Organización del Computador 1

Lógica digital
Algebra de Boole y circuitos combinatorios

Lógica digital

- La computadora está construida con circuitos que operan con 2 valores [0,1] que podemos interpretar como [Falso,Verdadero]
- Idea: implementar operaciones lógicas y matemáticas combinando circuitos

Álgebra de Boole

George Boole
1815-1864

- George Boole desarrolló un sistema algebraico para formalizar la lógica proposicional ("Análisis matemático de la lógica")
- El sistema consiste en un cálculo para resolver problemas de lógica proposicional con sólo 2 valores posibles
  - Da igual: 1 y 0, Verdadero y Falso, On y Off, Boca y River.
- 3 Operaciones:
  - AND
  - OR
  - NOT

Funciones booleanas: AND

- Función booleana AND (aka producto booleano)

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

- Notación: X.Y

Funciones booleanas: OR

- Función booleana OR (aka suma booleana)

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

- Notación: X+Y

Funciones booleanas: NOT

- Función booleana NOT (aka complemento booleano)

X	NOT
0	1
0	1
1	0
1	0

- Notación: ~X

Tablas de Verdad

- Una función f es una función matemática cuyo dominio e imagen está incluido en Bool {0,1}
- Se puede describir completamente unívocamente usando *tablas de verdad*
- Por ejemplo, F: Bool x Bool x Bool → Bool

X	Y	Z	F(X,Y,Z)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Expresiones booleanas

- Utilizan:
  - Variables booleanas (X,Y,Z, etc)
  - Las operaciones
    - AND (X.Y)
    - OR (X+Y)
    - NOT (~X)

Expresiones Booleanas

- ¿Qué expresión booleana representa a F?
- ¿Es única?

X	Y	Z	F(X,Y,Z)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Expresiones booleanas

- Las expresiones booleanas
  - (~X.Y.~Z)+(~X.Y.Z)+(X.~Y.Z)+(X.Y.~Z)+(X.Y.Z)
  - (X.~Z)+Y
- … son expresiones booleanas *distintas* que representan a la misma función booleana f(x,y,z)

Funciones booleanas: NAND y NOR

X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0

X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0

Propiedades del Algebra de Boole

Identidad	1.A=A	0+A=A
Nula	0.A=0	1+A=1
Idempotencia	A.A=A	A+A=A
Inversa	A.˜A=0	A+˜A=1
Conmutativa	A.B=B.A	A+B=B+A
Asociativa	(A.B)C=A.(B.C)	(A+B)+C=A+(B+C)
Distributiva	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	˜(A.B) = ˜A+˜B	˜(A+B) = ˜A.˜B

Ejemplo

- Usando las propiedades del algebra de Boole podemos reducir la expresión
  - $F(X,Y,Z)=(X+Y).(X+\sim Y).\sim(X.\sim Z)$

$(X+Y).(X+\sim Y).(\sim X+Z)$	DeMorgan
$(X.X + X.\sim Y+Y.X+Y.\sim Y).(\sim X+Z)$	Distributiva
$(X + X.\sim Y+Y.X + 0). (\sim X+Z)$	Indempotencia e Inversa
$(X + X.(\sim Y+Y)). (\sim X+Z)$	Nula y Distributiva
$(X).(\sim X+Z)$	Inversa, Identidad y Nula
$X.\sim X+X.Z$	Distributiva
XZ	Inversa e Identidad

Funciones universales

- Un conjunto de funciones booleanas se denomina “universal” si alcanza para definir ***cualquier*** función booleana usando solamente esas funciones
- <u>Preguntas:</u>
  - ¿Es {NOT,AND,OR} un conjunto universal?
  - ¿Es {NAND} un conjunto universal?
  - ¿Es {NOT} un conjunto universal?

Expresiones equivalentes

- Varias expresiones booleanas pueden representar la misma función booleana
  - Son lógicamente equivalentes
- En general se suelen elegir formas normals
  - Suma de productos
    - $F(X,Y,Z) = X.Y + X.Z + Y.Z$
  - Producto de sumas
    - $F(X,Y,Z) = (X+Y) . (X+Z) . (Y+Z)$

Suma de Productos

- Se obtiene fácilmente a partir de la tabla de verdad
- Elegimos filas que cuyo resultado es 1 y hacemos “producto” (AND) de las entradas
- Luego “sumamos” (OR) todas las expresiones

$F(x,y,z) = x\bar{z}+y$

x	y	z	$x\bar{z}+y$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

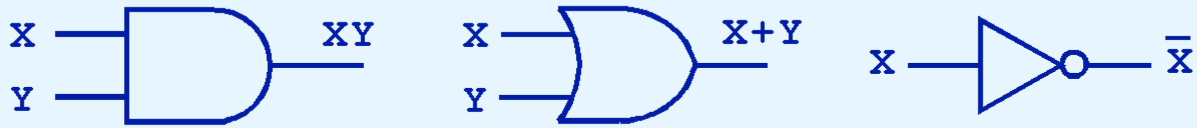
F(X,Y,Z) =(~X.Y.~Z)+(~X.Y.Z)+(X.~Y.~Z)+(X.Y.~Z)+(X.Y.Z)

