grapa

Examen 2010/11-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	29/01/2010	12:00

75.562 29 01 10 EX

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

Ficha técnica del examen

- **Comprueba que el código y el nombre de la asignatura corresponden a la asignatura de la cual estás matriculado.**
- **Debes pegar una sola etiqueta de estudiante en el espacio de esta hoja destinado a ello.**
- **No se puede añadir hojas adicionales.**
- **No se puede realizar las pruebas a lápiz o rotulador.**
- **Tiempo total** 2 **horas**
- **En el caso de que los estudiantes puedan consultar algún material durante el examen, ¿cuál o cuáles pueden consultar?:** No se puede consultar ningún material
- **Valor de cada pregunta:** Prob 1: 30%, Prob. 2: 35%, Prob. 3: 35%
- **En el caso de que haya preguntas tipo test: ¿descuentan las respuestas erróneas?** NO **¿Cuánto?**
- **Indicaciones específicas para la realización de este examen**
- No se puede utilizar ningún tipo de calculadora.
  Razonad las respuestas en cada ejercicio. Las respuestas sin justificar no recibirán puntuación.
- **Este enunciado corresponde también a los siguientes códigos de asignatura:** 81.518.

Enunciados

(En siguientes páginas)

Examen 2010/11-1

Asignatura	**Código**	**Fecha**	**Hora inicio**
Fundamentos de computadores	75.562	29/01/2010	12:00

PROBLEMA 1 [30%]

Dada la cadena de bits 001100110011 que representa un número fraccionario codificado en coma fija y signo y magnitud, en un formato con 4 bits fraccionarios.

a) **[5%]** ¿Qué valor decimal representa?

Primero separamos los bits de signo, de magnitud entera y de magnitud fraccionaria:

Signo: 0, por lo tanto signo positivo.
Magnitud entera: 0110011
Magnitud fraccionaria: 0011

A continuación calculamos el valor decimal de la magnitud aplicando el TFN:

Parte entera: $0110011_{(2} = 0\cdot2^6 + 1.2^5 + 1\cdot2^4 + 0\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\ 2^0 = 32 + 16 + 2 + 1 = 51_{(10}$
Parte fraccionaria: $0{,}0011_{(2} = 0\cdot2^{-1} + 0\cdot2^{-2} + 1\cdot2^{-3} + 1\cdot2^{-4} = 0{,}125 + 0{,}0625 = 0{,}1875_{(10}$

Por lo tanto, el valor decimal que representa es: $+51{,}1875_{(10}$

b) **[10%]** Representad este mismo número en el formato de coma flotante siguiente:

S	Exponente		Mantisa	
11	10	7	6	0

Donde:

- El bit de signo, *S*, vale 0 para cantidades positivas y 1 para las negativas.
- El exponente se representa en exceso a 8.
- La mantisa está normalizada en la forma 1,*X* con bit implícito.

Para representar el número en formato de coma flotante a partir de un número codificado en coma fija y signo y magnitud, el primero que hacemos es eliminar la información del signo (eliminar el bit de más peso) y normalizar la magnitud:

$$00110011{,}0011 = +0110011{,}0011_{(2} = +1{,}100110011_{(2} \cdot 2^5$$

A continuación, identificamos cada campo:

- Signo: 0
- Exponente: 5. Como hay que representarlo en exceso a 8, tenemos que sumarle 8. Por lo tanto, 5 + 8 = 13, que en base 2 es $1101_{(2}$.
- Mantisa: Como la mantisa es con bit implícito y tiene que tener 7 bits, cogemos los 7 bits de más peso de la parte fraccionaria normalizada y eliminamos el resto. Así pues, la mantisa será: 1001100.

El número en el formato solicitado es:

0	1101	1001100

c) **[5%]** ¿Se produce error de representación? En caso afirmativo, calculadlo y expresadlo **en decimal**.

Sí, ya que al representar la mantisa hemos eliminado bits de la parte fraccionaria.

El error producido es:

$$0{,}000000011_{(2} \cdot 2^{5} = 11_{(2} \cdot 2^{-9} \cdot 2^{5} = 3_{(10} \cdot 2^{-4} = 0{,}1875_{(10}.$$

d) **[10%]** Dados los números binarios *A* = 01011010 y *B* = 10111110, que representan números enteros codificados en signo y magnitud, realizad la resta *A* - *B* con el mismo número de bits. ¿Se produce desbordamiento? Si no se produce, indicad el resultado de la suma **en decimal**.

