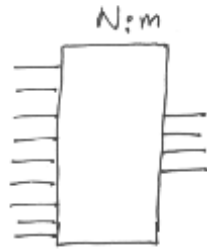
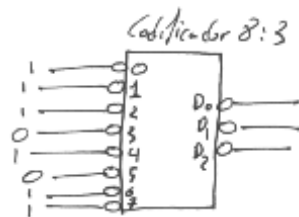


1) Un codificador es un circuito combinatorial con N entradas y n salidas ($N > n$). Cuando se activa una de las entradas en la salida aparece codificado (en binario, BCD, ...etc.) el código de la entrada activada.



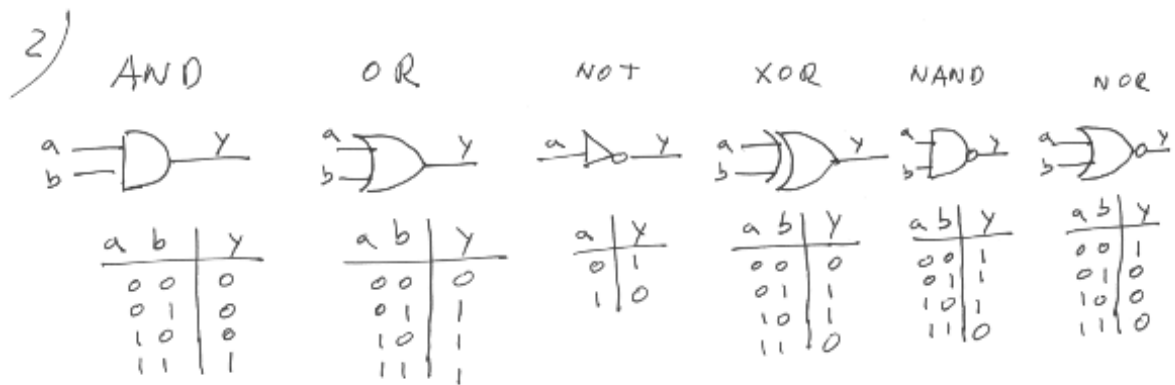
una de las entradas en la salida aparece codificado (en binario, BCD, ...etc.) el código de la entrada activada.

En un codificador sin prioridad si se activan dos o más entradas a la vez, la salida no es válida (normalmente será un OR bit a bit entre las codificaciones de las dos entradas activadas), mientras que en un



codificador prioritario se dará prioridad a la entrada de mayor peso, ignorando las demás.

Por ejemplo en un codificador 8:3, con las entradas 5 y 3 activadas (como en la figura), si el codificador es prioritario, la salida será 010 (el código del 5 en lógica negativa). Sin embargo si el codificador es no prioritario, la salida podría ser 000



3)

$$f = (cd \oplus b) + a\bar{b} = \bar{c}\bar{d} \cdot b + cd \cdot \bar{b} + a\bar{b} = (\bar{c} + \bar{d}) \cdot b + \bar{b}cd + a\bar{b} = b\bar{c} + b\bar{d} + \bar{b}cd + a\bar{b}$$

b	\bar{c}
x	10x
↓	
0	100
0	101
1	100
1	101

b	\bar{d}
x	1x0
↓	
0	100
0	110
1	100
1	110

\bar{b}	cd
x	011
↓	
0	011
1	011

a	\bar{b}
x	10xx
↓	
1	000
1	001
1	010
1	011

a	b	cd	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0
1	1	1	0

La función estará en su forma canónica cuando en cada uno de sus términos aparecen todas las variables. En una función por minterminos aparecerán tantos términos como unos hay en la tabla de verdad:

$$f(a,b,c,d) = \bar{a}\bar{b}cd + \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + \bar{a}b\bar{c}\bar{d} + a\bar{b}\bar{c}\bar{d} + a\bar{b}\bar{c}d + a\bar{b}c\bar{d} + a\bar{b}cd + ab\bar{c}\bar{d} + ab\bar{c}d + abc\bar{d} + abcd$$

