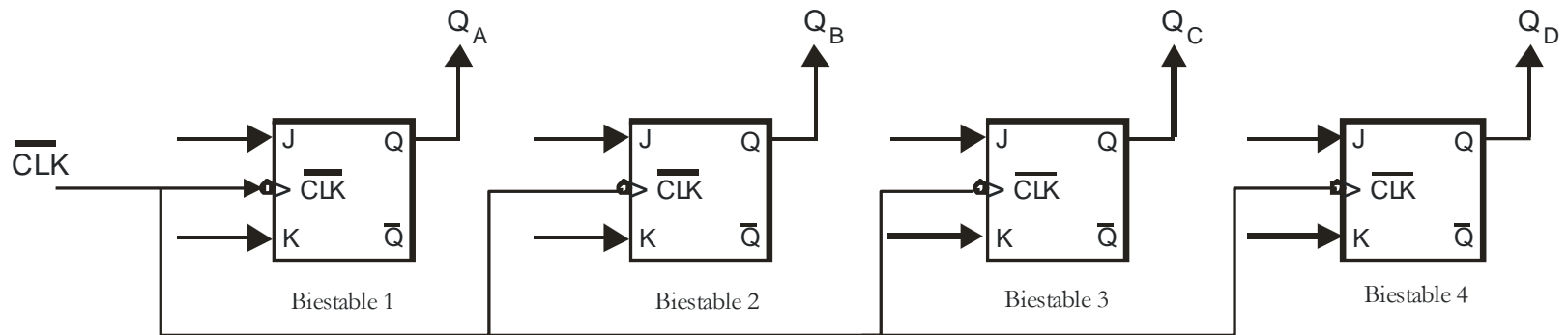


Contador Síncrono Ascendente J-K

Fundamentos de tecnología de
computadores

Contador síncrono ascendente



Transición			
Q_t	Q_{t+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Contador síncrono ascendente

t				t+1				B4		B3		B2		B1	
QD	QC	QB	QA	QD	QC	QB	QA	J	K	J	K	J	K	J	K
0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X
0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1
0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
0	0	1	1	0	1	0	0	0	X	1	X	X	1	X	1
0	1	0	0	0	1	0	1	0	X	X	0	0	X	1	X
0	1	0	1	0	1	1	0	0	X	X	0	1	X	X	1
0	1	1	0	0	1	1	1	0	X	X	0	X	0	1	X
0	1	1	1	1	0	0	0	1	X	X	1	X	1	X	1
1	0	0	0	1	0	0	1	X	0	0	X	0	X	1	X
1	0	0	1	1	0	1	0	X	0	0	X	1	X	X	1
1	0	1	0	1	0	1	1	X	0	0	X	X	0	1	X
1	0	1	1	1	1	0	0	X	0	1	X	X	1	X	1
1	1	0	0	1	1	0	1	X	0	X	0	0	X	1	X
1	1	0	1	1	1	1	0	X	0	X	0	1	X	X	1
1	1	1	0	1	1	1	1	X	0	X	0	X	0	1	X
1	1	1	1	0	0	0	0	X	1	X	1	X	1	X	1

Contador síncrono ascendente

J4

		Qb Qa			
Qd Qc		00	01	11	10
		0	1	3	2
00					
01				1	
11		X	X	X	X
10		X	X	X	X

K4

		Qb Qa			
Qd Qc		00	01	11	10
		X	X	X	X
00		X	X	X	X
01		X	X	X	X
11				1	
10					

J3

		Qb Qa			
Qd Qc		00	01	11	10
		0	1	3	2
00				1	
01		X	X	X	X
11		X	X	X	X
10				1	

K3

		Qb Qa			
Qd Qc		00	01	11	10
		X	X	X	X
00		X	X	X	X
01				1	
11				1	
10		X	X	X	X

J2

		Qb Qa			
Qd Qc		00	01	11	10
		0	1	3	2
00			1	X	X
01			1	X	X
11			1	X	X
10			1	X	X

K2

		Qb Qa			
Qd Qc		00	01	11	10
		X	X	1	
00		X	X	1	
01		X	X	1	
11		X	X	1	
10		X	X	1	

$$J4 = Qc \cdot Qb \cdot Qa$$

$$K4 = Qc \cdot Qb \cdot Qa$$

$$J2 = Qa$$

$$K2 = Qa$$

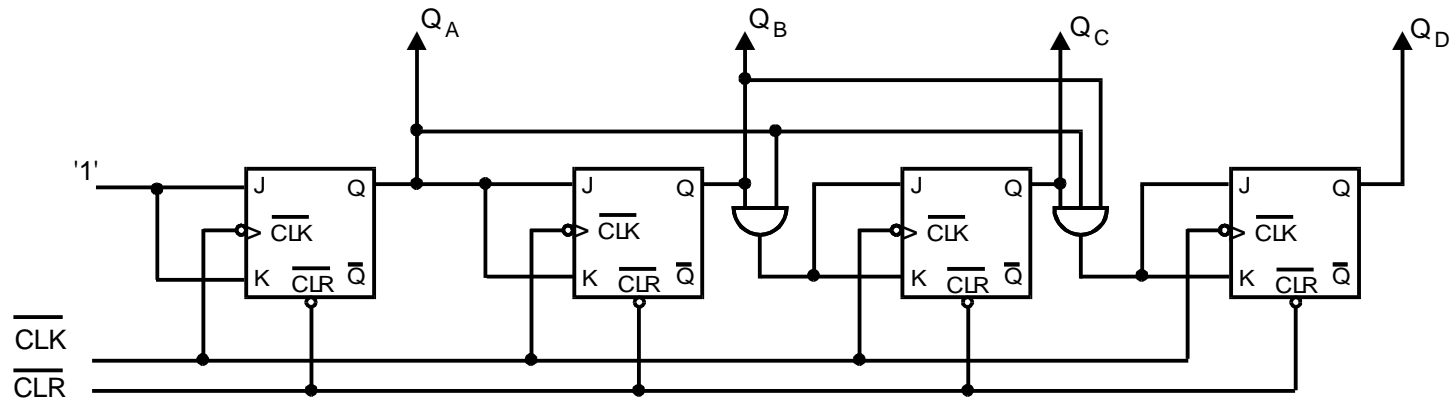
$$J3 = Qb \cdot Qa$$

$$K3 = Qb \cdot Qa$$

$$J1 = 1$$

$$K1 = 1$$

Contador síncrono ascendente



PROBLEMAS DE FUNDAMENTOS DE TECNOLOGÍA DE COMPUTADORES

2. SISTEMAS DE NUMERACIÓN Y OPERACIONES

EJERCICIO 1:

Se tiene un ordenador que representa los números enteros con 16 bits en signo-magnitud: 1 bit para el signo y 15 para la magnitud.

- a) Calcular el rango de representación para ese formato.*
- b) Representar en dicho formato los números 24 y -24*

EJERCICIO 2:

Se tiene un ordenador que representa los números enteros con 24 bits, representados en complemento a 1.

- a) Calcular el rango de representación para ese formato.*
- b) Representar en dicho formato los números 37 y -214*

EJERCICIO 3:

Se tiene un ordenador con los siguientes formatos de representación:

- *Números enteros con 8 bits, representados en complemento a 2.*
- a) Calcular el rango de representación para los números enteros.*
- c) Representar en dicho formato los números 235 y -144*

EJERCICIO 4:

Dado el número de 8 bits **1010 1011**

- a) Calcular el valor de dicho n° suponiendo que esta representado en Binario puro.
 - b) Calcular el valor de dicho n° suponiendo que esta representado en Signo Magnitud.
 - c) Calcular el valor de dicho n° suponiendo que esta representado en Complemento a 1.
 - d) Calcular el valor de dicho n° suponiendo que esta representado en Complemento a 2.
-

EJERCICIO 5:

Sean los números representados en C1 siguientes. $A = 0100\ 0110$ $B = 1111\ 1000$.

- a) Calcular $A + B$
 - b) Calcular $A - B$
 - c) Calcular $B - A$
 - d) ¿Se produce desbordamiento en algún caso?
-

EJERCICIO 6:

Sean los números representados en C2 siguientes. $A = 0100\ 0110$ $B = 1111\ 1000$.

- a) Calcular $A + B$
 - b) Calcular $A - B$
 - c) Calcular $B - A$
 - d) ¿Se produce desbordamiento en algún caso?
-

EJERCICIO 7:

Sea $A = 0110\ 1001$. Se pide cambiar de signo el número suponiendo que:

- a) El número está representado en binario puro
 - b) El número está representado en signo-magnitud
 - c) El número está representado en C1
 - d) El número está representado en C2
-

EJERCICIO 8:

Sean $A = 1AF7h$ y $B = FA59h$. Realizar la suma de $A + B$

PROBLEMAS DE FUNDAMENTOS DE TECNOLOGÍA DE COMPUTADORES

2. SISTEMAS DE NUMERACIÓN Y OPERACIONES

SOLUCIONES

EJERCICIO 1:

Se tiene un ordenador que representa los números enteros con 16 bits en signo-magnitud: 1 bit para el signo y 15 para la magnitud.

a) Calcular el rango de representación para ese formato.

$$[-(2^{15}-1) ; 0 ; 2^{15}-1]$$

b) Representar en dicho formato los números:

- 24

0	000 0000 0001 1000
---	--------------------

- -24

1	000 0000 0001 1000
---	--------------------

EJERCICIO 2

Se tiene un ordenador que representa los números enteros con 24 bits, representados en complemento a 1.

a) Calcular el rango de representación para ese formato.

$$[-(2^{23}-1) ; 0 ; 2^{23}-1]$$

b) Representar en dicho formato los números:

- 37

0000 0000 0000 0000 0010 0101

- -214

1111 1111 1111 1111 0010 1001

EJERCICIO 3:

Se tiene un ordenador con los siguientes formatos de representación:

- Números enteros con 8 bits, representados en complemento a 2.

a) Calcular el rango de representación para los números enteros.

$$[-2^7 ; -1 ; 0 ; 2^7-1]$$

c) Representar en dicho formato los números:

- 235. **No se puede representar, puesto que excede del rango de representación**
- -144 **No se puede representar, puesto que excede del rango de representación**

EJERCICIO 4:

Dado el número de 8 bits **1010 1011**

- a) Calcular el valor de dicho n° suponiendo que esta representado en Binario puro: **171**
- b) Calcular el valor de dicho n° suponiendo que esta representado en Signo Magnitud: **- 43**
- c) Calcular el valor de dicho n° suponiendo que esta representado en Complemento a 1: **- 84**
- d) Calcular el valor de dicho n° suponiendo que esta representado en Complemento a 2: **- 85**

EJERCICIO 5:

Sean los números representados en C1 siguientes. $A = 0100\ 0110$ $B = 1111\ 1000$.

- a) Calcular $A + B$
- b) Calcular $A - B$
- c) Calcular $B - A$
- d) ¿Se produce desbordamiento en algún caso?

a)

$$\begin{array}{r}
 1 1 \\
 0 1 0 0 0 1 1 0 \\
 1 1 1 1 1 0 0 0 + \\
 \hline
 1 0 0 1 1 1 1 1 0 \\
 0 0 1 1 1 1 1 1 +
 \end{array}$$

b)

$$-B = C1(B) = 0000\ 0111$$

$$\begin{array}{r}
 1 1 \\
 0 1 0 0 0 1 1 0 \\
 0 0 0 0 0 1 1 1 + \\
 \hline
 0 1 0 0 1 1 0 1
 \end{array}$$

c)

$$-A = C1(A) = 1011\ 1001$$

$$\begin{array}{r}
 1 1 1 1 1 \\
 1 0 1 1 1 0 0 1 \\
 1 1 1 1 1 0 0 0 + \\
 \hline
 1 1 0 1 1 0 0 0 1 \\
 1 0 1 1 0 0 1 0
 \end{array}$$

d) En ningún caso

EJERCICIO 6:

Sean los números representados en C2 siguientes. $A = 0100\ 0110$ $B = 1111\ 1000$.

Se pide:

- Calcular $A + B$
- Calcular $A - B$
- Calcular $B - A$
- ¿Se produce desbordamiento en algún caso?

a)

$$\begin{array}{r}
 1 1 \\
 0 1 0 0 0 1 1 0 \\
 1 1 1 1 1 0 0 0 + \\
 \hline
 \neq 0 0 1 1 1 1 1 1 0
 \end{array}$$

b)

$$-B = C2(B) = 0000\ 1000$$

$$\begin{array}{r}
 0 1 0 0 0 1 1 0 \\
 0 0 0 0 1 0 0 0 + \\
 \hline
 0 1 0 0 1 1 1 0
 \end{array}$$

c)

$$-A = C1(A) = 1011\ 1010$$

$$\begin{array}{r}
 1 1 1 1 1 \\
 1 0 1 1 1 0 1 0 \\
 1 1 1 1 1 0 0 0 + \\
 \hline
 \neq 1 0 1 1 0 0 1 0
 \end{array}$$

d) En ningún caso

EJERCICIO 7:

Sea $A = 1000\ 1010$. Se pide cambiar de signo el número suponiendo que:

- a) El número está representado en binario puro
- b) El número está representado en signo-magnitud
- c) El número está representado en C1
- d) El número está representado en C2

A=	1	0	0	0	1	0	1	0
a)	No se puede							
b)	0	0	0	0	1	0	1	0
c)	0	1	1	1	0	1	0	1
d)	0	1	1	1	0	1	1	0

EJERCICIO 8:

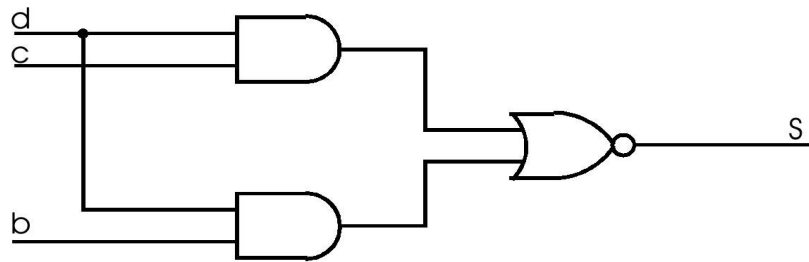
Sean $A = 1AF7h$ y $B = FA59h$. Se pide:

- a) Realizar la suma de $A + B$

1	1	1	1		
	1	A	F	7	
	F	A	5	9	+
<hr/>					
1	1	5	5	0	

Tema 3: Sistemas Combinacionales

1. Analizar el siguiente circuito indicando la expresión algebraica que implementa, la tabla de verdad correspondiente y la función lógica en sus dos formas canónicas



