From ZERO To ONE

Arquitectura de Computadores

Fundamentos

Basado en texto: "***Digital Design and Computer Architecture*, 2nd Edition**", David Money Harris and Sarah L. Harris

ELSEVIER

From ZERO To ONE

Tópicos

- **Conceptos básicos**
- **El plan de juego del curso**
- **El arte de manejar la complejidad**
- **La abstracción Digital**
- **Sistemas numéricos**
- **Compuertas Lógicas**
- **Niveles Lógicos**
- **Transistores CMOS**
- **Consumo de Potencia**

ELSEVIER

From ZERO To ONE

Conceptos básicos

- Los microprocesadores han revolucionado nuestro mundo
  - Teléfonos celulares, Internet, avances rápidos en medicina, etc.
- La industria de semiconductores ha crecido desde $21 billones en 1985 hasta los $300 billones en 2011

ELSEVIER

From ZERO To ONE

El Arte de Manejar la Complejidad

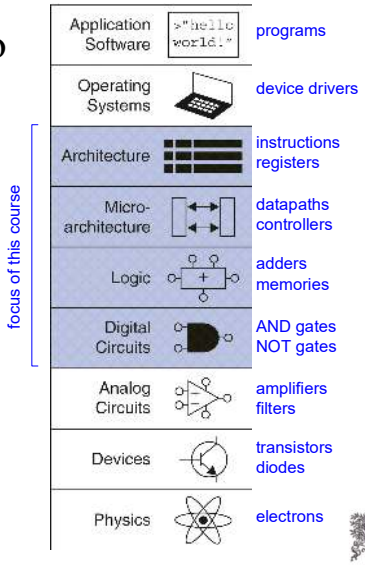
- Abstracción
- Disciplina
- Las tres y's
  - Jerarquía (Hierarch**y**)
  - Modularidad (Modularit**y**)
  - Regularidad (Regularit**y**)

ELSEVIER

From ZERO To ONE

Abstracción

- Ocultar detalles cuando estos no son importantes



From ZERO To ONE

Disciplina

- Intencionalmente restringir la selección del diseño
- Ejemplo: Disciplina digital
  - Voltajes discretos en vez de continuos
  - Es mas simple que diseñar circuitos analógicos – podemos construir mas sistemas sofisticados
  - Sistemas digitales reemplazan a sus predecesores análogos
    - i.e., cámaras digitales, televisión digital, teléfono celular, CDs

From ZERO To ONE

Las tres -y's

- **Jerarquia - Hierarchy**
- **Modularidad - Modularity**
- **Regularidad - Regularity**

From ZERO To ONE

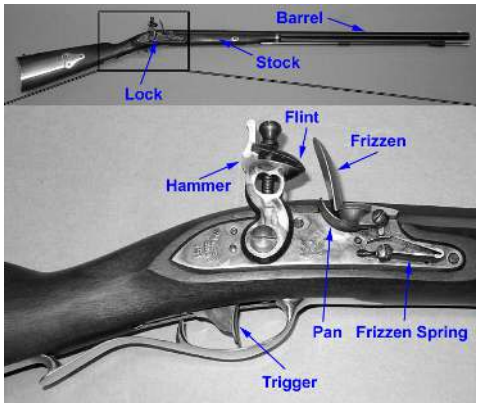
Las tres -y's

- **Jerarquía**
  - Un sistema es dividido en módulos y sub-módulos
- **Modularidad**
  - Poseer interfaces y funciones bien definidas
- **Regularidad**
  - Alentar la uniformidad, de modo que los módulos puedan ser fácilmente reusables

From ZERO To ONE

Ejemplo: Rifle Flintlock

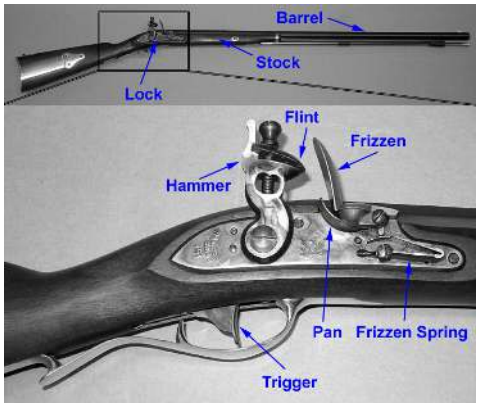
- **Jerarquía**
  - **Hay tres módulos principales:**acción (lock), culata (stock), y cañón (barrel)
  - **Submodulos de la acción:** martillo, percutor seguro, etc.



