

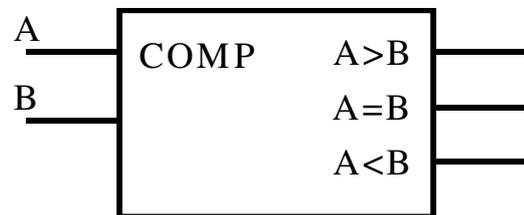
T6. CIRCUITOS ARITMÉTICOS

Son dispositivos **MSI** que pueden realizar operaciones aritméticas (suma, resta, multiplicación y división) con números binarios. De todos los dispositivos, nos centraremos en los **comparadores de magnitud, detectores y generadores de paridad, sumadores y ALU's**.

Comparadores de Magnitud

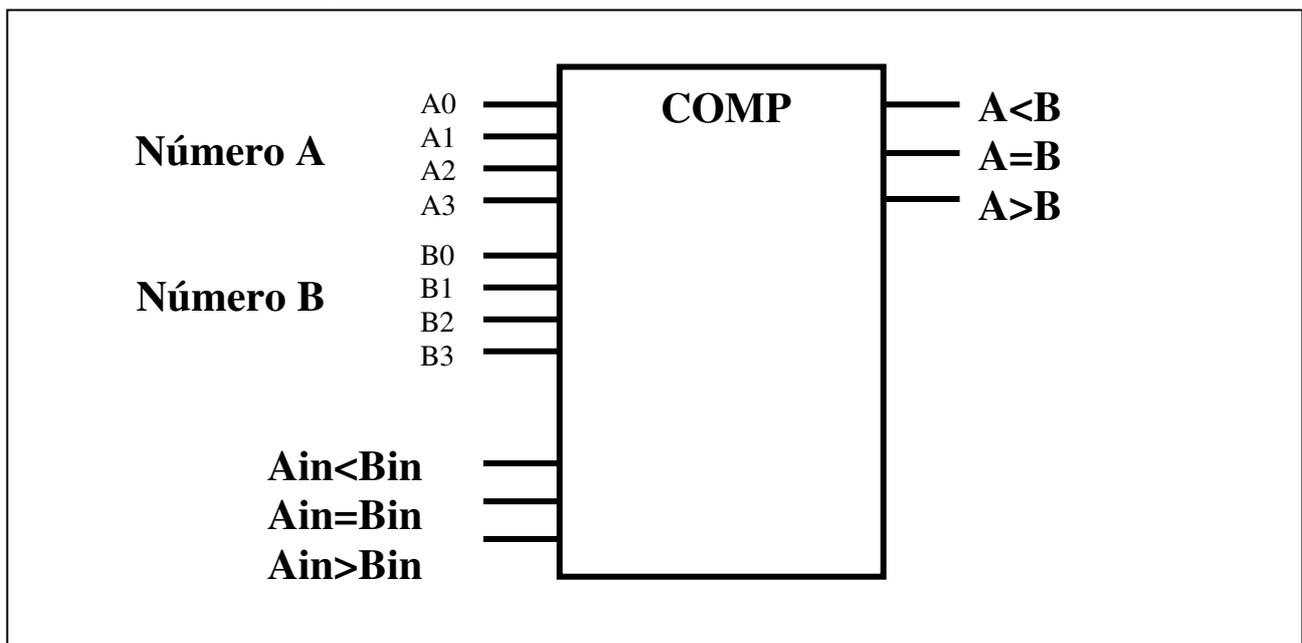
Son circuitos que comparan el valor binario de dos números, proporcionando información de cuál es mayor, menor, o si ambos son iguales. Son sistemas muy usados en ingeniería. Su bloque y tabla de funcionamiento básico son los siguientes

A	B	A>B	A=B	A<B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0



Existen comparadores de 4 bits y de 8 bits. Además de las correspondientes entradas de datos disponen de tres entradas más que pueden informar sobre una situación anterior, y que se usan para conectar en cascada distintos comparadores, de manera que pueda construirse uno de mayor capacidad.

Veámoslo con el comparador de 4 bits (7485):



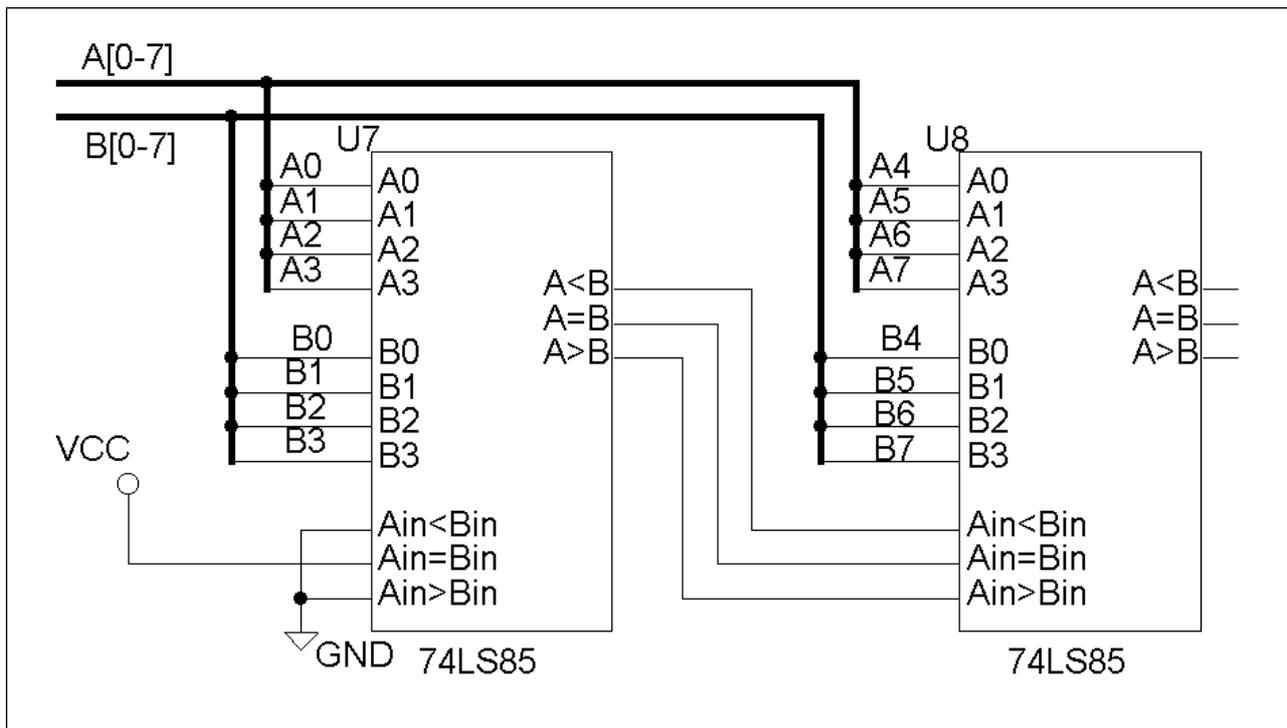
- Las Salidas del comparador se pueden deducir de las siguientes expresiones lógicas:

$$(A > B) = (N^{\circ} A > N^{\circ} B) \text{ or } [(N^{\circ} A = N^{\circ} B) \text{ and } (A_{in} > B_{in})]$$

$$(A = B) = (N^{\circ} A = N^{\circ} B) \text{ and } (A_{in} = B_{in})$$

$$(A < B) = (N^{\circ} A < N^{\circ} B) \text{ or } [(N^{\circ} A = N^{\circ} B) \text{ and } (A_{in} < B_{in})]$$

Haciendo uso de esas entradas de “comparación anteriores”, podemos diseñar un comparador de más bits, por ejemplo de 8.

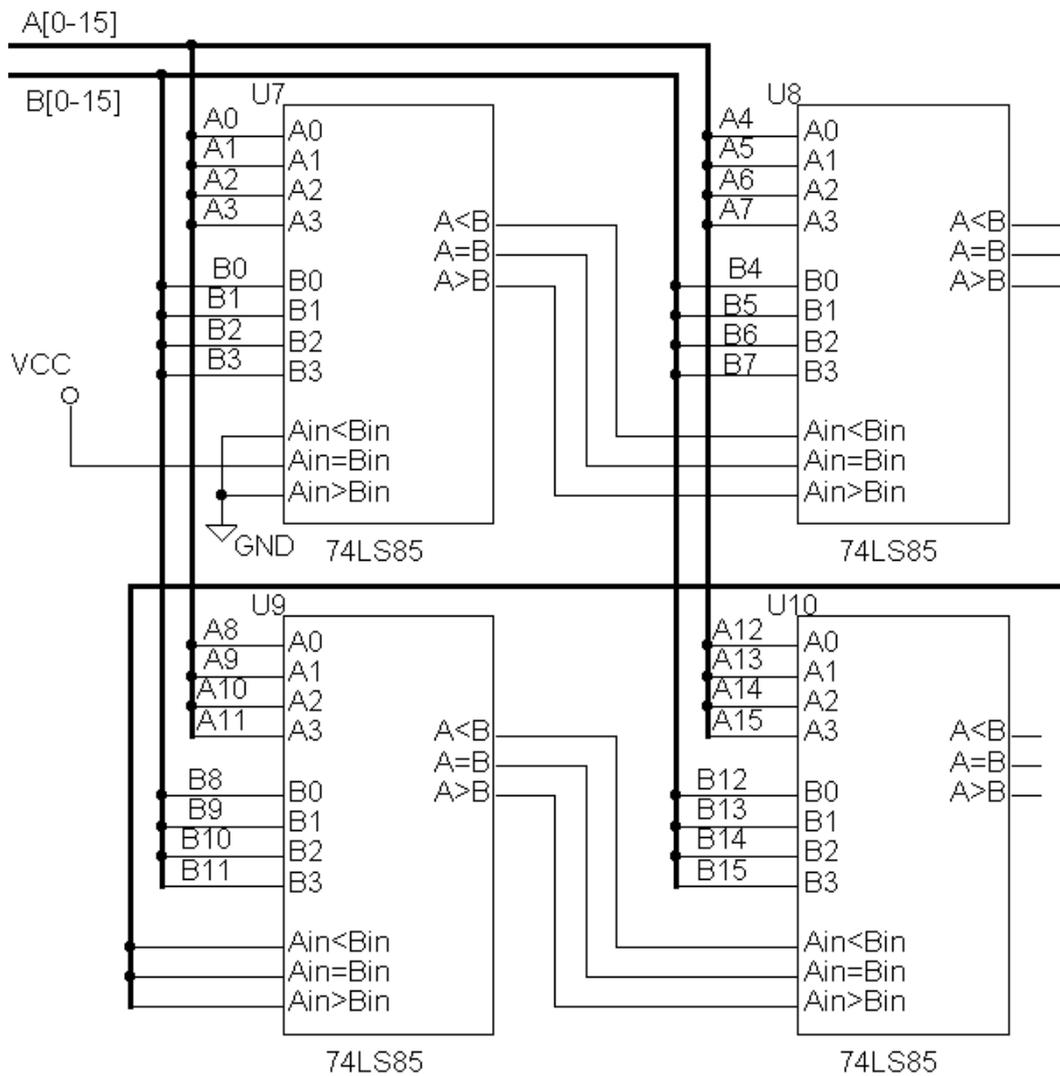


Así se comparan primero los bits menos significativos A[0-3] y B[0-3], para con la información obtenida comparar los más significativos; por ej.

