

Organización de Computadoras

SEMANA 1

UNIVERSIDAD NACIONAL DE QUILMES

¿Qué vimos?

- Sistema Binario
- Interpretación
- Representación
- Aritmética
- Sistema Hexadecimal

Hoy!

- Lógica proposicional
- Compuertas lógicas:
 - ¿Qué?
 - Compuertas básicas
- Circuitos
 - Formulas y tablas de verdad
 - Producto de sumas y suma de productos
 - Circuitos comunes
 - Circuitos aritméticos

Ver apunte: [Apunte sobre logica proposicional y digital](#)

Lógica Proposicional

Componentes de la lógica proposicional



- **Variables proposicionales:** Enunciados que pueden ser verdaderos o falsos
- **Operadores:** Conjunción, disyunción, negación, etc.

Lógica Proposicional

- Permite expresar situaciones **formalmente**
 - **A** El tanque está lleno
 - **B** La llave de paso está cerrada

$$A \rightarrow B$$

Lógica Proposicional

- La operación de la computadora está basada en el almacenamiento y procesamiento de datos binarios



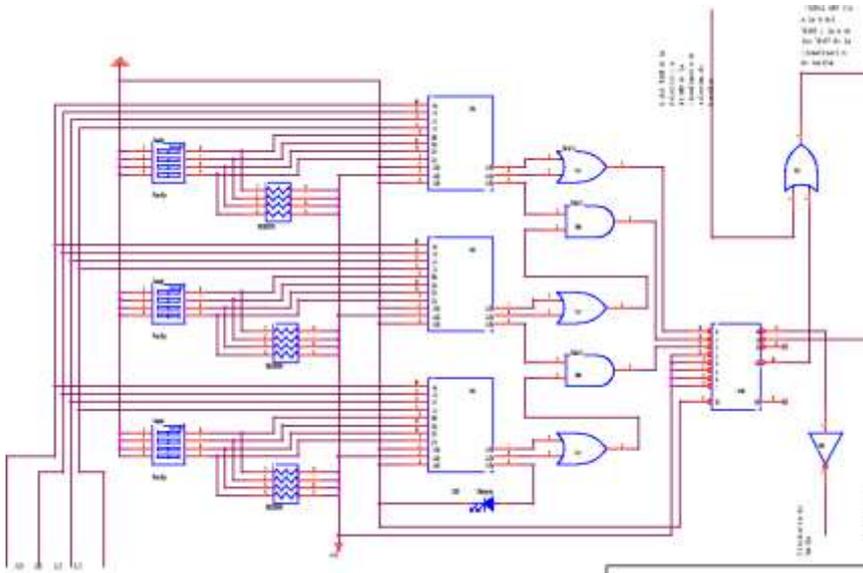
- Se utilizan circuitos



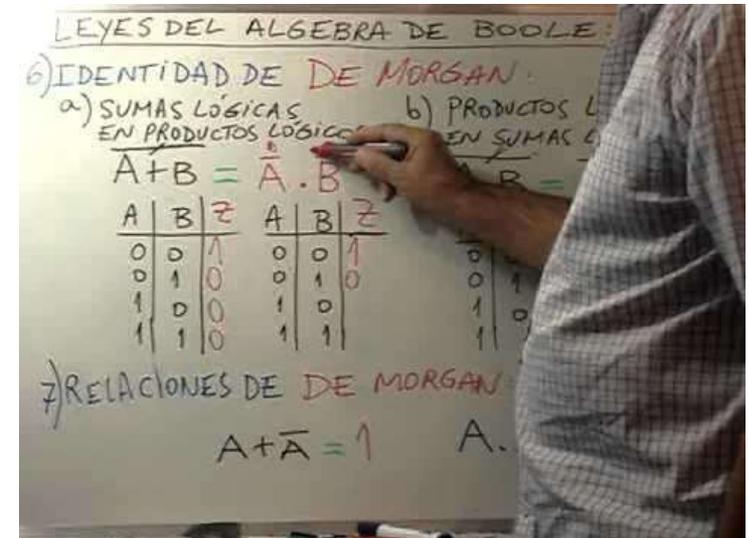
- Los circuitos se construyen mediante lógica digital

Lógica Proposicional

Lógica digital



Álgebra de Boole



- El álgebra de Boole (basada en la lógica proposicional) permite diseñar y analizar el comportamiento de los **circuitos digitales**.