From ZERO To ONE

Ejemplo: El Rifle Flintlock

- **Modularidad**
  - **Función de la culata:** montar el cañón con la acción
  - **Interfaz de la culata:** largo y ubicación delos pasadores de montaje
- **Regularidad**
  - Partes intercambiables



La Abstracción Digital

From ZERO To ONE

- La mayoría de las variables son **continuas**
  - Voltaje en un cable
  - Frecuencia de oscilador
  - Posición de un objeto
- La abstracción digital considera un **conjunto discreto** de valores

Disciplina Digital : Valores Binarios

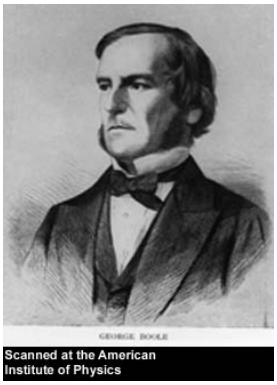
From ZERO To ONE

- **Dos valores discretos:**
  - 1's y 0's
  - 1, VERDADDERO (true), ALTO (high)
  - 0, FALSO (falso), BAJO (low)
- **1 y 0:** niveles de voltaje, engranajes giratorios, niveles de fluido, etc.
- Los circuitos digitales usan niveles de **voltaje** para representar 1 y 0
- ***Bit*:** Dígito *binario*

From ZERO To ONE

George Boole, 1815-1864

- Nació de en una familia de clase media
- Autodidacta de las matemática y fue profesor de la facultad del Queen's College en Ireland
- Escribió *An Investigation of the Laws of Thought* (1854)
- Propuso las variables binarias
- Propuso las tres operaciones lógicas básicas: AND, OR, and NOT



ELSEVIER

From ZERO To ONE

Sistemas numéricos

- Números decimales

1000's column 100's column 10's column 1's column

5374_{10} =

- Números binarios

8's column 4's column 2's column 1's column

1101_2 =

ELSEVIER

From ZERO To ONE

Sistemas numéricos

- Números decimales

1000's column 100's column 10's column 1's column

$$5374_{10} = 5 \times 10^3 + 3 \times 10^2 + 7 \times 10^1 + 4 \times 10^0$$

five thousands; three hundreds; seven tens; four ones

- Números binarios

8's column 4's column 2's column 1's column

$$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{10}$$

one eight; one four; no two; one one

ELSEVIER

From ZERO To ONE

Potencias de dos

- $2^0 =$
- $2^1 =$
- $2^2 =$
- $2^3 =$
- $2^4 =$
- $2^5 =$
- $2^6 =$
- $2^7 =$
- $2^8 =$
- $2^9 =$
- $2^{10} =$
- $2^{11} =$
- $2^{12} =$
- $2^{13} =$
- $2^{14} =$
- $2^{15} =$

ELSEVIER

From ZERO To ONE

Potencias de dos

- $2^0 = 1$
- $2^1 = 2$
- $2^2 = 4$
- $2^3 = 8$
- $2^4 = 16$
- $2^5 = 32$
- $2^6 = 64$
- $2^7 = 128$
- $2^8 = 256$
- $2^9 = 512$
- $2^{10} = 1024$
- $2^{11} = 2048$
- $2^{12} = 4096$
- $2^{13} = 8192$
- $2^{14} = 16384$
- $2^{15} = 32768$
- Es útil memorizar hasta 2^9

From ZERO To ONE

Grandes Potencias de Dos

- 2^{10} = 1 kilo ≈ 1000 (1024)
- 2^{20} = 1 mega ≈ 1 millón (1.048.576)
- 2^{30} = 1 giga ≈ Mil millones (1.073.741.824)

Estimando una potencia de Dos

- ¿Cual es el valor de 2^{24}?
- ¿Cuantos valores distintos puede representar una variable de 32-bit?

Estimando Potencias de Dos

- ¿Cual es el valor de 2^{24}?

  $2^4 \times 2^{20} \approx$ 16 millones

- ¿Cuantos valores distintos puede representar una variable de 32-bit?

  $2^2 \times 2^{30} \approx$ 4 mil millones

Conversión de números

- Conversión binaria a decimal:
  - Convertir 10011_2 a decimal
- Conversión decimal a binaria:
  - Convertir 47_{10} a binaria

Conversión numérica

- Conversión binaria a decimal:
  - Convertir 10011_2 a decimal
  - $16 \times 1 + 8 \times 0 + 4 \times 0 + 2 \times 1 + 1 \times 1 = 19_{10}$
- Conversión decimal a binaria:
  - Convertir 47_{10} a binario
  - $32 \times 1 + 16 \times 0 + 8 \times 1 + 4 \times 1 + 2 \times 1 + 1 \times 1 = 101111_2$

Rango y valores binarios

- Números decimales de N-dígitos
  - ¿Cuantos valores distintos?
  - ¿Rango?
  - Ejemplo: numero decimal de 3-dígitos:
- Números binarios de N-bits
  - ¿Cuantos valores distintos?
  - Rango:
  - Ejemplo: numero binario de 3-bits:

Valores Binarios y Rango

- Numero decimales de N-dígitos
  - ¿Cuantos valores distintos? 10^N
  - ¿Rango? $[0, 10^N - 1]$
  - Ejemplo: numero decimal de 3-dígitos:
    - $10^3 = 1000$ valores posibles
    - Rango: [0, 999]
- *Numero binario de N*-bit
  - ¿Cuantos valores distintos? 2^N
  - Rango: $[0, 2^N - 1]$
  - Ejemplo: numero binario de 3-bits:
    - $2^3 = 8$ valores posibles
    - Rango: $[0, 7] = [000_2$ al $111_2]$

From ZERO To ONE

Números Hexadecimales

Digito Hex	Equivalente Decimal	Equivalente Binario
0	0	
1	1	
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	
9	9	
A	10	
B	11	
C	12	
D	13	
E	14	
F	15	

  ELSEVIER

From ZERO To ONE

Números Hexadecimales

Digito Hex	Equivalente Decimal	Equivalente Binario
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
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

  ELSEVIER

From ZERO To ONE

Números Hexadecimales

- Base 16
- Notación abreviada para los representar números binarios

From ZERO To ONE

Conversión Hexadecimal a Binario

- Conversión Hexadecimal a binario:
  - Convertir $4AF_{16}$ (también se escribe como 0x4AF) a binario
- Conversión Hexadecimal a decimal:
  - Convertir 0x4AF a decimal

From ZERO To ONE

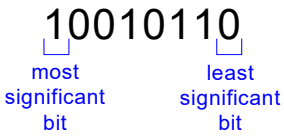
Conversión hexadecimal a binaria

- Conversión hexadecimal a binario:
  - Convertir $4AF_{16}$ (también se escribe como 0x4AF) a binario
  - $0100\ 1010\ 1111_2$
- Conversión hexadecimal a decimal
  - Convertir $4AF_{16}$ a decimal
  - $16^2 \times 4 + 16^1 \times 10 + 16^0 \times 15 = 1199_{10}$

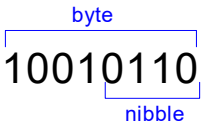
From ZERO To ONE

Bits, Bytes, Nibbles…

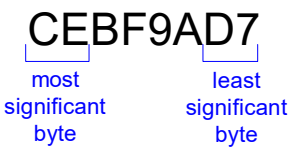
- Bits



- Bytes & Nibbles



- Bytes



¿Por que el byte mas significativo se compone de dos dígitos hexadecimales?

From ZERO To ONE

Suma

- Decimal

```
   3734
+  5168
------
```

- Binaria

```
   1011
+  0011
------
```


From ZERO To ONE

Suma

- Decimal

```
    11  ← carries
   3734
+  5168
------
   8902
```

- Binaria

```
    11  ← carries
   1011
+  0011
------
   1110
```


From ZERO To ONE

Ejemplos de Suma Binaria

- Sume los siguientes números binarios de 4-bit

```
  1001
+ 0101
------
```

- Sume los siguientes números binarios de 4-bit

```
  1011
+ 0110
------
```


From ZERO To ONE

Ejemplos de Suma Binaria

- Sume los siguientes números binarios de 4-bit

```
    1
  1001
+ 0101
------
 1110
```

- Sume los siguientes números binarios de 4-bit

```
  111
  1011
+ 0110
------
 10001
```

¡Desbordamiento!

Desbordamiento

- Los sistemas digitales operan sobre un **numero fijo de bits**
- Desbordamiento (Overflow): cuando el resultado es demasiado grande para calzar en los bits disponibles
- Vea el ejemplo previo de 11 + 6

Números Binarios con Signo

- Números con Signo/Magnitud
- Números en complemento de dos

From ZERO To ONE

Números con Signo/Magnitud

- 1 bit de signo, N-1 bits para magnitud
- Bit signo es el mas significativo, el bit mas a la izquierda

$$A: \{a_{N-1}, a_{N-2}, \cdots a_2, a_1, a_0\}$$

  - Numero positivo: bit signo = 0
  - Numero negativo: bit signo = 1

$$A = (-1)^{a_{n-1}} \sum_{i=0}^{n-2} a_i 2^i$$

- Ejemplo, representación $\pm$ 6 con sign/mag de 4 bits:
  +6 =
  - 6 =
- Rango de numero con signo/magnitud de N-bit:

ELSEVIER

From ZERO To ONE

Números con Signo/Magnitud

- 1 bit de signo, N-1 bits para magnitud
- Bit signo es el mas significativo, el bit mas a la izquierda

$$A: \{a_{N-1}, a_{N-2}, \ldots, a_2, a_1, a_0\}$$

  - Numero positivo: bit signo = 0
  - Numero negativo: bit signo = 1

$$A = (-1)^{a_{n-1}} \sum_{i=0}^{n-2} a_i 2^i$$

- Ejemplo, representación $\pm$ 6 con sign/mag de 4 bits:
  +6 = **0110**
  - 6 = **1110**
- Rango de numero con signo/magnitud de N-bit:
  $[-(2^{N-1}-1), 2^{N-1}-1$