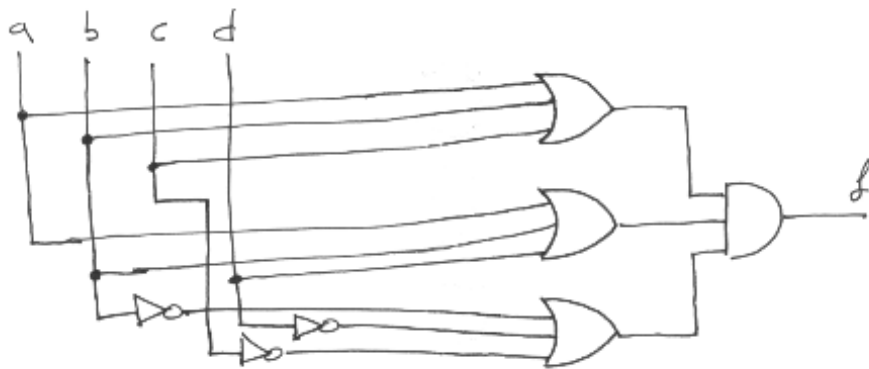
Para la realización del circuito es conveniente simplificarla primero:

ab \ cd	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	1	1	0	1
10	1	1	1	1

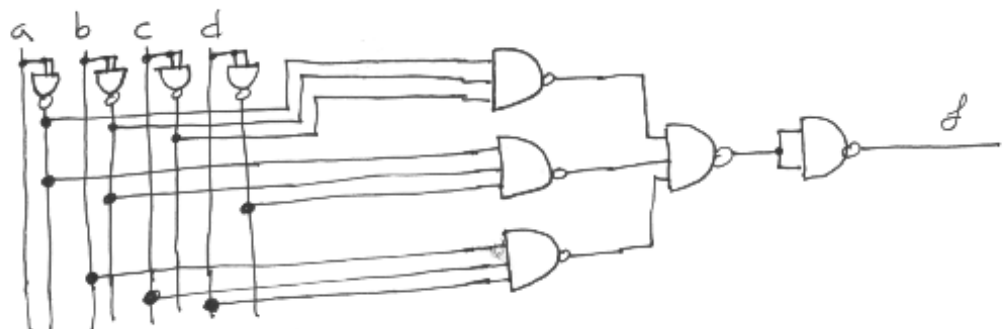
$\rightarrow (a+b+c)$ (points to the 0s in the first row)
 $\rightarrow (a+b+d)$ (points to the 0s in the third and fourth rows)
 $(\bar{b} + \bar{c} + \bar{d})$ (points to the 0s in the third and fourth columns)

$$f = (a+b+c) \cdot (a+b+d) \cdot (\bar{b} + \bar{c} + \bar{d})$$

El circuito sería:



Que, teniendo en cuenta que una puerta NAND  tiene una forma complementaria  se puede convertir en:



4) Tensiones de entrada y salida

V_{IHmin} → tensión mínima que se considera un "1" (nivel alto) válido

V_{ILmax} → tensión máxima que se considera un "0" (nivel bajo) válido

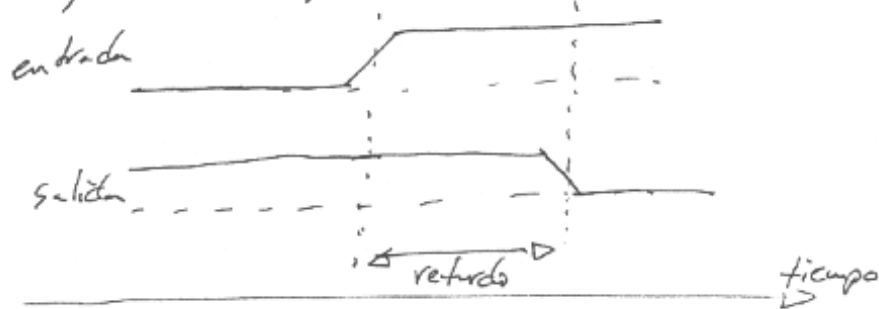
V_{OHmin} → tensión mínima que el fabricante garantiza que va a dar la puerta en caso de que su salida sea un "1"

V_{OLmax} → tensión máxima que el fabricante garantiza que va a dar la puerta a su salida en caso de que su salida sea un "0"

I_{ILmax} , I_{IHmax} corrientes máximas que absorberá la puerta por su entrada cuando esta sea un "0" (L) o un "1" (H).

