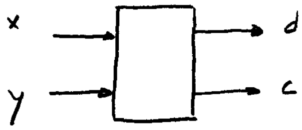
Problema 4.1

Semirestador



x	y	c	d
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$d = x - y$

$c = \bar{x} y \qquad d = x \oplus y$

Problema 4.2

Hacer un SSB - SRB seleccionado por una patilla de control

$$C \Rightarrow \begin{cases} c = 0 \Rightarrow SSB \\ c = 1 \Rightarrow SRB \end{cases}$$

$$SSB \begin{cases} c = xy \\ S = x \oplus y \end{cases} \qquad SRB \begin{cases} c = \bar{x} y \\ d = x \oplus y \end{cases}$$

$\Rightarrow$ la diferencia está en $C \Rightarrow$ que SSB se toma x en SRB se toma $\bar{x}$

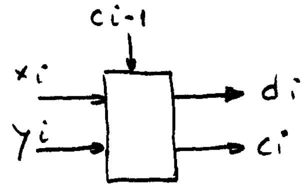
x	C	X
x	0	x
x	1	$\bar{x}$

$\Rightarrow$

x	C	X
0	0	0
1	0	1
0	1	1
1	1	0

$\Rightarrow \underline{\underline{\bar{x} = x \oplus C}}$

Problema 4.3

Restador completo (RBC) ⇒ 

x_i	y_i	c_{i-1}	d_i	c_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

d_i

	$\bar{c}_{i-1}$		c_{i-1}	
	$\bar{x}$	x	x	$\bar{x}$
$\bar{y}$	0	1	0	1
y	1	0	1	0

$$d_i = \bar{x} y \bar{c}_{i-1} + x \bar{y} \bar{c}_{i-1} + x y c_{i-1} + \bar{x}\bar{y} c_{i-1}$$

$$d_i = \bar{c}_{i-1}(x \oplus y) + c_{i-1}(x \odot y)$$

$$d_i = c_{i-1} \oplus x \oplus y$$

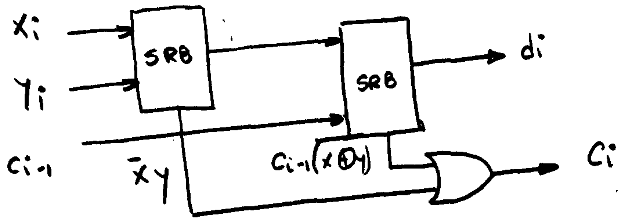
c_i

	$\bar{c}_{i-1}$		c_{i-1}	
	$\bar{x}$	x	x	$\bar{x}$
$\bar{y}$	0	0	0	1
y	1	0	1	1

$$c_i = c_{i-1} x + c_{i-1} y + \bar{x} y$$

$$c_i = \bar{x}\bar{y} c_{i-1} + \bar{x} y \bar{c}_{i-1} + \bar{x} y c_{i-1} + x y c_{i-1}$$

$$c_i = \bar{x} y + c_{i-1} \overline{(x \oplus y)}$$

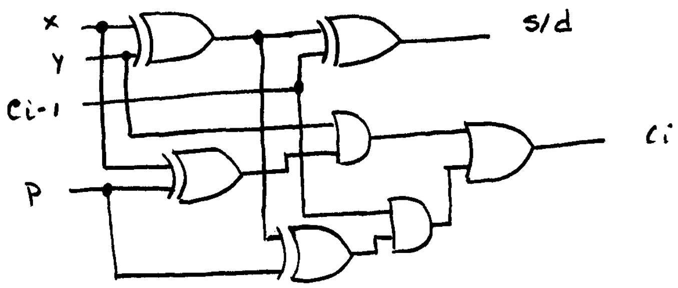


Problema 4.4

SBC - SRC con control p ⇒ $\begin{cases} p=0 \rightarrow SBC \\ p=1 \rightarrow SRC \end{cases}$

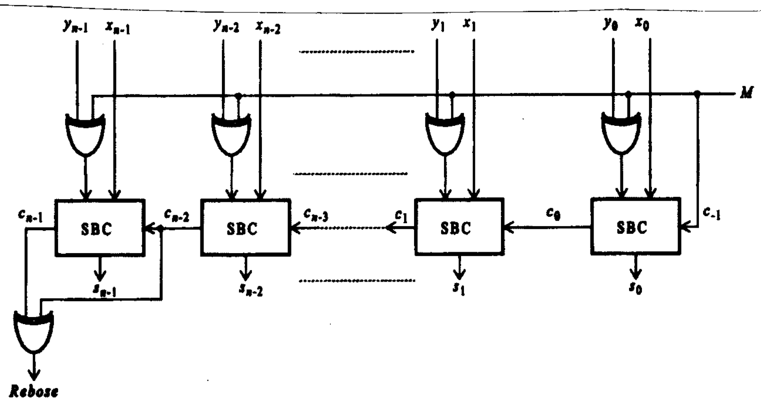
$$SBC \Rightarrow \begin{cases} s = c_{i-1} \oplus x_i \oplus y_i \\ c_i = xy + c_{i-1}(x \oplus y) \end{cases} \qquad SRC \Rightarrow \begin{cases} d = c_{i-1} \oplus x_i \oplus y_i \\ c_i = \bar{x} y + c_{i-1}\overline{(x \oplus y)} \end{cases}$$

$d = s$ / c_i { SRC invierte x y $x \oplus y$



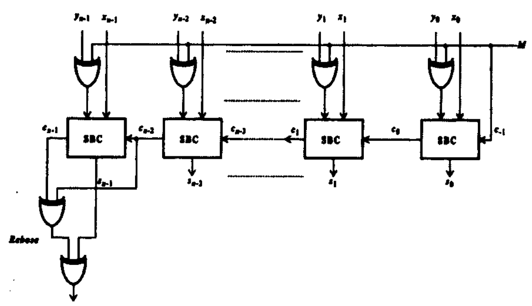
Problema 4.5

Completar el sumador-restador con un circuito que corrija el signo de la suma cuando se produce erróneamente.



$$\text{Rebose} \Rightarrow c_{n-1} \oplus c_{n-2}$$

$$\left.\begin{array}{l} s_{n-1}=1 \ \text{ y } \ \text{rebose}=1 \Rightarrow s=0 \\ s_{n-1}=0 \ \text{ y } \ \text{rebose}=1 \Rightarrow s=1 \\ s_{n-1}=0 \ \text{ y } \ \text{rebose}=0 \Rightarrow s=0 \\ s_{n-1}=1 \ \text{ y } \ \text{rebose}=0 \Rightarrow s=1 \end{array}\right\} \Rightarrow s = \text{Rebose} \oplus s_{n-1}$$



$s_c = s_{n-1} \oplus$ Rebose (s_c = signo corregido)

Problema 4.9

Determinar en las sumas siguientes:

1) Nº arrastres que comienzan simultáneamente

2) La secuencia de arrastre de mayor longitud

A)

```
  2      2     (4)  1 1
1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 1
0 1 0 0 1 1 1 1 0 1 0 0 1 0 1 1
-------------------------------
```