ELSEVIER

Números con Signo/Magnitud

- Problemas:
  - Suma no funciona, por ejemplo -6 + 6:

```
  1110
+ 0110
------
 10100 (¡error!)
```

  - Dos representaciones del 0 (± 0):

```
1000
0000
```


Números complemento de dos

- No tenemos los problemas de los números con signo/magnitud:
  - Suma funciona
  - Una sola representación para el 0

From ZERO To ONE

Números en Complemento de Dos

- Msb tiene el valor de -2^{N-1}

$$A = a_{n-1}\left(-2^{n-1}\right) + \sum_{i=0}^{n-2} a_i 2^i$$

- El mayor numero positivo de 4-bit:
- El numero mas negativo de 4-bit:
- El bit mas significativo aun indica el signo (1 = negativo, 0 = positivo)
- Rango de un numero de *N*-bit en complemento de dos:

From ZERO To ONE

Números en Complemento de Dos

- Msb tiene el valor de -2^{N-1}

$$A = a_{n-1}\left(-2^{n-1}\right) + \sum_{i=0}^{n-2} a_i 2^i$$

- El mayor numero positivo de 4-bit: **0111**
- El numero mas negativo de 4-bit: **1000**
- El bit mas significativo aun indica el signo (1 = negativo, 0 = positivo)
- Rango de un numero de *N*-bit en complemento de dos: **$[-(2^{N-1}), 2^{N-1}-1]$**

From ZERO To ONE

"Tomando el complemento de dos"

- Invierta el signo del numero en complemento de dos
- Método:
  1. Invertir los bits
  2. Sume 1
- Ejemplo: Invertir el signo de $3_{10} = 0011_2$

From ZERO To ONE

"Tomando complemento de dos"

- Invertir el signo del numero en complemento de dos
- Método:
  1. Invertir los bits
  2. Sume 1
- Ejemplo: Invierta el signo de $3_{10} = 0011_2$

  1. **1100**
  2. **+ 1**

  $1101 = -3_{10}$

From ZERO To ONE

Ejemplos de Complemento de Dos

- Tome el complemento de dos de $6_{10} = 0110_2$
- ¿Cual es el valor decimal de 1001_2?

From ZERO To ONE

Ejemplos de Complemento de Dos

- Tome el complemento de dos de $6_{10} = 0110_2$
  1. 1001
  2. + 1

  $1010_2 = -6_{10}$
- ¿Cual es el valor decimal del numero en complemento de dos 1001_2?
  1. 0110
  2. + 1

  $0111_2 = 7_{10}$, luego $1001_2 = -7_{10}$

Suma de Complemento de dos

- Sume 6 + (-6) con números en complemento de dos

```
  0110
+ 1010
------
```

- Sume -2 + 3 con numero en complemento de dos

```
  1110
+ 0011
------
```

From ZERO To ONE

Suma de Complementos de Dos

- Sume 6 + (-6) con números en complemento de dos

```
 111
  0110
+ 1010
------
 10000
```

- Sume -2 + 3 con números en complemento de dos

```
 111
  1110
+ 0011
------
 10001
```

From ZERO To ONE

From ZERO To ONE

Incrementar Ancho Bit

- **Extender numero de *N* a *M* bits (*M* > *N*) :**
  - Extensión de signo
  - Extensión de cero

From ZERO To ONE

Extensión de signo

- Bit Signo copiado al msb's
- Valor del numero es el mismo

- **Ejemplo 1:**
  - Representación del 3 en 4-bit = 0011
  - Valor con signo extendido de 8-bit: 00000011
- **Ejemplo 2:**
  - Representación de -5 en 4-bit = 1011
  - Valor con signo extendido de 8-bit: 11111011

From ZERO To ONE

Extensión Cero

- Ceros copiados al msb's
- Valor cambia para números negativos

- **Ejemplo 1:**
  - Valor 4-bit = $0011_2 = 3_{10}$
  - Valor con cero extendido de 8-bit: 00000011 = 3_{10}
- **Ejemplo 2:**
  - Valor 4-bit = $1011 = -5_{10}$
  - Valor con signo extendido de 8-bit: 00001011 = 11_{10}

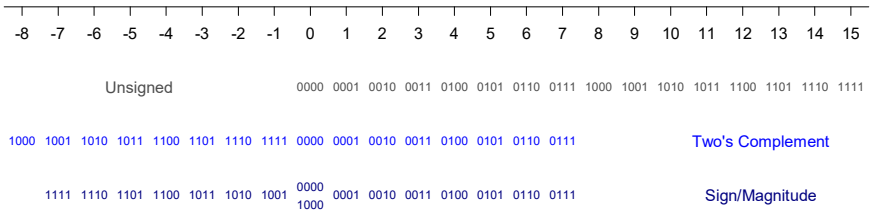
  ELSEVIER

From ZERO To ONE

Resumen: Comparación de Sistemas Numéricos

Sistema Numérico	Rango
Sin Signo	$[0, 2^N-1]$
Signo/Magnitud	$[-(2^{N-1}-1), 2^{N-1}-1]$
Complemento Dos	$[-2^{N-1}, 2^{N-1}-1]$

