Organización de computadoras

Clase 8

Universidad Nacional de Quilmes

Lic. Martínez Federico

dom	lun	mar	mié	jue	vie	sáb
¿Qué pasó la ultima clase?			1 Día del Trabajo	2	3	4
5	6	7	8	9	10 Dia de las Madres	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

¿Qué pasó ?

¿Qué pasó ?

- Signo Magnitud

¿Qué pasó ?

- Signo Magnitud
  - Interpretación

¿Qué pasó ?

- Signo Magnitud
  - Interpretación
  - Representación

¿Qué pasó ?

- Signo Magnitud
  - Interpretación
  - Representación
  - Rango

¿Qué pasó ?

- Signo Magnitud
  - Interpretación
  - Representación
  - Rango
- Notación científica

¿Qué pasó ?

- Signo Magnitud
  - Interpretación
  - Representación
  - Rango
- Notación científica
- Punto flotante

¿Qué pasó ?

- Signo Magnitud
  - Interpretación
  - Representación
  - Rango
- Notación científica
- Punto flotante
  - Interpretación

¿Qué pasó ?

- Signo Magnitud
  - Interpretación
  - Representación
  - Rango
- Notación científica
- Punto flotante
  - Interpretación
  - Rango

¿Qué pasó ?

- Signo Magnitud
  - Interpretación
  - Representación
  - Rango
- Notación científica
- Punto flotante
  - Interpretación
  - Rango
  - Resolución variable

¿Qué pasó ?

- Signo Magnitud
  - Interpretación
  - Representación
  - Rango
- Notación científica
- Punto flotante
  - Interpretación
  - Rango
  - Resolución variable
  - Mantisa entera y fraccionaria

¿Qué pasó ?

- Signo Magnitud
  - Interpretación
  - Representación
  - Rango
- Notación científica
- Punto flotante
  - Interpretación
  - Rango
  - Resolución variable
  - Mantisa entera y fraccionaria
  - Normalización

¿Qué pasó ?

- Signo Magnitud
  - Interpretación
  - Representación
  - Rango
- Notación científica
- Punto flotante
  - Interpretación
  - Rango
  - Resolución variable
  - Mantisa entera y fraccionaria
  - Normalización
  - Bit implícito

¿Qué pasó ?

- Signo Magnitud
  - Interpretación
  - Representación
  - Rango
- Notación científica
- Punto flotante
  - Interpretación
  - Rango
  - Resolución variable
  - Mantisa entera y fraccionaria
  - Normalización
  - Bit implícito
  - Representación

¿Qué se viene para hoy?

¿Qué se viene para hoy?

- Compuertas lógicas:

¿Qué se viene para hoy?

- Compuertas lógicas:
  - ¿Qué?

¿Qué se viene para hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR

¿Qué se viene para hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR
  - Compuerta AND

¿Qué se viene para hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR
  - Compuerta AND
  - Compuerta NOT

¿Qué se viene para hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR
  - Compuerta AND
  - Compuerta NOT
  - Otras compuertas

¿Qué se viene para hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR
  - Compuerta AND
  - Compuerta NOT
  - Otras compuertas
- Circuitos

¿Qué se viene para hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR
  - Compuerta AND
  - Compuerta NOT
  - Otras compuertas
- Circuitos
  - Formulas y tablas de verdad

¿Qué se viene para hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR
  - Compuerta AND
  - Compuerta NOT
  - Otras compuertas
- Circuitos
  - Formulas y tablas de verdad
  - Producto de sumas y suma de productos

¿Qué se viene para hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR
  - Compuerta AND
  - Compuerta NOT
  - Otras compuertas
- Circuitos
  - Formulas y tablas de verdad
  - Producto de sumas y suma de productos
  - Circuitos comunes

¿Qué se viene para hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR
  - Compuerta AND
  - Compuerta NOT
  - Otras compuertas
- Circuitos
  - Formulas y tablas de verdad
  - Producto de sumas y suma de productos
  - Circuitos comunes
  - Circuitos aritméticos

Niveles de abstracción

- La computadora puede considerarse desde distintos niveles de abstracción

Aplicaciones de Usuario


Lenguaje de alto nivel

string_comp.s

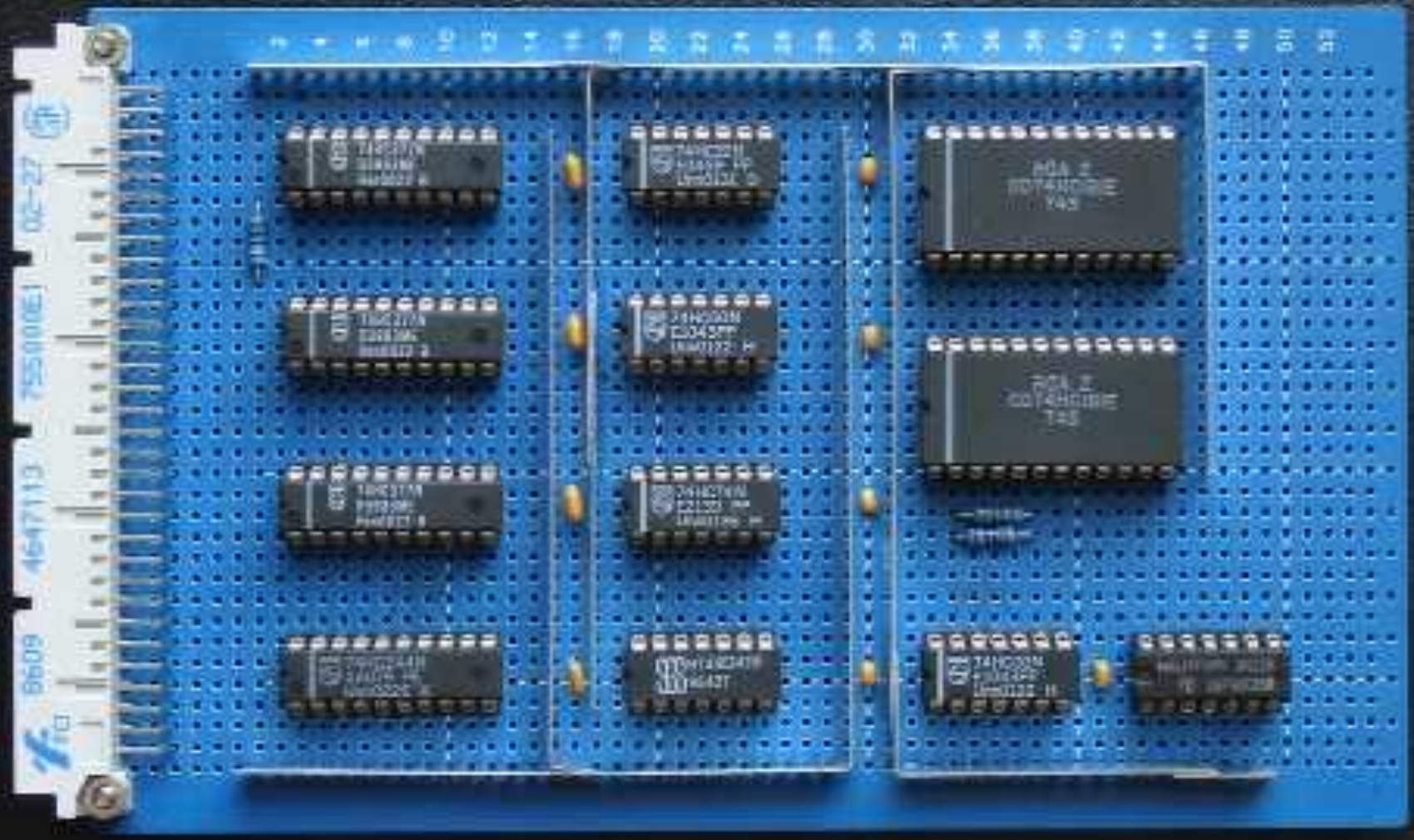
```
1  ; C Shared Library Example for ARM Linux
2  ;
3  ; Copyright (C) ARM Limited, 2007-2010. All rights reserved.
4
5  ; String comparison
6  ; This is pointless for a real application as you can just use strcmp, which is more e
7  ; However, this is a useful example of how to use assembly functions in a Linux applic
8
9      AREA strcomp, CODE, READONLY
10
11     EXPORT string_compare
12     CODE32
13
14 string_compare
15     ; int string_compare (char * a, char * b);
16     ; Compares a and b. Returns <0 if a < b, 0 if a = b, >0 if a > b
17
18     LDRB r2, [r0], #1
19     LDRB r3, [r1], #1
20     CMP r2, #1
21     CMPHS r2, r3
22     BEQ string_compare
23     SUB r0, r2, r3
24     BX r14
25
26     END
27
```

