

# FUNDAMENTOS Y ESTRUCTURA DE COMPUTADORES

## 1º G. I. Informática

Curso 2010 – 2011

Página 1 de 5

### Relación de ejercicios

#### Temas 6, 7 y 8: Diseño Combinacional (II)

---

##### Ejercicio 1

Diseñar un circuito sumador/restador completo de 1 bit. El circuito tendrá cuatro entradas: los operandos  $x$  e  $y$ , el acarreo/adeudo de entrada y una señal de control  $s/r$  para indicar suma ( $s/r = 0$ ) o resta ( $s/r = 1$ ). Tendrá asimismo dos señales de salida: el resultado de la operación de suma o resta y el acarreo/adeudo de salida.

##### Ejercicio 2

Diseñar un comparador de números de 1 bit que incorpore tres entradas para conexión en cascada:  $(X_{in} < Y_{in})$ ,  $(X_{in} = Y_{in})$  y  $(X_{in} > Y_{in})$ . Estas entradas indican el resultado de la comparación en los bits de peso inferior a los que se comparan. Tres serán las salidas:  $(X < Y)$ ,  $(X = Y)$  y  $(X > Y)$ .

##### Ejercicio 3

Diseñar un comparador de números de dos bits. Tendrá 4 entradas: dos para uno de los números ( $X_1, X_2$ ) y dos para el otro ( $Y_1, Y_2$ ). Tendrá asimismo tres salidas:  $(X < Y)$ ,  $(X = Y)$  y  $(X > Y)$ .

##### Ejercicio 4

Diseñar un comparador de números de dos bits utilizando los comparadores de 1 bit diseñados en el ejercicio 2.

##### Ejercicio 5

Utilizando comparadores de magnitud de datos de 4 bits más las puertas necesarias, diseñar un circuito que compruebe, dados tres datos de 4 bits, si son iguales.

##### Ejercicio 6

- Diseñar un conversor de código Gray de 4 bits a binario natural.
- Diseñar un conversor de código binario natural a Gray de 4 bits.

##### Ejercicio 7

Diseñar un conversor de código BCD a BCD exceso 3 utilizando un sumador.

# FUNDAMENTOS Y ESTRUCTURA DE COMPUTADORES

## 1º G. I. Informática

Curso 2010 – 2011

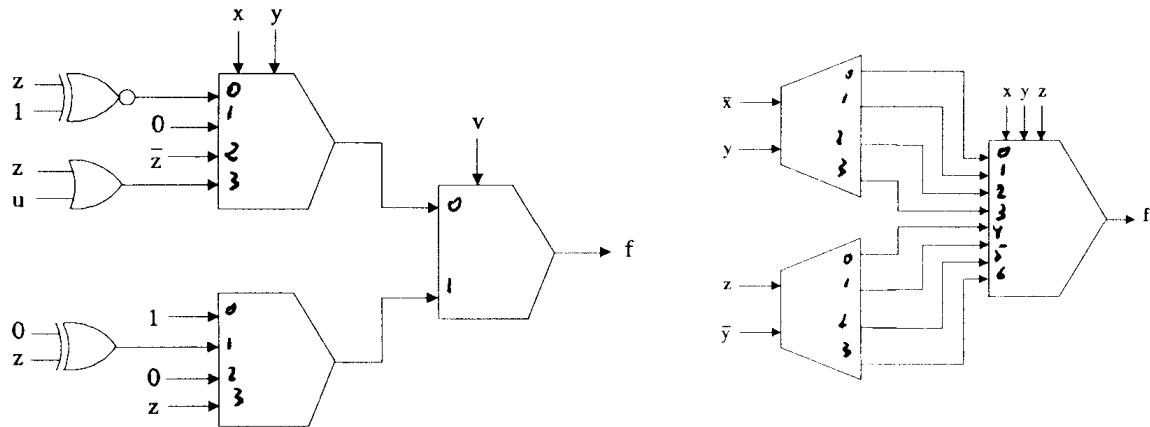
Página 2 de 5

### Relación de ejercicios

#### Temas 6, 7 y 8: Diseño Combinacional (II)

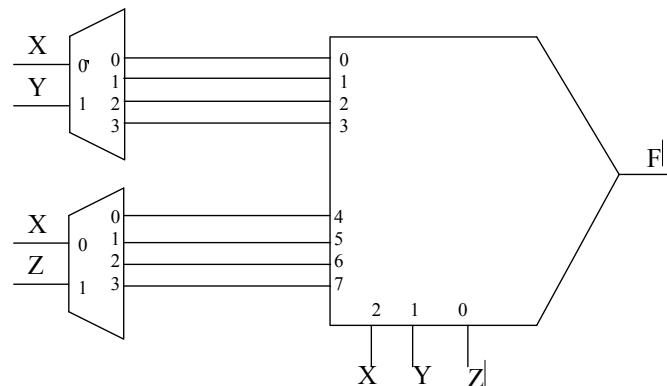
##### Ejercicio 8

Obtener las funciones sintetizadas por los circuitos de la figura siguiente:



##### Ejercicio 9

Obtener la función sintetizada por el circuito de la figura siguiente:



# FUNDAMENTOS Y ESTRUCTURA DE COMPUTADORES

## 1º G. I. Informática

Curso 2010 – 2011

Página 3 de 5

### Relación de ejercicios

#### Temas 6, 7 y 8: Diseño Combinacional (II)

---

##### Ejercicio 10

Implementar la función definida por la siguiente tabla de verdad empleando:

- a) Decodificadores
- b) Multiplexores

X	Y	Z		F
0	0	0		0
0	0	1		1
0	1	0		1
0	1	1		0
1	0	0		1
1	0	1		0
1	1	0		0
1	1	1		1

##### Ejercicio 11

Diseñar una unidad lógica de 1 bit que realice las operaciones lógicas siguientes en función de dos señales de control  $S_1$  y  $S_0$ :

$S_1$	$S_0$	Operación
0	0	$A$
0	1	$A \cdot B$
1	0	$A + B$
1	1	$A \oplus B$

- a) Realizarla con un multiplexor de 4 a 1 y las puertas necesarias.

b) Realizarla con un multiplexor del tamaño necesario aplicando la técnica de implementación de funciones mediante multiplexores.