Por ejemplo, representación con 4 bits:



  ELSEVIER

From ZERO To ONE

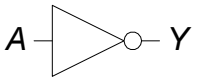
Compuertas Lógicas

- **Realizar funciones lógicas:**
  - inversión (NOT), AND, OR, NAND, NOR, etc.
- **Simple-entrada:**
  - Compuerta NOT, buffer
- **Dos-entradas:**
  - AND, OR, XOR, NAND, NOR, XNOR
- **Múltiples-entradas**

From ZERO To ONE

Compuertas Lógicas de una entrada

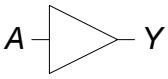
NOT



$Y = \overline{A}$

A	Y
0	
1	

BUF



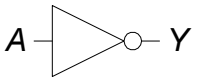
$Y = A$

A	Y
0	
1	

From ZERO To ONE

Compuertas Lógicas de una entrada

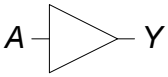
NOT



$Y = \overline{A}$

A	Y
0	1
1	0

BUF



$Y = A$

A	Y
0	0
1	1

From ZERO To ONE

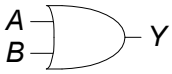
Compuertas Lógicas de Dos Entradas

AND

$Y = AB$

A	B	Y
0	0	
0	1	
1	0	
1	1	

OR



$Y = A + B$

A	B	Y
0	0	
0	1	
1	0	
1	1	

From ZERO To ONE

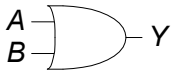
Compuertas Lógicas de Dos Entradas

AND

$Y = AB$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR



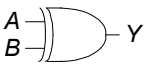
$Y = A + B$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

From ZERO To ONE

Mas Compuertas Lógicas de Dos Entradas

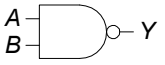
XOR



$Y = A \oplus B$

A	B	Y
0	0	
0	1	
1	0	
1	1	

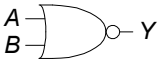
NAND



$Y = \overline{AB}$

A	B	Y
0	0	
0	1	
1	0	
1	1	

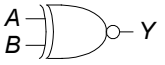
NOR



$Y = \overline{A + B}$

A	B	Y
0	0	
0	1	
1	0	
1	1	

XNOR



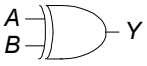
$Y = \overline{A \oplus B}$

A	B	Y
0	0	
0	1	
1	0	
1	1	

From ZERO To ONE

Mas Compuertas Lógicas de Dos Entradas

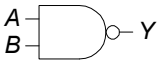
XOR



$Y = A \oplus B$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

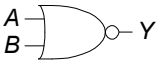
NAND



$Y = \overline{AB}$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

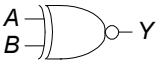
NOR



$Y = \overline{A + B}$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

XNOR



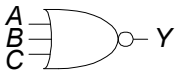
$Y = \overline{A \oplus B}$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

From ZERO To ONE

Compuertas Lógicas de Múltiples Entradas

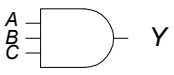
NOR3



$Y = \overline{A+B+C}$

A	B	C	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

AND3



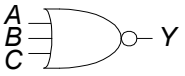
$Y = ABC$

A	B	C	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

From ZERO To ONE

Compuertas Lógicas de Múltiples Entradas

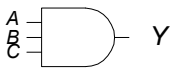
NOR3



$Y = \overline{A+B+C}$

A	B	C	Y
0	0	0	1
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	0

AND3



$Y = ABC$

A	B	C	Y
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

- XOR multi entrada: paridad impar

From ZERO To ONE

Niveles Lógicos

- Voltajes discretos representan un 1 y un 0
- Por ejemplo:
  - 0 = tierra/*ground* (GND) o 0 volts
  - 1 = V_{DD} o 5 volts
- ¿Que hay de 4,99 volts? Es eso un 0 o un 1?
- ¿Que hay de 3,2 volts?

From ZERO To ONE

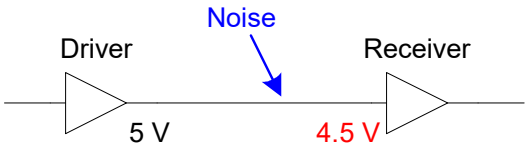
Niveles Lógicos

- *Rango de* voltajes para 1 y para 0
- Diferentes rangos para la entrada y la salida se definen debido al ruido

From ZERO To ONE

¿Que es el ruido?