Assembler

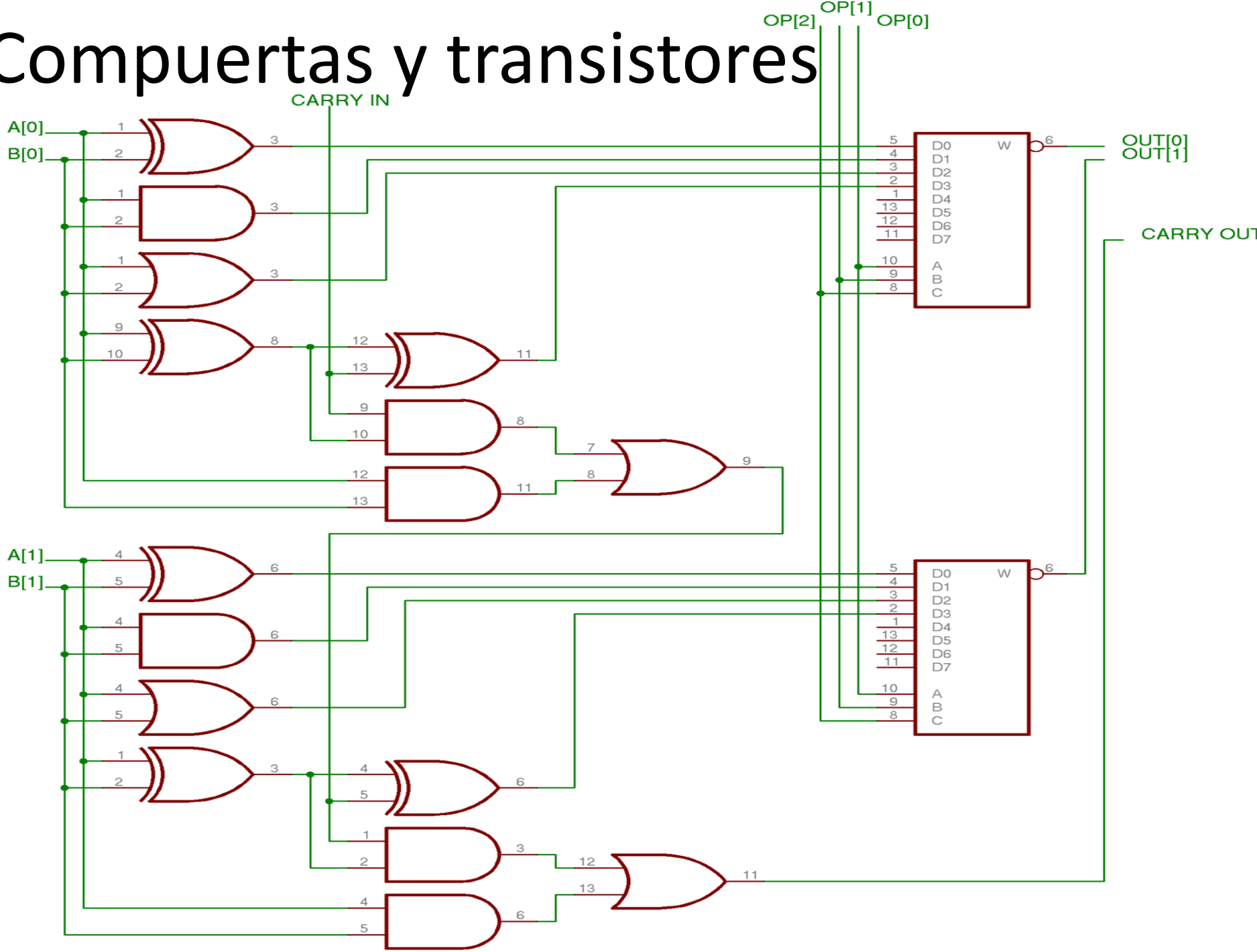
string_comp.s

Control:
Hardwiring y microporgramación

Unidades funcionales



Compuertas y transistores



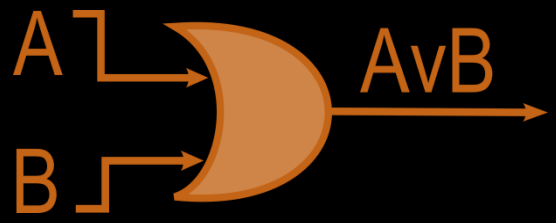
Compuertas

Compuertas

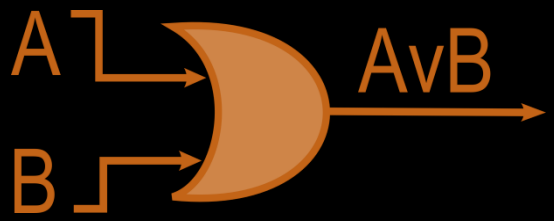
- es un dispositivo que implementa una función booleana simple.