2. Expresar en forma de *minterms* las siguientes funciones:

a)- $F(c,b,a) = \overline{((c+\bar{b}) \cdot \bar{c} + b + a + c \cdot b)}$

b)- $F(d,c,b,a) = (d + \bar{b}) \cdot \bar{c} + b + \bar{a}$

3. Convertir la siguiente función a su primera forma normal

- $F(a,b,c) = a \cdot b + c + a \cdot \bar{c} + \bar{a} \cdot b \cdot c$

4. Simplificar por el método de Karnaugh las siguientes funciones:

a)- $F(d,c,b,a) = \sum (0,1,4,5,6,8,9,13,14)$

b)- $F(d,c,b,a) = \sum (0,1,2,4,5,8,10)$

c)- $F(d,c,b,a) = \sum (0,1,3,4,5,7,8,9,14,15)$

d)- $F(d,c,b,a) = \sum (1,2,3,5,6,7,8,9,10,11,14)$

5. Simplificar la siguiente función por los métodos conocidos:

$F(d,c,b,a) = (\sum (0,2,5,7,8,10,13,15))$

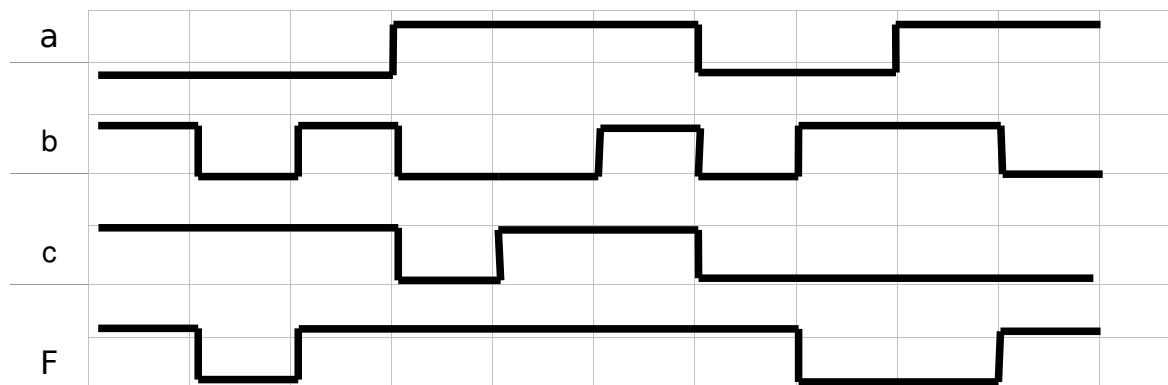
6. Diseñar un circuito compuesto por puertas lógicas AND y OR de cuatro entradas y dos salidas definido por las funciones siguientes:

a)- $F1(d,c,b,a) = \sum(0,1,4,5,6,8,9,13,14)$ (misma que 4.a)

b)- $F2(d,c,b,a) = \sum(0,1,2,4,5,8,10,13,14)$

c)- Rediseñarlos con puertas NAND exclusivamente

7. Dado el siguiente diagrama de tiempos para las señales de entrada a,b y c, y la de salida F, Obtener la expresión lógica más simple de F utilizando los diferentes métodos de simplificación conocidos.



8. Diseñar un decodificador de tres entradas que permita representar en un *display* de 7 segmentos el valor en binario puro de dichas entradas. (Hacer la tabla de verdad, obtener la expresión en minterms/maxterms para cada segmento - Fa, Fb..Fg-, simplificarlas y hacer los circuitos).
9. Diseñar un circuito que discrimine si una entrada de 4 bits representa o no un dígito BCD válido
10. Diseñar un circuito que sume dos números BCD natural y nos dé el resultado en código binario de 5 bits. Se pueden usar sumadores BCD, sumadores binarios de 4 bits y las puertas necesarias.
11. Dados dos números naturales de dos bits cada uno A (a2 a1) y B (b2 b1) diseñar un sistema combinacional que obtenga el valor absoluto de la diferencia entre ellos |A-B|.
12. Diseñar un circuito que sume dos números naturales de dos bits A (a2 a1) y B (b2 b1) proporcionando la salida en tres bits.
13. Diseñar un circuito que compare dos números naturales de dos bits A (a2 a1) y B (b2 b1) y proporcione las siguientes salidas:
- a)- En función de las entradas A y B:

- $S1 = 1$ si $A > B$ y 0 en cualquier otro caso
- $S2 = 1$ si $A = B$ y 0 en cualquier otro caso
- $S3 = 1$ si $A < B$ y 0 en cualquier otro caso

b)- Como una variante al diseño anterior, obtener $S2$ a partir de $S1$ y $S3$

14. Utilizando multiplexores y las puertas lógicas necesarias integrar los circuitos diseñados anteriormente en un único circuito combinacional: este ha de tener igualmente dos entradas A ($a_2 a_1$) y B ($b_2 b_1$), y 3 salidas ($S3$, $S2$ y $S1$) de datos pero además tendrá dos entradas de control $C2$ $C1$ que deberán seleccionar el tipo de funcionamiento del circuito:

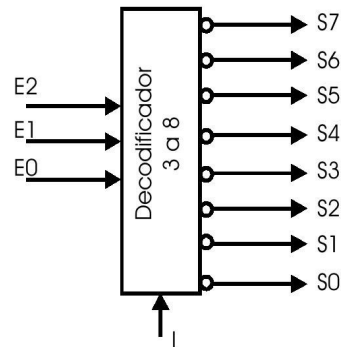
- si $C2 = 0$ y $C1 = 0$ --> las salidas $S = 111$
- si $C2 = 1$ y $C1 = 0$ --> las salidas mostrarán la suma de A y B (circuito ya diseñado en un problema anterior)
- si $C2 = 0$ y $C1 = 1$ --> las salidas mostrarán la comparación de A y B (circuito ya diseñado en un problema anterior)
- si $C2 = 1$ y $C1 = 1$ --> las salidas $S = 000$

15. Diseñar mediante puertas lógicas un circuito que tenga por entrada un número binario de 4 bits X (d, c, b, a) que realice las siguientes operaciones de salida:

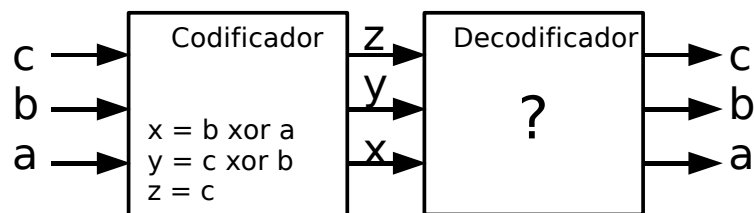
- si $X > 9$, se activa una línea de salida $S1$ que enciende una luz roja
- si $X < 9$, se activa una línea de salida $S2$ que enciende una luz verde
- si $X = 9$, se activa una línea de salida $S3$ que enciende una luz ámbar

16. Realizar un circuito que ante una entrada de 8 bits indique si esta información tiene paridad par o impar.

17. A partir de comparadores 7485 de números de 4 bits, realizar un comparador de magnitudes de 32 bits.
18. Dados dos decodificadores 3 a 8 como el de la figura, constrúyase un decodificador de 4 a 16.



19. Realizar un convertidor de código BCD natural a un código BCD con exceso a tres.
- con puertas lógicas
 - con circuitos multiplexores
20. Construir un decodificador para visualizar números binarios de 3 bits con un display 7 segmentos.
21. El bloque codificador de la figura es un circuito combinacional que realiza una codificación de las señales de entrada (a,b,c) según las ecuaciones siguientes:

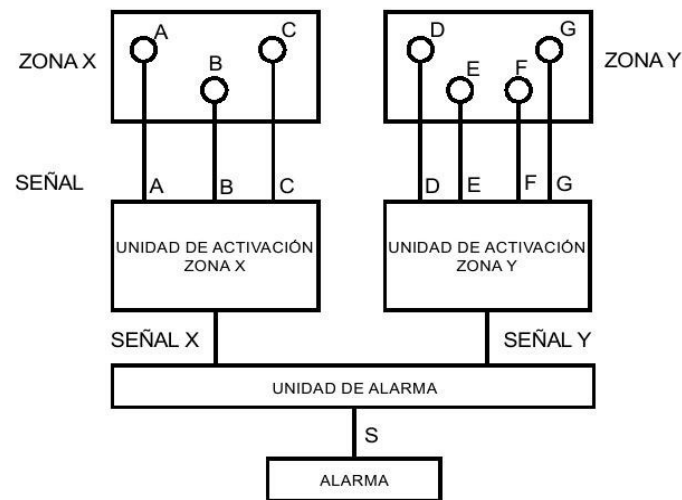


- $x = b \text{ xor } a$
- $z = c$ siendo c y z los bits más significativos

Se pide:

- Obtener la salida del codificador
- Diseñar el circuito decodificador de modo que permita obtener de nuevo el código original

22. Un banco desea instalar un sistema de alarma dotado de sensores de proximidad por rayos infrarrojos. Existen dos zonas de seguridad X e Y y la alarma de seguridad debe dispararse cuando se active cualquiera de ellas. La zona X tiene 3 sensores. A, B y C, mientras que la zona Y tiene 4 sensores: D, E, F y G. Para evitar falsas alarmas producidas por el disparo aleatorio de algunos sensores, el sistema activará cuando bien en la zona X o bien en la zona Y se activen al menos 2 sensores simultáneamente. Diseñar el circuito de control con la función más sencilla obtenida. rediseñar con puertas NOR únicamente.



Ejercicios Leyes de de-Morgan.

Transformar mientras sea posible.

1- $\overline{\overline{(A+BC)} + D(\overline{E+F})}$

2- $\overline{\overline{(A+B+C)} D}$

3- $\overline{(ABC + DFE)}$

4- $\overline{(A\overline{B} + \overline{C}D + EF)}$

5- $\overline{(\overline{ABC} + D + E)}$

6- $\overline{\overline{(\overline{A+B})} + \overline{C}}$

7- $\overline{(\overline{A+B+CD})}$

8- $\overline{((A+B)\overline{CD} + E + \overline{F})}$

9- $\overline{(\overline{AB} (C + \overline{D}) + E)}$

Ejercicios Álgebra de Boole:

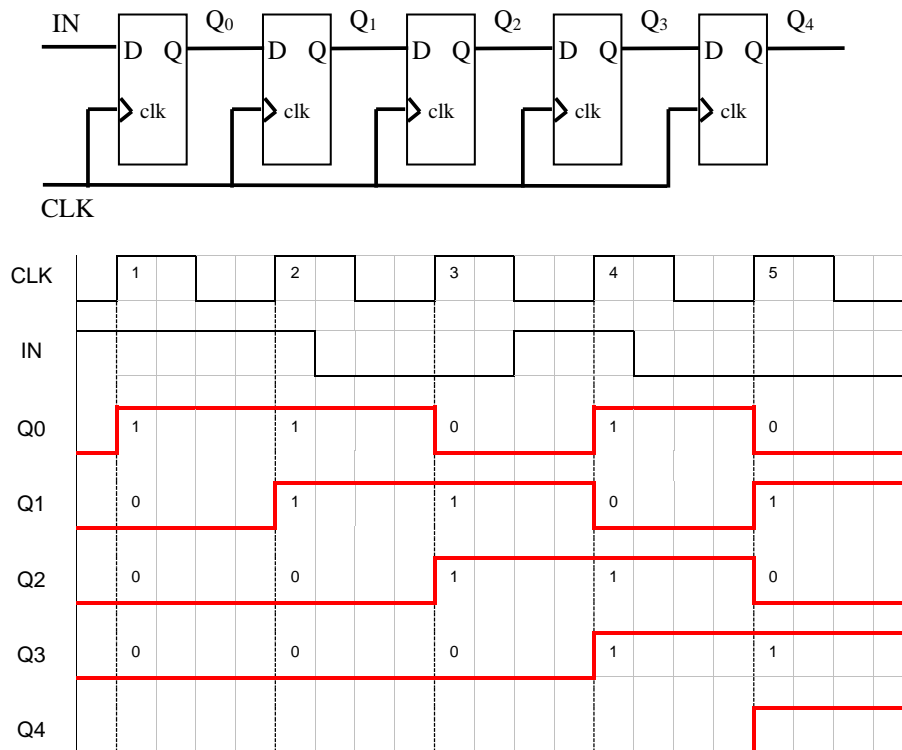
Reducir algebraicamente

Expresión	Solución
1- $AB + A(B+C) + B(B+C)$	$B + AC$
2- $A\overline{B} + A(\overline{B+C}) + B(\overline{B+C})$	$A\overline{B}$
3- $(A\overline{B} (C+BD) + \overline{A}\overline{B})C$	$\overline{B}C$
4- $CD[AB(C + \overline{BD}) + \overline{AB}]$	CD
5- $\overline{A}BC + A\overline{B}\overline{C} + \overline{A}\overline{B}\overline{C} + A\overline{B}C + ABC$	$BC + A\overline{B} + \overline{C}\overline{B}$
6- $ABC\overline{C} + \overline{A}\overline{B}C + \overline{A}BC + \overline{A}\overline{B}\overline{C}$	$ABC\overline{C} + \overline{A}C + \overline{A}\overline{B}$
7- $\overline{(AB+AC)} + \overline{A}\overline{B}C$	$\overline{A} + \overline{B}C$
8- $\overline{A}\overline{B} + \overline{A}C + \overline{A}\overline{B}\overline{C}$	$\overline{A} + \overline{B} + \overline{C}$

Soluciones de Ejercicios de cronogramas de Sistemas Digitales Secuenciales

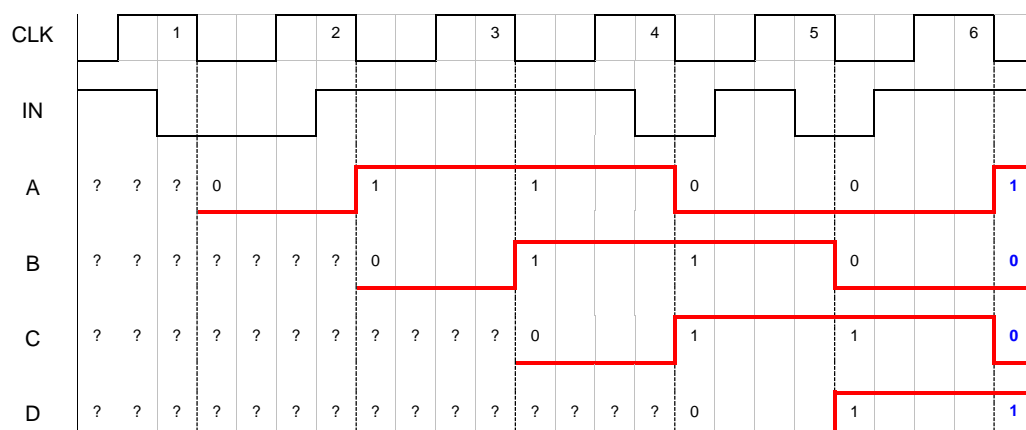
Ejercicio 4:

Obtener los estados del registro de 5 bits mostrado (inicialmente 0000), para las señales de reloj (CLK) y entrada de datos (IN) indicadas



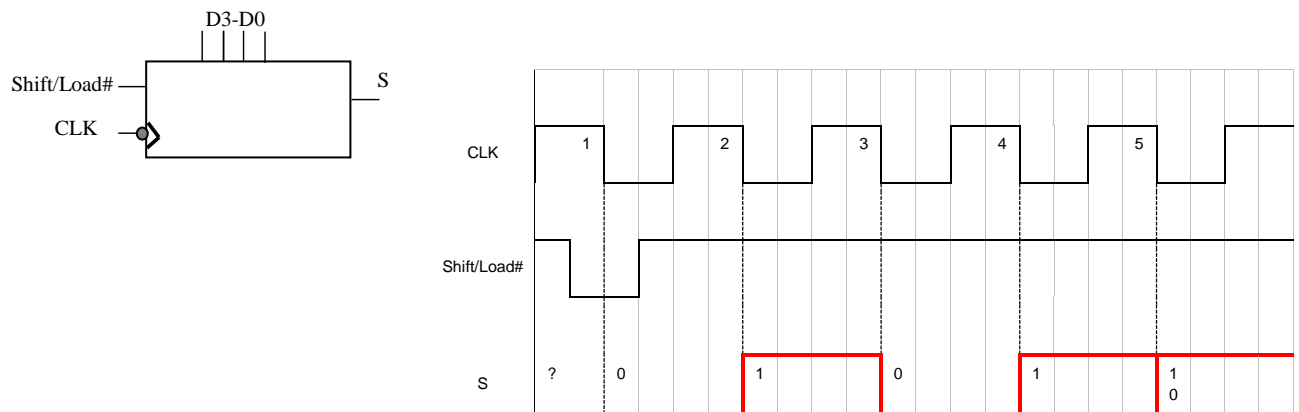
Ejercicio 5:

Hacer un diagrama de tiempos de un registro de desplazamiento con entrada serie y 4 salidas paralelas (A-B-C-D), suponiendo que está compuesto de biestables tipo D con entrada de reloj activa en flanco de bajada y que la entrada serie aplicada es la mostrada a continuación. ¿Qué dato se obtendría tras 6 ciclos de reloj?

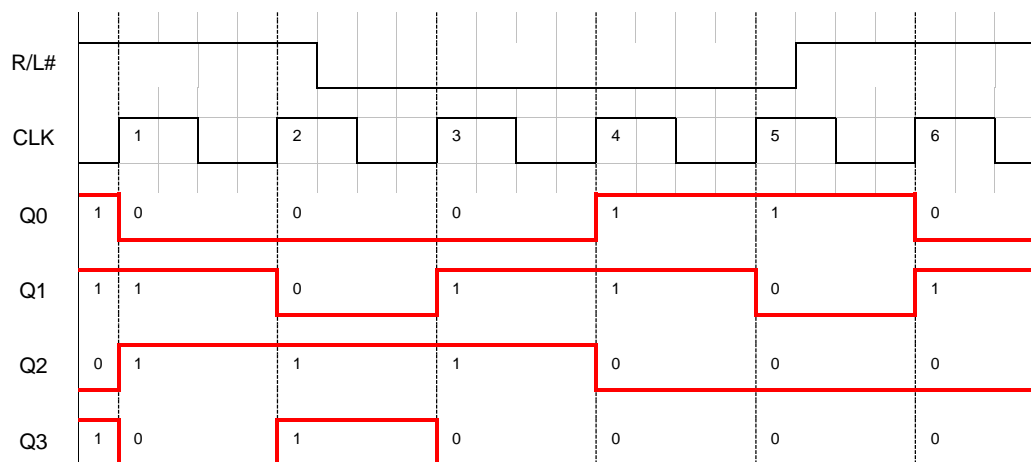


Ejercicio 6:

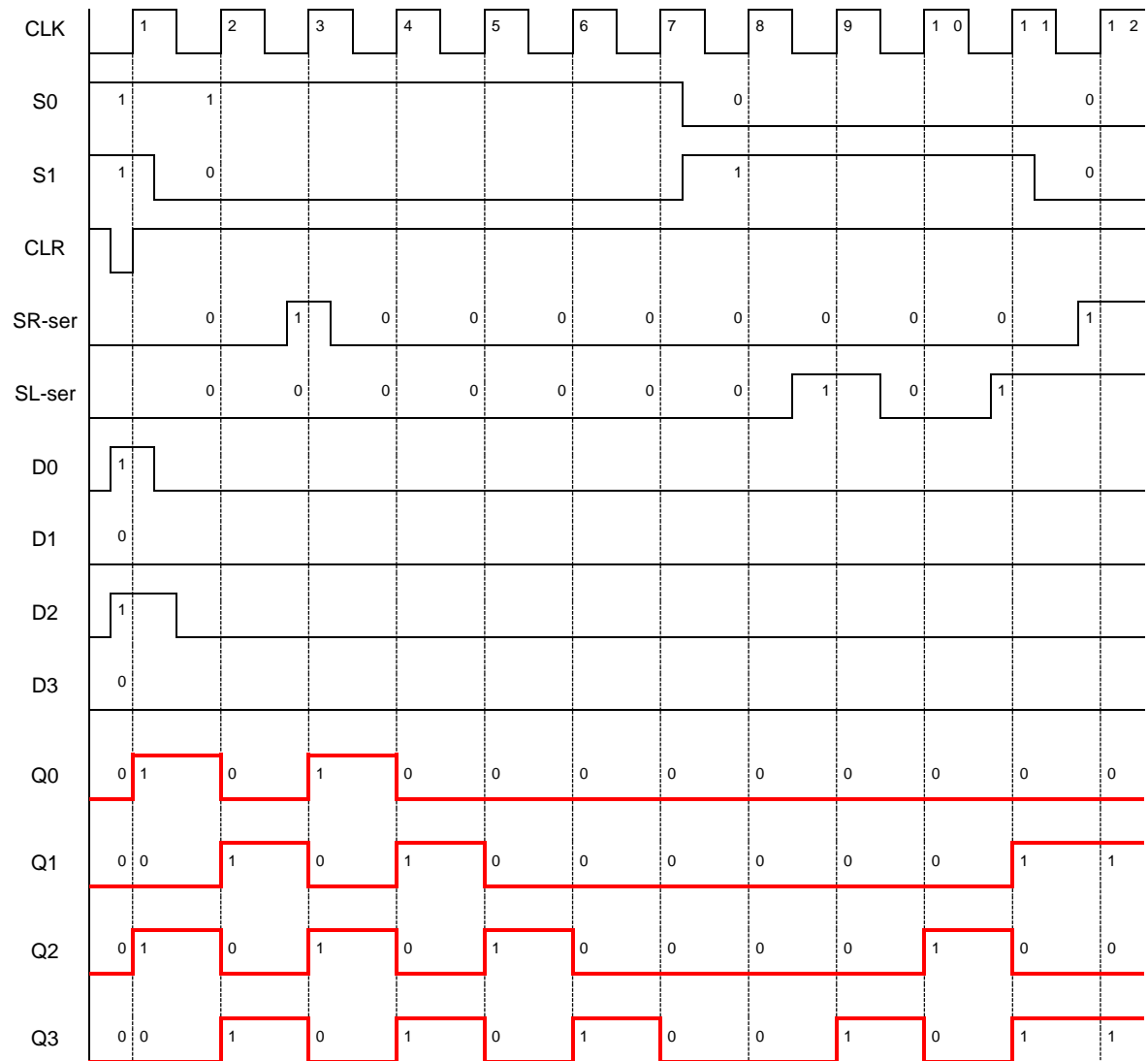
Determinar la forma de onda de la salida de datos de un registro entrada paralela / salida serie para las señales de entrada mostradas en la figura y suponiendo que las entradas D3-D0 tienen los datos 1010.

**Ejercicio 7:**

Determinar el estado en cada pulso de reloj de un registro de desplazamiento bidireccional de 4 bits en el que la entrada de datos está en nivel BAJO, si se aplica la señal de control RIGHT/LEFT# indicada. Suponer que el estado inicial del registro es Q0-Q3 = 1101 y que la sincronización es con el flanco de subida del reloj.

**Ejercicio 8:**

Determinar el estado en cada pulso de reloj de un registro de desplazamiento universal bidireccional de 4 bits en el que se aplican las señales de entrada y de control mostradas (sincronización con flanco de subida del reloj). Suponer que solo CLR es asíncrona y que las señales de control S0-S1 codifican el siguiente comportamiento: S0=S1=1 : Carga paralela (D0-D3); S0=S1=0 no hay cambios; S0=0 y S1=1 desplazamiento a la izquierda y S0=1 y S1=0 desplazamiento a la derecha. SR-ser y SL-ser son las entradas de datos serie derecha e izquierda respectivamente.



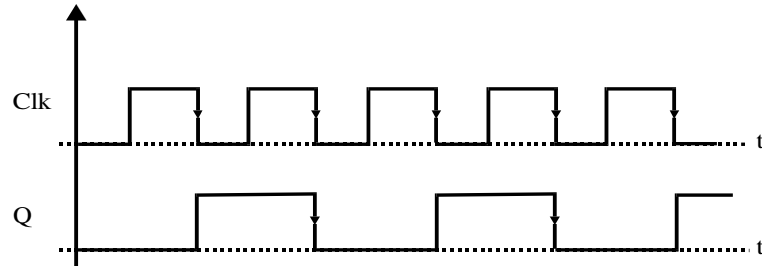
Sistemas Digitales Secuenciales

Ejercicio 1:

Siendo X un número binario de tres bits, diseñar la entrada lógica necesaria para poner a '1' un biestable J-K si X tiene un número impar de unos y ponerle a '0' en caso contrario.

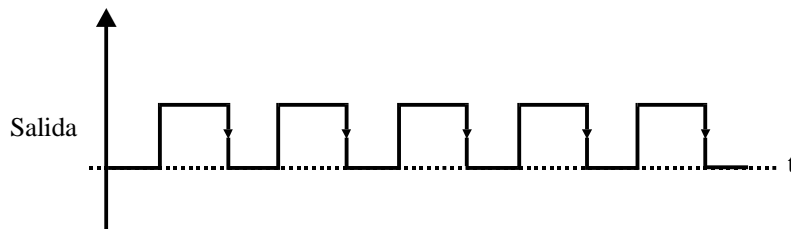
Ejercicio 2:

Se dispone del siguiente cronograma. Identifíquese que tipo de biestable se ha empleado y explique por qué.



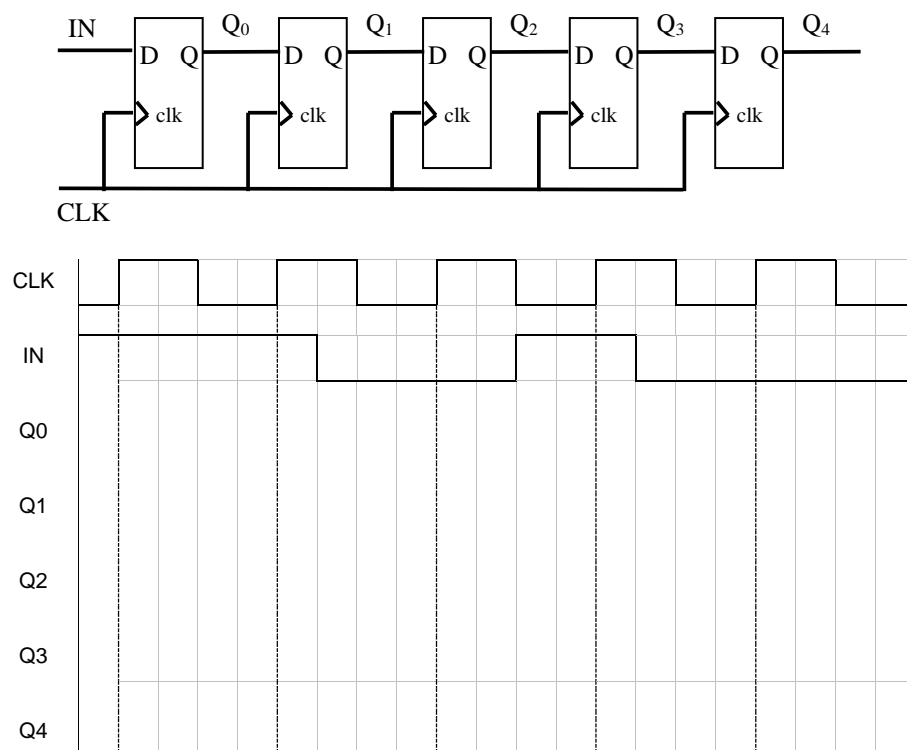
Ejercicio 3:

Se desea un circuito secuencial que tenga como señal de salida la representada en la figura. Diseñese dicho circuito digital a base de biestables y puertas lógicas, sabiendo que se trata de un divisor por 4.