Compuertas lógicas

- Una **compuerta** es un dispositivo electrónico que produce resultado en base a un conjunto de valores de entrada
- **Idea**: una compuerta implementa funciones del Álgebra de Boole “recurrentes”.
- Son el “átomo” de la computación

Compuertas lógicas

- Implementantan las funciones booleanas que vimos



X AND Y

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X OR Y

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

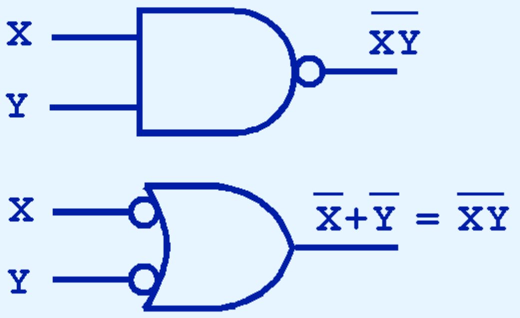
NOT X

X	$\overline{X}$
0	1
1	0

Compuertas lógicas

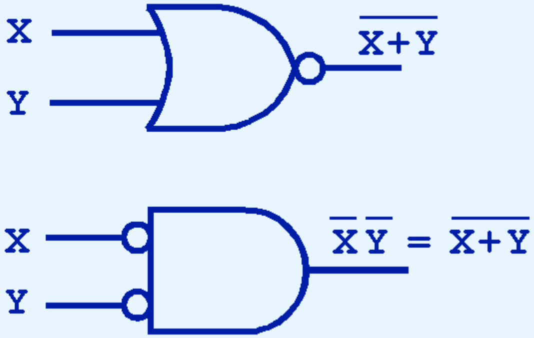
X NAND Y

X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0



X NOR Y

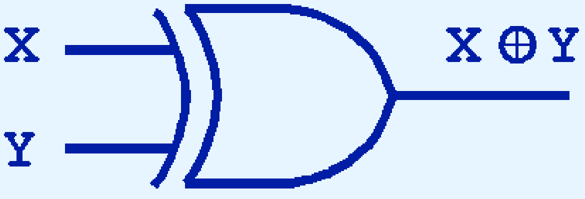
X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0



Compuertas Lógicas

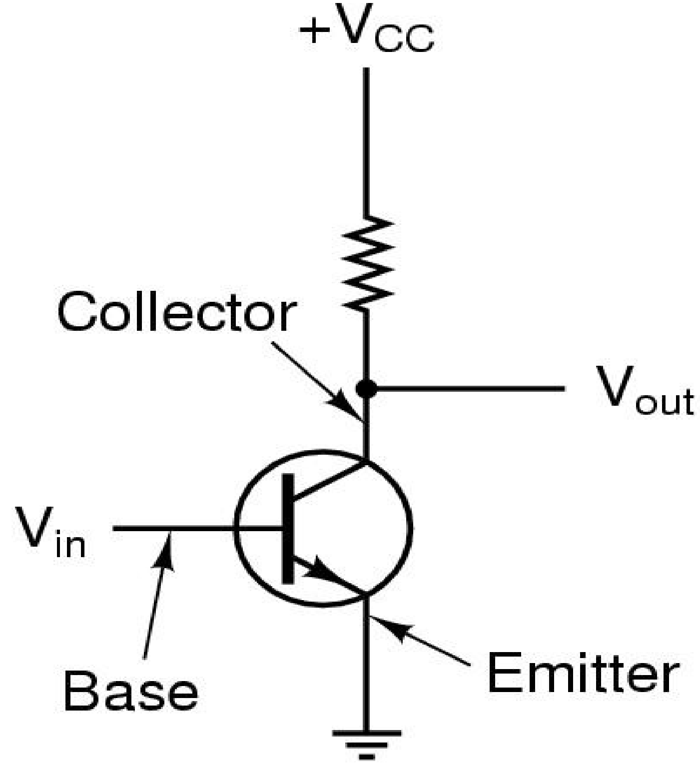
X XOR Y

X	Y	X ⊕ Y
0	0	0
0	1	1
1	0	1
1	1	0

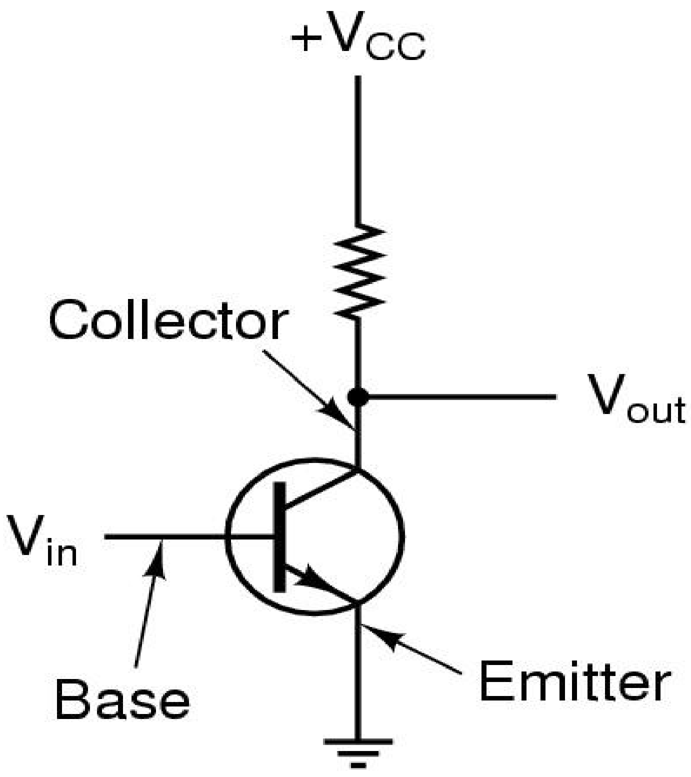


Transistores

- Las compuertas se construyen usando **transistores**
- ¿A qué compuerta equivale este circuito?



