

## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	20/06/2018	12:00

75.562 20 06 18 EX  
75.562 20 06 18 EX

Espacio para la etiqueta identificativa con el código  
personal del **estudiante**.  
Examen

**Este enunciado corresponde también a las siguientes asignaturas:**

- 81.518 - Fundamentos de computadores

### Ficha técnica del examen

- Comprueba que el código y el nombre de la asignatura corresponden a la asignatura matriculada.
- Debes pegar una sola etiqueta de estudiante en el espacio correspondiente de esta hoja.
- No se puede añadir hojas adicionales, ni realizar el examen en lápiz o rotulador grueso.
- Tiempo total: **2 horas** Valor de cada pregunta: **Prob. 1: 20%; Prob. 2: 35%; Prob. 3: 35%; Prob. 4: 10%.**
- En el caso de que los estudiantes puedan consultar algún material durante el examen, ¿cuáles son?: **No se puede consultar ningún tipo de material.**
- En el caso de poder usar calculadora, de que tipo? **NINGUNA**
- En el caso de que haya preguntas tipo test: ¿descuentan las respuestas erróneas? **NO** ¿Cuánto?

Indicaciones específicas

– **Razonad las respuestas en cada ejercicio. Las respuestas sin justificar no obtendrán puntuación.**

## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	20/06/2018	12:00

### Enunciados

#### PROBLEMA 1 [20%]

Considerad el número binario  $A = 01111010$  y el número decimal  $B = -13$ , junto con el formato de representación de los números enteros en **Signo y Magnitud y en 8 bits**. Contestad los siguientes apartados:

- a) [5%] Si  $A$  representa un número entero en el formato dado en este ejercicio, indicad cuál es el valor en decimal de este número entero.

El primer bit en SM2 codifica el signo. En  $01111010_{(SM2)}$ , el primer bit es 0, por lo tanto el número tiene un signo positivo.

El valor de la magnitud  $1111010_2$  lo podemos obtener mediante el TFN:

$$1111010_2 = 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^1 = 64 + 32 + 16 + 8 + 2 = 122_{(10)}$$

Añadiendo el signo, obtenemos  $01111010_{(SM2)} = +122_{(10)}$

- b) [5%] Representad el número entero  $B$  en el formato dado.

El signo se codifica con 1 en el bit de más peso, puesto que el número es negativo. La magnitud contiene la representación binaria.

Usamos el método de la división entera por 2 para obtener el valor binario de 13 representado en decimal:

$$\begin{array}{rcll} 13 & = & 6 \cdot 2 & + 1 \\ 6 & = & 3 \cdot 2 & + 0 \\ 3 & = & 1 \cdot 2 & + 1 \\ 1 & = & 0 \cdot 2 & + 1 \end{array} \quad \uparrow$$

Añadimos ceros y tenemos  $13_{(10)} = 0001101_2$  para la magnitud en 7 bits.

Añadimos el bit de signo  $-13 = 10001101_{(SM2)}$

- c) [10%] Realizad la resta de los dos números  $A - B$  usando el formato de representación dado. Indicad el resultado y si se produce desbordamiento en el cálculo de este resultado.

El primer operando es positivo y el segundo operando es negativo. La operación de resta se convierte en una operación de suma de las magnitudes y añadimos al resultado el signo positivo.

$$\begin{array}{r} 1 \ 1 \ 1 \ 1 \qquad \leftarrow \text{acarreo} \\ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \leftarrow A \\ + 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \leftarrow B \\ \hline 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \end{array}$$

Hay acarreo en la última etapa de la suma de las magnitudes, por lo tanto, hay desbordamiento.

## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	20/06/2018	12:00

### PROBLEMA 2 [35%]

- a) [12,5%] Dada la tabla de verdad siguiente, donde  $a$ ,  $b$ ,  $c$  y  $d$  son las entradas del sistema y  $f$  es la salida. Minimiza la salida  $f$  mediante el método de Karnaugh a dos niveles e implementa el resultado con puertas lógicas.

