



Tema: 06_Tarea: Cuadernillo

Autor/es:

Yanacallo Caiza Andy Esteban

Carrera: Tecnología Superior en Desarrollo de Software

ELECTRÓNICA-SISTEMAS DIGITALES

Clase 26/08/2024

booleanos

Not

Operaciones básicas

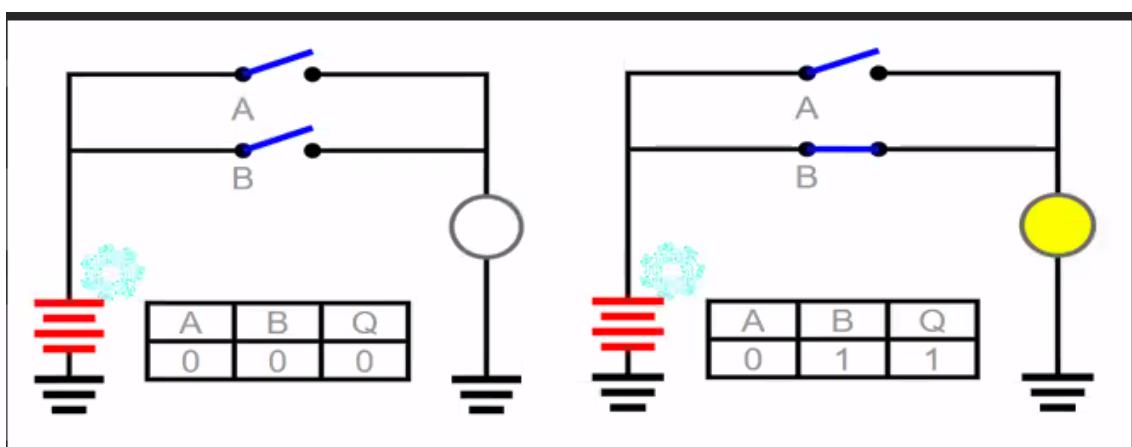
Complemento

Símbolo	Expresión	Tabla de verdad						
	$Q = \bar{A}$	<table border="1"> <thead> <tr> <th>A</th><th>Q</th></tr> </thead> <tbody> <tr> <td>0</td><td>1</td></tr> <tr> <td>1</td><td>0</td></tr> </tbody> </table>	A	Q	0	1	1	0
A	Q							
0	1							
1	0							

Compuerta or o suma logica

OR																						
Símbolo	Expresión	Tabla de verdad	A	B	Q																	
	$Q = A + B$	<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>Q</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	Q	0	0	0	0	1	1	1	0	1	1	1	1					
A	B	Q																				
0	0	0																				
0	1	1																				
1	0	1																				
1	1	1																				
2^6	2^6	2^5	2^4	2^3	2^2	2^1	2^0															
128	64	32	16	8	4	2	1															
						1	1															

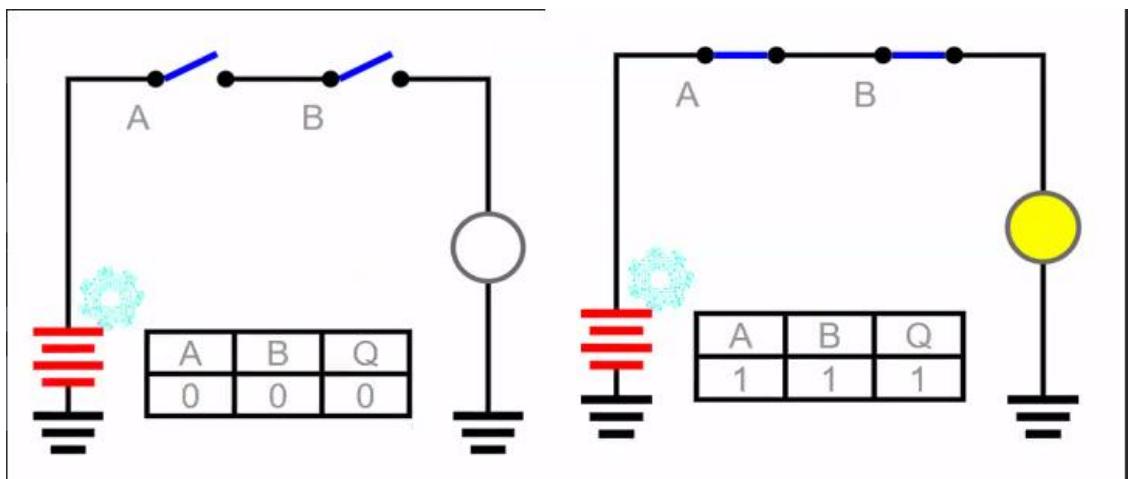
Circuito representativo compuerta or



Compuerta and

Símbolo	Expresión	Tabla de verdad															
	$Q = A \cdot B$	<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>Q</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	Q	0	0	0	0	1	0	1	0	0	1	1	1
A	B	Q															
0	0	0															
0	1	0															
1	0	0															
1	1	1															

Circuito representativo de la compuerta and



1: La ley asociativa:

$$\forall a, b, c \in \mathfrak{B} : (a + b) + c = a + (b + c)$$

$$\forall a, b, c \in \mathfrak{B} : (a \cdot b) \cdot c = a \cdot (b \cdot c)$$



2: Existencia del elemento neutro:

$$\forall a \in \mathfrak{B} : a + 0 = a$$

$$\forall a \in \mathfrak{B} : a \cdot 1 = a$$

3: La ley conmutativa:

$$\forall a, b \in \mathfrak{B} : a + b = b + a$$

$$\forall a, b \in \mathfrak{B} : a \cdot b = b \cdot a$$

Leyes del álgebra de bool

$$1. A + 0 = A$$

$$7. A \cdot A = A$$

$$2. A + 1 = 1$$

$$8. A \cdot \bar{A} = 0$$

$$3. A \cdot 0 = 0$$

$$9. \bar{\bar{A}} = A$$

$$4. A \cdot 1 = A$$

$$10. A + AB = A$$

$$5. A + A = A$$

$$11. A + \bar{A}B = A + B$$

$$6. A + \bar{A} = 1$$

$$12. (A + B)(A + C) = A + BC$$

Leyes de De Morgan

$$\forall a, b \in \mathfrak{B} : \overline{a + b} = \bar{a} \cdot \bar{b}$$

$$\forall a, b \in \mathfrak{B} : \overline{a \cdot b} = \bar{a} + \bar{b}$$

Resolución 1er ejercicio

Ejercicios de simplificación de funciones lógicas

Resolver:

$$AB + A(B+C) + B(B+C)$$

$$AB+AC+B+BC$$

$$AB+AC+B$$

$$B+AC$$

Salida: ▾

Format: Sum of products ▾

		B, C			
		00	01	11	10
A	0	0	0	1	1
	1	0	1	1	1
		B + AC			

[Fijar Como Expresión](#)