##### Ejercicio 12

Diseñar una unidad aritmética de 4 bits partiendo de un sumador completo de cuatro bits que realice las operaciones siguientes en función de dos señales de control  $S_1$  y  $S_0$ :

$S_1$	$S_0$	Operación
0	0	$A + 1$
0	1	$A + 2$
1	0	$A - 1$
1	1	$A - 2$

# FUNDAMENTOS Y ESTRUCTURA DE COMPUTADORES

## 1º G. I. Informática

### Relación de ejercicios

#### Temas 6, 7 y 8: Diseño Combinacional (II)

---

##### Ejercicio 13

Diseñar una unidad aritmética de 4 bits partiendo de un sumador completo de cuatro bits que realice las operaciones siguientes en función de dos señales de control  $S_1$  y  $S_0$ :

$S_1$	$S_0$	Operación
0	0	$A + 2$
0	1	$A + 3$
1	0	$A - 3$
1	1	$A - B$

##### Ejercicio 14

Diseñar una unidad aritmética de 4 bits partiendo de un sumador completo de cuatro bits que realice las operaciones siguientes en función de dos señales de control  $S_1$  y  $S_0$ :

$S_1$	$S_0$	Operación
0	0	$A + B + C_E$
0	1	$A + 1$
1	0	$A - B - \bar{C}_E$
1	1	$A - 1$

##### Ejercicio 15

Diseñar una unidad aritmética de 4 bits partiendo de un sumador completo de cuatro bits que realice las operaciones siguientes en función de dos señales de control  $S_1$  y  $S_0$ :

$S_1$	$S_0$	Operación
0	0	$A - B$
0	1	$-A + B$
1	0	$A + 1$
1	1	$B - 1$

# FUNDAMENTOS Y ESTRUCTURA DE COMPUTADORES

1º G. I. Informática

Curso 2010 – 2011

Página 5 de 5

## Relación de ejercicios

### Temas 6, 7 y 8: Diseño Combinacional (II)

---

#### Ejercicio 16

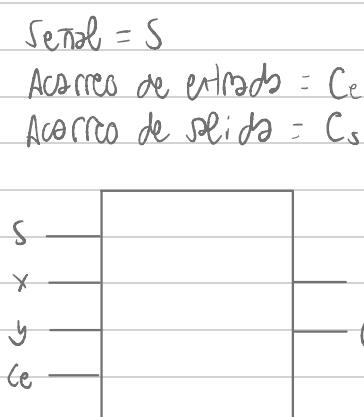
Diseñar una unidad aritmético-lógica de 4 bits que realice las operaciones siguientes en función de tres señales de control  $S_2$ ,  $S_1$  y  $S_0$ :

		Operaciones Lógicas	Operaciones Aritméticas
$S_1$	$S_0$	$S_2 = 0$	$S_2 = 1$
0	0	$A + B$	$A - 2$
0	1	$A \cdot B$	$-B + 1$
1	0	$A \oplus B$	$-A - 1$
1	1	$\overline{A \oplus B}$	$B - 1$

Realizar la unidad lógica empleando multiplexores del tamaño necesario aplicando la técnica de implementación de funciones mediante multiplexores. Para la unidad aritmética, partir de un sumador completo de cuatro bits.

## Ejercicio 1

Diseñar un circuito sumador/restador completo de 1 bit. El circuito tendrá cuatro entradas: los operandos  $x$  e  $y$ , el acarreo/adeudo de entrada y una señal de control  $s/r$  para indicar suma ( $s/r = 0$ ) o resta ( $s/r = 1$ ). Tendrá asimismo dos señales de salida: el resultado de la operación de suma o resta y el acarreo/adeudo de salida.



$R: S_x \setminus y \setminus C_e$	00	01	11	10	$C_s: S_x \setminus y \setminus C_e$	00	01	11	10
00	1		1		00		1		
01	1	1			01	1	1	1	
11	1		1		11		1		
10		1		1	10	1	1	1	1

$$R(S_x, x, y, C_e) = \bar{x}\bar{y}C_e + \bar{x}y\bar{C}_e + xyC_e + \bar{x}y\bar{C}_e$$

$$C_s(S_x, x, y, C_e) = yC_e + \bar{S}_x C_e + \bar{x}y + \bar{S}\bar{x}C_e + \bar{S}\bar{x}y$$

## Ejercicio 2

Diseñar un comparador de números de 1 bit que incorpore tres entradas para conexión en cascada:  $(X_{in} < Y_{in})$ ,  $(X_{in} = Y_{in})$  y  $(X_{in} > Y_{in})$ . Estas entradas indican el resultado de la comparación en los bits de peso inferior a los que se comparan. Tres serán las salidas:  $(X < Y)$ ,  $(X = Y)$  y  $(X > Y)$ .

$x$	$y$	$x > y_e$	$x = y_e$	$x < y_e$	$x > y$	$x = y$	$x < y$
0	0	0	0	0	—	—	—
0	0	0	0	1	0	0	1
0	0	0	1	0	0	1	0
0	0	0	1	1	—	—	—
0	0	1	0	0	1	0	0
0	0	1	0	1	—	—	—
0	0	1	1	0	—	—	—
0	0	1	1	1	—	—	—
0	1	0	0	0	—	—	—
0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0
0	1	0	1	1	—	—	—
0	1	1	0	0	1	0	0
0	1	1	0	1	—	—	—
0	1	1	1	0	—	—	—
0	1	1	1	1	—	—	—
1	0	0	0	0	—	—	—
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	1
1	0	0	1	1	—	—	—
1	0	1	0	0	0	0	1
1	0	1	0	1	—	—	—
1	0	1	1	0	—	—	—
1	1	0	0	0	—	—	—
1	1	0	0	1	0	0	1
1	1	0	1	0	0	1	0
1	1	0	1	1	—	—	—
1	1	1	0	0	1	0	0
1	1	1	0	1	—	—	—
1	1	1	1	0	—	—	—
1	1	1	1	1	—	—	—

Las combinaciones con ---

no son posibles

### Ejercicio 3

Diseñar un comparador de números de dos bits. Tendrá 4 entradas: dos para uno de los números ( $X_1, X_2$ ) y dos para el otro ( $Y_1, Y_2$ ). Tendrá asimismo tres salidas: ( $X < Y$ ), ( $X = Y$ ) y ( $X > Y$ ).