Transistores



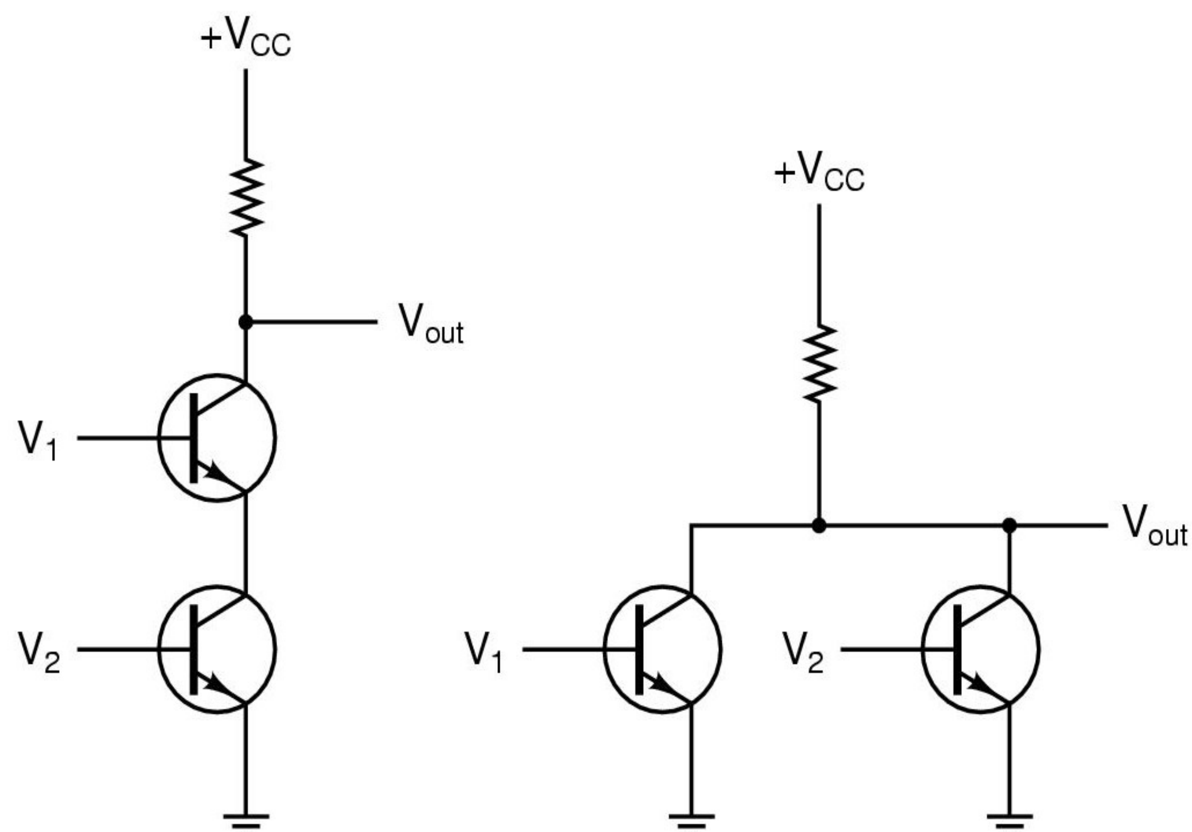
X	NOT
0	1
0	1
1	0
1	0

Compuertas lógicas

- ¿Cómo implemento el NAND? ¿y el NOR?

Compuertas lógicas

- ¿Cómo implemento el NAND? ¿y el NOR?