- Como el segundo operando es negativo, la operación de resta se convierte en una operación de suma donde el segundo operando se cambia de signo. En la representación con signo y magnitud, cambiar el signo sólo significa cambiar el bit de más peso, así que *B* cambiado de signo es 00111110. Por lo tanto, tenemos que sumar dos números positivos.

  Para sumar dos números codificados en signo y magnitud que tienen el mismo signo, seguiremos el procedimiento siguiente:

  1. El signo del resultado es el signo de los operandos. En nuestro caso, el resultado será positivo (bit de signo 0).

  2. Sumamos las magnitudes de los operandos:

$$\begin{array}{cccccccccl}
 & 1 & 1 & 1 & 1 & 1 & & & \leftarrow & \text{transportes} \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & \leftarrow & A \\
+ & 0 & 1 & 1 & 1 & 1 & 1 & 0 & \leftarrow & -B \\
\hline
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & &
\end{array}$$

     El hecho de que aparezca un transporte en la última etapa de la suma de las magnitudes en una operación de suma de operandos con el mismo signo, indica que hay desbordamiento

  3. Para obtener el resultado, tenemos que añadir el signo al resultado de la suma: $00011000_{(2}$

  Como hay desbordamiento, no hay que indicar el resultado en decimal.

Examen 2010/11-1

Asignatura	**Código**	**Fecha**	**Hora inicio**
Fundamentos de computadores	75.562	29/01/2010	12:00

PROBLEMA 2 [35%]

Dado el siguiente circuito, formado por puertas lógicas, un descodificador y un multiplexor.

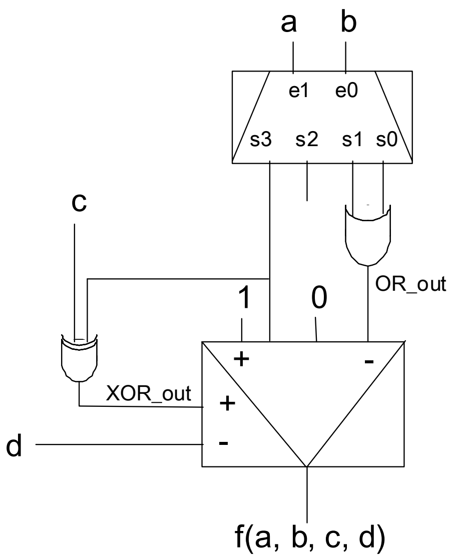


Tabla de verdad del descodificador:

e_1	e_0	s_3	s_2	s_1	s_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

a) **[15%]** Obtened la tabla de verdad de la función de salida *f*(*a*,*b*,*c*,*d*).

a	*b*	*c*	*D*	*OR_out*	*s3*	*XOR_out*	*f*
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				

Examen 2010/11-1

Asignatura	**Código**	**Fecha**	**Hora inicio**
Fundamentos de computadores	75.562	29/01/2010	12:00

Calculamos las señales intermedias que se indican en la tabla hasta llegar a la función de salida del circuito. La tabla de verdad resultante es la siguiente:

a	*b*	*c*	*d*	*OR_out*	*s3*	*XOR_out*	*f*
0	0	0	0	1	0	0	1
0	0	0	1	1	0	0	0
0	0	1	0	1	0	1	0
0	0	1	1	1	0	1	1
0	1	0	0	1	0	0	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	1	0
0	1	1	1	1	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
1	0	1	0	0	0	1	0
1	0	1	1	0	0	1	1
1	1	0	0	0	1	1	1
1	1	0	1	0	1	1	1
1	1	1	0	0	1	0	0
1	1	1	1	0	1	0	0

b) **[5%]** Expresad la salida *f* en suma de mintérminos.

La función *f* tiene 7 mintérminos. La expresión resultante para *f* es

$$f = a'b'c'd' + a'b'cd + a'bc'd' + a'bcd + ab'cd + abc'd' + abc'd$$

c) **[5%]** Indicad el tamaño mínimo de una memoria ROM que calcula la función *f* e indicad también su contenido.

