

Práctica 07

Operaciones Aritméticas

ALUMNO: MEZA VARGAS BRANDON DAVID

GRUPO: 2CM5

BOLETA: 2020630288



Fundamentos de Diseño Digital

1) Objetivo general

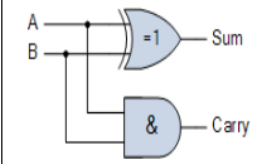
El alumno diseñará circuitos aritméticos programando en lenguaje VHDL y programando su GAL 22V10 para verificar el resultado.

2) Introducción Teórica

Medio Sumador

Un circuito básico se puede hacer de ANDS y Ex-OR lo que nos permite “añadir” juntos dos números binarios de un solo bit, A y B.

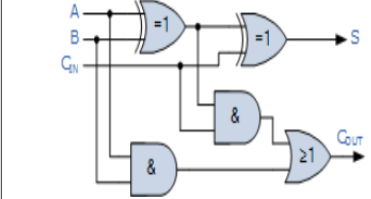
Medio Sumador

Symbol		Truth Table			
		B	A	SUM	CARRY
		0	0	0	0
		0	1	1	0
		1	0	1	0
		1	1	0	1

parece bastante completo, pero resulta que para sumar mas dígitos necesitamos algo que se llama:

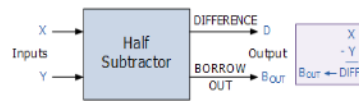
Sumador Completo

Este es más grande, pero te da la posibilidad de concatenarlos y así tener más dígitos.

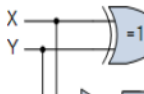
Symbol		Truth Table				
		C-in	B	A	Sum	C-out
		0	0	0	0	0
		0	0	1	1	0
		0	1	0	1	0
		0	1	1	0	1
		1	0	0	1	0
		1	0	1	0	1
		1	1	0	0	1
		1	1	1	1	1

Medio Restador

Como su nombre indica, un binario Subtractor es un circuito de toma de decisiones que resta de dos números binarios de la otra, por ejemplo, $X - Y$ para encontrar la diferencia resultante entre los dos números.

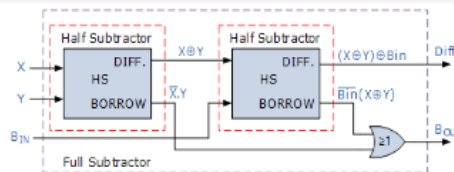


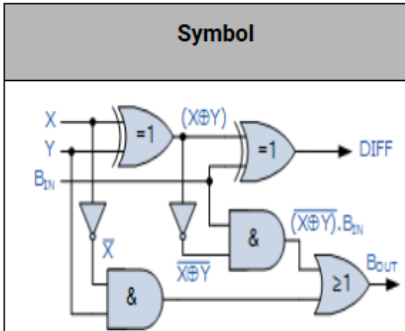
Medio Restador

Symbol	Truth Table			
	Y	X	Diff.	Borrow
	0	0	0	0
	0	1	1	0
	1	0	1	1
	1	1	0	0

Restador Completo.

Como el anterior este no se puede usar para más de 1bit.



Symbol	Truth Table				
	B-in	Y	X	Diff.	B-out
	0	0	0	0	0
	0	0	1	1	0
	0	1	0	1	1
	0	1	1	0	0
	1	0	0	1	1
	1	0	1	0	0
	1	1	0	0	1
	1	1	1	1	1

Sumadores de N bits

Para los sumadores de N bits un método natural de realizar la suma es situar sumadores completos en modo “ripple” o en serie, de forma que desde el bit menos significativo hacia el mas significativo el acarreo de salida del bit $j-1$.

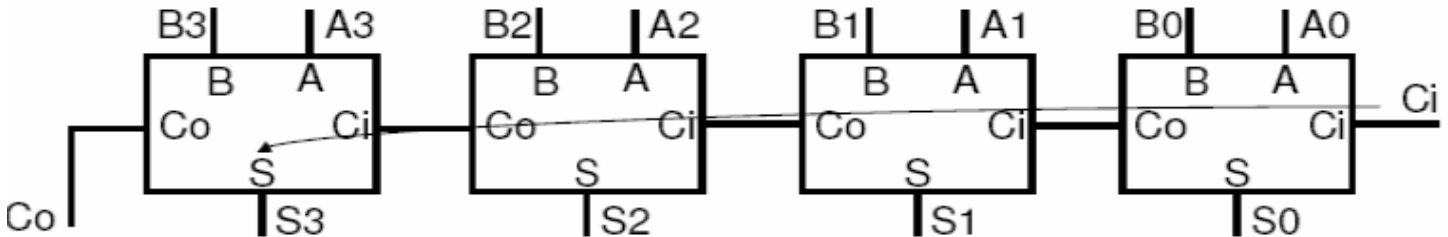
El primer bit se puede construir con un semisumador mientras que los demás bits requieren un sumador completo, en este caso se tiene un semisumador de N bits.

