



CLASE 5

LÓGICA COMBINACIONAL

CONTENIDO

Sumador

Restador

Multiplicador

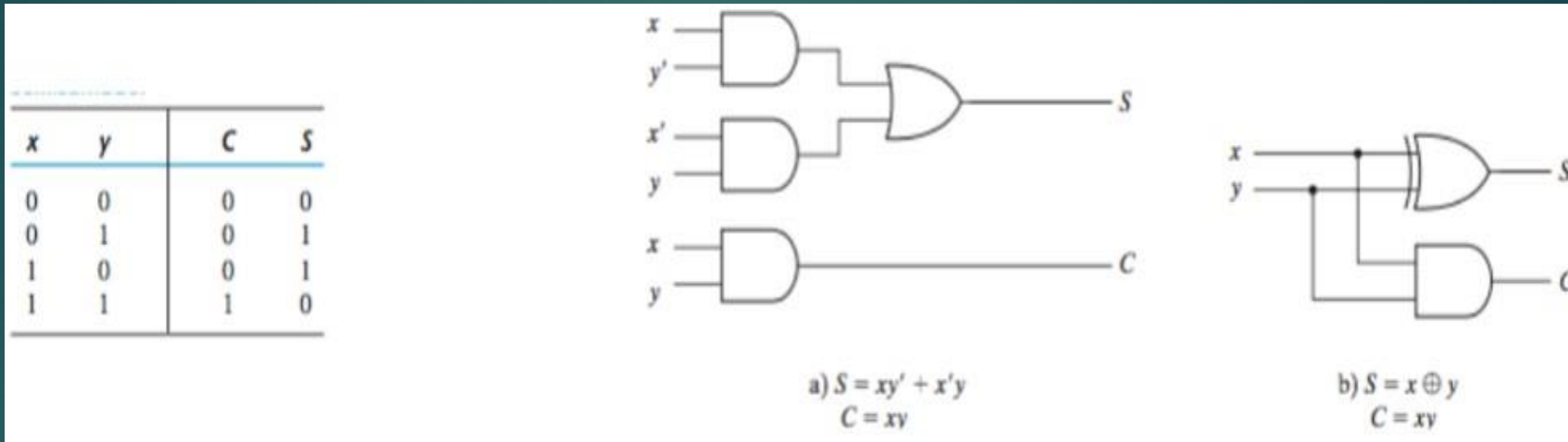
SUMADOR

- ❖ Recibe dos números binarios y nos devuelve la suma en binario.
- ❖ Semisumador:
 - Únicamente obtiene el resultado aritmético con su respectivo acarreo de salida (Carry-out o Cout)

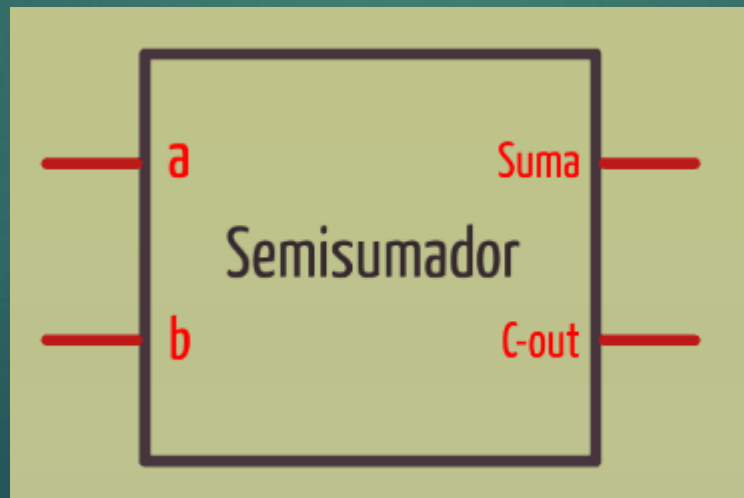
a	b	acarreo	suma
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Si vemos bien, el acarreo se comporta como una compuerta AND, y la suma como una compuerta XOR

- Semisumador de 1 bit:



Forma
Final:



❖ Sumador Completo:

- Un sumador completo es uno capaz de sumar tres bits: los dos de los sumandos; y un tercero, el acarreo de los sumandos anteriores.
- A la entrada adicional se le llama acarreo de entrada, Carry-In (Cin).

