

**CLASE 3**  
Unidad 1



# ARQUITECTURA Y SISTEMAS OPERATIVOS

**Profesor:** Pablo Andres Gonzales Camargo

# CLASIFICACION DE CIRCUITOS LOGICOS

## CIRCUITOS LOGICOS

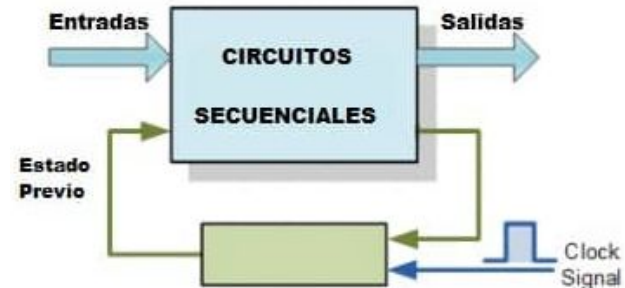
### COMBINACIONALES

El valor de las salidas, dependen únicamente del valor de las entradas



### SECUENCIALES

El valor de las salidas depende del valor de las entradas y de valores "pasados" de las salidas, por lo que se dice que tienen "memoria".



# FUNCIONES LÓGICAS

Con la combinación de varias operaciones AND, OR, NOT se pueden construir distintos circuitos lógicos que realizan distintas operaciones muy útiles y que se emplean en la construcción de sistemas digitales completos.

- Multiplexor MUX
- Demultiplexor DEMUX
- Codificador CODEC
- Sumador/Restador
- Registros REGs
- Contadores

COMBINACIONALES

SECUENCIALES

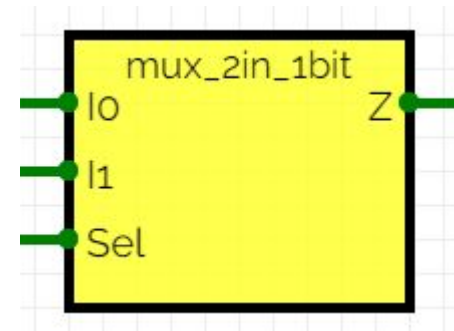


# MULTIPLEXOR

Un Multiplexor “MUX” pasa la información procedente de varias líneas de entrada  $I_0 \dots I_n$  a una única línea de salida  $Z$  según una entrada especial de Selección  $Sel$ .