$a$	$b$	$c$	$d$	$f$
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

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

$a \backslash b$ $c \backslash d$	00	01	11	10
00	1	0	0	1
01	0	1	0	0
11	0	0	1	1
10	1	0	0	1

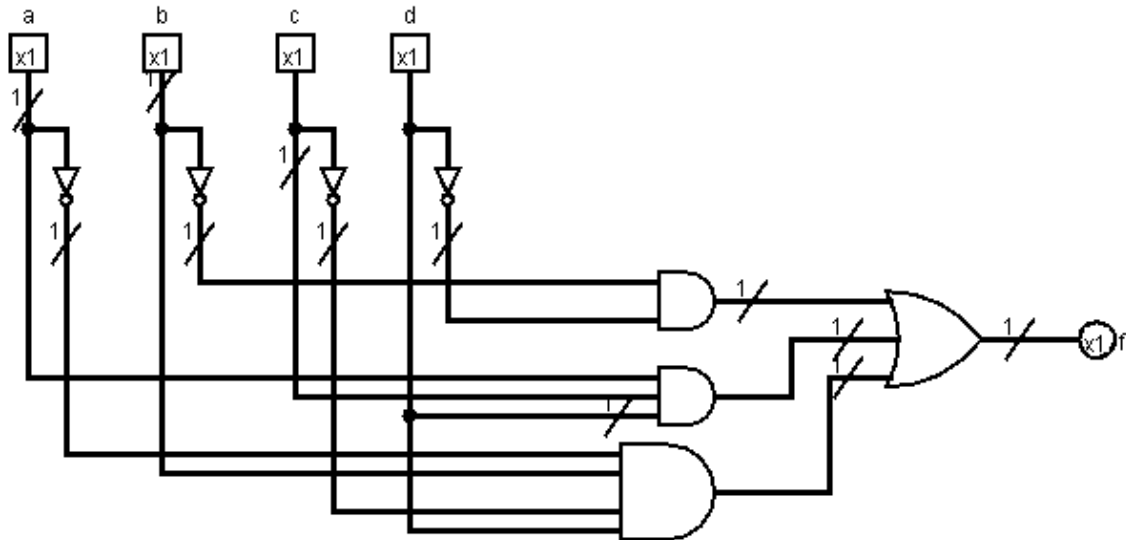
Y obtenemos la siguiente expresión mínima:

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

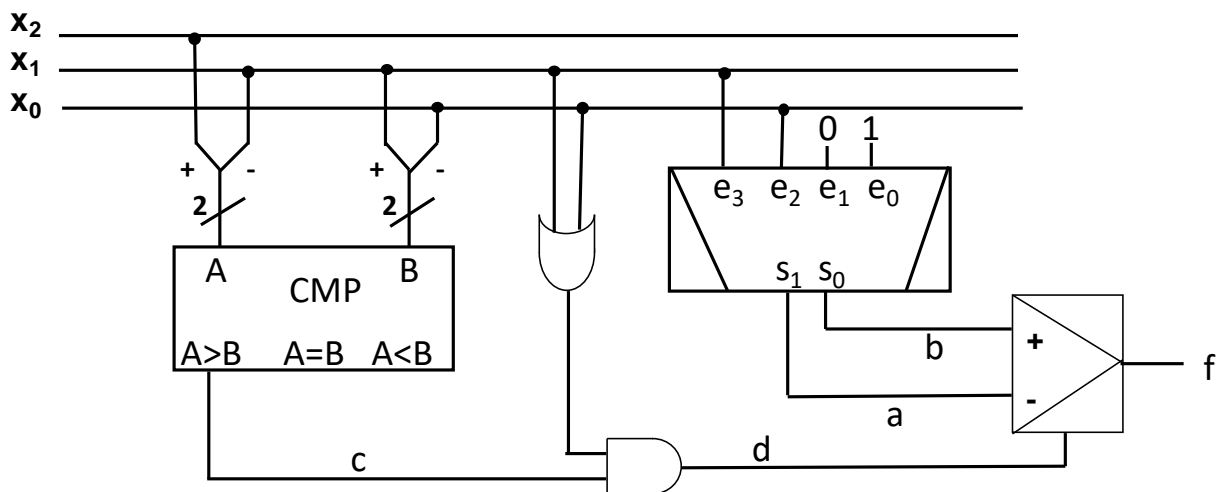
## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	20/06/2018	12:00