1) Comienzan simultáneamente ----→ 5

2) secuencia mayor longitud → (4)

Aquí no hay secuencia pq cuando llega el arrastre a -- 1 0 desde -- 1 1 ya anteriormente 1 + 1 era ∅ y había propagado el arrastre; cuando le llega un arrastre desde el anterior $+\,{}^{0}_{1}$ ⟶ ya no genera arrastre

B)

```
1 1        7         1 1    1 1
0 1  0 0 1 0 0 1 0 1 0 1 1 0 1 1
0 1  0 1 0 1 1 0 1 1 0 1 0 0 1 0
--------------------------------
```

4

(7)

Problema 4.12

Retardo de un sumador binario de n bits construido con k módulos SAD de m bits ($n = k \times m$ bits)

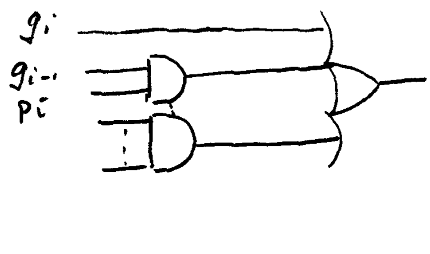
Supuesto T = retardo de una puerta lógica

① → $p_i = x_i \oplus y_i$ $g_i = x_i \cdot y_i$ ⟹ retardo de p_i y $g_i = T$

② → Propagación de arrastres ⟹ k módulos CAA ⟹

⟹ $c_i = g_i + p_i g_{i-1} + \dots p_i p_{i-1} \dots p_0 c_{i-1}$ ⟹

Suma de puertas AND



⟹ Cascada de 2 puertas

$\text{prog} = 2T$

k módulos ⟹ $2kT$

③ → retardo para s_{n-1} ⇒ $s_i = p_i \oplus c_{i-1}$ ⇒ T

Retardo total ⇒ suma de retardos

↳ $T_{①} + 2 * T_{②} + T_{③} = T(2+2K) =$

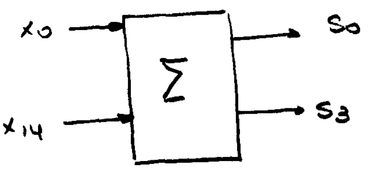
$= 2T(1+K)$

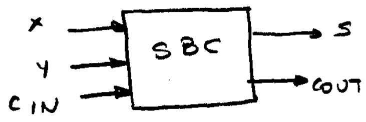
⇓

El retardo depende solo del nº de módulos SAA y no del tamaño de estos.

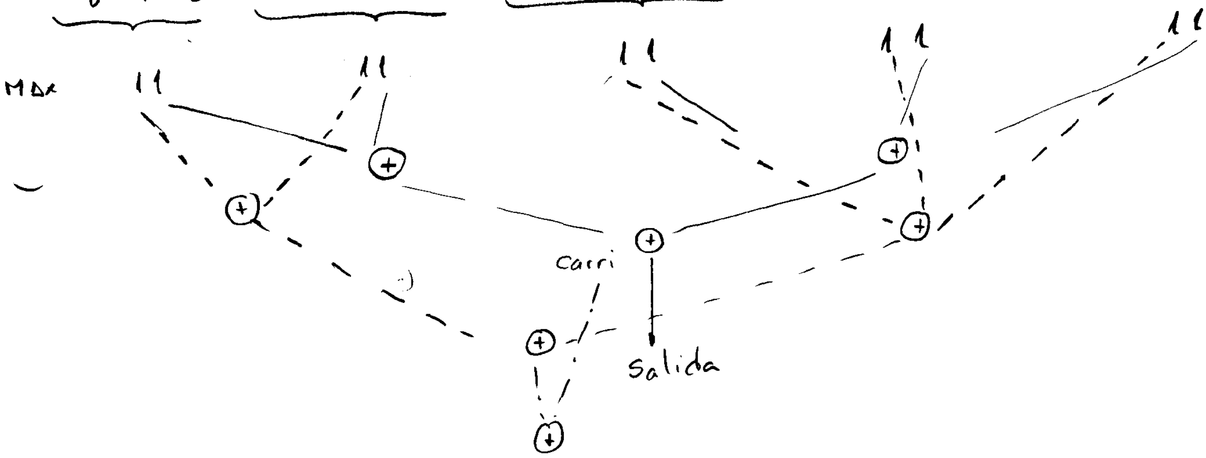
Problema 4.13

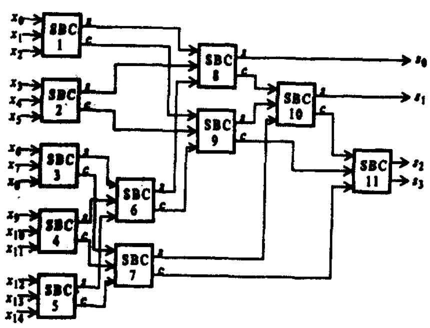
Sumador de 15 nºs 1 bit con módulos SBC. ⇒ Resultado ≤ 15
⇓
1111
⇓
4 bit salida





$\underbrace{x_0+x_1+x_2}+\underbrace{x_3+x_4+x_5}+\underbrace{x_6+x_7+x_8}+\underbrace{x_9+x_{10}+x_{11}}+\underbrace{x_{12}+x_{13}+x_{14}}$





Problema 4.14

Demostrar, que un sumador BCD debe sumar 6 para corregir el resultado.

- La suma de dos nºs binarios BCD (4 bits)

Cada nº menor de 10 ⇒ del 0000 al 1001 lo que implica que del último nº BCD 1001 (9) hasta el final de representación con 4 bit 1111 (15) faltan 6, por lo que al sumar 2 BCD si el resultado pasa de 9 habrá que sumar 6.

$5 + 5 = 10 \Rightarrow$

$$\begin{array}{rr} & 0101 \\ & 0101 \\ \hline & 1010 \\ + & 0110 \;(6) \\ \hline & \underbrace{1}_{1}\,\underbrace{0000}_{0} \end{array}$$

$5 + 6 = 11 \Rightarrow$

$$\begin{array}{rr} & 0101 \\ & 0110 \\ \hline & 1011 \\ + & 0110 \;(6) \\ \hline & \underbrace{1}_{1}\,\underbrace{0001}_{1} \end{array}$$

$5 + 7 =$

$$\begin{array}{rr} & 0101 \\ & 0111 \\ \hline & 1100 \\ + & 0110 \\ \hline & \underbrace{1}_{1}\,\underbrace{0010}_{2} \end{array}$$

... $8 + 8 \Rightarrow$

$$\begin{array}{rr} & 1000 \\ & 1000 \\ \hline & 10000 \\ + & 0110 \\ \hline & \underbrace{1}_{1}\,\underbrace{0110}_{6} \end{array}$$

... $9 + 9 \Rightarrow$

$$\begin{array}{rr} & 1001 \\ & 1001 \\ \hline & 10010 \\ + & 0110 \\ \hline & \underbrace{1}_{1}\,\underbrace{1000}_{8} \end{array}$$

Problema 4.17

Demostrar que $x \cdot y$ con n bits en base B tiene menos de $2n$ dígitos.

