Sumadores Lógicos

Electrónica Digital

G10 - Javier Frances Hevia

Elementos de la Presentación

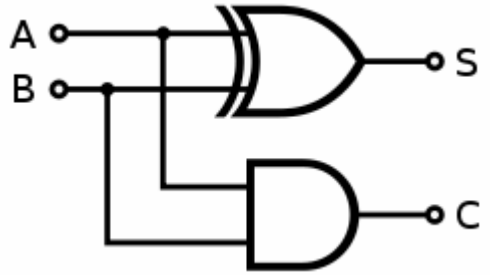
- Descripción conceptual
- Ejemplo comercial
- Aplicaciones
- Problemas propuestos

Descripción conceptual

- Los sumadores lógicos realizan la operación aritmética de la suma.
- Normalmente las operaciones se realizan en código binario, si bien pueden desarrollarse sumadores en otros formatos.
- Es importante evaluar el tamaño y la velocidad a la que trabajan en función de lo que busquemos
- El elemento mas simple de los circuitos sumadores es el Half-Adder. Realiza sumas de 2 elementos de 1 bit.

Sumador Half-Adder

- Realiza la operación aritmética de la suma de 2 operandos de 1 bit.
- Tiene 1 bit de acarreo de salida.
- Está formado por una puerta XOR y una AND.



A B	C S
0 0	0 0
0 1	0 1
1 0	0 1
1 1	1 0

$S = A \oplus B$

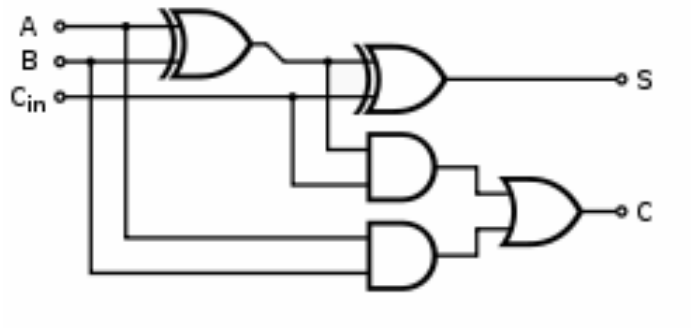
$C = A\,B$

Sumador Full-Adder

A B Ci	Co S
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

$$S = A \oplus B \oplus Ci$$
$$Co = A\,B + A\,Ci + B\,Ci$$

- Se forma con 2 Half-Adders y una OR
- Tiene 1 bit de acarreo de entrada y otro de salida.

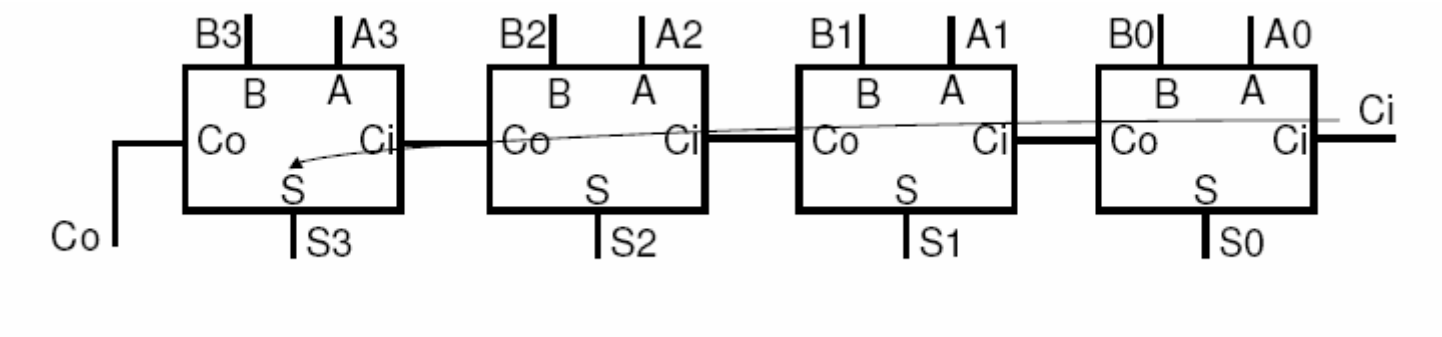


Sumadores de N bits

- Si se pretende hacer un sumador de N bits, el único problema a solventar es la aparición (y desplazamiento) de los acarreos.
- El Full-Adder es el elemento básico para realizar circuitos sumadores complejos.
- Lo mas sencillo es la distribución Ripple (serie)