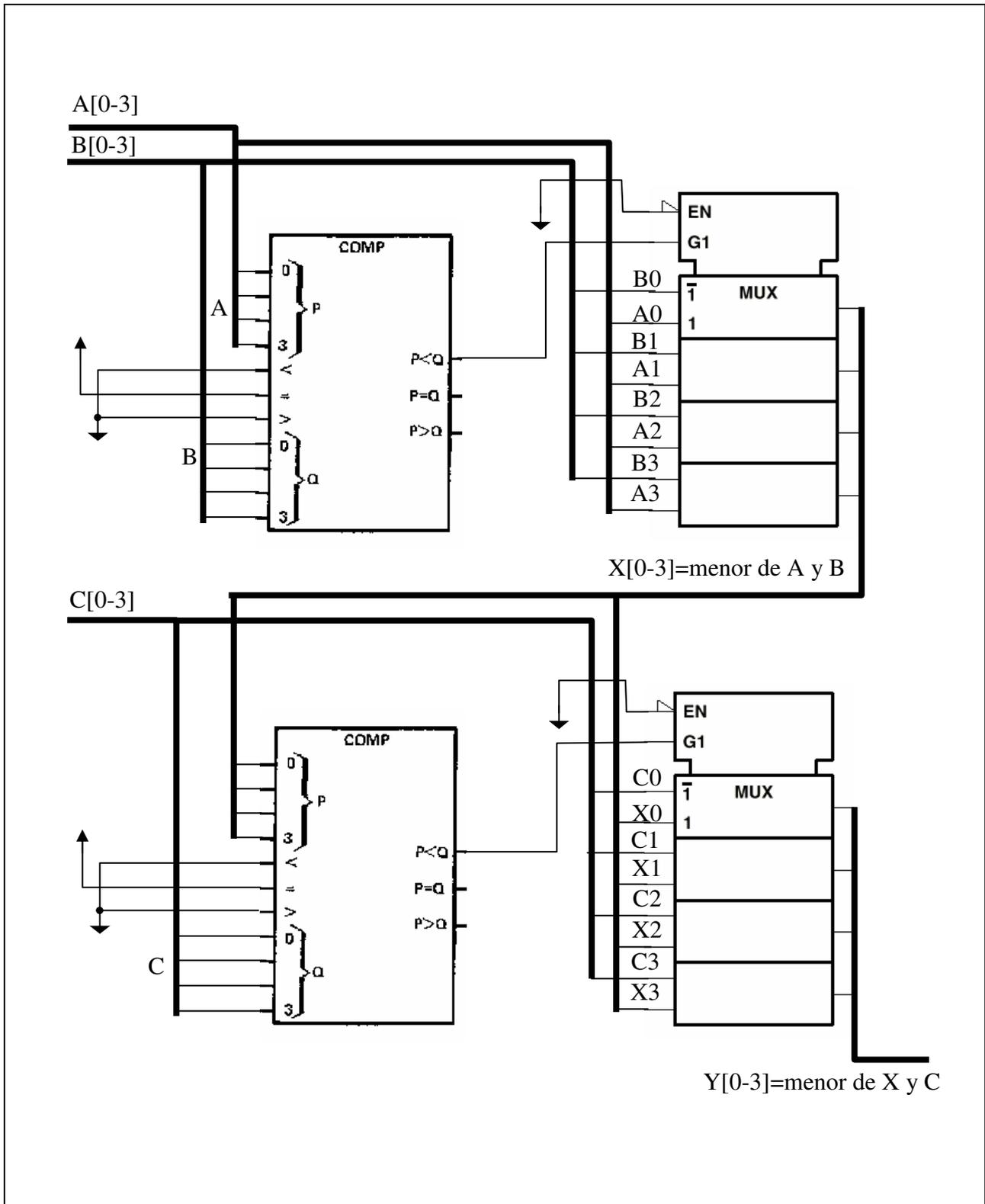
$$A: \quad 0010 \quad 1100 \quad \quad \quad y \quad \quad B: \quad 0010 \quad 0100$$

La comparación de los 4 bits menos significativos (1100 y 0100) activará la salida **A>B** del circuito de la izquierda y según las ecuaciones anteriores, al cumplirse que **n° A=n° B** (0010 y 0010), la salida que debe activarse en el segundo comparador es **A>B**. Es fácil hacer esta comprobación con todas las combinaciones posibles.

La extensión de esta construcción a comparadores de mayor número de bits es bien sencilla.



El siguiente circuito es una aplicación completa de comparadores, en la que se comparan tres números codificados en BCD (palabras de 4 bits) y se escoge el menor. Para ello se comparan dos números y se toma el menor y éste se compara con el tercero; se usa para “transmitir” el número seleccionado un MUX cuádruple de 2 a 1 líneas (se ha usado simbología según el estándar de IEEE):