La memoria ROM deberá tener tantas entradas de direcciones como entradas tiene el circuito. Cada posición de memoria almacenará un bit, que corresponde al valor de la función *f*. El tamaño de la ROM será de 2^4*1 bits.

El contenido de la memoria se muestra a continuación, teniendo en cuenta que las señales de entrada del circuito (*a*, *b*, *c*, *d*) se conectan a las entradas de direccionamiento de la memoria en orden de peso decreciente. De esta forma el contenido de la memoria coincide en orden con los valores obtenidos en la tabla de verdad de *f*.

Dirección	***Contenido***
0	1
1	0
2	0
3	1
4	1
5	0
6	0
7	1
8	0
9	0
10	0
11	1
12	1
13	1
14	0
15	0

d) **[10%]** Dada la siguiente tabla de verdad con las entradas x_2- x_1- x_0 y la salida *s*.

x_2	x_1	x_0	s
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Minimizad la salida *s* utilizando el método de Karnaugh y dibujad el circuito que calcula *s* con el mínimo número de puertas.

Examen 2010/11-1

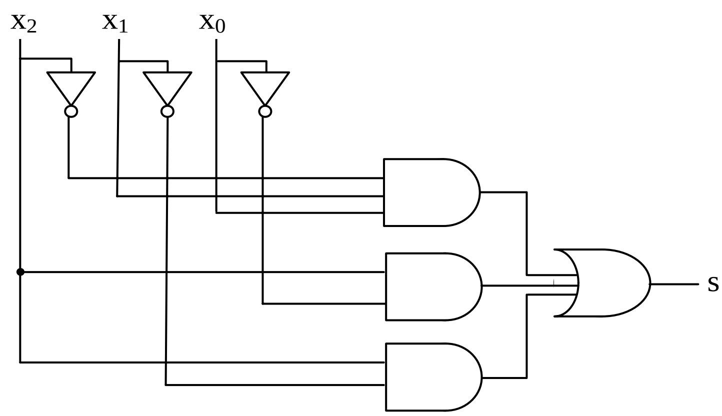
Asignatura	**Código**	**Fecha**	**Hora inicio**
Fundamentos de computadores	75.562	29/01/2010	12:00

El mapa de Karnaugh para la función *s* es el siguiente:

x_0 \ x_2x_1	**00**	**01**	**11**	**10**
0	0	0	1	1
1	0	1	0	1

La expresión algebraica minimizada de la función es $s = x_2'x_1x_0 + x_2x_0' + x_2x_1'$