Sumador Ripple

- Permite realizar la suma de 2 elementos de N bits.
- Se forma colocando N Full-Adders en serie.



Características:

- El primer término que se obtiene es el menos significativo y el último es el más significativo.
- La salida es de **N+1** bit, ya que el acarreo del último término es parte de la solución.
- No tenemos la solución correcta hasta que se cumplan los tiempos de propagación de todos los sumadores. Implica lentitud. **PROBLEMA**

Sumador Carry Look-Ahead

- El problema de la propagación se solventa anticipando el acarreo.
- Para cada bit, las salidas de acarreo y de suma son independientes de los bits anteriores.
- Por tanto, si el efecto de propagación se ha eliminado, se obtiene el resultado mas rápidamente.

El acarreo puede generarse por 2 motivos:

- Se genera acarreo en la propia etapa del sumador. **Generado**

$$Gj = Aj * Bj$$

- Proviene de la etapa anterior. **Propagado**

$$Pj = Aj \oplus Bj$$

Por tanto el acarreo producido en la etapa i-esima Ci será porque se genera o propaga y se expresa como:

$$C_i = G_i + P_i\, C_{i-1} = A_i\, B_i + (A_i + B_i)\, C_{i-1}$$

Ejemplo de Carry Look-Ahead de 2 entradas de 4 bits.

1) Anticipamos los acarreos de cada término:

T1

$$P_j = A_j \oplus B_j$$
$$G_j = A_j B_j$$

2) Generamos los acarreos de manera independiente de las entradas:

T2

$$C1 = G0 + P0Ci$$
$$C2 = G1 + P1\,C1 = G1 + P1\,(G0 + P0\,Ci) = G1 + P1G2 + P1P0Ci$$
$$C3 = G2 + P2\,C2 = G2 + P2G1 + P2P1G0 + P2P1P0Ci$$
$$Co = C4 = G3 + P3\,C3 = G3 + P3G2 + P3P2G1 + P3P2P1G0 + P4P3P2P1Ci$$

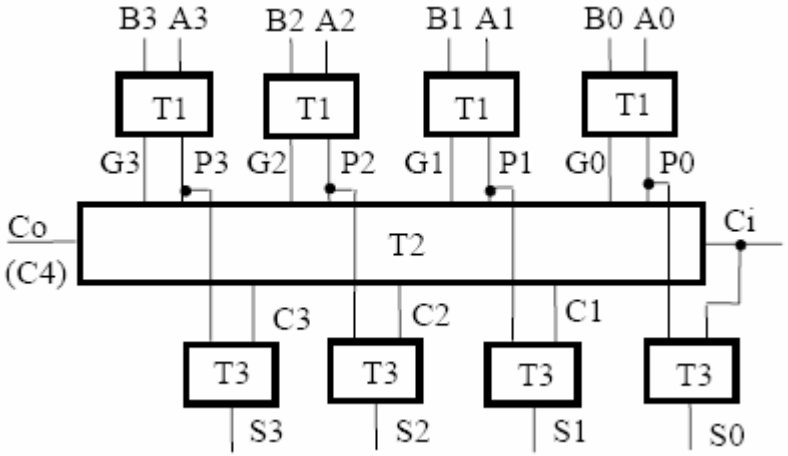
3) Generamos las salidas en función de los acarreos:

T3

$$S0 = P0 \oplus Ci$$
$$Sj = Pj \oplus Cj$$

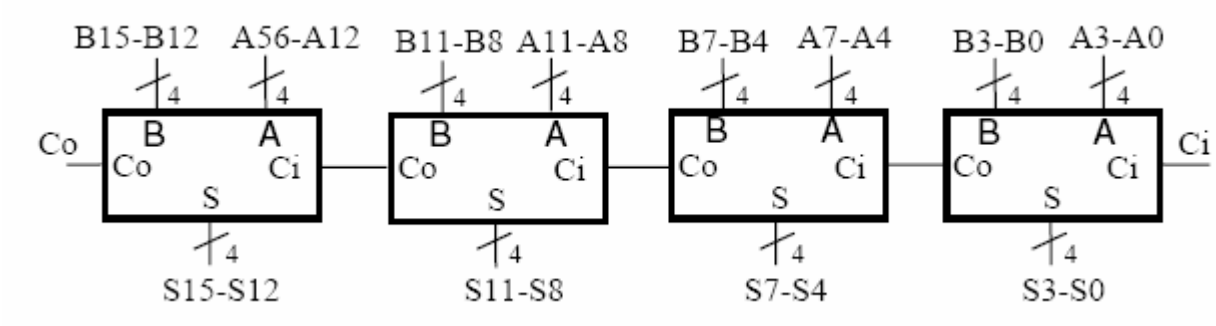
Ejemplo de Carry Look-Ahead de 2 entradas de 4 bits.

El circuito se forma de la siguiente manera a partir de T1, T2, T3



Con este circuito se obtiene la solución de manera mucho más rápida que con un circuito Ripple, si bien se utilizan muchos más elementos lógicos. **PROBLEMA**

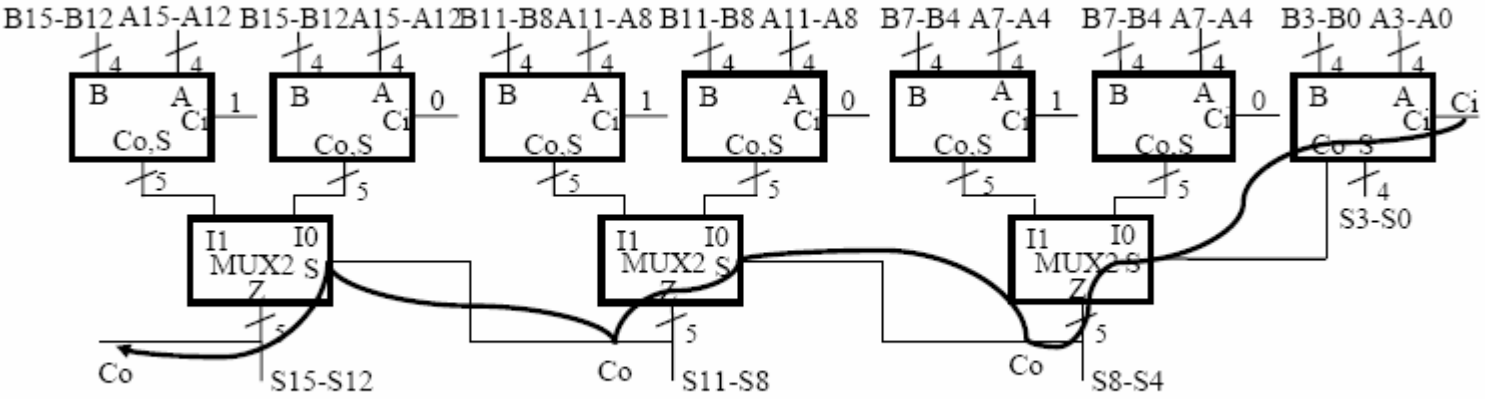
Concretamente, las ecuaciones que definen los acarreos son cada vez más grandes, luego no es rentable continuar aumentando el circuito. Una solución es poner sumadores Carry Look-Ahead de 4 bits en serie.