- **Cualquier cosa que degrada a una señal**
  - E.g., resistencia, ruido eléctrico de la fuente de poder, inducción por cables cercanos, etc.
- **Ejemplo:** Una compuerta (Driver) entrega 5 V de salida pero, debido a la resistencia eléctrica en un cable largo, el receptor obtiene 4.5 V



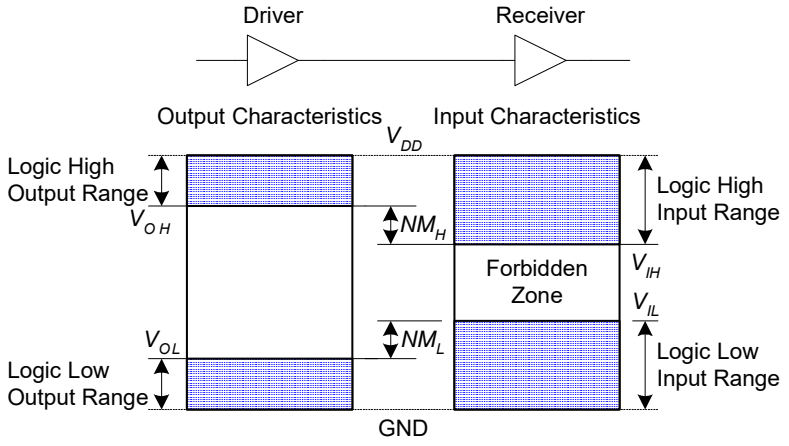
From ZERO To ONE

La Disciplina de la Estática

- Con entradas lógicas validas, cada elemento de circuito debe producir salidas lógicas validas
- Use rangos limitados de voltaje para representar valores discretos

From ZERO To ONE

Niveles Lógicos



From ZERO To ONE

Ajuste de V_{DD}

- En los años 70 y 80, V_{DD} = 5 V
- V_{DD} se ha ido reduciendo
  - Evitar freír a transistores muy pequeños
  - Ahorro de energía
- 3.3 V, 2.5 V, 1.8 V, 1.5 V, 1.2 V, 1.0 V, …
- Se debe ser cuidadoso al conectar chips con diferentes voltajes de alimentación

Los chips funcionan porque poseen un humo mágico

Demostración:

- Si se deja salir el humo mágico, el chip deja de funcionar

From ZERO To ONE

Ejemplo de Familias Lógicas

Familia Lógica	V_{DD}	V_{IL}	V_{IH}	V_{OL}	V_{OH}
TTL	5 (4,75 – 5,25)	0,8	2,0	0,4	2,4
CMOS	5 (4,5 - 6)	1,35	3,15	0,33	3,84
LVTTL	3.3 (3 – 3,6)	0,8	2,0	0,4	2,4
LVCMOS	3,3 (3 – 3,6)	0,9	1,8	0,36	2,7

From ZERO To ONE

Transistores y Relés

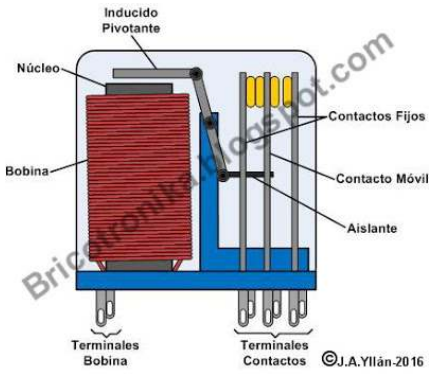
- Antiguamente las compuertas lógicas se construían con relés
- Hoy en día se construyen de transistores
- Tanto relés como los transistores pueden ser entendidos como interruptores controlados por voltaje o por una fuerza mecánica

From ZERO To ONE

Relé: Conmutador electromecánico

Un relé o relevador es básicamente **un interruptor** o conmutador accionado por un electroimán.

- El electroimán esta compuesto por una barra de hierro dulce, llamado núcleo,
- El núcleo esta rodeado por una bobina de hilo de cobre.



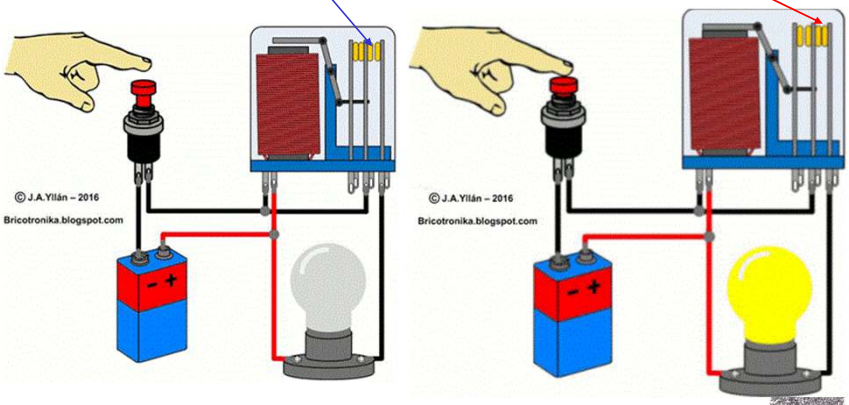
http://bricotronika.blogspot.com/2016/04/como-funciona-un-rele-electromecanico.html