Detectores y generadores de Paridad

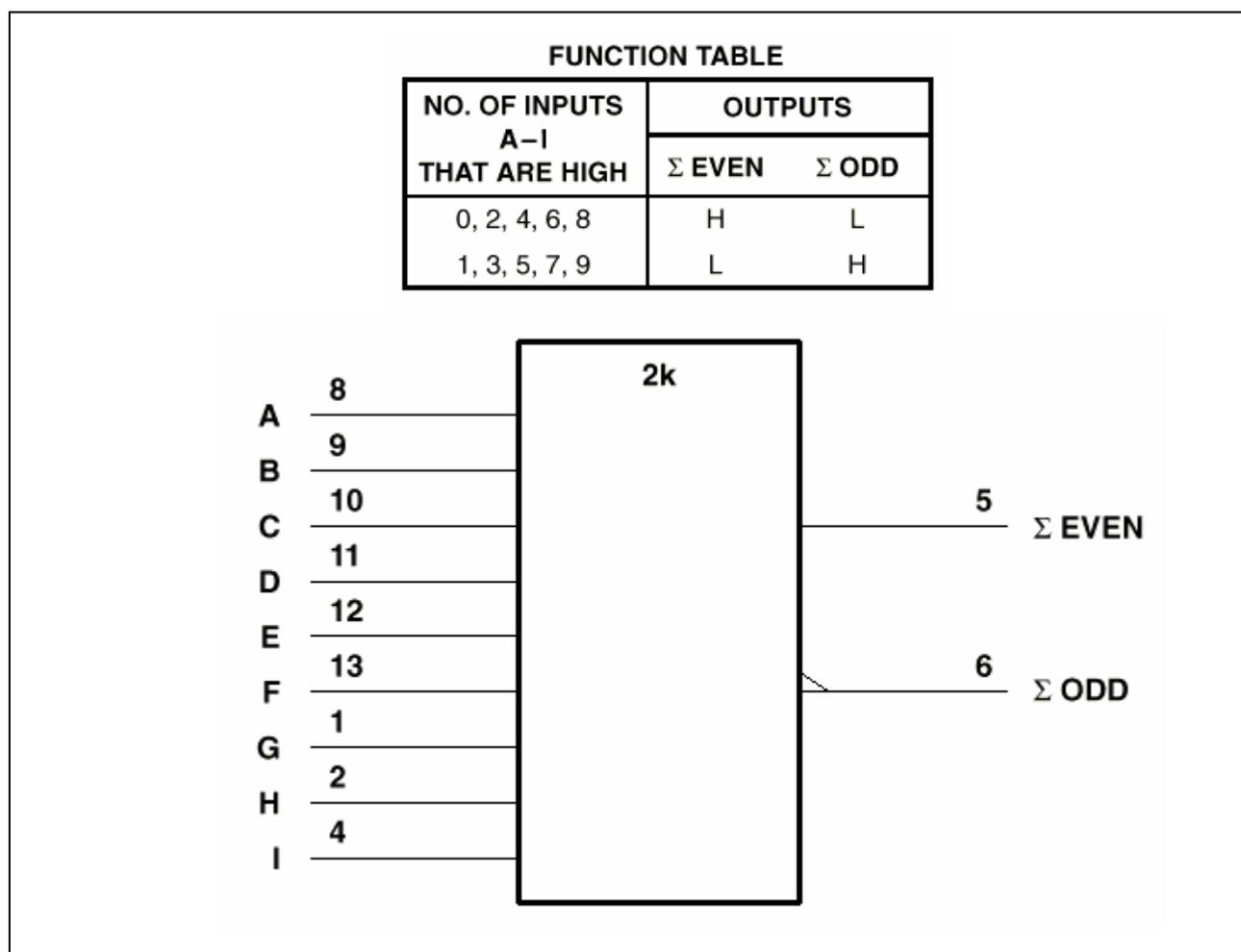
Son circuitos MSI que detectan si en la entrada hay un número par o impar de “unos”, o sea, detectan la paridad de una palabra digital. Se basan en la función EX-OR.

Su aplicación principal se basa en la transmisión y detección de códigos en las comunicaciones digitales. Un tipo de código muy usado en las transmisiones digitales es aquel que a una palabra digital le añade un bit que indique la paridad de la palabra. Cuando nuestro circuito genere el bit de paridad, funcionará como transmisor, y cuando tenga que detectarlo, funcionará como receptor. Supongamos que vamos a transmitir la palabra de 7 bits [1011110] con paridad par, el bit que debemos añadir debe ser un 1, para que el total de unos sea par.

Como Transmisor: 1011110 1 n° de unos 6

En el receptor recibimos una palabra de 8 bits [10111101] detectamos su paridad y si es par (como ocurre en este caso), admitimos la palabra como correcta.

El circuito de la figura corresponde al 74ALS280



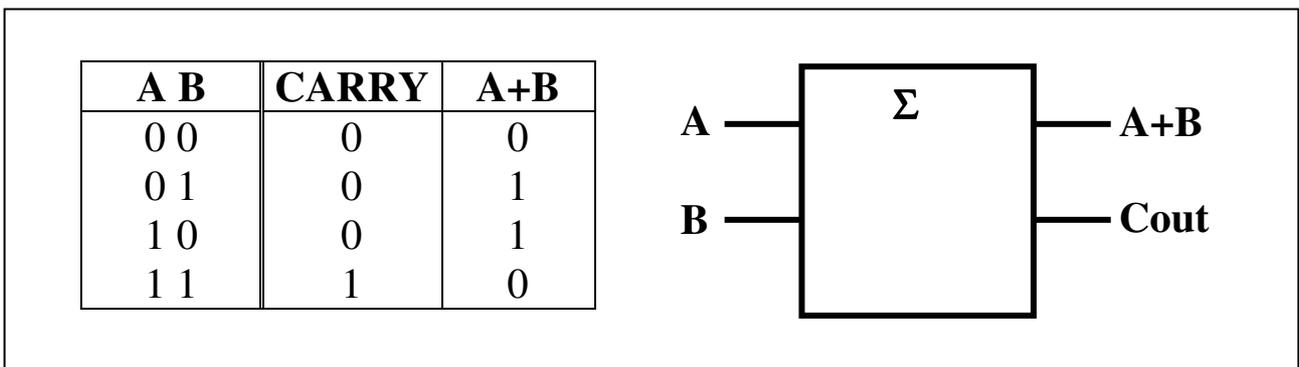
Sumadores

Un sumador es un circuito que realiza la suma de dos palabras binarias. Es distinta de la operación **OR** y con la que no nos debemos confundir. Está basada en la suma decimal que conocemos tan bien:

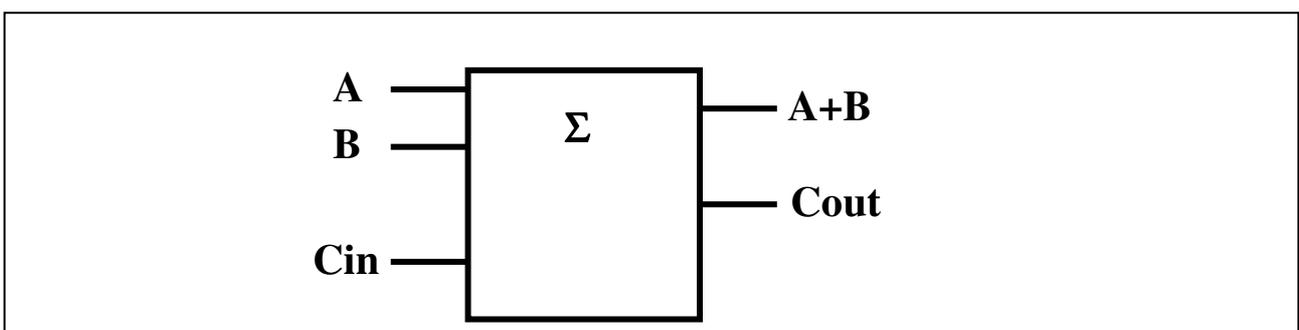
$$\begin{array}{r} 1 \\ 2 \\ \hline 3 \end{array} \quad \begin{array}{r} 01 \\ 10 \\ \hline 11 \end{array} \qquad \begin{array}{r} 2 \\ 2 \\ \hline 4 \end{array} \quad \begin{array}{r} 10 \\ 10 \\ \hline 100 \end{array}$$

Como vemos la suma de números binarios de un bit nos puede dar un número binario de tres bits. Así un sumador de dos bits debería tener 2 entradas y tres salidas. A este bit más significativo en la salida se le conoce como el **acarreo** (parecido al “me llevo una” de la suma decimal).

Asignemos una tabla de funcionamiento y un símbolo a un dispositivo que realice esta operación, al que se le suele denominar **medio sumador**:



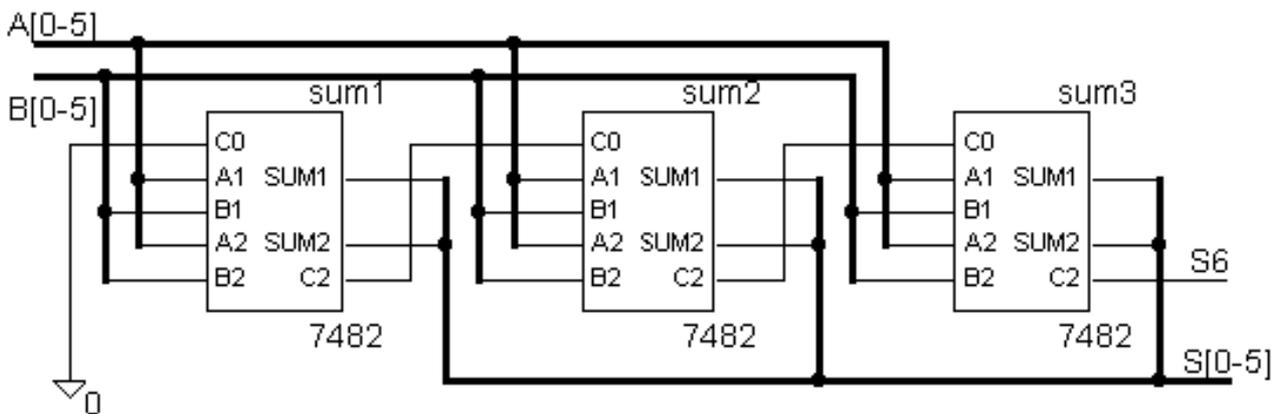
A la salida **A+B**, se le conoce también como **media suma** (HA). **Cout**, indica el acarreo de salida (carry out). Esta salida se puede usar para conectar en cascada distintos sumadores, de manera que podamos sumar palabras de más de un bit. Para ello necesitamos que el circuito disponga de una entrada extra por donde podamos informarle de la suma de los bits anteriores. A este dispositivo se le denomina **sumador completo**:



Su tabla de funcionamiento sería la siguiente:

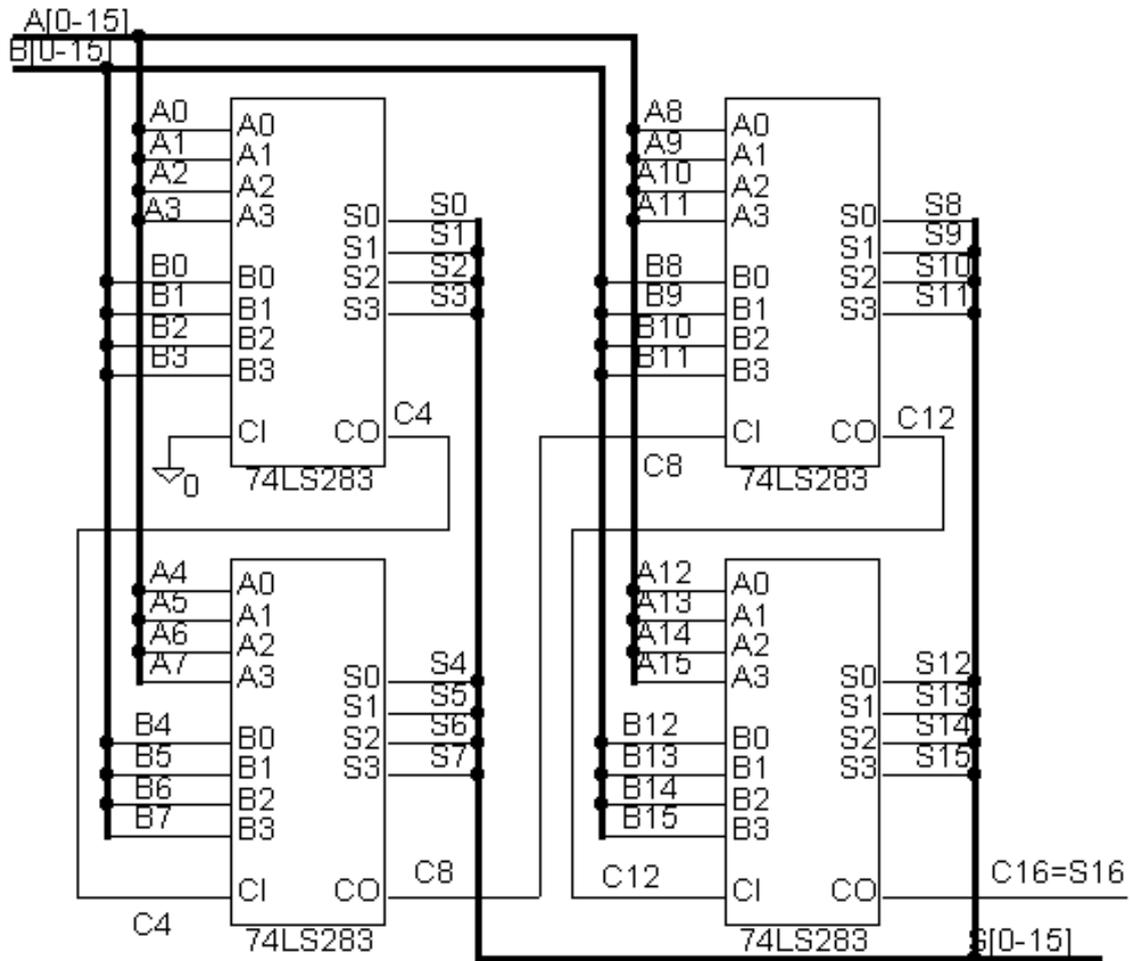
A	B	Cin	Cout	A+B
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Este sumador completo constituye el bloque básico para sumar palabras de mayor número de bits. Como ejemplo el siguiente circuito suma dos palabras de seis bits $A[0-5]$ y $B[0-5]$ para darnos una palabra de siete bits $S[0-6]$, los seis menos significativos corresponden a la media suma y el MSB es el del acarreo de salida.