- Traduce un conjunto de entradas (una o mas) en una salida

Compuerta OR

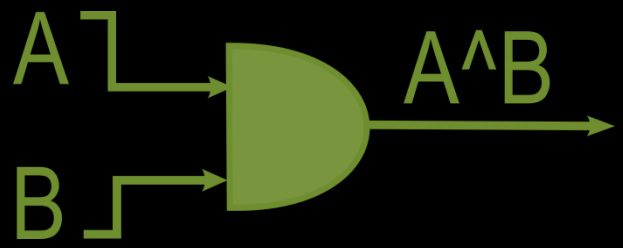


Compuerta OR

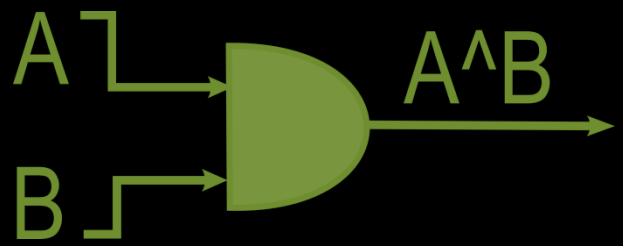


A	B	AvB
0	0	0
0	1	1
1	0	1
1	1	1

Compuerta AND

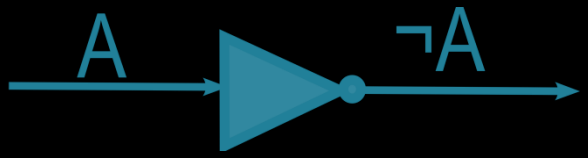


Compuerta AND

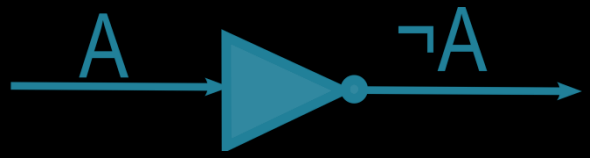


A	B	A^B
0	0	0
0	1	0
1	0	0
1	1	1

Compuerta NOT



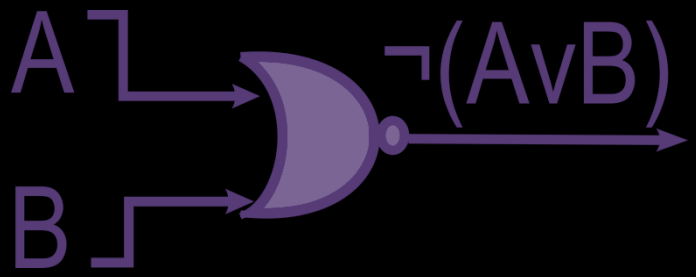
Compuerta NOT



A	¬A
0	1
1	0

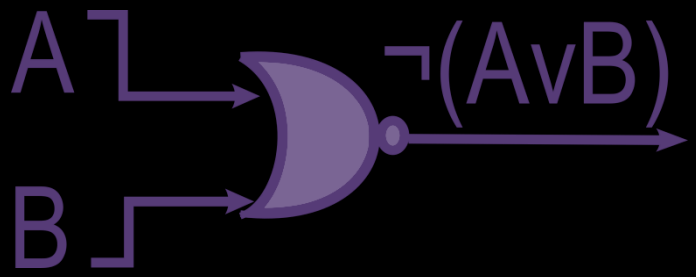
Otras

Otras

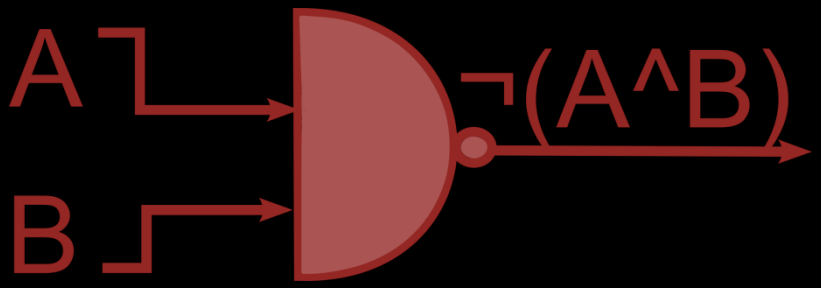


Compuerta NOR

Otras

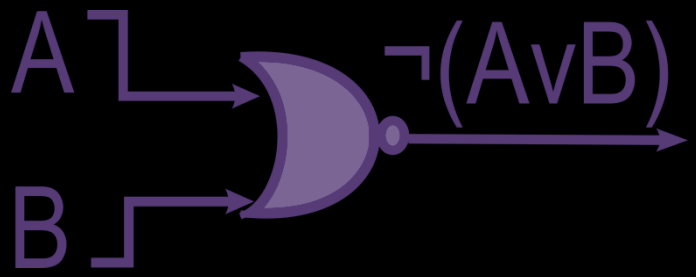


Compuerta NOR

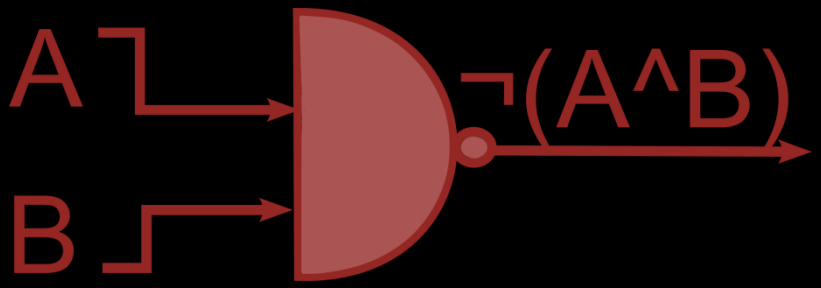


Compuerta NAND

Otras



Compuerta NOR



Compuerta NAND

Compuerta XOR


Circuitos

Circuitos

- Traducen un conjunto de entradas en un conjunto de salidas.

Circuitos

- Traducen un conjunto de entradas en un conjunto de salidas.

- Una o mas funciones booleanas.

Circuitos

- Traducen un conjunto de entradas en un conjunto de salidas.
- Una o mas funciones booleanas.
- Se obtienen combinando compuertas.