Sumador Carry Select

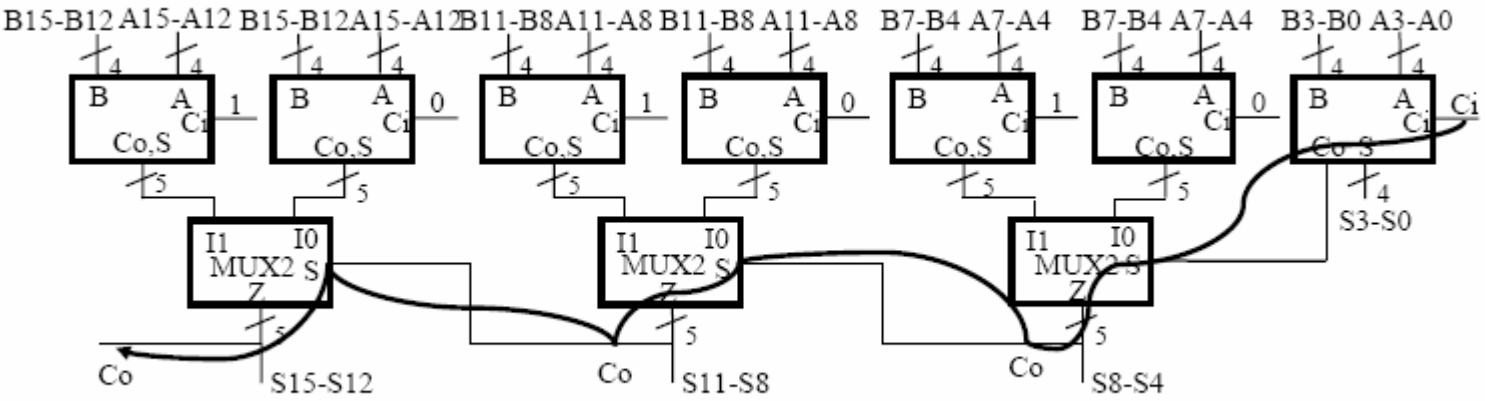
- En este circuito sumador se incluyen multiplexores y la idea es similar al del Carry Look-Ahead y se pretende anticipar el acarreo.

- Se realizan todas las sumas simultáneamente, con la particularidad de simularlas con o sin acarreo. Luego, el acarreo real determinará cual de las dos soluciones es la correcta.



Sumador Carry Select

- El tiempo de propagación de este sumador depende del tiempo de propagación de la primera etapa, más el tiempo de propagación de los multiplexores para propagación del acarreo. Sigue necesitando mucha mas lógica que el Ripple. **PROBLEMA**



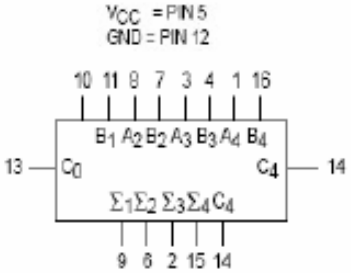
Sumador comercial - 74LS83

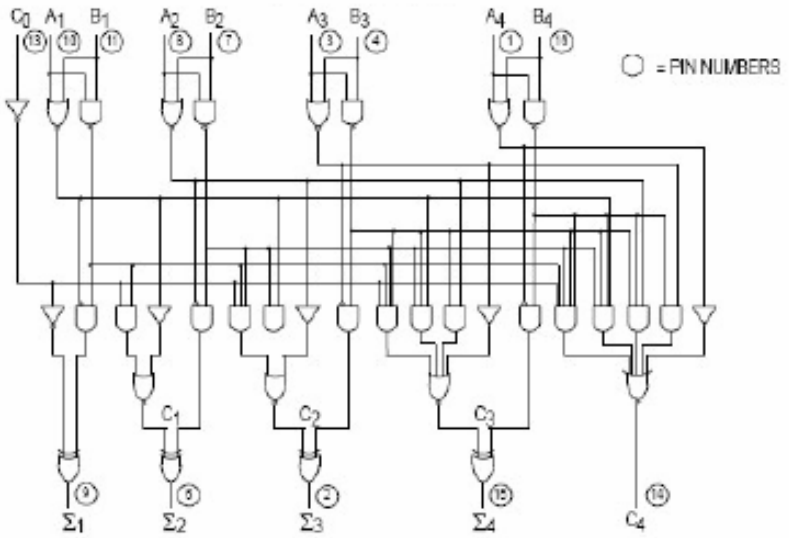
Circuitos comerciales: 74'83. Sumador de 4 bits con estructura interna de carry look-ahead. El circuito opera como sumador suponiendo las entradas y salidas en polaridad positiva o en polaridad negativa.