$X_1$	$X_2$	$Y_1$	$Y_2$	$X < Y$	$X = Y$	$X > Y$
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	0
1	0	0	0	0	0	1
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	1	0
1	1	1	0	0	0	1
1	1	1	1	0	1	0

$X < Y$	$x_1, x_2 \backslash y_1, y_2$	00	01	11	10
	$x_1, x_2 \backslash y_1, y_2$	00	01	11	10
		1	1	1	1
00		1	1	1	1
01			1		
11					
10			1		

$X = Y$	$x_1, x_2 \backslash y_1, y_2$	00	01	11	10
	$x_1, x_2 \backslash y_1, y_2$	00	01	11	10
		1			
00		1			
01		1	1	1	1
11			1		
10		1	1	1	1

$X > Y$	$x_1, x_2 \backslash y_1, y_2$	00	01	11	10
	$x_1, x_2 \backslash y_1, y_2$	00	01	11	10
00					
01		1		1	
11		1	1	1	1
10		1			

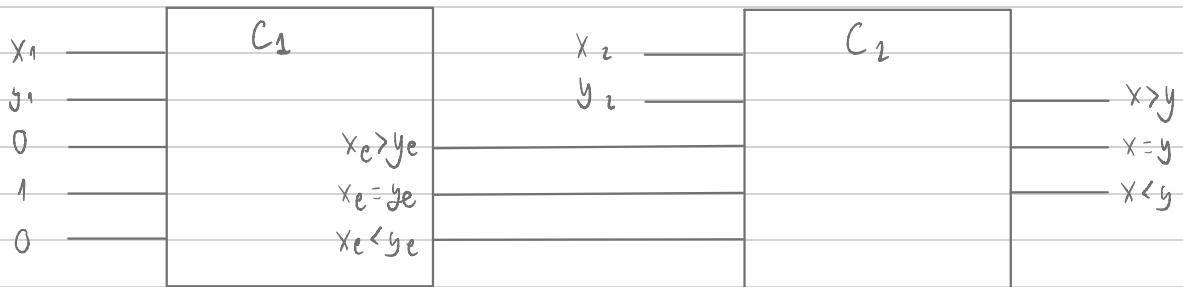
$$X < Y (X_1, X_2, Y_1, Y_2) = \bar{X}_1 \bar{X}_2 Y_2 + \bar{X}_1 X_2 \bar{Y}_1 + \bar{X}_1 Y_1 Y_2 + \bar{X}_1 Y_1 \bar{Y}_2$$

$$X = Y (X_1, X_2, Y_1, Y_2) = \bar{X}_1 \bar{X}_2 \bar{Y}_1 \bar{Y}_2 + \bar{X}_1 X_2 \bar{Y}_1 Y_2 + X_1 \bar{X}_2 Y_1 Y_2 + X_1 \bar{X}_2 Y_1 \bar{Y}_2 + \bar{X}_1 X_2 Y_1 \bar{Y}_2 + X_1 \bar{X}_2 \bar{Y}_1 Y_2$$

$$X > Y (X_1, X_2, Y_1, Y_2) = \bar{X}_1 X_2 \bar{Y}_2 + X_1 X_2 \bar{Y}_1 + X_1 \bar{Y}_1 \bar{Y}_2$$

## Ejercicio 4

Diseñar un comparador de números de dos bits utilizando los comparadores de 1 bit diseñados en el ejercicio 2.

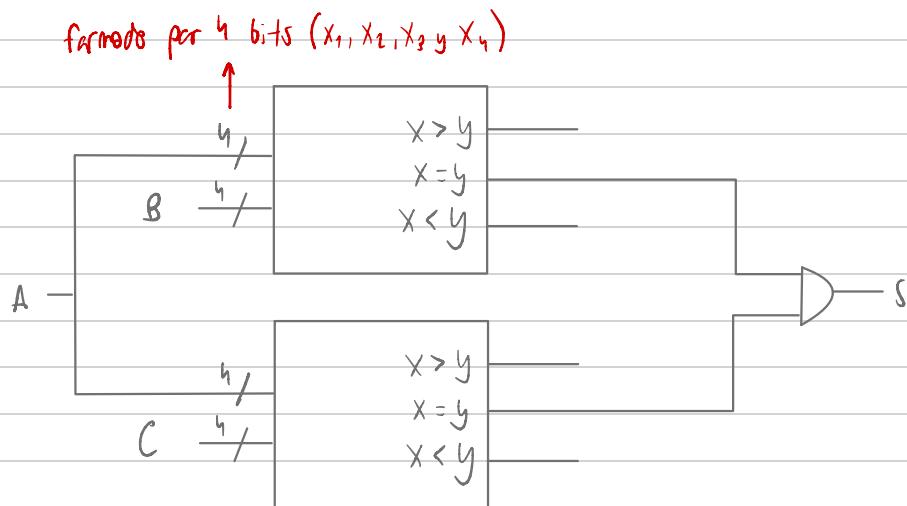


$C_1:$	$x_1$	$y_1$	0	1	0	$x_e > y_e$	$x_e = y_e$	$x_e < y_e$
	0	0	0	1	0	0	1	0
	0	1	0	1	0	0	0	1
	1	0	0	1	0	1	0	0
	1	1	0	1	0	0	1	0

$C_2:$	$x_2$	$y_2$	$x_e > y_e$	$x_e = y_e$	$x_e < y_e$	$x > y$	$x = y$	$x < y$
	0	0	1	0	0	1	0	0
	0	0	0	1	0	0	1	0
	0	0	0	0	1	0	0	1
	0	1	1	0	0	0	1	0
	0	1	0	1	0	0	0	1
	0	1	0	0	1	0	0	1
	1	0	1	0	0	1	0	0
	1	0	0	1	0	1	0	0
	1	0	0	0	1	0	1	0
	1	1	1	0	0	1	0	0
	1	1	0	1	0	0	1	0
	1	1	0	0	1	0	0	1

## Ejercicio 5

Utilizando comparadores de magnitud de datos de 4 bits más las puertas necesarias, diseñar un circuito que compruebe, dados tres datos de 4 bits, si son iguales.