$$AB' + A'B + AB$$

$$A \sim B + B(\sim A + A)$$

$$A \sim B + B$$

$$A+B$$

Salida: ▾

Format: Sum of products ▾

		B	
		0	1
A	0	0	1
	1	1	1
		B + A	

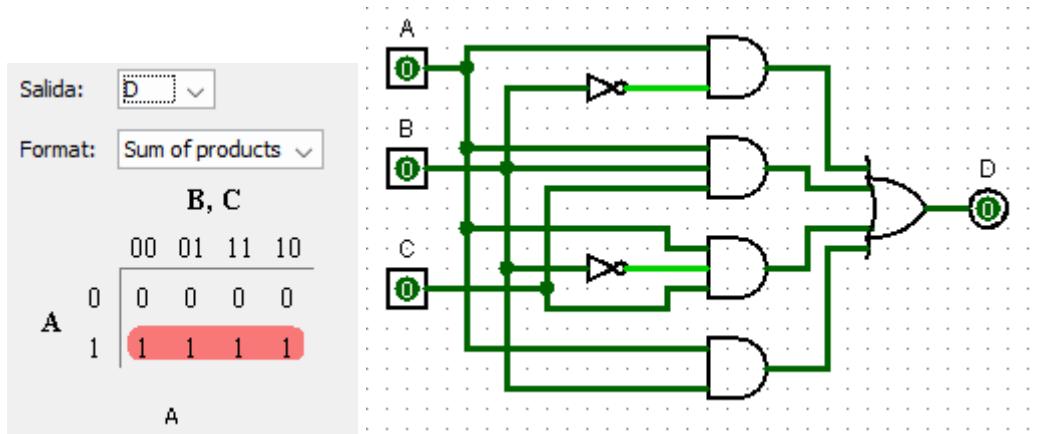
$$AB' + ABC + AB'C + AB$$

$$A \sim B + AB(C + C') + AB$$

$$A \sim B + AB + AB$$

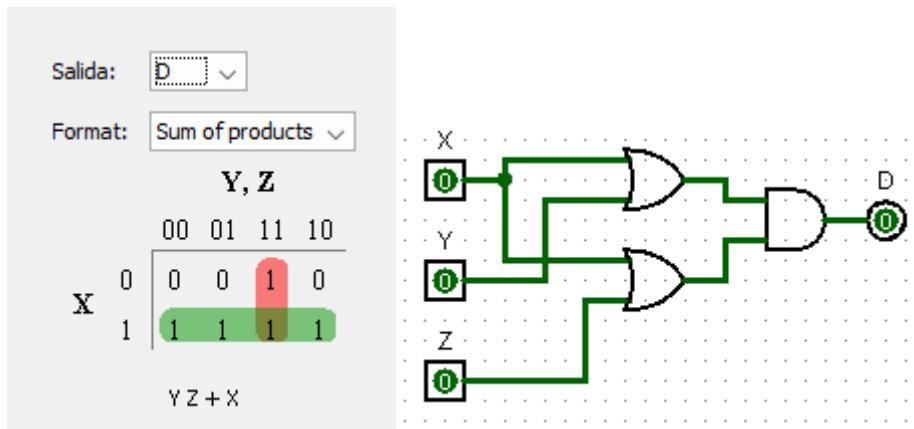
$$A + AB$$

$$A$$



$$(X+Y)(X+Z)$$

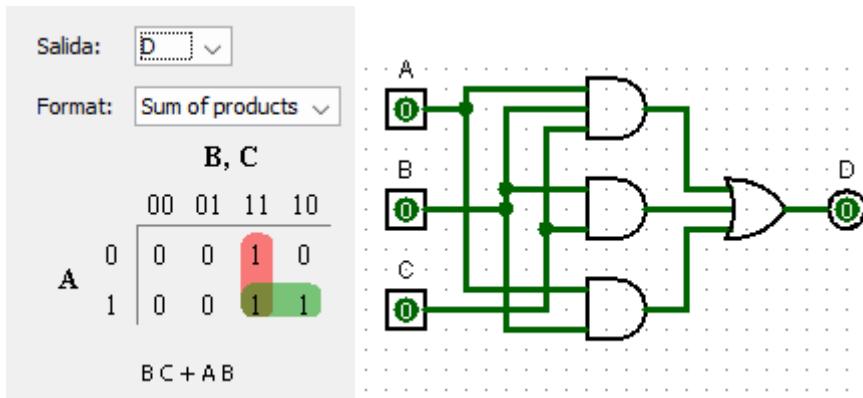
$$YZ + X$$



$$(A \cdot B \cdot C) + (B \cdot C) + (A \cdot B)$$

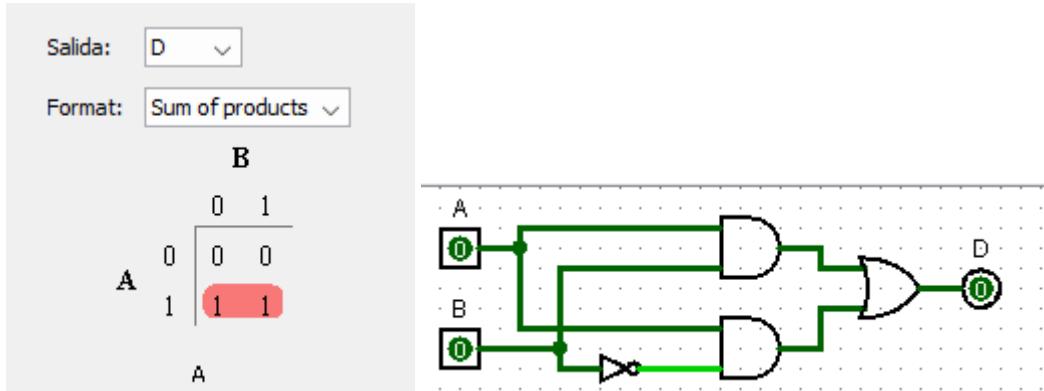
$$B(AC+C)+(AB)$$

$$BC+AB$$



$$AB + AB' = A$$

A



$$(A+B) * (A+B')$$

$$AA + A \sim B + BA + B \sim B$$

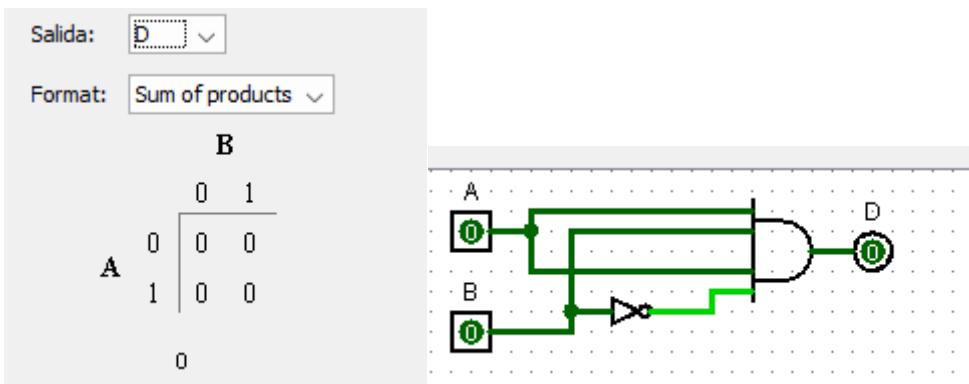
$$A + A \sim B + BA + B \sim B$$

$$A + A \sim B + BA + 0$$

$$A(B + \sim B)$$

$$A(1)$$

$$0$$



$$\overline{(AB+C)(A+BC)}$$

$$\sim((AB+C)(A+BC))$$

$$\sim(ABA+ABBC+CA+CBC)$$

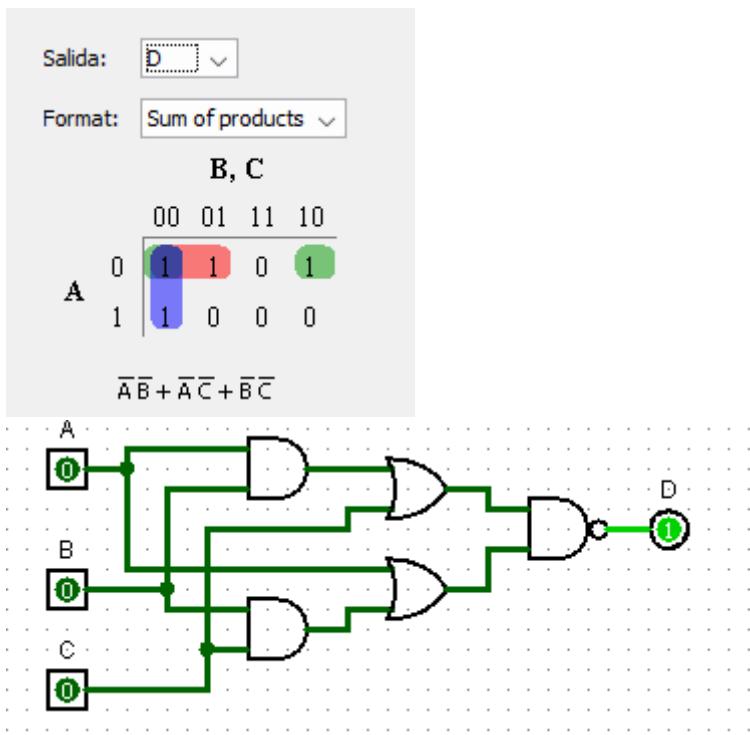
$$\sim(AB+ABC+CA+CB)$$

$$\sim(A(B+BC)+CA+CB))$$

$$\sim(A+CA+CB)$$

$$\sim (A+CB)$$

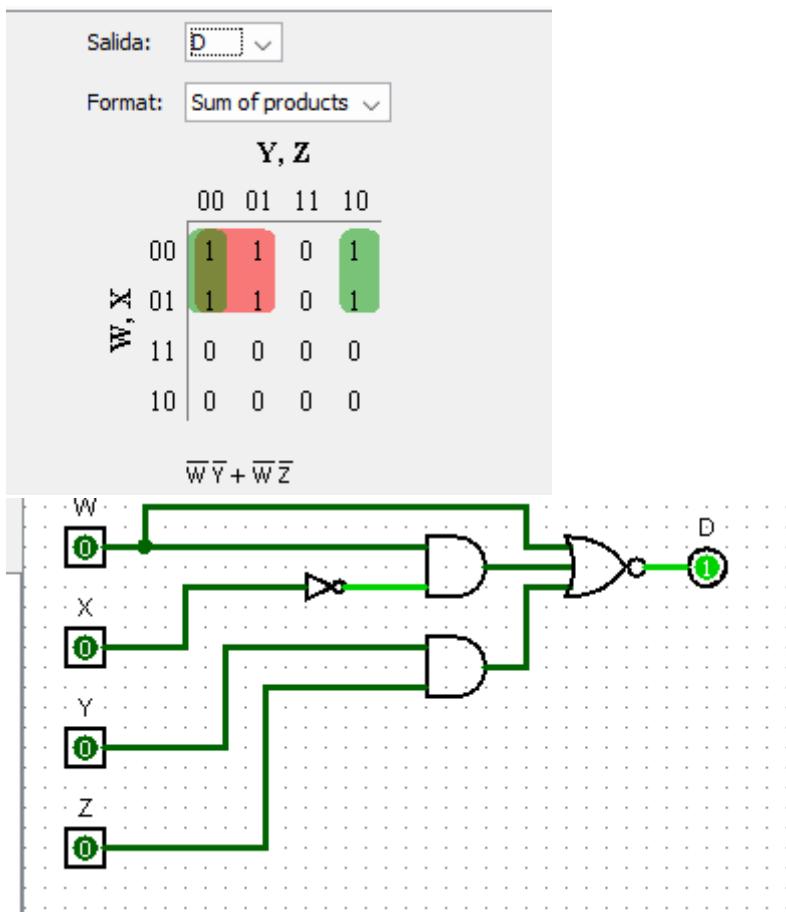
$$\sim A$$



$$(W+WX'+YZ)'$$

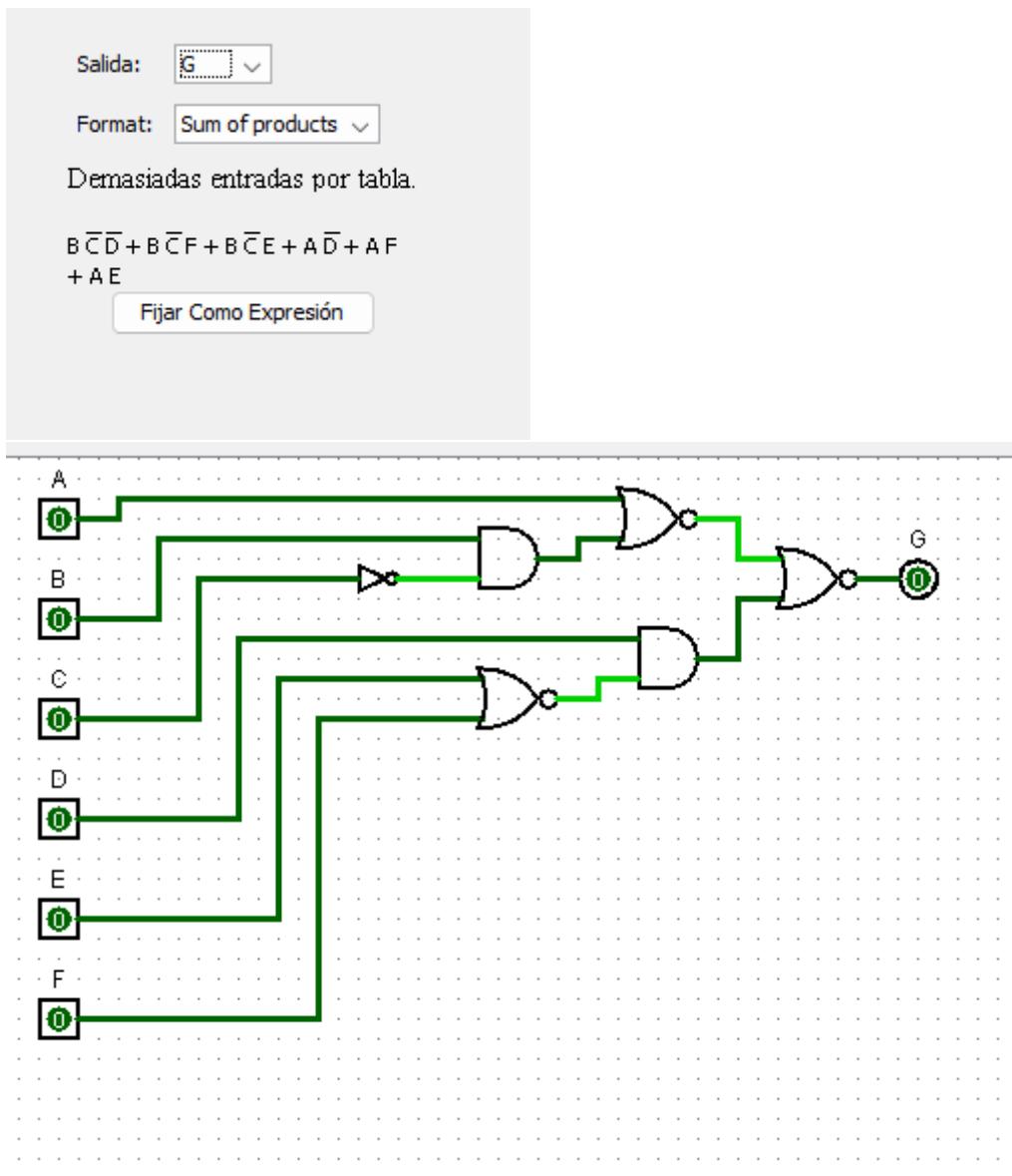
$$\sim (W+W^{\sim}X+YZ)$$

$$\sim W^{\sim}Y + \sim W^{\sim}Z$$



$$\overline{A+B\overline{C}} + \overline{D(E+F)}$$

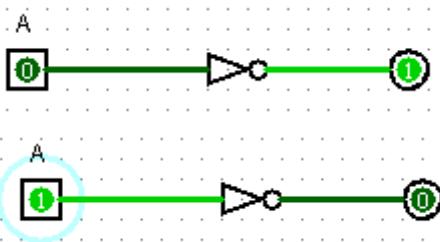
B~C~D+B~CF+B~C+A~D+AF+AE



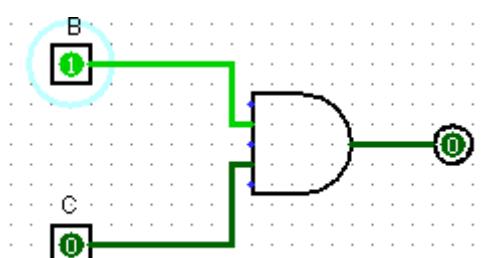
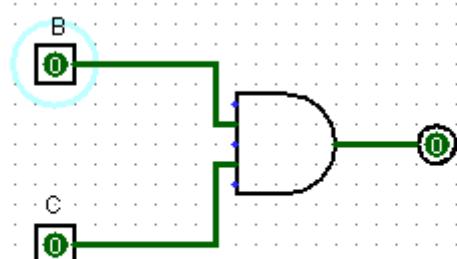
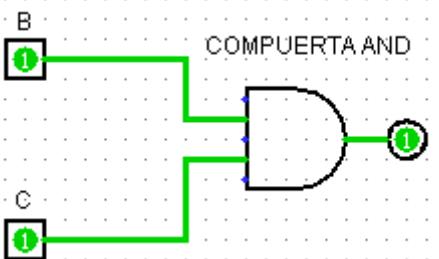
Clase 28/08/2024

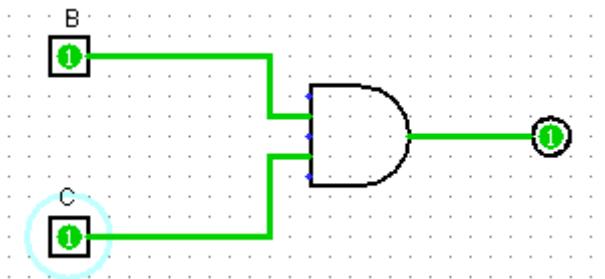
Circuitos combinacionales

SIMULACIÓN COMPUERTA NOT INVIERTE LA SALIDA

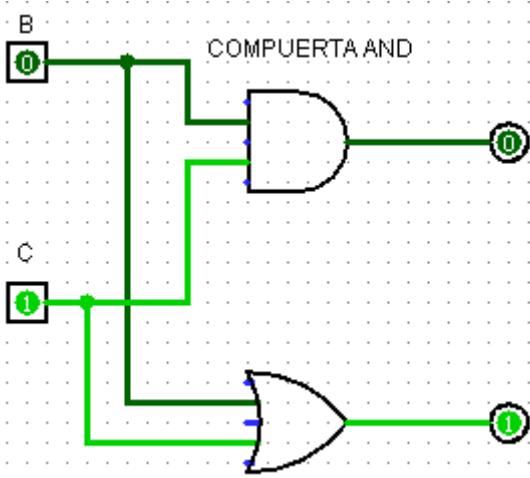
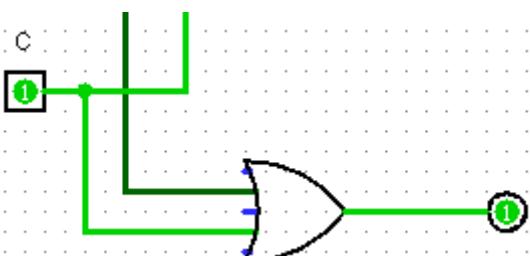


