Regla de la suma:

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10

Elementos de la Suma

1	1	1	0	0	Carry
	1	0	1	1	Numero A
	1	1	1	0	Numero B
1	1	0	0	1	∑

Ejemplos:

	1	1	1	1	1	1	0
		1	0	1	1	1	1
+		1	1	1	0	1	1
=	1	1	0	1	0	1	0

	10	11	11	11	11	11	10	0
			1	0	1	1	1	1
		1	1	1	0	0	0	1
		1	0	1	1	1	1	1
		1	1	1	0	1	1	1
	+		1	1	1	0	1	1
R//	10	0	1	1	0	0	0	1

SUMA DE BINARIO Circuito semi-sumador

Circuito Logico Semisumador

Este circuito lógico permite sumar 2 bits de 2 números compuestos de 1 bits (A y B) el cual solo genera un bit de salida (Carry out).

La operación del circuito semisumador se sintetiza en 2 operaciones booleanas:

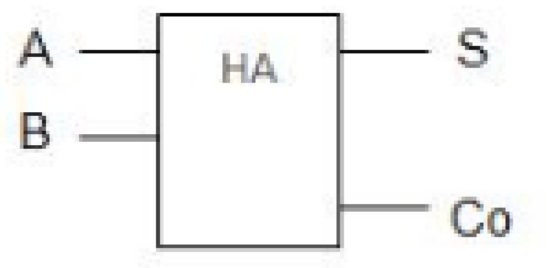
Suma= $A \oplus B$ la cual se lee: S = A X0R B

$$S = A\overline{B} + \overline{A}B$$

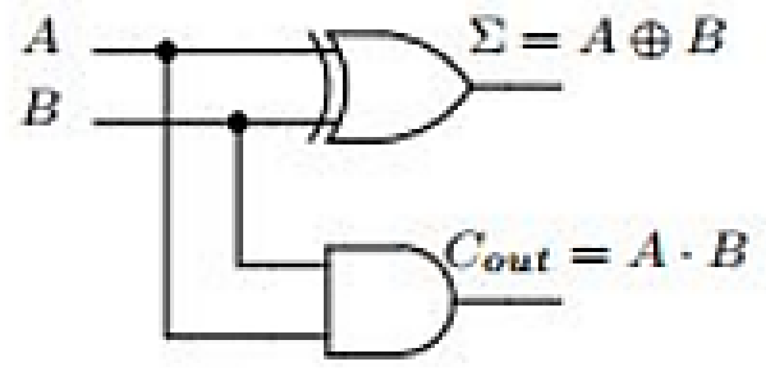
Y el carry sería: $C_{out} = A \, . \, B$

DIAGRAMAS DEL CIRCUITO SEMI-SUMADOR

Circuiro Estandarizado



Circuito no Estandarizado



Estos circuitos realizan operaciones booleanas es de hacer notar que estamos viendo 2 tipos de operaciones unas que son sumas binarias y otra que es suma lógica, que se generan por el funcionamiento de dichos circuitos lógicos en la computadora; en una operación binaria 1+1= 10 en una suma boolena 1+1= 1 Comprobando las operaciones si A=1 y B=1 entonces la operación binaria sería igual a 10; en donde el valor de 0= suma y el valor de 1 = carry de salida (C_{out})

