

Examen 2022/23-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	21/1/2023	12:30

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 2022/23-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	21/1/2023	12:30

Enunciados

PROBLEMA 1 [20 %]

- a) [2.5 %] Dado el valor $A = 10101010$, que representa un número binario entero codificado en formato de signo y magnitud con 8 bits, ¿qué número decimal representa?

Para calcular la representación decimal de un número entero codificado en signo y magnitud, aplicamos el TFN (multiplicar cada dígito por el peso de su posición y hacer la suma) a la magnitud (los 7 bits de menor peso).

Así pues, obtenemos el resultado siguiente: $0101010_{(2)} = 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^1 = 42_{(10)}$

Finalmente, consideramos el signo del número, el cual está indicado en el bit de más peso. Como es 1, el signo es negativo.

Por lo tanto: $10101010_{(SM2)} = -42_{(10)}$

- b) [2.5 %] ¿Qué número representaría en decimal, si el mismo valor A estuviera codificado en complemento a 2 con 8 bits?

Para calcular la representación decimal de un número entero codificado en Complemento a 2, podemos aplicar el TFN pero considerando que el bit de mayor peso puede indicar que sea negativo. Así pues, al ser el primer bit 1 se trata de un número negativo. Por lo tanto, aplicaremos el TFN considerando el bit de más peso negativo y el resto como positivos. En este caso tendríamos:

$10101010_{(Ca2)} = -1 \cdot 2^7 + 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^2 = -128 + 32 + 8 + 4 = -86_{(10)}$

- c) [10 %] Dados los valores $C = 10101101$ y $D = 01100111$, que representan dos números binarios enteros expresados en complemento a 2 con 8 bits, calcula $C - D$ utilizando el mismo formato. ¿Se produce desbordamiento? Justifica la respuesta.

Para restar dos números en formato de Ca2 convertimos la operación $C - D$ en $C + (-D)$, cambiando el signo del sustraendo.

Para cambiar de signo un número representado en Ca2 complementamos todos sus bits y sumamos 1 al resultado. Por lo tanto, para cambiar de signo el valor D :

$$\begin{array}{r}
 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \quad \leftarrow D \\
 \downarrow \\
 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \quad \leftarrow D' \\
 + \quad \quad \quad 1 \\
 \hline
 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \quad \leftarrow -D
 \end{array}$$

A continuación, hacemos la operación de suma de los dos números:

$$\begin{array}{r}
 1 \quad 1 \ 1 \ 1 \quad 1 \quad \leftarrow \text{acarreo} \\
 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \quad \leftarrow C \\
 + \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \quad \leftarrow -D \\
 \hline
 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0
 \end{array}$$

Examen 2022/23-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	21/1/2023	12:30

Sumando en Ca2, el acarreo de la última etapa se desprecia. Hay desbordamiento porque en la suma de dos números negativos codificados en Ca2 el resultado es positivo.

d) [5 %] Dado el formato de coma flotante siguiente:

S	Exponente				Mantisa		
15	14	...	10	9	...	0	0


Donde:

- El bit de signo, S, vale 0 para cantidades positivas y 1 para negativas.
- El exponente se representa en exceso a 16.
- Hay bit implícito.
- La mantisa está normalizada en la forma 1,X.


Representad el número $-24,125_{(10)}$ en este formato.

Según el formato dado, analizamos primero el número separando la parte entera y la parte fraccionaria:

1. Pasamos la parte entera a binario, aplicando el método de la división entera:

$$\begin{array}{rclcl}
 24 & = & 12 \cdot 2 & + & 0 \\
 12 & = & 6 \cdot 2 & + & 0 \\
 6 & = & 3 \cdot 2 & + & 0 \\
 3 & = & 1 \cdot 2 & + & 1 \\
 1 & = & 0 \cdot 2 & + & 1 \\
 \\
 24_{(10)} & = & 11000_{(2)}
 \end{array}$$