SIMULACION COMPUERTA AND MULTIPLICA LA SALIDA

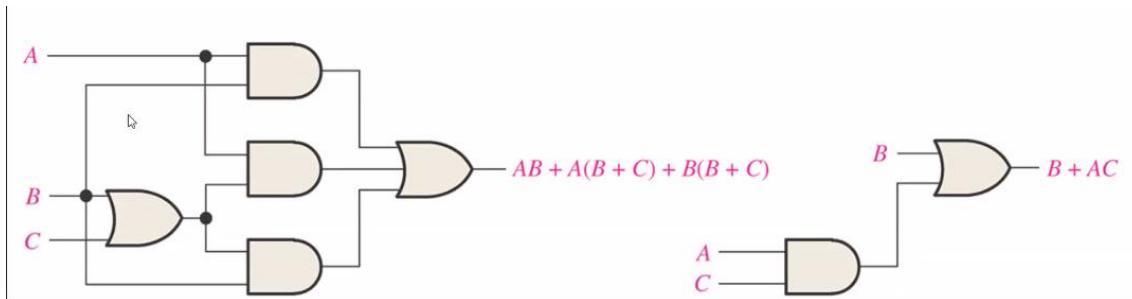




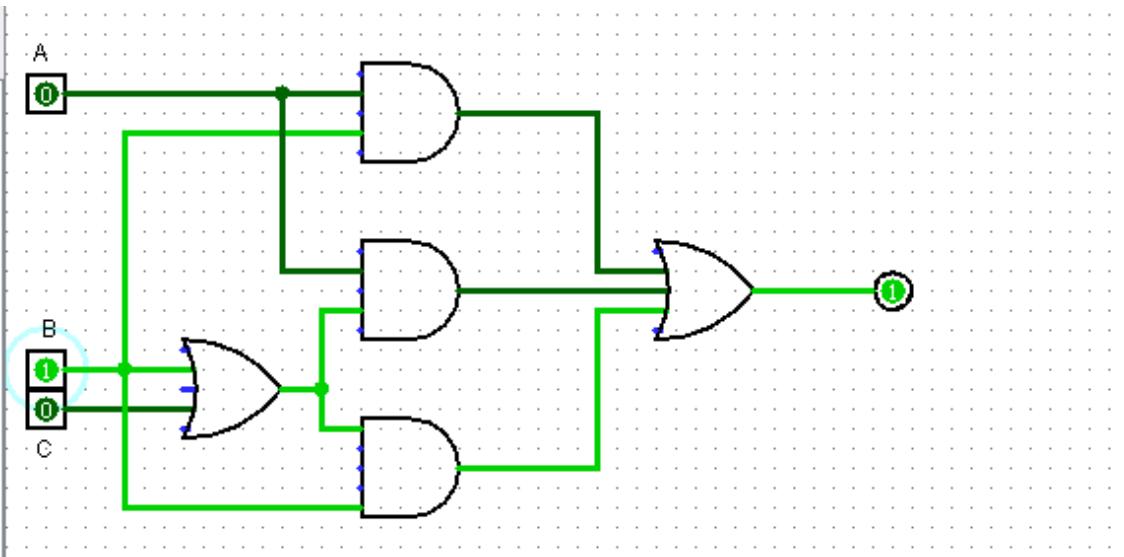
SIMULACION COMPUERTA OR SUMA LA SALIDA



SIMULACION DE CIRCUITO



EN QUE COMBINACION ME DA COMO RESULTADO UN 1 LOGICO



R: LA COMBINACION QUE ME DA UN 1 LOGICO ES 010

CIRCUITOS INTEGRADOS

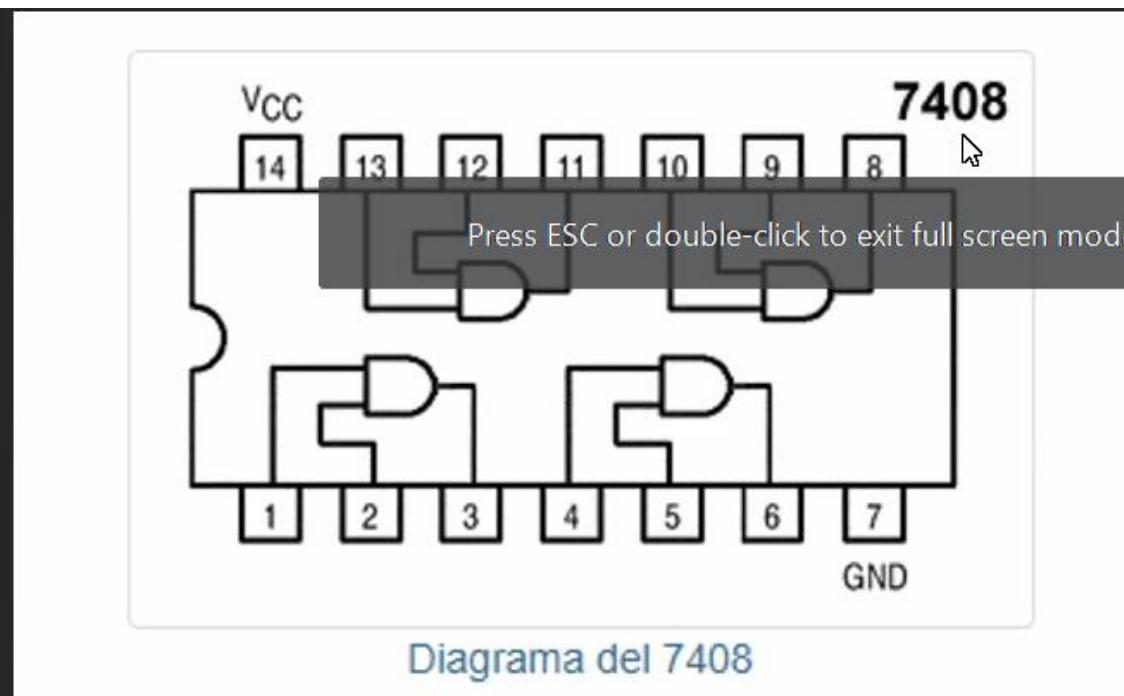
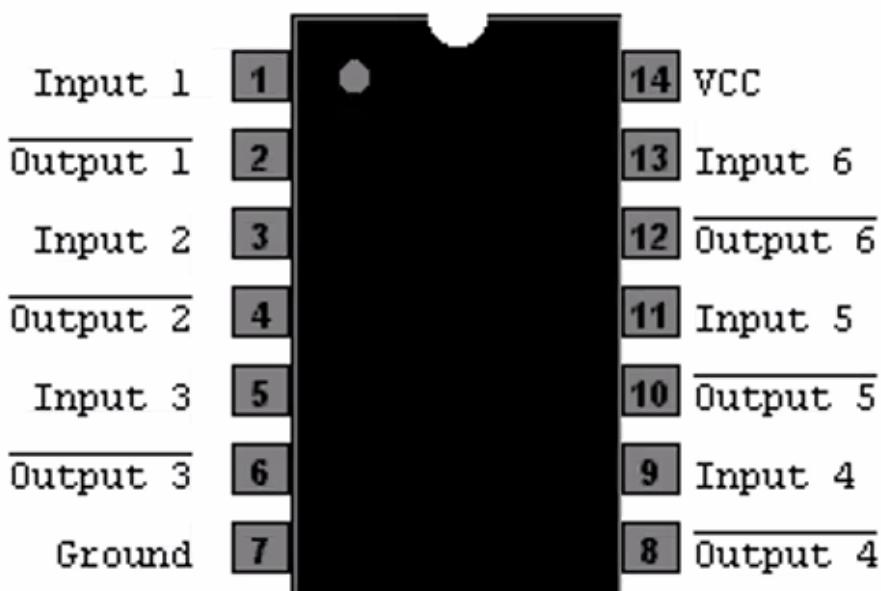
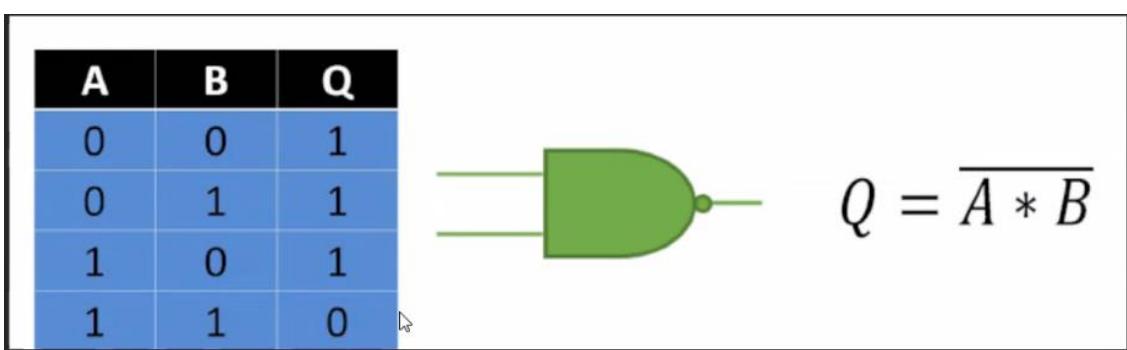


Diagrama del 7404



COMPUERTA NAND INVIerte EL VALOR DEL RESULTADO DE LA MULTIPLICACION



COMPUERTA NOR

2. Compuerta NOR

Así como vimos anteriormente, la compuerta OR también tiene su versión inversa. Esta compuerta cuando tiene sus entradas en estado 0 su salida estará en 1, pero si alguna de sus entradas pasa a un estado 1 sin importar en qué posición, su salida será un estado 0.

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0



$$Q = \overline{A} + B$$

3. Compuerta XOR

También llamada OR exclusiva, esta actúa como una suma binaria de un dígito cada uno y el resultado de la suma sería la salida. Otra manera de verlo es que con valores de entrada igual el estado de salida es 0 y con valores de entrada diferentes, la salida será 1.

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0



$$Q = A * \bar{B} + \bar{A} * B$$

Compuerta XNOR

Esta es todo lo contrario a la compuerta XOR, ya que cuando las entradas sean iguales se presentara una salida en estado 1 y si son diferentes la salida será un estado 0.

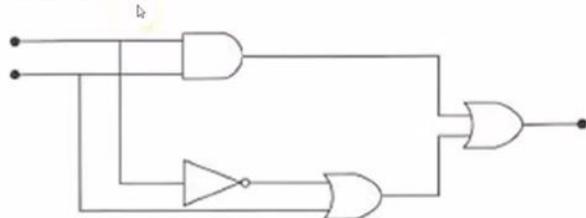
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1



$$Q = A * B + \overline{A} * \overline{B}$$

Ejercicios compuertas lógicas

Para el circuito mostrado en la figura: (a) identificar cuántas entradas y salidas posee, (b) identificar las compuertas lógicas presentes, (c) obtener la tabla de verdad, y (d) representar el circuito por medio de una ecuación.



RESPUESTA A)

2 ENTRADAS Y UNA SALIDA

RESPUESTA B)

2 COMPUERTAS OR

1 COMPUERTA AND

1 COMPUERTA NOT

RESPUESTA C)

TABLA DE VERDAD

A	B	$A \cdot B$	A'	R	B	F
0	0	0	1	1	0	1
0	1	0	1	1	1	1
1	0	0	0	0	0	0
1	1	1	0	0	1	1

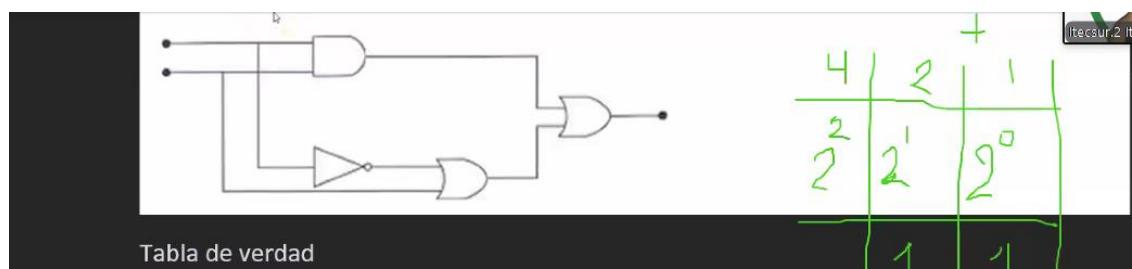
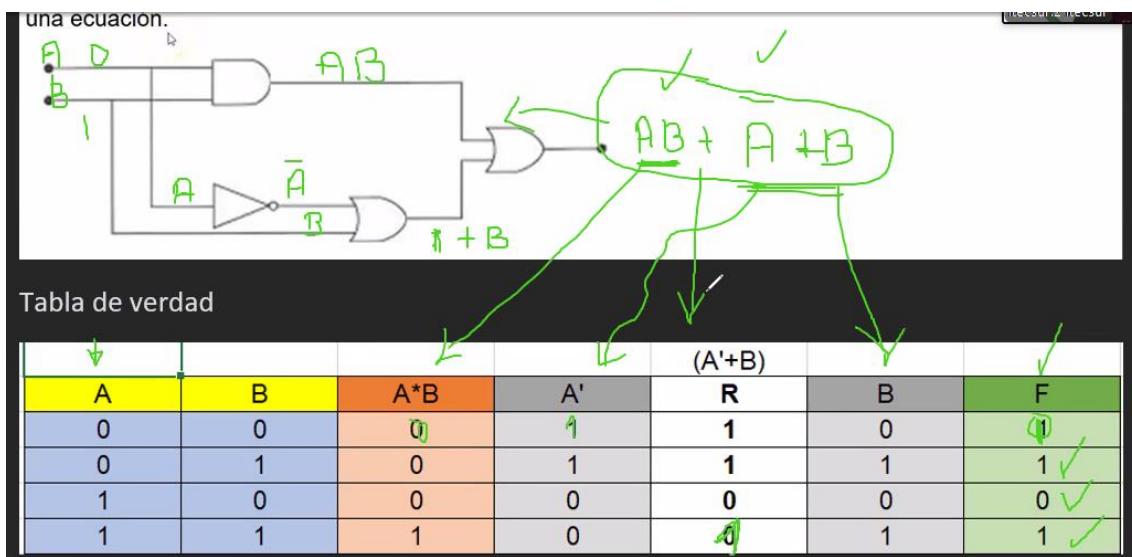