## Ejercicio 6

a) Diseñar un conversor de código Gray de 4 bits a binario natural.

b) Diseñar un conversor de código binario natural a Gray de 4 bits.

$G_3\ G_2\ G_1\ G_0$	$B_3\ B_2\ B_1\ B_0$	$G_f: \begin{array}{c} B_3\ B_0 \\ B_3\ B_1 \end{array}$	$00\ 01\ 11\ 10$	$G_l: \begin{array}{c} B_3\ B_0 \\ B_3\ B_1 \end{array}$	$00\ 01\ 11\ 10$
0 0 0 0	0 0 0 0		00		00
0 0 0 1	0 0 0 1		01		01
0 0 1 1	0 0 1 0		11	1 1 1 1	11 1 1 1
0 0 1 0	0 0 1 1		10	1 1 1 1	10
0 1 1 0	0 1 0 0				
0 1 1 1	0 1 0 1	$G_4: \begin{array}{c} B_3\ B_0 \\ B_3\ B_1 \end{array}$	00 01 11 10	$G_0: \begin{array}{c} B_3\ B_0 \\ B_3\ B_1 \end{array}$	00 01 11 10
0 1 0 1	0 1 1 0		00	1 1	00
0 1 0 0	0 1 1 1		01	1 1	01
1 1 0 0	1 0 0 0		11	1 1	11
1 1 0 1	1 0 0 1		10	1 1	10
1 1 1 1	1 0 1 0				
1 1 1 0	1 0 1 1	$B_3: \begin{array}{c} G_6\ G_0 \\ G_6\ G_1 \end{array}$	00 01 11 10	$B_2: \begin{array}{c} G_6\ G_0 \\ G_6\ G_1 \end{array}$	00 01 11 10
1 0 1 0	1 1 0 0		00		00
1 0 1 1	1 1 0 1		01		01
1 0 0 1	1 1 1 0		11	1 1 1 1	11
1 0 0 0	1 1 1 1		10	1 1 1 1	10

$$2) G_3(B_3, B_2, B_1, B_0) = B_3$$

$$G_2(B_3, B_2, B_1, B_0) = B_2$$

$$G_1(B_3, B_2, B_1, B_0) = B_2 \bar{B}_1 + \bar{B}_3 \bar{B}_2 B_0 + B_3 \bar{B}_2 B_1$$

$$G_0(B_3, B_2, B_1, B_0) = B_1 B_0 + B_1 \bar{B}_0$$

$$6) B_3(G_3, G_2, G_1, G_0) = G_3$$

$$B_2(G_3, G_2, G_1, G_0) = \bar{G}_3 G_2 + G_3 \bar{G}_2$$

$$B_1(G_3, G_2, G_1, G_0) = \bar{G}_3 \bar{G}_2 G_1 + \bar{G}_3 G_2 \bar{G}_1 + G_3 G_2 G_1 + G_3 \bar{G}_2 \bar{G}_1$$

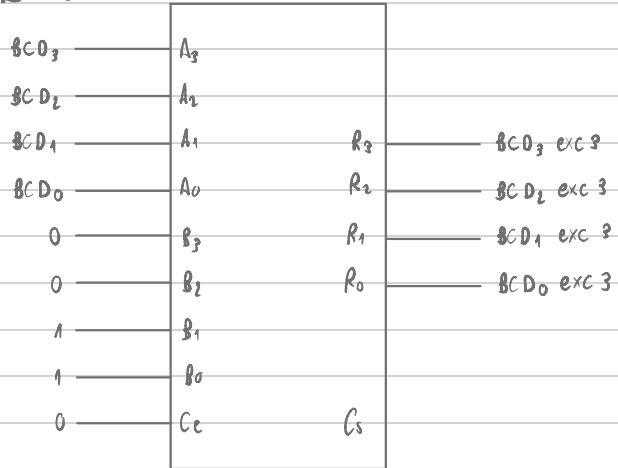
$$B_0(G_3, G_2, G_1, G_0) = G_3 \bar{G}_2 G_1 G_0 + G_3 G_2 \bar{G}_1 G_0 + G_3 \bar{G}_2 \bar{G}_1 \bar{G}_0 + \bar{G}_3 G_2 G_1 G_0 + \bar{G}_3 \bar{G}_2 G_1 \bar{G}_0 + G_3 \bar{G}_2 \bar{G}_1 \bar{G}_0 + \bar{G}_3 \bar{G}_2 \bar{G}_1 G_0$$

$B_4: \begin{array}{c} G_6\ G_0 \\ G_6\ G_1 \end{array}$	$00\ 01\ 11\ 10$	$B_0: \begin{array}{c} G_6\ G_0 \\ G_6\ G_1 \end{array}$	$00\ 01\ 11\ 10$
	00	1 1	00
	01	1 1	01
	11	1 1	11
	10	1 1	10

