

Examen 2024/25-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	22/1/2025	10:00

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%**
 - ¿Se puede consultar material durante la prueba? **No** ¿Qué materiales están permitidos?
Ninguno
 - ¿Puede utilizarse calculadora? **No** ¿De qué tipo?
 - Si hay preguntas tipo test, ¿descuentan las respuestas erróneas? ¿Cuánto?
 - Indicaciones específicas para la realización de este examen: **Razonad todas las respuestas. Las respuestas sin justificar no serán puntuadas.**
-

Examen 2024/25-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	22/1/2025	10:00

Enunciados

PROBLEMA 1 [20%]

Responded los apartados siguientes:

- a) [5 %] Dado el valor decimal $B = -137$, representad B en el formato de signo y magnitud con 10 bits.

En el formato signo y magnitud el primer bit indica el signo, 1 para los números negativos y 0 para los positivos. El resto de los 9 bits son para el valor absoluto del número.

Usamos el método de la división entera por 2 para obtener el valor binario de $137_{(10)}$:

$$\begin{array}{rcl}
 137 & = & 68 \cdot 2 + 1 \\
 68 & = & 34 \cdot 2 + 0 \\
 34 & = & 17 \cdot 2 + 0 \\
 17 & = & 8 \cdot 2 + 1 \\
 8 & = & 4 \cdot 2 + 0 \\
 4 & = & 2 \cdot 2 + 0 \\
 2 & = & 1 \cdot 2 + 0 \\
 1 & = & 0 \cdot 2 + 1 \\
 137_{(10)} & = & 10001001_{(2)}
 \end{array}$$

Así pues, la codificación en SM2 y 10 bits del valor $-137_{(10)}$ es **$1010001001_{(SM2)}$**

- b) [7.5 %] Dados los valores $C=00100100$ y $D=01111010$, que representan dos números binarios enteros expresados en formato de signo y magnitud, calculad $C-D$ utilizando el mismo formato. ¿Se produce desbordamiento? Justificad la respuesta

Para realizar la resta entre C y D , tenemos que restar sus magnitudes teniendo en cuenta cuál es la más grande. Hay que restar la magnitud menor de la mayor y aplicar el signo de la mayor. Podemos observar que la magnitud de D es más grande que la magnitud de C .

$$\begin{array}{r}
 \begin{array}{cccccccc}
 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
 & & & 1 & & & & \\
 - & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
 \hline
 1 & 0 & 1 & 0 & 1 & 1 & 0 &
 \end{array}
 \end{array}
 \begin{array}{l}
 \leftarrow |D| \\
 \leftarrow \text{acarreo} \\
 \leftarrow |C|
 \end{array}$$

Al resultado le ponemos el signo del número más grande D (1), la suma de un número negativo y un positivo no puede producir desbordamiento.

$$C - D = 11010110_{(SM2)}$$

Examen 2024/25-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	22/1/2025	10:00

- c) [7.5 %] Dado el número fraccionario binario 111100000101 que codifica un valor en coma flotante según el formato de representación siguiente:

S	Exponente	Mantisa
11	10	7 6 0

Donde:

- el bit de signo S vale 0 para los números positivos, y 1 para los negativos
 - método de aproximación por truncamiento
 - el exponente se codifica en exceso a 8, y
 - la mantisa está normalizada de la forma $1,M$ y con bit implícito.
- ¿A qué número decimal corresponde?

Los números en coma flotante toman la forma $\pm R \cdot b^e$ donde se indica el signo, R es un número fraccionario que recibe el nombre de mantisa, b es la base de la numeración y e es un número entero que recibe el nombre de exponente.

La identificación del formato de la numeración nos indica que el primer bit de la cadena de bits indica el signo. En este caso, el primer bit es un 1 y por tanto, el número es **negativo**.

Respecto la mantisa, vemos que ocupa las 7 posiciones menos significativas de la cadena de bits. Por lo tanto, se corresponde con la cadena: 0000101. Ahora bien, el formato indica que hay bit implícito y está normalizada de la forma $1,M$. Esto implica que solo muestra la parte variable de las mantisas normalizadas y se asume la parte fija como conocida y definida en el formato de representación. En este ejercicio el número representado por la mantisa es **1,0000101₂**.

Respecto al exponente, el formato nos indica que está representado por 4 bits y además se nos indica que se ha codificado con exceso a 8. Esto implica que el valor del exponente se conseguirá a partir del resto entre el número codificado con exceso y este exceso. En este ejercicio el exponente es $1110_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 14_{10}$ sabiendo que hay una codificación con exceso de 8, el exponente será igual a $14_{10} - 8 = 6_{10}$.