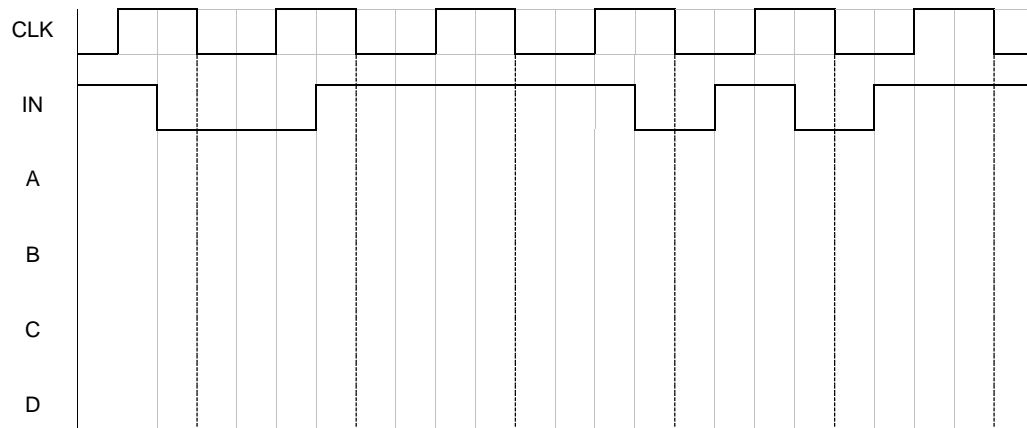
Ejercicio 4:

Obtener los estados del registro de 5 bits mostrado (inicialmente 0000), para las señales de reloj (CLK) y entrada de datos (IN) indicadas

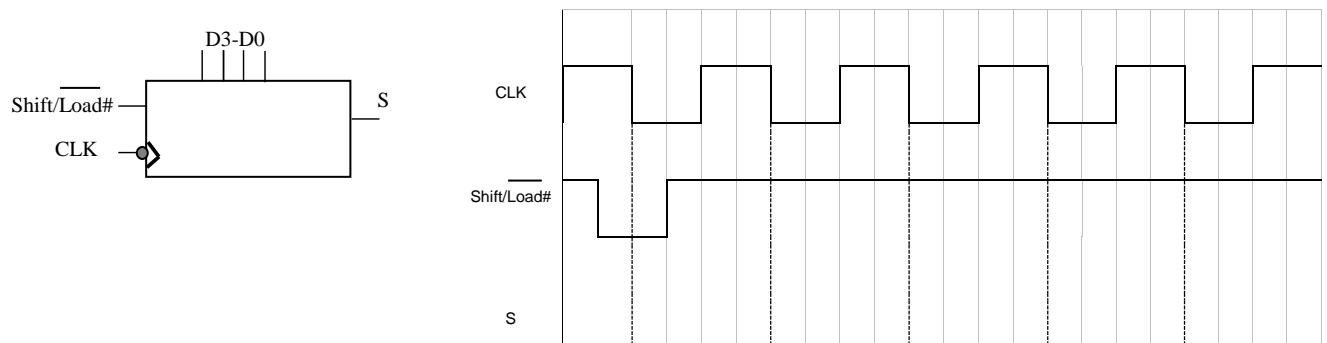


Ejercicio 5:

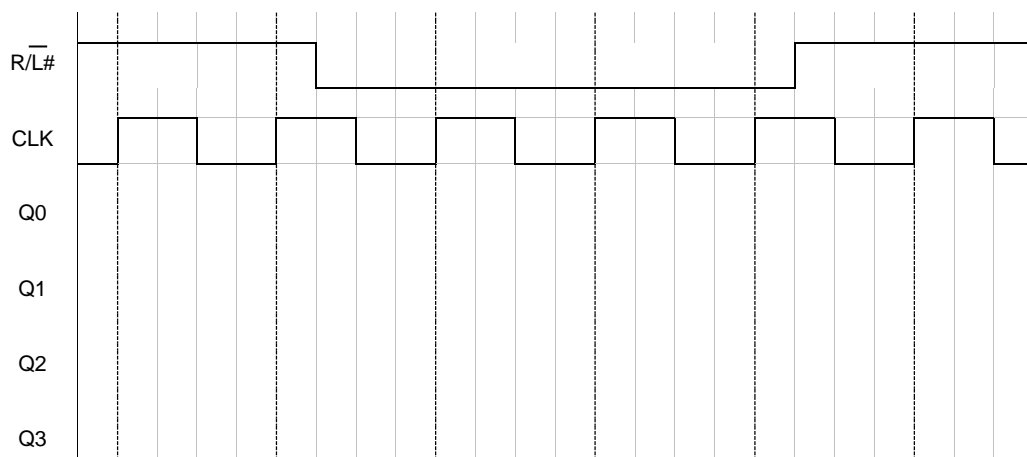
Hacer un diagrama de tiempos de un registro de desplazamiento con entrada serie y 4 salidas paralelas (A-B-C-D), suponiendo que está compuesto de biestables tipo D con entrada de reloj activa en flanco de bajada y que la entrada serie aplicada es la mostrada a continuación. ¿Qué dato se obtendría tras 6 ciclos de reloj?

**Ejercicio 6:**

Determinar la forma de onda de la salida de datos de un registro entrada paralela / salida serie para las señales de entrada mostradas en la figura y suponiendo que las entradas D3-D0 tienen los datos 1010.

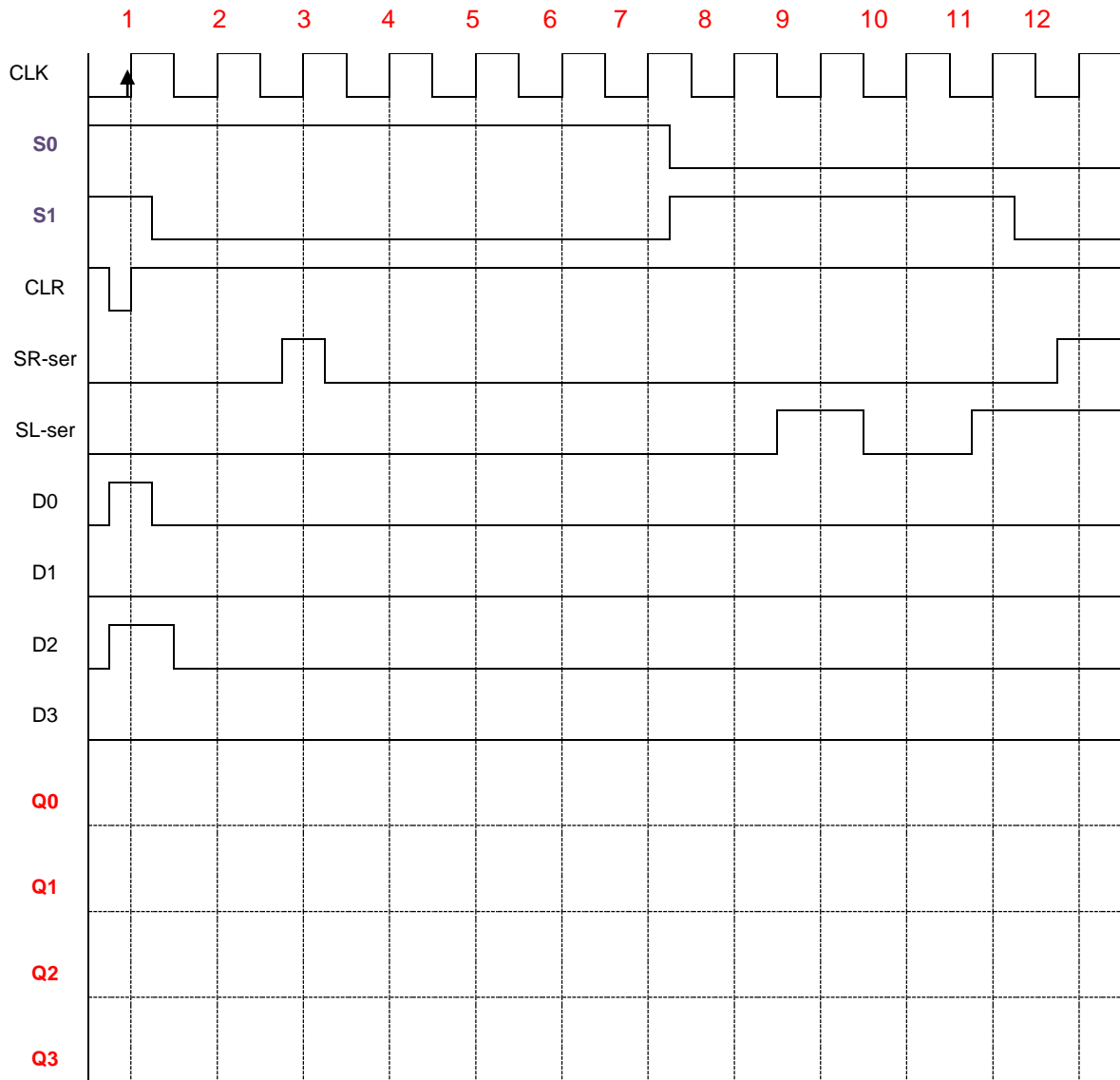
**Ejercicio 7:**

Determinar el estado en cada pulso de reloj de un registro de desplazamiento bidireccional de 4 bits en el que la entrada de datos está en nivel BAJO, si se aplica la señal de control RIGHT/LEFT# indicada. Suponer que el estado inicial del registro es Q0-Q3 = 1101 y que la sincronización es con el flanco de subida del reloj.



Ejercicio 8:

Determinar el estado en cada pulso de reloj de un registro de desplazamiento universal bidireccional de 4 bits en el que se aplican las señales de entrada y de control mostradas (sincronización con flanco de subida del reloj). Suponer que solo CLR es asíncrona y que las señales de control S0-S1 codifican el siguiente comportamiento: S0=S1=1 : Carga paralela (D0-D3); S0=S1=0 no hay cambios; S0=0 y S1=1 desplazamiento a la izquierda y S0=1 y S1=0 desplazamiento a la derecha. SR-ser y SL-ser son las entradas de datos serie derecha e izquierda respectivamente.

**Ejercicio 9:**

Diseñar un contador síncrono ascendente de módulo 4 con biestables D

Ejercicio 10:

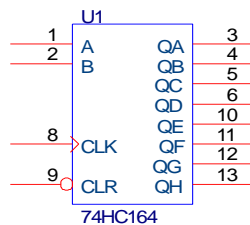
Diseñar un contador síncrono ascendente/descendente de 0 a 7 con biestables J-K.

Ejercicio 11:

Diseñar un contador síncrono con biestables D que cuente de 5 a 0 (descendente). También se requiere que si el contador está en algún estado distinto del 0 al 5, vuelva a 0 en el siguiente pulso de reloj.

Ejercicio 12:

Se dispone de dos conversores serie-paralelo de 8 bits como el de la figura. Diseñe un conversor serie-paralelo de 16 bits.

**Ejercicio 13:**

Diseñe un circuito con flip-flops JK que detecte en la entrada Y la secuencia "101" configurando la salida Z a 1.

Ejercicio 14:

Diseñe un circuito con flip-flops D que detecte en la entrada Y cuatro "1" consecutivos ajustando la salida Z a 1.

Ejercicio 15:

Diseñe un circuito con una entrada Y y una salida Z que debe tomar el valor "1" siempre que los dos últimos bits en la entrada sean iguales, y "0" en caso contrario.

PROBLEMAS DE FUNDAMENTOS DE TECNOLOGÍA DE COMPUTADORES

T5. MEMORIAS

Base teórica

La memoria es el lugar en el que se almacenan las instrucciones y los datos para que se puedan ejecutar los programas. Sin embargo, el sistema de memoria del computador está formado por varios tipos de memorias con diferentes capacidades y tiempos de acceso. La idea es que parezca que las referencias a memoria se sirven a velocidades cercanas a las de los registros del procesador, y que además se tiene un espacio de memoria casi ilimitado para los programas y los datos. La memoria caché es la responsable de la rapidez de los accesos y la memoria virtual la de la gran capacidad del sistema de memoria.

Para que todo el sistema funcione, se diseña de manera jerárquica, por el que los diferentes bloques de información se van moviendo

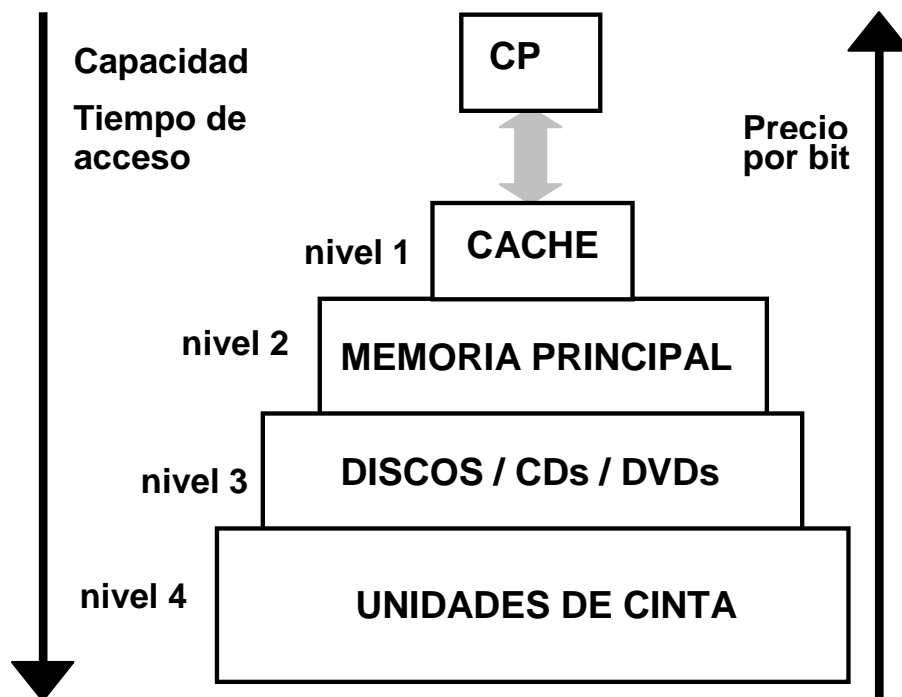


Figura 1: jerarquía de memoria

Memoria principal

Centrándose en la memoria principal, se distinguen dos grupos de memorias:

- Memorias de las que solamente se puede leer (ROM)
- Memorias en las que se puede leer y escribir (RAM)

La capacidad de la memoria principal se mide en cuantos bytes o palabras es capaz de almacenar. La manera habitual es indicar $nnK \times mm$ en el que nn son el número de direcciones que tiene la memoria con capacidad para poder almacenar mm bits en cada una.

La capacidad de memoria viene dada por el bus de direcciones que establece el máximo número de posiciones direccionables por el computador. Si se tienen n bits para el bus de direcciones, se podrá acceder hasta un máximo de 2^n posiciones.

Independientemente de cuanta memoria sea capaz de direccionar, los computadores no los entregan con el máximo de memoria disponible, por lo que habrá que definir como se accede a las posiciones ocupadas y a las que no están disponibles. Esa información se encuentra en el mapa de memoria del computador.

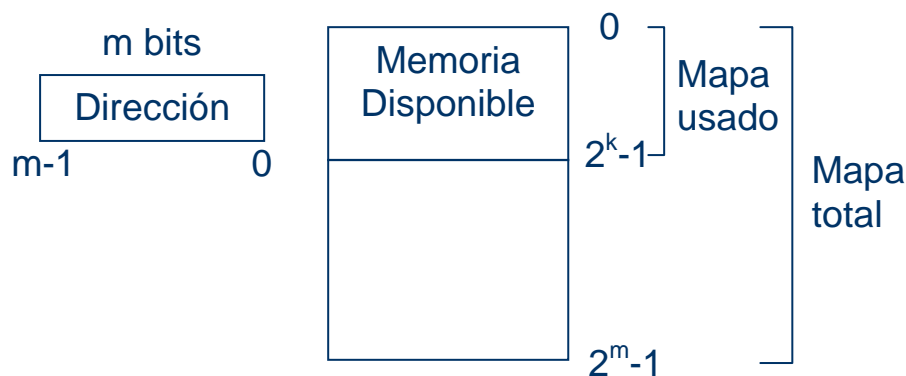


Figura 2: esquema de memoria de un computador.