2. Para la parte fraccionaria aplicamos el método correspondiente:

$$\begin{array}{rclcl}
 0,125 & \cdot 2 = & 0,25 & + & 0 \\
 0,25 & \cdot 2 = & 0,5 & + & 0 \\
 0,5 & \cdot 2 = & 0,0 & + & 1
 \end{array}$$


$$0,125_{(10)} = 0,001_{(2)}$$

Juntamos la parte entera y fraccionaria: $11000,001_{(2)}$

Para normalizar la mantisa hay que mover la coma 4 posiciones hacia la izquierda:

$$11000,001_{(2)} = 1,1000001_{(2)} \cdot 2^4$$

Finalmente, identificamos cada campo:

- Signo: Negativo, S=1.

Examen 2022/23-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	21/1/2023	12:30

- Exponente: 4. Hay que representarlo en exceso a 16. Por lo tanto, tenemos que sumarle el exceso, $16 + 4 = 20$, que en base 2 es 10100_2 .
- Mantisa: $1,1000001_2$. Como tenemos que representar el número con bit implícito eliminamos el 1 de la parte entera. Así pues, la mantisa será 1000001 . Como la mantisa tiene disponibles 10 posiciones, añadimos tres ceros por la derecha hasta completar el formato.

El número en el formato solicitado es:

1	10100	1000001000
---	-------	------------

PROBLEMA 2 [35 %]

a) [20 %] Dada la tabla de verdad siguiente:

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

- i. [10 %] Sintetizad la función f de manera mínima a dos niveles mediante el método de Karnaugh y dibujad el circuito resultante.

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

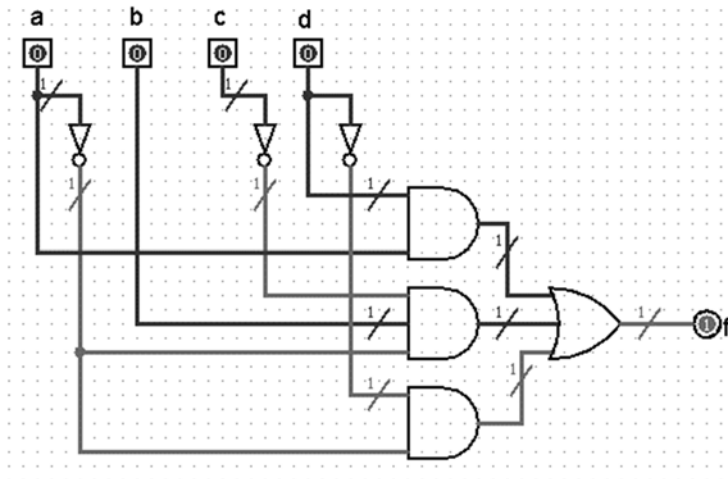
Examen 2022/23-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	21/1/2023	12:30

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

Y obtenemos esta expresión mínima y el correspondiente circuito:

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



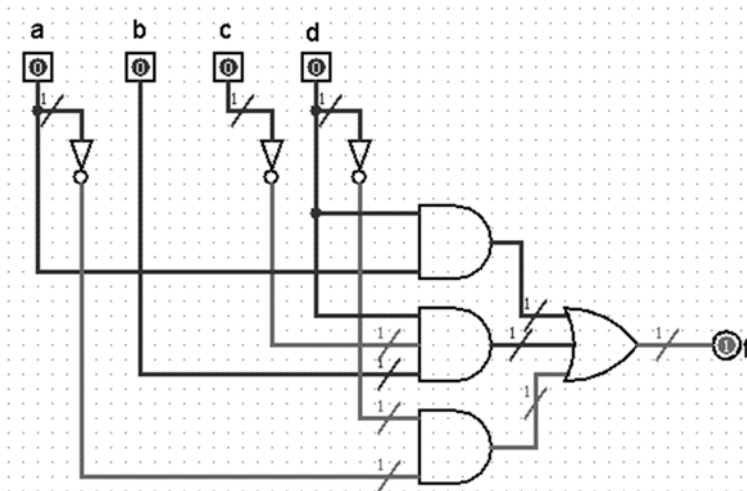
Habría una segunda solución de minimización de la función f .