Esta función la implementamos con el siguiente circuito:



- b) [10%] Dado el siguiente circuito lógico combinacional. Completad la siguiente tabla de verdad del sistema incluyendo los valores de las señales intermedias ( $a$ ,  $b$ ,  $c$  y  $d$ ):



$x_2$	$x_1$	$x_0$	$a$	$b$	$c$	$d$	$f$
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	20/06/2018	12:00

El valor de las salidas del codificador depende de cuál sea la entrada de más peso que esté a 1.

- A la entrada de mayor peso ( $e_3$ ) se le ha conectado  $x_1$ , de forma que siempre que  $x_1$  valga 1 saldrá el valor  $3_{10}$ , codificado en binario 11. Por lo tanto, si  $x_1$  vale 1, las señales  $a$  y  $b$  valdrán las dos 1.
- Cuando  $x_1$  valga 0, la salida depende de la entrada  $e_2$ , a la que se le ha conectado la entrada  $x_0$ . Por lo tanto, si  $x_1=0$  y  $x_0=1$  por el codificador saldrá un  $2_{10}$  codificado en binario 10, lo que hará que  $a=1$  y  $b=0$ .
- Si  $x_1$  y  $x_0$  valen 0, como la entrada  $e_1$  tiene conectada la constante 0 y la  $e_0$  la constante 1 la salida del codificador será 00. Por lo tanto, si  $x_1=0$  y  $x_0=0$  las señales  $a$  y  $b$  valdrán las dos 0.

La señal  $c$  es la salida de un comparador de dos bits,  $c$  valdrá 1 cuando  $[x_2, x_1]$  sea mayor que  $[x_1, x_0]$ . Esto sucede cuando  $[x_2, x_1]=10$  y  $[x_1, x_0]=00$ , cuando  $[x_2, x_1]=10$  y  $[x_1, x_0]=01$ , y cuando  $[x_2, x_1]=11$  y  $[x_1, x_0]=10$ .

La señal  $d$  es igual  $(x_1 \text{ OR } x_0) \text{ AND } c$ . Por lo tanto, solamente valdrá 1 cuando  $c=1$ , y valga 1  $x_1$  o  $x_0$ .

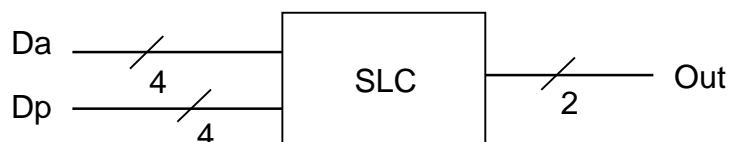
Finalmente, la salida del multiplexor  $f$  depende de la señal  $d$  que está conectada a su entrada de selección. La salida  $f=a$  si  $d=0$ , y  $f=b$  si  $d=1$ .

$x_2$	$x_1$	$x_0$	$a$	$b$	$c$	$d$	$f$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1
0	1	0	1	1	0	0	1
0	1	1	1	1	0	0	1
1	0	0	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	1	1	1	1	1
1	1	1	1	1	0	0	1

## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	20/06/2018	12:00

- c) **[12,5%]** Un sistema tiene dos entradas  $Da$  y  $Dp$  de cuatro bits que indican distancias en binario natural y una salida  $Out$  de dos bits.



El sistema internamente debe realizar la operación  $R = 5 \times (Dp + Da)$ . En función del valor de  $R$ , la señal  $Out$  tiene que tomar los siguientes valores:

- $Out = 10$  si  $R$  es mayor que 30.
- $Out = 00$  si  $R$  es igual a 30.
- $Out = 01$  si  $R$  es menor que 30.

Diseñad la salida  $Out$  del circuito indicando el ancho de todos los buses.