## Ejercicio 7

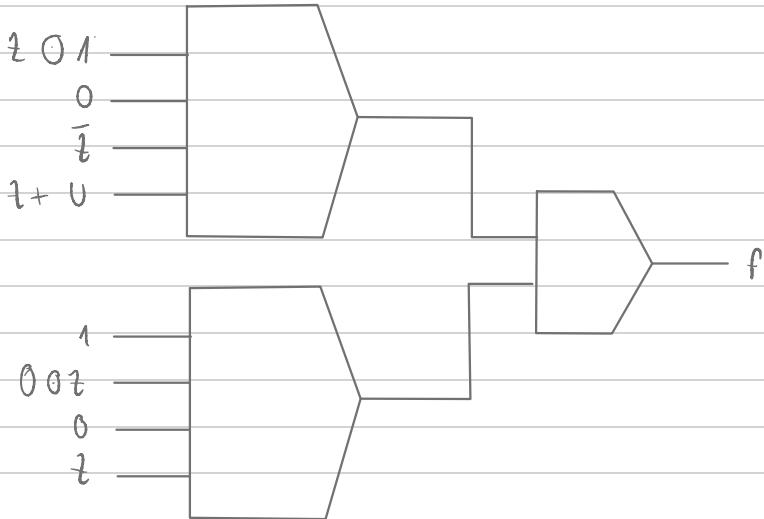
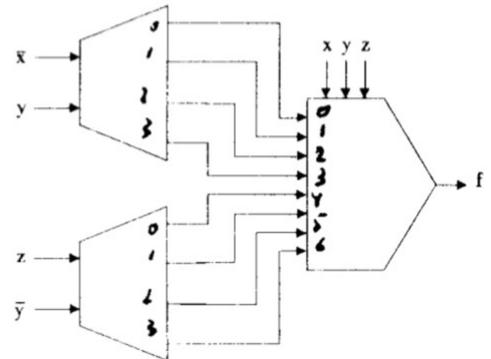
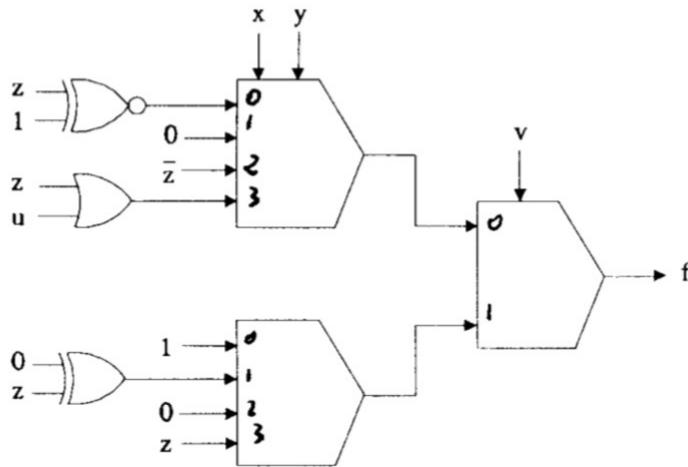
Diseñar un conversor de código BCD a BCD exceso 3 utilizando un sumador.

$$BCD_{exc\ 3} = BCD + 3$$

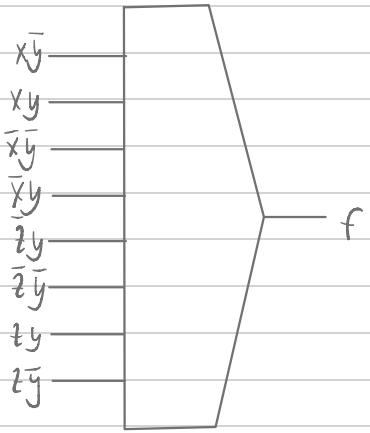


## Ejercicio 8

Obtener las funciones sintetizadas por los circuitos de la figura siguiente:



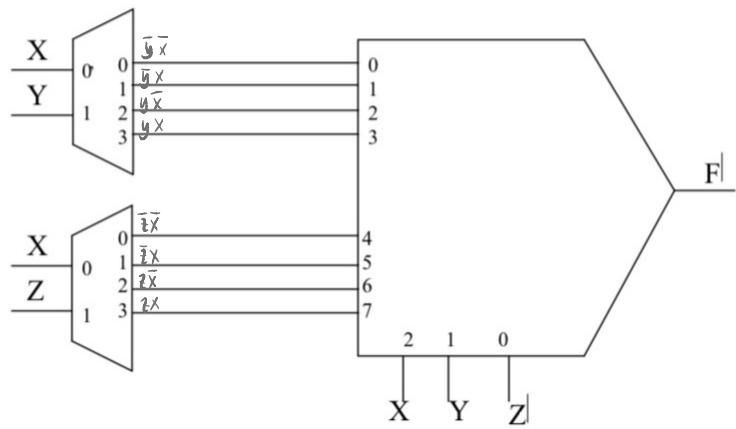
$$\begin{aligned}
 f &= ((\bar{z} \circ 1)(\bar{x}\bar{y}) + 0 + \bar{z}(x\bar{y}) + (\bar{z} + u)(xy))\bar{v} + (1(\bar{x}\bar{y}) + (0 \circ 2)(\bar{x}y) + 0 + 2(xy))v \\
 &= ((\bar{z} + \bar{x})(\bar{x}\bar{y}) + \bar{x}\bar{y}\bar{z} + (\bar{z} + \bar{z}v + u)(xy))\bar{v} + (1(\bar{x}\bar{y}) + \bar{z}\bar{v}(\bar{x}y) + 2(xy))v \\
 &= (\bar{x}\bar{y}\bar{z} + \bar{x}\bar{y} + x\bar{y}\bar{z} + xy\bar{z}v + xyv + xyz)\bar{v} + (\bar{x}\bar{y} + \bar{x}\bar{y}\bar{z} + xyz)v \\
 &= \cancel{\bar{x}\bar{y}\bar{z}\bar{v}} + \cancel{\bar{x}\bar{y}v} + \cancel{x\bar{y}\bar{z}\bar{v}} + \cancel{xy\bar{z}v} + \cancel{x\bar{y}v} + \cancel{\bar{x}\bar{y}v} + \cancel{xyz}v \\
 &= \bar{x}\bar{y} + xy\bar{z}v + x\bar{y}\bar{z} + xyv + x\bar{y}\bar{z}\bar{v}
 \end{aligned}$$



$$\begin{aligned}
 f &= (\bar{x}\bar{y})(\bar{x}\bar{y}\bar{z}) + (x\bar{y})(\bar{x}\bar{y}z) + (\bar{x}\bar{y})(\bar{x}y\bar{z}) + (\bar{x}y)(\bar{x}y\bar{z}) + (\bar{z}y)(x\bar{y}\bar{z}) + (\bar{z}\bar{y})(x\bar{y}\bar{z}) + (zy)(xy\bar{z}) + (z\bar{y})(xy\bar{z}) \\
 &= \cancel{\bar{y}\bar{z}} + \cancel{z} + \bar{x}\bar{z} + \cancel{\bar{x}y\bar{z}} + \cancel{x\bar{z}} + \cancel{x\bar{y}} + \cancel{xy} + \cancel{xz}
 \end{aligned}$$