Circuitos

- Se pueden construir a partir de una formula booleana o a partir de una tabla de verdad

- Ejemplo: Construir un circuito que compute cada una de las siguientes funciónes:
  - B^(C v A)
  - (A ^ B) v (¬A ^ C)

Circuitos

- ¿Cómo pasar de la tabla al circuito?

Circuitos

- Suma de productos:
  - Una formula tiene la forma de suma de productos si tiene la siguiente pinta:
    - A1 v A2 v A3 v …. An, donde cada Ai usa solo and y not
  - Ej: (A^¬B) v (¬A^C) v (B ^ C)

Circuitos

- Producto de sumas:
  - Una formula tiene la forma de producto de sumas si tiene la siguiente pinta:
    - A1 ^ A2 ^ A3 ^ …. An, donde cada Ai usa solo or y not
  - Ej: (Av¬B) ^ (¬AvC) ^ (B v C)

Circuitos

- ¿Cómo pasar de la tabla al circuito?
  1. Armamos la tabla
  2. Si hay mas filas con resultado 1:
     1. Escribimos un producto por cada una de estas filas
     2. Las sumamos
     3. Armamos el circuito a partir de la fórmula
  3. Si hay mas filas con resultado 0:
     1. Escribimos una suma por cada una de estas filas
     2. Hacemos el producto entre ellas
     3. Armamos el circuito a partir de la fórmula

Ejemplo

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

Ejemplo

A	B	C	F
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	0

Ejercicio

- Realizar un circuito de 3 entradas que compute la función mayoría, es decir, si dos o mas entradas valen 1 debe obtenerse un 1, y un 0 si no.