De forma Booleana si A= 1 y B=1

$$S = A\bar{B} + \bar{A}B$$

$$1=\bar{1} + \bar{1}.1$$

$$\cancel{1}.0 + \cancel{0}.1$$

$$0 + 0$$

$$S = 0$$

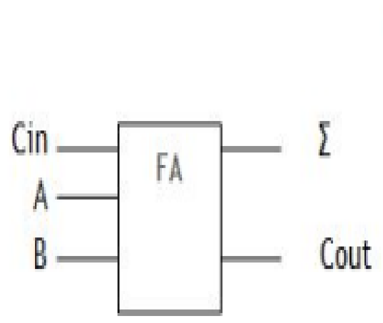
$$C_{out} = A.B$$

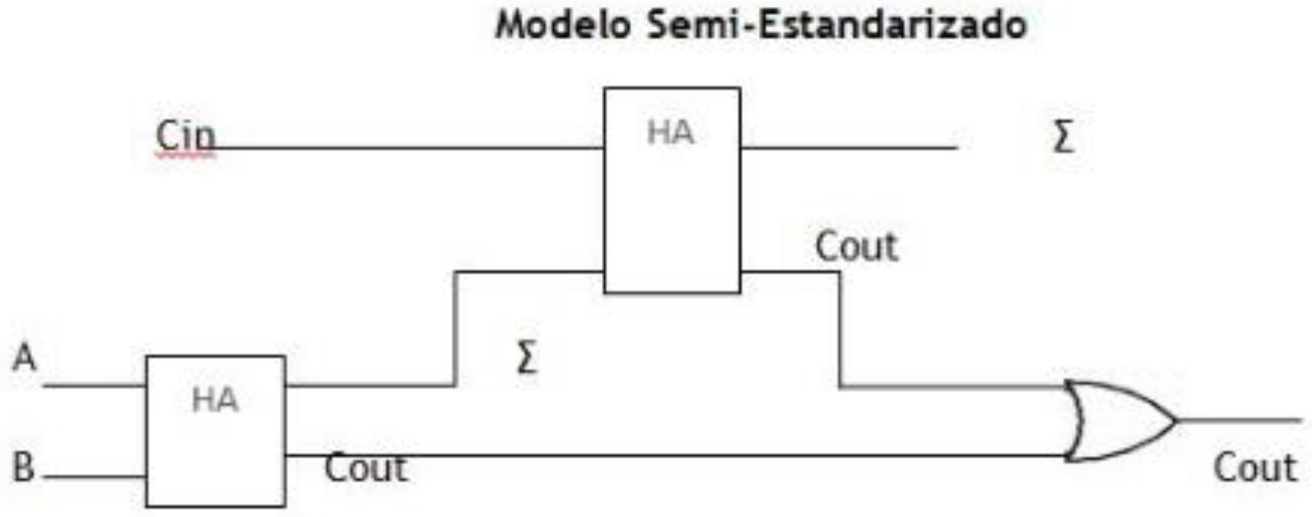
$$1.1$$

$$C_{out} = 1$$

Circuito Sumador Completo

Este circuito es el encargado de realizar la suma binaria de "n" números con "n" bits.
A diferencia del circuito lógico semisumador este implementa una variable adicional de entrada llamada Cin y sus entradas habituales A y B, sus salidas siempre son las mismas S (∑) y Cout.

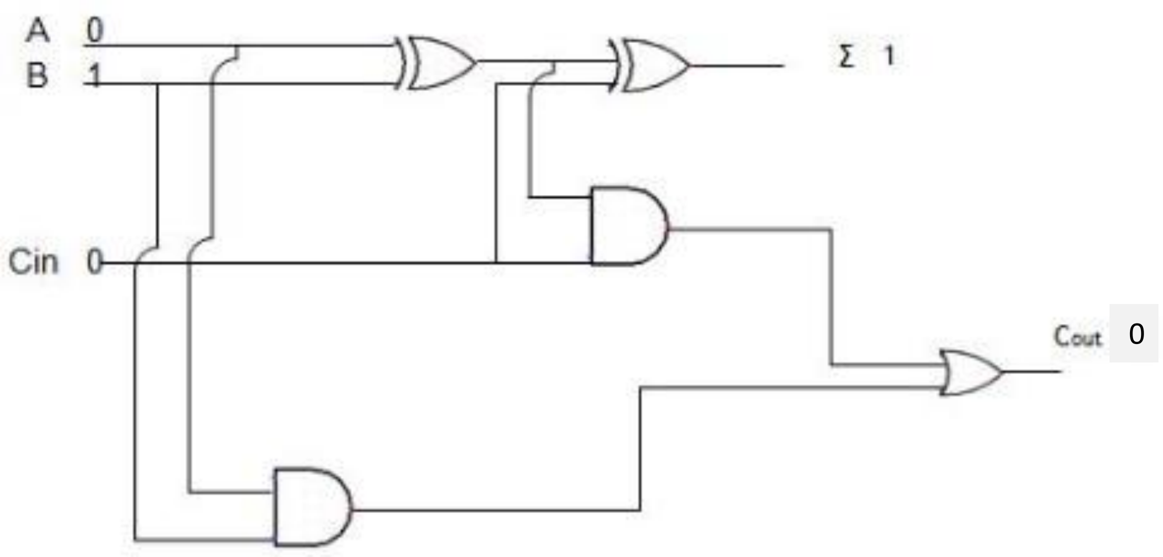




CIRCUITO SUMADOR COMPLETO DE 1 BIT

Si cada par de sumadores binarios pueden producir o generar un bit de acarreo, también debe de tener la c apacidad de reconocer la asignación de un bit de acarreo del sumador de nivel inferior.

