111111000010 - 111110100001110 \\
 &\quad 110111111000010 + 111110100001110 = \\
 &\quad \underline{1\ 110110011010000}
 \end{aligned}$$

Solución en Binario: **110110011010000**

Solución en Hexadecimal: **110 1100 1101 0000 → 6CD0**

$$7FC3_{(16)} + 642_{(10)} = 111111111000011 + 1010000010$$

Solución en Binario: **1000001001000101**

Solución en Hexadecimal: **8245**

Ejercicio 3

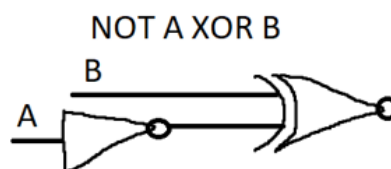
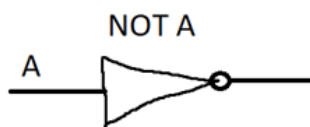
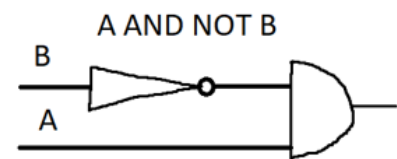
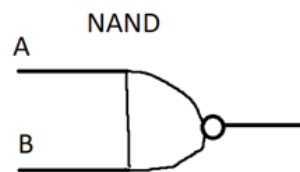
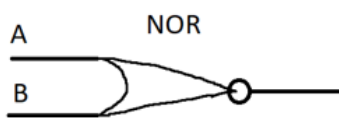
Sumar los números decimales 64 y -43 realizando la operación en binario representándolos en Ca1 en palabras de 8 bits.

$$\begin{aligned}
 64: & 1000000 & 1000000 - 00101011 \\
 43: & 0010\ 1011 & 1000000 - 11010100 \\
 & & 1000000 + 11010101 = 1\ 00010101 \rightarrow 00010101
 \end{aligned}$$

Ejercicio 4

Completa la tabla con las operaciones que se indican en la columna superior, con respecto a los valores de las columnas A y B y representa el circuito equivalente con puertas lógicas.

	Circuito 1	Circuito 2	Circuito 3	Circuito 4	Circuito 5
A B	A NOR B	A NAND B	A AND NOT B	NOT A	NOT A XOR B
0 0	1	1	0	1	1
0 1	0	1	0	1	0
1 0	0	1	1	0	0
1 1	0	0	0	0	1

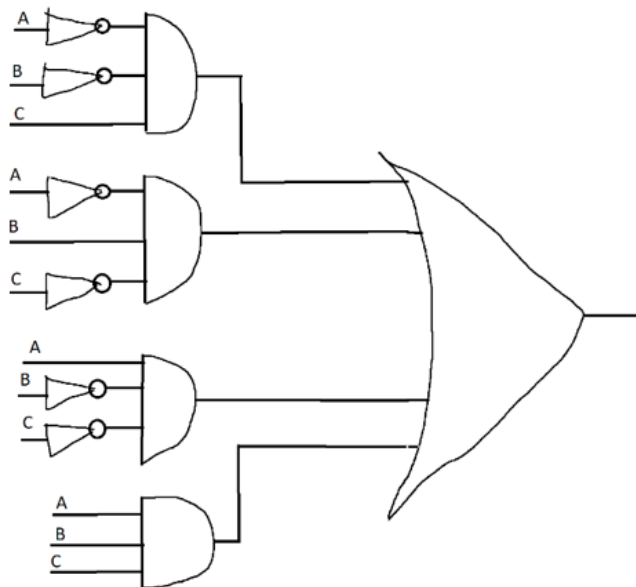
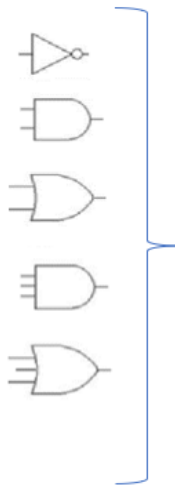


Ejercicio 5

Dada la siguiente tabla de verdad, representa la función equivalente y la representación con puertas lógicas utilizando puerta AND, NOT y OR.

A B C	F(S)
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	0
1 0 0	1
1 0 1	0
1 1 0	0
1 1 1	1

$F(S) =$



También se puede implementar con una puerta XOR (A, B, C).

Ejercicio 6

Para los valores $A = 44_{(10)}$, $B = 63_{(10)}$, $C = 31_{(10)}$, teniendo en cuenta la función obtenida y la tabla de verdad del apartado anterior. Obtener la función $F(S)$ de salida expresada en decimal, octal y hexadecimal.

$A)_{(2)} = 44 \rightarrow 44 - 32 = 12$; - 16 no; - 8 = 4; - 4 = 0; - 2 no; - 1 no

101100

$B)_{(2)} = 63 \rightarrow 63 - 32 = 31$; - 16 = 15; - 8 = 7; - 4 = 3; - 2 = 1; - 1 = 0

111111

$C)_{(2)} = 31 \rightarrow 11111$ (no pongo procedimiento puesto que es igual al anterior sin el primer dígito)

$F(S)_{(2)} =$

101100

111111

011111

001100

$F(S)_{(8)} = 001\ 100 \rightarrow 14$

$F(S)_{(16)} = 1100 \rightarrow C$

$F(S)_{(10)} = 2^2 + 2^3 = 12$

Notas: Tabla de potencias para cálculos:

28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20

256 128 64 32 16 8 4 2 1

Ejercicio 7

Dado los siguientes números en binarios, obtener las operaciones lógicas siguientes:

Operaciones Lógicas	A= 10011110 B= 01100011	A= 1100111101010101 B= 0011000111110000
A AND B (A & B)	00000010	0000000101010000
A OR B (A B)	11111111	1111111111110101
A NOR B	00000000	0000000000001010
A NAND B	11111101	1111111010101111
A XOR B	11111101	1111111010100101
NOT A	01100001	0011000010101010
NOT B	10011100	1100111000001111
(NOT A) (NOT B)	11111101	1111111010101111
(NOT A) & (NOT B)	00000000	0000000000001010
(NOT A) XOR (NOT B)	11111101	1111111010100101