Examen 2022/23-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	21/1/2023	12:30

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

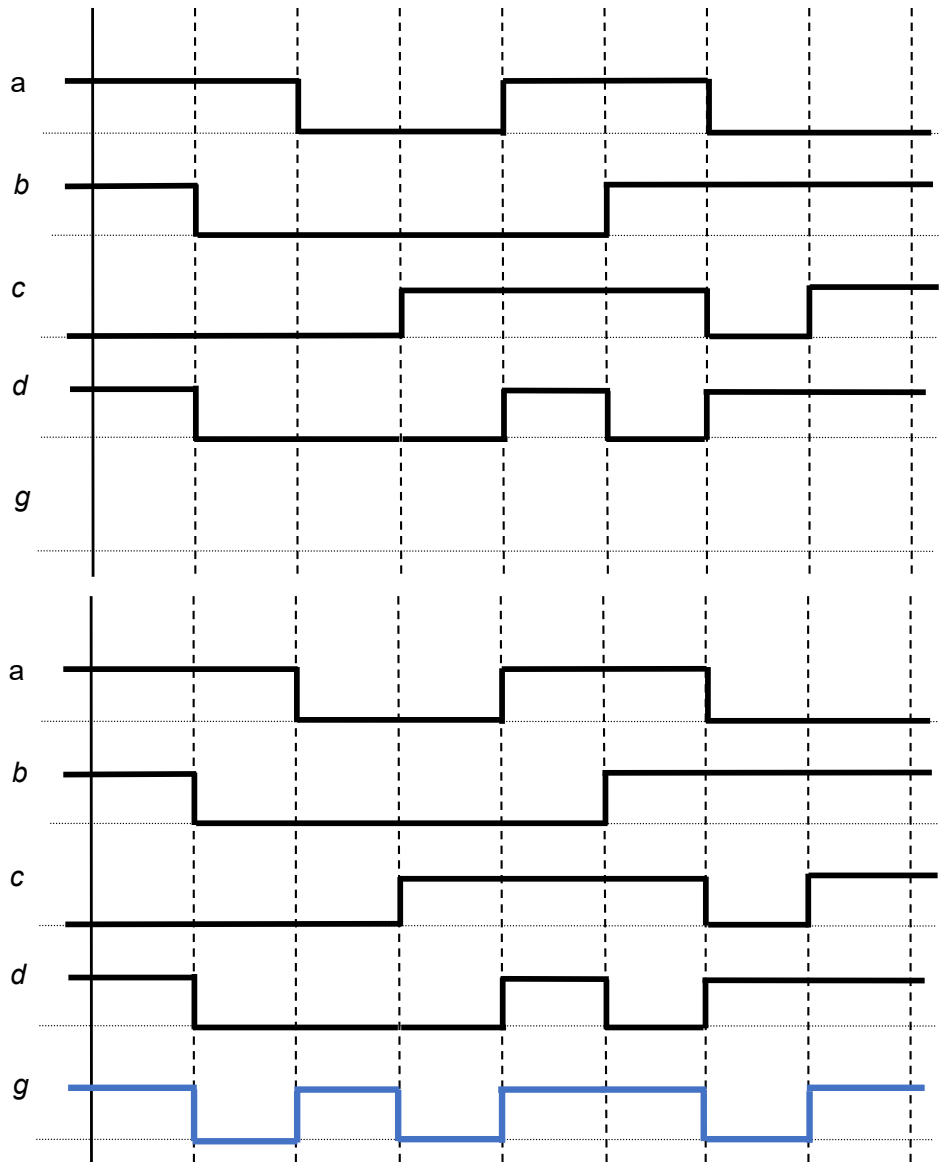
$$f = a \cdot d + a' d' + b \cdot c' d$$



Examen 2022/23-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	21/1/2023	12:30