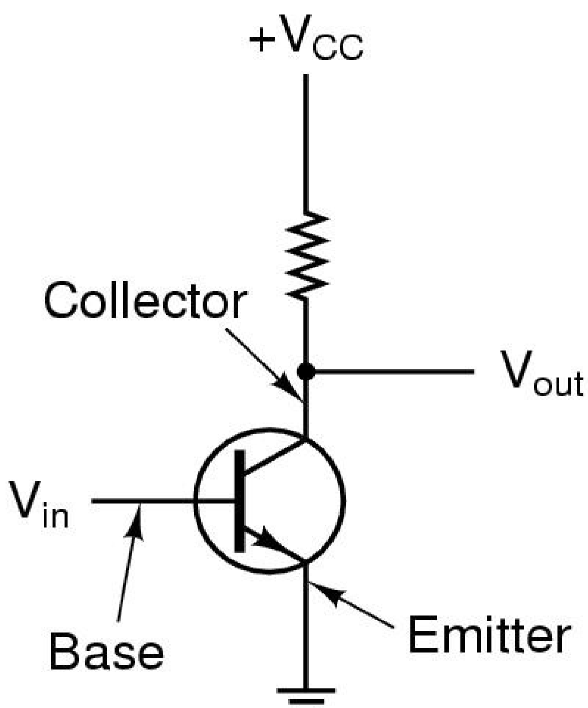


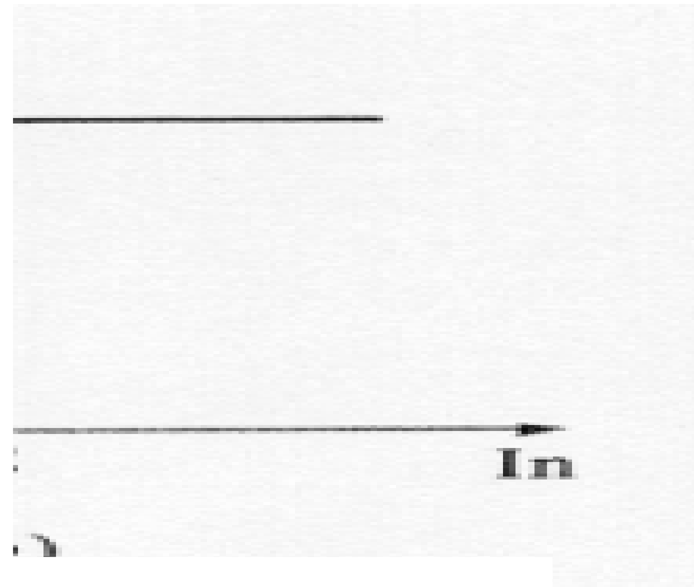
Circuitos Lógicos

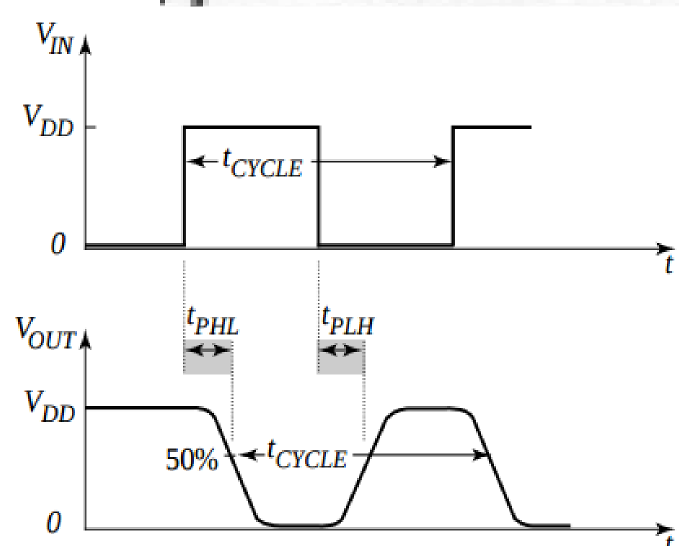
- Un **circuito** es una combinación de compuertas que implementan una función booleana.
- ¿Qué expresión booleana representa este circuito?

- ¿El valor de salida del circuito se propaga instantáneamente?

