

Examen 2023/24-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	29/6/2024	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%**
 - ¿Se puede consultar material durante la prueba? **NO** ¿Qué materiales están permitidos?
Cap
 - ¿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 2023/24-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	29/6/2024	12:30

Solución

PROBLEMA 1 [20%]

Contestad los apartados siguientes:

- a) **[12,5 %]** Dada la secuencia de bits 01011100, a qué número decimal equivale según los tipos de interpretaciones siguientes:

- i. **[5 %]** Si se trata de un número binario natural.

Para pasar de una base b en una base b' , usamos el método basado en TFN. Por lo tanto, para pasar de base 2 a base 10:

$$01011100_2 = 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 64 + 16 + 8 + 4 = 92_{10}$$

- ii. **[7,5 %]** Si se trata de un número binario codificado en el formato de coma flotante siguiente:

S		Exponente			Mantisa	
7	6		4	3		0

en el cual:

- El bit de signo, S, vale 0 en los números positivos y 1 en los negativos.
- El exponente se representa en exceso a 4.
- Hay bit implícito.
- La mantisa está normalizada en la forma 1,X.

Bit de signo (0): Vale 0, por lo tanto es un número positivo.

Bits del exponente (101): El exponente toma el valor $101_2 - 4_{10} = 5_{10} - 4_{10} = 1_{10}$

Bits de la mantisa (1100): Cómo es con bit implícito, la mantisa es $1,1100_2$

El número es $+1,1100_2 \cdot 2^1_{10} = 11,1_2 = 3,5_{10} \%$

- b) **[7,5 %]** Dados los números $A = 01011010$ y $B = 10110111$, que son enteros codificados en signo y magnitud de 8 bits, realizad la resta $A - B$ trabajando en la misma representación e indicad si se produce desbordamiento.

En este caso 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.

Sumamos las magnitudes de los operandos:

	1	1	1	1	1	1		← acarreo
		1	0	1	1	0	1	0
+	0	1	1	0	1	1	1	
	1	0	0	1	0	0	0	1

El resultado es: 10010001_{SM2}

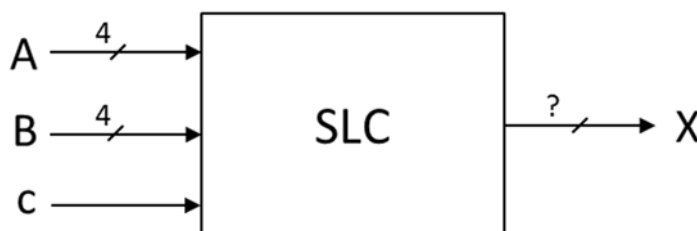
Examen 2023/24-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	29/6/2024	12:30

Hay desbordamiento, ya que hay acarreo en la última etapa de la suma de las magnitudes.

PROBLEMA 2 [35 %]

a) [25 %] Un sistema combinacional SLC tiene las entradas y salidas siguientes:



dónde A y B son números enteros de 4 bits codificados en complemento a 2. La salida X también es un entero en Ca_2 , donde su valor viene determinado por la tabla siguiente:

c	X
0	$(A-B) \cdot 6$
1	$2 \cdot A + 4 \cdot B$

i. [5 %] ¿Cuántos bits debe tener como mínimo la salida X para que no se pueda producir nunca desbordamiento?

El rango de representación de un número entero en Ca_2 es $[-2^{n-1}, 2^{n-1}-1]$. En este caso, con números codificados con 4 bits, el rango resultante es $[-8, 7]$.

Con la primera operación, el resultado estará entre $[(-8-7) \cdot 6, (7-0) \cdot 6] = [-90, 42]$, necesitando 8 bits para representar estos valores.

En el caso de la segunda operación, el rango de los resultados será $[-8 \cdot 2 + -8 \cdot 4, 7 \cdot 2 + 7 \cdot 4] = [-48, 42]$, que son representables con un mínimo de 7 bits.

Por eso, la salida debe tener **8 bits** para que no se pueda producir desbordamiento.

ii. [20 %] Implementad el circuito SLC sabiendo que podéis disponer de todas las puertas y bloques que se han explicado en el temario, excepto memorias ROM.

La salida del circuito está determinada por el valor de c . El circuito estará formado por dos partes que implementarán cada una de las dos funciones y un multiplexor seleccionará el valor correspondiente de estas dos partes en función del valor de c .