Fundamentos de Diseño Digital

Si en el primer bit se utiliza un sumador completo, el circuito dispone además de acarreo de entrada C_i y se tiene un sumador completo de N bits. Tener acarreo de entrada permite una mejor funcionalidad del circuito, como por ejemplo poner en serie dos (o más) sumadores de N bits para formar un sumador de $2N$ bits.

En cualquier caso, se tienen $N+1$ bits de salida: N de suma S y un acarreo de salida Co .

El tiempo de propagación de este sumador es proporcional al número N de sumadores en serie.



3) Materiales empleados

- ✓ 1 Circuito Integrado GAL22V10
- ✓ 15 LEDS de colores
- ✓ 15 Resistores de 330Ω
- ✓ 10 Resistores de $1K\Omega$
- ✓ 1 Dip switch de 8
- ✓ Alambre telefónico
- ✓ 1 Tablilla de Prueba (Protoboard)
- ✓ 1 Pinzas de punta
- ✓ 1 Pinzas de corte
- ✓ Cables Banana-Caimán (para alimentar el circuito)

4) Equipo empleado

- ✓ Multímetro
- ✓ Fuente de Alimentación de 5 Volts
- ✓ Manual de MOTOROLA, "FAST and LS TTL"
- ✓ Programador Universal

5) Desarrollo Experimental y Actividades

5.1.- Medio Sumador

- Arme su circuito y compruebe su tabla de verdad.

#	a	b	Suma	Carry
0	0	0	0	0
1	0	1	1	0
2	1	0	1	0
3	1	1	0	1

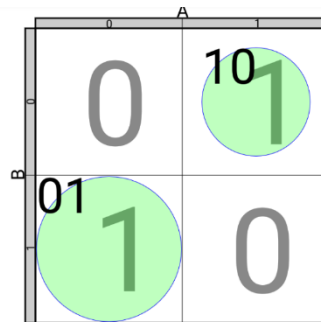
- Funciones Canónicas y mapas.

SUMA;

$$Suma(a, b) = \sum (1, 2)$$

$$Suma(a, b) = \bar{a}b + a\bar{b}$$

$$Suma(a, b) = a \text{ xor } b$$

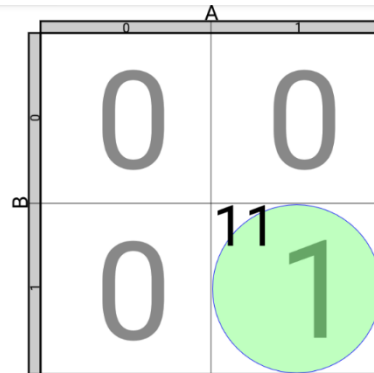


$$f = \bar{A} \cdot B + A \cdot \bar{B}$$

CARRY

$$Carry(a, b) = \sum (3)$$

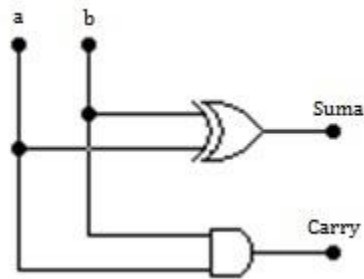
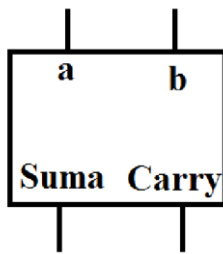
$$Carry(a, b) = ab$$