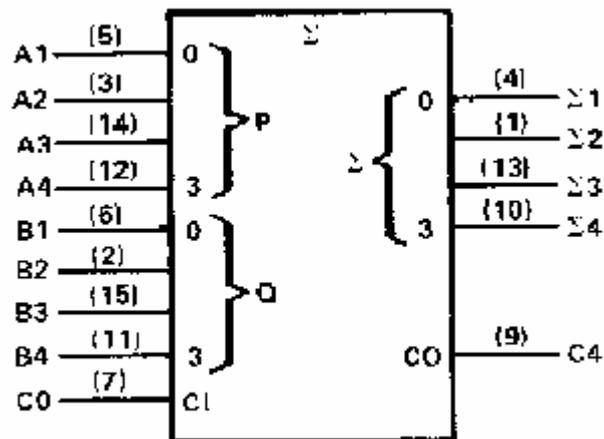
La suma que realiza este circuito se suele denominar suma con propagación de acarreo, puesto que como se ve, el acarreo se propaga de un sumador al siguiente. Aquí el acarreo se propaga en serie, pero puede propagarse en paralelo mediante un circuito de propagación del acarreo que veremos más adelante.

Este circuito suma 16 bits con propagación de acarreo serie, usando sumadores completos de 4 bits (74LS283).



Si el tiempo que tarda el circuito sumador en obtener, a partir de los datos de entrada, los valores de la salida lo denominamos τ , el tiempo que se tarda en obtener la palabra $S[0-16]$ como la suma binaria de $A[0-15]$ y $B[0-15]$, con este circuito es 4τ .

El símbolo estándar del 74LS283 es el que se tiene a continuación.



Los sumadores se pueden aplicar también a la conversión de códigos, ya que en muchas codificaciones se usa la suma de una palabra determinada para obtener otro código distinto.

Por ejemplo, para obtener una palabra del código **BCD-exceso3**, basta sumar la palabra **0011** al número **BCD-natural**.

Otros casos no son tan triviales; por ejemplo, veamos la realización de un convertidor de código BCD-natural a BCD-Aiken, usando un circuito sumador de 4 bits y puertas lógicas:

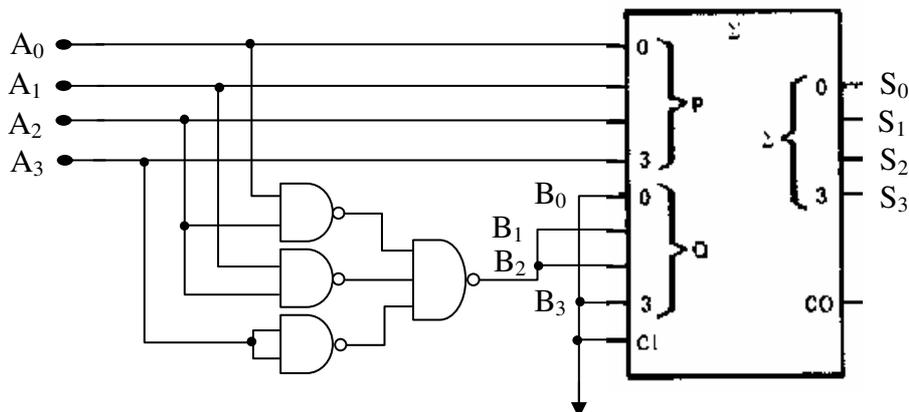
n	BCD-natural				Palabra para convertir				BCD-Aiken			
	A3	A2	A1	A0	B3	B2	B1	B0	S3	S2	S1	S0
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	0	1	0
3	0	0	1	1	0	0	0	0	0	0	1	1
4	0	1	0	0	0	0	0	0	0	0	1	0
5	0	1	0	1	0	1	1	0	1	0	1	1
6	0	1	1	0	0	1	1	0	1	1	0	0
7	0	1	1	1	0	1	1	0	1	1	0	1
8	1	0	0	0	0	1	1	0	1	1	1	0
9	1	0	0	1	0	1	1	0	1	1	1	1
X	1	0	1	0	X	X	X	X	X	X	X	X

De la tabla de funcionamiento se obtienen B_i a partir de las A_i :

$$\text{➤ } B_3 = B_0 = 0$$

$$\text{➤ } B_1 = B_2 = A_3 + A_2A_1 + A_2A_0$$

Y el circuito queda:

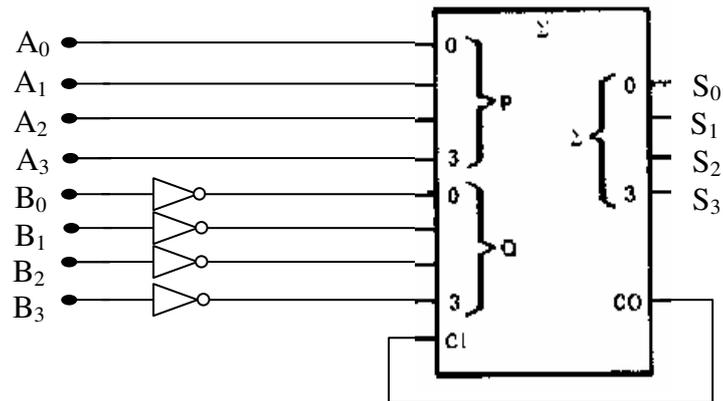


Restadores

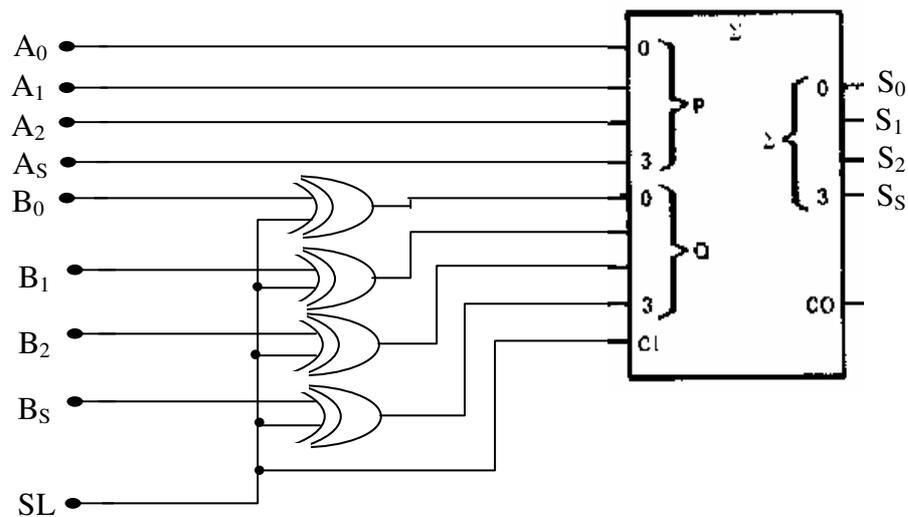
No existen en realidad, sino que se realizan mediante sumadores, ya que la resta de dos números es la suma de uno con el negativo del otro.

El negativo de un número binario se puede obtener en complemento a uno (inversión de todos los bits uno a uno); o, en complemento a dos (se añade un bit de signo).

Este circuito realiza la resta de $A[0-3]$ y $B[0-3]$ en complemento a uno:



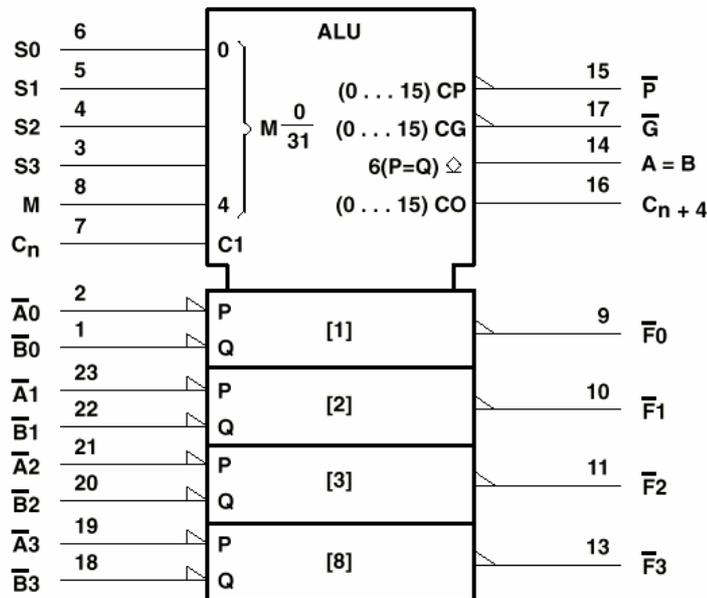
Y este otro es un sumador/restador de las palabras de 3 bits $A[0-2]$ y $B[0-2]$, en complemento a dos, nótese el uso del bit de signo. La entrada SL es 0 para obtener $A+B$ y 1 para obtener $A-B$.



ALU's

ALU son las siglas de Arithmetic Logic Unit, o sea, Unidad Lógico Aritmética. Se trata de un circuito MSI que puede realizar diferentes operaciones aritméticas y lógicas con dos palabras de n bits.

El más conocido es 74LS181, que es una ALU de 4 bits, que puede realizar hasta 32 funciones diferentes (16 lógicas y 16 aritméticas). Su símbolo lógico estándar y su tabla de funcionamiento se presentan a continuación