Tabla de verdad

D	A	B	$A \cdot B$	A'	R	B	F
0	✓	0	0	1	1	0	1
1	✓	1	0	1	1	1	1
2	✓	0	0	0	0	0	0
3	1	1	1	0	0	1	1



RESPUESTA D)

$$C = A + B(A' + B)$$

CONVERSIÓN A BINARIO

2*2*2			4 CEROS 8 4 UNOS		
A	B	C	DECIMAL		
0	0	0	0		
0	0	1	1	1	2
0	1	0		4	
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1	7		

BIT menos sig	bit mas signi
2^0	2^5
1	32
2^1	2^4
2	16
2^2	2^3
4	8
2^3	2^2
8	4
2^4	2^1
16	2
2^5	2^0
32	1

BMAS-S	BMENOS-S	1	1	1	0
		1	0	0	1

BIT MENOS SIGNIFICATIVO							BIT MAS SIGNIFICATIVO
	1	0	1	0	1	1	1

011110101

BIT MENOS SIGNIFICATIVO							BIT MAS SIGNIFICATIVO
	1	0	1	0	1	1	1

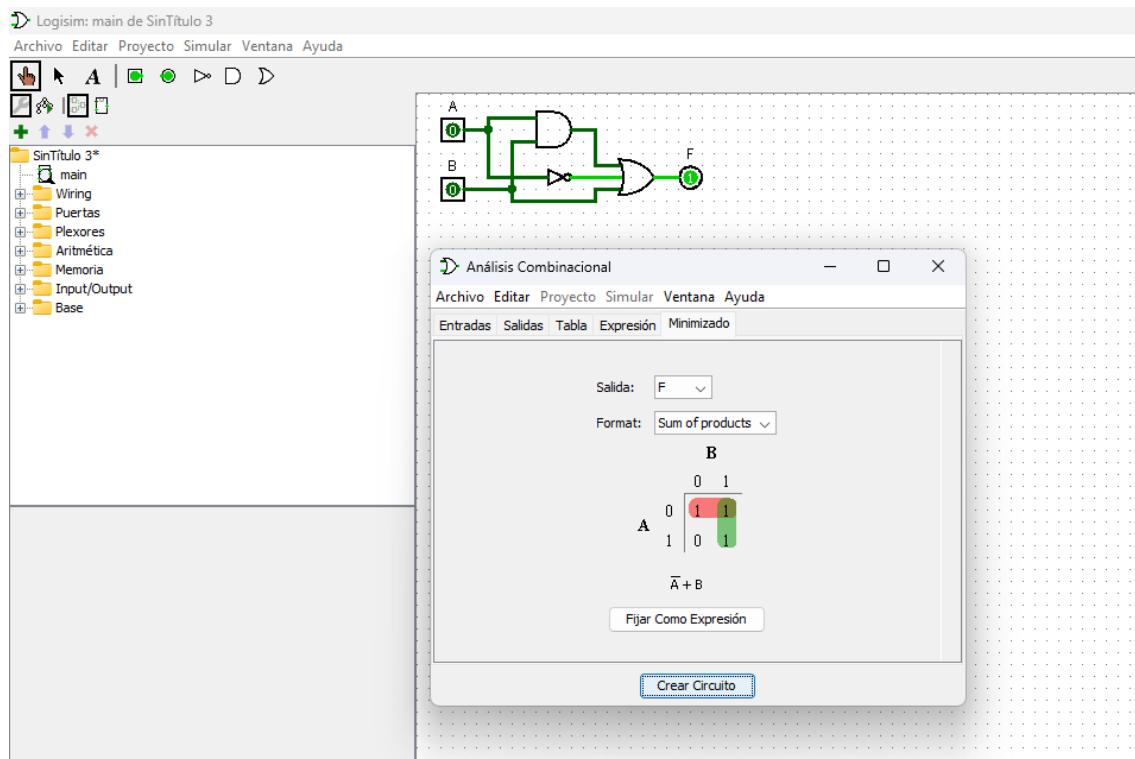
1	2	4	8	16	32	64	128
1	0	1	0	1	1	1	1



245

011110101	245								
BIT MENOS SIGNIFICATIVO					BIT MAS SIGNIFICATIVO				
1	2	4	8	16	32	64			128
1	0	1	0	1	1	1			1
1	0	1	0	1	1	1			1

CREACION DE CIRCUITO A PARTIR DE UNA EXPRESION



Clase 2/09/2024 tablas de verdad, representaciones y simplificación de términos.

Suma de productos

$$AB + \bar{A}BC + ABC\bar{C} + BC$$

La primera forma canónica es una representación de una función lógica que utiliza una suma de productos canónicos.

A	B	C	F
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1

Salida: F

$AB + \bar{A}BC + ABC\bar{C} + BC$

(A AND NOT B) OR (NOT A AND B AND C) OR (A AND B NOT C) OR (B AND C)

Suma de productos

$$AB + \bar{A}BC + ABC\bar{C} + BC$$

10 011 110 111

La primera forma canónica es una representación de una función lógica que utiliza una suma de productos canónicos.

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Salida: F

$AB + \bar{A}BC + ABC\bar{C} + BC$

(A AND NOT B) OR (NOT A AND B AND C) OR (A AND B NOT C) OR (B AND C)

Análisis Combinacional

Archivo Editar Proyecto Simular Ventana Ayuda

Entradas Salidas Tabla Expresión Minimizado

Salida: D

Formato: Sum of products

B, C

A	0	0	1	0
1	1	1	1	1

BC + A

Crear Circuito

A

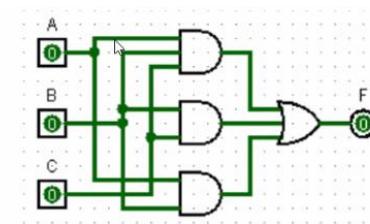
ACTIVIDAD EN CLASE



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

Realizar el siguiente ejercicio

$$(ABC) + (BC) + (A*B)$$



Salida: ▾
 $ABC + BC + AB$

(A AND B AND C) OR (B AND C) OR (A AND B)

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Salida: ▾
Format: ▾

B, C

	00	01	11	10
A	0	0	1	0
	1	0	0	1

$BC + AB$



Análisis Combinacional

Archivo Editar Proyecto Simular Ventana Ayuda

Entradas Salidas Tabla Expresión Minimizado

Salida: D

$A \bar{B} C + \bar{B} C + A \bar{B}$

(A AND B AND C) OR (B AND C) OR (A AND B)

Limpiar Recargar Intro

Crear Circuito

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



Análisis Combinacional

Archivo Editar Proyecto Simular Ventana Ayuda

Entradas Salidas Tabla Expresión Minimizado

Salida: D

Format: Sum of products

Y, Z

	00	01	11	10
X	0	0	0	1
1	1	1	0	1

$YZ + X\bar{Y}$

Fijar Como Expresión

Crear Circuito

Análisis Combinacional

Archivo Editar Proyecto Simular Ventana Ayuda

Entradas Salidas Tabla Expresión Minimizado

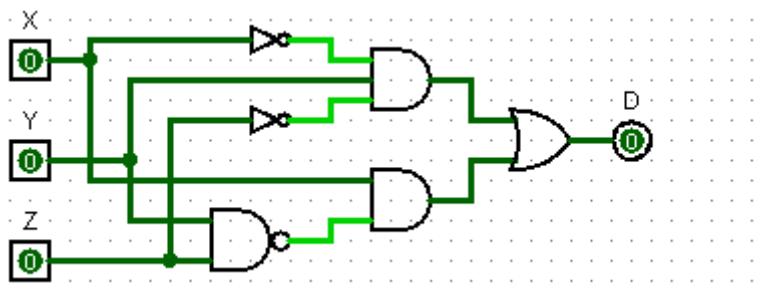
Salida: D

$\bar{X}YZ + X\bar{Y}Z$

$\sim X \text{ AND } Y \text{ AND } \sim Z \text{ OR } X \text{ AND } \sim(Y \text{ AND } Z)$

Limpiar Recargar Intro

Crear Circuito



Producto de sumas

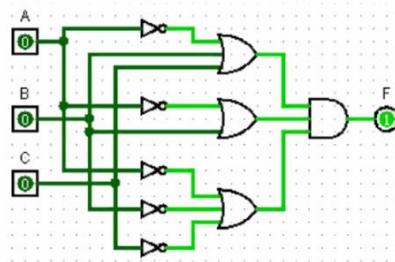
$$(\bar{A} + B + C)(\bar{A} + B)(\bar{A} + \bar{B} + \bar{C})$$

Salida: F
 $(\bar{A} + B + C)(\bar{A} + B)(\bar{A} + \bar{B} + \bar{C})$

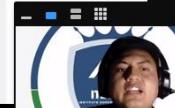
(NOT A OR B OR C) AND (NOT A OR B) (NOT A OR NOT B OR NOT C)

La función lógica se expresa como un producto de sumas canónicas, es decir, sumas que contienen las variables de la función en su forma "normal" o complementada.

A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0



Salida: F
Format: Sum of products
B, C
A 0 | 1 1 1 1
1 | 0 0 0 1
 $\bar{A} + B \bar{C}$



Análisis Combinacional

Archivo Editar Proyecto Simular Ventana Ayuda

Entradas Salidas Tabla Expresión Minimizado

Salida: D
Format: Sum of products
B, C
00 01 11 10
A 0 | 1 1 1 1
1 | 0 0 0 1
 $\bar{A} + B \bar{C}$

Fijar Como Expresión

Crear Circuito



Análisis Combinacional

Archivo Editar Proyecto Simular Ventana Ayuda

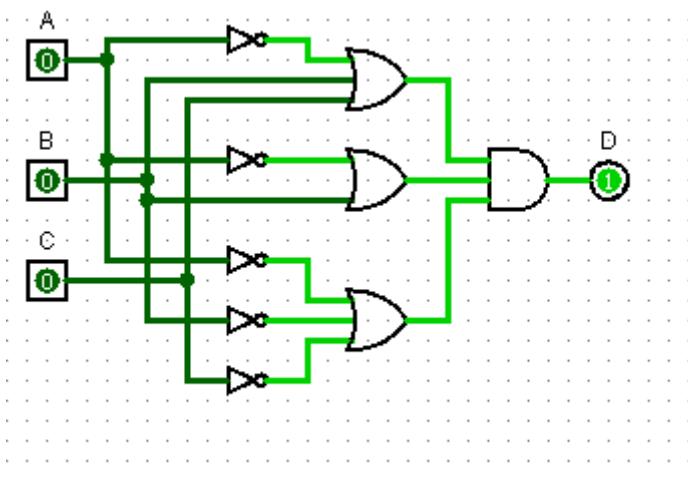
Entradas Salidas Tabla Expresión Minimizado

Salida: D

$(\bar{A} + B + C)(\bar{A} + B)(\bar{A} + \bar{B} + \bar{C})$

$(\sim A \text{ OR } B \text{ OR } C) \text{ AND } (\sim A \text{ OR } B) \text{ AND } (\sim A \text{ OR } \sim B \text{ OR } \sim C)$

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



Obtener la función booleana a partir de la tabla de verdad: Mintérminos



A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Mintérminos: Son todas las combinaciones de entrada que dan como salida 1

$$F = (\bar{A}\bar{B}\bar{C}) + (\bar{A}\bar{B}C) + (A\bar{B}\bar{C}) + (ABC)$$

$$010 \quad F = AC + A'B$$

011

101

Los minterminos y maxterminos son conceptos utilizados para representar funciones lógicas. Los minterminos de una función son todas las combinaciones para las cuales la función toma el valor de 1, mientras que los maxterminos son todas las combinaciones para las cuales la función toma el valor de 0