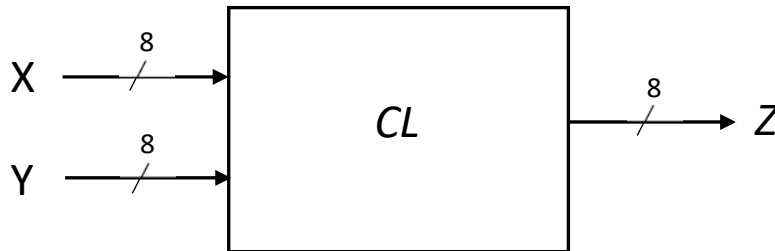
- ii. **[10 %]** Completad el siguiente cronograma para la función g . No hace falta que justifiquéis el resultado.



Examen 2022/23-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	21/1/2023	12:30

b) [15 %] Diseñad un circuito lógico combinacional, denominado *CL*, con la estructura siguiente:



Las entradas *X* e *Y* representan números enteros de 8 bits, codificados en Ca2. La salida *Z*, también en Ca2 de 8 bits, debe obtener el número más grande entre *X* e *Y*.

Se pide que diseñéis el circuito *CL* usando bloques y puertas combinacionales, y especificando claramente la dimensión de los buses utilizados.

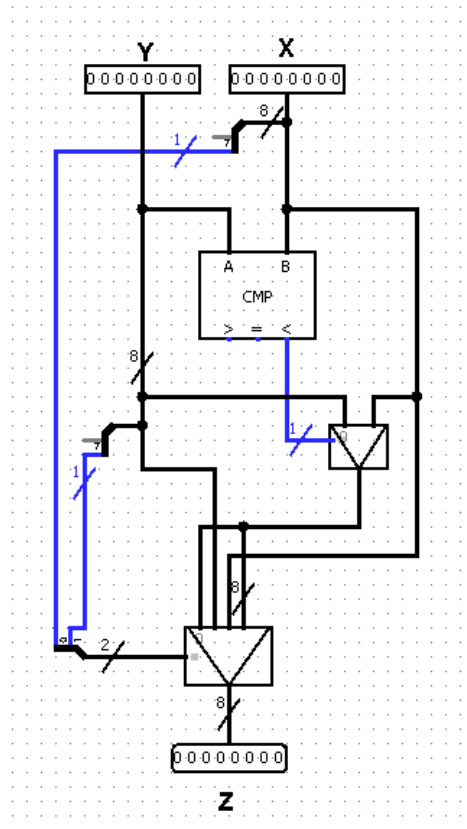
Para saber cuál de los dos números es mayor, consideraremos sus signos (el signo en Ca2 viene dado por el bit más significativo). Utilizaremos un multiplexor 4-1 donde las señales de control sean los bits de signo de los números *X* e *Y* (bits x_7 e y_7 , respectivamente). De este modo, hay cuatro posibilidades a tener en cuenta:

- $x_7 = y_7 = 0$: Los dos números son positivos. Para saber cuál de los dos es mayor comparamos sus magnitudes. Por este motivo se incluye un comparador para determinar cuál de los dos es mayor y es éste el que se propaga en la entrada con menor peso del multiplexor 4:1.
- $x_7 = 1, y_7 = 0$: El número *Y* es positivo y *X* es negativo. Por lo tanto, *Y* es mayor y lo propagamos en la salida.
- $x_7 = 0, y_7 = 1$: El número *X* es positivo e *Y* es negativo. Por lo tanto, *X* es mayor y lo propagamos en la salida.
- $x_7 = y_7 = 1$: Los dos números son negativos. Este caso es similar al primer caso cuando los dos son positivos. Para saber cuál de los dos es mayor aprovechando el mismo comparador ya que el orden de los números negativos en Ca2 es la misma que la representación de los números positivos en binario. El número mayor es el que se propaga en la entrada de más peso del multiplexor 4-1.

Hay que tener en cuenta que existen alternativas similares que también tendrán la funcionalidad pedida, y por tanto, serán válidas.