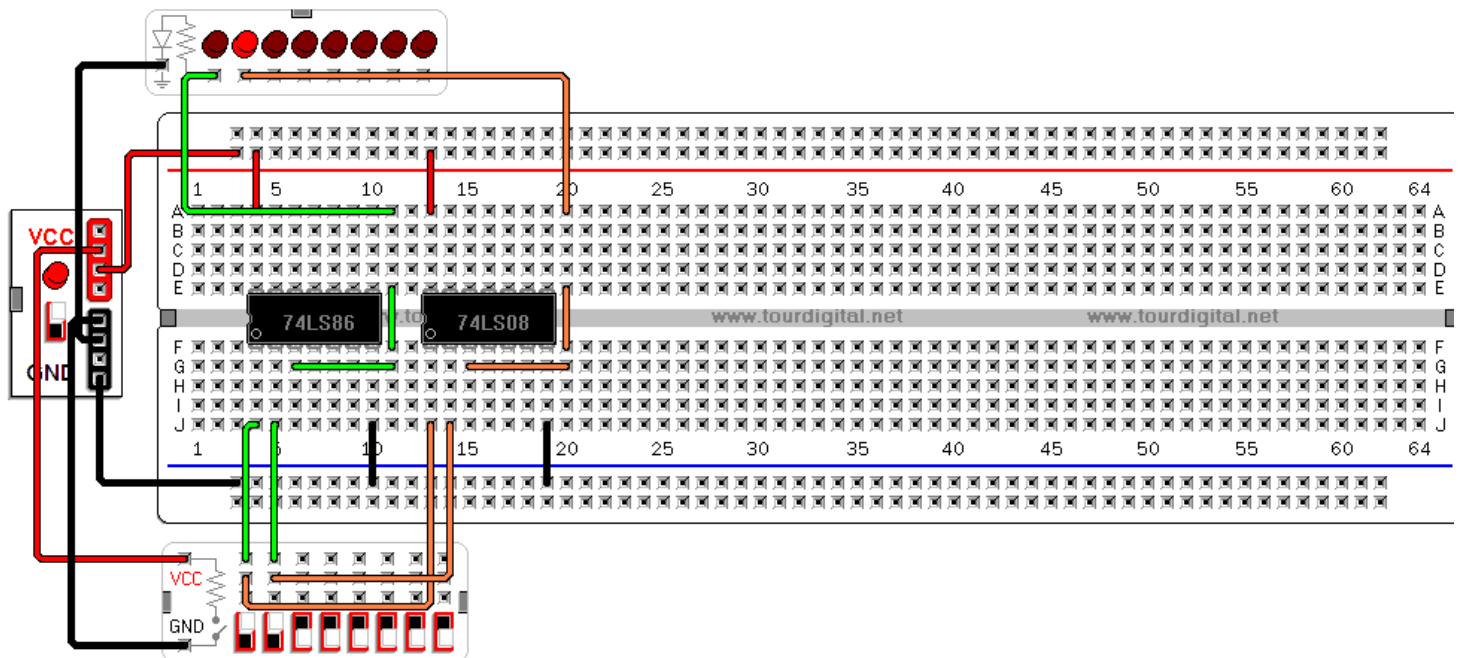
$$f = A \cdot B$$

Fundamentos de Diseño Digital

- Diseña y dibuja el siguiente circuito lógico para obtener sus ecuaciones lógicas.



- Armado con compuertas



Fundamentos de Diseño Digital

5.2.- Sumador Completo

- Arme su circuito y compruebe su tabla de verdad.

#	a	b	Cin	Suma	Cout
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

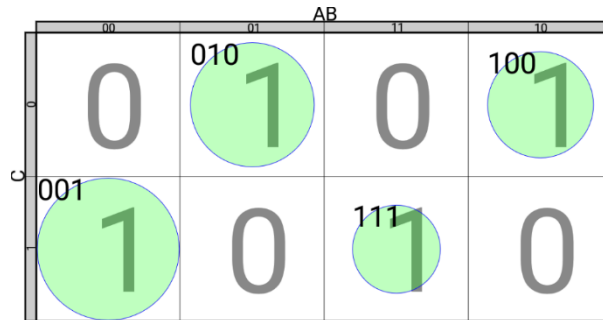
- Funciones Canónicas y mapas.

SUMA:

$$Suma(a, b, c) = \sum (1, 2, 4, 7)$$

$$Suma(a, b, c) = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc$$

$$Suma(a, b, c) = a \text{ xor } b \text{ xor } c$$



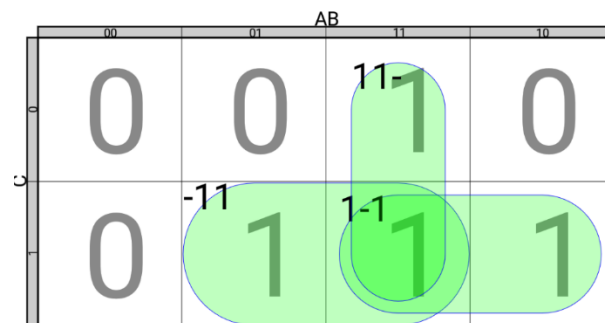
$$f = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

COUT:

$$Cout(a, b) = \sum (3, 5, 7)$$

$$Cout(a, b) = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc$$

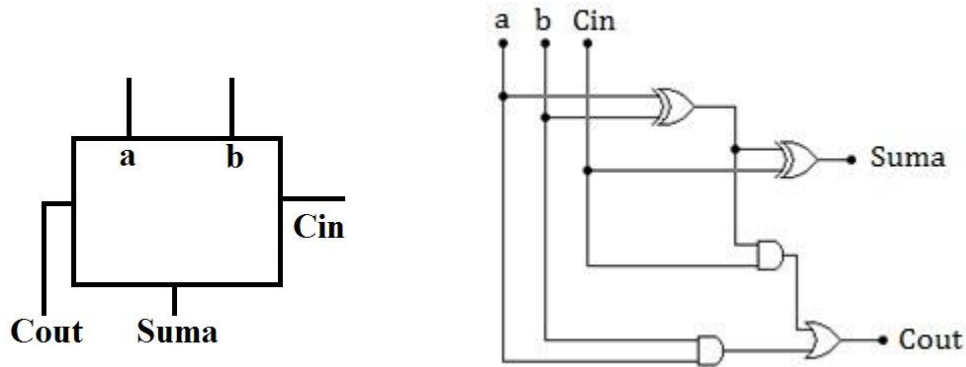
$$Cout(a, b) = bc + ac + ab$$



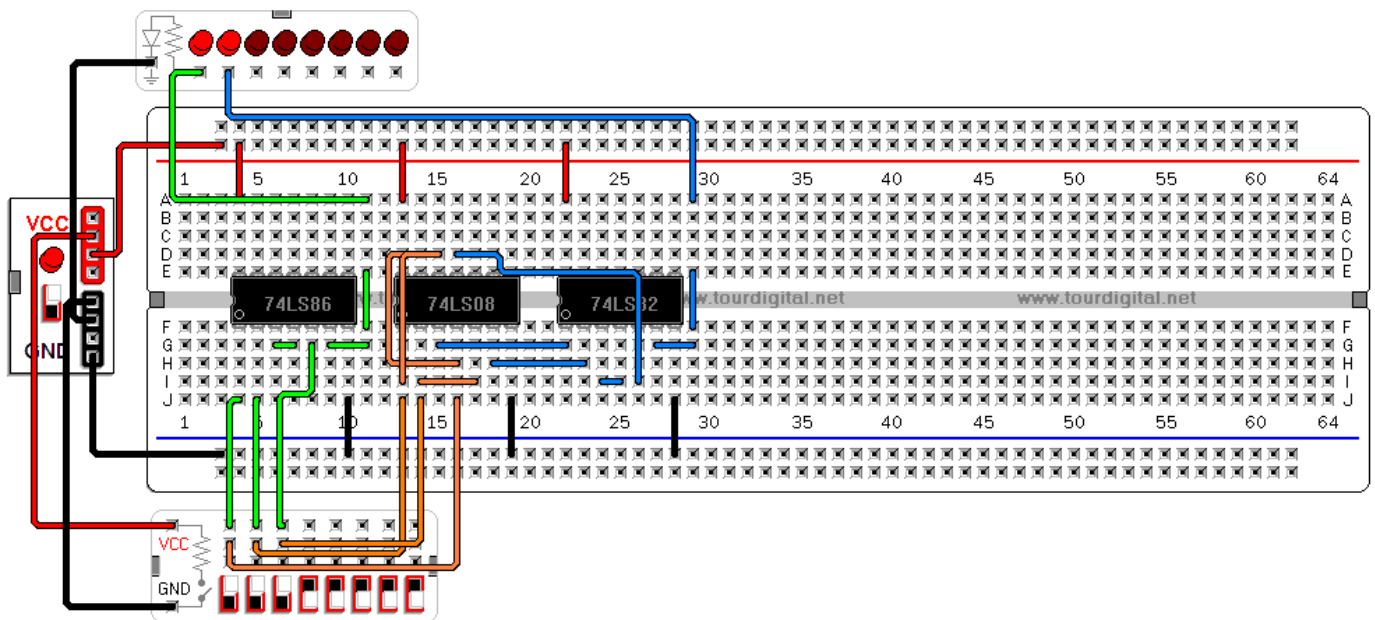
$$f = B\bar{C} + A\bar{C} + A\bar{B}$$

Fundamentos de Diseño Digital

- Diseña y dibuja el siguiente circuito lógico para obtener sus ecuaciones lógicas.



- Armado con compuertas



5.3.- Medio Restador

- Arme su circuito y compruebe su tabla de verdad.

#	x	y	Resta	Préstamo
0	0	0	0	0
1	0	1	1	1
2	1	0	1	0
3	1	1	0	0

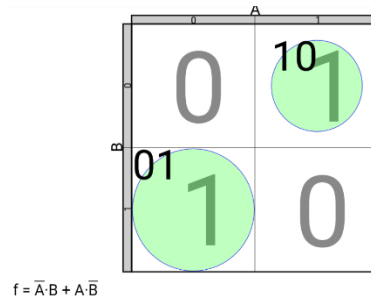
- Funciones Canónicas y mapas.

RESTA

$$Resta(x, y) = \sum (1, 2)$$

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

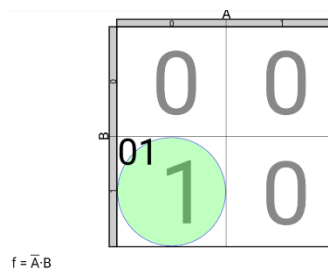
$$Resta(x, y) = x \text{ xor } y$$



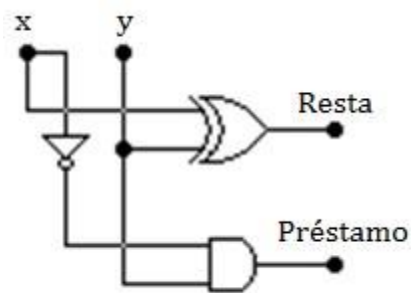
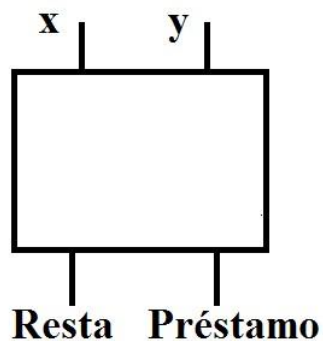
PRÉSTAMO