Nº máximo $= B^n - 1$

$$N^o \max \cdot N^o \max = (B^n - 1)(B^n - 1) = B^{2n} - B^n - B^n + 1 =$$

$$= B^{2n} - 2B^n + 1$$

Con $B^{2n} - 1 \Rightarrow 2n$ dígitos

$B^{2n} - 2B^n + 1 < B^{2n} - 1 \Rightarrow x \cdot y$ siempre menos $2n$ dígitos

Problema 4.18

Diseñar multiplicador binario de 2 x 2 bits con 4 puertas AND y 2 SBB.

$$\begin{array}{cccc}
 & & M_1 & M_0 \\
 & & m_1 & m_0 \\
\hline
 & & p_{01} & p_{00} \\
 & p_{11} & p_{10} & \\
\hline
P_3 & P_2 & P_1 & P_0
\end{array}$$

$p_{00} = m_0 \cdot M_0$ $\quad$ $p_{10} = m_1 M_0$

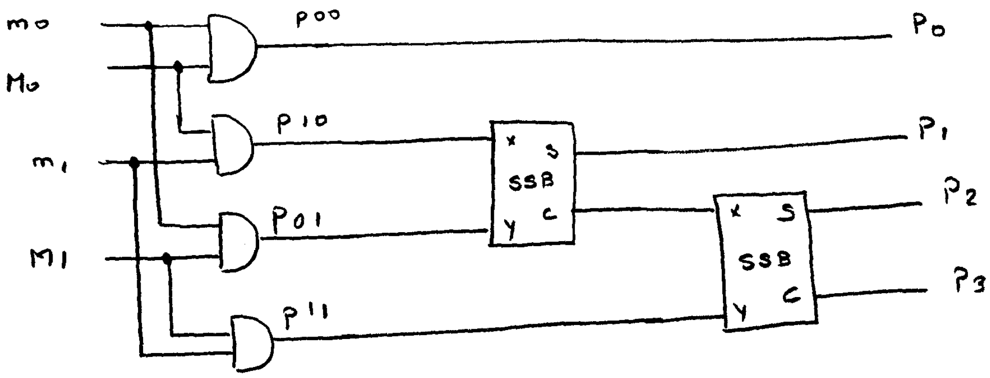
$p_{01} = m_0 \cdot M_1$ $\quad$ $p_{11} = m_1 M_1$

$P_0 = p_{00}$

$P_1 = p_{01} + p_{10}$

$P_2 = p_{11}$ + carry de P_1

P_3 = carry de P_2



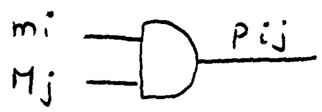
Problema 4.19

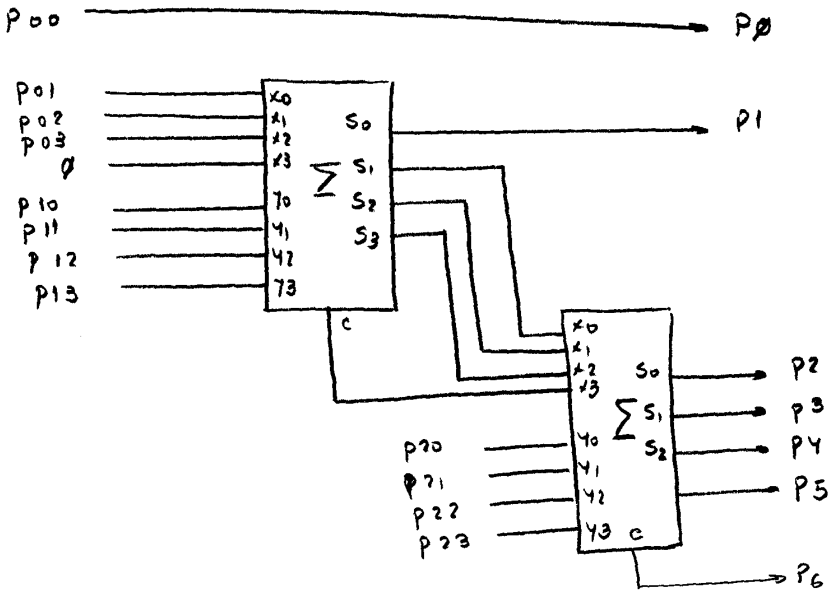
Multiplicador de 4 x 3 bit con 12 AND y 2 sumadores bin. paralelo de 4 bits cada uno

$$\begin{array}{ccccccc}
 & & & M_3 & M_2 & M_1 & M_0 \\
 & & & & m_2 & m_1 & m_0 \\
\hline
 & & & p_{03} & p_{02} & p_{01} & p_{00} \\
 & & p_{13} & p_{12} & p_{11} & p_{10} & \\
 & p_{23} & p_{22} & p_{21} & p_{20} & & \\
\hline
P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
\end{array}$$

$p_{ij} = m_i \cdot M_j \Rightarrow 12$ AND

Sum 4 bit





Problema 4.20

Diseñar un multiplicador de 4×8 bits, utilizando mem ROM de 256×8 y dos sumadores binarios paralelos de 4 bits

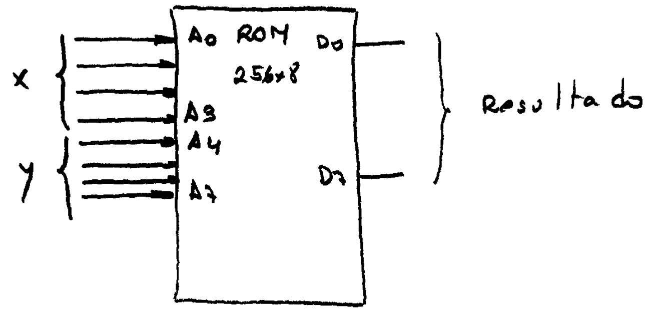
Multiplicador de 4×4 bit

En base a conformar con los (4+4) bits la dirección de memoria en la que se deberá escribir el resultado de la multiplicación correspondiente:

	x y		Dir.	contenido	
0×0=0 →	0000 × 0000	⇒	0000 0000	00000000	(0)
0×1=0 →	0000 × 0001	⇒	0000 0001	00000000	(0)
2×3=6 →	0010 × 0011	⇒	0010 0011	00000110	(6)
9×15= 135 →	1001 × 1111	⇒	1001 1111	10000111	(135)