Las medidas más empleadas para la capacidad de memoria son:

- Kilobyte (KB) = 2^{10} bytes
- Megabyte (MB) = 2^{20} bytes
- Gigabyte (GB) = 2^{30} bytes
- Terabyte (TB) = 2^{40} bytes

Aumento de la memoria en número de direcciones

Si se desea una memoria de 16Kx8 a partir de módulos de memoria de 8Kx8 para un procesador con un bus de direcciones de 14 bits, se tendrá que definir cuando se accede a un módulo o al otro.

0 8K-1	Módulo 0 de 8Kx8
8K 16K-1	Módulo 1 de 8Kx8

Figura 3. Esquema de memoria solicitado

Lo primero es ver si podemos direccionar 16K con 14 bits. Para ello, se sabe que $16K = 2^{14}$ con lo que si se pueden direccionar los 16K.

Para calcular cuantos módulos hacen falta, en este caso es muy sencillo y se ve a simple vista, se realiza la operación siguiente:

$$\frac{16k}{8k} \times \frac{8}{8} = 2 \times 1 = 2 \text{ módulos de } 8k \times 8$$

De la ecuación anterior se desprende que se necesitan dos módulos, tal y como reflejaba la figura 3.

Ahora se tiene que ver cuando se accede a un módulo o al otro. Ya que los dos módulos son de 8K, se necesitarán 13 bits del bus de direcciones para poder direccionar en cada módulo esos 8K ($=2^{13}$). De esa forma, nos queda únicamente un bit para determinar cuando se accede al módulo cero o al módulo uno.

La información de cómo acceder a los módulos se da en el mapa de memoria, tal y como muestra la tabla siguiente:

A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Módulo
1	1	1	1	1	1	1	1	1	1	1	1	1	1	Módulo 1
	
	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	1	1	1	1	1	1	1	1	1	1	1	1	1	Módulo 0
	
	0	0	0	0	0	0	0	0	0	0	0	0	0	

Tabla 1: mapa de memoria de 16Kx8 a partir de dos módulos de 8Kx8

El esquema de memoria quedaría:

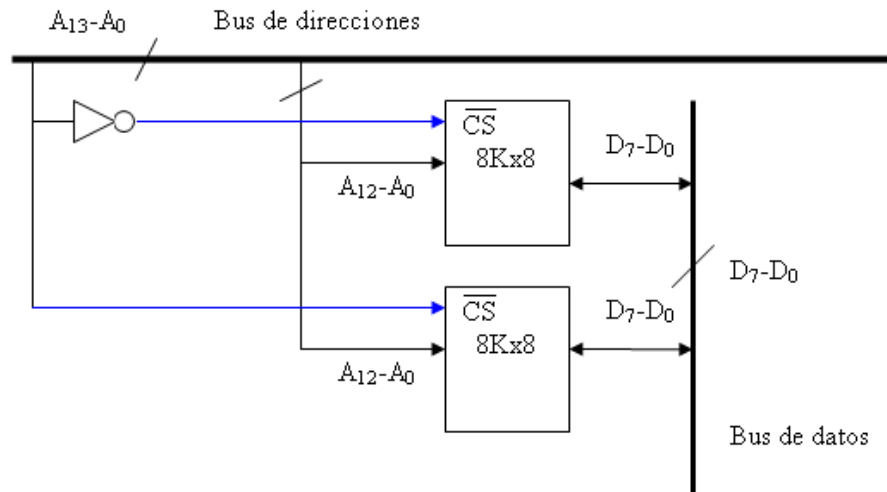


Figura 4 .Esquema de memoria de 16Kx8 a partir de dos módulos de 8Kx8

Aumento de la memoria en el ancho de palabra

Si se desea una memoria de 16Kx16 a partir de módulos de memoria de 16Kx8 para un procesador con un bus de direcciones de 14 bits, se tendrá que definir cuando se accede a un módulo o al otro y cuantos módulos harán falta.

$$\frac{16k}{16k} \times \frac{16}{8} = 1 \times 2 = 2 \text{ módulos de } 16k \times 8$$

En este caso, se deberá acceder a los dos módulos a la vez, dado que en un módulo se almacenarán los 8 bits superiores y en el otro los 8 bits de menor peso. El esquema quedaría de la forma:

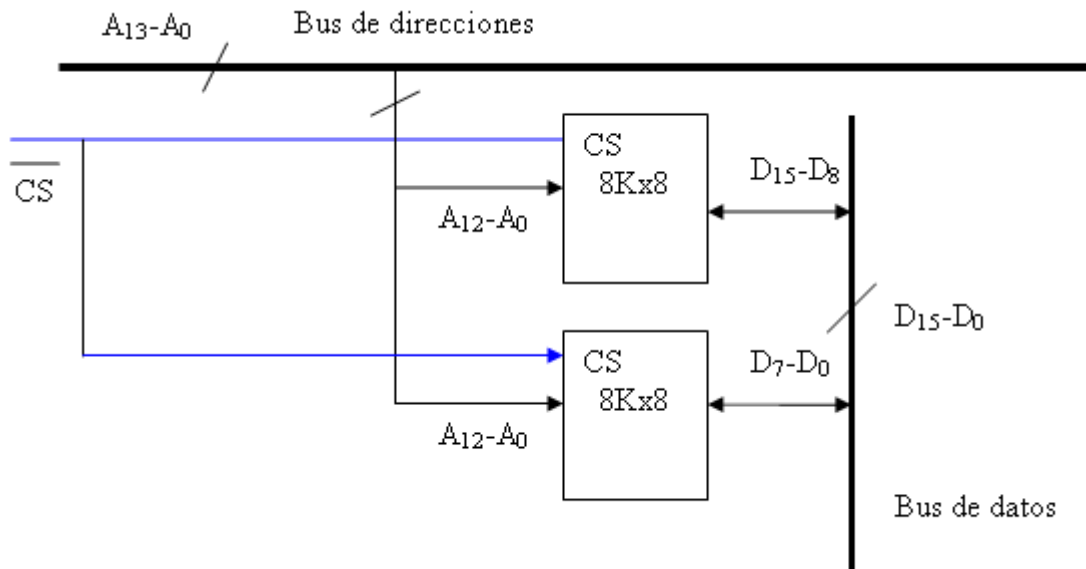


Figura 4 .Esquema de memoria de 16Kx16 con de dos módulos de 8Kx8

También es posible aumentar el tamaño de palabra y el número de posiciones de la memoria combinando los dos esquemas anteriores.

EJERCICIOS T5

1. Una CPU cuenta con un ancho de palabra de 16 bits. Se quiere dotar a esa CPU de una memoria con las siguientes características:

- 256 Kpalabras (256 K x 16) de memoria ROM.
- 512 Kpalabras (512 K x 16) de memoria RAM.

Diseñar la memoria con el menor número de pastillas, sabiendo que disponemos de las siguientes y que deseamos que la RAM ocupe las posiciones más bajas del mapa de memoria, seguida de la ROM:

Pastillas de memoria ROM	Pastillas de memoria RAM
64 K x 8	128 K x 1
128 K x 1	256 K x 8
128 K x 8	

2. La CPU de la figura cuenta con un bus de datos de 16 bits y un bus de direcciones de 20 bits.

Se quiere diseñar un computador con una memoria principal de las siguientes características:

- 256 Kpalabras (256 K x 16) de memoria ROM.
- 512 Kpalabras (512 K x 16) de memoria RAM.

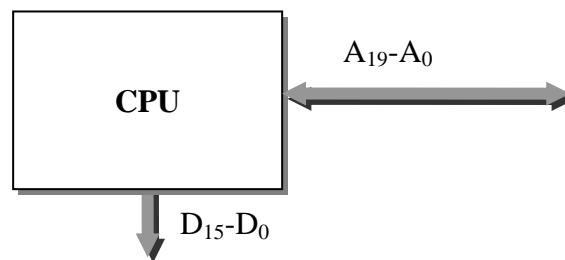
La memoria ROM debe situarse en las posiciones más altas del mapa de memoria direccionable y la memoria RAM debe situarse en las posiciones más bajas.

Diseñar la memoria con el menor número de pastillas sabiendo que disponemos de las siguientes:

Pastillas de memoria ROM	Pastillas de memoria RAM
64 K x 8	128 K x 1
128 K x 1	256 K x 8
128 K x 16	256 K x 16

3. La CPU de la figura se conecta al bus de datos con 16 bits, además, el bus de direcciones cuenta con 20 bits. Se quiere dotar a esa CPU de una memoria, sin multiplexación de buses, con las siguientes características:

- 128 Kpalabras (128K x 16) de memoria ROM.
- 640 Kpalabras (640K x 16) de memoria RAM.



La memoria ROM debe situarse en las posiciones más altas del mapa de memoria direccionable y la memoria RAM debe situarse en las posiciones más bajas.

Diseñar la memoria con el menor número de pastillas sabiendo que disponemos de:

Pastillas de memoria ROM	Pastillas de memoria RAM
64 k x 8	128 k x 1
128 k x 1	256 k x 8
128 k x 16	256 k x 16

4. Una CPU se conecta al bus de datos con 16 bits, además, su bus de direcciones cuenta con 20 bits. Se quiere dotar a esa CPU de una memoria con las siguientes características:

- 640 Kpalabras x 16 de memoria ROM.
- 384 Kpalabras x 16 de memoria RAM.

La memoria ROM debe situarse en las posiciones más altas del mapa de memoria direccionable y la memoria RAM debe situarse en las posiciones más bajas.

Diseñar la memoria con el menor número de pastillas sabiendo que disponemos de:

Pastillas de memoria RAM	Pastillas de memoria ROM
64 k x 1	32 k x 1
128 k x 8	64 k x 1
512 k x 8	128 k x 8

5. Se quiere diseñar una memoria para un circuito electrónico que se adaptará a una tarjeta de felicitación de cumpleaños de forma que al abrirla suene la melodía del cumpleaños feliz. El archivo de sonido es de tipo WAV y ocupa 384 K y el programa que reproducirá la melodía ocupará, como máximo, 128 K.

Además se tienen que tener en cuenta los siguientes puntos:

- El cliente puede pedir que el fichero de sonido sea en formato MP3 192 K con lo que el programa ocupará un máximo de 320 K. Dado que no existirán diferentes tarjetas de felicitación se deberá cambiar el contenido de la memoria.
- Se podrá cambiar las baterías sin que se pierda la información de las memorias.

Se sabe que el bus de datos es de 16 bits y el de direcciones de 20 bits y que se cuenta con los siguientes módulos de memoria disponibles para realizar el circuito:

Memoria ROM	Memoria EEPROM	Memoria RAM
128 K x 8	128 K x 8	128 K x 8
256 K x 8	256 K x 8	256 K x 8
512 K x 8	512 K x 8	512 K x 16

- Razonar brevemente el tipo de módulos de memoria que se deberán emplear para realizar el circuito con las especificaciones requeridas.
- Calcular el menor número de módulos necesario para realizar las especificaciones anteriores.
- Realizar el mapa de memoria con una capacidad máxima para almacenar el fichero de sonido (WAV o MP3) y el programa correspondiente.



- Se desea situar el bloque de memoria de 8Kx8 mostrado en la figura 1, al principio del mapa de memoria del procesador, es decir, ocupando desde la posición 0000h a la 1FFFh. Diseñar la lógica auxiliar que controle la señal CS mediante las líneas de mayor peso del bus de direcciones no usadas en el direccionamiento de la posición interna.

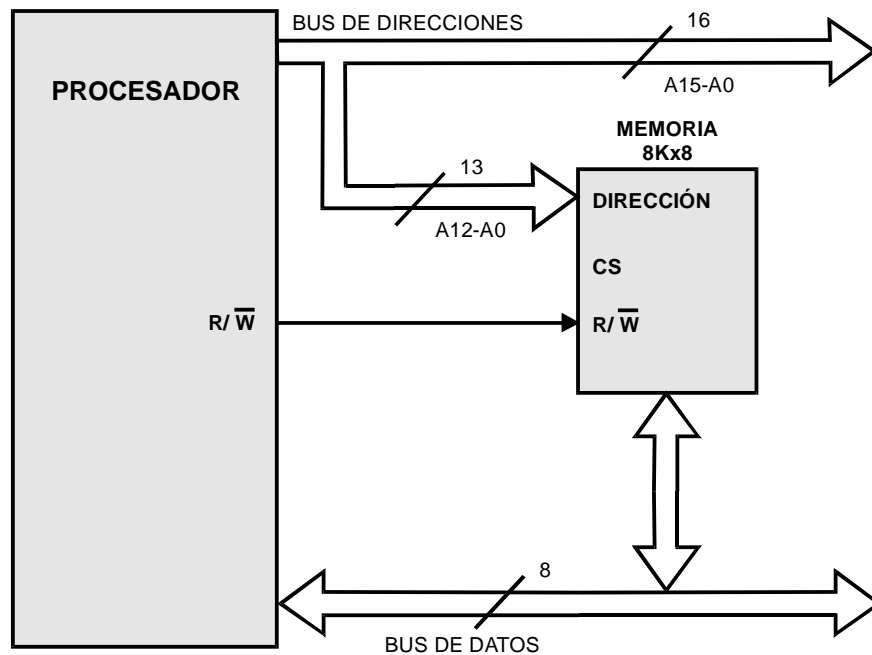


Figura 1. Sistema ejercicio 1

7. Se dispone de dos módulos de memoria M1 y M2 de 256x8 de capacidad, que se desean ubicar en el mapa de memoria correspondiente a un procesador que tiene un bus de direcciones de 10 líneas y un bus de datos de ocho. M1 hay que situarlo al principio del mapa, o sea, en las 256 primeras posiciones, mientras que M2 hay que situarlo al final del mapa.

Diseñar el esquema de conexionado correspondiente.

8. Se dispone de un subsistema de memoria como el mostrado en la figura 2. Averiguar el rango de direcciones a los que corresponden los módulos M1 y M2.