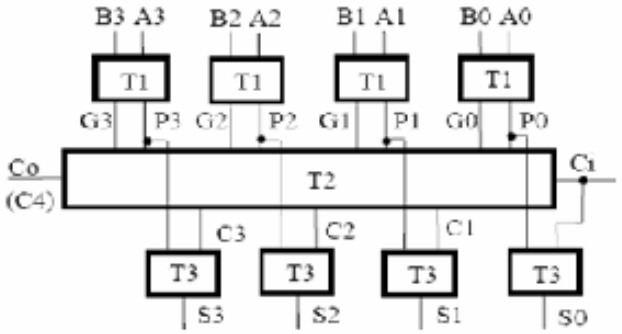
$C_0 + (A_1+B_1)+2(A_2+B_2)+4(A_3+B_3)+8(A_4+B_4) = \Sigma_1+2\Sigma_2+4\Sigma_3+8\Sigma_4+16C_4$

Where: (+) = plus

	C_0	A_1	A_2	A_3	A_4	B_1	B_2	B_3	B_4	Σ_1	Σ_2	Σ_3	Σ_4	C_4	
Logic Levels	L	L	H	L	H	H	L	L	H	H	H	L	L	H	
Active HIGH	0	0	1	0	1	1	0	0	1	1	1	0	0	1	(10+9 = 19)
Active LOW	1	1	0	1	0	0	1	1	0	0	0	1	1	0	(carry+5+6 = 12)