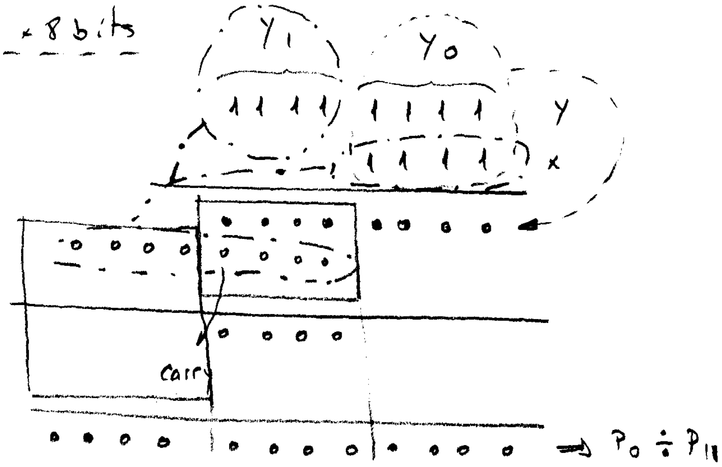
⇓

10000111



Multiplicar 4 x 8 bits

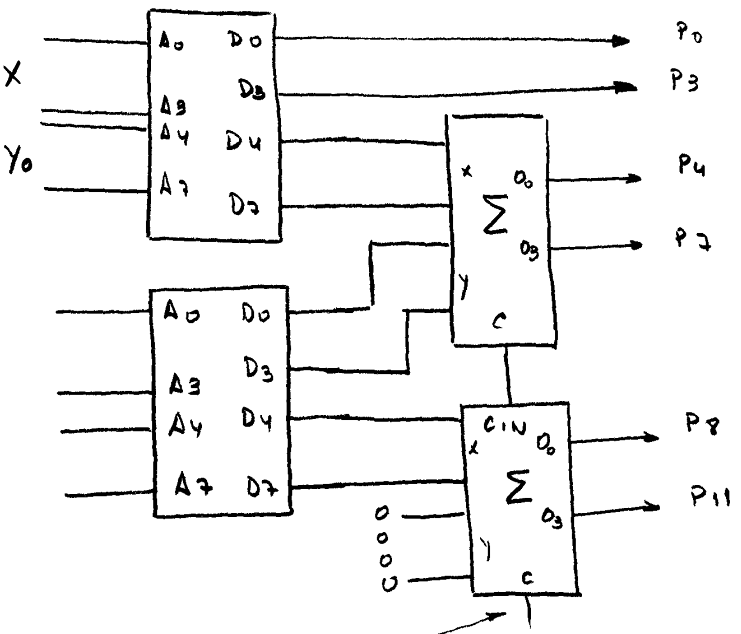
$$\begin{array}{r} y_1 \; y_0 \\ x \\ \hline x y_0 \\ x y_1 \quad \\ \hline x y_1 \;\; x y_0 \end{array}$$



$x Y_0 \Rightarrow$ 1 ROM

$x Y_1 \Rightarrow$ 1 ROM

- 1 sumador para parte alta de $x y_0$ + baja de $x y_1$
- 1 sumador para parte alta de $x y_1$ + carry de suma anterior



No lleva P_{12} pq el c nunca puede ser 1 ya que multiplicar 2 nos de 4 x 8 bits siempre 12 bits o menos.

PR. ALU 9

Problema 4.21

Mostrar paso a paso el proceso de multiplicación con el algoritmo de "lápiz y papel".

$(+15) \times (+13) \Rightarrow M \times m \Rightarrow 01111 \times 01101$

C	A	m	
0	00000	01101	← Inicial
0	01111	01101	A ← A+M
0	00111	10110	
0	00011	11011	Desplaz.
0	10010	11011	A ← A+M
0	01001	01101	
0	11000	01101	A ← A+M
0	01100	00110	
0	00110	00011	

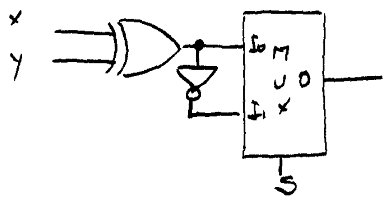
$\boxed{0\ 00110\ 00011}$ ⇓ 195

$13 \times 15 = 195$

$$\begin{array}{r} 00011 \\ 1111 \\ \hline 10010 \end{array} \qquad \begin{array}{r} 01001 \\ 1111 \\ \hline 11000 \end{array}$$

Problema 4.29

- ALU para una XOR y XNOR



$S=0 \Rightarrow$ XOR

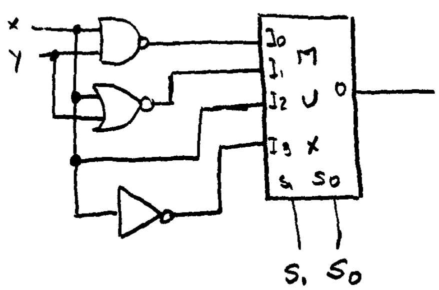
$S=1 \Rightarrow$ XNOR

- ALU para una NAND, NOR, transferencia y complem de x

⇓

4 operaciones ⇒ MUX 4 a 1 ⇒ 2 select

S_1	S_0	Operación
0	0	$\overline{x \cdot y}$
0	1	$\overline{x + y}$
1	0	x
1	1	$\bar{x}$



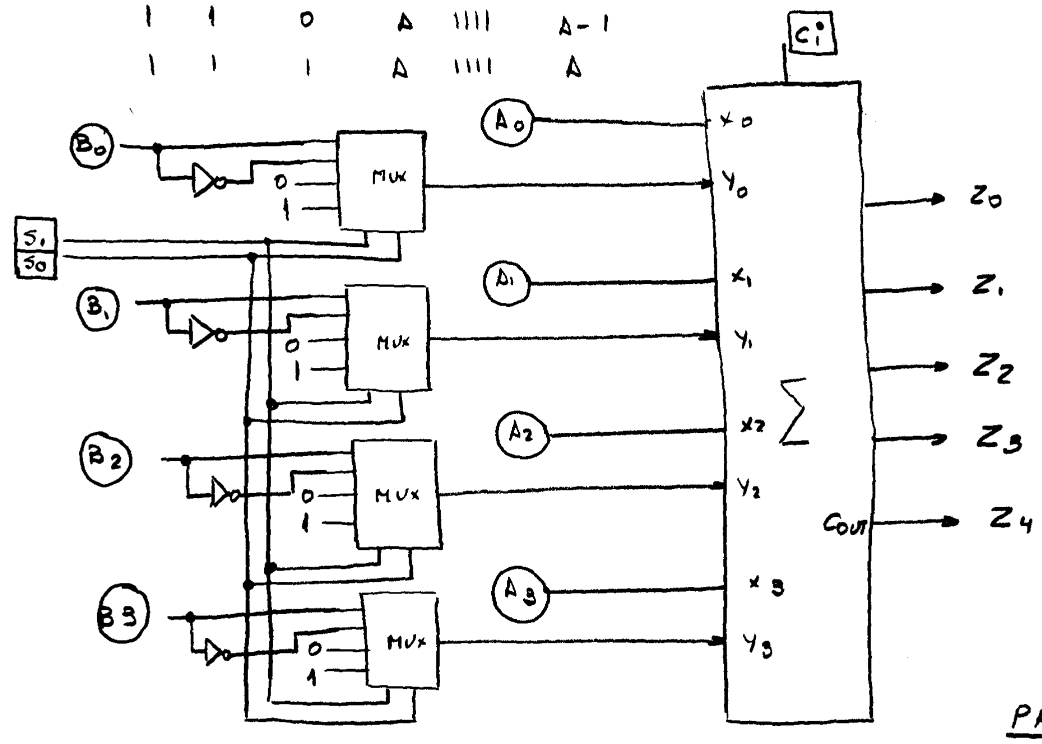
Problema 4.30

ALU 4 bits para tabla

Operación		Descripción
1	$A+B$	suma aritm
2	$A+B+1$	" " con arrastre
3	$A+\bar{B}$	Resta $A-B$ con arras
4	$A+\bar{B}+1$	Resta $A-B$
5	A	
6	$A+1$	Incrementa A
7	$A-1$	Decrementa A
8	A	

