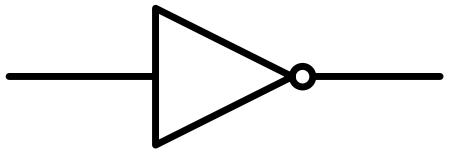
Práctica 2: Lógica Digital - Combinatorios

Organización del Computador I
DC - UBA

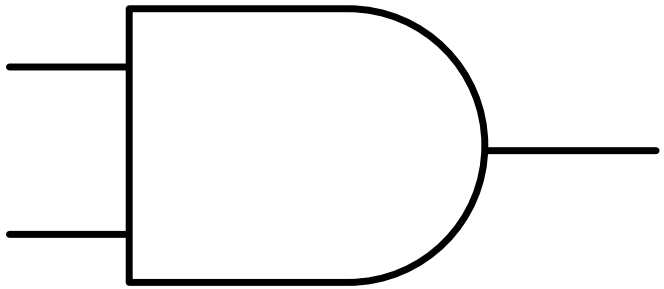
Segundo Cuatrimestre 2020

Compuertas - NOT



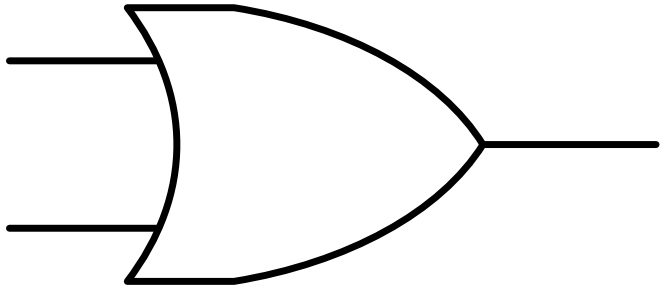
A	NOT A
0	1
1	0

Compuertas - AND



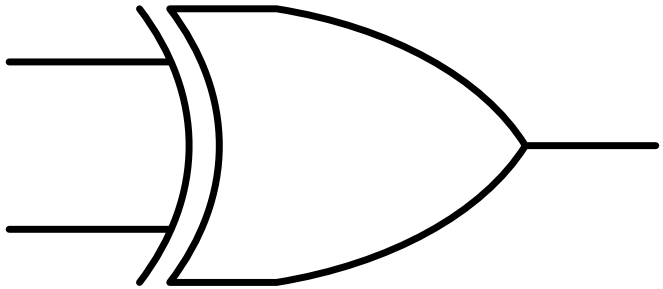
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Compuertas - OR



A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Compuertas - XOR u OR-EXCLUSIVA



A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Notación

$$A + B \equiv A \ OR \ B$$

$$AB \equiv A.B \equiv A \ AND \ B$$

$$\overline{A} \equiv NOT \ A$$

Propiedades

Propiedad	AND	OR
Identidad	$1.A = A$	$0 + A = A$
Nulo	$0.A = 0$	$1 + A = 1$
Idempotencia	$A.A = A$	$A + A = A$
Inverso	$A.\overline{A} = 0$	$A + \overline{A} = 1$
Conmutatividad	$A.B = B.A$	$A + B = B + A$
Asociatividad	$(A.B).C = A.(B.C)$	$(A + B) + C = A + (B + C)$
Distributividad	$A + (B.C) = (A + B).(A + C)$	$A.(B + C) = A.B + A.C$
Absorción	$A.(A + B) = A$	$A + A.B = A$
De Morgan	$\overline{A.B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}.\overline{B}$

Tarea: ¡Demostrarlas!