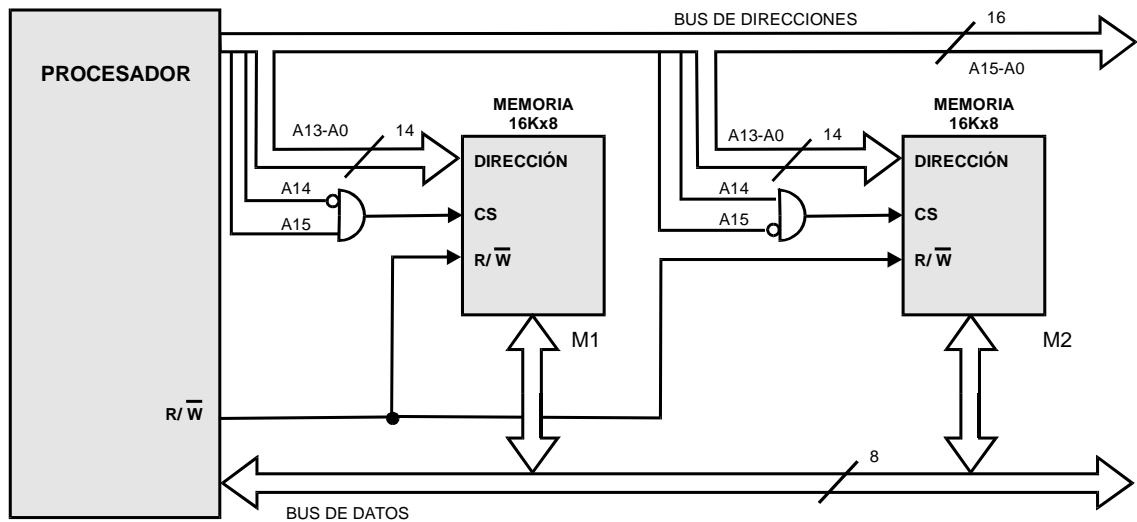
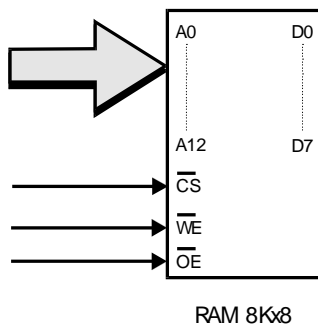


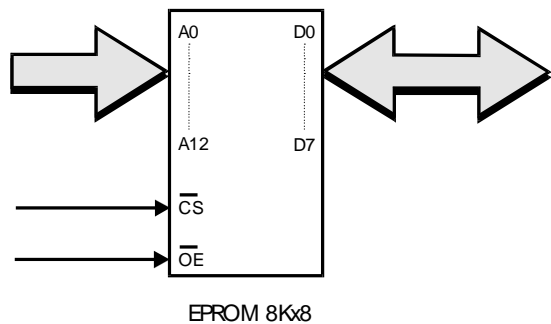
Figura 2. Sistema ejercicio 10

9. Se desea dotar a un sistema microprocesador basado en una CPU de 16 bits de datos y 16 bits de direcciones, de un banco de memoria RAM de 16 kpalabras y un banco de memoria EPROM de 8 kpalabras. Para ello se dispone de circuitos integrados como los mostrados en las figuras 3a y 3b.



RAM 8Kx8

Figura 3a. Memoria RAM



EPROM 8Kx8

Figura 3b. Memoria EPROM

La CPU tiene una línea de lectura y una de escritura, ambas activas a nivel bajo (\overline{RD} y \overline{WR} respectivamente). Los rangos de direcciones que se desean para ambos bancos son los siguientes:

	Dirección inicio	Dirección fin
RAM	2000h	5FFFh
EPROM	8000h	9FFFh

- Diseñar el sistema de memoria que se pide empleando puertas lógicas para el circuito decodificador de direcciones.
- Diseñar el mismo sistema que en el apartado anterior pero recurriendo al decodificador 3 a 8 74LS138 (figura 4).

Inputs					Outputs							
Enable		Select										
G1	G2 (Note 1)	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	L	H	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

Nota 1: $G2 = G2A + G2B$

Figura 4. Decodificador 3 a 8 74LS138

**SOLUCIÓN PROBLEMAS DE
FUNDAMENTOS DE TECNOLOGÍA DE
COMPUTADORES**

T5. MEMORIAS

Solución ejercicio 1

1. Comprobar que existe solución

Para poder comprobarlo, debemos fijarnos en el número de bits que tenemos en el bus de direcciones, y ver que con ese número de bits, podemos direccionar todo el mapa de memoria.

El bus de direcciones tiene 20 líneas, desde la A_{19} hasta la A_0 , lo que supone una capacidad de direccionamiento de 2^{20} palabras = 1 Mpalabras. En el caso que nos ocupa, necesitamos direccionar 512 Kpalabras de memoria RAM y 256 Kpalabras de memoria ROM, en total 768 Kpalabras. Por lo tanto, vemos que con los 20 bits del bus de direcciones sí se puede direccionar la memoria que se nos pide.

Por otro lado, tenemos que comprobar que el bus de datos soporte el ancho de una palabra de memoria (16 bits). Por tanto, vamos a suponer que el bus de datos cuenta con 16 líneas, desde la D_{15} hasta la D_0 .

2. Calcular los módulos de memoria que serán necesarios

Para la memoria RAM, disponemos de los siguientes módulos de memoria:

- 128 K x 1

$$\frac{512 \text{ K}}{128 \text{ K}} \times \frac{16 \text{ bits}}{1 \text{ bit}} = 4 \times 16 = 64 \text{ pastillas de } 128 \text{ K x } 1$$

Con esta posibilidad se necesitarían 4 filas de módulos de 128 K posiciones de memoria (para poder direccionar las 512 K posiciones de memoria) y 16 módulos por cada una de las filas para almacenar los 16 bits que componen una palabra.

Con lo cual, el mapa de memoria RAM resultante contaría con $4 \times 16 = 64$ pastillas de 128 K x 1.

- 256 K x 8

$$\frac{512 \text{ K}}{256 \text{ K}} \times \frac{16 \text{ bits}}{8 \text{ bits}} = 2 \times 2 = 4 \text{ pastillas de } 256 \text{ K x } 8$$

Con esta posibilidad se necesitarían 2 filas de módulos de 256 K posiciones de memoria (para poder direccionar las 512 K posiciones de memoria) y 2 módulos por cada una de las filas para almacenar los 16 bits que componen una palabra (en cada uno de los módulos se almacenarán los 8 bits que componen las partes alta y baja de una palabra).

Con lo cual, el mapa de memoria RAM resultante contaría con $2 \times 2 = 4$ pastillas de 256 K x 8.

Para la memoria ROM, disponemos de los siguientes módulos de memoria:

- 128 K x 1

$$\frac{256 \text{ K}}{128 \text{ K}} \times \frac{16 \text{ bits}}{1 \text{ bit}} = 2 \times 16 = 32 \text{ pastillas de } 128 \text{ K x } 1$$

- 64 K x 8

$$\frac{256 \text{ K}}{64 \text{ K}} \times \frac{16 \text{ bits}}{8 \text{ bit}} = 4 \times 2 = 8 \text{ pastillas de } 64 \text{ K x } 8$$

- 128 K x 8

$$\frac{256 \text{ K}}{128 \text{ K}} \times \frac{16 \text{ bits}}{8 \text{ bits}} = 2 \times 2 = 4 \text{ pastillas de } 128 \text{ K x } 8$$

La solución más sencilla es utilizar:

- $2 \times 2 = 4$ pastillas de 256 K x 8 para la memoria RAM
- $2 \times 2 = 4$ pastillas de 128 K x 8 para la memoria ROM

3. Diseñar el mapa de memoria

Para diseñar el mapa de memoria, supondremos que la memoria RAM ocupa las posiciones más bajas del mapa de memoria y que la memoria ROM se encuentra a continuación.

En la siguiente figura se muestra un esquema del mapa de memoria resultante:

FFFFF h	Libre	1024 K – 1
C0000 h		768 K
BFFFF h	ROM	768 K – 1
80000 h		512 K
7FFFF h	RAM	512 K – 1
00000 h		0

Finalmente, la distribución del mapa de memoria resultante es la que aparece en la siguiente figura:

	A19	A18	A17	A16	...	A0		
FFFFF h C0000 h	1	1	1	1	...	1	1024 K – 1 768 K	Libre
BFFFF h A0000 h	1	0	1	1	...	1	768 K – 1 640 K	2ª fila pastillas ROM
9FFFF h 80000 h	1	0	0	1	...	1	640 K – 1 512 K	1ª fila pastillas ROM
7FFFF h 40000 h	0	1	1	1	...	1	512 K – 1 256 K	2ª fila pastillas RAM
3FFFF h 00000 h	0	0	1	1	...	1	256 K – 1 0	1ª fila pastillas RAM

Para poder direccionar las 256 Kpalabras ($= 2^{18}$ Kpalabras) de cada una de las pastillas de memoria RAM necesitaremos 18 bits, que se corresponden con las líneas $A_{17}..A_0$ del bus de direcciones. Mientras que para poder acceder a las 128 Kpalabras ($= 2^{17}$ Kpalabras) de cada una de las pastillas de memoria ROM necesitaremos 17 bits, son las líneas $A_{16}..A_0$ del bus de direcciones.

Las líneas A_{19} y A_{18} del bus de direcciones, nos permitirán seleccionar la fila de pastillas sobre la que se realizará la lectura o escritura de los datos.

Por otro lado, la primera pastilla de cada fila del mapa de memoria almacenará la parte alta de una palabra y la segunda la parte alta.

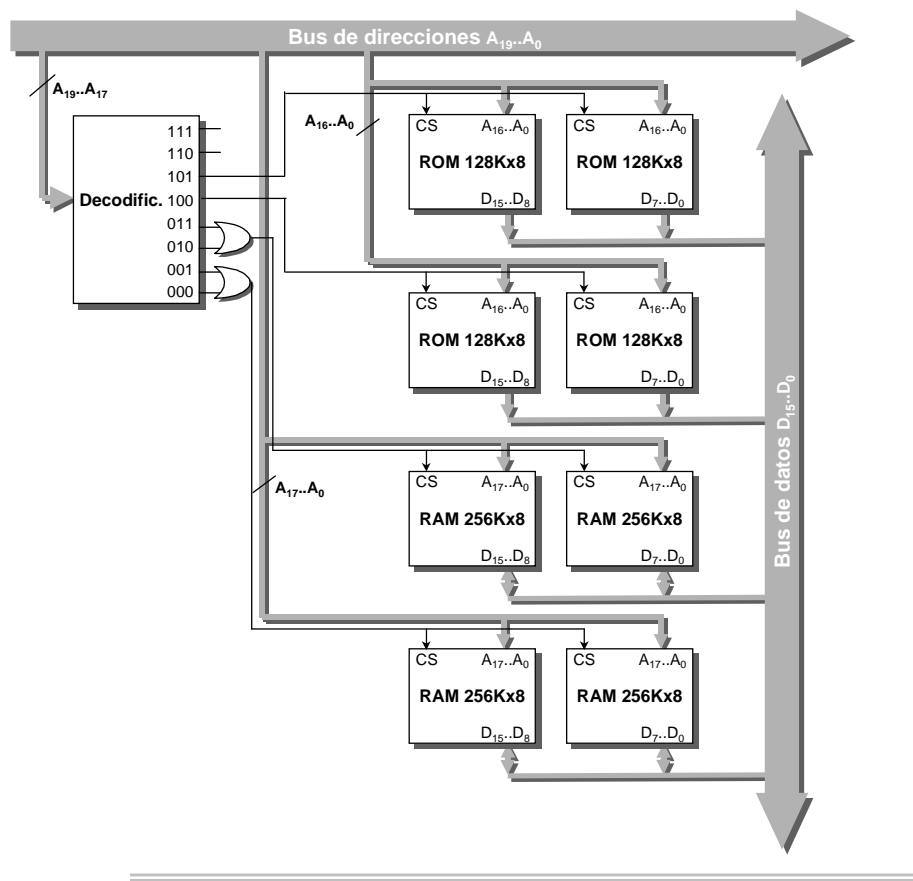
4. Dibujar el esquema del mapa de memoria

Utilizando un decodificador de 3 entradas, líneas $A_{19} .. A_{17}$.

Debemos considerar que en el esquema falta una línea de lectura conectada a todas las pastillas de memoria RAM y ROM y una línea de escritura conectada a todas las pastillas de memoria RAM.

Se utiliza un decodificador de tres entradas ($A_{19}..A_{17}$) y dos puertas OR para seleccionar las filas de pastillas de memoria RAM.

Las filas de memoria ROM se seleccionan con las salidas del decodificador 101 y 100.



Solución ejercicio 2

1. Comprobar que existe solución

El bus de direcciones tiene 20 líneas, desde la A_{19} hasta la A_0 , lo que supone una capacidad de direccionamiento de 2^{20} palabras = 1 Mpalabras. En el caso que nos ocupa, necesitamos direccionar 512 Kpalabras de memoria RAM y 256 Kpalabras de memoria ROM, en total 768 Kpalabras. Por lo tanto, vemos que con los 20 bits del bus de direcciones sí es posible.

Por otro lado, una dato de memoria puede transmitirse por el bus de dato puesto que cuenta con 16 líneas, desde la D_{15} hasta la D_0 .