Examen 2022/23-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	21/1/2023	12:30



Examen 2022/23-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	21/1/2023	12:30

PROBLEMA 3 [35 %]

- a) [17,5 %] Dibujad el grafo de estados de un circuito secuencial, con dos entradas, a y b , de un bit cada una, y una salida S de dos bits. El circuito leerá en secuencia, por cada entrada, un número entero de dos bits codificado en signo y magnitud (en el primer ciclo leerá los bits de mayor peso y en el segundo ciclo los de menor peso).

Inicialmente la salida tendrá el valor 00, y mantendrá ese valor mientras no se hayan leído los dos números A y B completos. Después, la salida tiene que tomar en el siguiente ciclo uno de estos valores:

- $S = 10$, si $A > B$, o bien
- $S = 01$, si $B > A$, o bien
- $S = 11$, si $A = B$.

Recordad que en signo y magnitud las codificaciones 00 y 10 representan el mismo número.

Para simplificar el número de aristas del grafo, podéis asumir que el circuito hará una única comparación de los dos números, es decir: una vez leídos y comparados los números A y B , el circuito se parará.

Como los números A y B están codificados en signo y magnitud, el bit de más peso es el bit de signo. Para saber qué número es mayor tendremos que recordar el signo de cada uno. Por lo tanto, tenemos que crear 4 estados, uno para cada posible combinación de signos, a los que se llegará desde el estado inicial después de leer los bits de signo de ambos números

A continuación, llegarán los bits correspondientes a las magnitudes de A y B , que añadidas a la información que nos han dado los signos ya nos permitirán decidir cuál deberá ser el valor de las salidas, de entre los tres posibles. Cada uno se dará en un estado distinto, claro está.

Por lo tanto, los estados serán los siguientes:

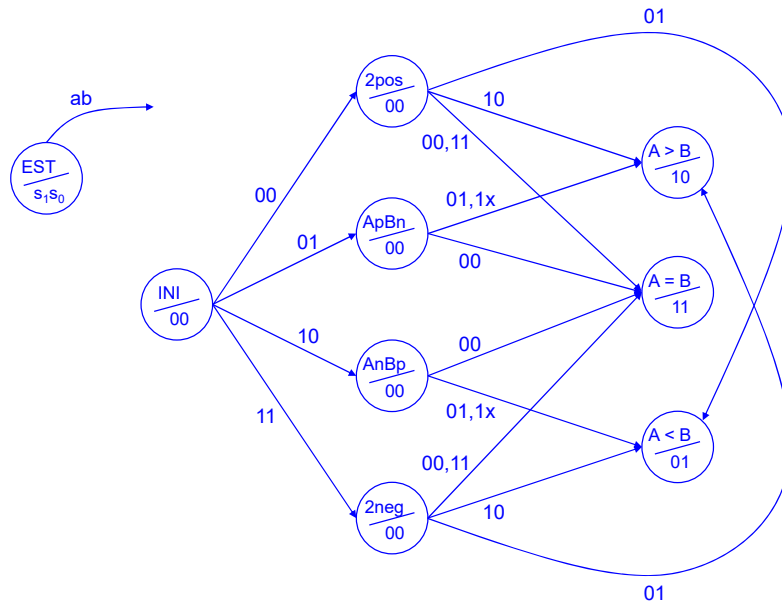
INI	Estado inicial
2pos	Los dos números A y B son positivos
ApBn	A es positivo y B negativo
AnBp	A es negativo y B positivo
2neg	Los dos son negativos
A=B	Los dos números son iguales
A>B	A es mayor que B
A<B	A es menor que B

Las transiciones hacia los estados finales se derivan inmediatamente de la comparación entre dos números, recordando que los valores que se pueden codificar en signo y magnitud y 2 bits son $-1 < 0 < 1$, y que el 0 tiene dos codificaciones (00 y 10).