Circuito ideal / Mundo real

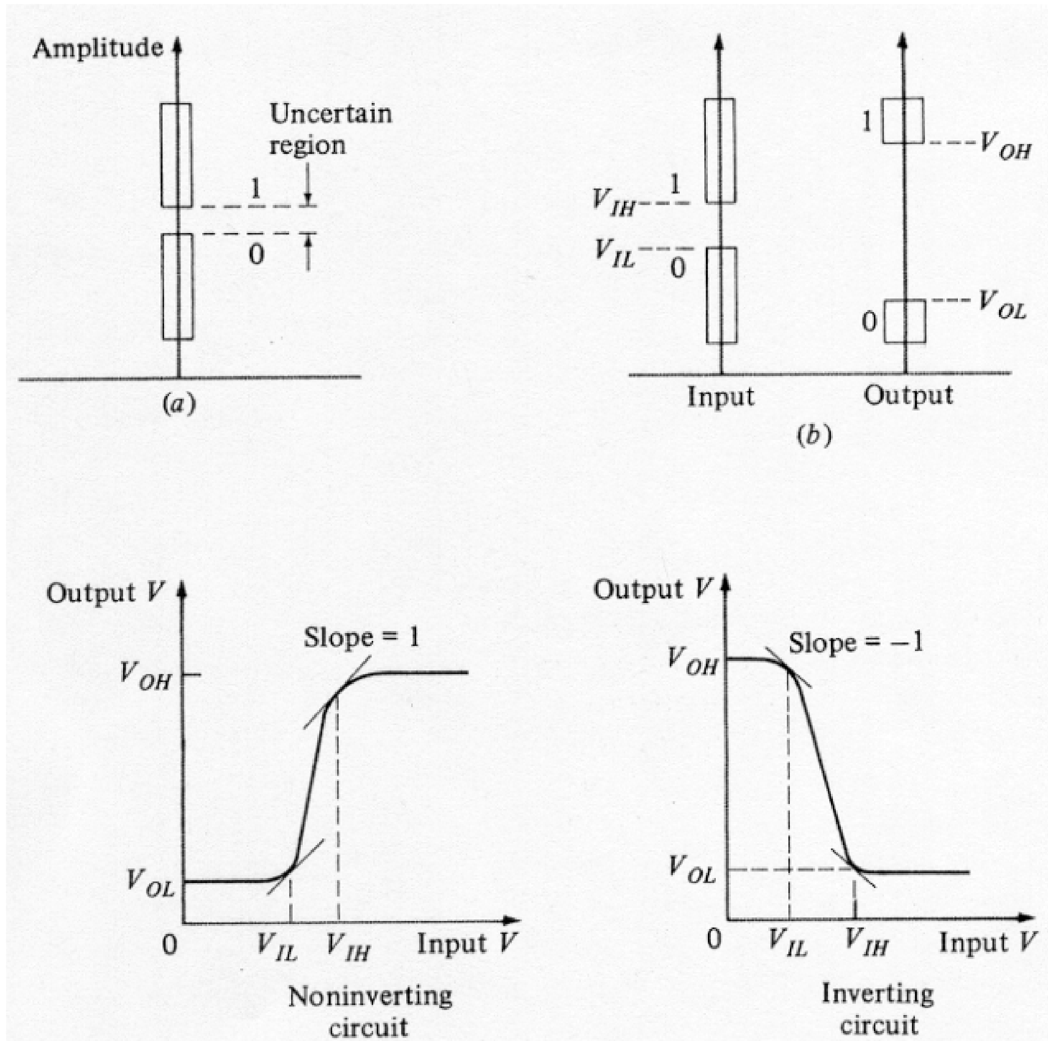






$$\mathbf{t_p} = \frac{1}{2}\left(\mathbf{t_{PHL}} + \mathbf{t_{PLH}}\right)$$

Circuitos digitales: resumiendo



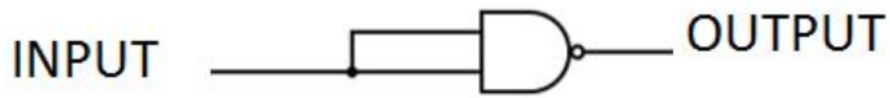
Ejercicio

- Implementar una compuerta NOT usando solamente compuertas NAND

- Ejercicio: Idem para AND y OR

Ejercicio

- Implementar una compuerta NOT usando solamente compuertas NAND



- Ejercicio: Idem para AND y OR

Recap.

- Funciones booleanas
- Expresiones booleanas
- Circuitos combinatorios

- La aritmética (cuentas) y las operaciones lógicas del CPU se implementan usando Circuitos Combinatorios

Sumadores

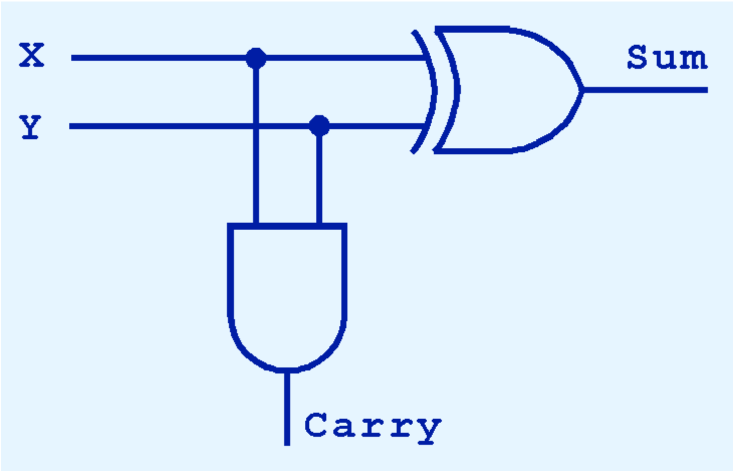
- ¿Cómo podemos construir un circuito que sume 2 bits?

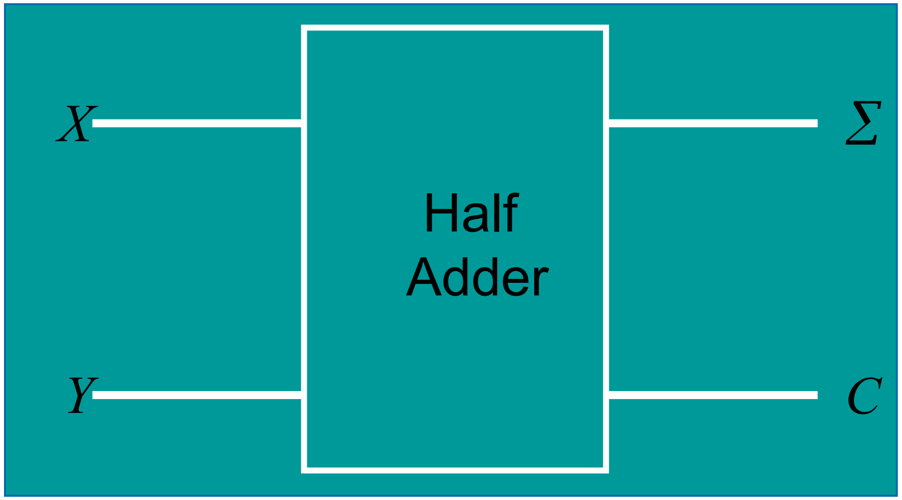
Inputs		Outputs	
X	Y	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Half-Adder