El circuito que obtenemos a partir de esta expresión es

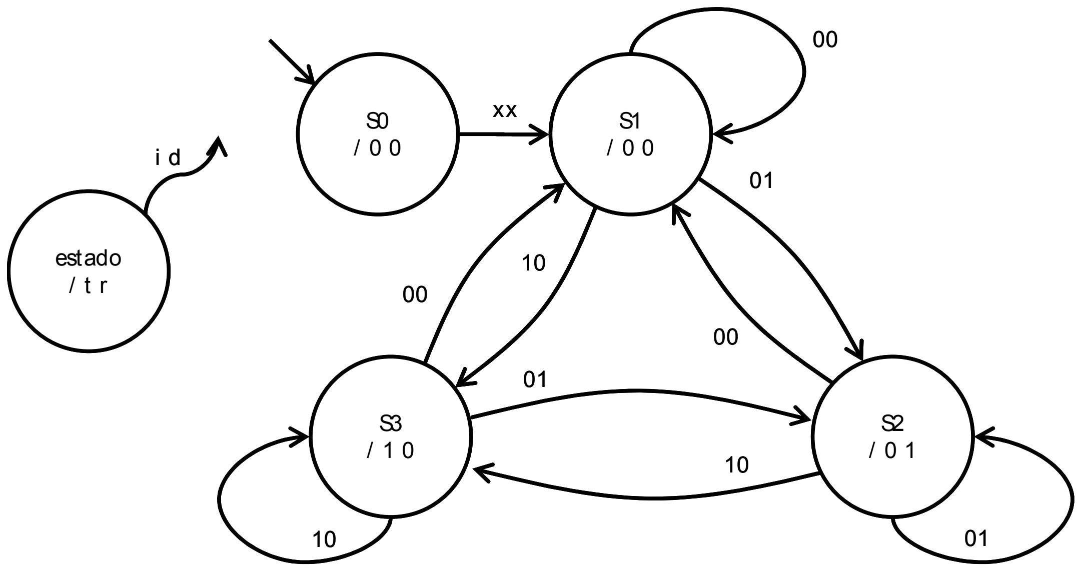


Examen 2010/11-1

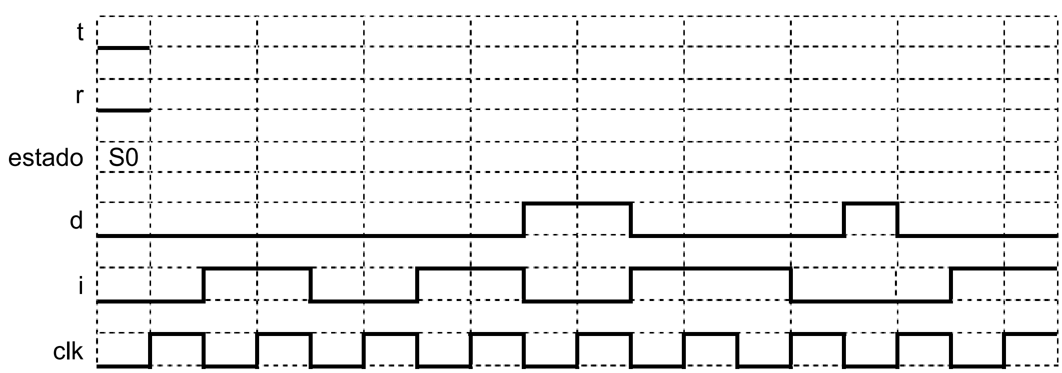
Asignatura	**Código**	**Fecha**	**Hora inicio**
Fundamentos de computadores	75.562	29/01/2010	12:00

PROBLEMA 3 [35%]

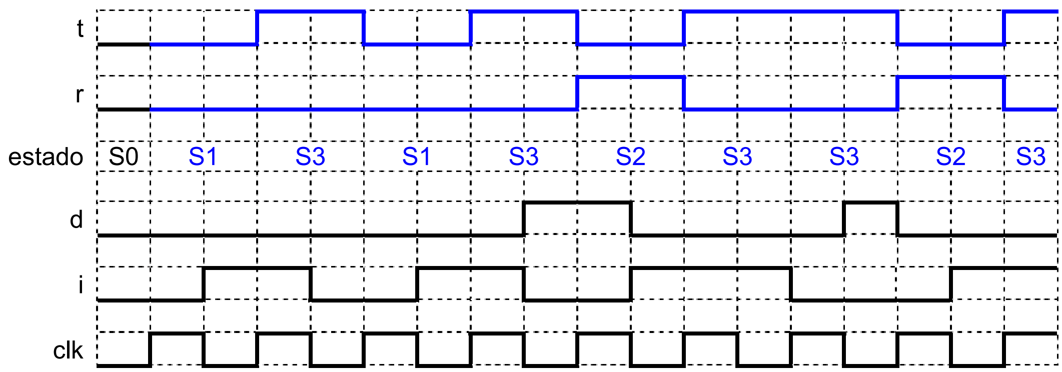
a) **[15%]** El grafo de transición de estados siguiente se corresponde con el de un detector de sentido de giro. Las entradas (*y*, *d*) indican si se detecta un incremento o una disminución en la posición absoluta del rotor respecto del ciclo anterior de reloj. Las salidas *t* y *r* se actualizan según si el motor gira en el sentido de las agujas del reloj (*r*=1) o si lo hacen en sentido contrario (*t*=1). Cuando no hay variación, la salida es 00.



Completad el cronograma siguiente para la secuencia de entradas que se dan.