Circuitos útiles

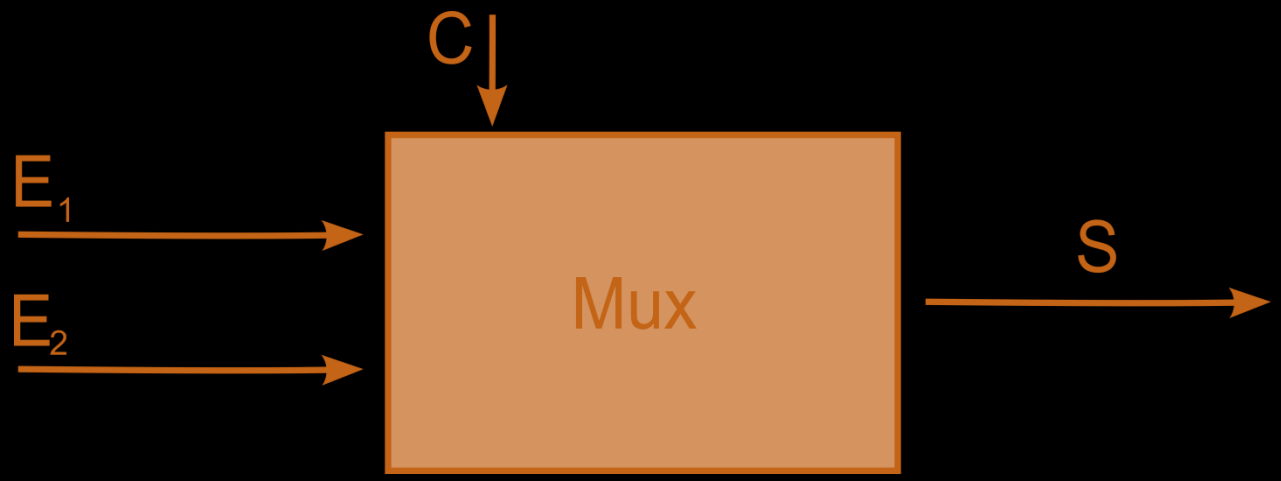
Multiplexor simple

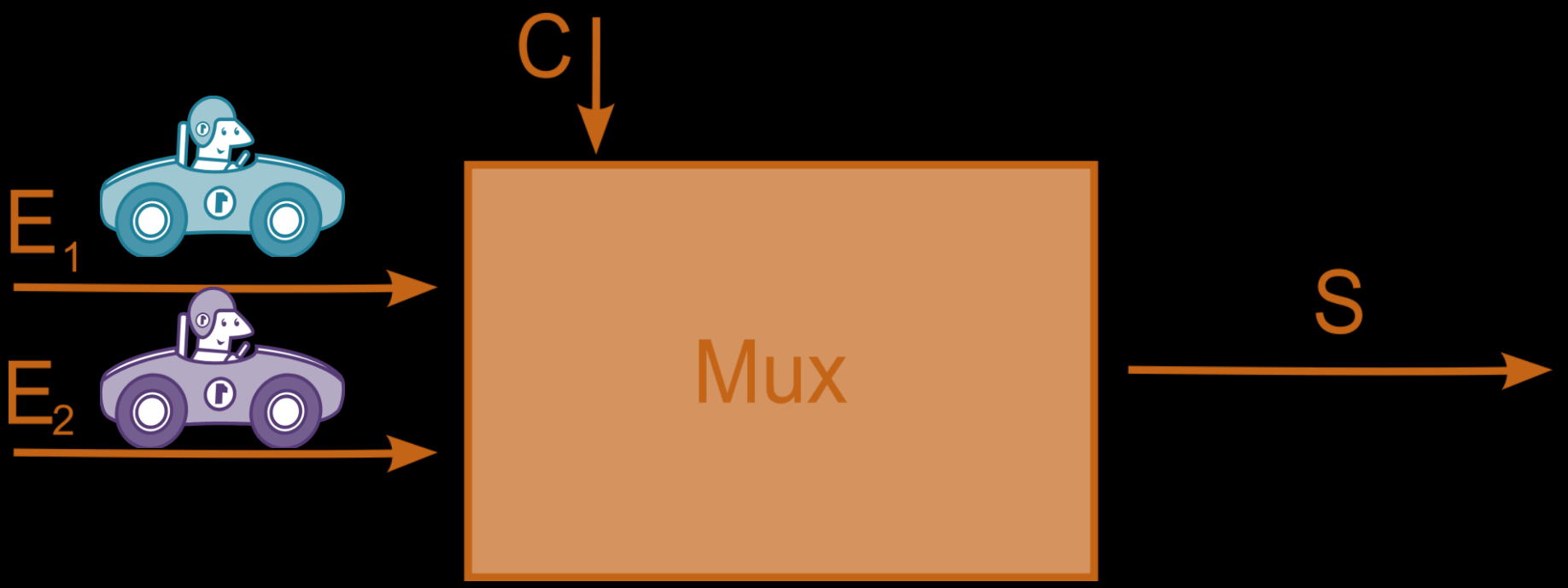
Multiplexor simple

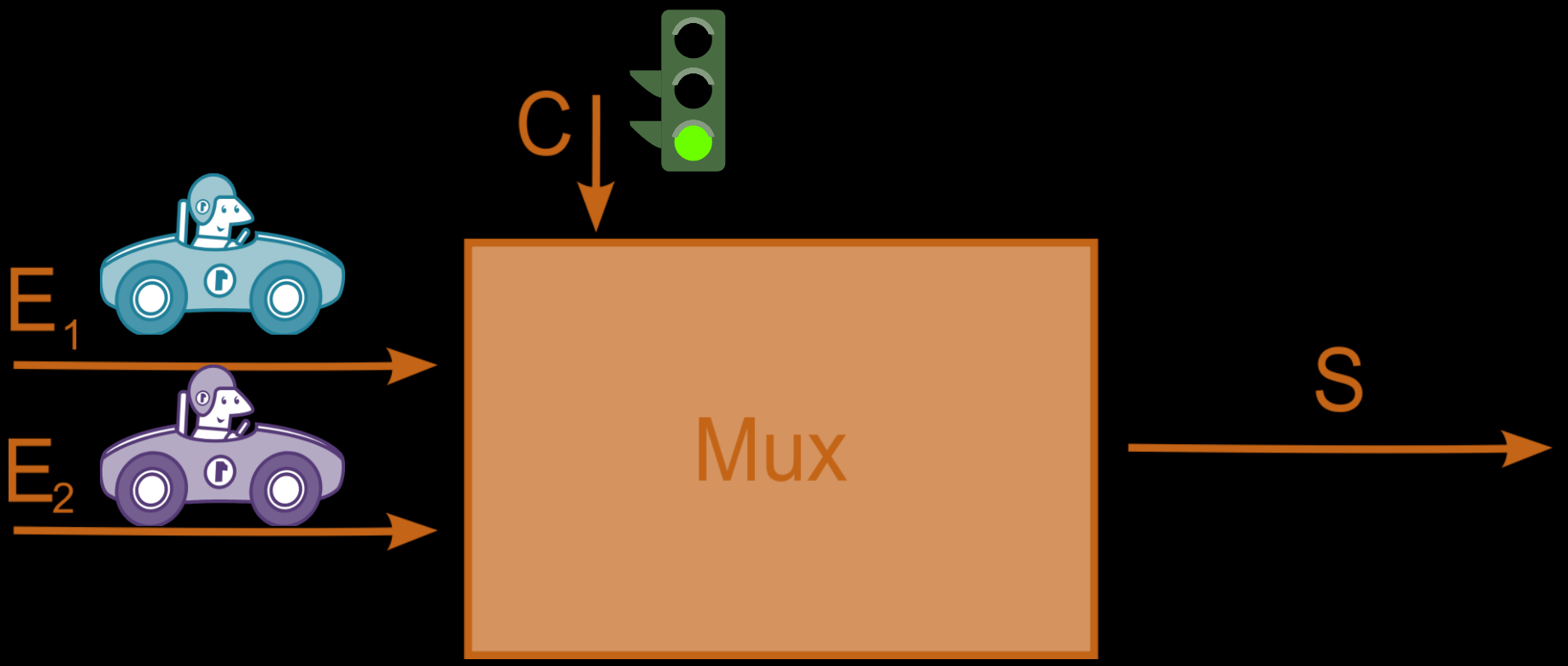
- 2 entradas
- 1 salida
- Una línea de control que elige cuál de las entradas se proyecta a la salida.