Para implementar este circuito primero hemos de calcular  $R$  y a continuación en función de su valor se generará la salida  $Out$ .

La representación de la información en binario nos facilita realizar productos con potencias de 2. Por lo tanto, para calcular  $R$  haremos:  $R = (Dp + Da) + 4 \times (Dp + Da)$ .

Con un sumador de 4 bits sumaremos  $Da$  y  $Dp$ , Como necesitamos 1 bit más para la suma agregamos el  $Cout$  como bit de más peso, por lo que el resultado de la suma,  $Da + Dp$ , será un bus de 5 bits.

Para multiplicar por 4, agregamos dos bits a 0 en los dos bits de menor peso (pasa de 5 a 7 bits). También se puede hacer la multiplicación con un desplazador de 2 bits a la izquierda.

A continuación, hemos de sumar  $4 \times (Da + Dp)$  con  $(Da + Dp)$ . Para poder sumar, los dos operandos han de tener el mismo tamaño, por lo tanto, tenemos que ampliar el bus que lleva el operando  $Da + Dp$  a 7 bits, para esto le agregamos 2 bits a 0 por la izquierda.

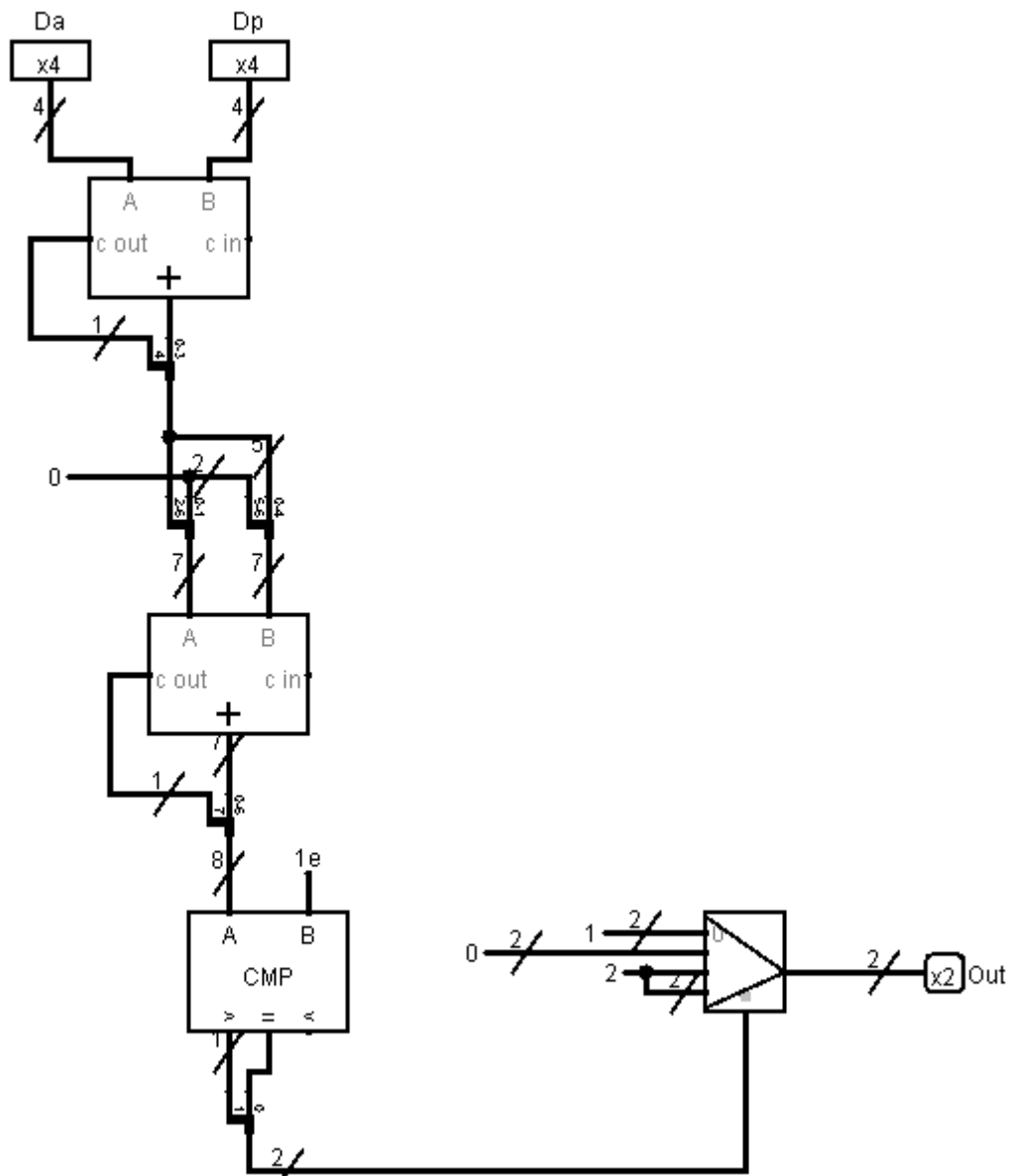
Una vez transformados los operandos los conectamos a un sumador de 7 bits, e igual que en el sumador anterior necesitamos 1 bit más para la suma, por lo que también agregaremos el  $Cout$  como bit de más peso, pasando el bus a 8 bits.