- Al circuito que suma 2 bits se lo llama “Half-Adder”

Inputs		Outputs	
X	Y	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



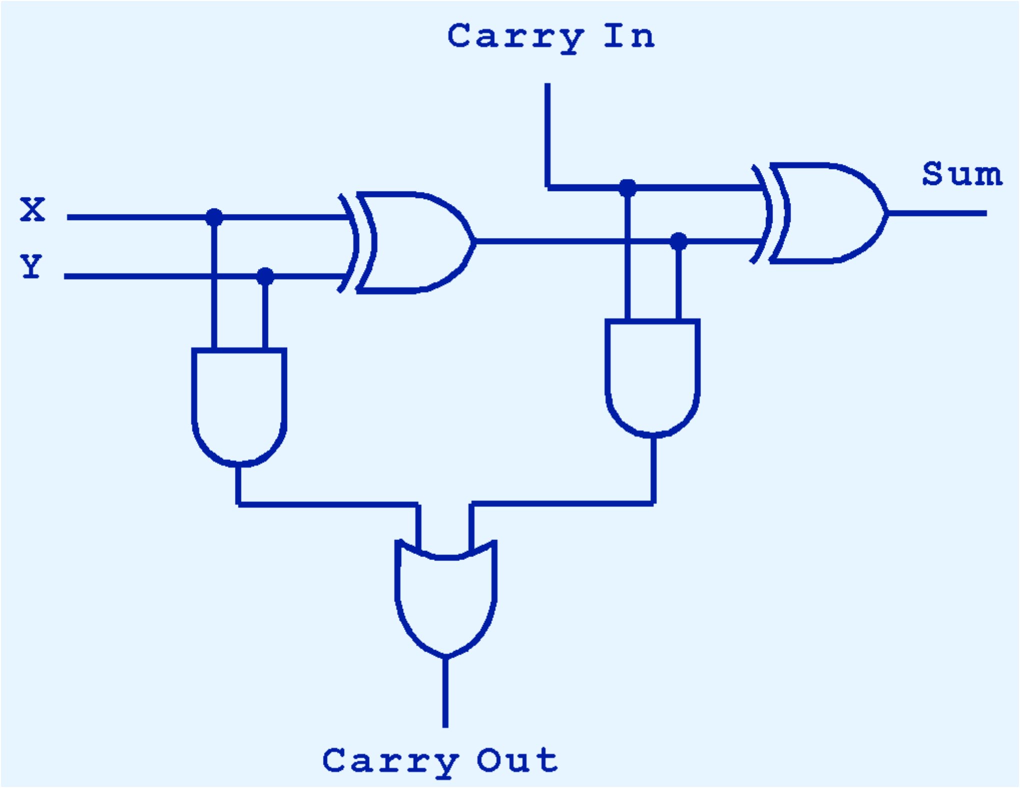


Full-Adder

- ¿Cómo podemos hacer el circuito que sume 3 bits?

Inputs			Outputs	
X	Y	Carry In	Sum	Carry Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

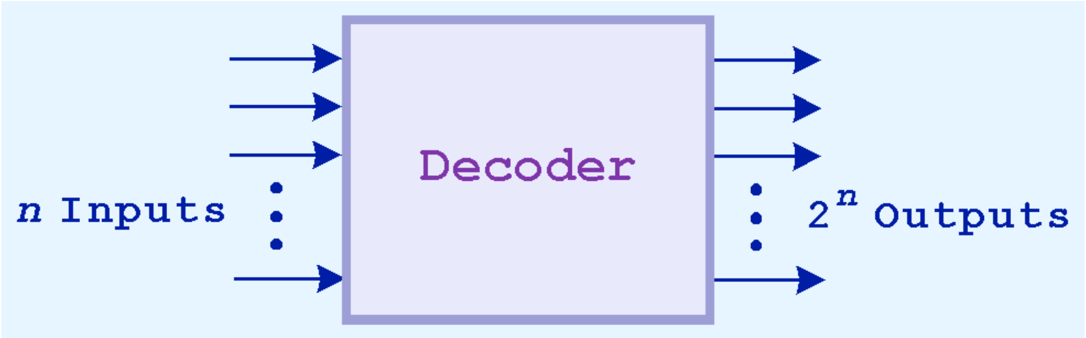
Full Adders



¿Cómo puedo hacer un sumador de 4 (cuatro bits) usando full adders y half adders?