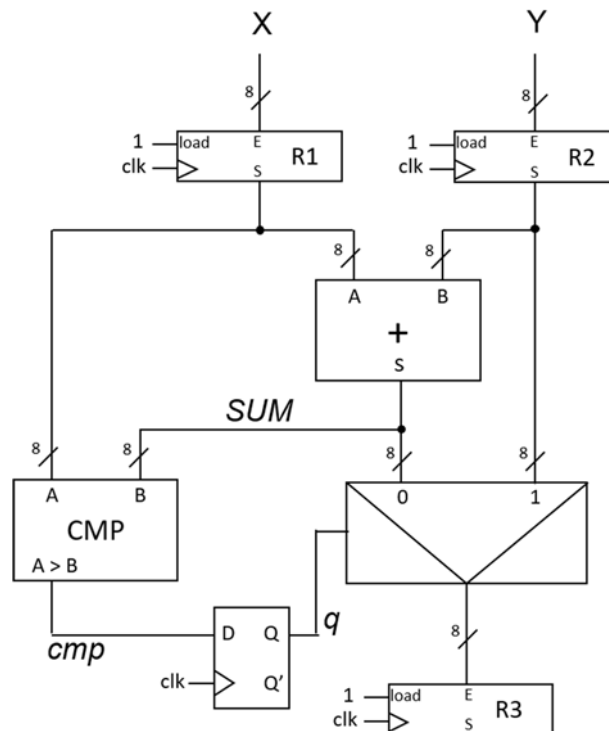
A continuación, se muestra el grafo completo. Recordemos que el enunciado indica que mientras no se hayan leído ambos bits de cada número, la salida se debe mantener a 00.

Examen 2022/23-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	21/1/2023	12:30



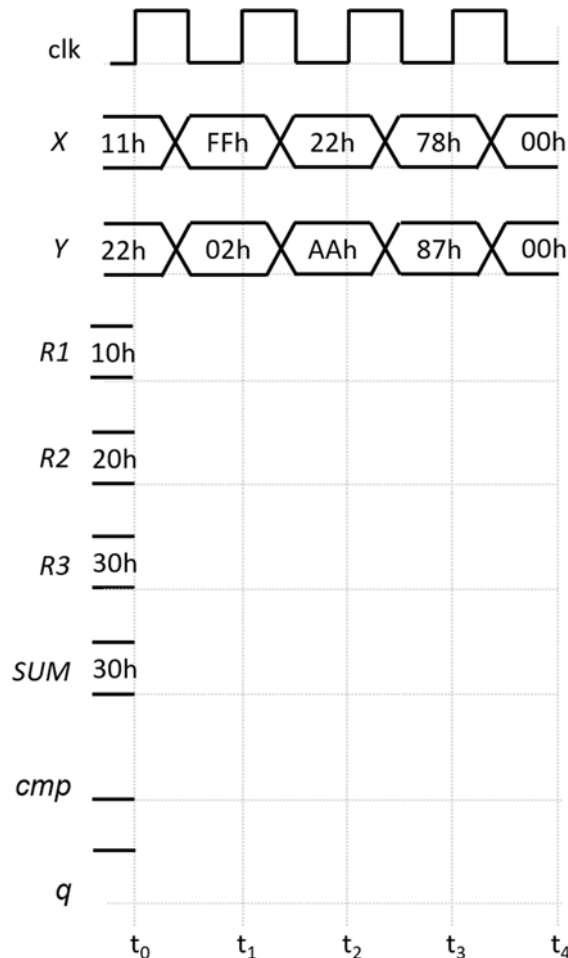
b) [17,5 %] Dado el circuito siguiente:



Examen 2022/23-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	21/1/2023	12:30

Completad el cronograma siguiente expresando los valores de los registros y de *SUM* en hexadecimal, e incluyendo las señales *cmp* y *q*. No hace falta que justificuéis el resultado.