a	b	Cin	Cout	Suma
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Karnaugh

C-out

C-in \ ab	00	01	11	10
0	0	0	1	0
1	0	1	1	1

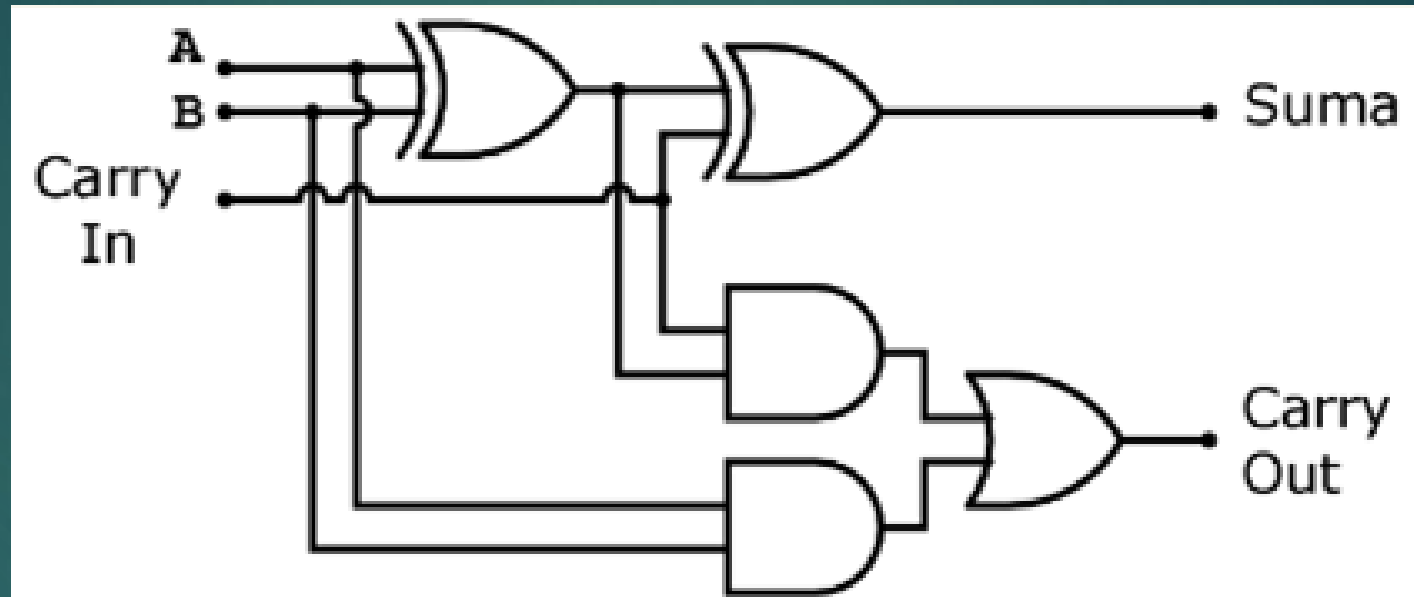
→ $(a.b) + (C\text{-in}.(a \oplus b))$

Suma

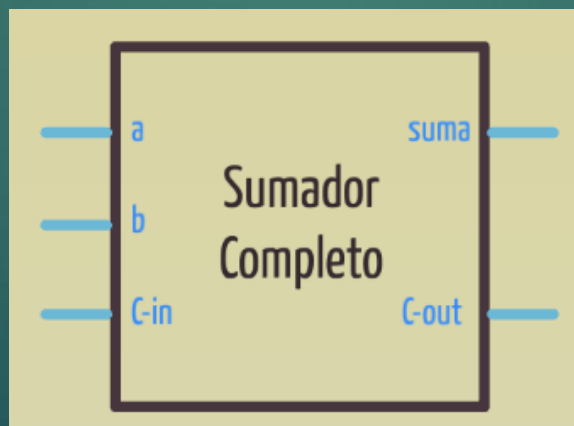
C-in \ ab	00	01	11	10
0	0	1	0	1
1	1	0	1	0

→ $(a \oplus b) \oplus C\text{in}$

Circuito Resultante:

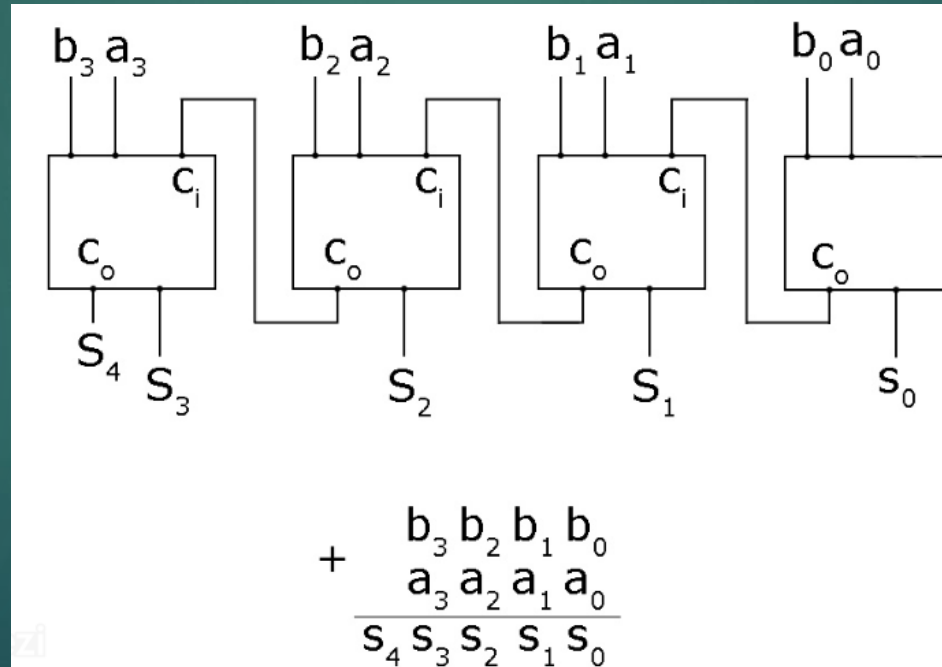


Forma
Final:



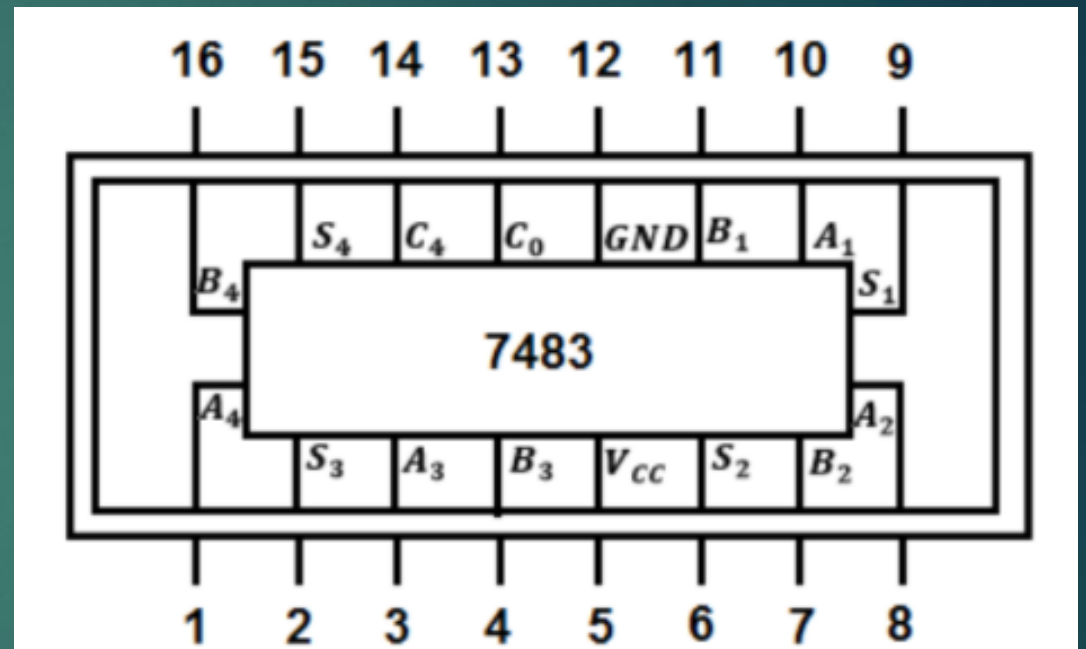
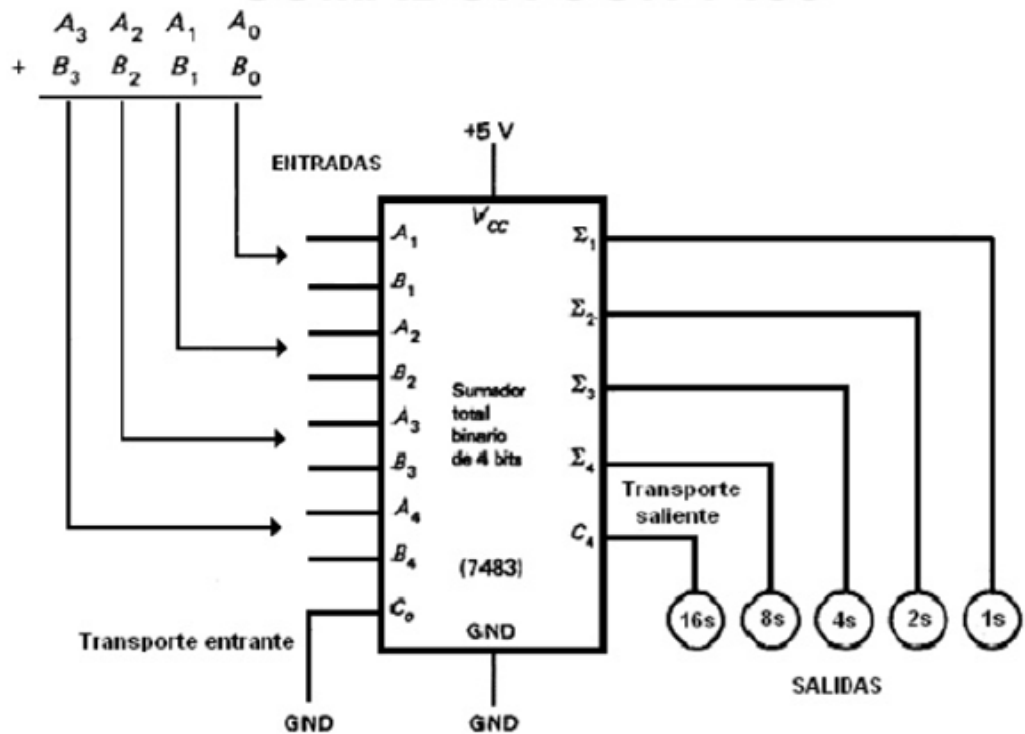
SUMADORES EN CASCADA.