$$Préstamo(x, y) = \sum (1)$$

$$Préstamos(x, y) = \bar{x}y$$

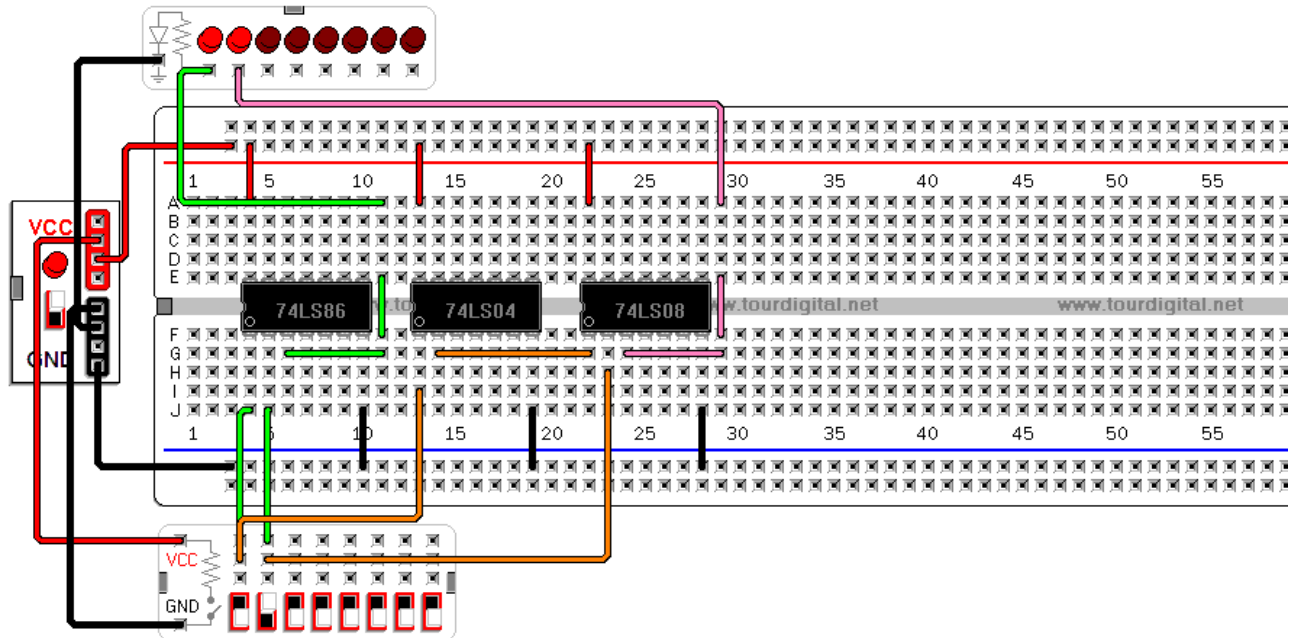


- Diseñe y dibuje el siguiente circuito lógico para obtener sus ecuaciones lógicas.



Fundamentos de Diseño Digital

- Armado con compuertas



5.4.- Restador Completo

- Arme su circuito y compruebe su tabla de verdad.

#	x	y	Pin	Resta	Pout
0	0	0	0	0	0
1	0	0	1	1	1
2	0	1	0	1	1
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	0
6	1	1	0	0	0
7	1	1	1	1	1

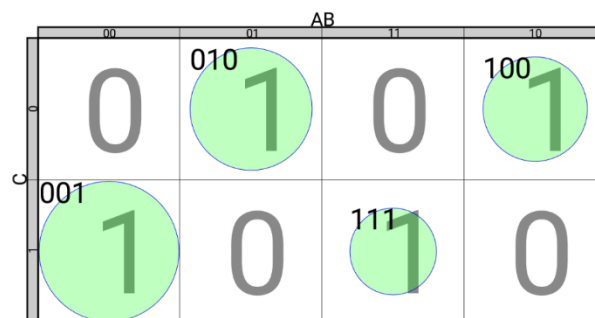
- Funciones Canónicas y mapas.