111



Obtener la función booleana a partir de la tabla de verdad

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

000
001
100
110

$$F = (A+B+C)*(A+B+\bar{C})*\\(\bar{A}+B+C)*(\bar{A}+\bar{B}+C)$$

$$F = (ABC) + (BC)$$

Maxterminos: Son todas las combinaciones de entrada que dan como salida 0

Obtener la función booleana a partir de tabla de verdad

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

000
001
100
110
 $\bar{A} \bar{B} \bar{C}$
 $\bar{A} \bar{B} C$
 $\bar{A} B \bar{C}$
 $A \bar{B} \bar{C}$

$$D = \bar{A} \bar{B} \bar{C}$$

$$1 = \text{output}$$

$$F = (A+B+C)*(A+B+\bar{C})*\\(\bar{A}+B+C)*(\bar{A}+\bar{B}+C)$$

$$\bar{A} \bar{B} \bar{C}$$

$$\bar{A} \bar{B} \bar{C} = A + B + C$$

Maxterminos: Son todas las combinaciones de entrada que dan como salida 0



Mapas K con dos variable

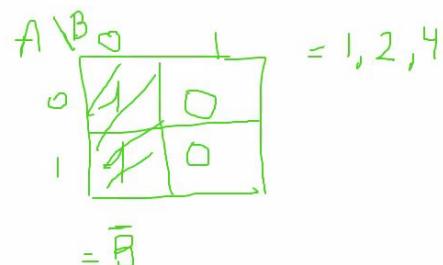


A	B	AB		
0	0	0		
0	1	1		
1	0	0		
1	1	1		

	B		
A		0	1
0		0	1
1		0	1

Press ESC or double-click to exit full screen mode

Ejercicio: $AB' + A'B + AB$



Análisis Combinacional

Archivo Editar Proyecto Simular Ventana Ayuda

Entradas Salidas Tabla Expresión Minimizado

Salida: D

$A\bar{B} + \bar{A}B + AB$

A AND ~B OR ~A AND B OR A AND B

Limpiar Recargar Intro

Crear Circuito



Análisis Combinacional

Archivo Editar Proyecto Simular Ventana Ayuda

Entradas Salidas Tabla Expresión Minimizado

Salida: D

Format: Sum of products

B

	0	1
0	0	1
1	1	1

A

B + A

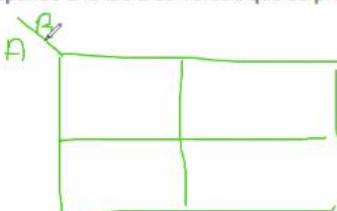
Fijar Como Expresión

Crear Circuito

Ejercicio

1.- Indica cuál de las funciones siguientes corresponde a la tabla de verdad que se presenta:

a	b	f
0	0	1
0	1	0
1	0	1
1	1	1



- a) $f = a + b + c$
- b) $f = a + b'$
- c) $f = a \cdot b$



Análisis Combinacional

Archivo Editar Proyecto Simular Ventana Ayuda

Entradas Salidas Tabla Expresión Minimizado

Salida: D

Format: Sum of products

B

A

	0	1
0	1	0
1	1	1

$\bar{B} + A$

Fijar Como Expresión

Crear Circuito

Mapas K con tres variables



Ejercicio
 $F = AB' + ABC + AB'C + AB$

- a) $AB + BC + A'B'C'$
- b) $B' + A'C' + AC$
- c) $A'B + A'C' + AB'D$
- d) $D' + AB'C + BC'$

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

BC		[0...0]	[0...1]	[1...1]	[1...0]
A		0	0	1	1
0	0	0	0	1	1
1	0	1	1	1	0



Análisis Combinacional

Archivo Editar Proyecto Simular Ventana Ayuda

Entradas Salidas Tabla Expresión Minimizado

Salida: D

Format: Sum of products

B, C

	00	01	11	10
A	0	0	0	0
	1	1	1	1

A

Fijar Como Expresión

Crear Circuito

HACER ESTOS EJERCICIOS

- a) $AB + BC + A'B'C'$
- b) $B' + A'C' + AC$
- c) $A'B + A'C' + AB'D$
- d) $D' + AB'C + BC'$

$$AB + BC + \sim A \sim B \sim C$$



Análisis Combinacional

Archivo Editar Proyecto Simular Ventana Ayuda

Entradas Salidas Tabla Expresión Minimizado

Salida: D

Format: Sum of products

B, C

	00	01	11	10
A	0	1	0	1
1	0	0	1	1

$\bar{A}\bar{B} + B\bar{C} + A\bar{B}$

Fijar Como Expresión

Crear Circuito

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

$$\sim B + \sim A \sim C + AC$$

Análisis Combinacional

Archivo Editar Proyecto Simular Ventana Ayuda

Entradas Salidas Tabla Expresión Minimizado

Salida: D

Format: Sum of products

B, C

	00	01	11	10
A	0	1	1	0
1	1	1	1	0

$\bar{A}\bar{C} + \bar{B} + AC$

Fijar Como Expresión

Crear Circuito

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

$$\sim AB + \sim A \sim C + A \sim BD$$

Salida: E

Format: Sum of products

C, D

	00	01	11	10
B, A	00	1 1 0 0		
	01	1 1 1 1		
	11	0 0 0 0		
	10	0 1 1 0		

$\bar{A}\bar{C} + \bar{A}B + A\bar{B}D$

Fijar Como Expresión

A	B	C	D	E
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

 $\sim D + A\sim BC + B\sim C$

Salida:	E							
Format:	Sum of products							
C, D								
00	01	11	10	A	B	C	D	E
00	1	0	0	1	0	0	0	1
01	1	1	0	1	0	0	0	0
11	1	1	0	1	0	1	0	1
10	1	0	1	1	1	0	1	0
$\bar{D} + B\bar{C} + A\bar{B}\bar{C}$		0	1	0	0	1	1	1
Fijar Como Expresión		1	1	1	0	1	0	0

 $\sim A\sim BCD + \sim AB\sim C\sim D + AB\sim C\sim D + ABCD + AB\sim C\sim D + \sim A\sim B\sim CD + A\sim BC\sim D$

Salida:	E							
Format:	Sum of products							
C, D								
00	01	11	10	A	B	C	D	E
00	0	1	1	0	0	0	0	0
01	1	0	0	0	0	0	1	1
11	1	1	1	0	0	1	0	0
10	0	0	0	1	0	0	0	0
$\bar{A}\bar{B}\bar{D} + B\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + ABD$		0	1	0	0	1	1	1
Fijar Como Expresión		1	1	0	1	0	1	0



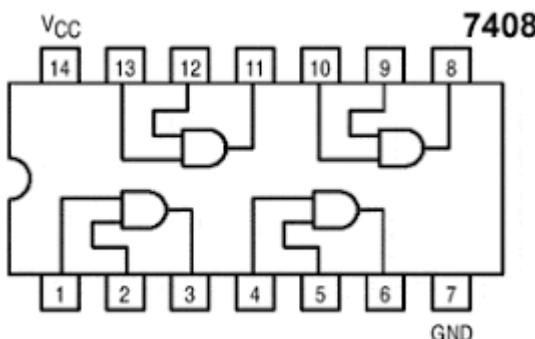
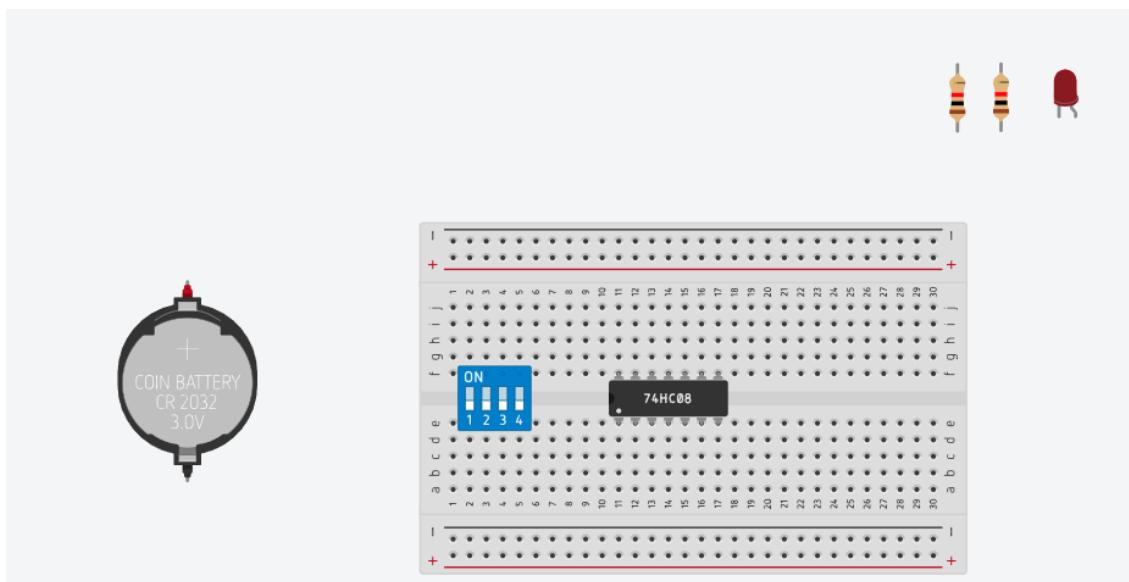
Mapas K de 4 variables

A	B	C	D	F	
0	0	0	0	0	$\bar{A}B\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}CD + A\bar{B}CD + A\bar{B}CD + A\bar{B}CD$
0	0	0	1	1	
0	0	1	0	0	
0	0	1	1	1	
0	1	0	0	0	
0	1	0	1	0	
0	1	1	0	1	
0	1	1	1	0	
1	0	0	0	1	
1	0	0	1	0	
1	0	1	0	1	
1	0	1	1	0	
1	1	0	0	1	
1	1	0	1	0	
1	1	1	0	1	
1	1	1	1	1	

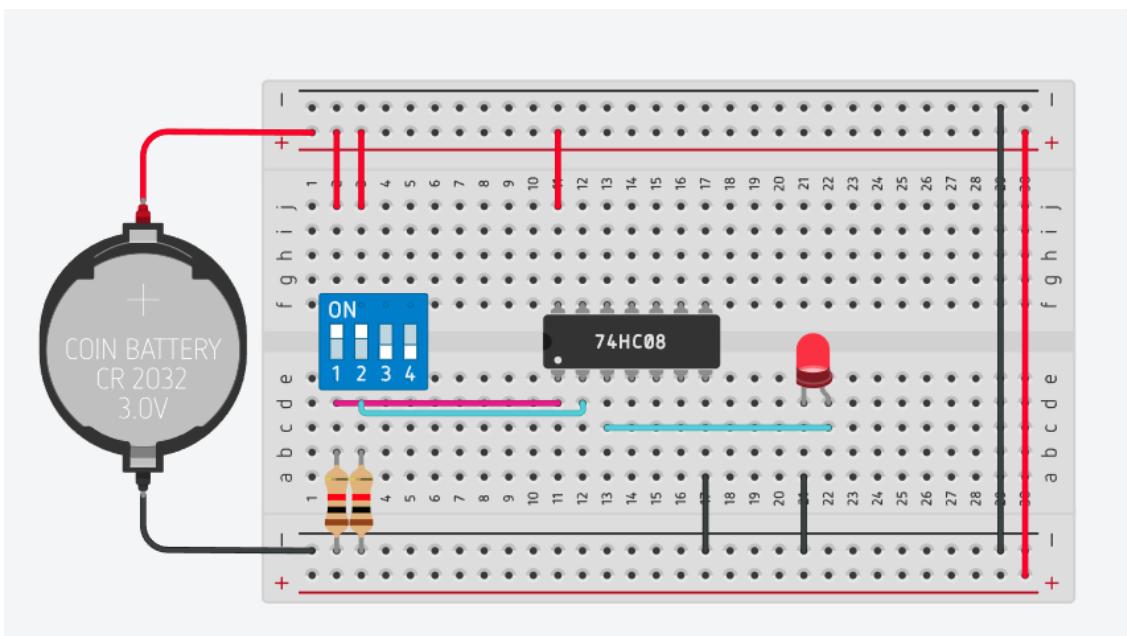
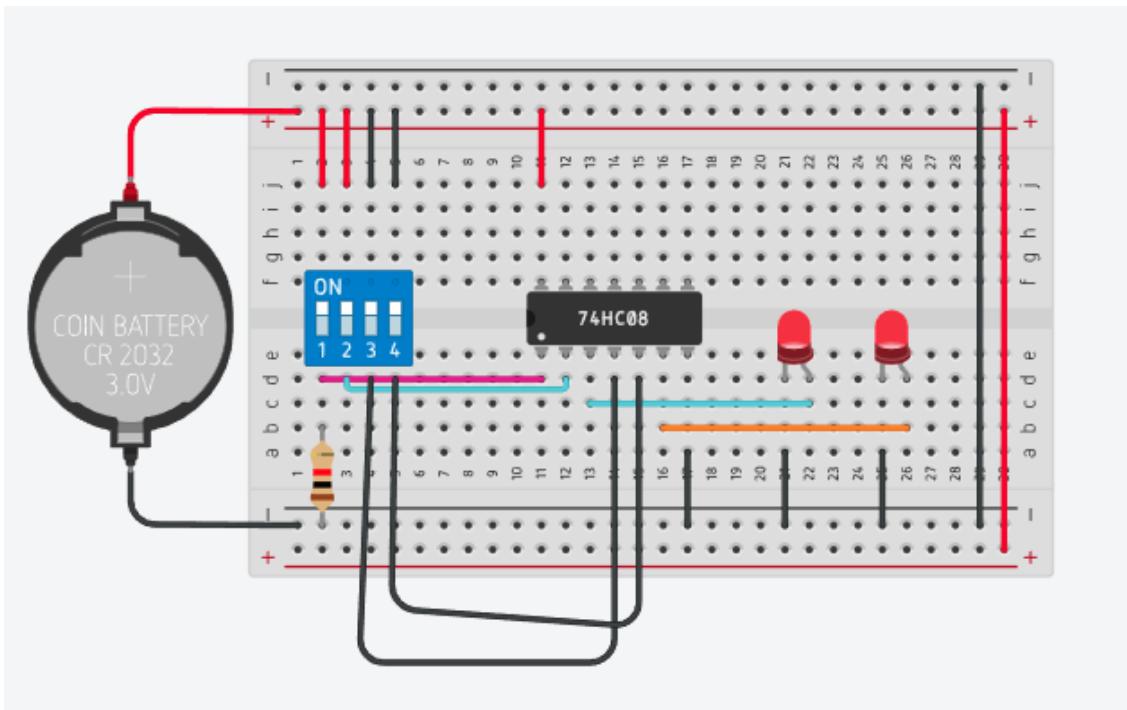
AB \ CD	[0...0]	[0...1]	[1...1]	[1...0]
[0...0]	0	1	1	0
[0...1]	0	0	0	1
[1...1]	1	0	1	1
[1...0]	1	0	0	1

