

Examen 2022/23-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	28/6/2023	17:00

Este enunciado también corresponde a las siguientes asignaturas:

- 81.518 - Fundamentos de computadores

Ficha técnica del examen

- No es necesario que escribas tu nombre. Una vez resuelta la prueba final, solo se aceptan documentos en formato .doc, .docx (Word) y .pdf.
 - Comprueba que el código y el nombre de la asignatura corresponden a la asignatura de la que te has matriculado.
 - Tiempo total: **2 horas** Valor de cada pregunta: **P1:20%; P2:35%; P3:35%; P4:10%**
 - ¿Puede consultarse algún material durante el examen? **NO** ¿Qué materiales están permitidos? **Ninguno**
 - ¿Puede utilizarse calculadora? **NO** ¿De qué tipo? **NINGUNO**
 - Si hay preguntas tipo test, ¿descuentan las respuestas erróneas? **NO** ¿Cuánto?
 - Indicaciones específicas para la realización de este examen: **Razonad todas las respuestas. Las respuestas sin justificar no serán puntuadas.**
-

Examen 2022/23-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	28/6/2023	17:00

Enunciados

PROBLEMA 1 [20%]

- a) **[10%]** Considerad los números $A = 01010110$ y $B = 10101101$, que representan números binarios enteros codificados en signo y magnitud con 8 bits. Realizad la operación $A + B$ en signo y magnitud manteniendo el formato de representación con 8 bits. Explicad si se produce o no desbordamiento y por qué razón. Indicad cuál es el rango del resultado de esta operación expresándolo en decimal.

El primer bit codifica el signo así que A es positivo y B es negativo. Los dos números son de signo diferente, con lo cual para obtener la magnitud de la suma de los dos números hay que restar de la magnitud de A , que es la magnitud mayor, la magnitud de B , que es la menor ($|A| - |B|$). Así pues:

$$\begin{array}{r}
 \begin{array}{ccccccc}
 -1 & & -1 & & -1 & & \leftarrow \text{acarreo} \\
 1 & 0 & 1 & 0 & 1 & 1 & 0 \leftarrow |A| \\
 - & 0 & 1 & 0 & 1 & 1 & 0 \leftarrow |B| \\
 \hline
 0 & 1 & 0 & 1 & 0 & 0 & 1
 \end{array}
 \end{array}$$

No hay acarreo final y, por lo tanto, **no hay desbordamiento**. El resultado final es el resultado de esta operación añadiendo el signo del primer operando (A), el que tiene la magnitud mayor. Es decir: **00101001**_(SM2).

El rango de representación de un número en signo y magnitud es $[-(2^{n-1}-1)..2^{n-1}-1]$. El rango de representación del resultado de esta operación es el mismo puesto que podemos obtener estos mismos valores extremos con múltiples combinaciones de los operandos de entrada. Por lo tanto, en este caso que tenemos números de 8 bits ($n=8$) el rango es $[-(2^7-1)..2^7-1] = [-127..127]$.

- b) **[5%]** Dado el número $C = 01010110$, que representa un número binario entero codificado en signo y magnitud con 8 bits, indicad su valor en base 8.

En SM2, el primer bit codifica el signo y el resto de bits la magnitud. Así pues, es un número positivo, con la magnitud 1010110. Puesto que podemos transformar directamente un número binario a un número octal (base 8), no hace falta obtener su valor decimal. Tan solo tenemos que hacer grupos de tres bits de derecha a izquierda y sustituir cada grupo de 3 bits por su valor correspondiente:

$$\begin{array}{r}
 001 \ 010 \ 110 \leftarrow \text{binario} \\
 1 \quad 2 \quad 6 \leftarrow \text{octal}
 \end{array}$$

Así pues, el resultado es **+126**₍₈₎.

- c) **[5%]** Dado el formato de coma flotante siguiente:

S	Exponente			Mantisa		
12	11		8	7		0

Donde:

- El bit de signo (posición 12), S , vale 0 para cantidades positivas y 1 para negativas.
- El exponente se representa en exceso a 8, con 4 bits de la posición 11 a la 8.

Examen 2022/23-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	28/6/2023	17:00

- La mantisa está normalizada en la forma $1,X$. Se representa con bit implícito y con 8 bits de la posición 7 a la 0.