RESTA

$$\text{Resta}(x, y, z) = \sum (1, 2, 4, 7)$$

$$\text{Resta}(x, y, z) = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$$

$$\text{Resta}(x, y, z) = x \text{ xor } y \text{ xor } z$$



$$f = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C$$

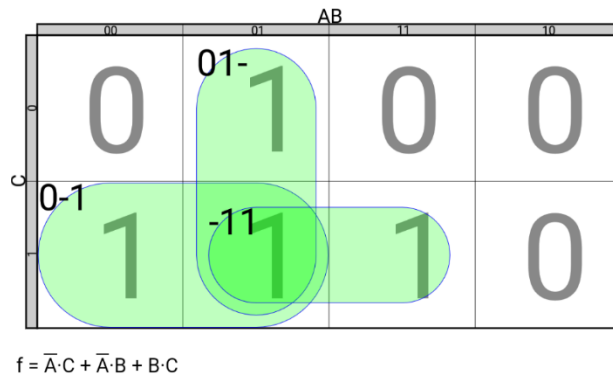
Fundamentos de Diseño Digital

PRÉSTAMO

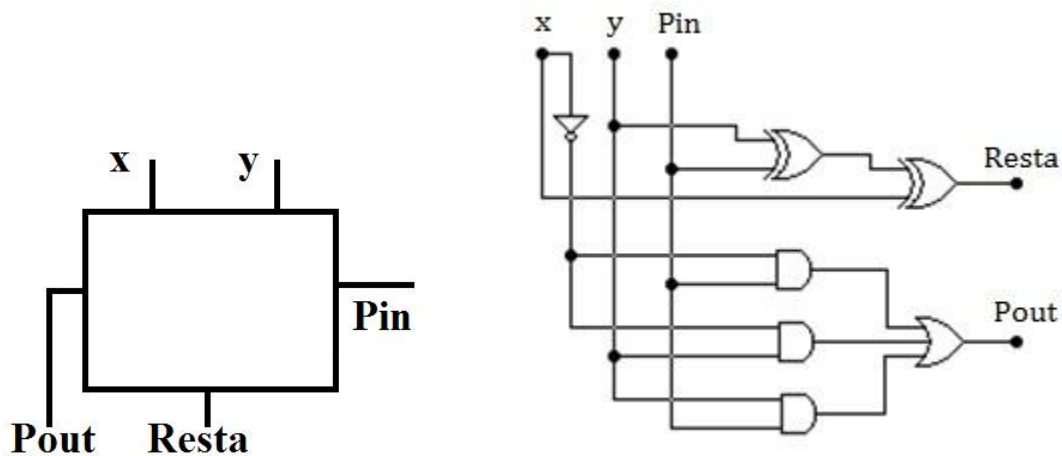
$$\text{Préstamo}(x, y, z) = \sum (1, 2, 3, 7)$$

$$\text{Préstamos}(x, y, z) = \bar{x}yz + \bar{x}y\bar{z} + \bar{x}yz + xyz$$

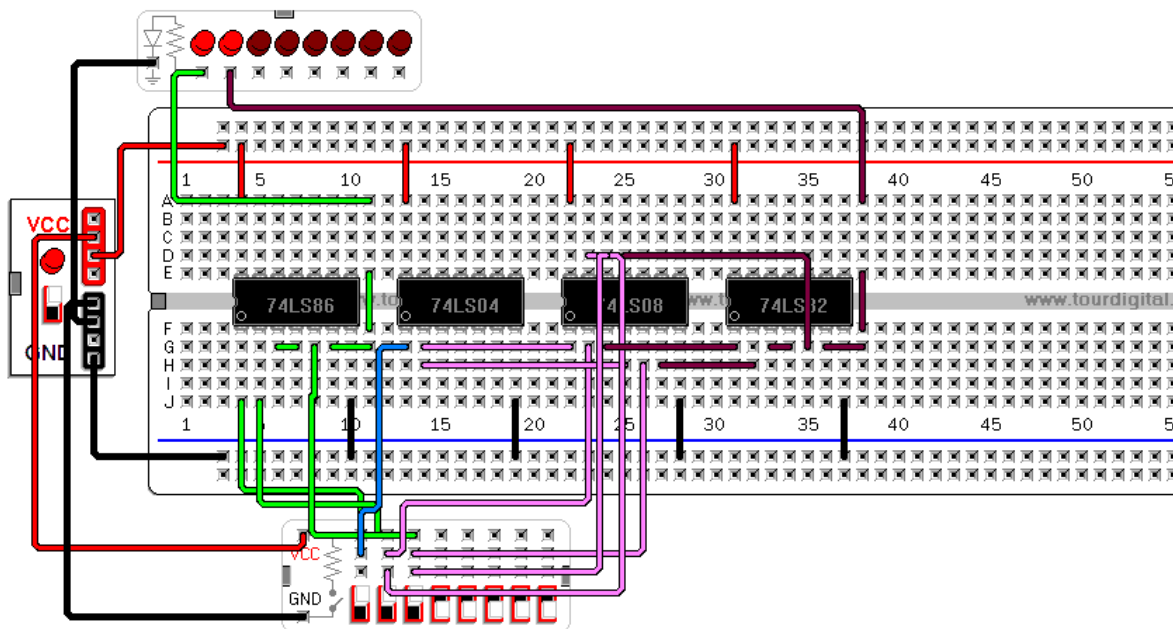
$$\text{Préstamos}(x, y, z) = \bar{x}z + \bar{x}y + yz$$



- Diseñe y dibuje el siguiente circuito lógico para obtener sus ecuaciones lógicas.