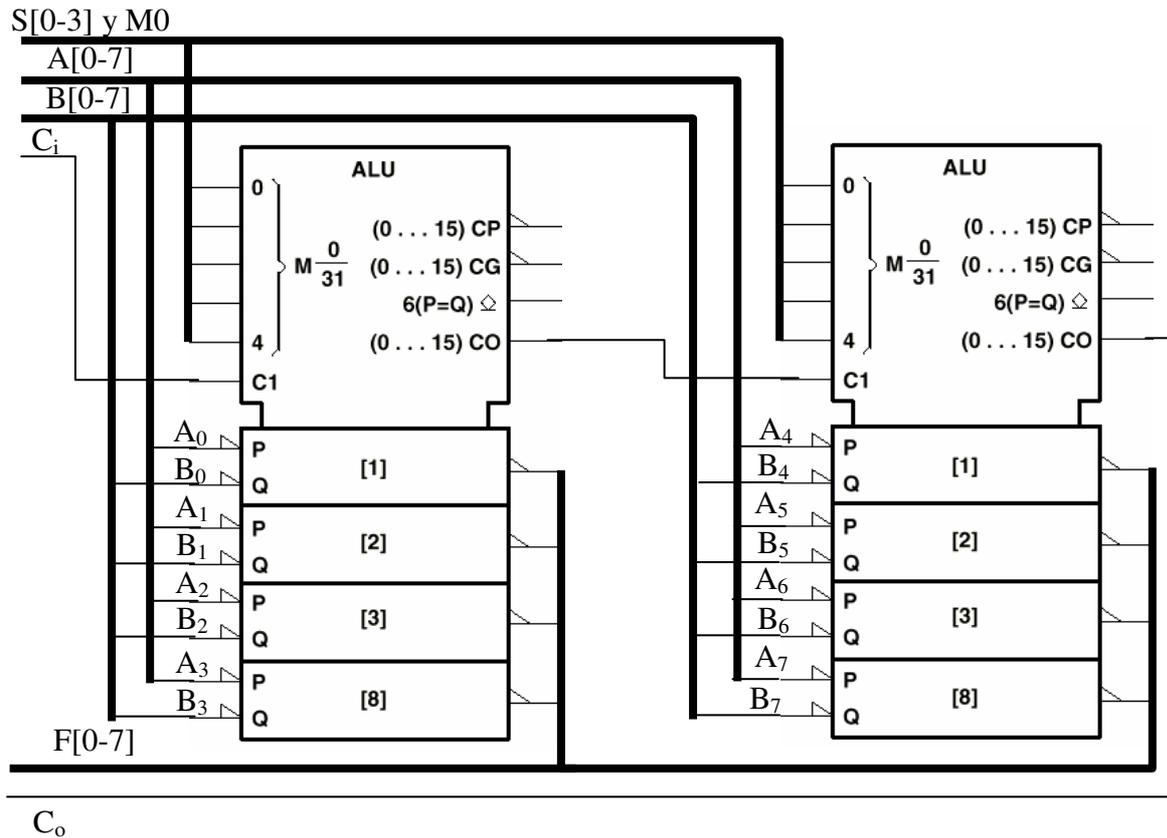


SELECTION				ACTIVE-LOW DATA		
				M = H LOGIC FUNCTIONS	M = L; ARITHMETIC OPERATIONS	
S3	S2	S1	S0		C _n = L (no carry)	C _n = H (with carry)
L	L	L	L	$F = \bar{A}$	$F = A \text{ MINUS } 1$	$F = A$
L	L	L	H	$F = \overline{AB}$	$F = AB \text{ MINUS } 1$	$F = AB$
L	L	H	L	$F = \bar{A} + B$	$F = \overline{AB} \text{ MINUS } 1$	$F = \overline{AB}$
L	L	H	H	$F = 1$	$F = \text{MINUS } 1 \text{ (2's COMP)}$	$F = \text{ZERO}$
L	H	L	L	$F = \overline{A + B}$	$F = A \text{ PLUS } (A + \bar{B})$	$F = A \text{ PLUS } (A + \bar{B}) \text{ PLUS } 1$
L	H	L	H	$F = \bar{B}$	$F = AB \text{ PLUS } (A + \bar{B})$	$F = AB \text{ PLUS } (A + \bar{B}) \text{ PLUS } 1$
L	H	H	L	$F = \overline{A \oplus B}$	$F = A \text{ MINUS } B \text{ MINUS } 1$	$F = A \text{ MINUS } B$
L	H	H	H	$F = A + \bar{B}$	$F = A + \bar{B}$	$F = (A + \bar{B}) \text{ PLUS } 1$
H	L	L	L	$F = \overline{AB}$	$F = A \text{ PLUS } (A + B)$	$F = A \text{ PLUS } (A + B) \text{ PLUS } 1$
H	L	L	H	$F = A \oplus B$	$F = A \text{ PLUS } B$	$F = A \text{ PLUS } B \text{ PLUS } 1$
H	L	H	L	$F = B$	$F = \overline{AB} \text{ PLUS } (A + B)$	$F = \overline{AB} \text{ PLUS } (A + B) \text{ PLUS } 1$
H	L	H	H	$F = A + B$	$F = (A + B)$	$F = (A + B) \text{ PLUS } 1$
H	H	L	L	$F = 0$	$F = A \text{ PLUS } A^\dagger$	$F = A \text{ PLUS } A \text{ PLUS } 1$
H	H	L	H	$F = \overline{AB}$	$F = AB \text{ PLUS } A$	$F = AB \text{ PLUS } A \text{ PLUS } 1$
H	H	H	L	$F = AB$	$F = \overline{AB} \text{ PLUS } A$	$F = \overline{AB} \text{ PLUS } A \text{ PLUS } 1$
H	H	H	H	$F = A$	$F = A \text{ PLUS } 1$	$F = A \text{ PLUS } 1$

† Each bit is shifted to the next more significant position.

La asociación de ALU's, para operar con más bits, se puede hacer en serie (acarreo serie) o en paralelo (acarreo paralelo o acarreo rápido).

Para el primero, basta usar las entradas de acarreo anterior (C1) y las salidas de acarreo serie (CO), para realizar una conexión similar a la de los sumadores:



Para que el procesamiento de los datos sea más rápido, se usa un circuito de propagación y generación de acarreo.

Para ello se usan las salidas G y P de las ALU's:

- G se activa si la ALU genera un acarreo, o sea, si se produce un acarreo saliente (CO=1), independientemente de si hay o no un acarreo entrante (C1=X).
- P se activa si la ALU propaga un acarreo, o sea, se producirá un acarreo saliente si hay un acarreo entrante.

Cuando las ALU's se conectan para procesar en paralelo (acarreo rápido) se usa un circuito Generador y Propagador de acarreo, que toma información de las ALU's, para generar y propagar su acarreo. Este circuito es el 74S182, y permite conectar hasta cuatro ALU's (procesamiento de 16 bits).