¿A qué número decimal corresponde la secuencia de bits 0101001010100 codificado en este formato de coma flotante?

Analizando el número según el formato tenemos:

- Signo: positivo, $S=0$
- Exponente: 1010_2 en exceso a 8, $10-8 = 2_{(10)}$.
- Mantisa: $1,01010100_2$.

Si juntamos la mantisa y el exponente, obtenemos el número:

$$1,01010100_2 \cdot 2^2 = 101,010101_2$$

Y si aplicamos el TFN:

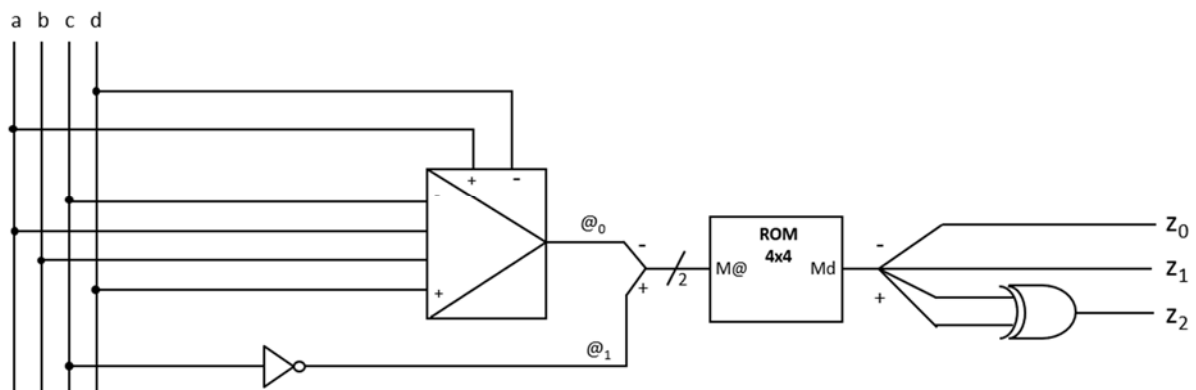
Parte entera: $1 \cdot 2^2 + 1 \cdot 2^0 = 4 + 1 = 5$

Parte decimal: $1 \cdot 2^{-2} + 1 \cdot 2^{-4} = 0,25 + 0,0625 = 0,3125$

Por lo tanto, en decimal el número que nos han dado es **+5,3125₍₁₀₎**.

PROBLEMA 2 [35%]

a) [10%] Dado el circuito lógico combinacional siguiente:



Donde el contenido de la memoria ROM, especificado en hexadecimal, es:

M@	[Md]
0	7
1	A
2	D
3	0

Completad la tabla de verdad que especifica las salidas z_0 , z_1 , z_2 en función de las entradas a , b , c y d . Calculad también, previamente, los valores intermedios indicados en el circuito y añadidos a la tabla de verdad siguiente:

Examen 2022/23-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	28/6/2023	17:00

a	b	c	d	@ ₁	@ ₀	Z ₂	Z ₁	Z ₀
0	0	0	0					
0	0	0	1					
0	0	1	0					
0	0	1	1					
0	1	0	0					
0	1	0	1					
0	1	1	0					
0	1	1	1					
1	0	0	0					
1	0	0	1					
1	0	1	0					
1	0	1	1					
1	1	0	0					
1	1	0	1					
1	1	1	0					
1	1	1	1					

En primer lugar, para completar las columnas de los valores intermedios, buscamos sus funciones lógicas:

- $@_0 = ca'd' + aa'd + bad' + dad = a'cd' + abd' + ad$
- $@_1 = c'$

Y evaluamos estas funciones para cada entrada de la tabla de verdad:

a	b	c	d	@ ₁	@ ₀	Z ₂	Z ₁	Z ₀
0	0	0	0	1	0			
0	0	0	1	1	0			
0	0	1	0	0	1			
0	0	1	1	0	0			
0	1	0	0	1	0			
0	1	0	1	1	0			
0	1	1	0	0	1			
0	1	1	1	0	0			
1	0	0	0	1	0			
1	0	0	1	1	1			
1	0	1	0	0	0			
1	0	1	1	0	1			
1	1	0	0	1	1			
1	1	0	1	1	1			
1	1	1	0	0	1			
1	1	1	1	0	1			