- Armado con compuertas



Fundamentos de Diseño Digital

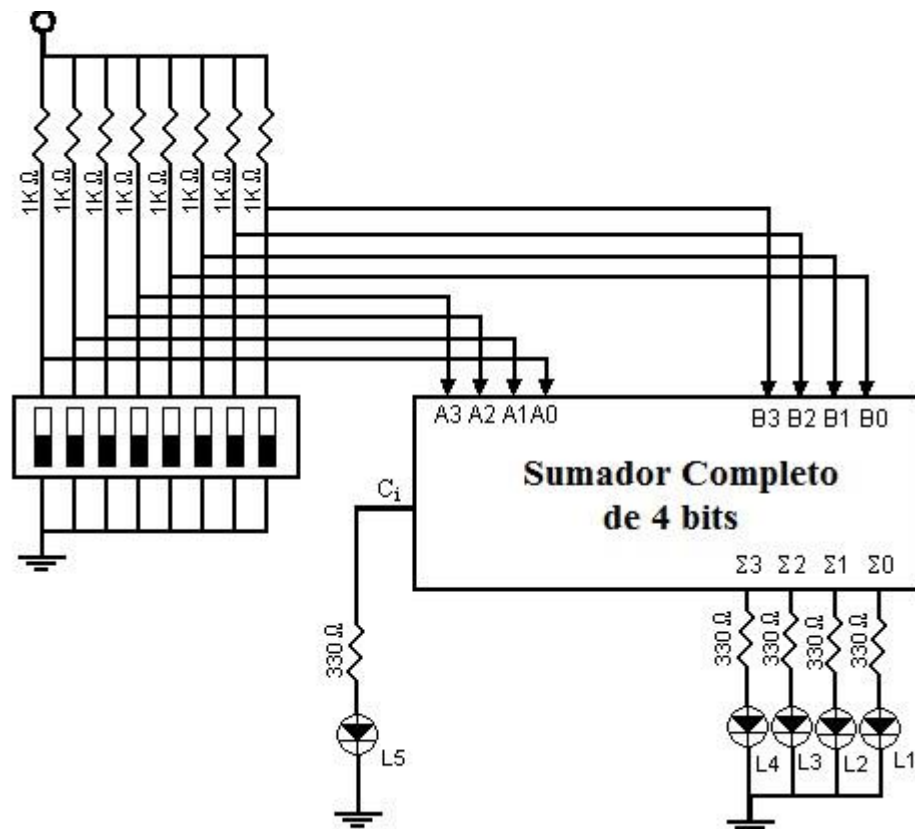
- Implemente su solución usando VHDL, este es el programa que contiene los 4; medio sumador, sumador total, medio restador y restador total.

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity operaciones is
5  port(a,b,c: in std_logic;
6       ms,s,cout,carry,mr,p,r,pout: out std_logic);
7  end operaciones;
8
9  architecture ope of operaciones is
10
11  begin
12      ms <= a xor b;
13      carry <= a and b;
14      s <= a xor b xor c;
15      cout <= (b and c) or (a and c) or (a and b);
16      mr <= a xor b;
17      p <= (not a) and b;
18      r <= a xor b xor c;
19      pout <= ((not a) and c) or ((not a) and b) or (b and c);
20  end ope;
```

5.5.- Sumador Completo

a) Diseñe en VHDL y arme el siguiente circuito.

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity sumador is
5  port (A,B:in std_logic_vector(3 downto 0);
6        S:out std_logic_vector(3 downto 0);
7        Cout: out std_logic);
8  end sumador;
9
10 architecture arqsuma of sumador is
11  signal C:std_logic_vector(2 downto 0);
12
13  ATTRIBUTE synthesis_off: BOOLEAN;
14  ATTRIBUTE synthesis_off OF C:SIGNAL IS true;
15
16 begin
17  S(0) <= A(0) xor B(0);
18  C(0) <= A(0) and B(0);
19  S(1) <= (A(1) xor B(1)) xor C(0);
20  C(1) <= (A(1) and B(1)) or (C(0) and (A(1) xor B(1)));
21  S(2) <= (A(2) xor B(2)) xor C(1);
22  C(2) <= (A(2) and B(2)) or (C(1) and (A(2) xor B(2)));
23  S(3) <= (A(3) xor B(3)) xor C(2);
24  Cout <= (A(3) and B(3)) or (C(2) and (A(3) xor B(3)));
25 end arqsuma;
```