Ahora se puede unir el signo, el exponente y la mantisa para obtener el número representado

$$- 1,0000101_2 \cdot 2^6$$

Haremos un cambio de base para obtener el valor decimal

$$\begin{aligned} - 1,0000101_2 \cdot 2^6 &= - (1 \cdot 2^0 + 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 0 \cdot 2^{-3} + 0 \cdot 2^{-4} + 1 \cdot 2^{-5} + 0 \cdot 2^{-6} + 1 \cdot 2^{-7}) \cdot 2^6 \\ &= \mathbf{-66,5_{10}} \end{aligned}$$

Examen 2024/25-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	22/1/2025	10:00

PROBLEMA 2 [35%]

a) [10%] Minimizar la siguiente función por Karnaugh y hacer la síntesis del circuito a dos niveles:

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

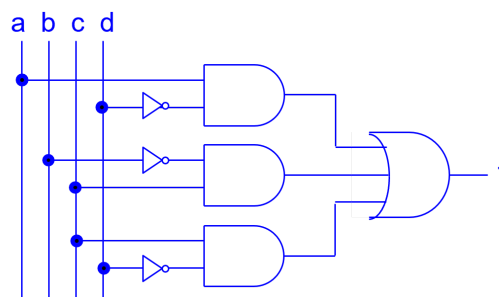
El mapa de Karnaugh para la función f es el siguiente:

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

Y obtenemos esta expresión mínima:

$$f = ad' + b'c + cd'$$

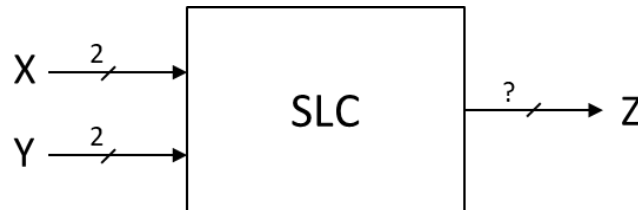
El circuito que la implementa a dos niveles con puertas lógicas es el siguiente:



Examen 2024/25-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	22/1/2025	10:00

b) **[25%]** Un sistema combinacional SLC responde al siguiente bloque:



donde X e Y son números naturales de 2 bits. La salida Z también es un número natural.

La salida Z viene determinada por la siguiente expresión numérica:

$$Z = \text{MÁXIMO}(3 \cdot X^2, 2 \cdot Y^2 + 1)$$

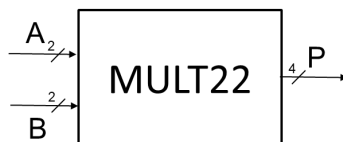
b.1) **[5%]** Indicad la dimensión de la señal de salida Z para poder representar correctamente la salida con el mínimo número de bits.

Para calcular la dimensión de la señal de salida tenemos que calcular cual puede ser el resultado máximo que se puede producir en este circuito. El resultado máximo vendrá dado o bien por la función $3 \cdot X^2$ o bien por la función $2 \cdot Y^2 + 1$, y se dará cuando $X = 3$ (para la primera función) o cuando $Y = 3$ (para la segunda). En este caso:

$$\begin{aligned} 3 \cdot X^2 &= 3 \cdot 3^2 = 3 \cdot 9 = 27 \\ 2 \cdot Y^2 + 1 &= 2 \cdot 3^2 + 1 = 2 \cdot 9 + 1 = 18 + 1 = 19 \end{aligned}$$

Así pues, el valor máximo que puede generar este circuito es el número 27. Para poder representar correctamente el número 27 necesitamos un mínimo de $\log_2(27)$ bits, es decir, **necesitaremos 5 bits**.

b.2) **[20%]** Realizad el circuito SLC utilizando bloques MULT22 como los utilizados en la PEC2, que realizaba la multiplicación de números naturales de 2 bits:



Adicionalmente podéis usar cualquiera de los bloques y puertas explicados en el temario.

Se valorará que todos los buses internos del circuito, así como la salida, tengan definidos el número de bits correctos para que no se produzca nunca desbordamiento en los cálculos realizados.

Utilizaremos los bloques MULT22 para calcular las operaciones X^2 e Y^2 . Lamentablemente no los podremos utilizar para hacer los cálculos $3 \cdot X^2$ ni tampoco $2 \cdot Y^2$, ya que tanto X^2 como Y^2 son valores de 4 bits y no nos cabrían en los bloques MULT22.