$$AD' + A'B'D + A'B$$

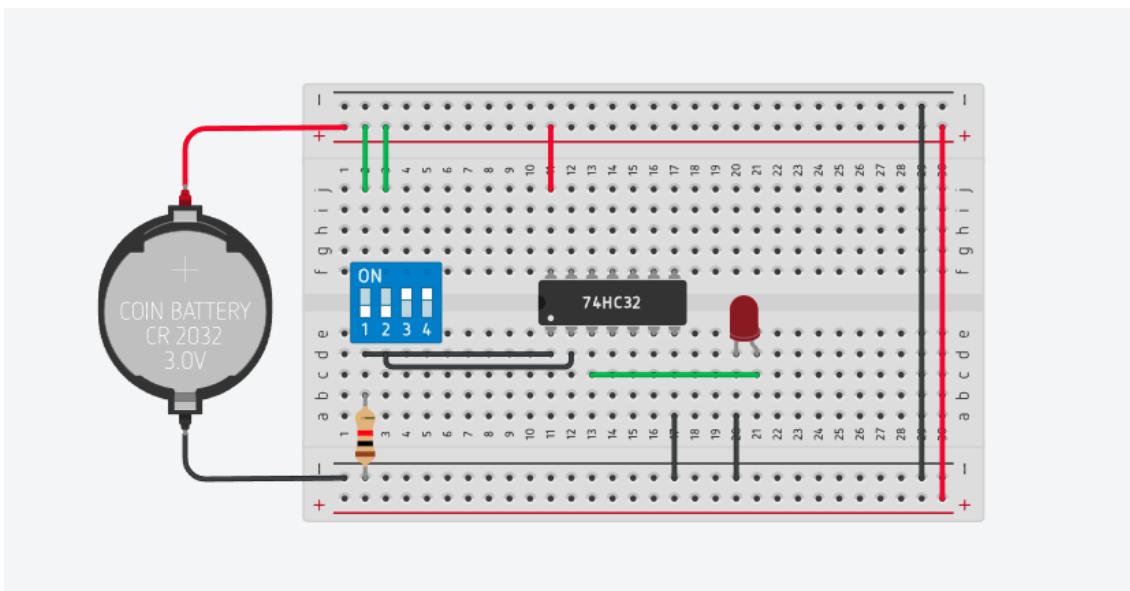
CLASE 04/09/2024 TINKERCAT



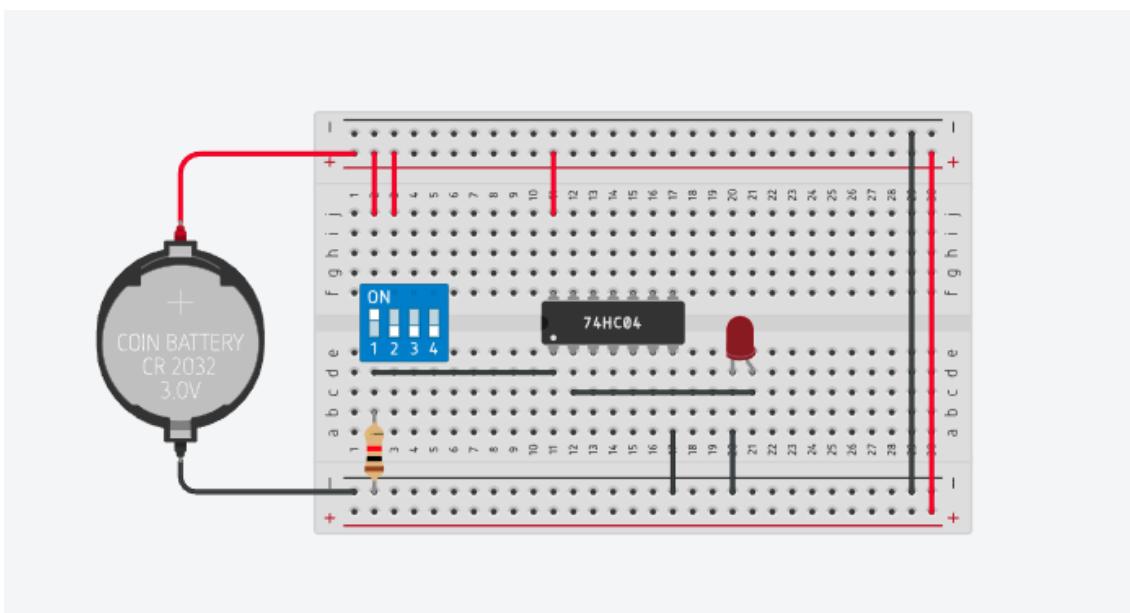
COMPROBACION COMPUERTA AND

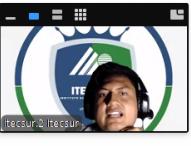


Simular el comportamiento de la compuerta or



Simulación de la compuerta not





binario	octal	hexadecimal	decimal
11010101	325	D5	213
100101010	452	12A	298
10001001	211	89	137
1100011	143	63	99

DECIMAL	HEX	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Sistemas de representación numérica



• Sistema Decimal:

- Es el sistema más utilizado en la vida diaria.
- Utiliza 10 símbolos (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) para representar cualquier cantidad.
- El valor de un número depende de su posición y del valor de la base (10).
- Ejemplo: $532 = 5 \times 10^2 + 3 \times 10^1 + 2 \times 10^0 = 532$.

Sistemas de representación numérica



• 2. Sistema Binario:

- Se utiliza en informática y electrónica.
- Utiliza solo 2 símbolos (0 y 1) para representar cualquier cantidad.
- El valor de un número depende de su posición y del valor de la base (2).
- Ejemplo: $1011 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11$.



02:14:38

Sistemas de representación numérica



• 3. Sistema Octal:

- Se utiliza en algunos sistemas informáticos y de telecomunicaciones.
- Utiliza 8 símbolos (0, 1, 2, 3, 4, 5, 6, 7) para representar cualquier cantidad.
- El valor de un número depende de su posición y del valor de la base (8).
- Ejemplo: $753 = 7 \times 8^2 + 5 \times 8^1 + 3 \times 8^0 = 491$.

• 4. Sistema Hexadecimal:

- Se utiliza en informática y diseño web.
- Utiliza 16 símbolos (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) para representar cualquier cantidad.
- El valor de un número depende de su posición y del valor de la base (16).
- Ejemplo: $A2F = 10 \times 16^2 + 2 \times 16^1 + 15 \times 16^0 = 2607$.

Recording 01:44:13 View

Talking: itesur.2 Itesur

Conversión entre bases: Binario a Octal

• Agrupación por tres:

1. Divide el número binario en grupos de tres dígitos empezando por la derecha. Si te quedan dígitos al final que no llegan a formar un grupo de tres, complétalos con ceros a la izquierda.
2. A cada grupo de tres dígitos binarios, asígnale su equivalente en octal utilizando la siguiente tabla:
3. Escribe los valores octales obtenidos debajo de cada grupo de tres dígitos binarios.
4. El número octal final se obtiene leyendo los dígitos octales de izquierda a derecha.
5. Ejercicio 11010101 ,111010111, 1011101101, 111011101101

000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7





Conversión entre bases: Binario a Octal

011 111 001
(3) (7) (1)₃

000 0

001 → 1

010 2

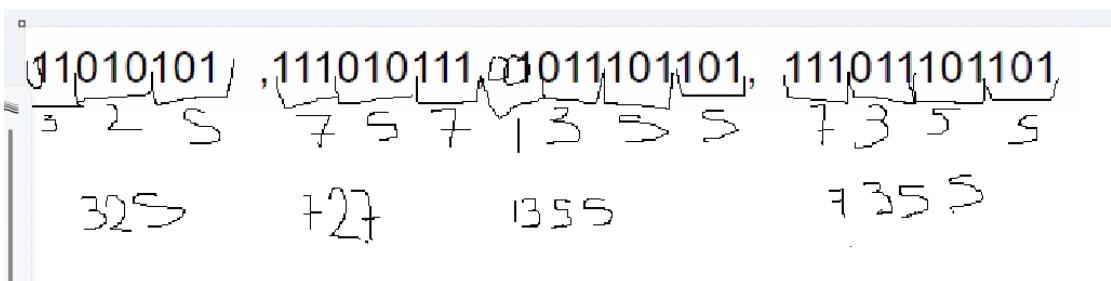
011 3

100 4

101 5

110 6

111 → 7



Conversión entre bases: Binario a Decimal



• Potencia de 2:

1. Escribe el número binario de derecha a izquierda.
 2. Asigna a cada dígito binario una potencia de 2 según su posición: el dígito más a la derecha tiene exponente 0, el siguiente 1, etc.
 3. Multiplica cada dígito binario por su correspondiente potencia de 2.
 4. Suma los productos obtenidos en el paso anterior.
- Ejercicio 1101010111, 11010111, 101101101, 10011010101

1	1	0	1	1
2^4	2^3	2^2	2^1	2^0
16	8	4	2	1
16	8	0	2	1

el número binario 10011010101 es igual a 1237

el número binario 101101101 es igual a 365

el número binario 1101011 es igual a 107

el número binario 1101010111 es igual a 855



Binario a hexadecimal

•Agrupación por cuatro:

1. Divide el número binario en grupos de cuatro dígitos empezando por la derecha. Si te quedan dígitos al final que no llegan a formar un grupo de cuatro, complétalos con ceros a la izquierda.
2. A cada grupo de cuatro dígitos binarios, asígnale su equivalente en hexadecimal utilizando la siguiente tabla:
3. Escribe los valores hexadecimales obtenidos debajo de cada grupo de cuatro dígitos binarios.
4. El número hexadecimal final se obtiene leyendo los dígitos hexadecimales de izquierda a derecha.
5. Ejercicio 1111101011011, 10101101101, 1001101011101



Binario	Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

1111101011011, 10101101101, 1001101011101

F S B 5 6 D 3 5 D

Octal a binario

1. Busca el equivalente binario de cada dígito octal en una tabla de conversión.
2. Escribe los equivalentes binarios de cada dígito octal debajo de cada dígito original.
3. Lee todos los dígitos binarios de izquierda a derecha para obtener el número binario final.
4. Ejemplo 40->100000
5. Ejercicio 44, 54,23 714,1235



000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7



44, 54, 23 714, 1235

100 | 00

| 0 | 1 0 0

0 | 0 0 | |

|| 0 0 | | 0 0

• Ejercicio 45, 23, 125, 145

$$45=37$$

$$23=21$$

$$125=85$$

$$145=101$$



Octal a Hexadecimal



- Convierte el número octal a binario: Cada dígito octal se convierte en su equivalente binario de 3 bits.
1. Por ejemplo, si tienes el número octal 753, se convierte a binario como sigue:
 1. 7 se convierte en 111
 2. 5 se convierte en 101
 3. 3 se convierte en 011
 2. Por lo tanto, el número octal 753 en binario es 111101011.
 3. Agrupa los dígitos binarios en grupos de 4, comenzando desde la derecha. Si el último grupo no tiene 4 dígitos, agregue ceros a la izquierda para completar el grupo.
 4. Convierte cada grupo de 4 dígitos binarios en su equivalente hexadecimal.
 5. Ejercicio 452 ,125,749,456

Hexadecimal a octal



• Agrupar dígitos y convertir individualmente:

1. **Agrupar los dígitos hexadecimales en grupos de 4:** Empezar por la derecha y agrupar los dígitos juntos, añadiendo ceros a la izquierda si es necesario para formar grupos de 4.
2. **Convertir cada grupo a su equivalente octal:** Buscar el equivalente octal para cada grupo hexadecimal en una tabla de conversión o memorizarlos (0000 = 0, 0001 = 1, ..., FFF0 = 777, FFF1 = 7777).
3. **Combinar los equivalentes octales:** Unir los equivalentes octales de cada grupo, manteniendo su orden de derecha a izquierda.
4. Convertir "A3F" a binario: A en binario = 1010 3 en binario = 0011 F en binario = 1111
5. Por lo tanto, "A3F" en binario es "101000111111".
6. Agrupar en grupos de tres de derecha a izquierda: 101 000 111 111
7. Convertir cada grupo binario a su equivalente en octal: 101 en octal = 5 000 en octal = 0 111 en octal = 7 111 en octal = 7
8. Por lo tanto, "A3F" en octal es "5077".