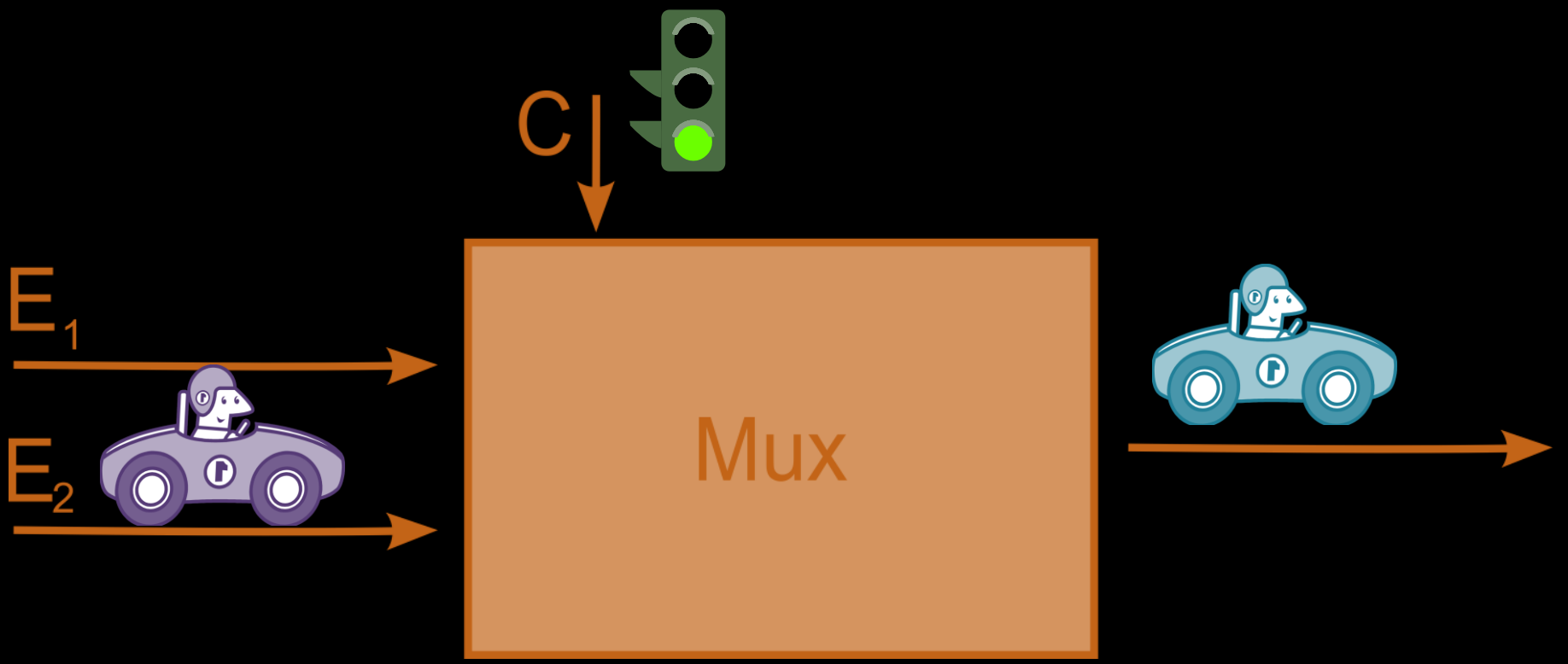
Multiplexor simple

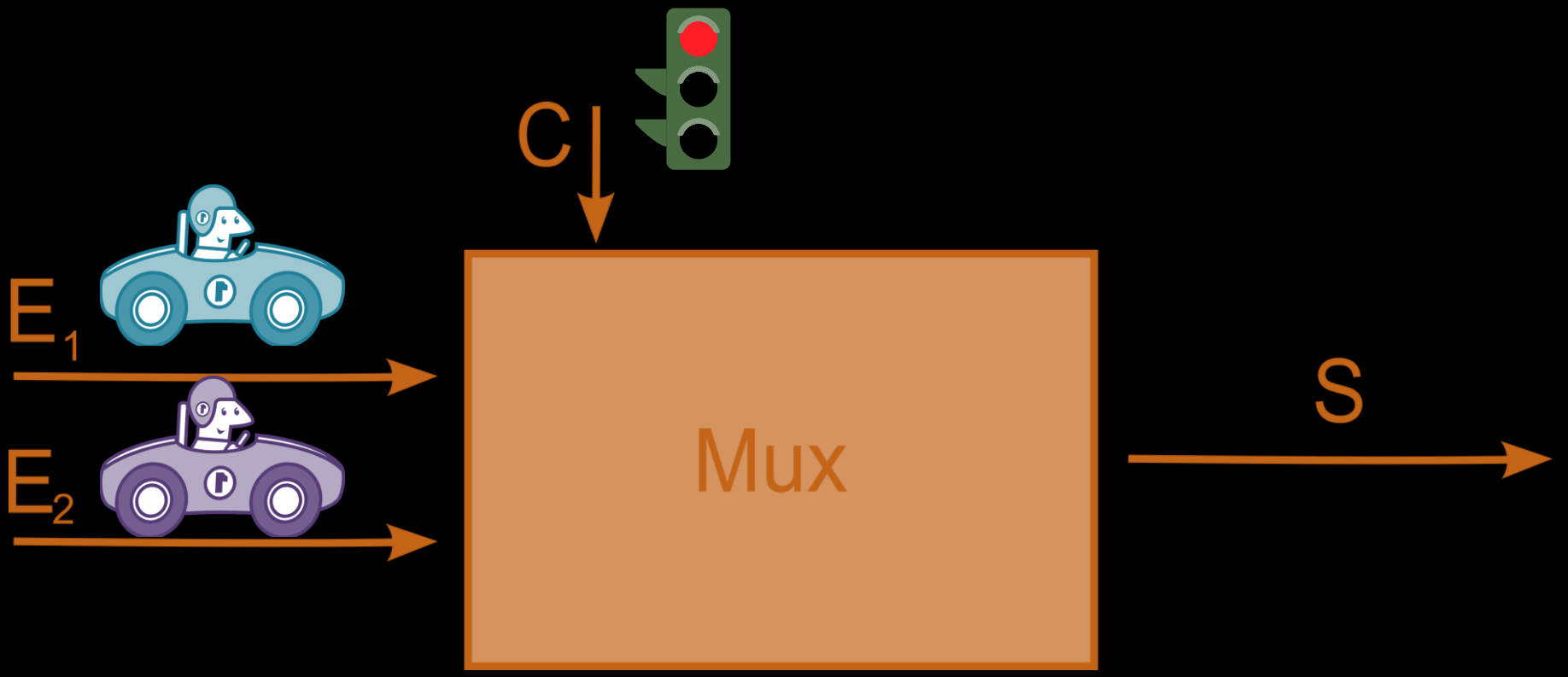
- 2 entradas
- 1 salida
- Una línea de control que elige cuál de las entradas se proyecta a la salida.

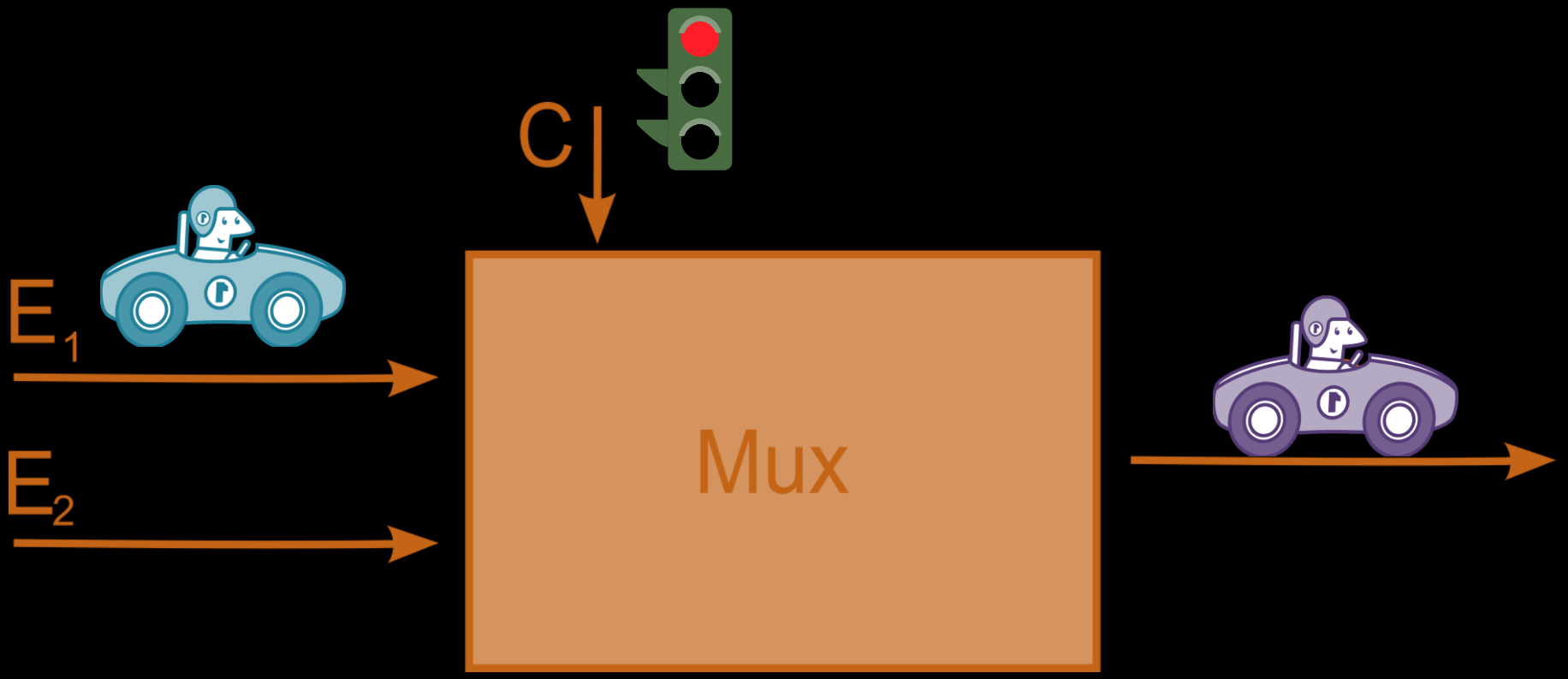



C
E1
E2
Mux
S


C
E1
E2
Mux
S


C
E1
E2
Mux


C
E1
E2
Mux
S


C
E1
E2
Mux

Multiplexor simple

- Tabla abreviada:

C	S
0	E1
1	E2

Multiplexor Simple

- Tabla completa:

C	E1	E2	S
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

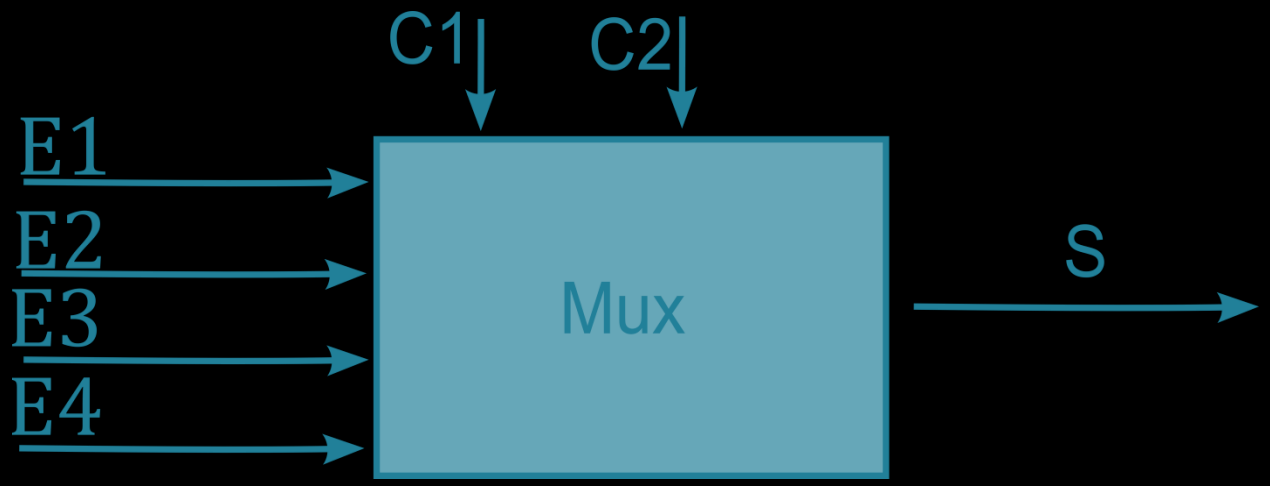
Multiplexor Simple

- Tabla completa:

C	E1	E2	S
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 de 4 entradas

- 4 entradas
- 1 salida
- 2 líneas de control



Multiplexor de 4 entradas

- Tabla abreviada:

C1	C2	S
0	0	E1
0	1	E2
1	0	E3
1	1	E4

Multiplexor de 4 entradas

- Tabla complete y circuito

Multiplexor de 4 entradas

- Tabla complete y circuito

¡¡¡TAREA!!!

Decodificador

- 2 entradas
- 4 salidas

Decodificador

- 2 entradas
- 4 salidas

Decodificador

- 2 entradas
- 4 salidas

Decodificador

- 2 entradas
- 4 salidas

Decodificador

- 2 entradas
- 4 salidas

00 -> 0

Decodificador

- 2 entradas
- 4 salidas

00 -> 0

Decodificador

- 2 entradas
- 4 salidas

00 -> 0
01 -> 1

Decodificador

- 2 entradas
- 4 salidas

00 -> 0
01 -> 1

Decodificador

- 2 entradas
- 4 salidas

00 -> 0
01 -> 1
10 -> 2

Decodificador

- Tabla:

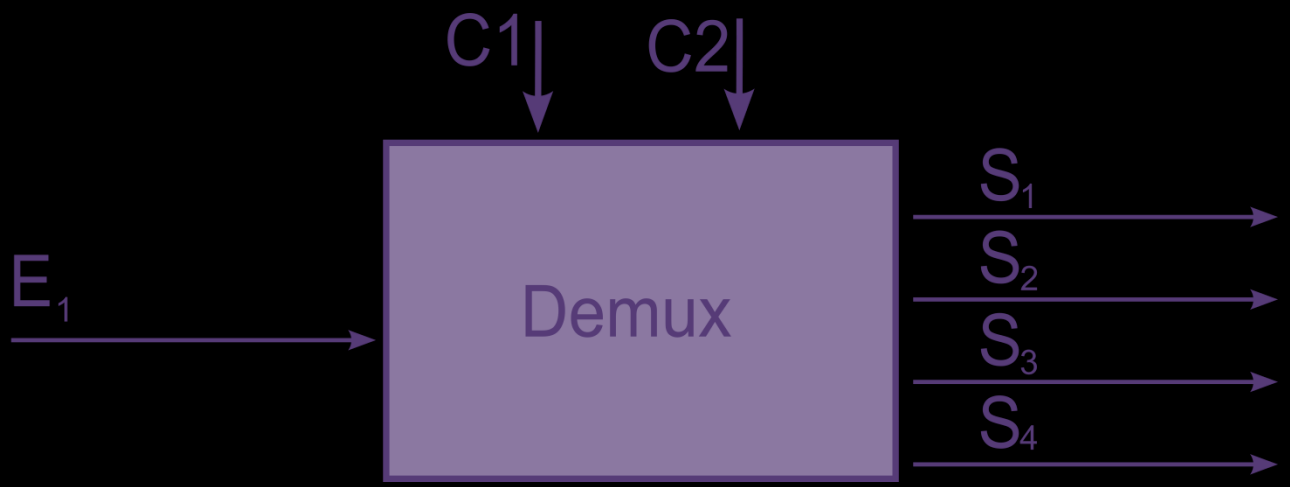
E1	E2	S1	S2	S3	S4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Demultiplexor

- 1 Entrada
- 2 Entradas de control
- 4 salidas

Demultiplexor

- 1 Entrada
- 2 Entradas de control
- 4 salidas



Demultiplexor

- Tabla:

E	C1	C2	S1	S2	S3	S4
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Circuitos aritméticos