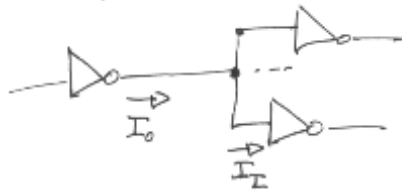
I_{OLmax} , I_{OHmax} corriente máxima que puede entregar la puerta a su salida cuando esta esté a "0" (L) o a "1" (H)

El retardo de una puerta es el tiempo que pasa desde que se produce una modificación en sus entradas hasta que esa modificación se refleja en la salida.



si la salida pasa de nivel alto a nivel bajo, ese retardo se llama t_{PHL} y si pasa de nivel bajo a nivel alto t_{PLH}

El fan-out (abánico de salida) es el número de puertas que podemos conectar a la salida de otra puerta.

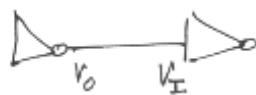


$$\text{F.O. a nivel bajo} = \frac{I_{OL\max}}{I_{IL\max}}$$

$$\text{F.O. a nivel alto} = \frac{I_{OH\max}}{I_{IH\max}}$$

y el fan-out será el valor mínimo de los dos.

El margen de ruido es el margen de error existente entre las tensiones de salida de una puerta y las de entrada de la



puerta a la que ataca.

$$NM_H = V_{OH\min} - V_{IH\min} \rightarrow \text{margen de ruido a nivel alto}$$

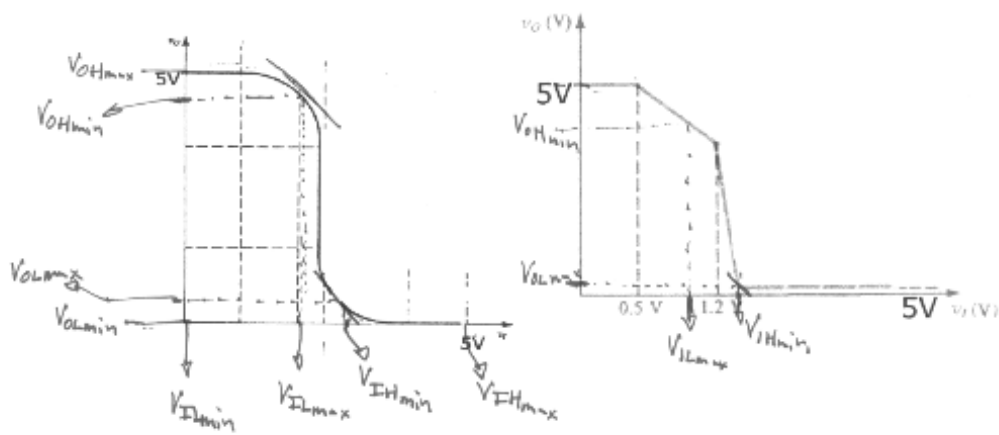
$$NM_L = V_{IH\max} - V_{OH\max} \rightarrow \text{margen de ruido a nivel bajo}$$

El margen de ruido será el menor de los dos.

I_{cc} es la corriente (media) de alimentación.

P_{cc} es el consumo (medio) de potencia.

- 5) La curva de curva de un inversor CMOS es simétrica (conmuta a $V_{cc}/2$) y mucho más abrupta que la del inversor TTL, por lo tanto la curva de la izquierda corresponde a un inversor CMOS y la de la derecha a un inversor TTL



Los márgenes de ruido también se pueden sacar de estos gráficos, a partir de V_{IH} , V_{IL} , V_{OH} , V_{OL} . El resto de parámetros pedidos (comiadas, fan-out, etc.) no es posible sacarlos de estos gráficos.

De las dos gráficos se ve que los márgenes de ruido son mayores en CMOS que en TTL, por lo que las características de CMOS (por lo que puede verse en los gráficos) son mejores.

6) Los valores absolutos de los números en binario puro son:

$$\begin{array}{r} 27 \overline{) 27} \\ \underline{1} \\ 13 \overline{) 13} \\ \underline{1} \\ 6 \overline{) 6} \\ \underline{0} \\ 3 \overline{) 3} \\ \underline{1} \\ 1 \end{array}$$

$$27 = 11011$$

Se necesitarán 6 bits

$$\begin{array}{r} 24 \overline{) 24} \\ \underline{0} \\ 12 \overline{) 12} \\ \underline{0} \\ 6 \overline{) 6} \\ \underline{0} \\ 3 \overline{) 3} \\ \underline{1} \\ 1 \end{array}$$