Compuertas lógicas

- Es un dispositivo que implementa una función booleana simple.
- Traduce un conjunto de entradas (una o mas) en **una sola salida**
- Son la implementación en “Los Fierros” de las operaciones que hacemos con la máquina

Compuertas lógicas

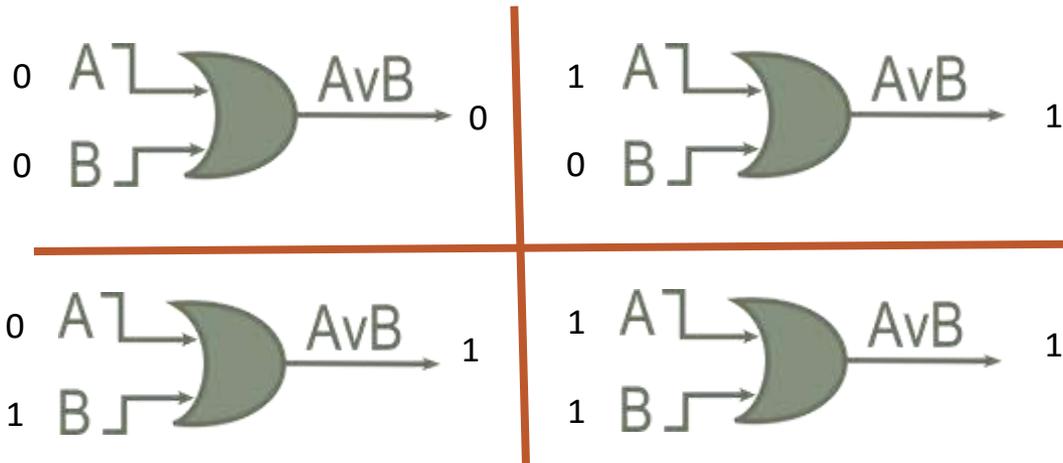
Compuerta OR



A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Compuertas lógicas

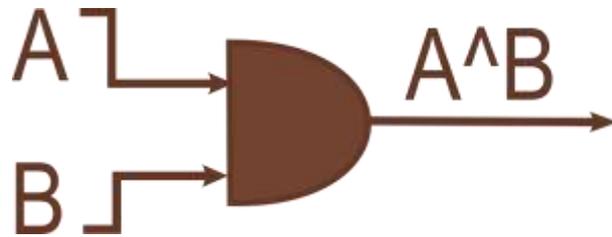
Compuerta OR



A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Compuertas lógicas

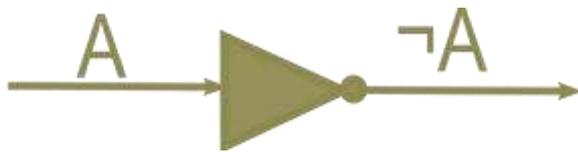
Compuerta AND



A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Compuertas lógicas

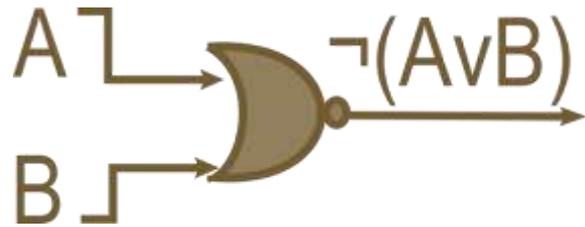
Compuerta NOT



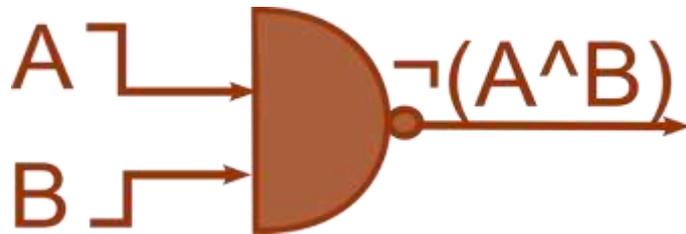
A	$\neg A$
0	1
1	0

Compuertas lógicas

Otras



Compuerta NOR



Compuerta NAND



Compuerta XOR

Circuitos

¿Qué son los circuitos?

- **Circuito lógico:**

- Traduce un conjunto de entradas en un conjunto de salidas de acuerdo a una o mas funciones lógicas
- Cada salida es estrictamente una función de las entradas
- Las salidas se actualizan de inmediato luego de que cambien las entradas
- Se obtienen combinando compuertas

Circuitos

¿Qué son los circuitos?

- Se pueden construir a partir de una fórmula booleana o a partir de una tabla de verdad
- Ejemplo: Construir un circuito que compute cada una de las siguientes funciones:
 - $B \wedge (C \vee A)$
 - $(A \wedge B) \vee (\neg A \wedge C)$
- **¿Cómo pasar de la tabla al circuito?**

Circuitos

¿Cómo pasar de la tabla al circuito?

Lenguaje natural



Tabla de verdad



Fórmula booleana

SOP / POS

Circuitos

Suma de Productos (SOP)

- Una fórmula tiene la forma de suma de productos si tiene la siguiente forma:
 - $A_1 \vee A_2 \vee A_3 \vee \dots \vee A_n$, donde cada A_i usa solo **and** y **not**
 - Ej: $(A \wedge \neg B) \vee (\neg A \wedge C) \vee (B \wedge C)$

Circuitos

Productos de Sumas (POS)

- Una fórmula tiene la forma de producto de sumas si tiene la siguiente forma:
 - $A_1 \wedge A_2 \wedge A_3 \wedge \dots \wedge A_n$, donde cada A_i usa solo **or** y **not**
 - Ej: $(A \vee \neg B) \wedge (\neg A \vee C) \wedge (B \vee C)$

Circuitos

¿Cómo pasar de la tabla al circuito?