- ❖ Conectamos dos o mas sumadores en cascada para poder realizar la suma de 2 números binarios, en donde cada sumador por si solo es incapaz de sumarlos, debido a la mayor cantidad de bits de los sumandos.



SUMADOR COMPLETO DE 4 BITS

SUMADOR CON 7483



CIRCUITO DIGITAL BINARIO A BCD



RESTADOR

- ▶ La operación de resta con circuitos integrados puede ser bastante completa, debido a esto, la resta se realiza por medio de un método alternativo, que se basa en realizar la resta utilizando un sumador donde uno de los sumandos se hace negativo, esto se puede observar en el sistema decimal de la siguiente forma:

- ▶ $25 - 7 = 25 + (-7) = 18$


RESTADOR

- ❖ En los circuitos digitales se utilizan los *complementos*, que son el equivalente a tener números negativos. Ya que esta es la forma más sencilla de realizar la resta de números binarios.
- ❖ Existen dos tipos de complementos:
 - ❖ Complemento a la base (complemento a 2) (* porque se utiliza la base binaria)
 - ❖ Complemento a la base disminuida (complemento a 1)(* 2-1)

RESTADOR

- Complemento a la base en el Sistema decimal:
 - Se quiere realizar la resta $45 - 15$.
 - Se convierte el sustraendo(15) a su complemento:
 - Complemento a 10 de 2 dígitos : $10^2 - 15 = 85$
 - Ahora se puede sumar normalmente:
 - $45 + 85 = (1)30$
 - Se ignora el acarreo (1), ya que unicamente se estan utilizando 2 digitos.

RESTADOR

-  Complemento a la base en el Sistema binario:
 - Se quiere realizar la resta $1010 - 0101$ (10-5).
 - Se convierte el sustraendo(0101) a su complemento:
 - Complemento a 2 de 4 dígitos : $2^4 - 0101 = 10000 - 0101 = 1011$
 - Ahora se puede sumar normalmente:
 - $1010 + 1011 = (1)0101$
 - Se ignora el acarreo (1), ya que unicamente se estan utilizando 4 digitos.