Ejercicio I

Armar un inversor de 3 bits. Este circuito invierte o no las tres entradas de acuerdo al valor de una de ellas que actúa como control. En otras palabras, un inversor de k-bits es un circuito de k entradas $(e_k, ..., e_0)$ y $k-1$ salidas $(s_k - 1, ..., s_0)$ que funciona del siguiente modo:

- Si $e_k = 1$, entonces $s_i = not(e_i)$ para todo $i < k$
- Si $e_k = 0$, entonces $s_i = e_i$ para todo $i < k$

Ejemplo:
inversor(1,011)=100
inversor(0,011)=011
inversor(1,100)=011
inversor(1,101)=010

Ejercicio I

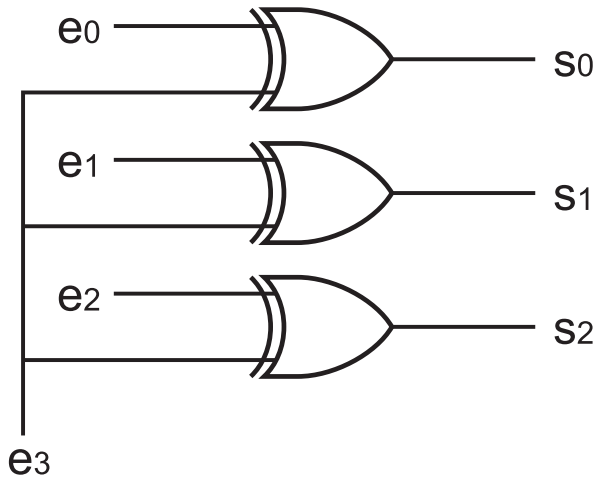
Primero pensar como invertir un bit,

ei	ek	si
0	0	0
0	1	1
1	0	1
1	1	0

Como suma de productos,
$(\overline{ei} \cdot ek) + (ei \cdot \overline{ek})$

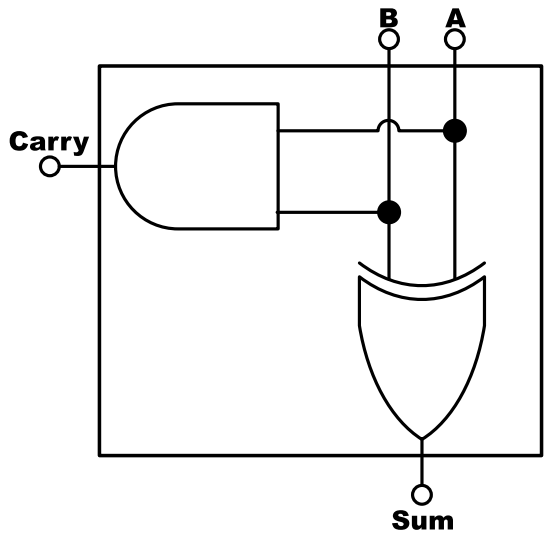
¡Oh! casualidad, es una XOR ($\oplus$)
$(\overline{A} \cdot B) + (A \cdot \overline{B}) = A \oplus B$

Solucion con XOR:



Ejercicio II - Sumador Simple

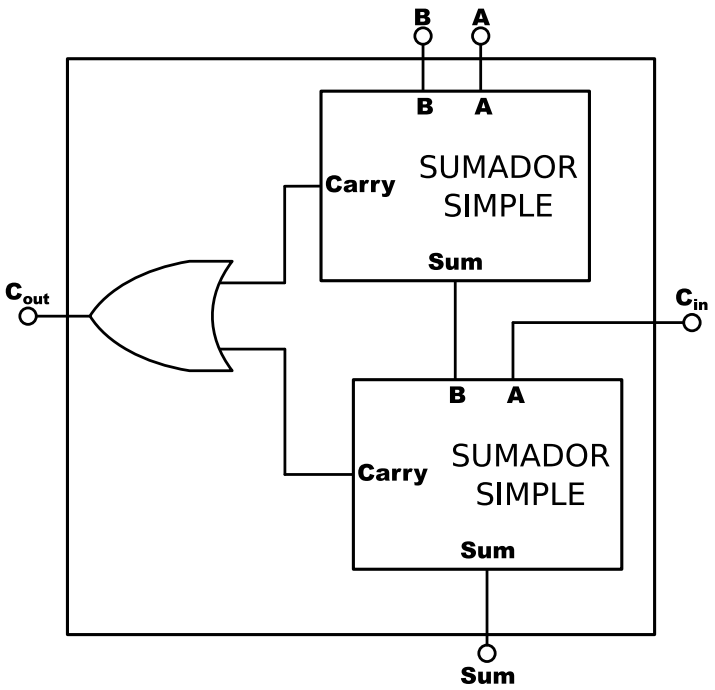
Armar un **sumador de 1 bit**. Tiene que tener dos entradas de un bit y dos salidas, una para el resultado y otra para indicar si hubo o no acarreo.
Solución:



Ejercicio III - Sumador Completo

Teniendo dos sumadores simples (de 1 bit) y sólo una compuerta a elección, arme un **sumador completo**. Tiene 2 entradas de 1 bit y una tercer entrada interpretada como C_{In}, tiene como salida C_{Out} y S.

Solución:



Ejercicio IV - Sumador Completo de 3 bits

Armar un sumador completo de 3 bits.
Solución:

¡Tarea!

Ejercicio IV - Shift

Armar un circuito de 3 *bits*. Este deberá mover a izquierda o a derecha los bits de entrada de acuerdo al valor de una de ellas que actúa como control. En otras palabras, un shift izq-der de k-bits es un circuito de $k+1$ entradas $(e_k, ..., e_0)$ y k salidas $(s_{k-1}, ..., s_0)$ que funciona del siguiente modo:

- Si $e_k = 1$, entonces $s_i = e_{i-1}$ para todo $0 < i < k$ y $s_0 = 0$
- Si $e_k = 0$, entonces $s_i = e_{i+1}$ para todo $0 \leq i < k-1$ y $s_{k-1} = 0$

Ejemplos:
shift_lr(1,011)= 110
shift_lr(0,011)= 001
shift_lr(1,100)= 000
shift_lr(1,101)= 010

Ejercicio IV - Shift

Armar un circuito de 3 *bits*. Este deberá mover a izquierda o a derecha los bits de entrada de acuerdo al valor de una de ellas que actúa como control. En otras palabras, un shift izq-der de k-bits es un circuito de $k+1$ entradas $(e_k, ..., e_0)$ y k salidas $(s_{k-1}, ..., s_0)$ que funciona del siguiente modo:

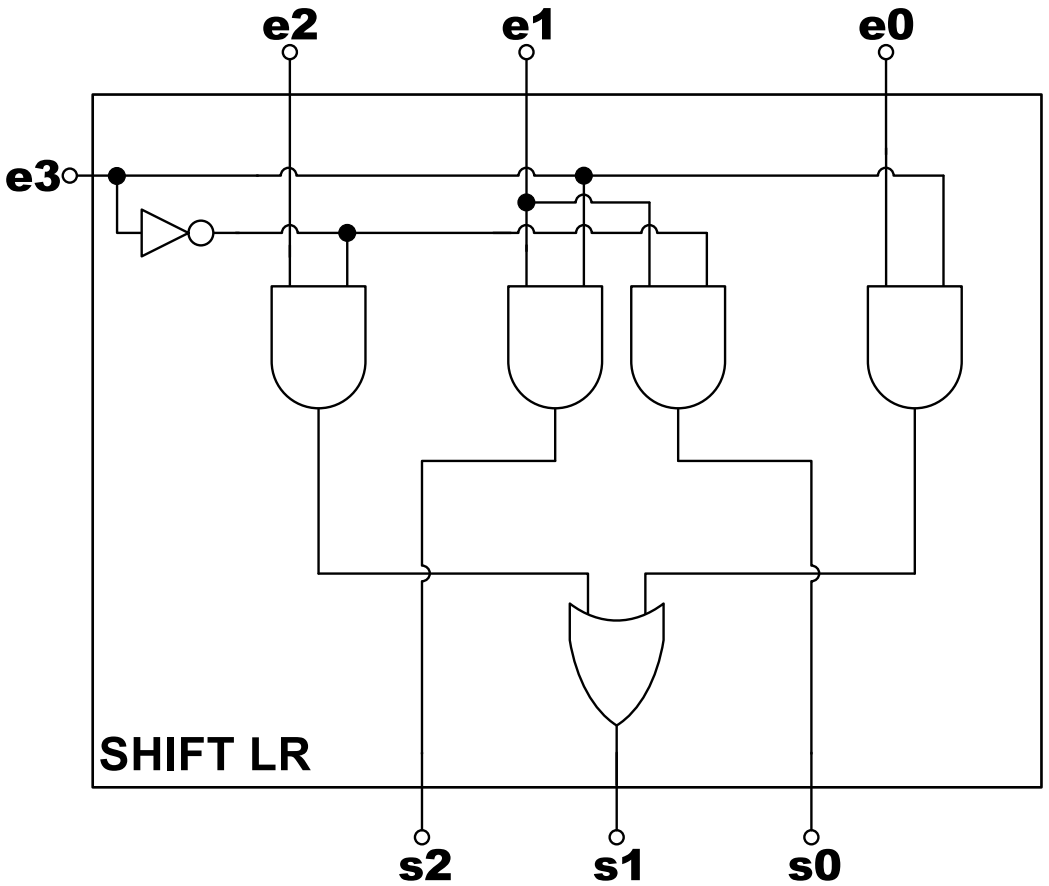
- Si $e_k = 1$, entonces $s_i = e_{i-1}$ para todo $0 < i < k$ y $s_0 = 0$
- Si $e_k = 0$, entonces $s_i = e_{i+1}$ para todo $0 \leq i < k-1$ y $s_{k-1} = 0$