En este caso, para multiplicar un cierto número N de cuatro bits por dos se hace desplazando un bit a la izquierda (o bien añadiendo un cero como bit de menor peso) de forma que el resultado tendrá 5 bits. Similarmente, para multiplicar un número N por tres se hace sumando $(N \cdot 2) + N$. Como que $N \cdot 2$ es un valor de 5 bits, la suma $(N \cdot 2) + N$ se tiene que hacer con un sumador de 5 bits.

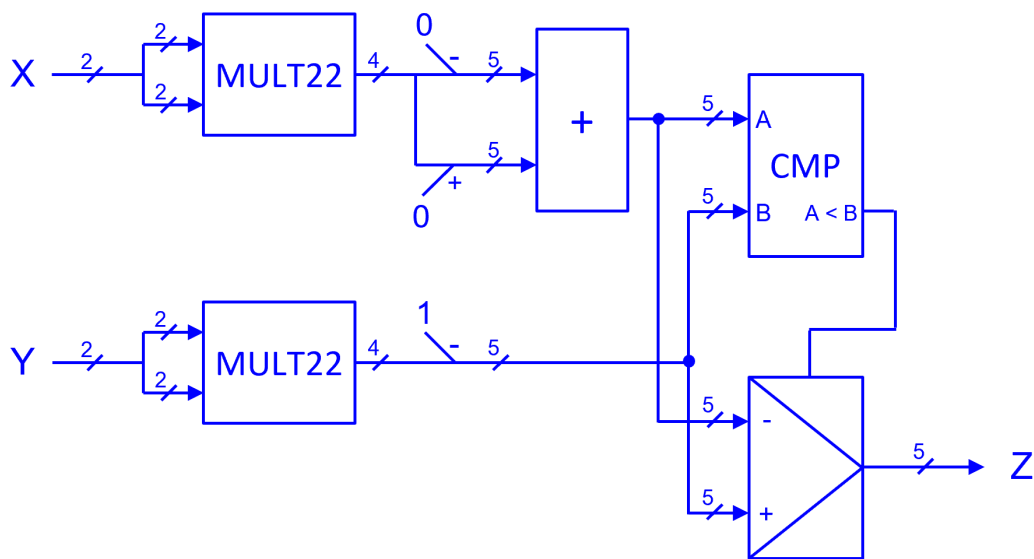
Examen 2024/25-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	22/1/2025	10:00

Una mejora que se puede hacer al calcular $2 \cdot X^2 + 1$ es que, en lugar de añadir un 0 como bit de menor peso al multiplicar X^2 por dos, añadir un 1, y así ya queda hecho el incremento +1.

Finalmente, la operación MÁXIMO se implementa mediante un comparador y un multiplexor, de forma que la función Z coge el valor máximo entre las dos funciones calculadas.

El circuito resultante es lo siguiente:



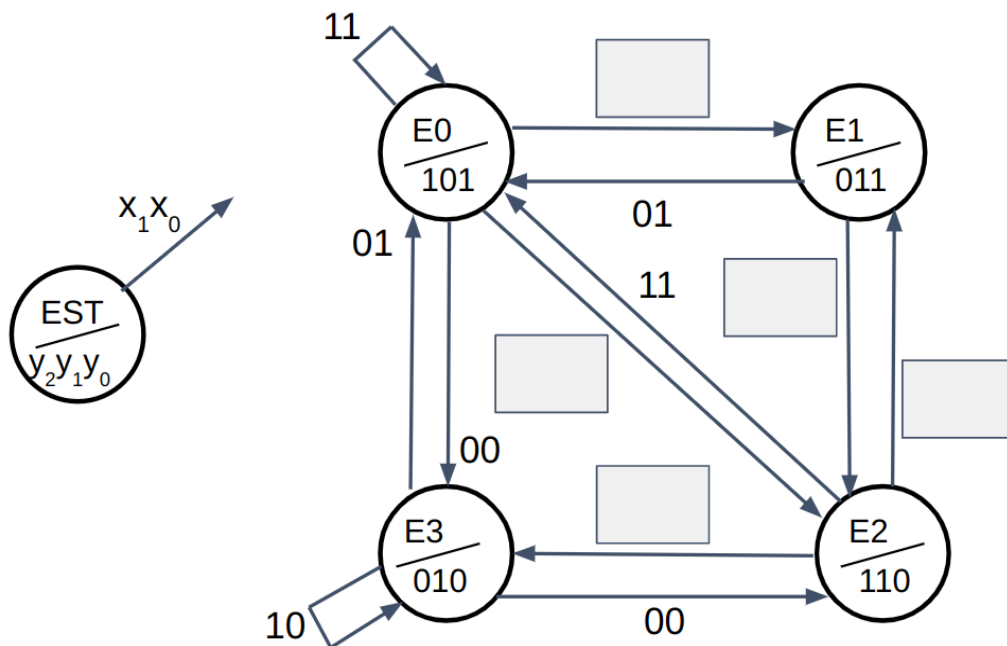
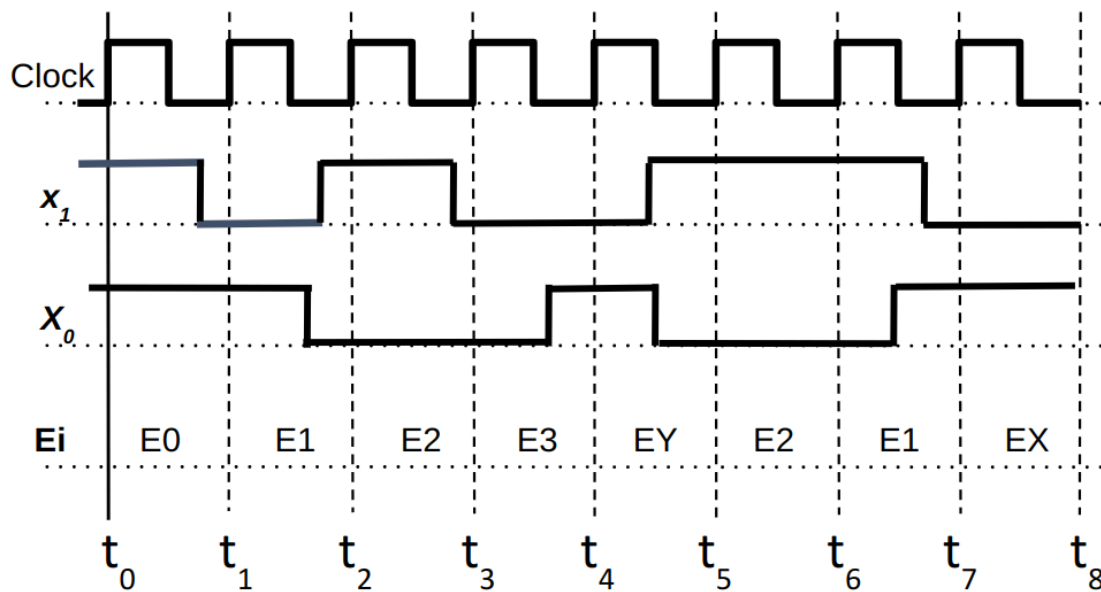
Como soluciones alternativas, las multiplicaciones por 2 y por 3 se pueden hacer mediante dos o tres sumas, respectivamente. En este caso, hay que gestionar correctamente los posibles acarreos de cada sumador, así como el correspondiente número de bits en cada salida de los sumadores.

Examen 2024/25-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	22/1/2025	10:00

PROBLEMA 3 [35%]

a) [15%] Dados el cronograma y el grafo de estados siguientes:



Examen 2024/25-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	22/1/2025	10:00

i) **[10%]** Completad el grafo de estados a partir del cronograma e indicad a qué estados del grafo corresponden los estados *EX* y *EY* del cronograma.

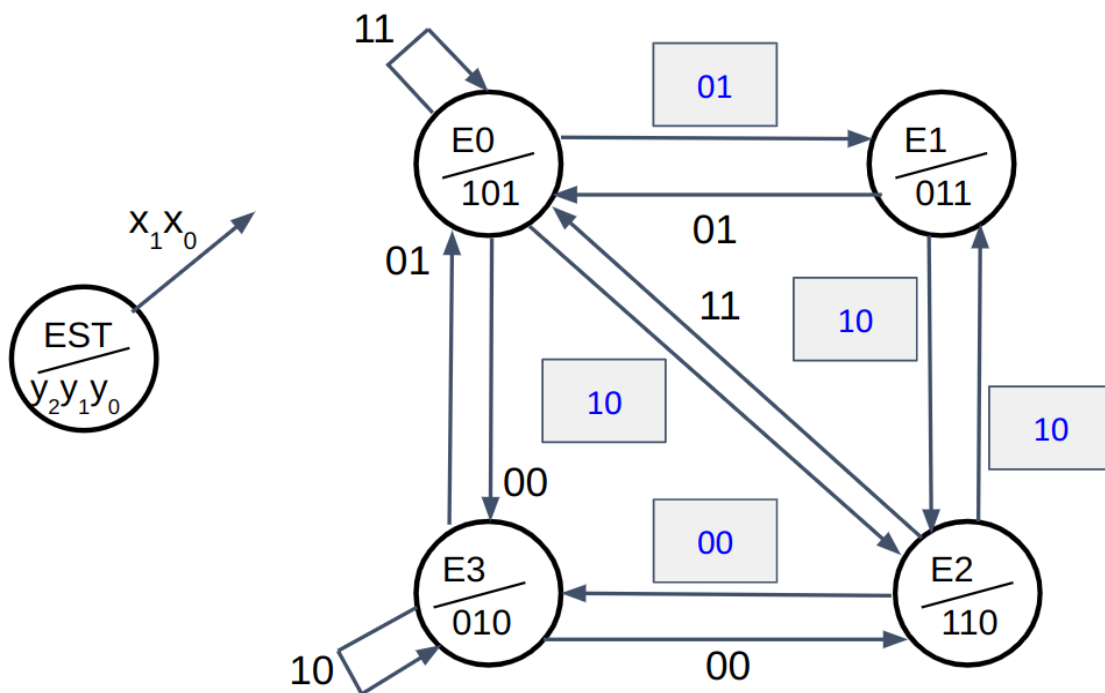
Nota: Las únicas combinaciones de entrada posibles son las indicadas en el grafo y las que se pueden deducir del cronograma.