$$24 = 11000$$

Se necesitarán 6 bits

$$\begin{array}{r} 39 \overline{) 39} \\ \underline{1} \\ 19 \overline{) 19} \\ \underline{1} \\ 9 \overline{) 9} \\ \underline{1} \\ 4 \overline{) 4} \\ \underline{0} \\ 2 \overline{) 2} \\ \underline{0} \\ 1 \end{array}$$

$$39 = 100111$$

Se necesitarán 7 bits

$$\begin{array}{r} 85 \overline{) 85} \\ \underline{1} \\ 42 \overline{) 42} \\ \underline{0} \\ 21 \overline{) 21} \\ \underline{0} \\ 10 \overline{) 10} \\ \underline{0} \\ 5 \overline{) 5} \\ \underline{1} \\ 2 \overline{) 2} \\ \underline{0} \\ 1 \end{array}$$

$$85 = 1010101$$

Se necesitarán 8 bits

Los números positivos se representan igual en C-2 que en binario puro:

$$+24 = 011000$$

$$+39 = 0100111$$

Los números negativos se construyen a partir del número positivo complementando a 2.

$$+27 = 011011$$

$$-27_{C-1} = 100100$$

$$+1$$

$$-27_{C-2} = 100101$$

$$+85 = 01010101$$

$$-85_{C-1} = 10101010$$

$$+1$$

$$-85_{C-2} = 10101011$$

En octal y hexadecimal

$$+24_{10} = \underbrace{011}_2 \underbrace{000}_2 = 30_8 = 18_{16}$$

$$+39_{10} = \underbrace{010}_2 \underbrace{011}_2 = 47_8 = 27_{16}$$

Para representar 12.0054 en coma flotante, convertimos a binario por un lado la parte entera y por otro la parte fraccionaria

$$\begin{array}{r} 12 \overline{) 12} \\ 9 \quad 6 \overline{) 6} \\ \quad 0 \quad 3 \overline{) 3} \\ \quad \quad 1 \quad 1 \end{array}$$

$$12 = 1100$$

0.0054	x 2	0
0.0108	x 2	0
0.0216	x 2	0
0.0432	x 2	0
0.0864	x 2	0
0.1728	x 2	0
0.3456	x 2	0
0.6912	x 2	1
0.3824	x 2	0
0.7648	x 2	1

0.5296	x 2	1
0.0592	x 2	0
0.1184	x 2	0
0.2368	x 2	0
0.4736	x 2	0
0.9472	x 2	1
0.8944	x 2	1
0.7888	x 2	1
0.5776	x 2	1
0.1552	x 2	0
0.3104	x 2	0

$$\begin{array}{r|l}
 0.6208 \times 2 & 1 \\
 0.2416 \times 2 & 0 \\
 0.4832 \times 2 & 0 \\
 0.9664 \times 2 & 1 \\
 0.9328 \times 2 & 1 \\
 0.8656 & \\
 \dots &
 \end{array}$$

$$12,0054 = 1100,0000001011000011110010011 \dots$$

Pasado a forma normalizada queda:

$$1,10000000001011000011110010011 \cdot 2^3$$

De donde:

$$\text{Exponente} = 3, \text{ que en exceso } 127 \text{ es } 130_{10} = 10000010_2$$

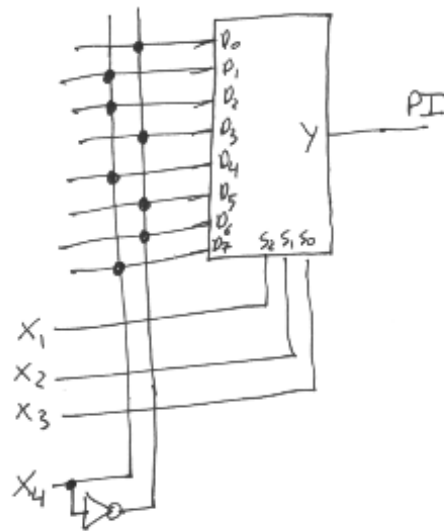
$$\text{Signo} = 0 \text{ (positivo).}$$

$$\text{Mantisa} = 10000000001011000011110$$

$$12,0054 = \begin{array}{c}
 \begin{array}{|c|c|c|}
 \hline
 S & E & M \\
 \hline
 0 & 10000010 & 10000000001011000011110 \\
 \hline
 \end{array}
 \end{array}$$

7) El criterio de paridad impar es que el número total de unos de un número (incluido el bit de paridad) sea impar.