Para obtener los valores de las salidas z₂, z₁ y z₀ analizamos cuál es la salida de la ROM en cada caso, y evaluamos las funciones lógicas de las tres salidas:

- $z_0 = Md_0$
- $z_1 = Md_1$
- $z_2 = Md_2 \text{ (xor) } Md_3 = Md_2'Md_3 + Md_2Md_3'$

Examen 2022/23-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	28/6/2023	17:00

a	b	c	d	@ ₁	@ ₀	Md	z ₂	z ₁	z ₀
0	0	0	0	1	0	(D) 1101	0	0	1
0	0	0	1	1	0	(D) 1101	0	0	1
0	0	1	0	0	1	(A) 1010	1	1	0
0	0	1	1	0	0	(7) 0111	1	1	1
0	1	0	0	1	0	(D) 1101	0	0	1
0	1	0	1	1	0	(D) 1101	0	0	1
0	1	1	0	0	1	(A) 1010	1	1	0
0	1	1	1	0	0	(7) 0111	1	1	1
1	0	0	0	1	0	(D) 1101	0	0	1
1	0	0	1	1	1	(0) 0000	0	0	0
1	0	1	0	0	0	(7) 0111	1	1	1
1	0	1	1	0	1	(A) 1010	1	1	0
1	1	0	0	1	1	(0) 0000	0	0	0
1	1	0	1	1	1	(0) 0000	0	0	0
1	1	1	0	0	1	(A) 1010	1	1	0
1	1	1	1	0	1	(A) 1010	1	1	0

b) [10%] Dada la tabla de verdad siguiente:

a	b	c	d	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	X
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	X
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Sintetizad de manera mínima a dos niveles la función f mediante el método de Karnaugh. No hace falta que implementéis el circuito.

El mapa de Karnaugh de la función f es el siguiente:

Examen 2022/23-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	28/6/2023	17:00

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

Y obtenemos esta expresión mínima:

$$f = a'd' + b'c'd$$

- c) [15%] Se quiere diseñar un circuito combinacional llamado *MULT* con la estructura siguiente:



Las entradas *A* y *B* representan números naturales de 8 y 2 bits, respectivamente. La salida *X*, que representa también un número binario natural de 10 bits, tiene que ser el resultado de multiplicar $A * B$. Diseñad el circuito *MULT*, especificando claramente la dimensión de todos los buses utilizados y usando los bloques y puertas combinacionales vistos en clase (es decir, no se puede asumir que existe un bloque combinacional que realiza multiplicaciones).

Para implementar este circuito, consideraremos cuatro casos en función de los valores que tome la entrada *B*. Habrá una parte del circuito que implementará la función necesaria para cada uno de estos casos, y después tendremos que seleccionar el caso específico en función del valor de *B*.

Si la entrada $B=0$, el resultado de la operación tiene que ser 0.

Si la entrada $B=1$, el resultado de la operación tiene que ser *A*, pero con una precisión de 10 bits. Es decir, tenemos que añadir dos bits más a la izquierda con valor 0.

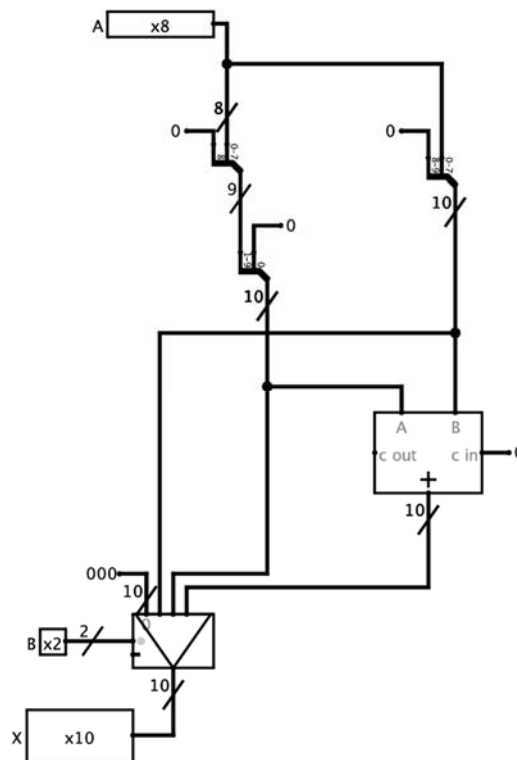
Si la entrada $B=2$, el resultado de la operación tiene que ser el doble de *A*. Si consideramos que la multiplicación por 2 de un número binario se realiza con un desplazamiento a la izquierda, el resultado para este caso se obtiene añadiendo un cero a la derecha y otro a la izquierda (para conseguir una precisión de 10 bits).