El corazón es un sumador de 4 bits (SBC) al que hay que alimentar con por un lado siempre A y por otro lado $B/B+1/\bar{B}/\bar{B}+1/1/$ com 1; o sea unas veces con B o $\bar{B}$ y con C_{IN} ó $\emptyset$. ⇒ sumador con ci que se pondrá a $\emptyset$ en las operaciones impares 1, 3, 5, 7 y a 1 en las pares 2, 4, 6, 8.

S_1	S_0	C_i	X	Y	sali
0	0	0	A	B	$A+B$
0	0	1	A	B	$A+B+1$
0	1	0	A	$\bar{B}$	$A+\bar{B}$
0	1	1	A	$\bar{B}$	$A+\bar{B}+1$
1	0	0	A	$\emptyset$	$A+\emptyset$
1	0	1	A	$\emptyset$	$A+1$
1	1	0	A	1111	$A-1$
1	1	1	A	1111	A



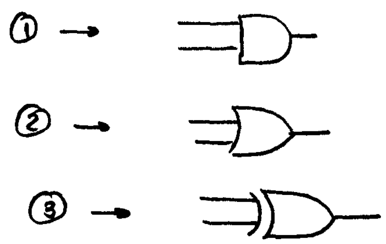
Problema 4.31

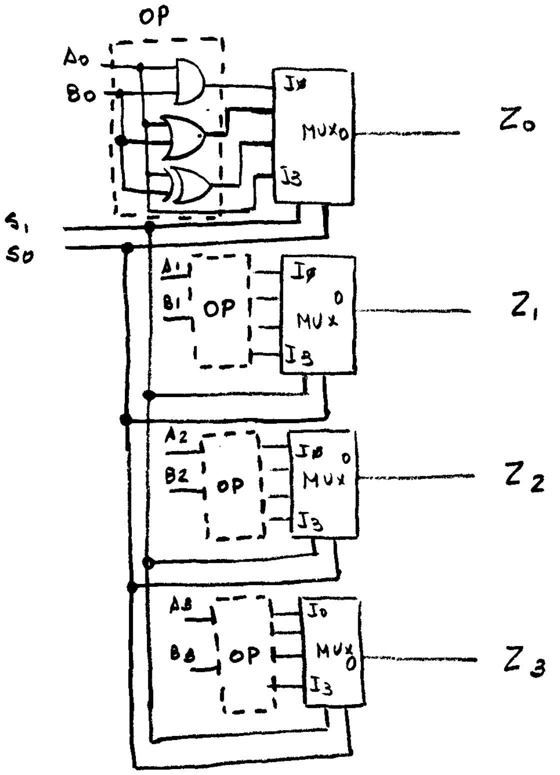
ALU de 4 bits para →

	Operación	Descripción
①	$A \wedge B$	"A" AND "B"
②	$A \vee B$	"A" OR "B"
③	$A \oplus B$	"A" XOR "B"
④	$\bar{A}$	

⇓

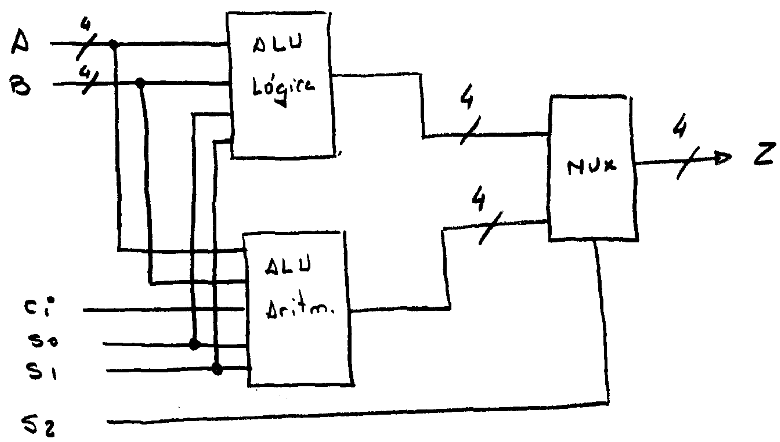
4 operaciones ⇒ MUX de 4 a 1 con 2 pin selección S_1 S_0





Problema 4.32

Hacer una ALU con las 2 anteriores ⇒ Un MUX de 2 a 1 que elija la salida de una ALU o la otra



└ selección de ALU

Problema 4. 33

Con el 74181 hacer un sumador-restador de 16 bits para nºs en complemento a 2.

En la tabla de funciones del ALU 74181 se puede ver que:

Funciones aritme ⇒ $M=0$

c_n
$c_e = 0 \rightarrow$ Suma $A+B$ $\Rightarrow$ $\overset{s}{c_3}\overset{s}{c_2}\overset{s}{c_1}\overset{s}{c_0} = 1001$

c_n
$c_e = 1 \rightarrow$ Resta $A-B$ $\Rightarrow$ $\overset{s}{c_3}\overset{s}{c_2}\overset{s}{c_1}\overset{s}{c_0} = 0110$

⇓

c_i de suma $= \overline{c_i}$ de resta.

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	$F = \overline{A}$	F = A MINUS 1	F = A
L	L	L	H	$F = \overline{AB}$	F = AB MINUS 1	F = AB
L	L	H	L	$F = \overline{A} + B$	F = $A\overline{B}$ MINUS 1	F = $A\overline{B}$
L	L	H	H	F = 1	F = MINUS 1 (2's COMP)	F = ZERO
L	H	L	L	$F = \overline{A+B}$	F = A PLUS $(A + \overline{B})$	F = A PLUS $(A+\overline{B})$ PLUS 1
L	H	L	H	$F = \overline{B}$	F = AB PLUS $(A+\overline{B})$	F = AB PLUS $(A+\overline{B})$ PLUS 1
L	H	H	L	$F = \overline{A \oplus B}$	F = A MINUS B MINUS 1	F = A MINUS B
L	H	H	H	$F = A + \overline{B}$	$F = A + \overline{B}$	F = $(A+\overline{B})$ PLUS 1
H	L	L	L	$F = \overline{A}B$	F = A PLUS (A + B)	F = A PLUS (A + B) PLUS 1
H	L	L	H	$F = A \oplus B$	F = A PLUS B	F = A PLUS B PLUS 1
H	L	H	L	F = B	F = $A\overline{B}$ PLUS (A + B)	F = $A\overline{B}$ PLUS (A + B) PLUS 1
H	L	H	H	F = A + B	F = (A + B)	F = (A + B) PLUS 1
H	H	L	L	F = 0	F = A PLUS A†	F = A PLUS A PLUS 1
H	H	L	H	$F = A\overline{B}$	F = AB PLUS A	F = AB PLUS A PLUS 1
H	H	H	L	F = AB	F = $A\overline{B}$ PLUS A	F = $A\overline{B}$ PLUS A PLUS 1
H	H	H	H	F = A	F = A	F = A PLUS 1