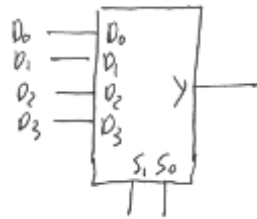
x_1	x_2	x_3	x_4	PI	
0	0	0	0	1) \bar{x}_4
0	0	0	1	0	
0	0	1	0	0) x_4
0	0	1	1	1	
0	1	0	0	0) x_4
0	1	0	1	1	
0	1	1	0	1) \bar{x}_4
0	1	1	1	0	
1	0	0	0	0) x_4
1	0	0	1	1	
1	0	1	0	1) \bar{x}_4
1	0	1	1	0	
1	1	0	0	1) \bar{x}_4
1	1	0	1	0	
1	1	1	0	0) x_4
1	1	1	1	1	



8) Un multiplexor es un circuito combinatorial con N entradas, 1 salida y n líneas de selección, donde $N = 2^n$. Las líneas de selección permiten seleccionar cual de los datos de las entradas de datos pasará a la salida.

Los multiplexores pueden utilizarse como selectores de datos, para generar funciones lógicas o para multiplexar señales en el tiempo.

Un multiplexor 4:1 es un circuito como este:



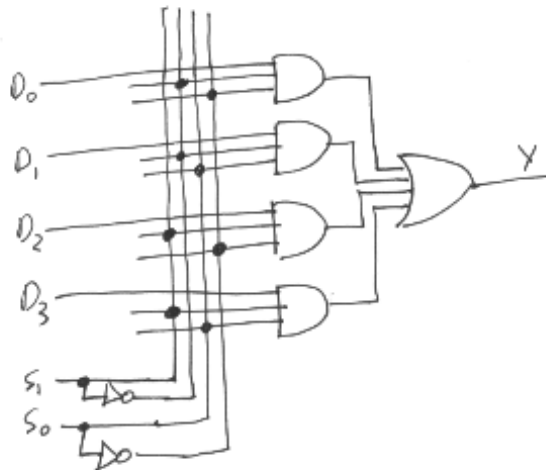
La tabl. de verdad ser :

D_0	D_1	D_2	D_3	S_1	S_0	Y
0	x	x	x	0	0	0
1	x	x	x	0	1	0
x	0	x	x	1	0	0
x	1	x	x	1	1	0
x	x	0	x	0	0	1
x	x	1	x	0	1	1
x	x	x	0	1	0	1
x	x	x	1	1	1	1

De donde se puede sacar:

$$Y = D_0 \bar{S}_1 \bar{S}_0 + D_1 \bar{S}_1 S_0 + D_2 S_1 \bar{S}_0 + D_3 S_1 S_0$$

Y el circuito ser :



El retardo de propagaci n desde una entrada D hasta la salida Y es de dos puertas, esto es 40ns. El retardo de propagaci n desde una entrada de selecci n S hasta la salida Y es, en el peor de los casos de 3 puertas, esto es 60ns.

9) Un sumador es un circuito que tiene dos entradas de N bits ($A+B$) y una salida de N bits (suma) más una salida extra de acarreo, que se activa cuando la suma de los dos números tiene un bit más.

El sumador completo de un bit será el siguiente circuito



donde A y B son dos sumandos
 C_i es el acarreo de entrada (el tercer sumando)

S es el resultado de la suma

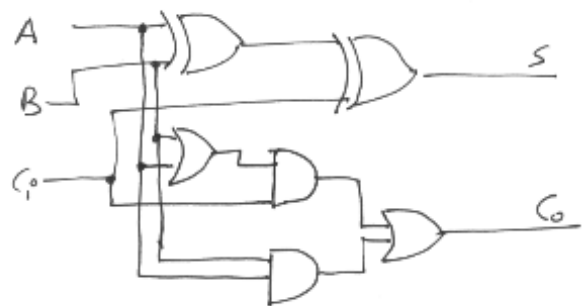
C_o es el acarreo de salida (un 2º bit o la suma, en este caso el de mayor peso,

La tabla de verdad del sumador completo de 1 bit es:

A	B	C_i	S	C_o
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

De donde $S = A \oplus B \oplus C_i$

$$C_o = AB + (A+B) \cdot C_i =$$

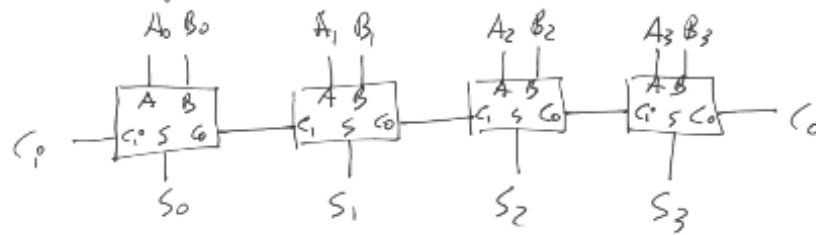


Un sumador completo de 4 bits será:



Hay dos formas de realizarla a partir de sumadores de 1 bit:

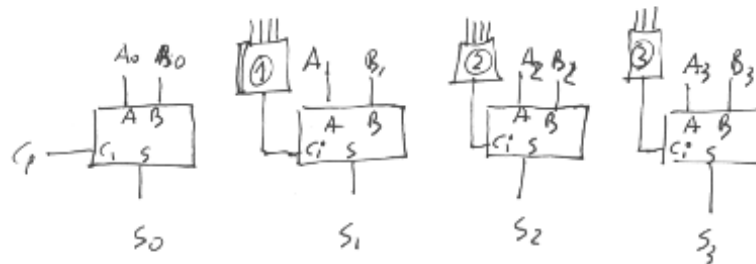
- Sumador paralelo con acarreo serie:



El circuito es muy sencillo, pero al propagarse el acarreo en serie cada sumador de 1 bit tiene que esperar a que el anterior termine de sumar, por lo tanto el retardo total del sumador = retardo del sumador de 1 bit \times número de bits.

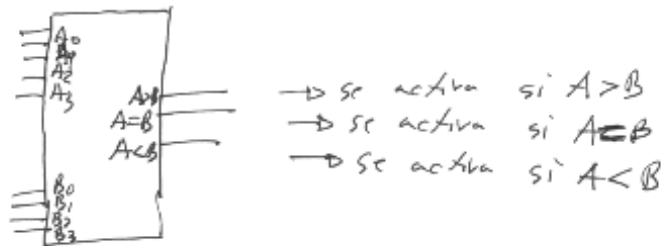
- Sumador paralelo con acarreo paralelo (sumador rápido)

Se basa en calcular el acarreo de entrada de cada una de las etapas sin esperar a que las anteriores hagan la suma. Básicamente será:

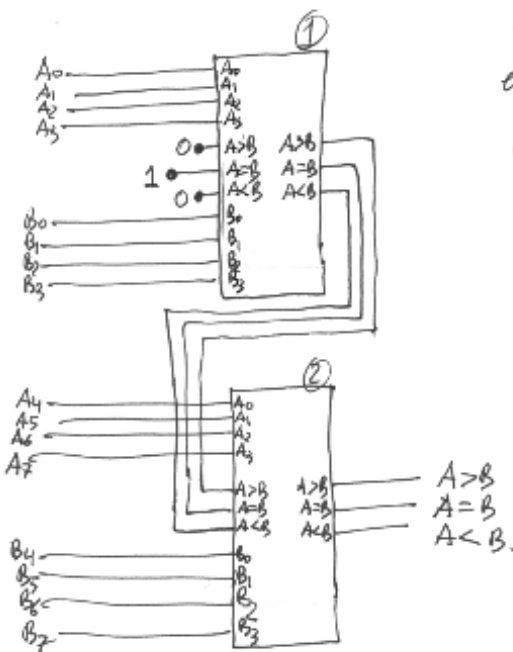


En cada uno de los bloques ①, ② y ③ entran las entradas A_i , B_i , los sumadores anteriores y el acarreo de entrada del circuito. La suma se hace de una forma más rápida pero el circuito es más complejo.

10) Un comparador realiza la comparación entre dos números de n bits e indica si son iguales, y si no lo son cual de los dos es mayor.



Los comparadores suelen tener unas entradas para conexión en cascada que permiten su conexión para realizar comparaciones de números de más bits, por ejemplo:



El comparador ② hace la comparación entre los bits de mayor peso. Si en esos bits $A > B$ o $A < B$ activa la salida correspondiente, pero si en los bits de mayor peso $A = B$, el comparador dará en su salida lo que le entre por la entrada de conexión en cascada, permitiendo al comparador ① que compare los bits de menor peso.