Examen 2022/23-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	28/6/2023	17:00

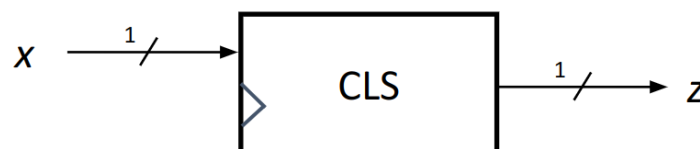
Finalmente, si la entrada es $B=3$, es necesario hacer la multiplicación por 3. Una multiplicación por 3 es igual a hacer una multiplicación por 2 y añadir una vez el valor a multiplicar. Es decir, $A \times 3 = A \times 2 + A$. Así pues, en este caso podemos coger el resultado del caso anterior, y sumar el valor de A .

Finalmente, para poder seleccionar entre cada uno de los cuatro casos posibles, tenemos que añadir un MUX con dos bits de selección. La entrada B del circuito irá conectada a la entrada de selección del MUX.



PROBLEMA 3 [35%]

- a) **[17.5%]** Se quiere diseñar el grafo de estados de un circuito lógico secuencial (CLS) que mantenga a cero su señal de salida z , de un bit, mientras no reciba dos ceros o dos unos consecutivos en su entrada x , también de un solo bit.



Tras recibir 00 o 11, la salida z continuará a cero, pero acto seguido y durante los dos ciclos de reloj siguientes, la salida z valdrá el mismo valor que la entrada x . Una vez finalizados estos dos ciclos de reloj, independientemente del valor de la entrada x , tiene que volver al estado inicial.

Examen 2022/23-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	28/6/2023	17:00

A continuación, se muestra un ejemplo del funcionamiento del circuito, en que cada columna corresponde a un ciclo de reloj y donde con gris oscuro se muestra la detección de la secuencia 00 o 11 y con gris claro los dos ciclos siguientes donde el valor de x se copia en la salida z :

entrada x	0	1	1	0	1	0	1	0	0	1	1	0	0	1	0	
salida z	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0

Diseñad el grafo de estados del circuito CLS, especificando claramente el significado de cada estado.

Para conseguir el funcionamiento deseado, el circuito debe tener los siguientes estados:

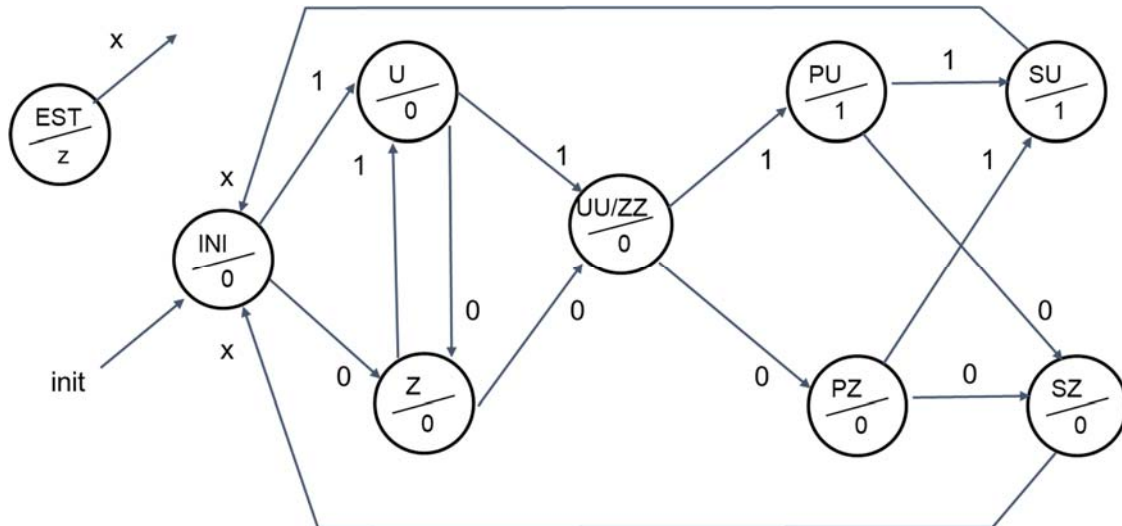
Estado	Descripción	Salida
<i>INI</i>	Estado inicial.	0
<i>Z</i>	El valor de la entrada x era 0.	0
<i>U</i>	El valor de la entrada x era 1.	0
<i>UU/ZZ</i>	La entrada x ha tenido la secuencia 00 o 11 en los dos estados anteriores	0
<i>PZ</i>	Primer valor donde la salida z toma el mismo valor de x , donde el valor anterior de x era 0.	0
<i>PU</i>	Primer valor donde la salida z toma el mismo valor de x , donde el valor anterior de x era 1.	1
<i>SZ</i>	Segundo valor donde la salida z toma el mismo valor de x , donde el valor anterior de x era 0.	0
<i>SU</i>	Segundo valor donde la salida z toma el mismo valor de x , donde el valor anterior de x era 1.	1