1. Armamos la tabla de verdad
2. Si hay menos 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 menos 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

Circuitos

¿Cómo pasar de la tabla al circuito?

Ejemplo:

E_1	E_2	E_3	S	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$(\bar{E}_1 \wedge E_2 \wedge E_3)$
1	0	0	0	
1	0	1	1	$(E_1 \wedge \bar{E}_2 \wedge E_3)$
1	1	0	1	$(E_1 \wedge E_2 \wedge \bar{E}_3)$
1	1	1	1	$(E_1 \wedge E_2 \wedge E_3)$

$$s = (\bar{E}_1 \wedge E_2 \wedge E_3) \vee (E_1 \wedge \bar{E}_2 \wedge E_3) \vee (E_1 \wedge E_2 \wedge \bar{E}_3) \vee (E_1 \wedge E_2 \wedge E_3)$$

Circuitos

¿Cómo pasar de la tabla al circuito?

Ejemplos:

1)

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

2)

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

Circuitos

¿Cómo pasar de la tabla al circuito?

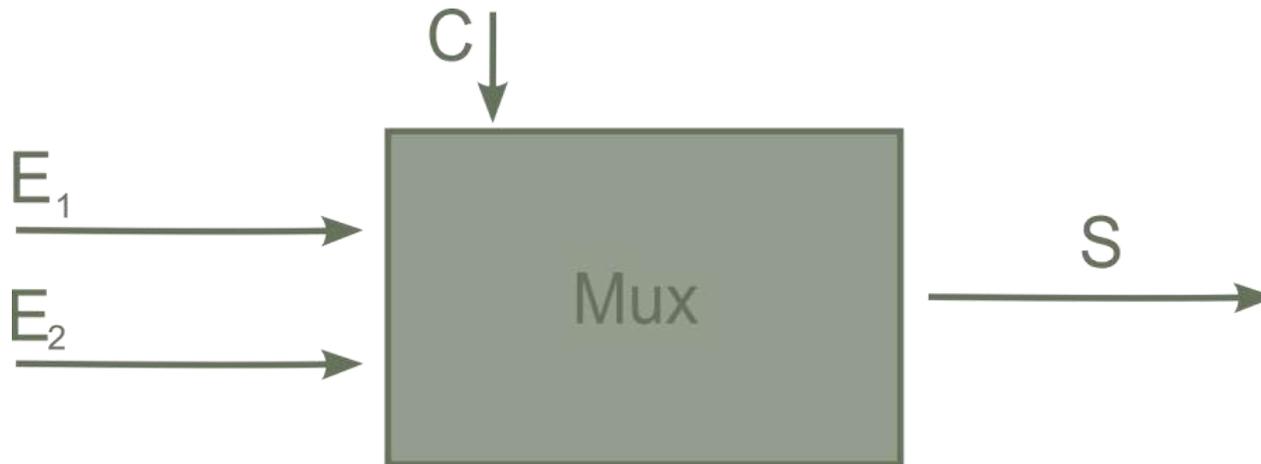
Ejercicio:

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

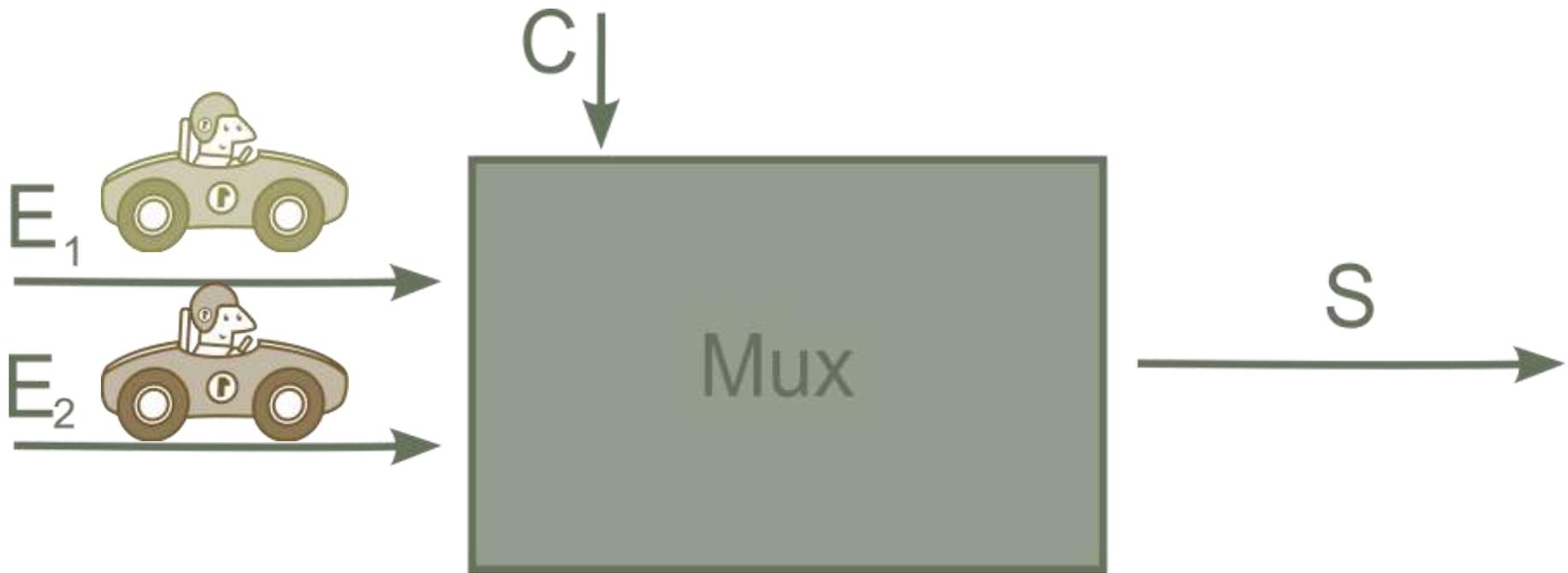
Multiplexor simple

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



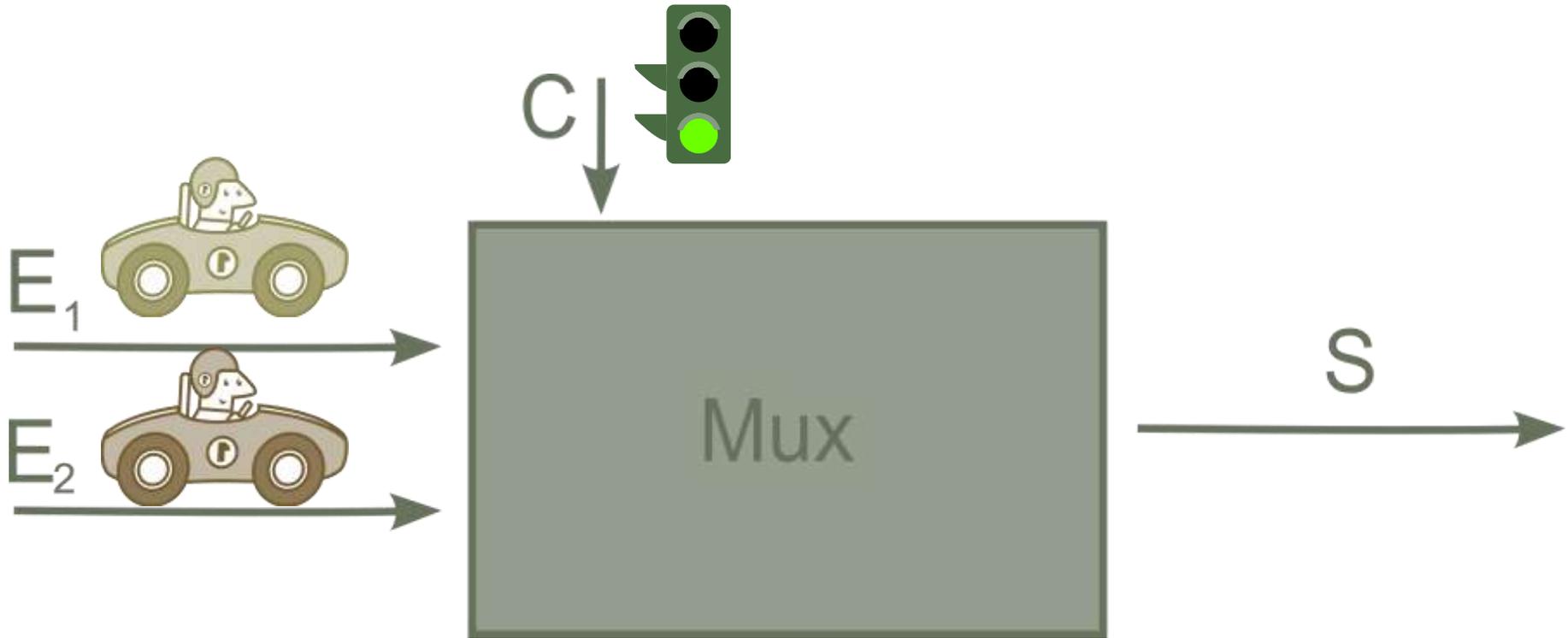
Circuitos útiles

Multiplexor simple



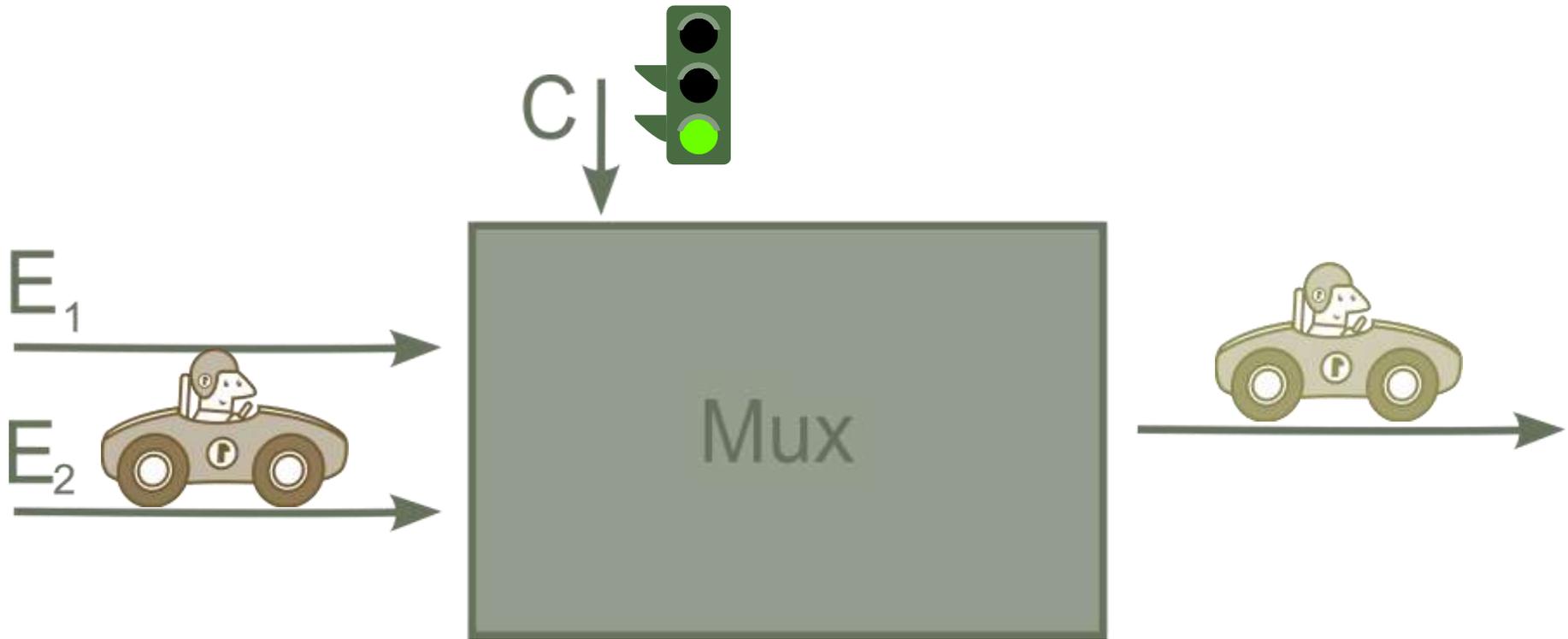
Circuitos útiles

Multiplexor simple



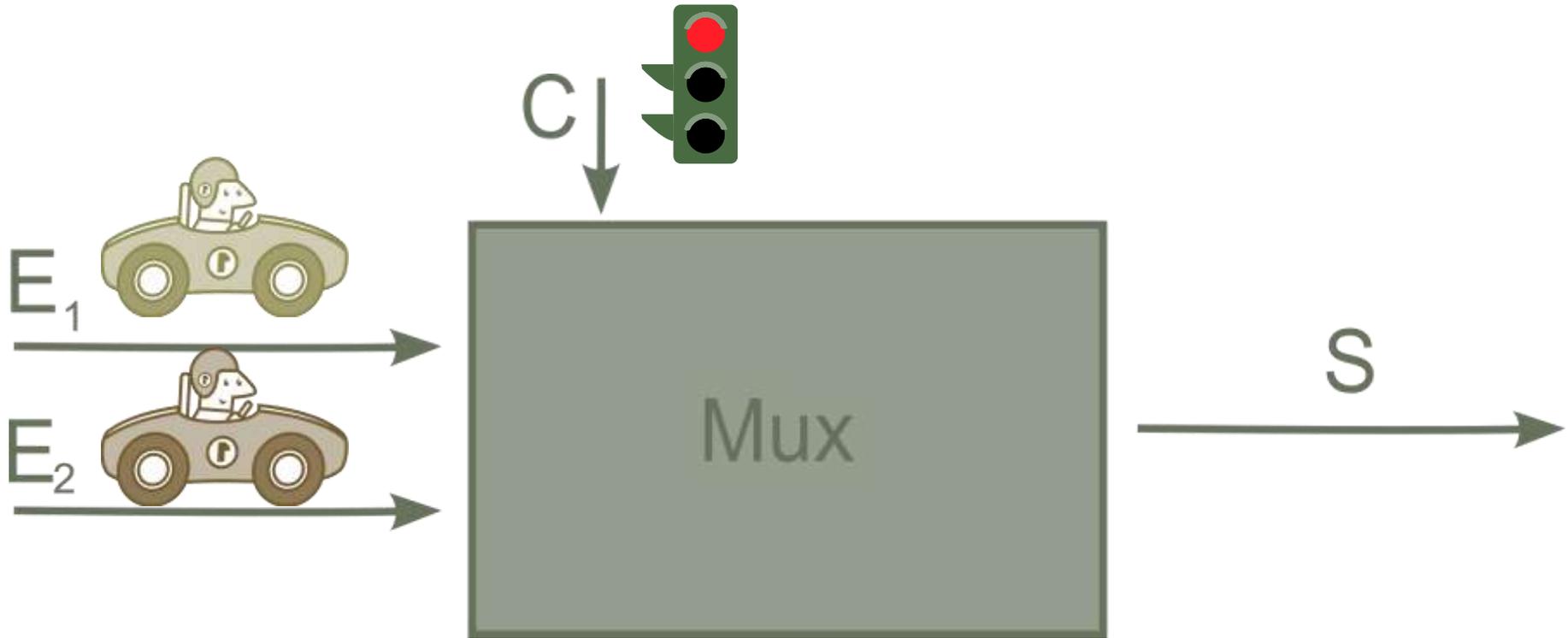