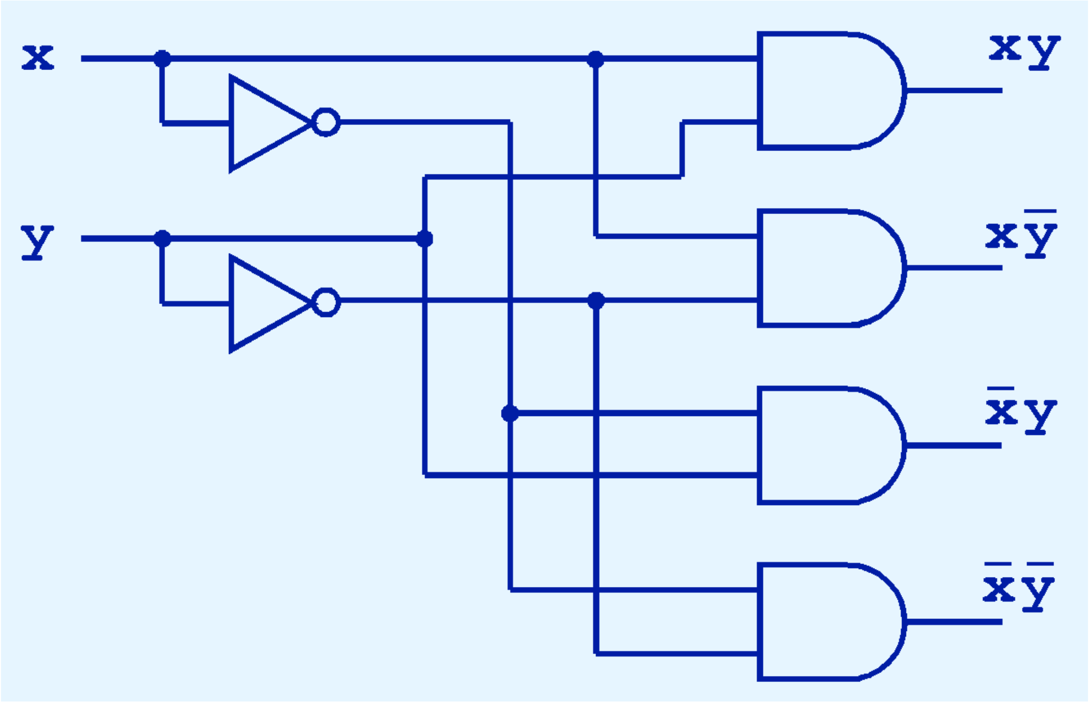
Decodificadores

- Un decodificador de n entradas puede seleccionar una de 2^n salidas
- Son muy importantes
  - Por ejemplo: seleccionar banco de memoria a partir de dirección



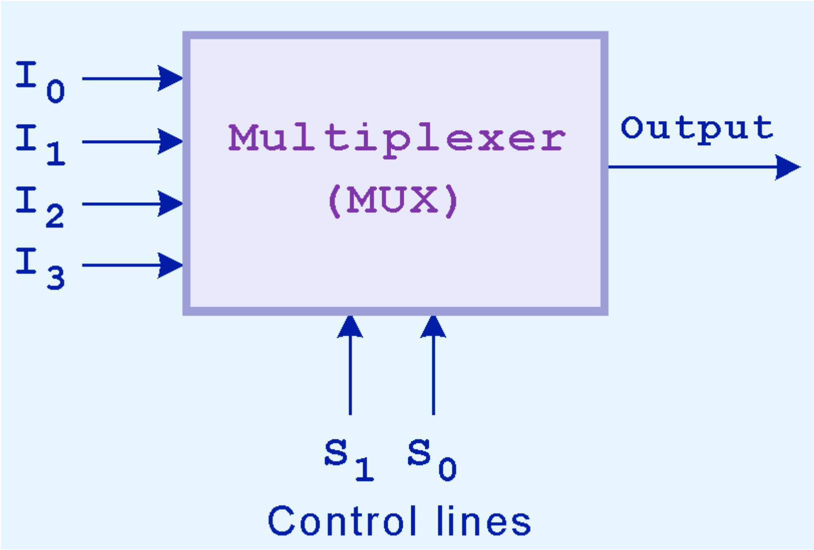
Decodificador de 2 bits

- Si x=0 e y=1 ¿qué salida queda habilitada?



Multiplexores

- Se “forwardea” una línea de entrada a la línea de salida de acuerdo al valor de las líneas de control



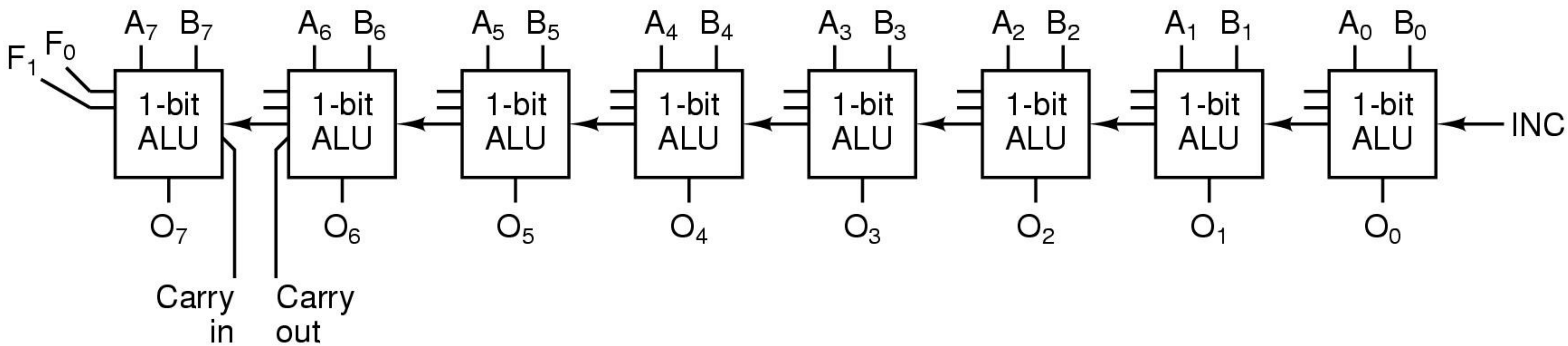
- Demultiplexor: es lo inverso al multiplexor