A partir del estado inicial *INI*, haremos una transición a *Z* o *U*, en función del valor de la entrada x . Al estado *INI* no se puede volver a no ser que se haya producido la secuencia 00 o 11 y hayan pasado dos ciclos de reloj más. En el estado *INI*, haremos una transición hacia los estados *Z* y *U*, en función del valor de x . A partir de aquí, haremos transiciones entre *Z* y *U* mientras la entrada x no tenga el mismo valor durante dos ciclos consecutivos de reloj. Si esto pasa, entonces hacemos una transición hacia el estado *UU/ZZ*. En este punto, en los dos próximos ciclos de reloj haremos las transiciones hacia los estados donde la salida z tomará el mismo valor que haya tenido la entrada x . De esta manera, una vez en el estado *UU/ZZ*, haremos la primera transición hacia *PZ* o *PU*, y desde estos dos estados haremos la segunda transición hacia *SZ* o *SU*. *Estando en SZ o SU*, en el siguiente ciclo de reloj haremos una transición hacia el estado inicial *INI*.

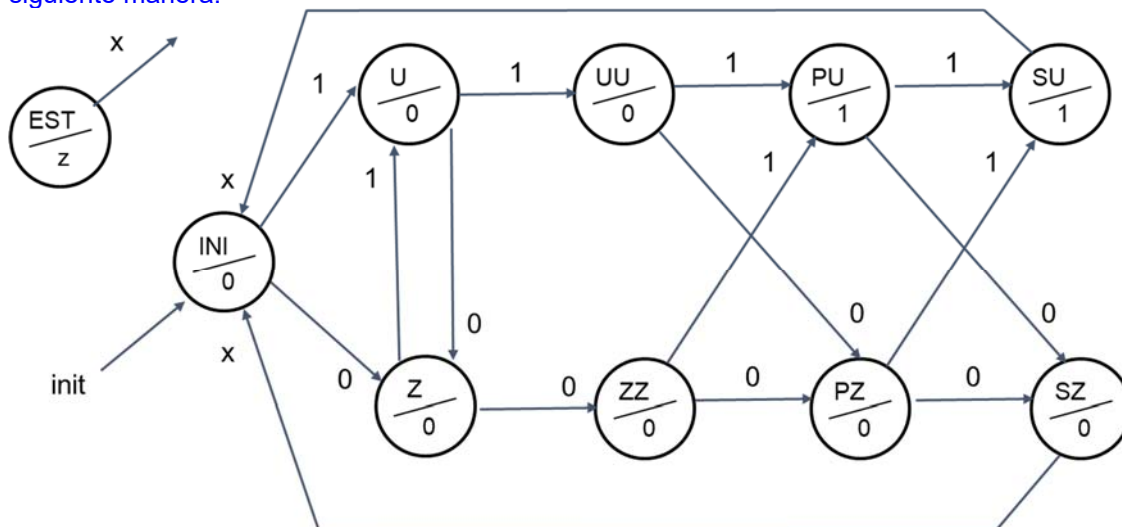
El siguiente grafo representa el comportamiento deseado:

Examen 2022/23-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	28/6/2023	17:00