Finalmente, conectaremos  $R$  a un comparador de 8 bits para compararlo con 30, la salidas  $[>,=]$  de este comparador se conectarán a la entrada de selección de un multiplexor para escoger el valor de  $Out$ .

- Si  $[>,=] = 00$ , significa que  $R < 30$ . Por lo tanto, hemos de conectar 01 a la entrada 0 del multiplexor.
- Si  $[>,=] = 01$ , significa que  $R = 30$ . Por lo tanto, hemos de conectar 00 a la entrada 1 del multiplexor.
- Si  $[>,=] = 10$ , significa que  $R > 30$ . Por lo tanto, hemos de conectar 10 a la entrada 2 del multiplexor.
- Es imposible que  $[>,=] = 11$ , ya en un comparador solamente una de las salidas vale 1. Por lo tanto, a la entrada 3 del multiplexor se le puede conectar cualquier valor.

## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	20/06/2018	12:00

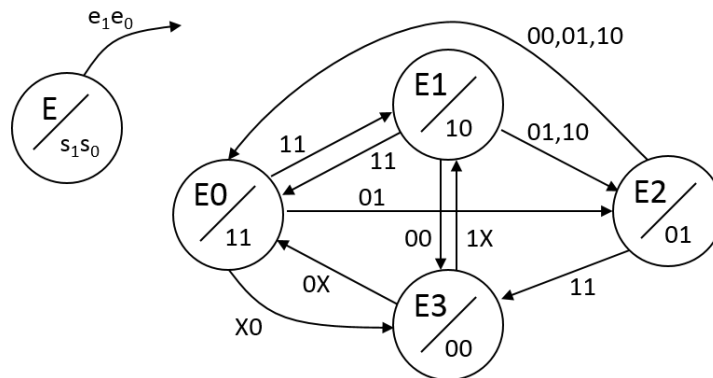


## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	20/06/2018	12:00

### PROBLEMA 3 [35%]

a) [15%] Dado el grafo de estados siguiente:



- i) [5%] ¿Cuántos bits de entrada tiene el circuito que implementa este grafo? ¿Cuántos bits de salida? ¿Cuál será el número mínimo de biestables para implementarlo? Si lo implementamos usando una memoria ROM, ¿cuál será la medida (bus de direcciones, dimensión de las palabras) de esta memoria?

El circuito tiene dos bits de entrada, denominados  $e_1 e_0$ . Los bits de salida son dos también, denominados  $s_1 s_0$ . Como que el circuito tiene 4 estados necesitamos 2 bits para representarlos y, por lo tanto, necesitamos 2 biestables para poder almacenar estos 2 bits.

La memoria ROM que puede implementar este circuito tendría 4 bits de direccionamiento, 2 bits por el estado más dos bits para las entradas. Cada entrada de la ROM tendría que contener 4 bits también, 2 bits para guardar la salida y 2 bits para codificar el estado futuro.