Lo que debemos buscar es un circuito que con una señal

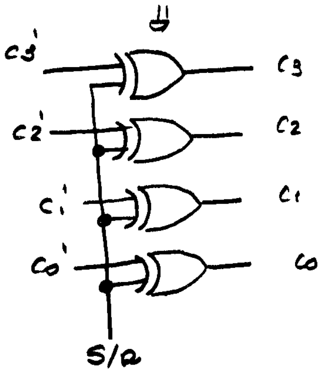
$S/R \begin{cases} 0 \Rightarrow \text{suma} \\ 1 \Rightarrow \text{resta} \end{cases}$ que

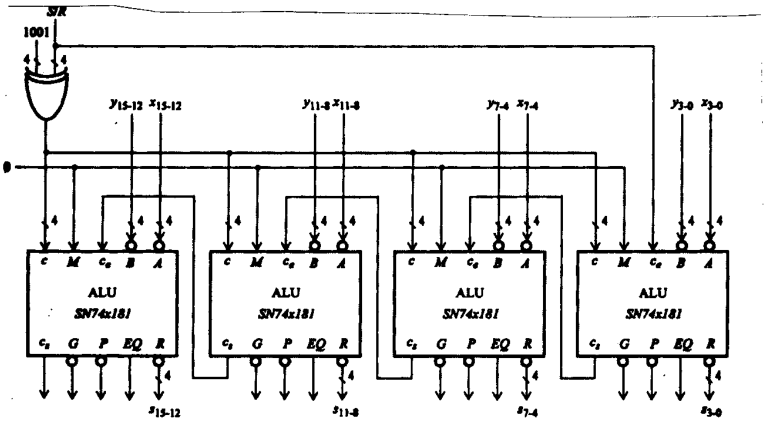
se conectará al ce del primer c.I. de forma que con

$0 \Rightarrow c_e = 0$ y $\overset{s}{c_3}\overset{s}{c_2}\overset{s}{c_1}\overset{s}{c_0} = 1001$
$1 \Rightarrow c_e = 1$ y $\overset{s}{c_3}\overset{s}{c_2}\overset{s}{c_1}\overset{s}{c_0} = 0110$

} debe invertir $c_3 c_2 c_1 c_0$

⇓





Problema 4.39

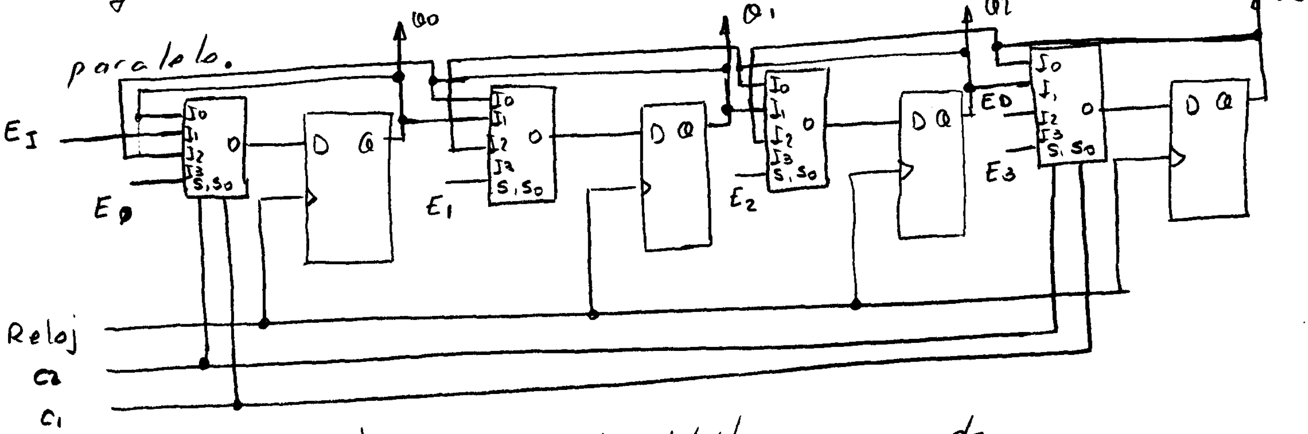
Registro desplazamiento de 8 bits de entrada paralelo/salida paralelo y capaz de realizar los desplazamientos

LJAD (Lógico - Izda - Abierto - Doble)
LDAD (" - Dcha - " - ")

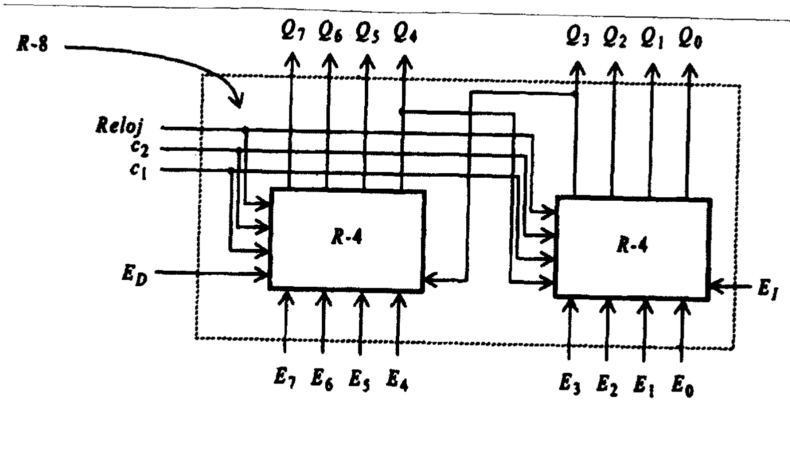
- Se va hacer en base a 2 registros de 4 bits
- Se van a elegir 2 patillas de elección de operación

Operación	c_2	c_1	
NOP	0	0	→ salida de Q_i unida a D_i
LIAS	0	1	→ Desplaz. izda → Salida Q_i a D_{i+1}
LDAS	1	0	→ " dcha → " Q_i a D_{i-1}
CARGA	1	1	→ carga paralelo → Entrada E_i a D_i

Cada bit será una báscula "D" y mediante un multiplexor elegiremos entrada desde báscula anterior, posterior o entrada paralelo.