Para lograr este propósito se implementa el siguiente circuito con sus salidas.



ENTRADAS			SALIDAS	
A	B	Cin	Cout	suma
0	0	0		
0	0	1		
0	1	0	0	1
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	0		
1	1	1		

Operaciones booleanas que se llevan acabo por parte del circuito sumador completo.

$S = A\bar{B} + \bar{A}B.$

$C = A \,.\, B$

Operaciones Booleanas

$S1 = A\bar{B} + \bar{A}B$

$0.\bar{1} + \bar{0}.1$

$00 + 1.1$

$0 + 1$

1

$S2 = A\bar{B} + \bar{A}B$

$1.\bar{0} + \bar{1}0$

$1.1 + 00$

$1 + 0$

1

$Coit1 = S1 \,.\, Cin$

$1 \,.\, 0$

0

$Coit2 = A.B$

$0 \,.\, 1$

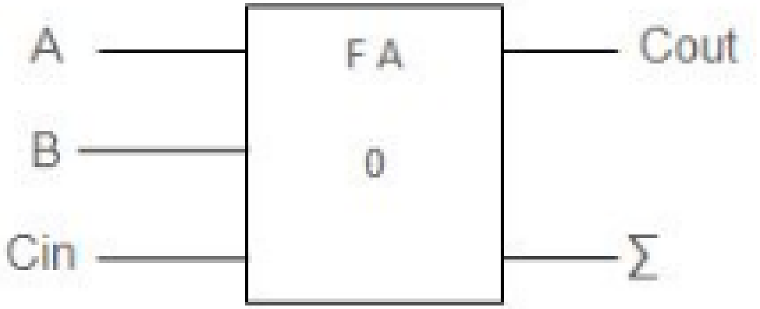
0

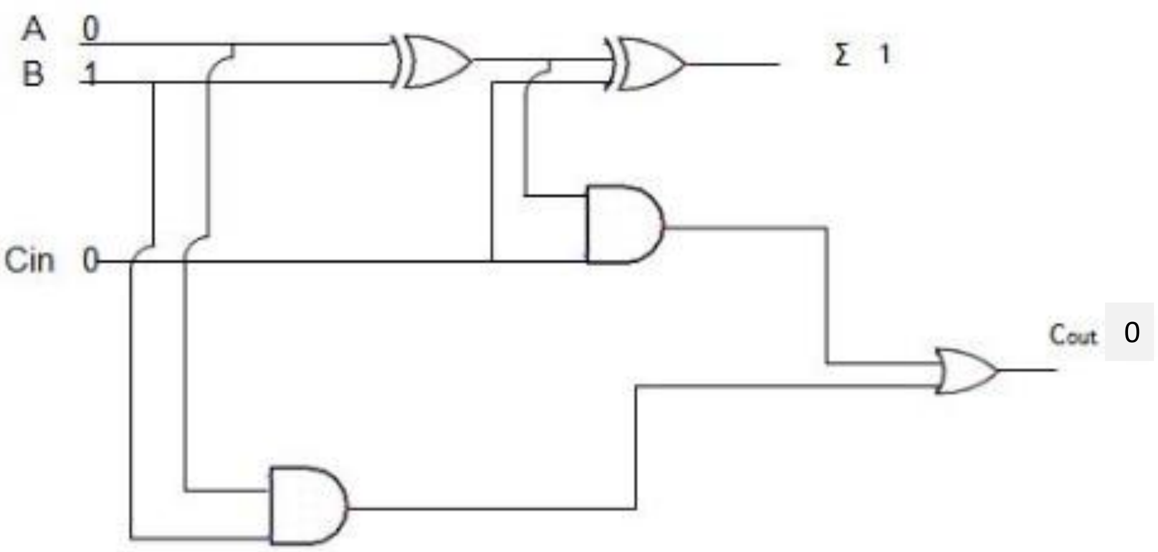
$Coit3 = Coit1 \times Coit2$

$0 \,.\, 0$

0

El circuito anterior es un poco com plicado de diseñar y graficar así que se puede reemplazar por una caja negra con 3 entradas y 2 salidas. (dos entradas cuando son 2 números los que se desean sumar)