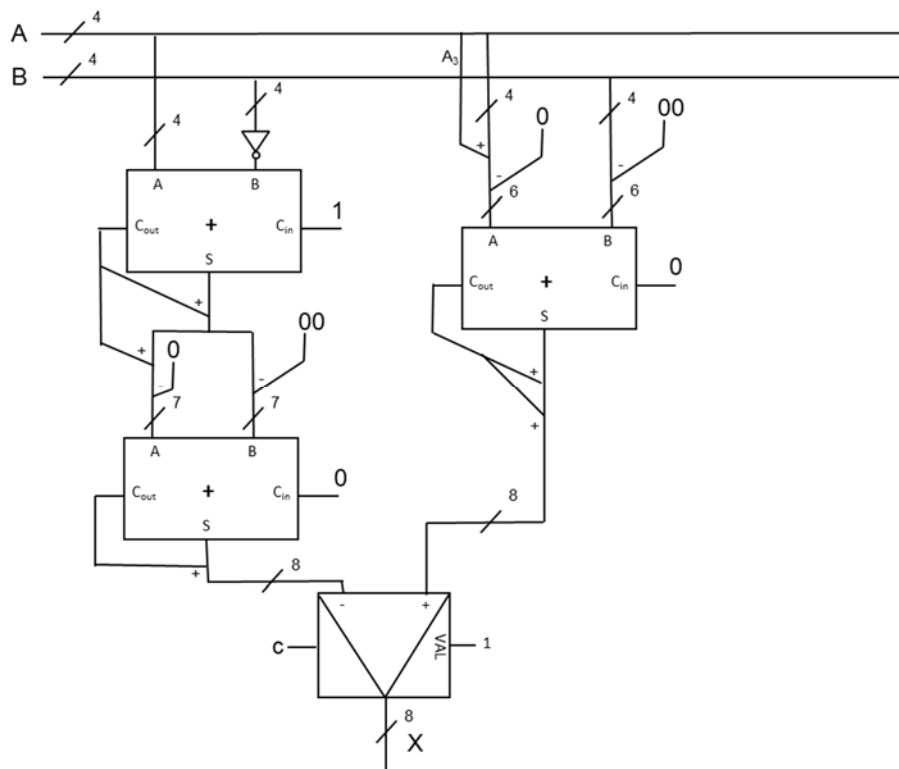
Para la primera parte, $c=0$, se debe hacer la resta y multiplicar el resultado por 6. La resta se puede implementar con un sumador cuyo segundo operando se cambia de signo ($A-B = A+(-B)$). El cambio de signo en Ca_2 se hace negando todos los bits y sumando el valor 1. Por eso, el sumador de la primera parte del circuito tiene puertas de negación para los bits del segundo operando y el valor de c_{in} activado. Para garantizar que no hay desbordamiento, se debe incrementar en un bit la precisión del resultado, incorporando el valor de c_{out} como bit más significativo de la salida.

Examen 2023/24-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	29/6/2024	12:30

Una vez que hemos calculado $A-B$, tenemos que multiplicar este valor por 6. Hacer una multiplicación por 6 es lo mismo que multiplicar por 2, multiplicar por 4 y sumar ($6 \cdot X = 2 \cdot X + 4 \cdot X$). Las multiplicaciones por valores potencias de 2 se pueden hacer con desplazamientos a la izquierda. Multiplicar por 2 es hacer un desplazamiento a la izquierda de un bit, y multiplicar por 4 es hacer un desplazamiento a la izquierda de dos bits. Fijaros que se tiene que incrementar en un bit la precisión del operador correspondiente a la multiplicación por 2 para que ambos operandos sean de 7 bits. Finalmente, estos dos valores son las entradas del sumador que implementa $2 \cdot X + 4 \cdot X$. Para evitar desbordamiento, añadiremos C_{out} con bit más significativo de la suma.

En la segunda parte del circuito, $c=1$, también hacemos las multiplicaciones con desplazamientos a la izquierda de los operandos. En el caso de A , hacemos un desplazamiento de 1 bit a la izquierda, y en el caso de B , un desplazamiento de 2 bits a la izquierda. Con estos desplazamientos A tendrá una precisión de 5 bits y B de 6 bits. Para poder sumarlos tenemos que hacer la extensión de signo de un bit al primero de ellos. Finalmente, para evitar desbordamiento en la salida del sumador, tenemos que incrementar la precisión del resultado hasta 8 bits incorporando el valor de C_{out} dos veces como bit más significativo del resultado.