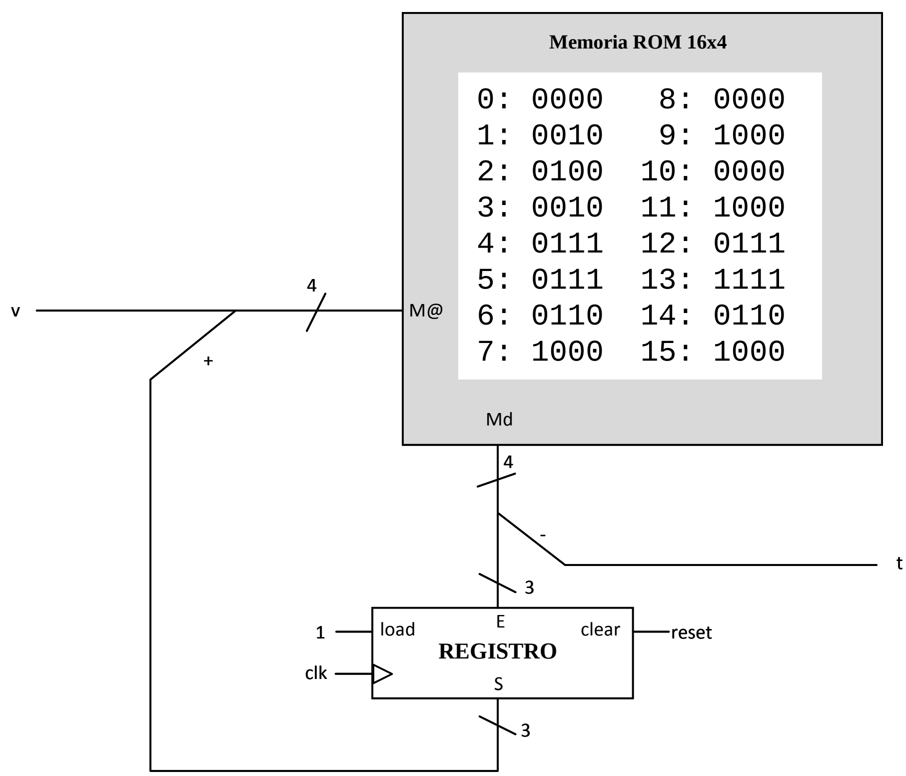


Hay que tener en cuenta, cuando una señal de entrada cambia de valor coincidiendo con un flanco ascendente (como la entrada *d* en el penúltimo flanco ascendente del cronograma), que el valor que se tiene que considerar, para evaluar la transición que se produce, es el que tiene a la izquierda del flanco.



Los cambios en las salidas *t* y *r* dependen exclusivamente del valor del estado y, por lo tanto, una vez que se sabe el estado de la máquina, se sabe también el valor.

b) **[20%]** El circuito siguiente es un controlador de un timbre de aviso de entrada de vehículos en una estación de servicio. El timbre suena cada vez que recibe un pulso de la salida *t* del controlador. Obtened el grafo de transición de estados correspondiente.



El controlador detecta los vehículos con un sensor de paso de ejes de ruedas que activa la entrada *v* poniéndola a 1 mientras haya un vehículo pasando.

De la ROM se puede obtener directamente la siguiente tabla. En el esquema podemos ver que los 3 bits de más peso de la salida de la ROM proporcionan el estado futuro, mientras que el bit de menor peso codifica el valor de la señal *t*.

M@				Md			
q_2	q_1	q_0	v	q_2^+	q_1^+	q_1^+	t
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	1
0	1	1	0	0	1	1	0
0	1	1	1	1	0	0	0
1	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	0	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	0	1	1	1
1	1	0	1	1	1	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	0

Denominaremos *Si* en el estado correspondiente a la codificación de [*q*2 *q*1 *q*0] que leída en binario corresponde al número i. El estado inicial se identifica con *S*0 y se corresponde con el de código 000, para ser el estado en que el circuito se encuentra al ponerse en marcha (después de un *reset*). Para construir el grafo, se parte de *S*0 y se examinan las transiciones según la entrada correspondiente, siguiendo la tabla.

La salida es 1 para los estados *S*2 (010) y *S*6 (110).

Al seguir la tabla, sin embargo, se puede ver que, una vez que la máquina ha llegado al estado *S*4, o bien permanece en *S*4 o bien pasa a *S*0. En consecuencia, los estados *S*5, *S*6 y *S*7 son inalcanzables.