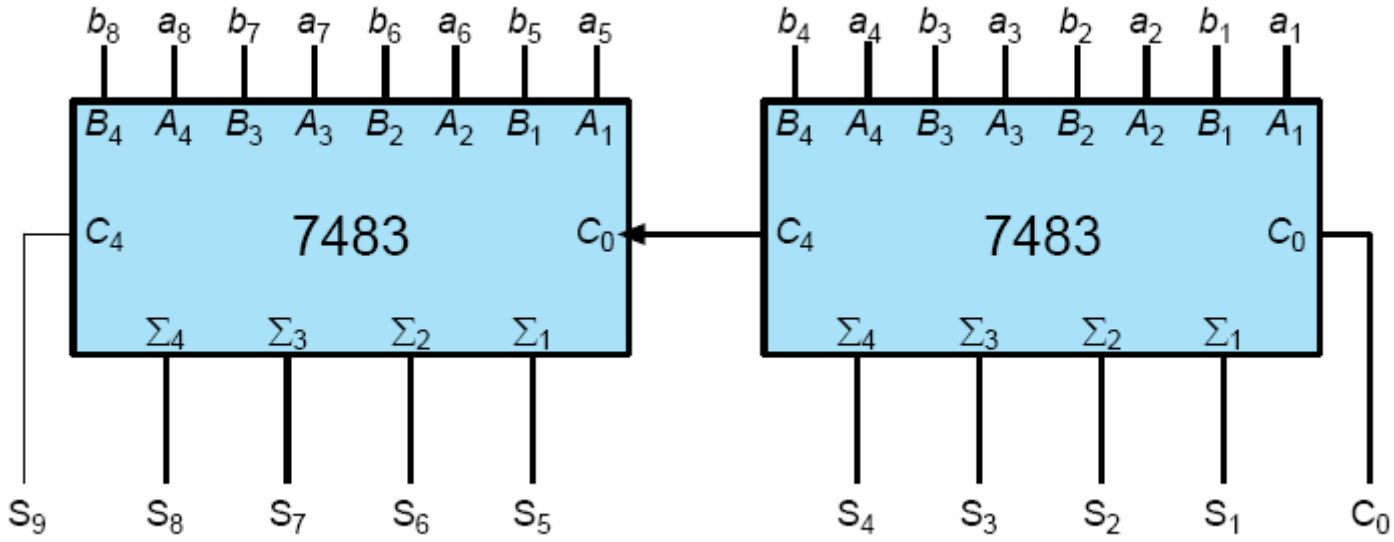




Co y Ci están cambiados en el ejemplo

Sumador comercial - 74LS83

- En este ejemplo se realiza un sumador de 8 bits con un integrado 74LS83

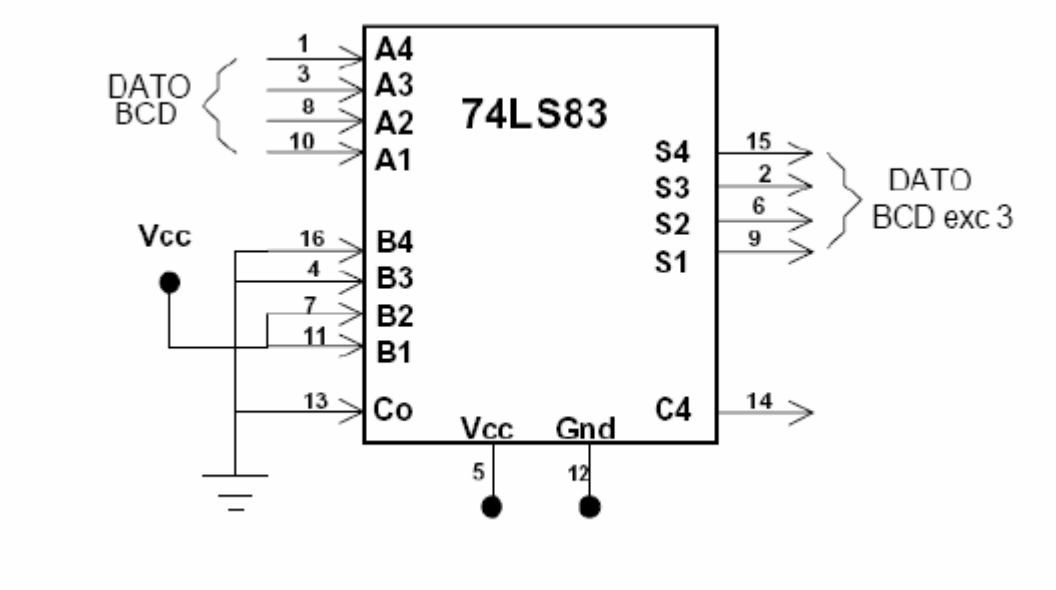


Aplicaciones de los sumadores

- Los sumadores son la base de cualquier ALU (unidad aritmético lógica)
- Con ellos podemos realizar otros operadores, como una resta o una multiplicación.
- Por ejemplo otra aplicación sería el conversor BCD en exceso a 3