From ZERO To ONE

Relé: funcionamiento

Contactos normalmente abiertos

Al presionar interruptor: contactos se cierran

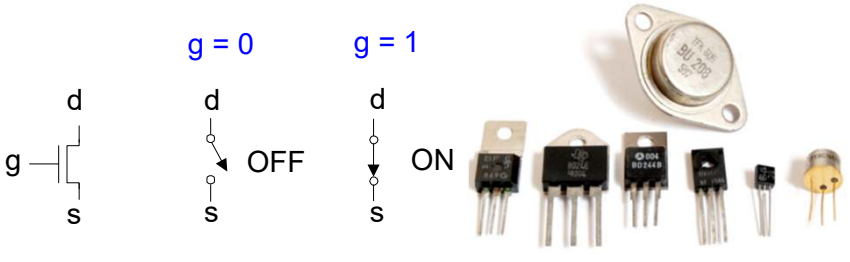


https://bit.ly/2kqWxyC

From ZERO To ONE

Transistor

- Interruptor de 3 puertos controlado por voltaje
  - 2 puertos se conectan dependiendo del voltaje del 3ro
  - d y s se conectan (ON) cuando g es 1



From ZERO To ONE

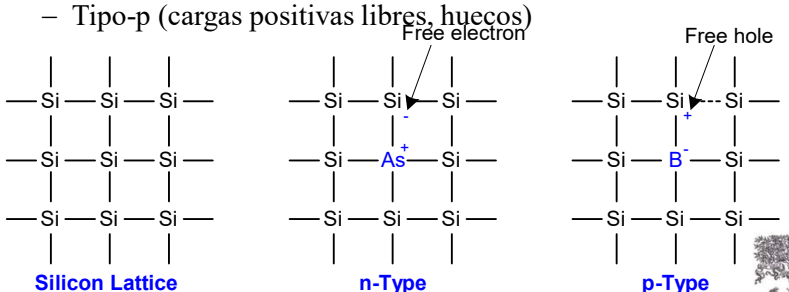
Robert Noyce, 1927-1990

- Apodado "El alcalde de Silicon Valley"
- Co-fundo Fairchild Semiconductor en 1957
- Co-fundo Intel en 1968
- Co-inventó el circuito integrado

From ZERO To ONE

Silicio

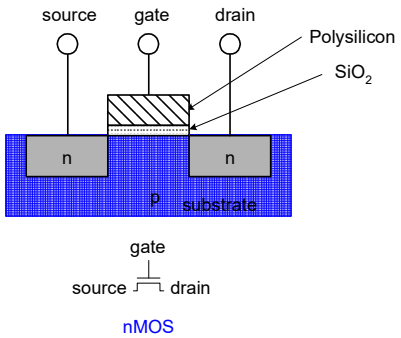
- Los transistores se construyen de silicio, que es un semiconductor
- El silicio puro es un conductor muy pobre (sin cargas libres)
- El silicio dopado es un buen conductor (cargas libres)
  - Tipo-n (cargas negativas libres, electrones)
  - Tipo-p (cargas positivas libres, huecos)



From ZERO To ONE

Transistores MOS

- **Transistores Metal oxido silicio (MOS):**
  - Compuerta de Polisilicio (solía ser **metal**) e
  - Aislante **Oxido** (dioxido de silicio)
  - **Silicio** dopado

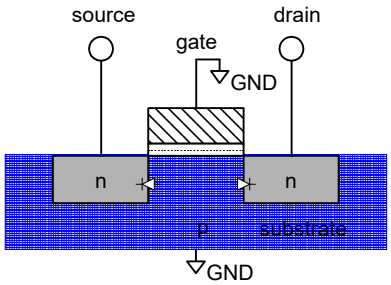


From ZERO To ONE

Transistores: nMOS

Gate (puerta) = 0

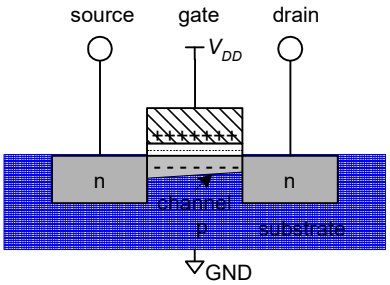
OFF (no hay conexión entre source/emisor y drain/colector)



Interruptor abierto: No circula corriente entre emisor y colector

Gate (puerta) = 1

ON (se crea canal entre source/emisor y drain/colector)