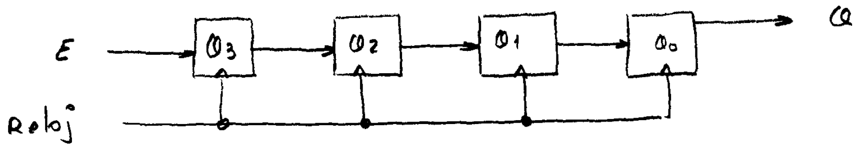
Para 8 bit poner 2 de 4 bits en cascada



Problema 4.40

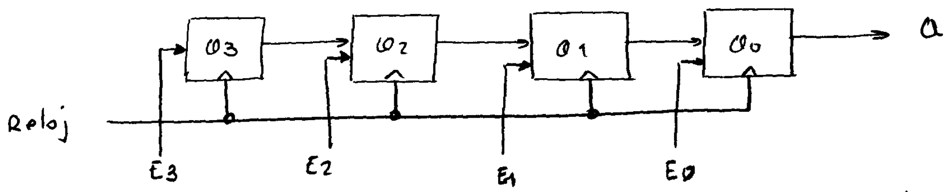
En el registro del problema anterior para las estructuras:

1) Entrada serie/salida serie



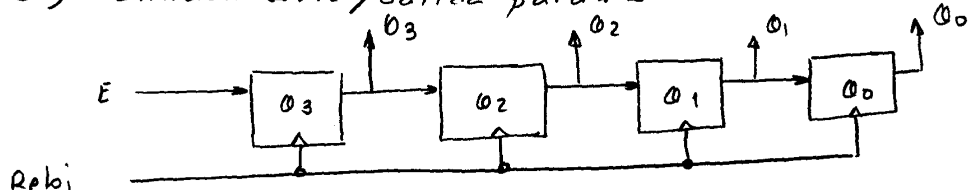
Será el anterior con:
- $C_2\ C_1 \Rightarrow 10 \Rightarrow$ desplaz dcha
- salida por Q_0
- 4 pulsos de reloj

2) Entrada paralelo/salida serie



Será igual pero:
- 1º Carga paralelo $\Rightarrow$ $C_2\ C_1 \Rightarrow 11$; pulso de reloj
- 2º Desplaz dcha $\Rightarrow$ $C_2\ C_1 \Rightarrow 10$; salida $\Rightarrow Q_0$; 4 pulsos reloj

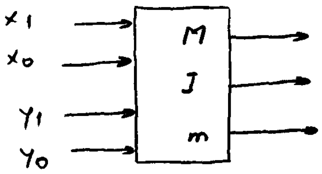
3º) Entrada serie/salida paralelo



Igual pero:
- 1º carga serie $\Rightarrow$ $C_2\ C_1 \Rightarrow 10$; 4 pulsos
- 2º Las salidas de Q_3 a Q_0 ya tienen la información en paralelo

Problema 4.41

Comparador de 2 nºs de 2 bits



$x > y \Rightarrow M = 1$

$x = y \Rightarrow J = 1$

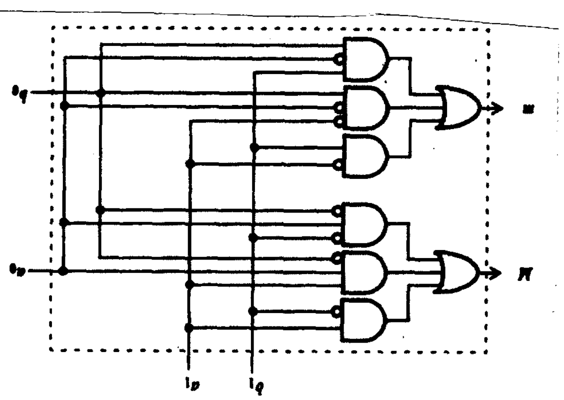
$x < y \Rightarrow m = 1$

Opciones:

Ⓐ Diseño completo en base a las funciones:

x_1	x_0	y_1	y_0	M	J	m
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

$J \Rightarrow$ cuando ni M ni $m \Rightarrow J = \overline{M + m}$



Ⓜ

		$\bar{x}_1 \bar{x}_0$	$\bar{x}_1 x_0$	$x_1 x_0$	$x_1 \bar{x}_0$
$\bar{y}_1$	$\bar{y}_0$	0	1	1	1
$\bar{y}_1$	y_0	0	0	1	1
y_1	y_0	0	0	0	0
y_1	$\bar{y}_0$	0	0	1	0

$M = x_1\bar{y}_1 + \bar{y}_1\bar{y}_0 x_0 + \bar{y}_0 x_1 x_0$

Ⓙ

		$\bar{x}_1 \bar{x}_0$	$\bar{x}_1 x_0$	$x_1 x_0$	$x_1 \bar{x}_0$
$\bar{y}_1$	$\bar{y}_0$	1	0	0	0
$\bar{y}_1$	y_0	0	1	0	0
y_1	y_0	0	0	1	0
y_1	$\bar{y}_0$	0	0	0	1

ⓜ

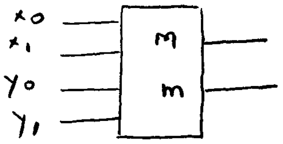
		$\bar{x}_1 \bar{x}_0$	$\bar{x}_1 x_0$	$x_1 x_0$	$x_1 \bar{x}_0$
$\bar{y}_1$	$\bar{y}_0$	0	0	0	0
$\bar{y}_1$	y_0	1	0	0	0
y_1	y_0	1	1	0	1
y_1	$\bar{y}_0$	1	1	0	0

$m = \bar{x}_1 y_1 + y_0 \bar{x}_1 \bar{x}_0 + y_1 y_0 \bar{x}_0$

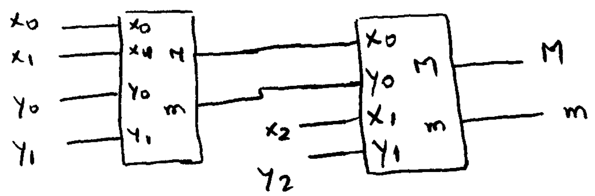
Problema 4.43

Diseñar un comparador serie y otro paralelo para comparar dos nºs X e Y representados en magnitud y signo

Comparador palab. 2 bit



Comp. pal. 3 bit



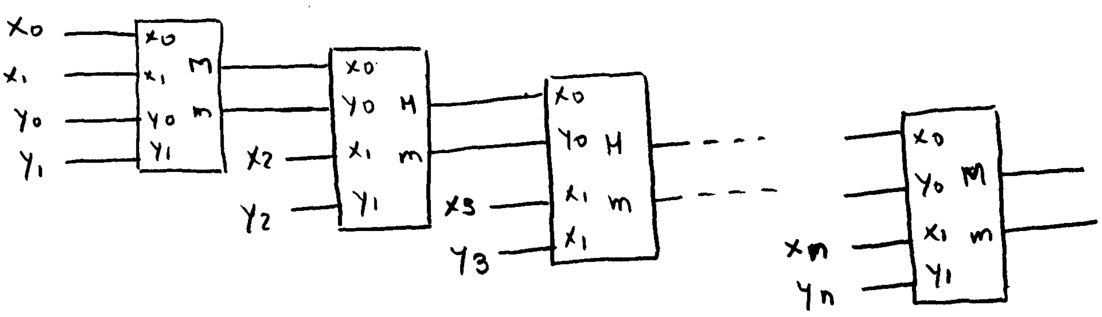
Y_2	Y_1	Y_0	X_2	X_1	X_0	M	m
1	x	x	0	x	x	0	1
0	x	x	1	x	x	1	0
1	x	x	1	x	x	→ comparación entre X_1X_0 Y_1Y_0	
0	x	x	0	x	x	" " " "	