ENTRADAS			SALIDAS	
A	B	Cin	Coit	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

IMPLEMENTACIÓN DEL CIRCUITO SUMADOR COMPLETo CON " N " BIT´S

El **sumador binario completo de n bits** se basa en el sumador binario completo de 1 bit

El sumador completo de " n " bit´s es una concatenación de " n " sumadores binarios completos de 1 bits. La concatenación se realiza a través de los terminales de acarreo saliente y entrantes.

Sumador de 4 bit´s: (N = 4) Suma 2 números binarios de 4 bit´s cada uno.

$A = A_3\ A_2\ A_1\ A_0$; $B = B_3\ B_2\ B_1\ B_0$; Entonces la suma será:

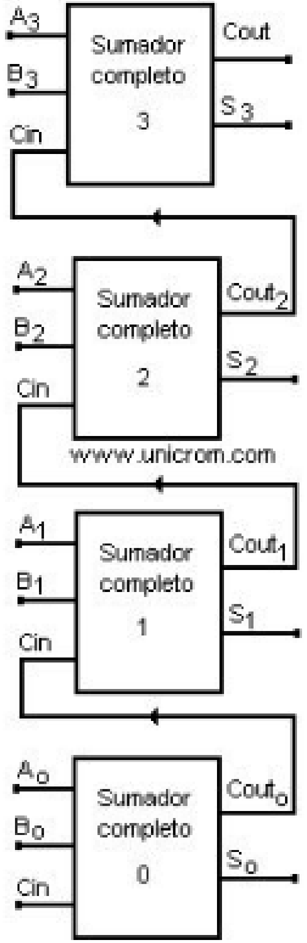
$S = Cout3,\ S_3\ S_2\ S_1\ S_0$

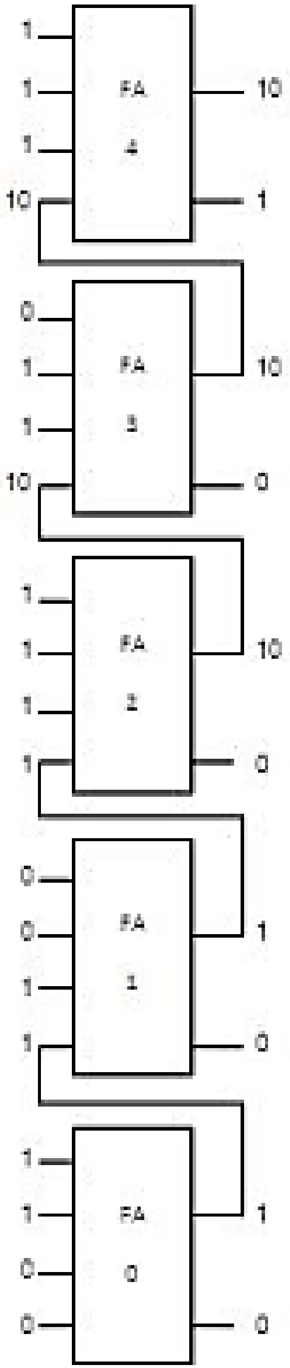
El bit menos significativo en los sumadores A y B son y $A_0\ B_0$ y el bit mas significativo es $A_3\ B_3$.

La suma inicia en el FA0 con la suma $\mathbf{A_0{+}B_0}$, si esta suma tuviese acarreo **Cout = 1** este pasaría al FA1 y así sucesivamente hasta llegar al FA3.

Si el sumador superior tiene acarreo (" 1 ") este se refleja en la suma al lado izquierdo de la suma final.

IMPLEMENTADO.





R// = 1010000

Tratemos de hacer lo siguiente:

Realizar la siguiente suma implementando el FA para el siguiente número

```
  10111
  11101
  11011
+ 10111
-------
```

Implementar el FA para las siguientes sumas:

$$\begin{array}{r} 10101 \\ 11101 \\ +\ 11110 \\ \hline \end{array}$$

Resta de binarios

Resta de Binarios

La resta o sustracción de números binarios es similar a los números decimales.
La diferencia radica en que, en binario, cuando el minuendo es menor que el sustraendo, se produce un **barrow o préstamo de 2**, mientras que en decimal se produce los prestamos de 10.