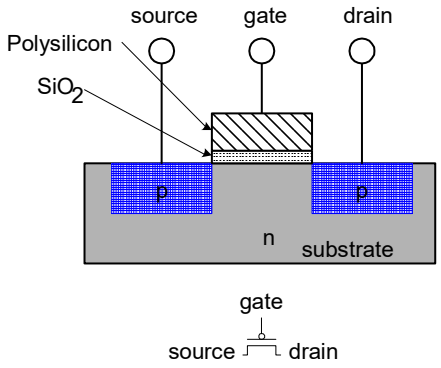


Interruptor cerrado: Si circula corriente entre emisor y colector

From ZERO To ONE

Transistores: pMOS

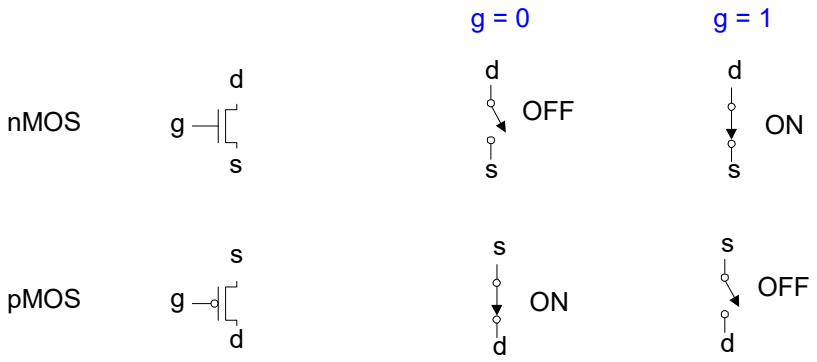
- El transistor pMOS opera de forma contraria
  - ON cuando Gate = 0
  - OFF cuando Gate = 1



ELSEVIER

From ZERO To ONE

Resumen: Función del Transistor

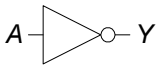


ELSEVIER

From ZERO To ONE

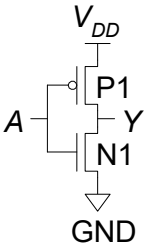
Compuertas CMOS: Compuerta NOT

NOT



$Y = \overline{A}$

A	Y
0	1
1	0

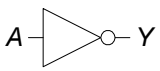


A	P1	N1	Y
0			
1			

From ZERO To ONE

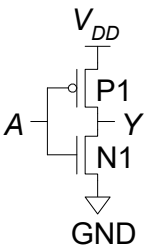
Compuertas CMOS: Compuerta NOT

NOT



$Y = \overline{A}$

A	Y
0	1
1	0

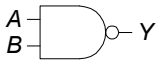


A	P1	N1	Y
0	ON	OFF	1
1	OFF	ON	0

From ZERO To ONE

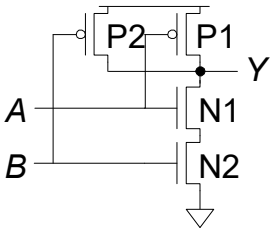
Compuertas CMOS: Compuerta NAND

NAND



$Y = \overline{AB}$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

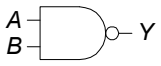


A	B	P1	P2	N1	N2	Y
0	0					
0	1					
1	0					
1	1					

From ZERO To ONE

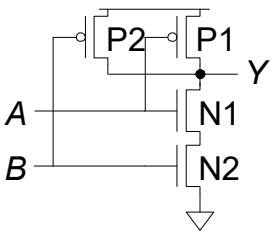
Compuertas CMOS: Compuerta NAND

NAND



$Y = \overline{AB}$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



A	B	P1	P2	N1	N2	Y
0	0	ON	ON	OFF	OFF	1
0	1	ON	OFF	OFF	ON	1
1	0	OFF	ON	ON	OFF	1
1	1	OFF	OFF	ON	ON	0

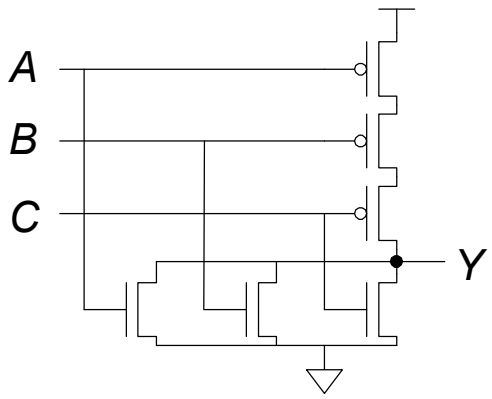
From ZERO To ONE

Compuerta NOR

¿Como construir un compuerta NOR de tres entradas?

From ZERO To ONE

Compuerta NOR3



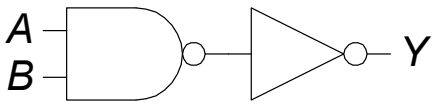
From ZERO To ONE

Otras Compuertas CMOS

¿Como construir una compuerta AND de dos entradas?

From ZERO To ONE

Compuerta AND2



From ZERO To ONE

Resumen

- Puede construir cualquier compuerta lógica (NOT, AND, OR) solo con interruptores
  - Antiguamente Relés
  - Hoy en día transistores

- En realidad usted necesita solo construir compuertas NAND o NOR
  - Lo veremos formalmente mas adelante (algebra de Boole)