- Implementan funciones aritméticas, como la suma

Half adder

- Suma dos bits
- 2 Entradas:
  - Los bits a sumar
- 2 Salidas:
  - La suma
  - El carry

Half adder

- Tabla:

X1	X2	S	C
0	0		
0	1		
1	0		
1	1		

Half adder

- Tabla:

X1	X2	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Full adder

- Suma dos bits
- 3 entradas
  - Los dos bits a sumar
  - El carry “anterior”
- 2 Salidas:
  - La suma
  - El carry de salida

Full adder

- Tabla:

X1	X2	Ci	S	Co
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	1		

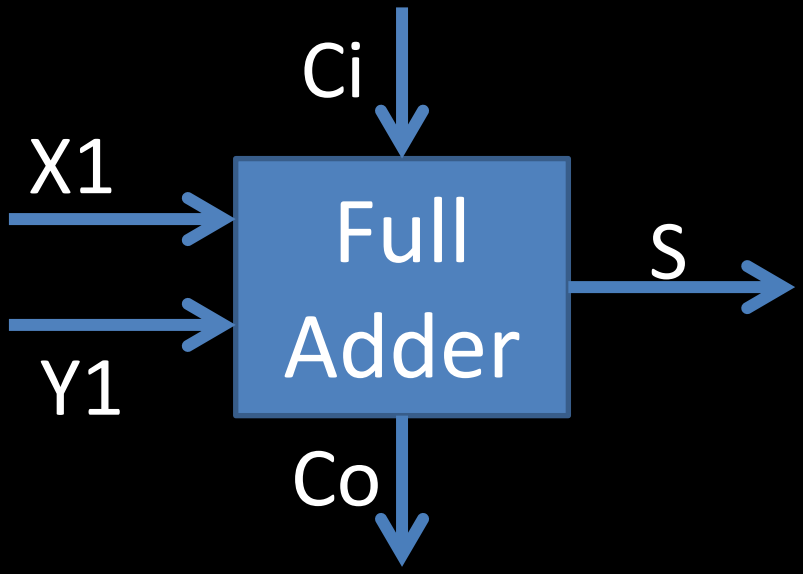
Full adder

- Tabla:

X1	X2	Ci	S	Co
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	1	1	1

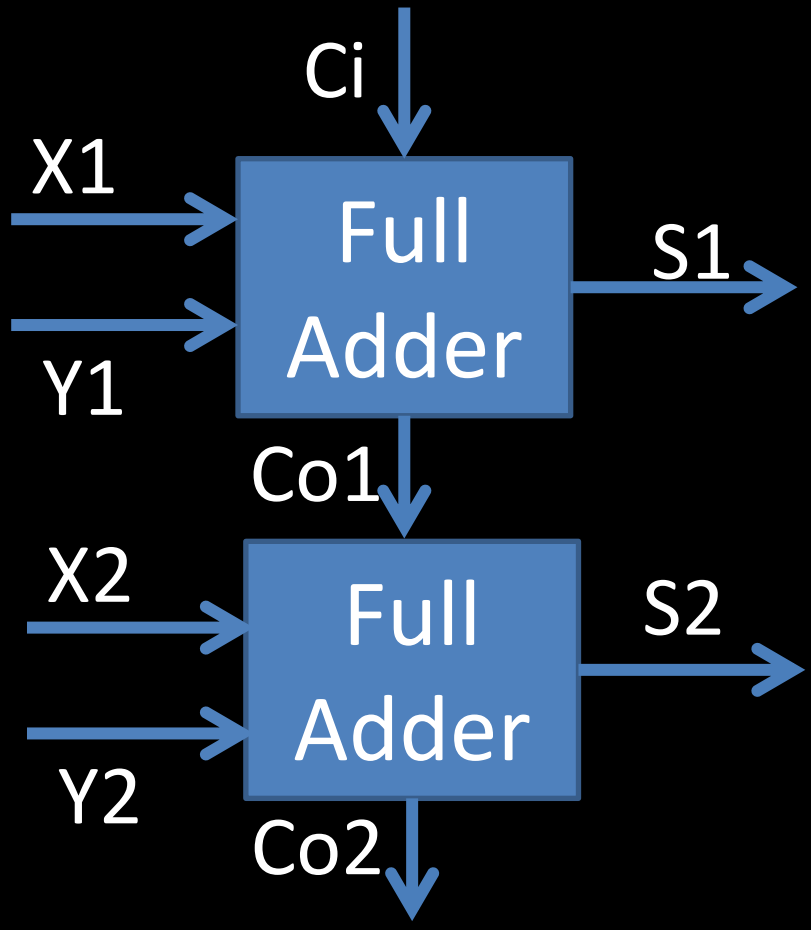
Sumar varios bits

- Una vez que ya tenemos armado el full adder de un bit, ¿Cómo puedo sumar varios bits?



Sumar varios bits

- Una vez que ya tenemos armado el full adder de un bit, ¿Cómo puedo sumar varios bits?



Restador

- Resta dos bits
- 2 Entradas:
  - Los bits a restar
- 2 Salidas:
  - La resta
  - El borrow

Restador

- Tabla:

X1	X2	R	B
0	0		
0	1		
1	0		
1	1		

Restador

- Tabla:

X1	X2	R	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

¿Qué pasó hoy?

- Compuertas lógicas:

¿Qué pasó hoy?

- Compuertas lógicas:
  - ¿Qué?

¿Qué pasó hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR

¿Qué pasó hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR
  - Compuerta AND

¿Qué pasó hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR
  - Compuerta AND
  - Compuerta NOT

¿Qué pasó hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR
  - Compuerta AND
  - Compuerta NOT
  - Otras compuertas

¿Qué pasó hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR
  - Compuerta AND
  - Compuerta NOT
  - Otras compuertas
- Circuitos

¿Qué pasó hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR
  - Compuerta AND
  - Compuerta NOT
  - Otras compuertas
- Circuitos
  - Formulas y tablas de verdad

¿Qué pasó hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR
  - Compuerta AND
  - Compuerta NOT
  - Otras compuertas
- Circuitos
  - Formulas y tablas de verdad
  - Producto de sumas y suma de productos

¿Qué pasó hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR
  - Compuerta AND
  - Compuerta NOT
  - Otras compuertas
- Circuitos
  - Formulas y tablas de verdad
  - Producto de sumas y suma de productos
  - Circuitos comunes

¿Qué pasó hoy?

- Compuertas lógicas:
  - ¿Qué?
  - Compuerta OR
  - Compuerta AND
  - Compuerta NOT
  - Otras compuertas
- Circuitos
  - Formulas y tablas de verdad
  - Producto de sumas y suma de productos
  - Circuitos comunes
  - Circuitos aritméticos