- ii) [10%] Asumid que la codificación de cada estado corresponde al número del estado (por ejemplo, el estado  $E0$  se codifica con el valor 00) y un diseño habitual del circuito secuencial. Indicad en la tabla adjunta, el contenido **en binario** de las 10 primeras posiciones de la memoria ROM. Considerad el orden de peso de los bits que se utiliza en los materiales, es decir, para la dirección primero va la codificación del estado y después las entradas del circuito y para el contenido primero va la codificación del estado y después las salidas.

Dirección	Contenido
0x00	
0x01	
0x02	
0x03	
0x04	
0x05	



## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	20/06/2018	12:00

0x06	
0x07	
0x08	
0x09	

El direccionamiento de la memoria ROM viene definido por las señales  $q_1$   $q_0$   $e_1$   $e_0$  especificados en orden de mayor a menor peso. Los dos primeros bits representan el estado y los otros dos son las entradas.

Cada posición de la memoria almacena los bits  $q_1^+$   $q_0^+$   $s_1$   $s_0$ , especificados en orden de mayor a menor peso. Los dos primeros bits especifican el estado futuro y los otros dos son las salidas.

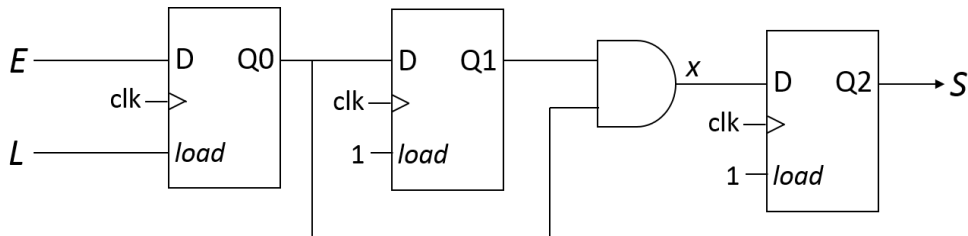
El contenido de las 10 primeras posiciones de la memoria ROM será:

Dirección	Contenido
0x00	11 11
0x01	10 11
0x02	11 11
0x03	01 11
0x04	11 10
0x05	10 10
0x06	10 10
0x07	00 10
0x08	00 01
0x09	00 01

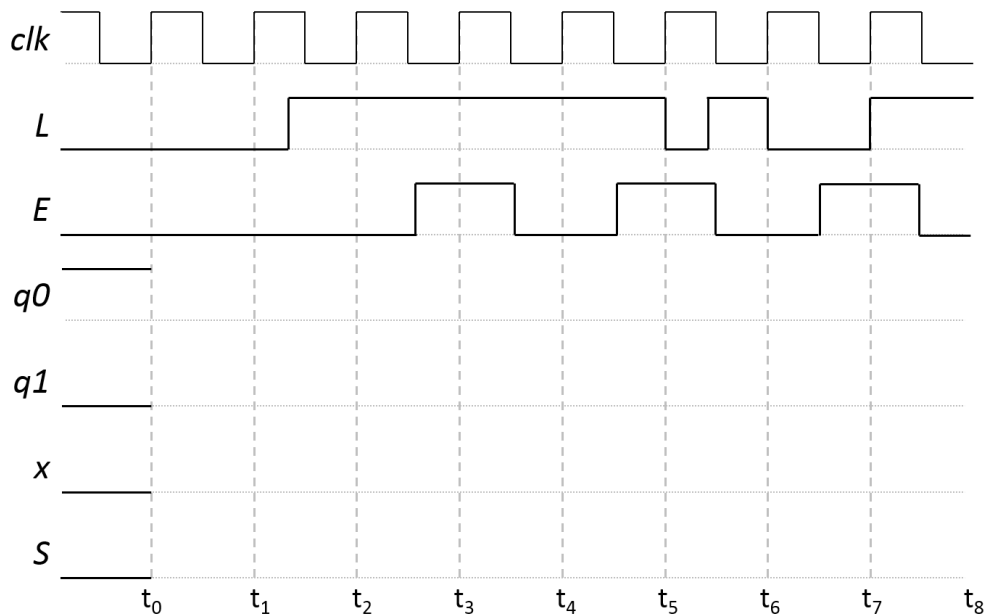
## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	20/06/2018	12:00

b) [20%] Dado el circuito siguiente:



Completad el cronograma siguiente:



Empezamos analizando la señal Q0. El biestable asociado a esta señal se carga con la entrada  $E$  si la señal de carga  $L=1$ . Esta señal no depende de ninguna otra señal y, por lo tanto, podemos hacer el cronograma de  $Q0$  para todos los ciclos. Los instantes en que el biestable se carga son:  $t_2$ ,  $t_3$ ,  $t_4$ ,  $t_5$ , y  $t_6$ .

El biestable asociado a la señal  $Q1$  se carga a cada ciclo de reloj con la señal  $Q0$ . Como que ya hemos calculado la señal  $Q0$  para todo el cronograma, ahora podemos completar la señal  $Q1$  para todo el cronograma también.

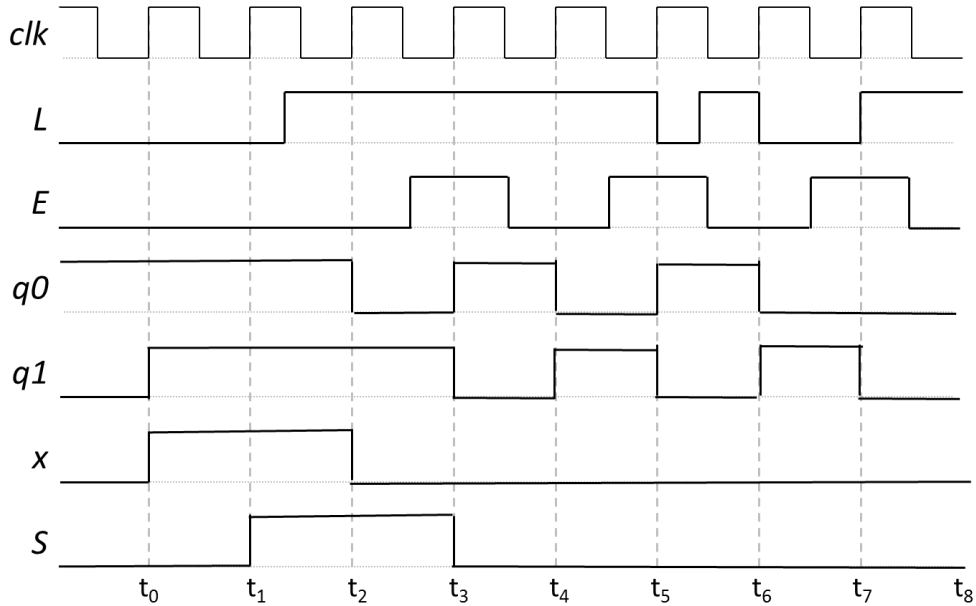
Para hacer la señal  $Q2$  necesitamos calcular antes la señal  $X$ . Esta señal  $X$  es la salida de una puerta AND entre los bits  $Q1$  y  $Q0$ . Ahora, por lo tanto, ya podemos completar esta señal combinacional  $X$  para todo el cronograma.

Por último, pasamos a razonar la señal  $Q2$ . El biestable asociado se carga siempre con el valor de la señal  $X$ . Por lo tanto, la salida  $S$ , que es la señal  $Q2$ , pasará a ser la señal  $X$  desplazada un ciclo de reloj.

El cronograma completo es el siguiente:

## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	20/06/2018	12:00



### PROBLEMA 4 [10%]

a) [5%] ¿Cuál es el primer paso en el ciclo de ejecución de instrucciones?

Lectura de la instrucción a ejecutar.

b) [5%] ¿Para qué se usan los módulos de acceso directo a memoria (DMA)?

Para liberar a las CPU de las transferencias de información entre periféricos y memoria principal.