Circuitos útiles

Multiplexor simple



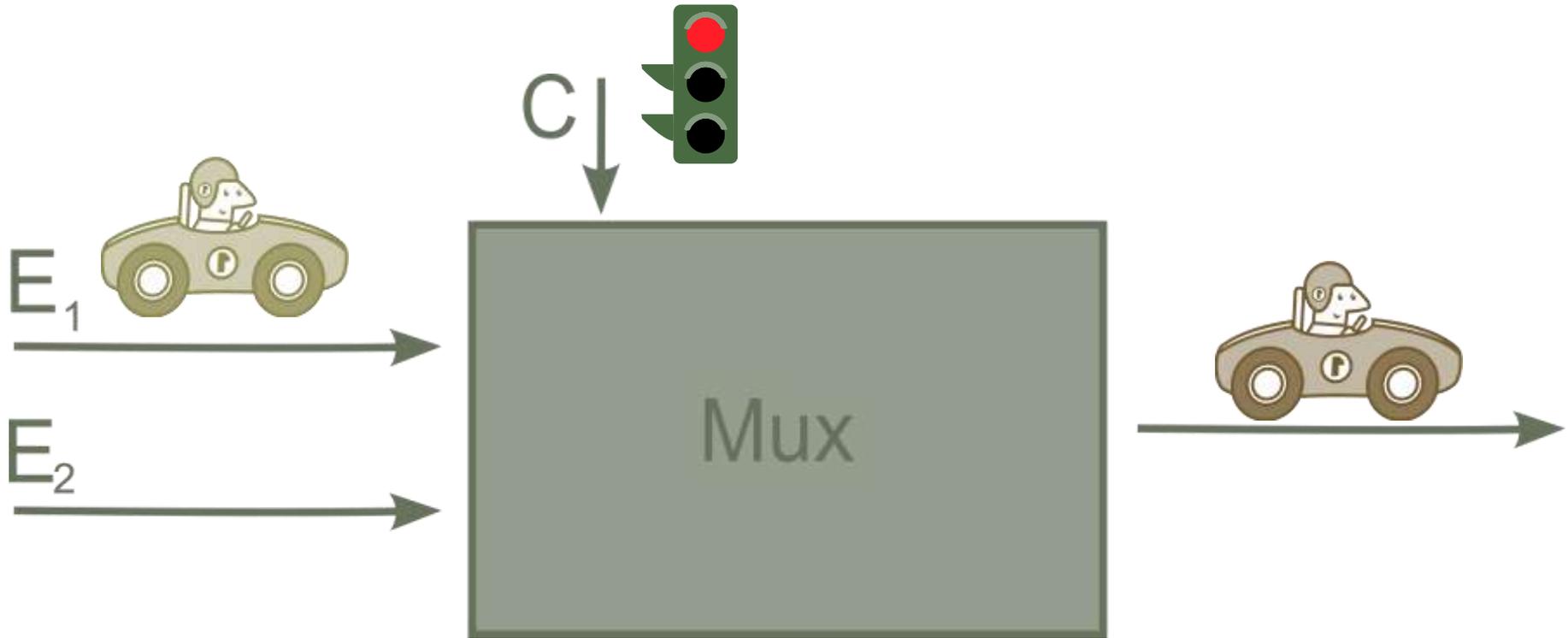
Circuitos útiles

Multiplexor simple



Circuitos útiles

Multiplexor simple



Circuitos útiles

Multiplexor simple

- Tabla abreviada

C	S
0	E1
1	E2

Circuitos útiles

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	

Circuitos útiles

Multiplexor simple