Para rellenar el cronograma miramos en qué momentos se cargan los registros y con qué valores. En cuanto a los momentos, vemos que se cargan en cada flanco de reloj, ya que a la entrada load de los tres registros está conectado un 1. El biestable también se cargará en cada flanco, porque no se ha dibujado su entrada load, lo cual indica que siempre le llega 1.

En cuanto a los valores que se cargan, vemos que en la entrada de R1 y de R2 llegan siempre las señales X e Y respectivamente. Por lo tanto, podemos escribir ya de entrada todos sus valores en el cronograma (en cada flanco se escribe el valor de X o de Y en el instante anterior al flanco).

Como $SUM = R1 + R2$, también podemos escribir todos sus valores en el cronograma una vez hemos puesto los de R1 y R2. También podemos dibujar los valores de *cmp*, que sólo depende de los valores de R1 y de SUM en cada momento (*cmp* vale 1 si $R1 > SUM$ y 0 en caso contrario). Por ejemplo, antes de t_0 *cmp* vale 0, en el ciclo que va de t_0 a t_1 también ($11h < 33h$), después de t_1 *cmp* vale 1 ($FFh > 01h$), y así sucesivamente.

Examen 2022/23-1

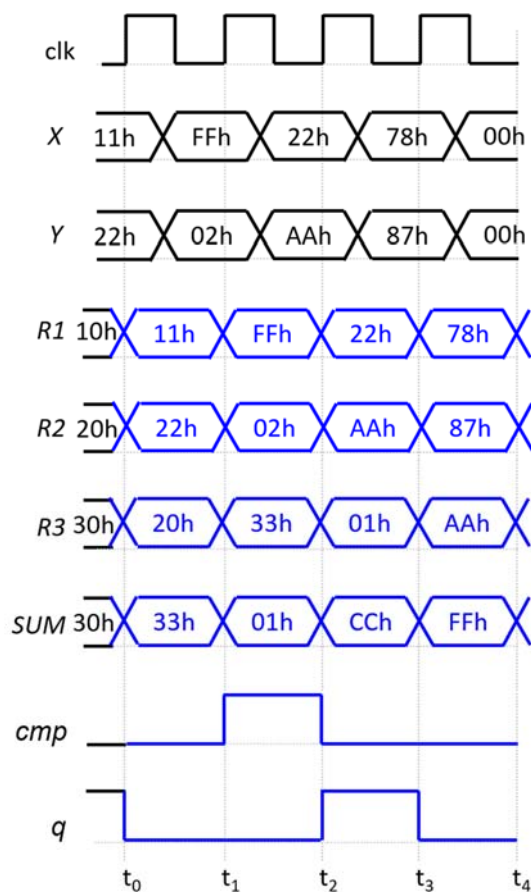
Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	21/1/2023	12:30

A continuación, podemos completar también la línea correspondiente a q , que en cada flanco tomará el valor de cmp . Es decir, tendrá los mismos valores que cmp pero desplazados un ciclo.

Ahora solo nos falta completar la línea del cronograma correspondiente a $R3$. El valor que se cargará en cada flanco depende de q : si al llegar el flanco $q=0$ tomará el valor de SUM , mientras que si $q=1$ tomará el de $R2$.

Por ejemplo, al llegar al instante t_0 $q=1$, de forma que en este flanco lo que se escribe en $R3$ es $R2$, que antes de t_0 vale $20h$. Al llegar al instante t_1 $q=0$, de forma que en este flanco se escribe SUM , que antes de t_1 vale $33h$.

Razonando de manera análoga completamos toda la línea correspondiente a $R3$.



PROBLEMA 4 [10%]

- a) [5%] ¿Qué característica es más exclusiva de las arquitecturas de conjuntos de instrucciones (ISA) reducidos (RISC) respecto de los complejos (CISC)?

Tener dos tipos de instrucciones para acceder a datos en memoria: uno para lectura y otro para escritura.

Examen 2022/23-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	21/1/2023	12:30

b) **[5%]** ¿Para qué se usa la memoria caché?

Para proporcionar a la CPU un acceso más rápido a la información de la memoria principal.