## Ejercicio 9

Obtener la función sintetizada por el circuito de la figura siguiente:



$$\begin{aligned} f &= (\bar{y}\bar{x})(\bar{x}\bar{y}\bar{z}) + (\bar{y}x)(\bar{x}\bar{y}\bar{z}) + (y\bar{x})(\bar{x}\bar{y}\bar{z}) + (yx)(\bar{x}\bar{y}\bar{z}) + (\bar{z}\bar{x})(x\bar{y}\bar{z}) + (\bar{z}x)(x\bar{y}\bar{z}) + (z\bar{x})(x\bar{y}\bar{z}) + (zx)(x\bar{y}\bar{z}) \\ &= \bar{x}\bar{y}\bar{z} + \bar{y}z + \bar{x}y\bar{z} + yz + \bar{y}\bar{z} + x\bar{y} + y + \cancel{xyz} \\ &= \bar{x}\bar{y} + y + z + \bar{x}\bar{z} + \bar{y}\bar{z} \end{aligned}$$

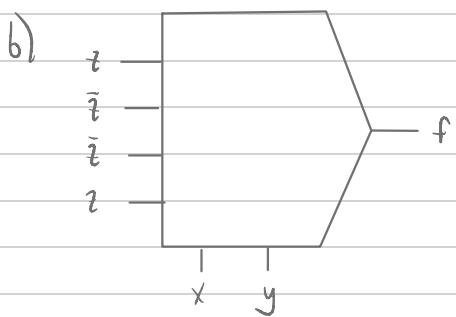
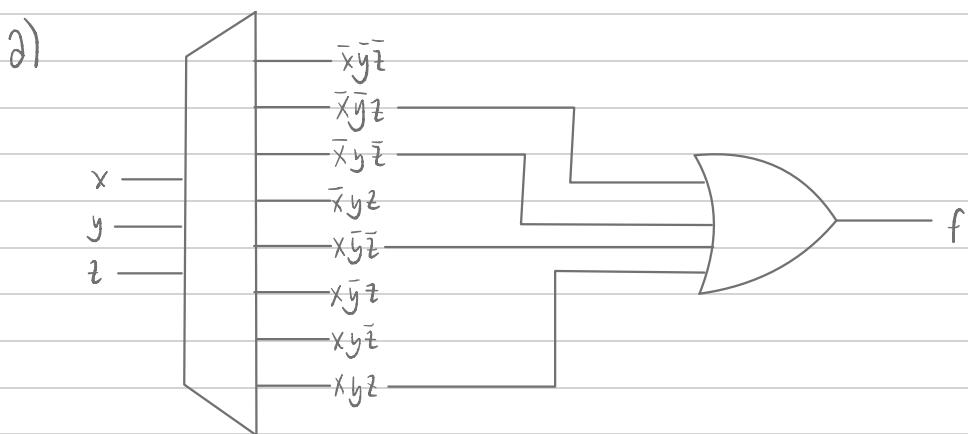
## Ejercicio 10

Implementar la función definida por la siguiente tabla de verdad empleando:

- a) Decodificadores
  - b) Multiplexores

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$x$	$y$	$t$	0	1	
0	0	0	0	1	$\bar{x}\bar{y}z$
0	1	1	1	0	$\bar{x}y\bar{z}$
1	0	0	0	1	$xy\bar{z}$
1	1	1	1	0	$xy\bar{z}$



### Ejercicio 11

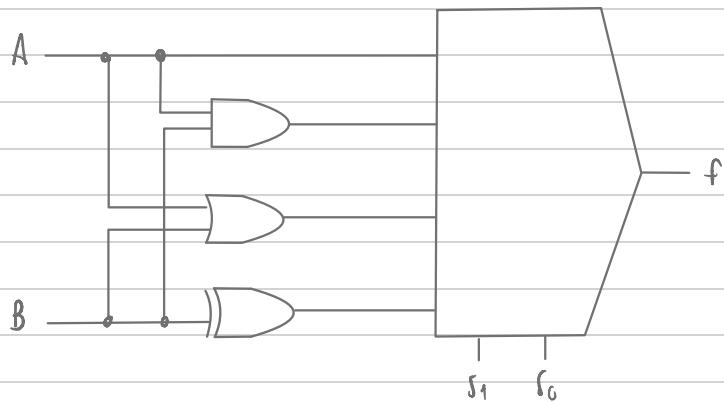
Diseñar una unidad lógica de 1 bit que realice las operaciones lógicas siguientes en función de dos señales de control  $S_1$  y  $S_0$ :

$S_1$	$S_0$	Operación
0	0	$A$
0	1	$A \cdot B$
1	0	$A + B$
1	1	$A \oplus B$

a) Realizarla con un multiplexor de 4 a 1 y las puertas necesarias.

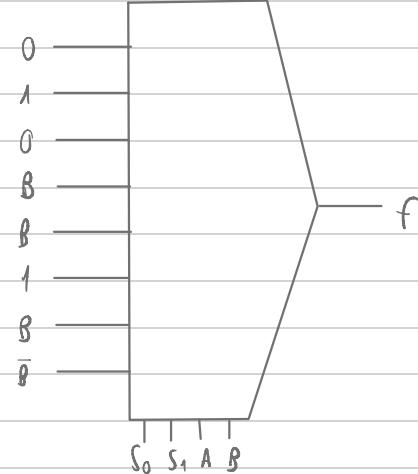
b) Realizarla con un multiplexor del tamaño necesario aplicando la técnica de implementación de funciones mediante multiplexores.

d)



b)

$S_0$	$S_1$	$A$	$B$	$F$
0	0	0	0	0 } 0
0	0	0	1	0 } 0
0	0	1	0	1 } 1
0	0	1	1	1 } 1
0	1	0	0	0 } 0
0	1	0	1	0 } 0
0	1	1	0	0 } 0
0	1	1	1	1 } 1
1	0	0	0	0 } 0
1	0	0	1	1 } 1
1	0	1	0	1 } 1
1	0	1	1	1 } 1
1	1	0	0	0 } 0
1	1	0	1	1 } 1
1	1	1	0	1 } 1
1	1	1	1	0 } 0