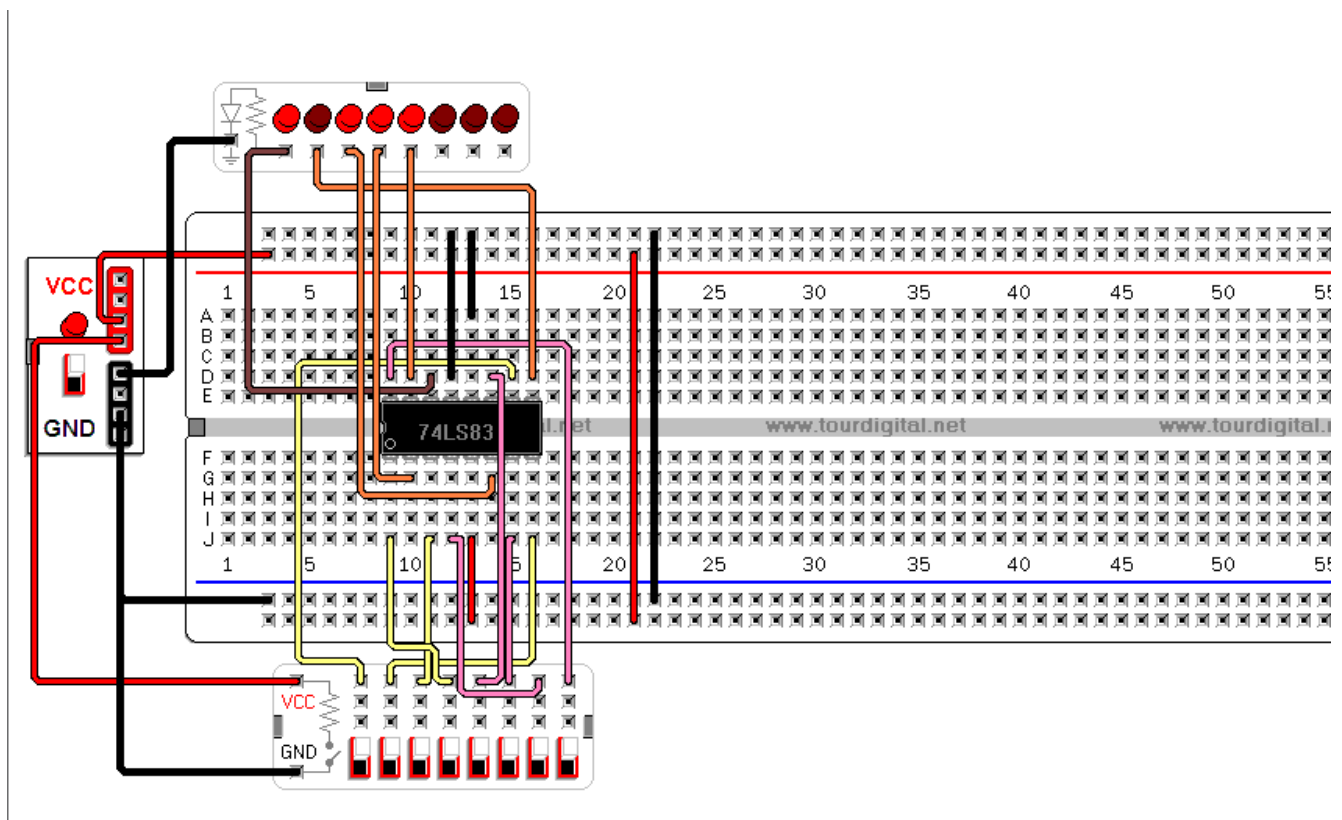


Fundamentos de Diseño Digital

Verifique algunas sumas que usted establezca y confirme sus resultados

#	Co	A3	A2	A1	A0	B3	B2	B1	B0	Ci	$\Sigma 3$	$\Sigma 2$	$\Sigma 1$	$\Sigma 0$
10	0	0	1	0	1	0	1	0	1	0	1	0	1	0
30	0	1	1	1	1	1	1	1	1	1	0	1	1	0
14	0	0	1	1	1	0	1	1	1	0	1	1	1	0
8	0	0	1	0	0	0	1	0	0	0	0	1	0	0
24	0	1	1	0	0	1	1	0	0	1	1	0	0	0
17	0	1	0	1	0	0	1	1	1	1	0	0	0	1
20	0	1	0	1	0	1	0	1	0	1	0	1	0	0
27	0	1	1	1	1	1	1	0	0	1	1	0	1	1
12	0	0	1	1	0	0	1	1	0	0	1	1	0	0
21	0	0	1	1	0	1	1	1	1	1	0	1	0	1
28	0	1	1	1	0	1	1	1	0	1	1	1	0	0

- Armado con 74LS83



6) Conclusiones Individuales.

A partir de los ejercicios realizados durante la práctica con ayuda del video del profesor, aprendí a diferencias entre un restador y sumador, además de la funcionalidad de cada uno en la electrónica digital, así mismo, obtuve un poquito más de experiencia programando con VHDL al realizar el código para los sumadores y restadores.

7) Bibliografía.

Anton, R. (2018). "Sumadores". Recuperado de:
https://personales.unican.es/manzanom/Planantiguo/EDigital/Sum_G5_08.pdf

Laniz, A. (2017). "Sumadore y Restador". Recuperado de:
<https://compilandoconocimiento.com/2017/06/13/sumador-y-restador/>

8) ANEXOS.