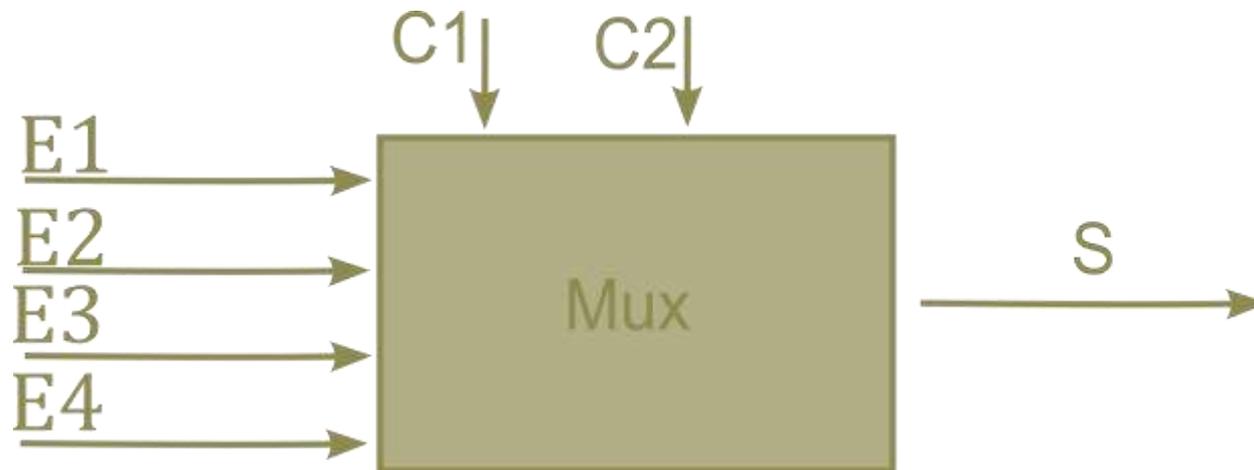
- Tabla completa

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

Circuitos útiles

Multiplexor de 4 entradas

- 4 entradas
- 1 salida
- 2 línea de control



Circuitos útiles

Multiplexor de 4 entradas

- Tabla abreviada

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

- Tabla completa y circuito → TAREA!!