D S

B A B C D E F G H I J K L M N O P Q R S

A U

O B I O Q D D



$$\begin{aligned} 30 &= 17 \\ &= 48 \\ &= 49 \end{aligned}$$

Hexadecimal a decimal

- **Representación de valor posicional:**

1. **Identificar los dígitos hexadecimales individuales:** Descomponer el n hexadecimal en sus dígitos individuales.
2. **Asignar valores posicionales:** Cada dígito en un número hexadecimal representa una potencia de 16. El dígito más a la derecha tiene un valor 16^0 , el segundo dígito tiene un valor de 16^1 , y así sucesivamente.
3. **Multiplicar cada dígito por su valor posicional:** Multiplicar cada dígito hexadecimal por su potencia de 16 correspondiente.
4. **Sumar los productos:** Sumar los productos obtenidos en el paso 3 para el equivalente decimal.

- **Ejemplo:** Convertir el número hexadecimal A3F a decimal.

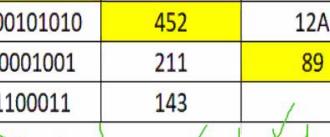
1. Dígitos individuales: A, 3, F
2. Valores posicionales: 16^2 , 16^1 , 16^0
3. Productos: $10 * 16^2 = 2560$, $3 * 16^1 = 48$, $15 * 16^0 = 15$
4. Equivalente decimal: $2560 + 48 + 15 = 2623$

Código: BCD8421



- **BCD8421** (Binary Coded Decimal 8421) es un código binario utilizado para representar números decimales. Se basa en un sistema de ponderación en el que cada dígito decimal se codifica con 4 bits binarios, de acuerdo al siguiente esquema:
 - **Bit 4:** Peso de 8 (MSB)
 - **Bit 3:** Peso de 4
 - **Bit 2:** Peso de 2
 - **Bit 1:** Peso de 1 (LSB)
- El número decimal "5" en BCD utilizando el código 8421 es "0101".
- El código BCD (Binary-Coded Decimal) 8421 asigna un código de 4 bits a cada dígito decimal del 0 al 9, donde cada dígito se representa por su equivalente binario de 4 bits. En este caso, el número decimal "5" se representa como "0101" en BCD utilizando el código 8421.



binario	octal	hexadecimal	decimal
11010101	325	D5	213
100101010	452	12A	298
10001001	211	89	137
1100011	143		63
			99
			
			
Binario	Octal		
000	0		
001	1		
010	2		
011	3		
100	4		
101	5		
110	6		
111	7		

DECIMAL	HEX	
0	0	
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111



Clase 09/09/2024

The screenshot shows a Microsoft Excel spreadsheet titled "Libro1 - Excel". The spreadsheet contains a table with four columns: BINARIO, OCTAL, HEXADECIMAL, and DECIMAL. Row 6 contains the binary value 10101010, which is converted to 252 in octal and 273 in decimal. Row 7 contains the binary value 10111011. The user interface includes a ribbon menu with tabs like Archivo, Inicio, Insertar, etc., and a toolbar with various formatting options. At the top, there is a video conference interface with three participants: Andy Yanacallo, Itecsur.2 Itecsur, and Lenin Albán, each with their respective profile pictures. A green box highlights the video conference area, and a blue circular button with a right-pointing arrow is visible in the top right corner.

	BINARIO	OCTAL	HEXADECIMAL	DECIMAL
6	10101010	252		273
7	10111011			
8				
9				
10				
11				
12				
13				
14				
15				
16				

Tabla de conversiones:

Decimal	Binario	Octal	Hexadecimal
170	10101010	252	aa
187	10111011	273	bb



binario	octal	hexadecimal	decimal
11010101	325	D5	213
100101010	452	12A	298
10001001	211	89	137
1100011	143	63	99

11

Binario	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

DECIMAL	HEX	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

2^7 2^6 2^5 2^4 2^3 2^2 2^1 $2^0 +$

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1

		2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
		256	128	64	32	16	8	4	2	1
S I E S M A Y O R O I G U A L C O L O C O 1		0	1	0	0	1	0	1	1	0
S I E S N O E S M A Y O R C O L O C O U N C E R O										
			22			6		2		0



binario	octal	hexadecimal	decimal	DECIMAL	HEX	BINARY
11010101	325	D5	213	0	0	0000
100101010	452	12A	298	1	1	0001
10001001	211	89	137	2	2	0010
1100011	143		99	3	3	0011
		Binario	Octal	4	4	0100
		000	0	5	5	0101
		001	1	6	6	0110
		010	2	7	7	0111
		011	3	8	8	1000
		100	4	9	9	1001
		101	5	10	A	1010
		110	6	11	B	1011
		111	7	12	C	1100
				13	D	1101
				14	E	1110
				15	F	1111

Clase 11/09/2024

BINARIO	OCTAL	HEXADECIMAL	DECIMAL
101010101	525	155	341
1010101	125	55	85
11101	35	1d	29
100000	40	20	32
101000	50	28	40
10101101	255	ad	173
101101	55	2d	45
111111	77	3f	63
10010110	226	96	150
101011110	536	15e	350

Uso de Github



```
MINGW64:/c/Users/Sigfred/documents/SEGUNDO SEMESTRE TSDS/ELECTRÓNICA Y SISTEMAS... - □ ×  
Sigfred@Esteban_Caiza MINGW64 ~/documents/SEGUNDO SEMESTRE TSDS/ELECTRÓNICA Y SISTEMAS DIGITALES/deberes/Ejemplo (master)  
$ git remote add origin https://github.com/elver159123/Ejercicio-git.git  
  
Sigfred@Esteban_Caiza MINGW64 ~/documents/SEGUNDO SEMESTRE TSDS/ELECTRÓNICA Y SISTEMAS DIGITALES/deberes/Ejemplo (master)  
$ git push -u origin master  
Enumerating objects: 3, done.  
Counting objects: 100% (3/3), done.  
Delta compression using up to 12 threads  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (3/3), 276.10 KiB | 25.10 MiB/s, done.  
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
To https://github.com/elver159123/Ejercicio-git.git  
 * [new branch]      master -> master  
branch 'master' set up to track 'origin/master'.  
  
Sigfred@Esteban_Caiza MINGW64 ~/documents/SEGUNDO SEMESTRE TSDS/ELECTRÓNICA Y SISTEMAS DIGITALES/deberes/Ejemplo (master)  
$ git status  
On branch master  
Your branch is up to date with 'origin/master'.  
nothing to commit, working tree clean  
  
Sigfred@Esteban_Caiza MINGW64 ~/documents/SEGUNDO SEMESTRE TSDS/ELECTRÓNICA Y SISTEMAS DIGITALES/deberes/Ejemplo (master)  
$ |
```

Clase 16/09/2024

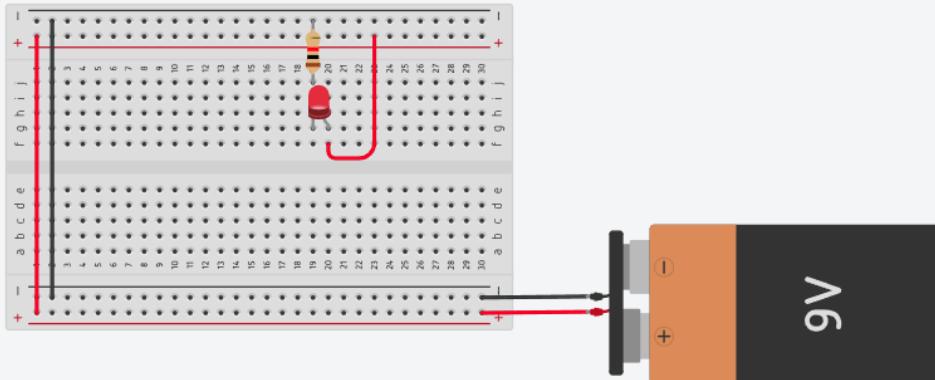
The screenshot shows a Google Slides presentation titled "Electrónica-Arduino - Presentación". The slide content is as follows:

Encender un led

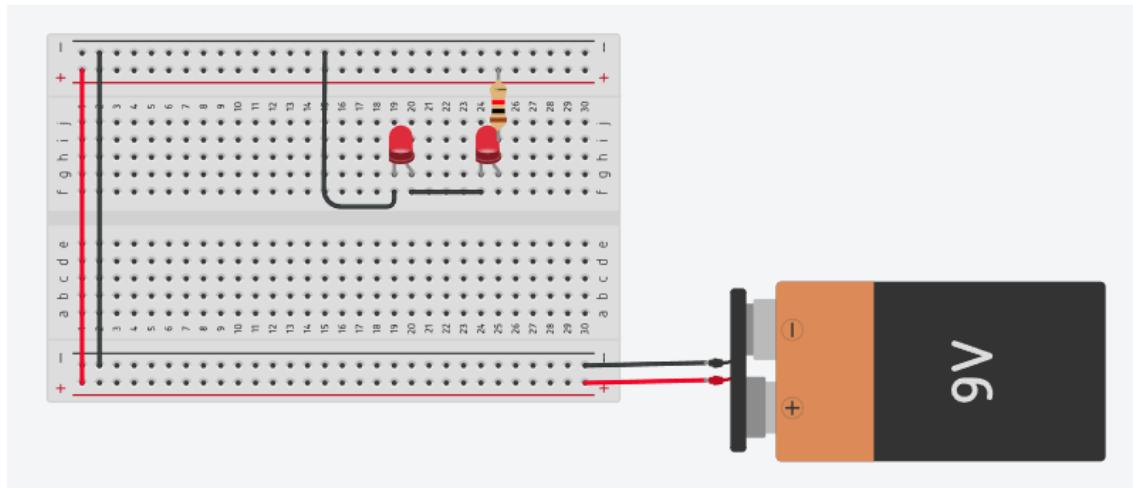
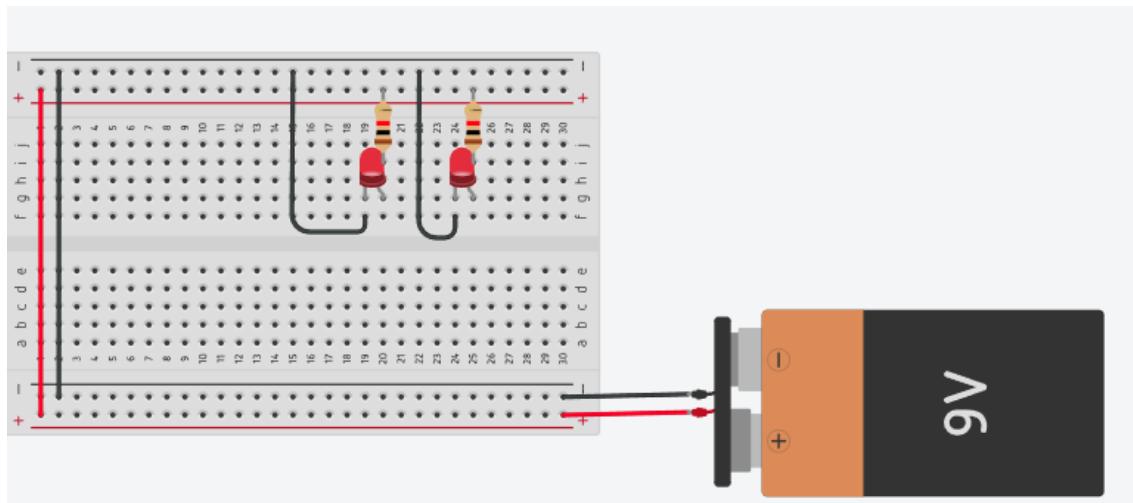
Diagram illustrating the circuit to connect an LED to a 9V battery. The LED is connected in series with a resistor and a switch, with the positive terminal of the LED connected to the positive terminal of the battery.

Below the diagram, there is a note: "Haz clic para añadir notas del orador".

The sidebar on the left contains steps 3, 4, and 5, which are part of a larger sequence of slides. Step 3 shows a circuit diagram for connecting an LED. Step 4 shows a note: "Conectar dos resistencias en serie". Step 5 shows a screenshot of a smartphone displaying a circuit diagram for connecting an LED.