## Ejercicio 12

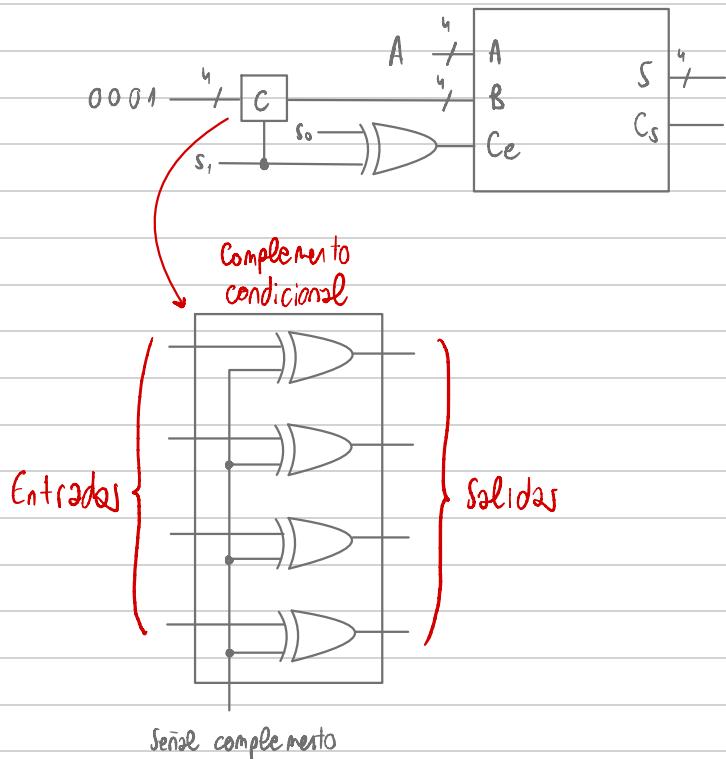
Diseñar una unidad aritmética de 4 bits partiendo de un sumador completo de cuatro bits que realice las operaciones siguientes en función de dos señales de control  $S_1$  y  $S_0$ :

$S_1$	$S_0$	Operación
0	0	$A+1$
0	1	$A+2$
1	0	$A-1$
1	1	$A-2$

$S_1$	$S_0$	$A$	$B$	$C_e$	Operación
0	0	A	1	0	$A+1$
0	1	A	1	1	$A+2 = A + 1 + 1$
1	0	A	$1_{C_1}$	1	$A-1 = A + 1_{C_1} + 1$
1	1	A	$1_{C_1}$	0	$A-2 = A - 1 - 1 = A + 1_{C_1} - 1 + 1$

↓

Puesta XOR



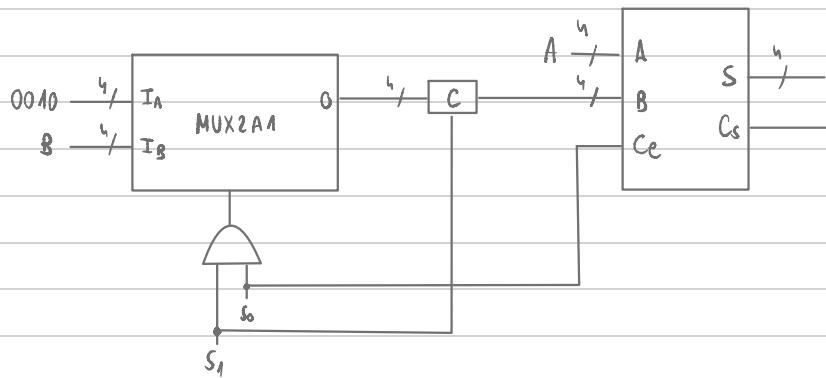
### Ejercicio 13

Diseñar una unidad aritmética de 4 bits partiendo de un sumador completo de cuatro bits que realice las operaciones siguientes en función de dos señales de control  $S_1$  y  $S_0$ :

$S_1$	$S_0$	Operación
0	0	$A + 2$
0	1	$A + 3$
1	0	$A - 3$
1	1	$A - B$

$S_1$	$S_0$	$A$	$B$	$C_e$	Operación
0	0	A	2	0	$A + 2$
0	1	A	2	1	$A + 3 = A + 2 + 1$
1	0	A	$2C_1$	0	$A - 3 = A - 2 - 1 = A + 2C_1 + 1 - 1$
1	1	A	$B_{C_1}$	1	$A - B = A - B = A + B_{C_1} + 1$

↑  
 $S_0$

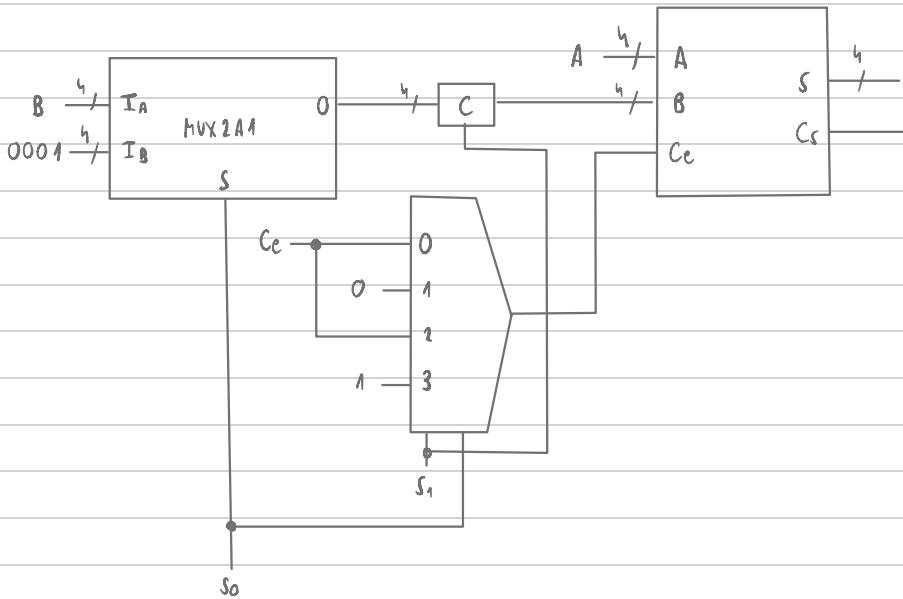


### Ejercicio 14

Diseñar una unidad aritmética de 4 bits partiendo de un sumador completo de cuatro bits que realice las operaciones siguientes en función de dos señales de control  $S_1$  y  $S_0$ :