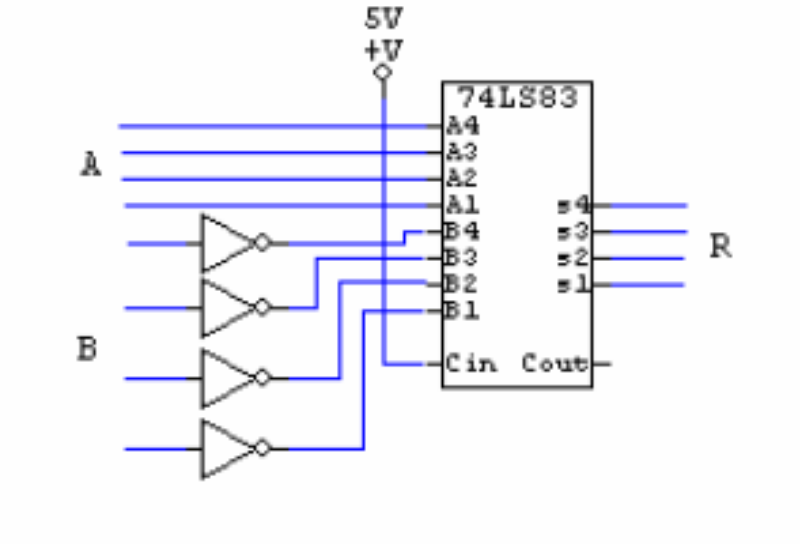
Conversor BCD exceso a 3

- El código BCD en exceso a 3 es ampliamente utilizado en electrónica digital. Y simplemente con un sumador obtenemos un conversor entre BCD y BCD en exceso a 3.



Restador

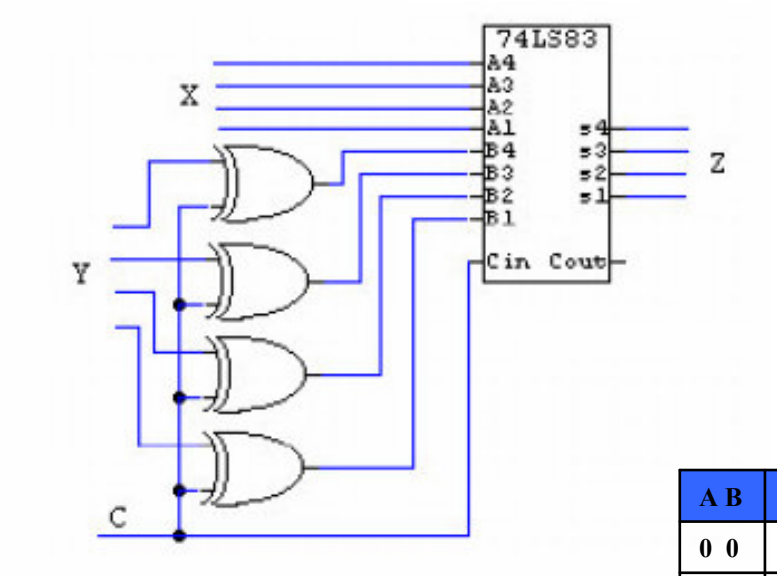
- Un restador se realiza con un sumador suponiendo los operandos en complemento-2.
- En este caso, al estar A y B en 4 bits en c-a-2, el resultado de la resta R tiene tantos bits como las entradas, el acarreo de salida no se utiliza y existe la posibilidad de que se produzca “overflow”.



Sumador / Restador

- Siguiendo el esquema anterior podemos realizar un sumador / restador con un selector.

- Si C=0 realiza una suma (Y se queda tal cual)
- Si C=1 realiza una resta (Y cambia 1's por 0'a)



A B	X
0 0	0
0 1	1
1 0	1
1 1	0

Ejercicios Propuestos

- A partir de un Half-Adder diseñar un Half-Substrator (realizar primero la tabla de verdad)
- Diseñar un sumador de 8 bits con dos sumadores de 4 bits mediante el método Ripple y Carry-Select. Comparar numero de niveles, puertas y tiempos de propagación.
- Diseñar un multiplicador de 2 entradas de 3 bits con semisumadores y sumadores completos de 1 bit
- Diseñar un circuito que realice la siguiente operación con operandos de 2 bits y usado sumadores de dos bits:

  $Z=5*A+2*B+C$
- Diseñar un sumador de códigos BCD (Nota: el código BCD está definido de 0 a 9)

Fuentes consultadas:

- Apuntes de la asignatura:
  http://personales.unican.es/manzanom/EDigitalII/Tema_V.pdf
  http://personales.unican.es/manzanom/EDigitalII/Sum_G5_08.pdf
  http://personales.unican.es/manzanom/EDigitalII/Sum_G11_08.pdf

Otras fuentes:

- http://es.wikipedia.org
- http://www.terra.es/personal3/rtamayo/Archivos/Tema4.PDF
- http://www.uned.es/ca-bergara/ppropias/Morillo/web_etc_II/4_alu/transp_alu.pdf