Examen 2023/24-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	29/6/2024	12:30

b) [10 %] Minimiza la función siguiente por el método de Karnaugh y haz la síntesis del circuito a dos niveles:

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

Esta función lógica tiene una única solución que se muestra en el siguiente mapa de Karnaugh:

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

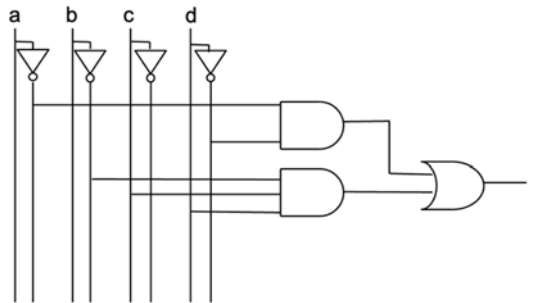
Dónde la expresión mínima que tiene este mapa de Karnaugh es:

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

El circuito a dos niveles que implementa esta función con puertas lógicas es el siguiente:

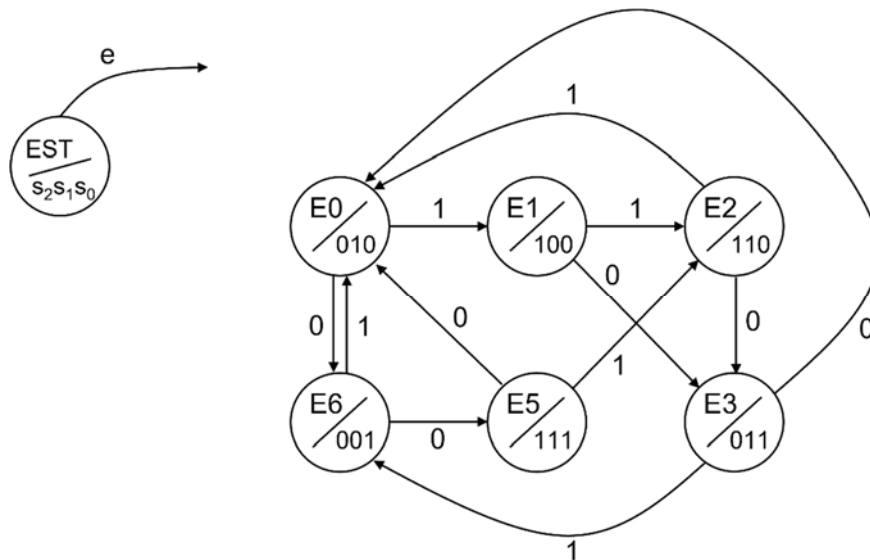
Examen 2023/24-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	29/6/2024	12:30



PROBLEMA 3 [35 %]

a) [17,5 %] Dado el grafo de estados siguiente:



Se pide:

- i) [10 %] Escribid la tabla de transiciones y la tabla de salidas del circuito descrito por el grafo, codificando los estados según su índice asociado. Al escribir la tabla de transiciones, poned en primer lugar las variables que codifican el estado, en el orden de pesos habitual, y a continuación las variables de entrada.

Puesto que el grafo tiene 6 estados, harán falta 3 bits para codificarlos ($\log_2(6)$ es un número real entre 2 i 3). Así, el circuito deberá tener 3 biestables, uno por cada bit de codificación de los estados. Les llamaremos q_2 , q_1 y q_0 .

Puesto que nos piden codificar los estados según el índice asociado a cada uno, la codificación será la siguiente:

Examen 2023/24-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	29/6/2024	12:30

Estado	q ₂	q ₁	q ₀
E0	0	0	0
E1	0	0	1
E2	0	1	0
E3	0	1	1
E5	1	0	1
E6	1	1	0

Para rellenar la tabla de transiciones, observamos en cada estado del grafo a qué estados vamos con cada valor de la variable de entrada, y escribimos la codificación correspondiente en la tabla. Las filas correspondientes a codificaciones de estados con un subíndice que no está en el grafo corresponden a combinaciones *don't care*.

q ₂	q ₁	q ₀	e	q ₂ ⁺	q ₁ ⁺	q ₀ ⁺
0	0	0	0	1	1	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	1	1	0
1	0	0	0	x	x	x
1	0	0	1	x	x	x
1	0	1	0	0	0	0
1	0	1	1	0	1	0
1	1	0	0	1	0	1
1	1	0	1	0	0	0
1	1	1	0	x	x	x
1	1	1	1	x	x	x

Para rellenar la tabla de salidas, observamos en el grafo el valor que toman las salidas en cada estado y lo ponemos en la tabla. Al igual que en la tabla de transiciones, las codificaciones de los subíndices 4 y 7 corresponden a condiciones *don't care*.