Circuitos útiles

Decodificador

- 2 entradas
- 4 salidas



00 -> 0

Circuitos útiles

Decodificador

- 2 entradas
- 4 salidas



00 -> 0 01 -> 1

Circuitos útiles

Decodificador

- 2 entradas
- 4 salidas



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

Circuitos útiles

Decodificador

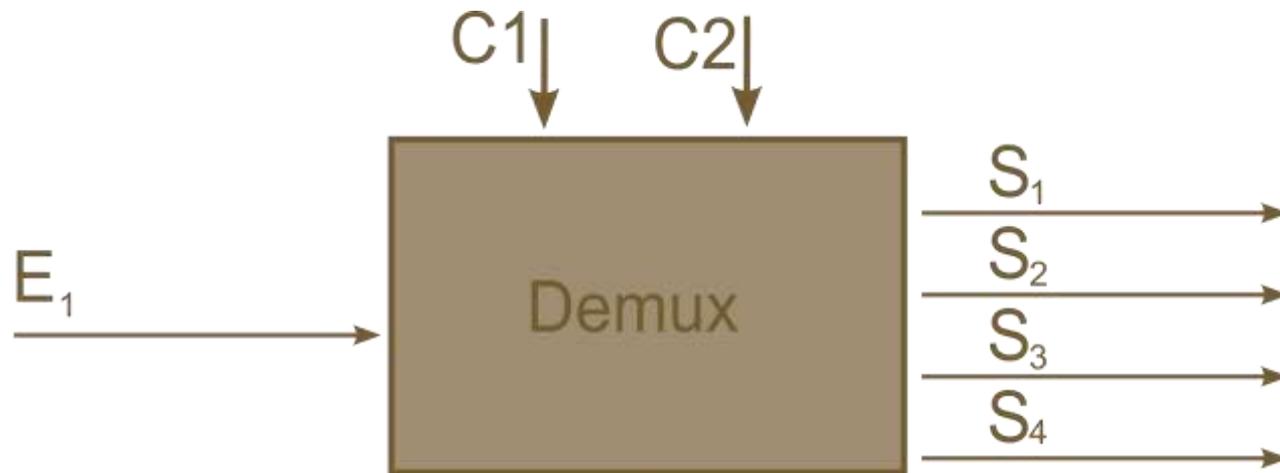
- Tabla

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

Circuitos útiles

Demultiplexor

- 1 Entrada
- 2 Entradas de control
- 4 salidas



Circuitos útiles

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

Circuitos Aritméticos

Half adder

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

Circuitos Aritméticos

Half adder

- Tabla

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

Circuitos Aritméticos

Half adder

- Tabla

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

Circuitos Aritméticos

Full adder

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

Circuitos Aritméticos

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	0		
1	1	1		

Circuitos Aritméticos

Full adder

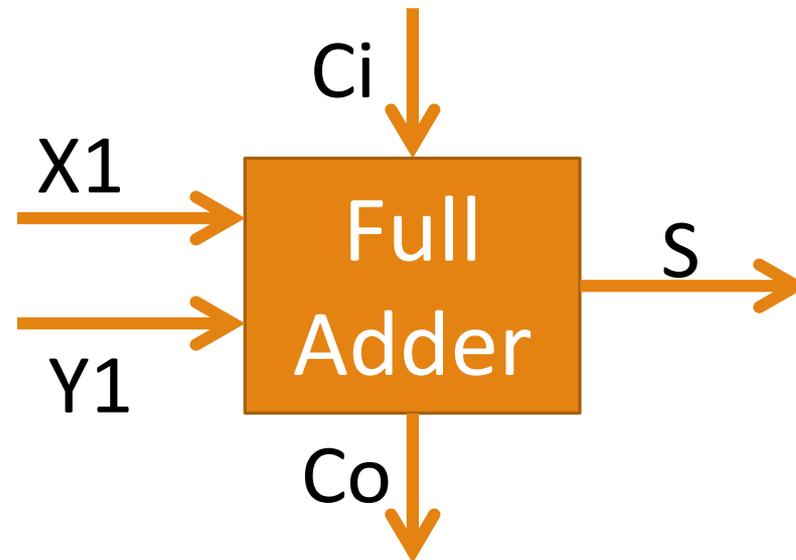
- Tabla