$$Z = Sel' I_0 + Sel I_1$$

- a. Construir la tabla de verdad
- b. Construir el circuito equivalente



# MULTIPLEXOR

Circuito Lógico

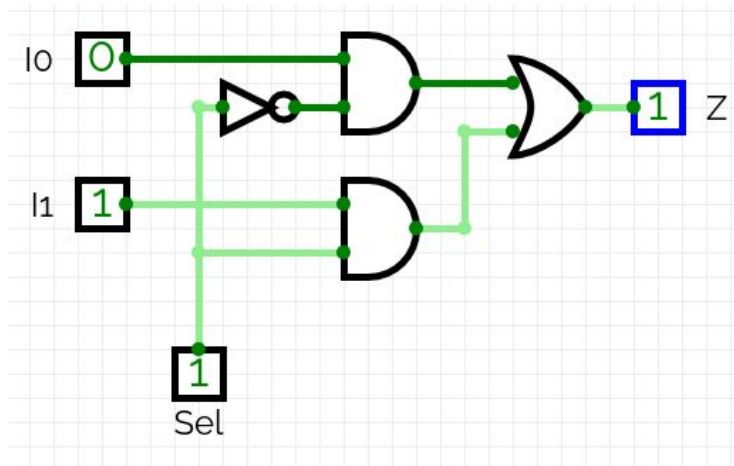
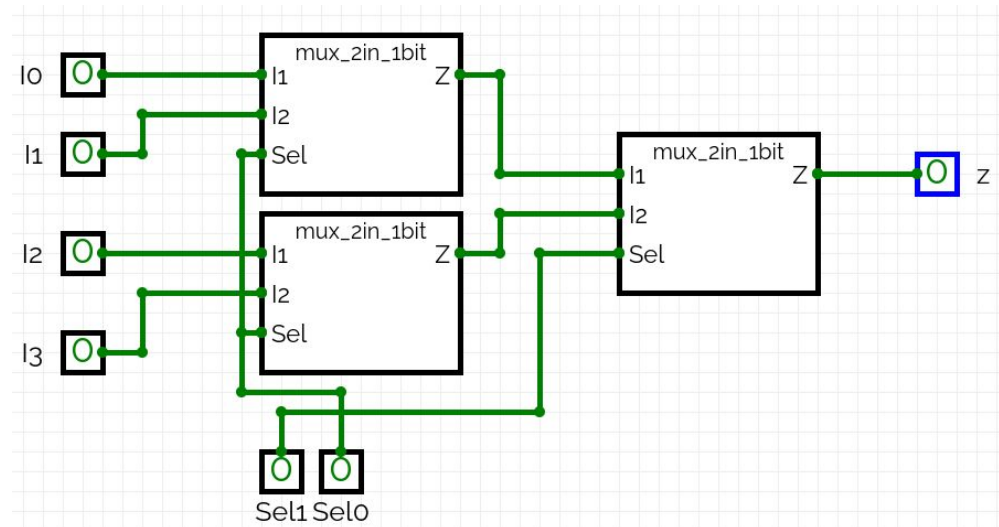
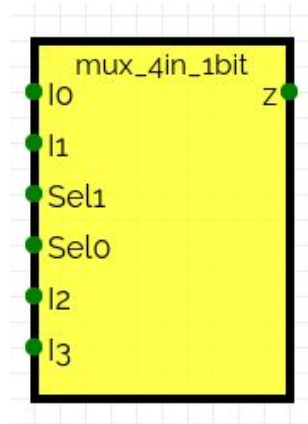


Tabla de verdad

Entradas			Salid
Sel	$I_0$	$I_1$	$Z$
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

# MULTIPLEXOR (varias entradas)

- ¿Cómo se construye por ejemplo un **MULTIPLEXOR (MUX) de 4 entradas de 1 bit?** u otro de más entradas como 8,16, 32, etc
- Las tablas de verdad tendrían miles de filas por analizar!! ( $2^4=16$  filas;  $2^8=256$ )



# SUMADOR

Este circuito realiza (simula) la SUMA ARITMÉTICA (de cantidades) entre 2 números de entrada **A** y **B** junto con un acarreo de entrada  $C_{in}$  en BINARIO y devuelve un resultado **S** y un acarreo de salida  $C_{out}$