Ejercicio

- Construir ALU (Unidad Aritmético Lógica) de 1 bit
- 3 entradas: A, B y carry_in
- 2 salida: resultado y carry_out
- 2 líneas de control:

c_0	c_1	operación
0	0	A.B
0	1	A+B
1	0	~B
1	1	sumar(A,B,carry_in)

ALU 8 bits

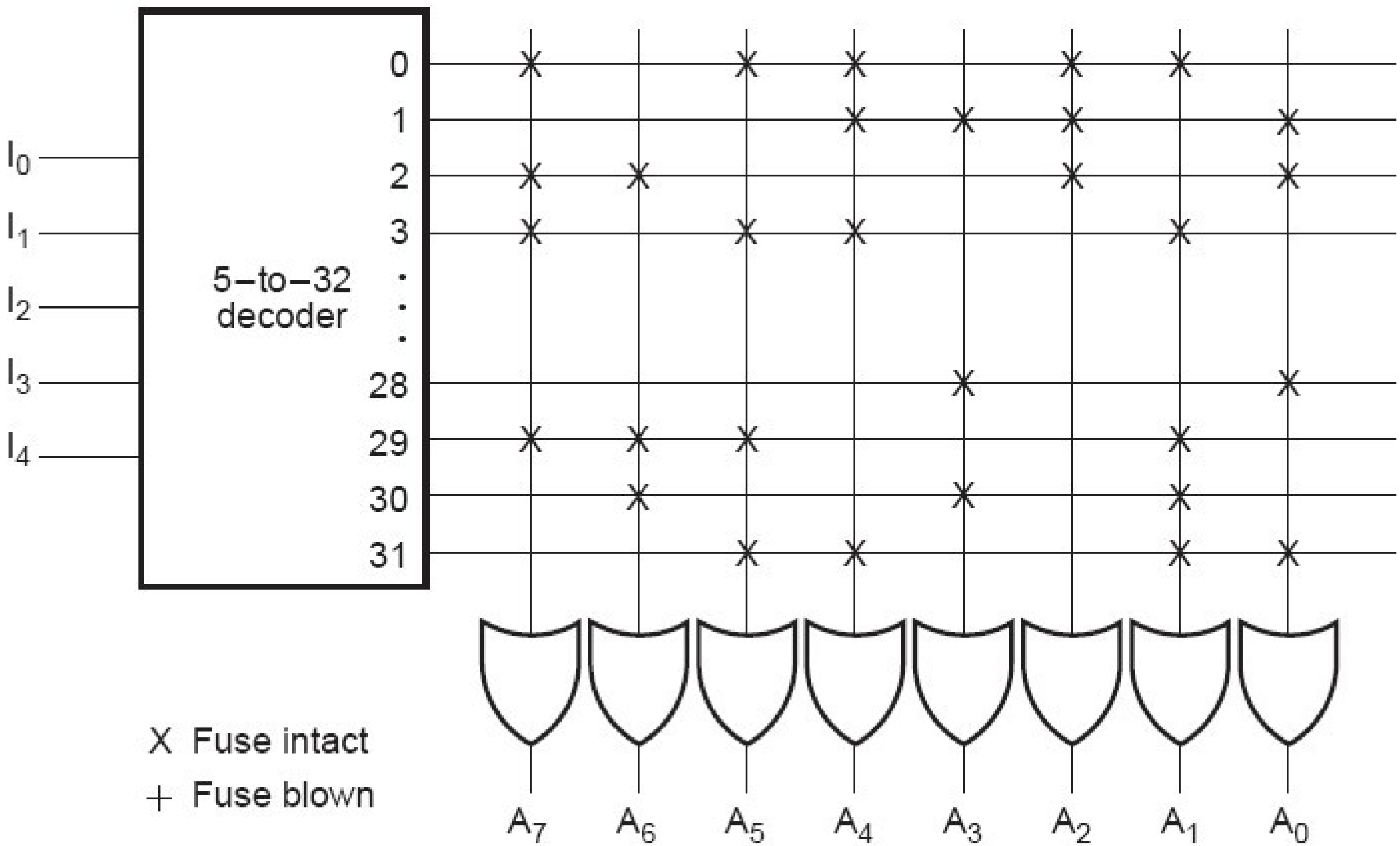


F1
F0
A7 B7
A6 B6
A5 B5
A4 B4
A3 B3
A2 B2
A1 B1
A0 B0
1-bit ALU
1-bit ALU
1-bit ALU
1-bit ALU
1-bit ALU
1-bit ALU
1-bit ALU
1-bit ALU
INC
O7
O6
O5
O4
O3
O2
O1
O0
Carry in
Carry out

Memoria ROM

Inputs					Outputs							
I_4	I_3	I_2	I_1	I_0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0
		.							.			
		.							.			
		.							.			
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1

ROM usando un decoder



Bibliografía obligatoria

- Leer capítulo 3 del Libro Null hasta la sección 3.5 INCLUSIVE (Circuitos Combinatorios)