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	0	0	1
1	1	1	1	1

Circuitos Aritméticos

Full adder

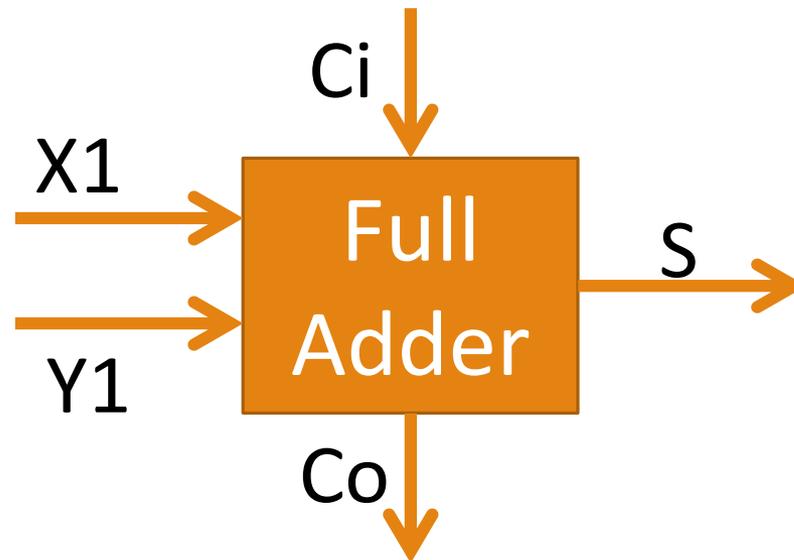
- Circuito



Circuitos Aritméticos

Full adder

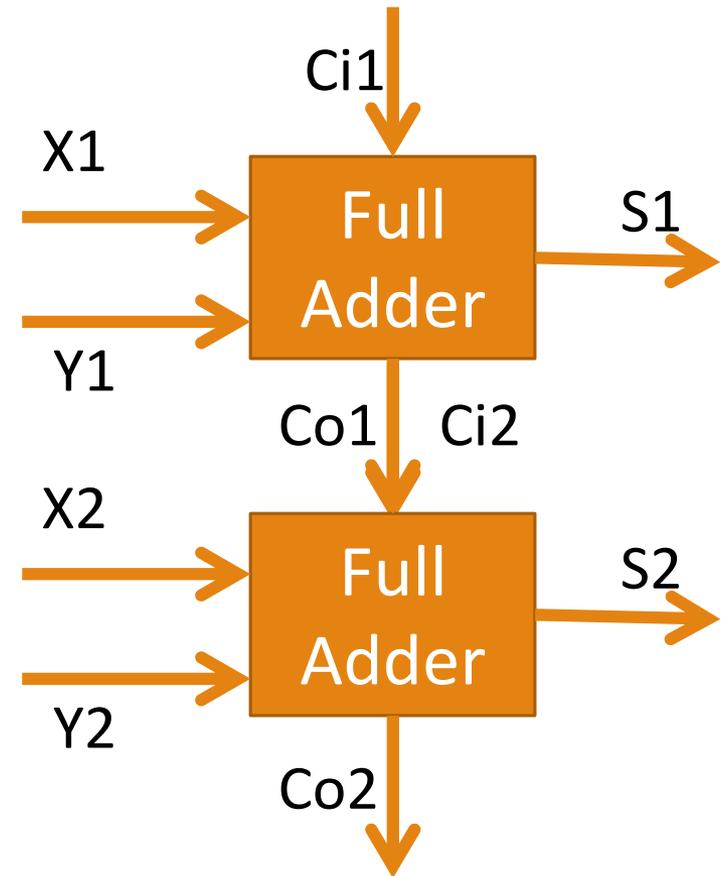
- Una vez que ya tenemos armado el full adder de un bit, ¿Cómo puedo sumar varios bits?



Circuitos Aritméticos

Full adder

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



Circuitos Aritméticos

Restador

- Resta dos bits
- 2 Entradas:
 - Los bits a restar
- 2 Salidas:
 - La resta
 - El borrow (Le pedí uno al compañero)

Circuitos Aritméticos

Restador

- Tabla

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

Circuitos Aritméticos

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?
 - Compuertas básicas
- Circuitos
 - Formulas y tablas de verdad
 - Producto de sumas y suma de productos
 - Circuitos comunes
 - Circuitos aritméticos

Quiero saber más!!!!

- Organización y Arquitectura de computadoras, Stallings, Apéndice A: Lógica digital (Notar que el libro muestra mas circuitos que los vistos en clase y llega a un nivel de detalle mayor)
- Apunte en el blog!