## FUNCIONES NO REDUCIDAS

$$S = A'B'C_{in} + A'BC_{in}' + AB'C_{in}' + ABC_{in}$$

$$C_{out} = A'BC_{in} + AB'C_{in} + ABC_{in}' + ABC_{in}$$

## FUNCIONES REDUCIDAS

$$S = (A \oplus B) \oplus C_{in}$$

$$C_{out} = (A \oplus B)C_{in} + AB$$



- Construir la tabla de verdad
- Construir el circuito equivalente

# SUMADOR

Circuito Lógico

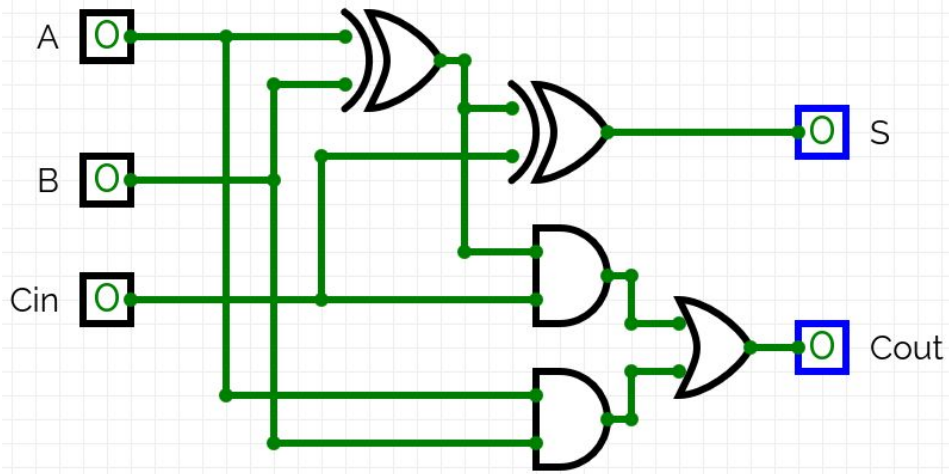


Tabla de verdad

Entradas			Salidas	
A	B	Cin	Cout	S
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



# COMPARADOR DE 1 BIT

- Es un circuito que compara 2 entradas de 1 bit cada una.
- Verifica si son iguales, si uno es mayor que el otro o si es menor.
- Para cada uno de estos casos tiene 3 salidas que se activan sólo una a la vez.

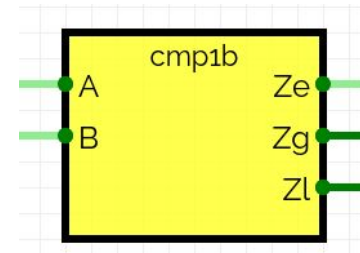
$$Z_e = AB + A'B'$$

$$Z_g = AB'$$

$$Z_l = A'B$$

\*(**e**=equals, **g**=A greater than B, **l**= A lesser than B)

- Construir la tabla de verdad
- Construir el circuito equivalente



# COMPARADOR DE 1 BIT

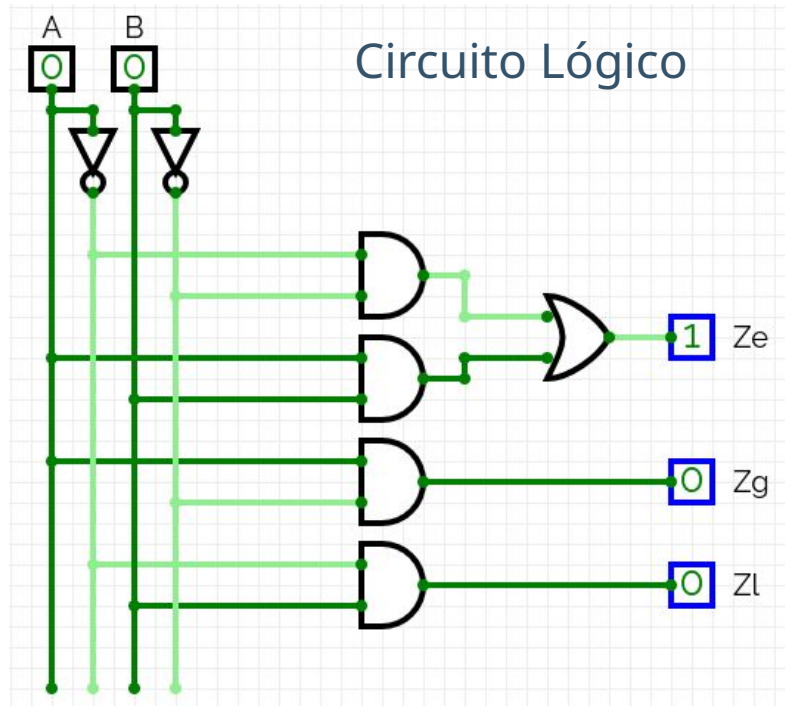


Tabla de verdad

Entradas		Salidas		
A	B	Ze	Zg	Zl
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

# UNA ALU MUY SIMPLE DE 1 BIT

- Combinando los circuitos combinacionales vistos, se construyó una pequeña ALU de 1 bit que realiza las siguientes operaciones: AND, OR, XOR y SUMA/RESTA.
- Tiene un total de 6 entradas y 2 salidas (todas de 1 bit)
- La operación se elige con los bits Oper que forman un selector.
- En la imagen se seleccionó hacer una SUMA (+/- debe ser '0')