Examen 2023/24-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	29/6/2024	12:30

q ₂	q ₁	q ₀	s ₂	s ₁	s ₀
0	0	0	0	1	0
0	0	1	1	0	0
0	1	0	1	1	0
0	1	1	0	1	1
1	0	0	x	x	x
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	x	x	x

- ii) **[7,5 %]** Suponiendo que implementamos el circuito descrito por el grafo usando una memoria ROM, especificad en hexadecimal en la tabla siguiente el contenido de las palabras que se indican de la memoria ROM.

Dirección	Contenido
02h	
03h	
04h	
05h	

Al implementar un circuito secuencial usando una memoria ROM, al bus de direcciones de la ROM se conectan las señales que codifican el estado en los bits de más peso, y en los bits de menos peso las señales de entrada. En nuestro caso, por lo tanto, se conectan las señales q_2 , q_1 , q_0 y e .

Por otro lado, las palabras de la ROM tienen que indicar el estado futuro (q_2^+ , q_1^+ i q_0^+) y el valor de las 3 señales de salida (s_2 , s_1 i s_0) en el estado actual.

Por lo tanto, la dirección 02h corresponde a la combinación [q_2 q_1 q_0 e] = [0010], y el contenido de esta palabra de la ROM tendrá que estar formado por la codificación del estado al cual se va desde el estado E1 cuando la entrada e vale 0 (es decir, la codificación del estado E3, que es [011]), y el valor de las salidas en el estado E1 (es decir, [100]).

Concatenando todos estos bits obtenemos 011100, que en hexadecimal es 1Ch. A la práctica, una forma fácil de obtener los 3 primeros bits es mirando la fila correspondiente a la combinación [0010] de la tabla de transiciones, y para obtener los 3 últimos bits podemos mirar la fila [001] de la tabla de salidas.

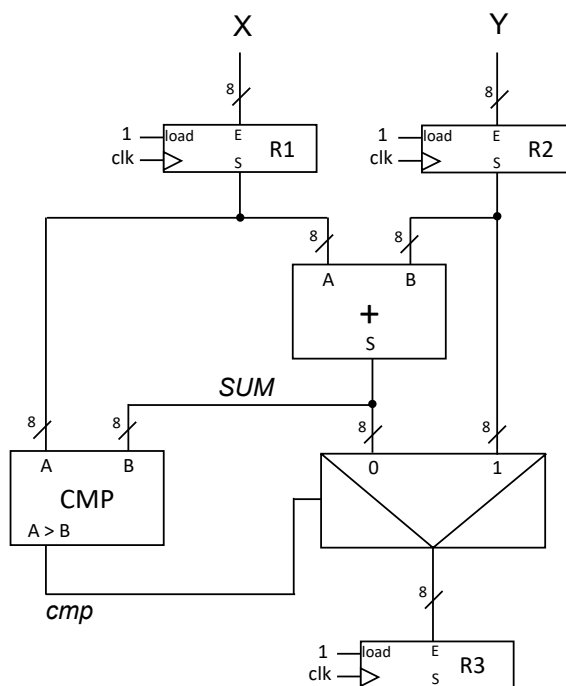
Razonando de forma análoga obtenemos el contenido del resto de palabras que se nos piden.

Examen 2023/24-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	29/6/2024	12:30

Dirección	Contenido
02h	1Ch
03h	14h
04h	1Eh
05h	06h

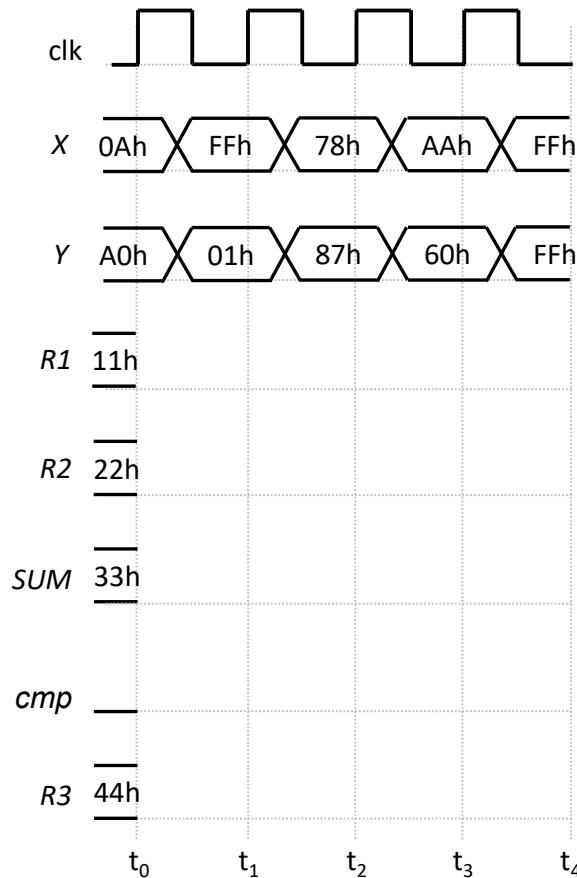
b) [17,5 %] Dado el circuito siguiente:



Completad el cronograma siguiente expresando los valores de los registros R1 y R2 en hexadecimal.