Regla de la suma:

Reglas de la Resta:

$0 - 0 \rightarrow 0$

$1 - 0 \rightarrow 1$

$1 - 1 \rightarrow 0$ $\qquad 10 - 1 \rightarrow 1$

Es necesario hacer notar que cuando se presenta el caso de la operación 0 - 1 = ? en teoría el bit más significativo le debe hacer un préstamo a cero por lo que 0 se convierte en 10 por lo que la operación se hace 10-1= 1

Elementos de la Resta Binaria

Elementos de la Resta Binaria:

		0	10	0	10	→ Barrow
	1	1	0	1	0	→ Minuendo (+)
-			1	0	1	→ Sustraendo (-)
	1	0	1	0	1	→ Diferencia.

Ejemplos: $\mathbf{1110101_2 - 111111_2}$

$$
\begin{array}{ccccccccc}
 & & 10 & 10 & 1 & 10 & & & \\
 & 0 & 0 & 0 & \cancel{10} & 0 & 10 & & \\
 & \cancel{1} & \cancel{1} & \cancel{1} & 0 & \cancel{1} & 0 & 1 & \\
- & & 1 & 1 & 1 & 1 & 1 & 1 & \\
\hline
 & & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \\
 & & & & & & & & 2
\end{array}
$$

Ejemplos: **11100000100**$_2$ **-111111011**$_2$

			10	1	1	1	1			1	
		0	0	~~10~~	~~10~~	~~10~~	~~10~~	10	0	~~10~~	10
	1	~~1~~	~~1~~	0	0	0	0	0	~~1~~	0	0
-			1	1	1	1	1	1	0	1	1
	1	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1** $_2$

Realizar los siguientes ejercicios.

	1	1	0	0	0	0	0	0	0	1
-			1	1	1	1	1	1	1	1

Restar en Binario

$2AB_{16} + 37_8 - 31F_{16} - 77_8 + 8_{16}$

	1	1	1	1	1	1	0	0	0
-	1	0	0	0	0	0	0	1	1
+		1	0	1	0	1	1	0	1
+	1	1	1	1	0	0	1	1	1
-			1	1	1	1	0	0	0
-						1	1	1	1

RESTA DE BINARIO CIRCUITO SEMI-RESTADOR

Circuito Logico Semi-Restador

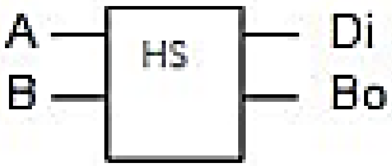
Es el circuito encargado de sustraer un bit de B de un bit de A y suministrar en bit de diferencia (Di) y un bit de préstamo la operación que se realiza es :

Diferencia = $A \oplus B$ es decir que : **Di** = A X0R B y **Bo** = $\overline{A}.B$