$$s_2 = \begin{cases} 0 & \text{si } e_3 = 0 \\ e_1 & \text{si } e_3 = 1 \end{cases} \qquad s_0 = \begin{cases} 0 & \text{si } e_3 = 1 \\ e_1 & \text{si } e_3 = 0 \end{cases} \qquad s_1 = \begin{cases} e_0 & \text{si } e_3 = 1 \\ e_2 & \text{si } e_3 = 0 \end{cases}$$

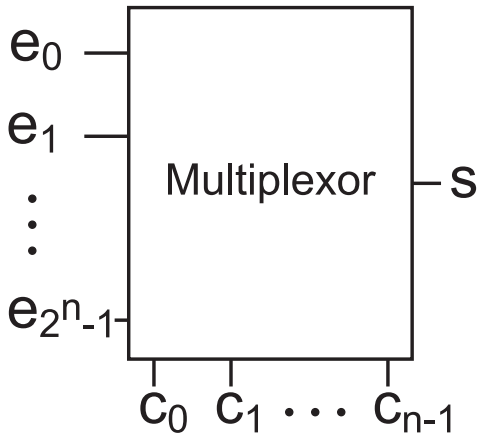
$$e_3.e_1 \qquad \overline{e_3}.e_1 \qquad e_3.e_0 + \overline{e_3}.e_2$$

Ejercicio IV - Solución

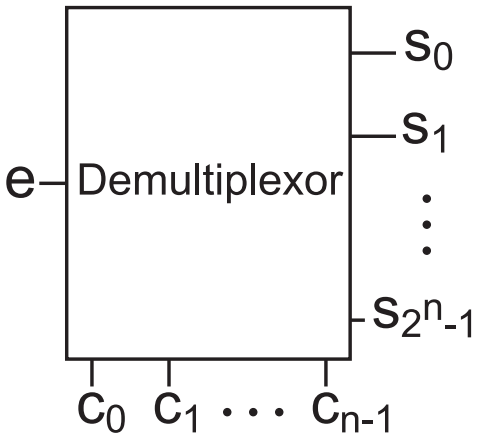
Solución:



Multiplexor y Demultiplexor



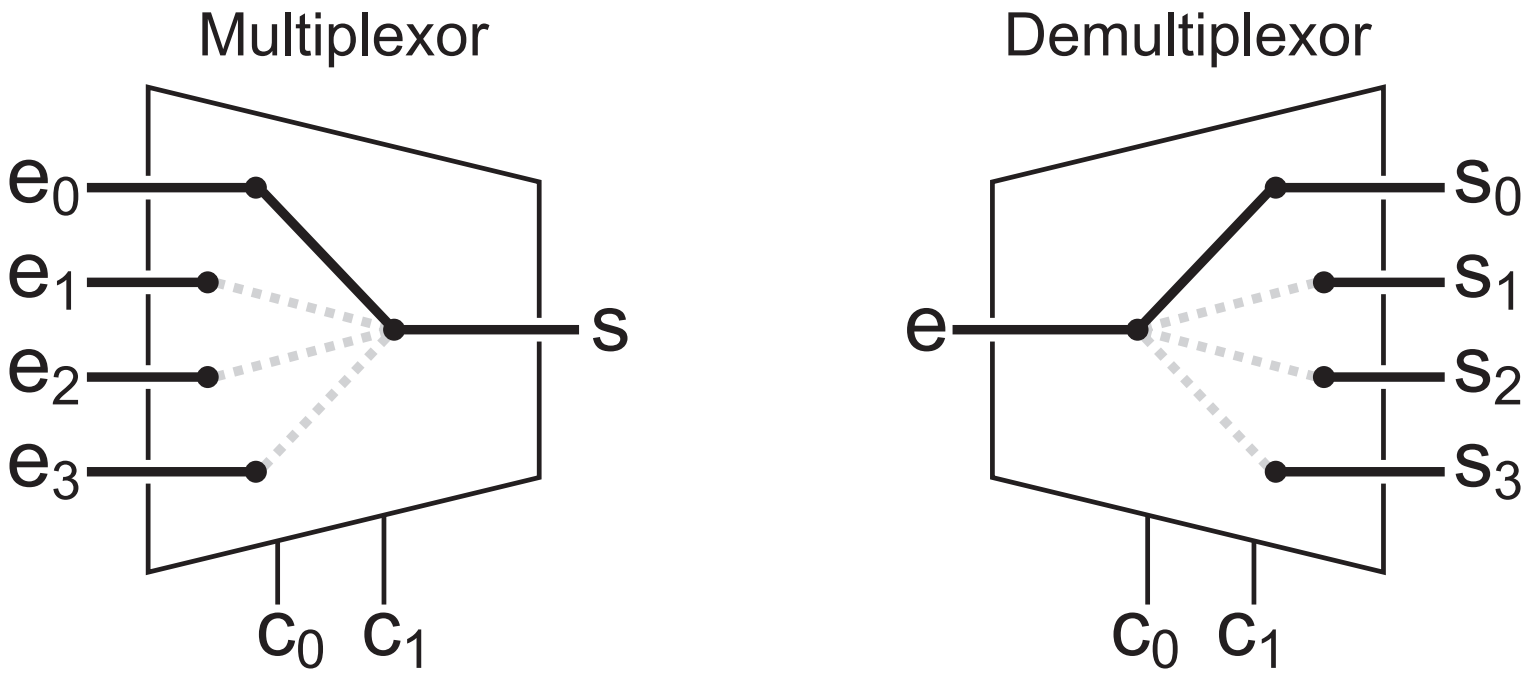
Las líneas de control c permiten seleccionar una de las entradas e, la que corresponderá a la salida s.



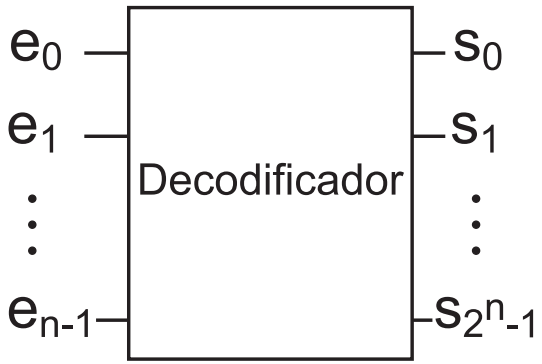
Las líneas de control c permiten seleccionar cual de las salidas s tendrá el valor de e.