RESTADOR

- ❖ La ventaja que nos da utilizar complementos para convertir una resta en una suma no tendría sentido si para obtener los complementos, debemos realizar una resta. Para ello existen 2 algoritmos:

ALGORITMO 1(Complemento a 2)

1. Se Copian los bits del LSB a MSB hasta encontrar un bit con valor de 1.
2. Se prosigue invirtiendo(negando) los bits restantes. (0->1 | 1->0)

EJEMPLO:

- Se desea obtener el complemento a 2 de 1011 0100 :
- 1011 0100
- **0100 1100**

ALGORITMO 2 (Complemento a 1)

1. Se obtiene el complemento a 1 del numero invirtiendo todos los bits.
2. Al resultado anterior se le suma 1.

EJEMPLO:

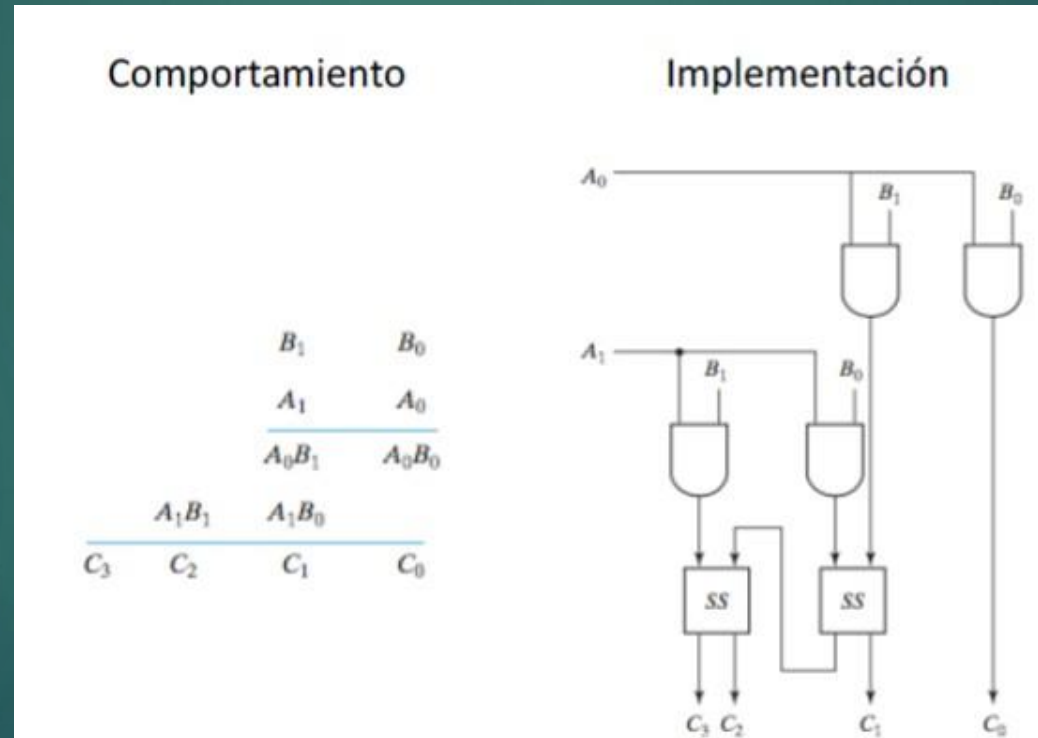
- Se desea obtener el complemento a 1 de 1011 0100 :
- 1011 0100
- **0100 1011 +1**
- **0100 1100**



CIRCUITO DIGITAL SUMADOR - RESTADOR

MULTIPLICADOR

- Un multiplicador es un circuito combinacional que forma la multiplicación aritmética de n bits.





Forma Genérica:

$$\begin{array}{cccc}
 & & A_3 & A_2 & A_1 & A_0 \\
 & & B_3 & B_2 & B_1 & B_0 \\
 \hline
 & & & A_3 B_0 & A_2 B_0 & A_1 B_0 & A_0 B_0 \\
 & & & & A_3 B_1 & A_2 B_1 & A_1 B_1 & A_0 B_1 \\
 & & & & & A_3 B_2 & A_2 B_2 & A_1 B_2 & A_0 B_2 \\
 & & & & & & A_3 B_3 & A_2 B_3 & A_1 B_3 & A_0 B_3 \\
 \hline
 P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
 \end{array}$$

Ejemplo

-
-

$$\begin{array}{r}
 1101 \\
 1011 \\
 \hline
 1101 \\
 1101 \\
 0000 \\
 1101 \\
 \hline
 10001111
 \end{array}$$



DUDAS?