Di = $A\overline{B} + \overline{A}B$

DIAGRAMAS DEL CIRCUITO SEMI-RESTADOR

Modelo Estandarizado



$Di = A.\overline{B} + \overline{A}.B$

$0.\overline{1} + \overline{0}.1$

$0.0 + 1.1$

$0 + 1$

$Di = 1$

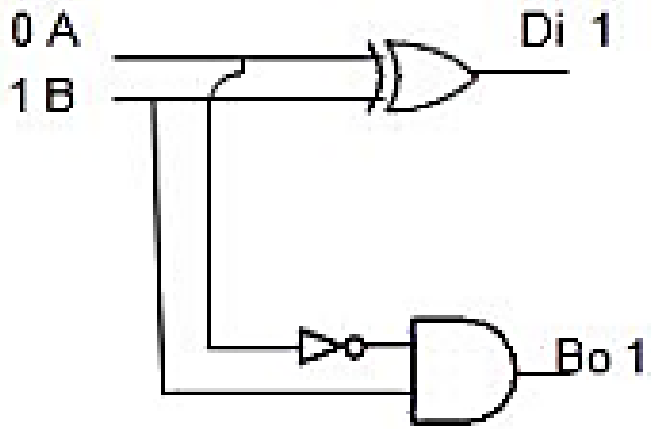
$Bo = \overline{A}.B$

$\overline{0}.1$

1.1

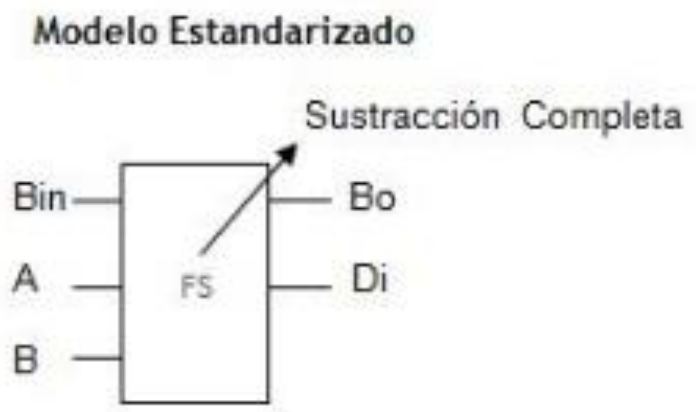
$Bo = 1$

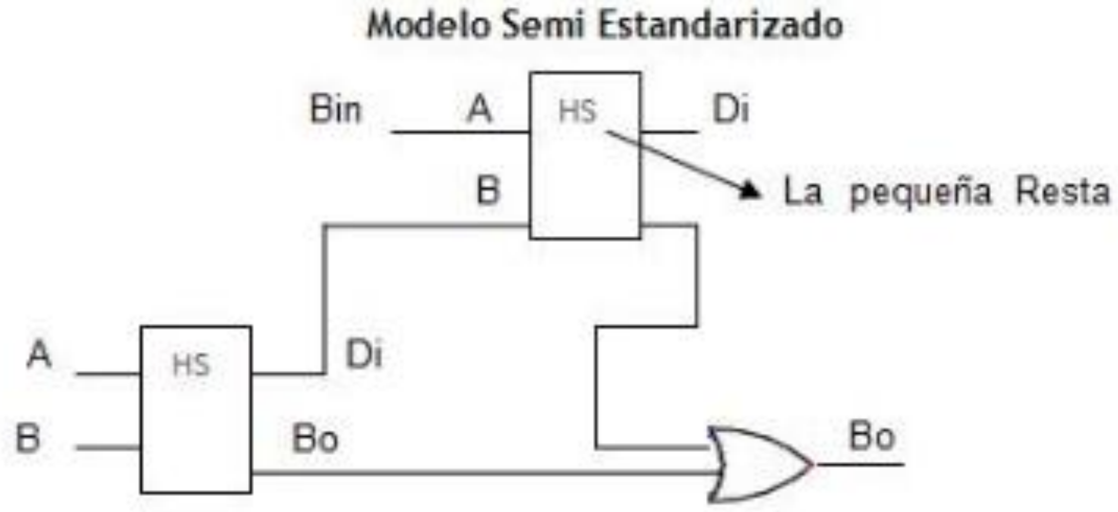
Modelo No Estandarizado



CIRCUITO LÓGICO RESTADOR COMPLETO

Al igual que el FA el FS hace uso de una variable adicional denominada Bin (barrow de entrada). Sus entradas habituales A y B y las Salidas Di y Bo.





Sistema de Conversión Numérico con Complemento a 2

Sistemas Numéricos con Complemento

Este sistema nos permite representar los números binarios de forma negativa, en donde el MSB es el bit del signo.

Si este bit es “ 0 “ entonces el numero binario es positivo (+) si el bit signo es 1, entonces el numero binario es negativo (-). Los siguientes bit´s restantes del registro representa la magnitud del numero.

COMPLEMENTO A1:

El complemento a1 de un numero binario se obtiene cambiando cada 0 por 1 y viceversa. Se cambia cada bit número por su complemento.

EJEMPLO:

10111001 → Numero Original

01000110 → Complemento a1

COMPLEMENTO A2:

El complemento a2 de un numero binario se obtiene tomando el complemento a1 del numero y sumando 1 al bit menos significativo. Al bit mas significativo se le agrega el bit “ 1 “.

EJEMPLO:

9 = 1001   Encontrar el C2 del numero.

Método Abreviado del C2. → 01001 → C2

10111

C1 = 0110

+ 1

C2 10111 ≃ -9