Esto es → si $X_2 > Y_2$ no importa lo que valgan X_1X_0 Y_1Y_0 que $X > Y$

→ si $X_2 < Y_2$ " " " " " " "" $X < Y$

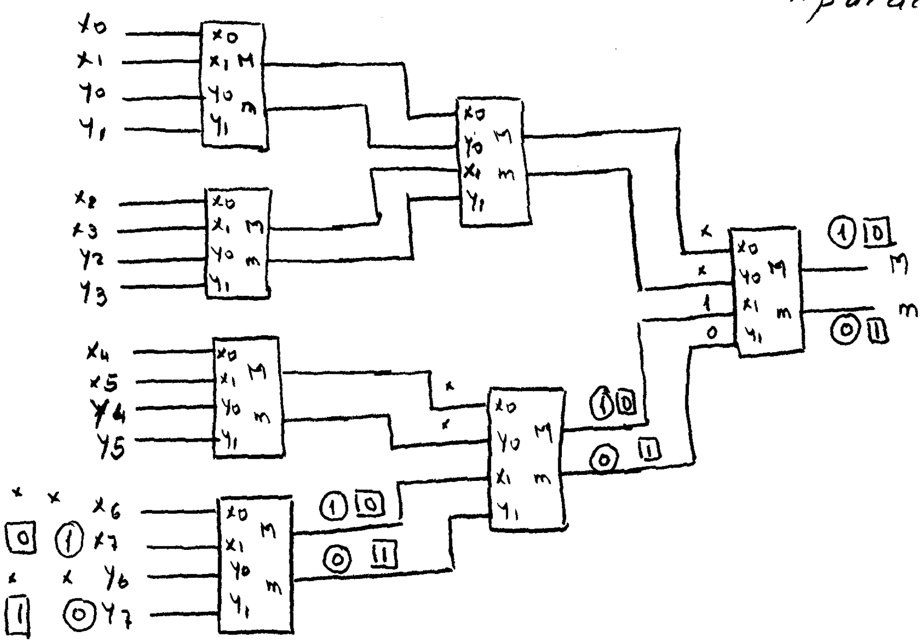
→ solo en el caso de que $X_2 = Y_2 \Rightarrow$ el resultado final de la comparación es el estado de comparar los bits inferiores a X_2 e Y_2

Comparador pal. de n bits



Comparador paralelo

Consiste en comparar partes de la palabra y luego comparar el resultado de las comparaciones.



Si $x_7 > y_7 \Rightarrow M = 1 \quad m = 0$ si $y_7 > x_7 \Rightarrow M = 0 \quad m = 1$

- Solo en el caso de que $x_7 = y_7$ el valor de salida será el resultado de la comparación de los bit $x_6 \cdots x_0$ e $y_6 \cdots y_0$.
- Para comparar con signo ⇒ tratar al signo con x_8 e y_8

Problema 4-44

Con una ROM de 16×4 diseñar un comparador de 2 bit que genere M y m. y $x = y$

Se conectar los 4 bits $x_1\ x_0$ e $y_1\ y_0$ al bus de direcciones y en cada de las direcciones se graba el valor correspondiente al obtenido de la tabla de la verdad, teniendo en cuenta que:

$x < y \longrightarrow D_2$

$x = y \longrightarrow D_1$

$x > y \longrightarrow D_0$

A_3	A_2	A_1	A_0	D_3	D_2	D_1	D_0
x_1	x_0	y_1	y_0	0	$x<y$	$x=y$	$x>y$
0	0	0	0	0	0	1	0
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	1
1	0	1	0	0	0	1	0
1	0	1	1	0	1	0	0
1	1	0	0	0	0	0	1
1	1	0	1	0	0	0	1
1	1	1	0	0	0	0	1
1	1	1	1	0	0	1	0

Direccion	Conte. Hex
0	2
1	4
2	4
3	4
4	1
5	2
6	4
7	4
8	1
9	1
A	2
B	4
C	1
D	1
E	1
F	2

Problema 4-45

Supuesta longitud de palabra de 8 bits y representación sin signo. Verificar las siguientes relaciones por el método de comparación por suma.

1) $x = 57$ $\quad y = 23$

Se suma x + complemento A 2 de $y \Rightarrow \begin{cases} Z=1 \Rightarrow x=y \\ C=1 \Rightarrow x>y \end{cases}$

57 = 00111001

23 = 00010111 ⇒ 11101000 + 1 = 11101001

⇒

$$\begin{array}{r} 00111001 \\ 11101001 \\ \hline 00100010 \end{array}$$

$C = 1$ $\quad Z = \emptyset$

$x > y$

2) $x = 23$ e $y = 57$

$x = 0\,0\,0\,1\,0\,1\,1\,1$

$y = 0\,0\,1\,1\,1\,0\,0\,1 \rightarrow$ C. a 2 $\rightarrow$

$$\begin{array}{r} 11000110 \\ 1 \\ \hline 11000111 \end{array}$$

$$\begin{array}{r} 00010111 \\ 11000111 \\ \hline 11011110 \end{array}$$

$C = 0$ $\quad Z \neq 0 \Rightarrow x < y$

Problema 4-46

Suponiendo una longitud de palabra de 8 bits y que los nos se representan en complemento a 2, comparar los nos siguientes por el método de comparación por suma.

1) $x = -57 \longrightarrow -57 \Rightarrow 00111001 \xrightarrow{Ca2}$

$$\begin{array}{r} 11000110 \\ + \quad 1 \\ \hline 11000111 \end{array}$$

$y = 23$

$x + (-y) \Rightarrow -23 \rightarrow 00010111 \xrightarrow{Ca2}$

$$\begin{array}{r} 11101000 \\ 1 \\ \hline 11101001 \end{array}$$

$x + (-y) \Rightarrow$

$$\begin{array}{r} 11000111 \\ 11101001 \\ \hline 10110000 \end{array} \Rightarrow \left\{ \begin{array}{l} C = 1 \\ Z = 0 \\ N = 1 \\ V = 0 \end{array} \right\} \Rightarrow N \oplus V = 1 \Rightarrow x < y$$

2) $x = 23 \longrightarrow 00010111$

$y = -57 \longrightarrow x + (-y) \Rightarrow x + y \Rightarrow 57 \rightarrow 00111001$

$$\begin{array}{r} 00010111 \\ 00111001 \\ \hline 01010000 \end{array} \Rightarrow \left\{ \begin{array}{l} C = 0 \\ Z = 0 \\ V = 0 \\ N = 0 \end{array} \right. \Rightarrow N \oplus V = 0 \Rightarrow x > y$$