$S_1$	$S_0$	Operación
0	0	$A + B + C_E$
0	1	$A + 1$
1	0	$A - B - \bar{C}_E$
1	1	$A - 1$

$S_1$	$S_0$	$A$	$B$	$C_E$	Operación
0	0	$A$	$B$	$C_E$	$A + B + C_E$
0	1	$A$	$1$	0	$A + 1$
1	0	$A$	$B_{C1}$	$C_E$	$A - B - \bar{C}_E = A + B_{C1} + 1 - \bar{C}_E$
1	1	$A$	$1_{C1}$	1	$A - 1 = A + 1_{C1} + 1$

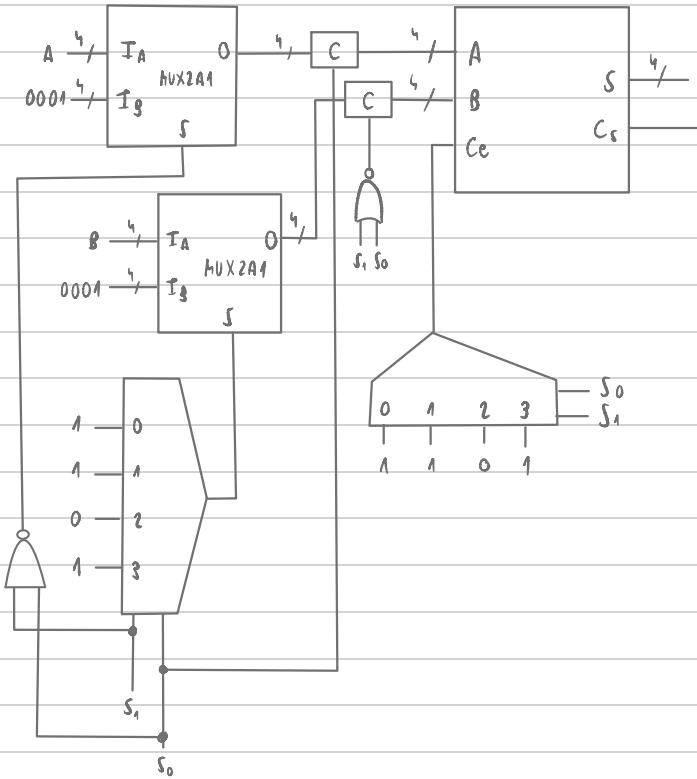


### Ejercicio 15

Diseñar una unidad aritmética de 4 bits partiendo de un sumador completo de cuatro bits que realice las operaciones siguientes en función de dos señales de control  $S_1$  y  $S_0$ :

$S_1$	$S_0$	Operación
0	0	$A - B$
0	1	$-A + B$
1	0	$A + 1$
1	1	$B - 1$

$S_1$	$S_0$	$A$	$B$	$C_e$	Operación
0	0	$A$	$B_{C_1}$	1	$A - B = A + B_{C_1} + 1$
0	1	$A_{C_1}$	$B$	1	$B - A = B + A_{C_1} + 1$
1	0	$A$	1	0	$A + 1$
1	1	$1_{C_1}$	$B$	1	$B - 1 = B - 1 = B + 1_{C_1} + 1$



## Ejercicio 16

Diseñar una unidad aritmético-lógica de 4 bits que realice las operaciones siguientes en función de tres señales de control  $S_2, S_1$  y  $S_0$ :

		Operaciones Lógicas	Operaciones Aritméticas
$S_1$	$S_0$	$S_2 = 0$	$S_2 = 1$
0	0	$A + B$	$A - 2$
0	1	$A \cdot B$	$-B + 1$
1	0	$A \oplus B$	$-A - 1$
1	1	$\overline{A \oplus B}$	$B - 1$

Realizar la unidad lógica empleando multiplexores del tamaño necesario aplicando la técnica de implementación de funciones mediante multiplexores. Para la unidad aritmética, partir de un sumador completo de cuatro bits.

Control MUX					
$S_2$	$S_1$	$A_0$	$A$	$B$	$F$
0	0	0	0	0	0 } $B$
0	0	0	0	1	1 }
0	0	0	1	0	1 }
0	0	0	1	1	1 }
0	0	1	0	0	0 } 0
0	0	1	0	1	0 }
0	0	1	1	0	0 }
0	0	1	1	1	1 }
0	1	0	0	0	0 } $B$
0	1	0	0	1	1 }
0	1	0	1	0	1 }
0	1	0	1	1	0 }
0	1	1	0	0	1 }
0	1	1	0	1	0 }
0	1	1	1	0	0 }
0	1	1	1	1	1 }

$A + B$   
 $A \cdot B$   
 $\bar{B}$   
 $A \oplus B$   
 $\overline{A \oplus B} = A \ominus B$

$S_2$	$S_1$	$S_0$	$A$	$B$	$C_e$	Operaciones
1	0	0	$A$	$\bar{A}_{C1}$	0	$A - 2 = A - 1 - 1 = A + \bar{A}_{C1} + 1 - 1 = A - \bar{A}_{C1}$
1	0	1	1	$\bar{B}_{C1}$	1	$1 - B = 1 + \bar{B}_{C1} + 1$
1	1	0	$A_{C1}$	0	0	$-A - 1 = A_{C1} + 1 - 1 = A_{C1}$
1	1	1	$\bar{A}_{C1}$	$B$	1	$B - 1 = B + \bar{A}_{C1} + 1$