Examen 2023/24-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	29/6/2024	12:30



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, porque en la entrada *load* de los tres registros hay conectado un 1.

En cuanto a los valores que se cargan, vemos que a la entrada de *R1* y de *R2* llegan 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 justo anterior al flanco).

A la entrada de *R3* llega un valor que depende de *R1*, *R2*, *SUM* y *cmp*, de la forma siguiente: si *cmp* = 0 en *R3* se escribirá *SUM*, y si *cmp* = 1 se escribirá *R2*. Analicemos cómo evolucionan estas dos señales a lo largo del tiempo.

La señal *SUM* es la suma de *R1* y *R2*. Puesto que ya hemos escrito en el cronograma todos los valores de estos dos registros, podemos escribir también todos los valores de *SUM*.

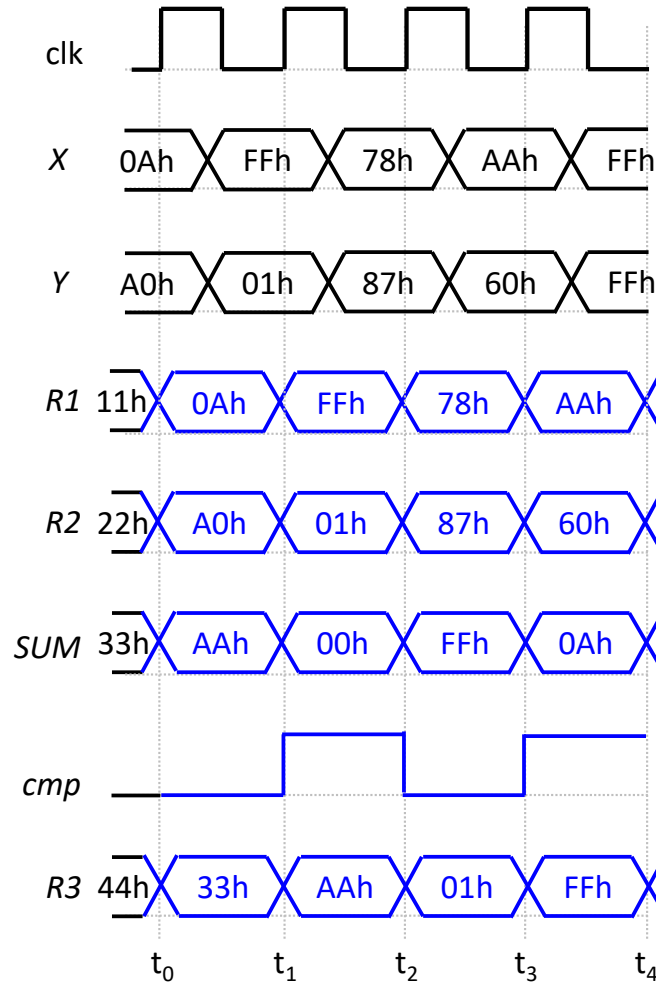
La señal *cmp* corresponde a la salida *A > B* de un comparador al que llegan *R1* y *SUM* por las entradas *A* y *B* respectivamente. Por lo tanto, *cmp* = 1 siempre que *R1* > *SUM*, y *cmp* = 0 en caso contrario. Puesto que ya tenemos escritos en el cronograma todos los valores de *R1* y de *SUM*, podemos escribir ya también todos los valores de *cmp*.

Nos falta solamente poner los valores de *R3*, cosa que haremos a partir de la descripción que hemos obtenido de los valores que se escriben en *R3*, observando cuánto valen todas las señales involucradas en el instante anterior a cada flanco.

El cronograma completo se muestra a continuación.

Examen 2023/24-2

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	29/6/2024	12:30



PROBLEMA 4 [10 %]

a) [5 %] ¿Cuál es la función de los buses en un computador?

Conectar dos o más componentes entre ellos.

b) [5 %] ¿Qué tipo de procesador permitiría ejecutar varios programas simultáneamente?

Uno con varias unidades de procesamiento o *cores*.