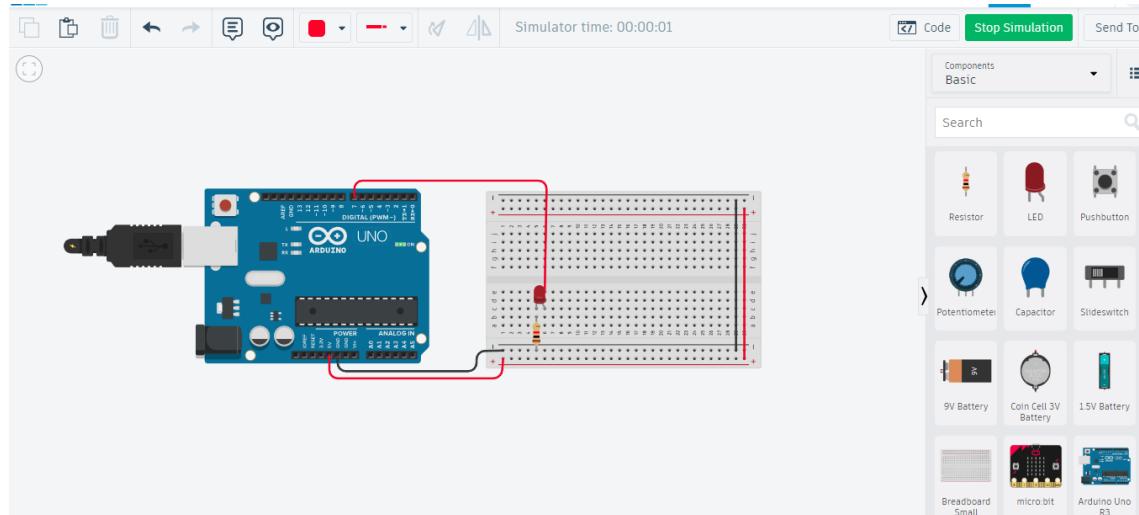


Conexión en serie

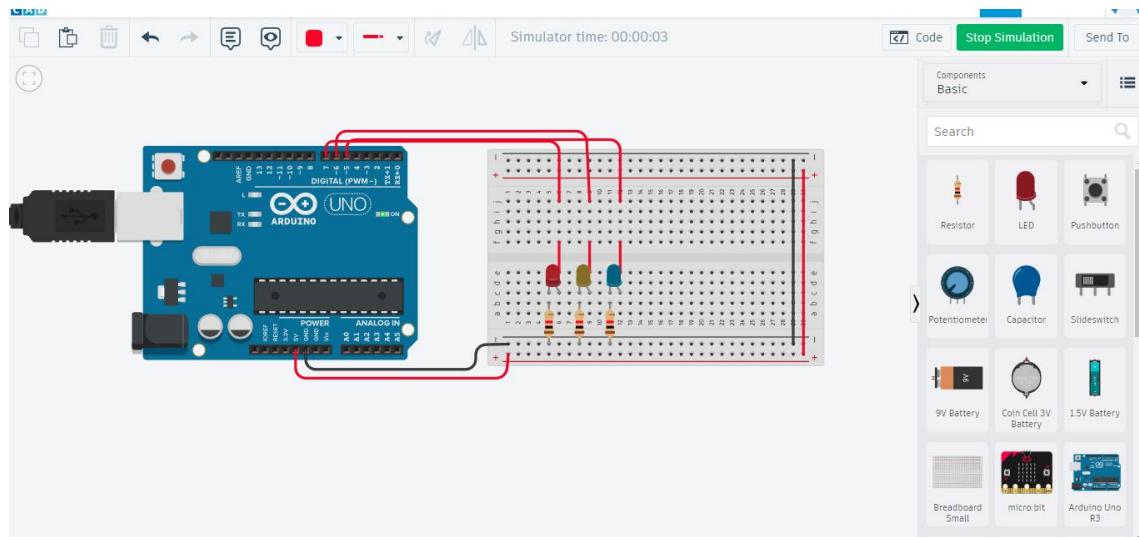




Encender el led



Ejercicio encender 3 leds





```
//  
void setup()  
{  
    for(int i=5;i<=7;i++){  
        pinMode(i, OUTPUT);  
    }  
  
}  
  
void loop()  
{  
    for(int i=5;i<=7;i++){  
        digitalWrite(i, HIGH);  
        delay(1000); // Wait for 1  
        digitalWrite(i, LOW);  
        delay(1000); // Wait fo  
    }  
}
```

```
1 void setup()  
2 {  
3     for (int i=5;i<=7;i++){  
4         pinMode (i, OUTPUT);  
5     }  
6     }  
7 }  
8  
9 void loop()  
10 {  
11     for(int i=5;i<=7;i++){  
12         digitalWrite(i, HIGH);  
13         delay(500);  
14         digitalWrite(i, LOW);  
15         delay(500);  
16     }  
17  
18     for(int i=7;i>=5;i--){  
19         digitalWrite(i, HIGH);  
20         delay(500);  
21         digitalWrite(i, LOW);  
22         delay(500);  
23     }  
24 }
```

Tarea en clase apagar y encender 7 luces led con el ciclo while



```
int led1 = 2;
int led2 = 3;
int led3 = 4;
int led4 = 5;
int led5 = 6;
int led6 = 7;
int led7 = 8;

void setup()
{
    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);
    pinMode(led3, OUTPUT);
    pinMode(led4, OUTPUT);
    pinMode(led5, OUTPUT);
    pinMode(led6, OUTPUT);
    pinMode(led7, OUTPUT);
}

void loop()
{
    int i = 0;
    while (i < 7)
    {
        digitalWrite(led1 + i, HIGH);
        delay(500);
        digitalWrite(led1 + i, LOW);
        i++;
    }
}
```

Actuación en clase



Actividad: Encender una alarma si la distancia es menor de 120cm, si es mayor encender un foco

```
int led1 = 5;
int buzzer=4;
void setup()
{
    Serial.begin(9600);
    pinMode(5, OUTPUT);
    pinMode(4, OUTPUT);
}
int distancia, tiempo;
void loop()
{
    pinMode(8, OUTPUT);
    digitalWrite(8, LOW);
    delayMicroseconds(2);
    digitalWrite(8, HIGH);
    delayMicroseconds(10);
    pinMode(8, INPUT);
    tiempo = pulseIn(8, HIGH);
    distancia = tiempo / 29 / 2;
    Serial.println(String(distancia) + " cm");
    if (distancia >= 120)
    {
        digitalWrite(5, HIGH);
        digitalWrite(4, LOW);
    }
    else
    {
        digitalWrite(5, LOW);
    }
}
```

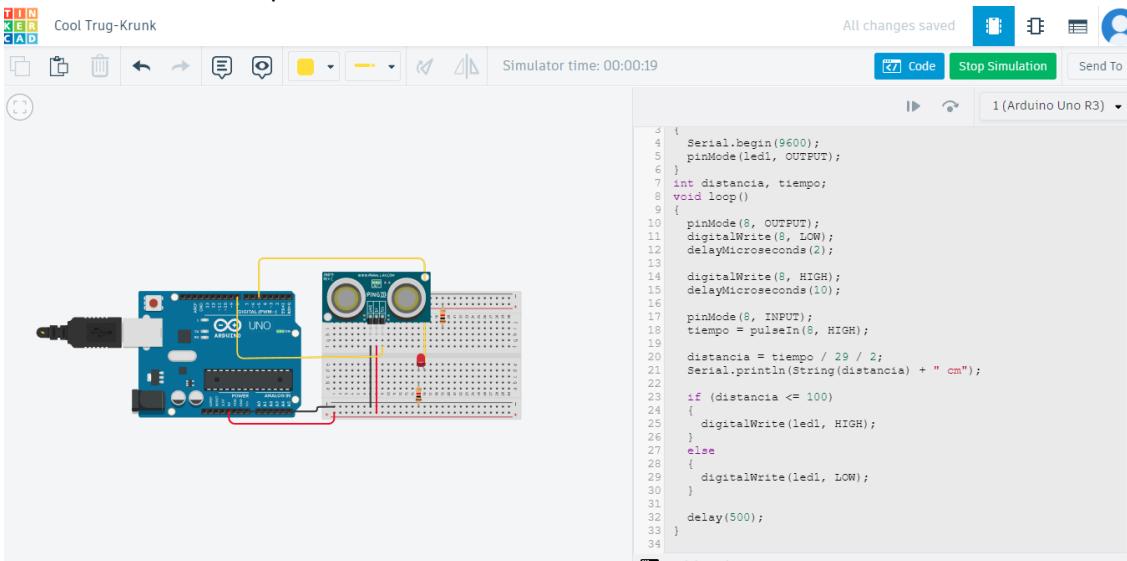
Sensor de distancia PING



- El **sensor Ping** es un dispositivo ultrasónico que se utiliza para medir distancias mediante la emisión y recepción de ondas de sonido.
- Funciona enviando pulsos de sonido a una frecuencia muy alta, y luego mide el tiempo que tarda el eco de ese sonido en regresar después de haber rebotado en un objeto.
- Configuramos el pin digital como salida porque el sensor ultrasónico requiere que se le envié un pulso de salida para iniciar la medición de distancia
- Al colocar el pin en **LOW** y esperar 2 microsegundos, nos aseguramos de que el pin comience en un estado estable y sin actividad eléctrica. Es una buena práctica inicializar el pin en **LOW** para garantizar que no se generen pulsos no deseados en el sensor antes de comenzar el proceso de medición.



Tarea en clase hacer que el led se encienda cuando este a menos de 100 metros



```
1 // Cool Trug-Krunk
2
3 // This sketch demonstrates how to use the PING module with an Arduino Uno.
4 // It sends a pulse from pin 9, receives an echo on pin 10, and prints the distance
5 // to the Serial Monitor.
6
7 // The PING module uses a 40KHz signal to measure distance. It sends a short
8 // burst of pulses and then waits for the echo. The time it takes for the echo to
9 // return is proportional to the distance.
10
11 // The PING module has three pins: Trig (pin 9), Echo (pin 10), and Vcc (pin 5).
12 // Pin 9 is an output pin that generates a short pulse when triggered.
13 // Pin 10 is an input pin that receives the echo signal from the PING module.
14 // Pin 5 is a power supply pin that must be connected to 5V.
15
16 // To use the PING module, you need to connect it to the Arduino Uno as follows:
17 // Trig (pin 9) to Digital Pin 9
18 // Echo (pin 10) to Digital Pin 10
19 // Vcc (pin 5) to 5V
20
21 // The code below initializes the serial connection, sets up the pins, and then
22 // enters a loop where it sends a pulse and receives an echo.
23
24 // The distance is calculated using the formula: distance = (time / 2) * speed_of_sound
25 // where speed_of_sound is approximately 343 meters per second at room temperature.
26
27 // Note: The PING module has a built-in 2-microsecond delay between sending
28 // a pulse and receiving an echo. This is accounted for in the code.
29
30 // You can change the threshold distance in the if statement to trigger the LED
31 // at different distances.
32
33 // The code also includes a delay of 500ms between measurements.
34
35 // To use this sketch, upload it to your Arduino Uno and connect the PING module
36 // to the pins as specified above. Then open the Serial Monitor to see the
37 // distance measurements.
38
39 // Author: Cool Trug-Krunk
40 // License: CC-BY-SA
41
42 // This sketch is based on the PING module example provided by Arduino.
43 // The original code is available at: https://www.arduino.cc/en/Tutorial/Ping
44
45 // Modified by Cool Trug-Krunk to demonstrate distance measurement with
46 // an Arduino Uno and a PING module.
```

Pantalla lcd



```
#include <LiquidCrystal.h>
LiquidCrystal lcd (7, 6, 5, 4, 3, 2);
void setup()
{
  Serial.begin(9600);
  lcd.begin(16,2);
}
int distancia,tiempo;
void loop()
{
  // establecer el cursor primero es columna y despues la fila
  lcd.setCursor(3,0);
  lcd.print("Hola mundo");
  lcd.setCursor(5,1);
  lcd.print("Steban");
}
```

Actividad 3 Arduino

Utilizar el sensor Ping y
Mostrar la distancia en
cm en el lcd

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
long sensorPing()
{
  long tiempo, distancia;
  pinMode(8, OUTPUT);
  digitalWrite(8, LOW);
  delayMicroseconds(2);
  digitalWrite(8, HIGH);
  delayMicroseconds(10);
  pinMode(8, INPUT);
  tiempo = pulseIn(8, HIGH);
  distancia = tiempo / 58;
  return distancia;
}
void setup()
{
  lcd.begin(16, 2);
}
void loop()
```



The screenshot displays a software environment for electronic circuit simulation and programming. On the left, a breadboard circuit is shown connected to an Arduino Uno microcontroller. The circuit includes a 16x2 LCD display, a numeric keypad, a red LED, a buzzer, and a power source. On the right, the Arduino IDE shows the following code:

```
69  digitalWrite(13, HIGH); // apaga el buzzer
70  lcd.clear();
71  lcd.print("Benvenido");
72  delay(1000);
73  lcd.clear();
74
75  }
76
77  else
78  {
79      intentos--;
80      lcd.clear();
81      lcd.print("Clave incorrecta");
82      lcd.setCursor(0, 1);
83      delay(1000);
84      lcd.clear();
85      lcd.print("Acceso denegado");
86      digitalWrite(12, HIGH); // encender led rojo
87      digitalWrite(13, HIGH); //encender buzzer
88      delay(500);
89      tone(13,220,100);
90      digitalWrite(12, LOW); // apagar led rojo
91      digitalWrite(13, LOW); //apagar buzzer
92      lcd.clear();
93      if (intentos > 0)
94      {
95          lcd.print("Te quedan ");
96          lcd.print(intentos);
97          lcd.print(" intentos");
98          delay(2000);
99          lcd.clear();
100     }
101 }
```