Multiplexor y Demultiplexor

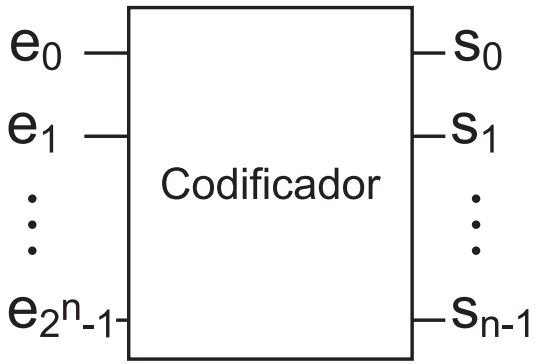
- Ejemplo,



Codificador y Decodificador



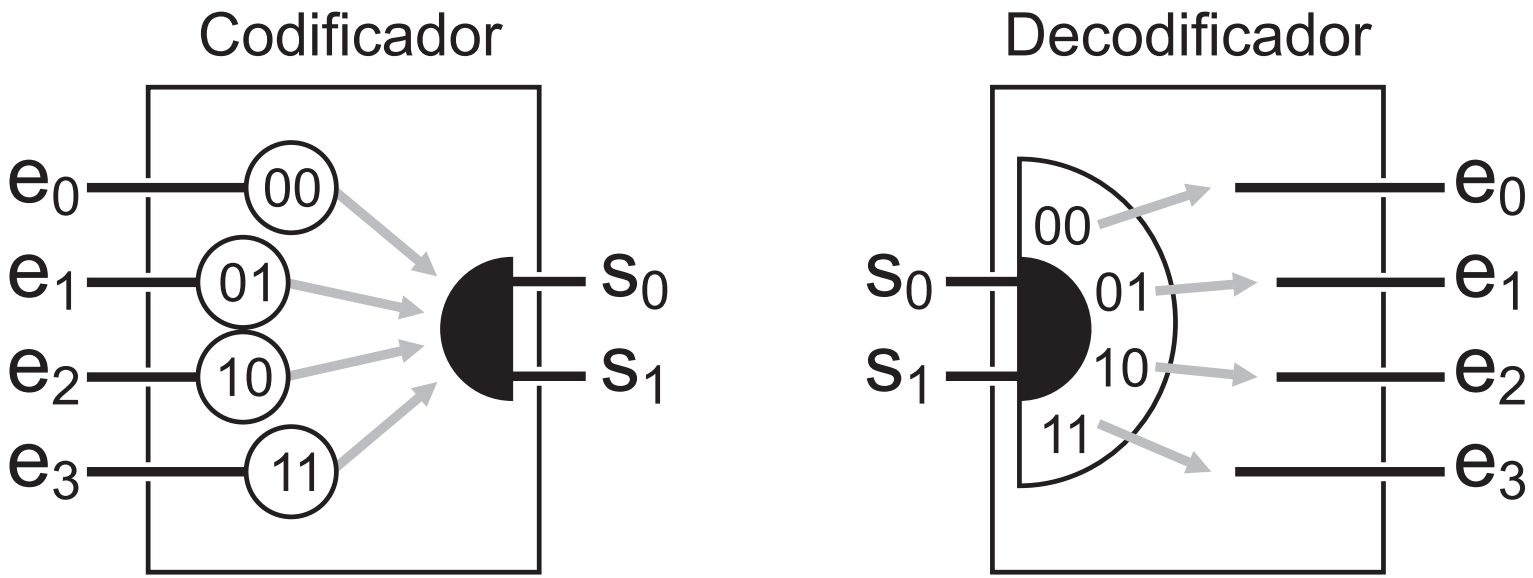
Cada combinación de las líneas *e* corresponderá a una sola línea en alto de la salida *s*.



Una y sólo una línea en alto de *e* corresponderá a una combinación en la salida *s*.

Codificador y Decodificador

- Ejemplo,



La práctica...

- Con lo visto hoy pueden realizar la parte A de la práctica 2.
- Pueden usar el Logisim para probar sus circuitos.
- El **MARTES 15 de Septiembre** tenemos el primer taller de la materia. **Obligatorio**
  - Tienen que tener el Logisim instalado.
  - Van a tener que compartirnos pantalla.
  - ¡Los queremos escuchar! Prueben los micrófonos de sus compus/celulares.
- Bibliografía recomendada: The Essentials of Computer Organization and Architecture - Linda Null - Capítulo 3

¡Eso es todo amigos!

¿Preguntas?