2. Calcular los módulos de memoria que serán necesarios

Para la memoria RAM, disponemos de los siguientes módulos de memoria:

128 K x 1

$$\frac{512 \text{ K}}{128 \text{ K}} \times \frac{16 \text{ bits}}{1 \text{ bit}} = 4 \times 16 = 64 \text{ pastillas de } 128 \text{ K x } 1$$

256 K x 8

$$\frac{512 \text{ K}}{256 \text{ K}} \times \frac{16 \text{ bits}}{8 \text{ bits}} = 2 \times 2 = 4 \text{ pastillas de } 256 \text{ K x } 8$$

256 K x 16

$$\frac{512 \text{ K}}{256 \text{ K}} \times \frac{16 \text{ bits}}{16 \text{ bits}} = 2 \times 1 = 2 \text{ pastillas de } 256 \text{ K x } 16$$

Para la memoria ROM, disponemos de los siguientes módulos de memoria:

128 K x 1

$$\frac{256 \text{ K}}{128 \text{ K}} \times \frac{16 \text{ bits}}{1 \text{ bit}} = 2 \times 16 = 32 \text{ pastillas de } 128 \text{ K x } 1$$

64 K x 8

$$\frac{256 \text{ K}}{64 \text{ K}} \times \frac{16 \text{ bits}}{8 \text{ bits}} = 4 \times 2 = 8 \text{ pastillas de } 64 \text{ K x } 8$$

128 K x 16

$$\frac{256 \text{ K}}{128 \text{ K}} \times \frac{16 \text{ bits}}{16 \text{ bits}} = 2 \times 1 = 2 \text{ pastillas de } 128 \text{ K x } 16$$

- 2 x 1 = 2 pastillas de 256 K x 16 para la memoria RAM
- 2 x 1 = 2 pastillas de 128 K x 16 para la memoria ROM

3. Diseñar el mapa de memoria

En la siguiente figura se muestra un esquema del mapa de memoria resultante:

FFFF h	ROM	1024 K – 1
C0000 h		768 K
BFFFF h	Libre	768 K – 1
80000 h		512 K
7FFFF h	RAM	512 K – 1
00000 h		0

La distribución del mapa de memoria resultante es la que aparece a continuación:

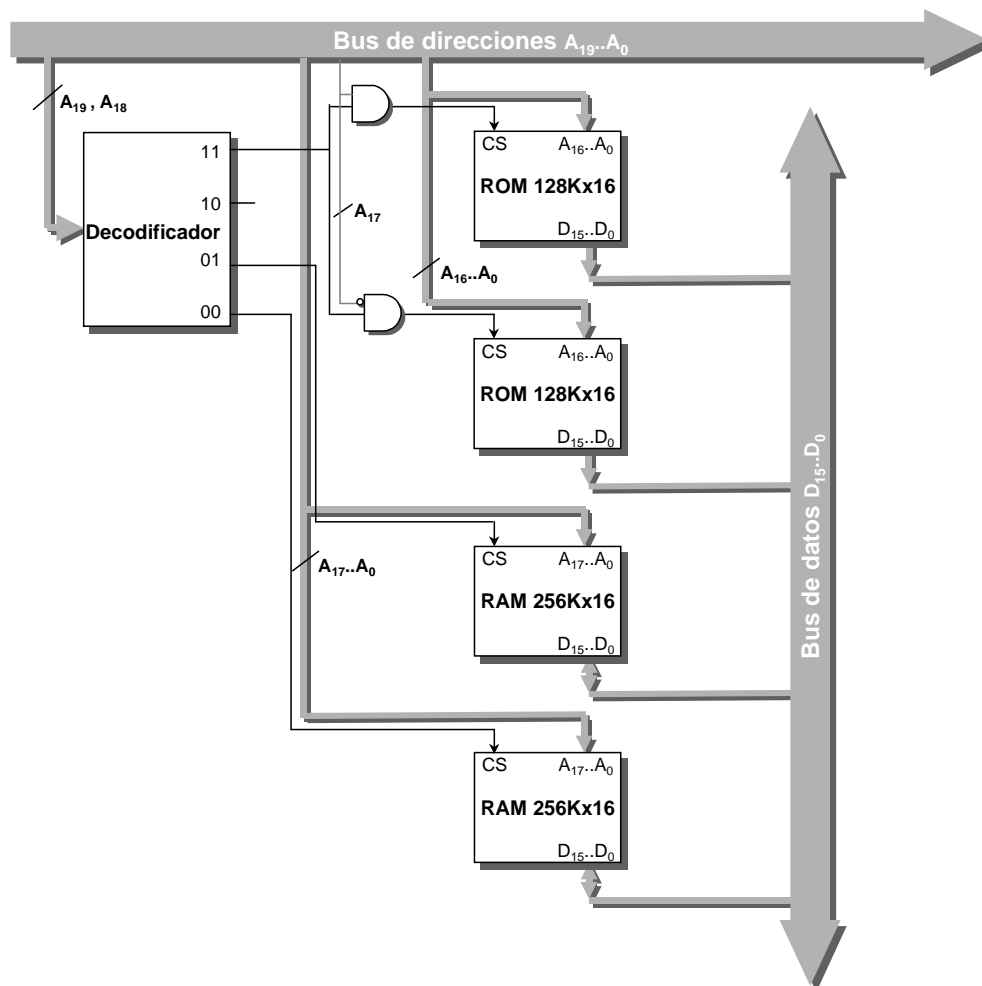
	A19	A18	A17	A16	...	A0		
FFFFF h	1	1	1	1	...	1	1024 K –1	2ª fila
E0000 h	1	1	1	0	...	0	896 K	pastillas ROM
DFFFF h	1	1	0	1	...	1	896 K –1	1ª fila
C0000 h	1	1	0	0	...	0	768 K	pastillas ROM
BFFFF h	1	0	1	1	...	1	768 K – 1	Libre
80000 h	1	0	0	0	...	0	512 K	
7FFFF h	0	1	1	1	...	1	512 K – 1	2ª fila
40000 h	0	1	0	0	...	0	256 K	pastillas RAM
3FFFF h	0	0	1	1	...	1	256 K – 1	1ª fila
00000 h	0	0	0	0	...	0	0	pastillas RAM

Para poder direccionar las 256 Kpalabras ($= 2^{18}$ Kpalabras) de cada una de las pastillas de memoria RAM necesitaremos 18 bits, que se corresponden con las líneas $A_{17}..A_0$ del bus de direcciones. Mientras que para poder acceder a las 128 Kpalabras ($= 2^{17}$ Kpalabras) de cada una de las pastillas de memoria ROM necesitaremos 17 bits, son las líneas $A_{16}..A_0$ del bus de direcciones.

Las líneas A_{19} y A_{18} del bus de direcciones, nos permitirán seleccionar la fila de pastillas sobre la que se realizará la lectura o escritura de los datos.

4. Dibujar el esquema del mapa de memoria

Utilizando un decodificador de 2 entradas, líneas $A_{19} .. A_{18}$ debemos considerar que faltaría añadir una línea de lectura conectada a todas las pastillas de memoria RAM y ROM y una línea de escritura conectada a todas las pastillas de memoria RAM.



Solución ejercicio 3

1. Comprobar que existe solución

El bus de direcciones tiene 20 líneas, desde la A_{19} hasta la A_0 , lo que supone una capacidad de direccionamiento de 2^{20} palabras = 1 Mpalabras. En el caso que nos ocupa, necesitamos direccionar 640 Kpalabras de memoria RAM y 128 Kpalabras de memoria ROM, en total 768 Kpalabras.

Por lo tanto, vemos que con los 20 bits del bus de direcciones sí es posible.

2. Calcular los módulos de memoria que serán necesarios

Para la memoria RAM, disponemos de los siguientes módulos de memoria:

128 K x 1

$$\frac{640 \text{ K}}{128 \text{ K}} \times \frac{16 \text{ bits}}{1 \text{ bit}} = 5 \times 16 = 80 \text{ pastillas de } 128 \text{ K x } 1$$

256 K x 8

$$\frac{640 \text{ K}}{256 \text{ K}} \times \frac{16 \text{ bits}}{8 \text{ bits}} = 3 \times 2 = 6 \text{ pastillas de } 256 \text{ K x } 8$$

256 K x 16

$$\frac{640 \text{ K}}{256 \text{ K}} \times \frac{16 \text{ bits}}{16 \text{ bits}} = 3 \times 1 = 3 \text{ pastillas de } 256 \text{ K x } 16$$

Para la memoria ROM, disponemos de los siguientes módulos de memoria:

128 K x 1

$$\frac{128 \text{ K}}{128 \text{ K}} \times \frac{16 \text{ bits}}{1 \text{ bit}} = 1 \times 16 = 16 \text{ pastillas de } 128 \text{ K x } 1$$

64 K x 8

$$\frac{128 \text{ K}}{64 \text{ K}} \times \frac{16 \text{ bits}}{8 \text{ bits}} = 2 \times 2 = 4 \text{ pastillas de } 64 \text{ K x } 8$$

128 K x 16

$$\frac{128 \text{ K}}{128 \text{ K}} \times \frac{16 \text{ bits}}{16 \text{ bits}} = 1 \times 1 = 1 \text{ pastilla de } 128 \text{ K x } 16$$

- 3 x 1 = 3 pastillas de 256 K x 16 para la memoria RAM
- 1 x 1 = 1 pastilla de 128 K x 16 para la memoria ROM

3. Diseñar el mapa de memoria

FFFFFF h	ROM	1024 K – 1
E0000 h		896 K
DFFFF h	Libre	896 K – 1
A0000 h		640 K
9FFFF h	RAM	640 K – 1
00000 h		0

	A19	A18	A17	A16	...	A0		
FFFFF h	1	1	1	1	...	1	1024 K – 1	Fila
E0000 h				0	...	0	896 K	pastillas
DFFFF h	1	1	0	1	...	1	896 K – 1	ROM
C0000 h				0	...	0	768 K	Libre
BFFFF h	Zona de RAM no direccionable						768 K – 1	
A0000 h							640 K	
9FFFF h	1	0	0	1	...	1	640 K – 1	3ª fila
80000 h			0	0	...	0	512 K	pastillas
7FFFF h	0	1	1	1	...	1	512 K – 1	2ª fila
40000 h			0	0	...	0	256 K	pastillas
3FFFF h	0	0	1	1	...	1	256 K – 1	1ª fila
00000 h			0	0	...	0	0	pastillas
								RAM

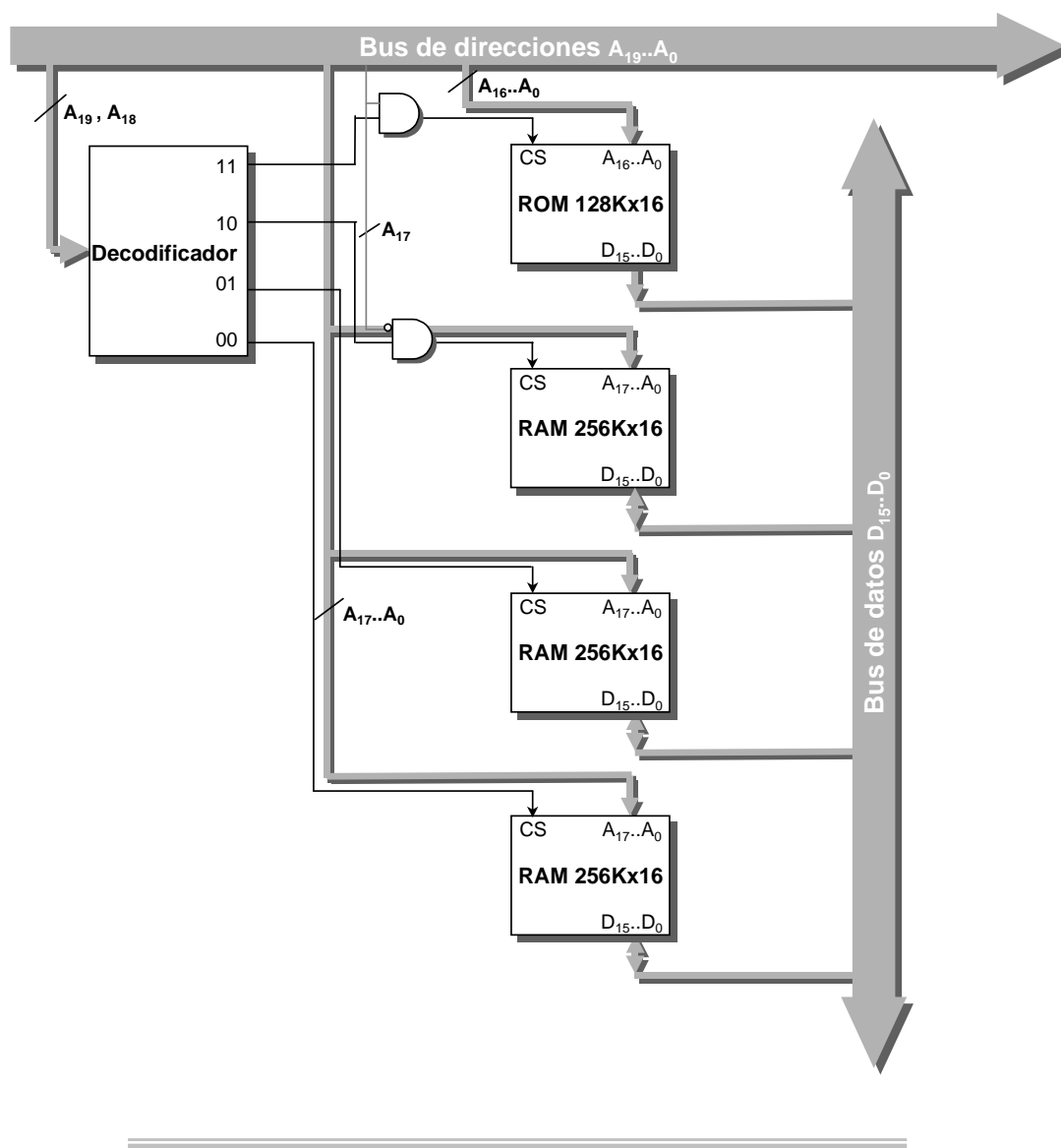
Como puede observarse en el mapa de memoria, la tercera fila de pastillas de memoria RAM no se utiliza de manera completa.

Para poder direccionar las 256 Kpalabras ($= 2^{18}$ Kpalabras) de cada una de las pastillas de memoria RAM necesitaremos 18 bits, que se corresponden con las líneas $A_{17}..A_0$ del bus de direcciones. Mientras que para poder acceder a las 128 Kpalabras ($= 2^{17}$ Kpalabras) de cada una de las pastillas de memoria ROM necesitaremos 17 bits, son las líneas $A_{16}..A_0$ del bus de direcciones.

Las líneas A_{19} y A_{18} del bus de direcciones, nos permitirán seleccionar la fila de pastillas sobre la que se realizará la lectura o escritura de los datos.

4. Dibujar el esquema del mapa de memoria

En la siguiente figura se muestra el esquema correspondiente a utilizar un decodificador de 2 entradas, líneas $A_{19} .. A_{18}..$. Debemos considerar que faltaría añadir una línea de lectura conectada a todas las pastillas de memoria RAM y ROM y una línea de escritura conectada a todas las pastillas de memoria RAM.



Solución ejercicio 4**1. Comprobar que existe solución**

El bus de direcciones tiene 20 líneas, desde la A_{19} hasta la A_0 , lo que supone una capacidad de direccionamiento de 2^{20} palabras = 1 Mpalabras. En el caso que nos ocupa, necesitamos direccionar 640 Kpalabras de memoria ROM y 384 Kpalabras de memoria RAM, en total 1024 Kpalabras. Por lo tanto, vemos que con los 20 bits del bus de direcciones sí se puede direccionar la memoria que se nos pide.