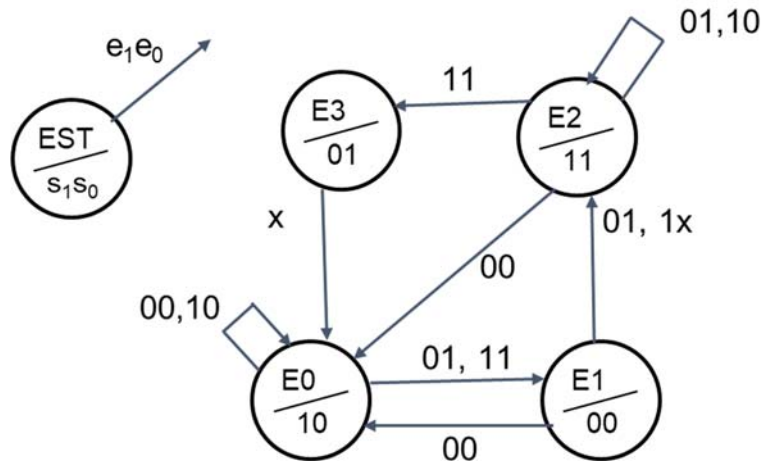
El comportamiento se puede representar también con un grafo donde se diferencia cuando la entrada x ha tenido la secuencia 00 (estado ZZ) o 11 (Estado UU). En este caso el diagrama quedaría de la siguiente manera.



Examen 2022/23-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	28/6/2023	17:00

b) [17.5%] Dado el grafo de estados siguiente:



- i. [10%] Rellenad la tabla de transiciones y salidas siguiente (fijaos que se han unificado en una única tabla de excitaciones), codificando los estados según su índice asociado. No hace falta que justifiquéis la respuesta.

q_1	q_0	e_1	e_0	q_1^+	q_0^+	s_1	s_0

En la tabla, las columnas q_1q_0 representan el estado y las columnas $q_1^+q_0^+$ el estado futuro, es decir, el estado en el ciclo siguiente a partir del valor de las entradas e_1e_0 . Para obtener la tabla de transiciones observamos en el grafo a qué estado futuro se llega desde cada estado con cada combinación de valores de las entradas. Vemos que estando en el estado $E3$ para cualquier valor de las entradas (condiciones don't care) se llega al estado $E0$.

La tabla de salidas expresa el valor de las salidas s_1s_0 en cada estado, teniendo en cuenta que las salidas dependen del estado actual, no del futuro.

q_1	q_0	e_1	e_0	q_1^+	q_0^+	s_1	s_0
0	0	0	0	0	0	1	0

Examen 2022/23-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	28/6/2023	17:00

0	0	0	1	0	1	1	0
0	0	1	0	0	0	1	0
0	0	1	1	0	1	1	0
0	1	0	0	0	0	0	0
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	0
0	1	1	1	1	0	0	0
1	0	0	0	0	0	1	1
1	0	0	1	1	0	1	1
1	0	1	0	1	0	1	1
1	0	1	1	1	1	1	1
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	0	1

- ii. **[7.5%]** Implementad el circuito representado por la tabla de salidas y excitaciones del apartado anterior usando una memoria ROM. Especificad en hexadecimal el contenido de las primeras seis posiciones de la memoria ROM.

El circuito tiene dos bits de entrada, denominados e_1 y e_0 y dos bits de salida, denominados s_1 y s_0 . Como el circuito tiene 4 estados, $\log_2(4) = 2$, necesitamos 2 bits para representarlos y, por lo tanto, necesitamos 2 biestables para poder almacenar estos 2 bits. La memoria ROM que implemente este circuito tendrá 4 bits de direccionamiento, 2 bits para el estado más 2 bits para la entrada. El ancho de las palabras de la ROM debe ser de 4 bits, 2 bits para guardar la salida y 2 bits para codificar el estado futuro.

El circuito que implementa el sistema secuencial se muestra a continuación.

Examen 2022/23-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	28/6/2023	17:00

PROBLEMA 4 [10%]

- a) **[5%]** En un procesador microprogramado, ¿qué función tiene el secuenciador?

[De controlador de una máquina de interpretación de instrucciones.](#)

- b) **[5%]** ¿Qué es la arquitectura de Von Neumann?

[Una manera de construir máquinas que tienen una memoria común para las instrucciones y los datos.](#)