2. Calcular los módulos de memoria que serán necesarios

Para la memoria RAM, disponemos de los siguientes módulos de memoria:

64 K x 1

$$\frac{384 \text{ K}}{64 \text{ K}} \times \frac{16 \text{ bits}}{1 \text{ bit}} = 6 \times 16 = 96 \text{ pastillas de } 64 \text{ K x } 1$$

512 K x 8

$$\frac{384 \text{ K}}{512 \text{ K}} \times \frac{16 \text{ bits}}{8 \text{ bits}} = 1 \times 2 = 2 \text{ pastillas de } 512 \text{ K x } 8$$

128 K x 8

$$\frac{384 \text{ K}}{128 \text{ K}} \times \frac{16 \text{ bits}}{8 \text{ bits}} = 3 \times 2 = 6 \text{ pastillas de } 128 \text{ K x } 8$$

Para la memoria ROM, disponemos de los siguientes módulos de memoria:

32 K x 1

$$\frac{640 \text{ K}}{32 \text{ K}} \times \frac{16 \text{ bits}}{1 \text{ bit}} = 20 \times 16 = 320 \text{ pastillas de } 32 \text{ K x } 1$$

64 K x 1

$$\frac{640 \text{ K}}{64 \text{ K}} \times \frac{16 \text{ bits}}{1 \text{ bit}} = 10 \times 16 = 160 \text{ pastillas de } 64 \text{ K x } 1$$

128 K x 8

$$\frac{640 \text{ K}}{128 \text{ K}} \times \frac{16 \text{ bits}}{8 \text{ bits}} = 5 \times 2 = 10 \text{ pastillas de } 128 \text{ K x } 8$$

La solución más sencilla es utilizar:

- 1 x 2 = 2 pastillas de 512 K x 8 para la memoria RAM
- 5 x 2 = 10 pastillas de 128 K x 8 para la memoria ROM

3. Diseñar el mapa de memoria

En las siguientes figuras se muestra un esquema del mapa de memoria resultante:

FFFFF h	ROM	1024 K – 1
60000 h		384 K
5FFFF h	RAM	384 K – 1
00000 h		0

	A19	A18	A17	A16	...	A0		
FFFFF h	1	1	1	1	...	1	1024 K – 1	5ª fila
E0000 h	1	1	1	0	...	0	896 K	pastillas ROM
DFFFF h	1	1	0	1	...	1	896 K – 1	4ª fila
C0000 h	1	1	0	0	...	0	768 K	pastillas ROM
BFFFF h	1	0	1	1	...	1	768 K – 1	3ª fila
A0000 h	1	0	1	0	...	0	640 K	pastillas ROM
9FFFF h	1	0	0	1	...	1	640 K – 1	2ª fila
80000 h	1	0	0	0	...	0	512 K	pastillas ROM
7FFFF h	0	1	1	1	...	1	512 K – 1	1ª fila
60000 h	0	1	1	0	...	0	384 K	pastillas ROM
	Zona de RAM no direccionable							
5FFFF h	0	1	0	1	...	1	384 K – 1	Fila pastillas RAM
00000 h	0	0	X	0	...	0	0	

Como puede observarse en el mapa de memoria anterior, existe una zona de las pastillas de memoria RAM que no va a poder utilizarse para almacenar datos.

Para poder direccionar las 512 Kpalabras ($= 2^{19}$ Kpalabras) de cada una de las pastillas de memoria RAM necesitaremos 19 bits, que se corresponden con las líneas $A_{18}..A_0$ del bus de direcciones. Mientras que para poder acceder a las 128 Kpalabras ($= 2^{17}$ Kpalabras) de cada una de las pastillas

de memoria ROM necesitaremos 17 bits, son las líneas $A_{16}..A_0$ del bus de direcciones.

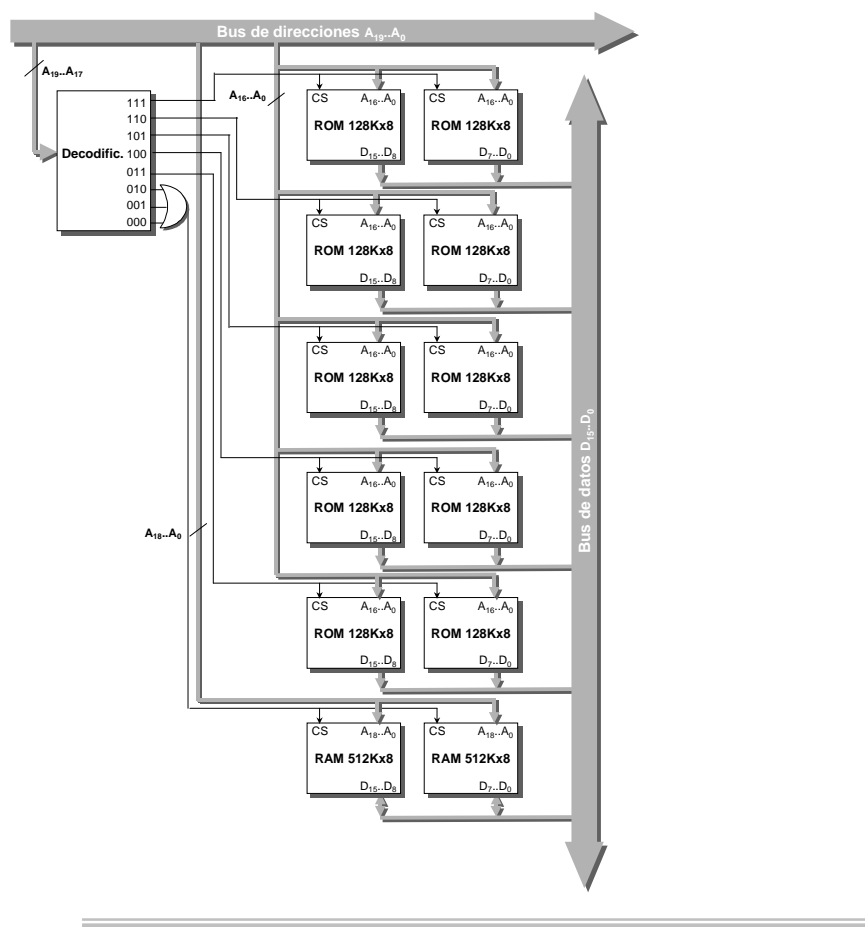
Las líneas A_{19} y A_{18} del bus de direcciones, nos permitirán seleccionar la fila de pastillas sobre la que se realizará la lectura o escritura de los datos.

Por otro lado, la primera pastilla de cada fila del mapa de memoria almacenará la parte alta de una palabra y la segunda la parte alta.

4. Dibujar el esquema del mapa de memoria

El esquema de éste mapa de memoria admite diferentes soluciones, la solución elegida utiliza un decodificador de 3 entradas, líneas $A_{19} .. A_{17}$, tal y como se muestra en la siguiente figura.

Debemos considerar que en ambos esquemas falta una línea de lectura conectada a todas las pastillas de memoria RAM y ROM y una línea de escritura conectada a todas las pastillas de memoria RAM



Solución ejercicio 6

Apartado a)

Dado que el contenido de la memoria se podrá cambiar a petición del cliente y que el contenido de la memoria debe permanecer aunque se cambie la pila, se necesitará una memoria permanente de tipo EEPROM.

Apartado b)

En los dos casos, las especificaciones de memoria, nos dan una necesidad de 512K.

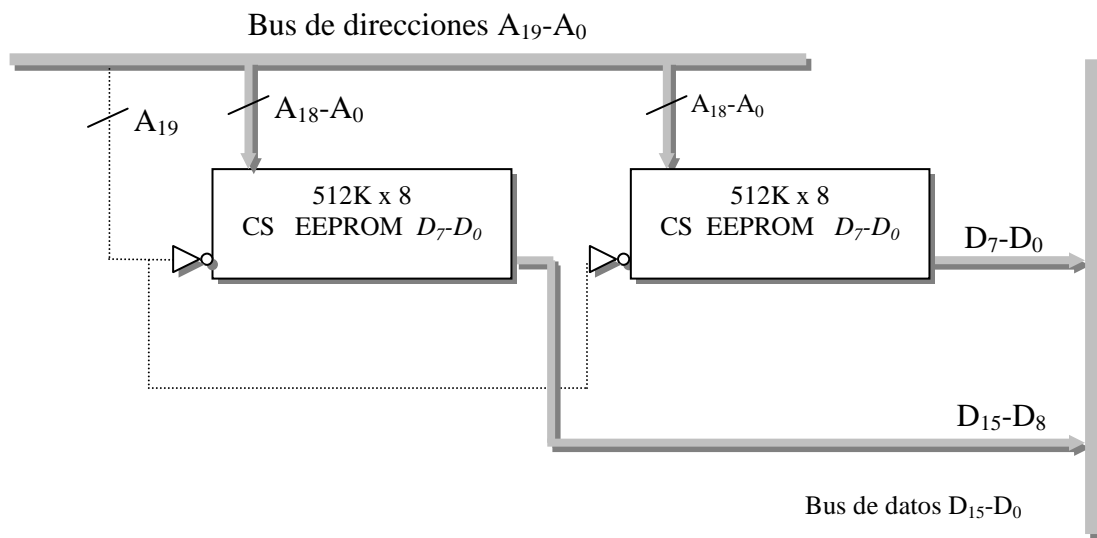
Como el bus de direcciones es de 20 bits que permite direccionar un máximo de $1\text{M} \times 16$. podremos direccionar los 512K que nos piden (solamente 19 bits A18-A0)

El menor número de módulos de memoria, dado que tenemos que escoger la opción de módulos EEPROM se da con dos módulos de 512K x 8 (tipo EEPROM)

Apartado c)

El mapa de memoria es el que se requiere para poder direccionar una pastilla de 512Kx8 de EEPROM. Es decir, tendré una fila formada por dos pastillas de memoria que se direccionarán a la vez, una contendrá los 8 bits superiores y otra los 8 bits inferiores de cada dirección.

	A19	A18	...	A0		
FFFFF h	1	0	...	1	1024 K – 1	Libre
80000 h		1		0	512 K	
7FFFF h	0	0	...	1	512 K – 1	Fila de pastillas EEPROM
00000 h		1		0	0 K	



Tema 3: Sistemas Combinacionales

Solución de algunos ejercicios

$$1. \quad S(b, c, d) = \bar{b}\bar{c}\bar{d} + \bar{b}\bar{c}d + \bar{b}c\bar{d} + b\bar{c}\bar{d} + bc\bar{d} = (b + \bar{c} + \bar{d}) \cdot (\bar{b} + c + \bar{d})$$

$$2. \quad a) \quad F(c, b, a) = \bar{c}ba + \bar{c}b\bar{a} + c\bar{b}\bar{a}$$

$$4. \quad a) \quad F(d, c, b, a) = cb\bar{a} + \bar{c}\bar{b} + \bar{b}a + \bar{d}\bar{b}$$

$$6. \quad b) \quad F(d, c, b, a) = db\bar{a} + c\bar{b}a + \bar{c}\bar{a} + \bar{d}\bar{b}$$

$$7. \quad F(a, b, c) = \bar{b}\bar{c} + bc + a\bar{b} = \bar{b}\bar{c} + bc + ac$$

$$F_a(A, B, C) = \sum(0, 2, 3, 5, 7) = \prod(1, 4, 6)$$

$$F_b(A, B, C) = \sum(0, 1, 2, 3, 4, 7) = \prod(5, 6)$$

$$F_c(A, B, C) = \sum(0, 1, 3, 4, 5, 6, 7) = \prod(2)$$

$$F_d(A, B, C) = \sum(0, 2, 3, 5, 6) = \prod(1, 4, 7)$$

$$F_e(A, B, C) = \sum(0, 2, 6) = \prod(1, 3, 4, 5, 7)$$

$$F_f(A, B, C) = \sum(0, 4, 5, 6) = \prod(1, 2, 3, 7)$$

$$8. \quad F_g(A, B, C) = \sum(2, 3, 4, 5, 6) = \prod(0, 1, 7)$$

y simplificar por Karnaugh

$$9. \quad F(a, b, c, d) = \bar{a} + \bar{b}\bar{c}$$

10. Basta con utilizar un sumador binario de cuatro bits en el que introducimos cada uno de los números BCD en cada uno de los sumandos. La suma son los cuatro bits del sumador y el acarreo de salida como MSB de la suma (que como máximo valdrá $9+9=18$ (10010)).

11. Se hace la tabla de verdad que tiene como entrada los cuatro bits de los dos números a restar (a_2, a_1, b_2, b_1), y como salidas los dos bits del valor absoluto de la resta (r_2, r_1) (que como máximo puede valer $3-0=3$). A partir de la tabla de verdad se obtienen las expresiones canónicas para r_2 y r_1 ,

$$r_1(a_2, a_1, b_2, b_1) = \sum(2, 3, 7, 8, 12, 13)$$

$$r_2(a_2, a_1, b_2, b_1) = \sum(1, 3, 4, 6, 9, 11, 12, 14)$$

se simplifican y se diseña el circuito.


12. Véanse los apuntes de teoría y la práctica 3 del laboratorio.
13. a) Se hace la tabla de verdad que tiene como entrada los cuatro bits de los dos números a comparar (a_2, a_1, b_2, b_1), y como salidas los tres bits de la comparación (S_1, S_2, S_3). A partir de la tabla de verdad se obtienen las expresiones canónicas para (S_1, S_2, S_3),

$$S_1(a_2, a_1, b_2, b_1) = \sum (4, 8, 9, 12, 13, 14)$$

$$S_2(a_2, a_1, b_2, b_1) = \sum (0, 5, 10, 15)$$

$$S_3(a_2, a_1, b_2, b_1) = \sum (1, 2, 3, 6, 7, 11)$$

se simplifican y se diseña el circuito.

b) Fijándonos en la comparación, se ve que S_2 sólo puede ser 1 cuando S_1 y S_3 son los dos cero, por lo tanto ~~$S_2 = (S_1 S_3)$~~ 

14. A las entradas de un multiplexor 4 --> 1 (con tres bits para cada entrada y la salida) se conectan '111', la salida del sumador de A y B, del comparador de A y B, y '000'. Según las entradas de control C_2 y C_1 obtendremos una u otra salida de tres bits en el multiplexor.
15. Muy parecido a ejercicios 11 y 13.
16. Se va evaluando la paridad de cada dos bits con puertas XOR cuyas salidas se conectan a las entradas de otra fila de XOR hasta obtener un único bit que indica la paridad total (3 filas de XOR con 7 puertas en total).
17. Consultando la hoja de características del comparador 7485, conectar 8 de ellos en cascada mediante la unión de los pines adecuados para obtener un comparador de 32 bits.
18. Los tres bits de menor peso del número de cuatro bits a decodificar (A_2, A_1, A_0) se conectan a las entradas de cada uno de los dos decodificadores 3 a 8 a utilizar. El bit de mayor peso A_3 se conecta negado a la entrada 'I' del decodificador 3 a 8 que proporcionará las 8 salidas de menor peso (S_0-S_7), y sin negar al que proporcionará las 8 salidas de mayor peso (S_8-S_{15}).
19. Se hace la tabla de verdad que tiene como entrada los cuatro bits del número BCD (A, B, C, D), y como salidas los cuatro bits del mismo número pero representado en exceso a tres (a, b, c, d). A partir de la tabla de verdad se obtienen las expresiones canónicas para (a, b, c, d), se simplifican y se diseña el circuito.
20. Igual que el problema 8.