Para obtener los valores que han de tomar las entradas para pasar de *E0* a *E1*, observamos en el cronograma los valores que toman x_1 y x_0 por la izquierda, (justo antes del flanco ascendente de reloj que se produce al pasar de *E0* a *E1*) que en este caso resultan ser los valores $[x_1 \ x_0] = [0 \ 1]$. Se sigue un razonamiento similar para completar los recuadros del grafo de estados.

En el cronograma se puede observar que, estando en el estado *E3*, cuando se produce la combinación de entradas $[x_1 \ x_0] = [0 \ 1]$ se produce una transición hacia el estado *EY*. Por otra parte, en el grafo de estados se puede observar que en el estado *E3*, cuando el circuito recibe como entradas $[x_1 \ x_0] = [0 \ 1]$, se produce una transición hacia el estado *E0*, por lo tanto, **EY es E0**.

Siguiendo un razonamiento similar al anterior para las transiciones desde el estado *E1* se deduce que **EX es E0**.

Finalmente, se obtiene el grafo de estados siguiente:



ii) **[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, ¿cuántos bits de direcciones y cuántos bits de datos necesitará esta memoria?

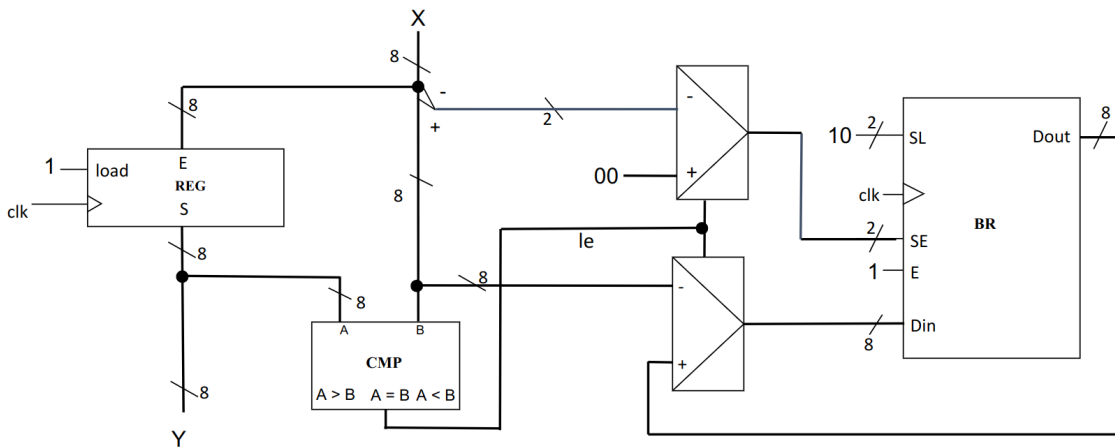
A partir de la leyenda vemos que el circuito tiene **dos bits de entrada**, denominados x_1 y x_0 , y **tres bits de salida**, denominados y_2 , y_1 y y_0 . Como el circuito tiene 4 estados, $\log_2(4) \approx 2$, necesitamos 2 bits para representarlos y, por lo tanto, necesitamos **2 biestables** para poder almacenar estos 2 bits.

Examen 2024/25-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	22/1/2025	10:00

La memoria ROM que puede implementar 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 tendrá que ser de 5 bits, 3 bits para guardar la salida y 2 bits para codificar el estado futuro.

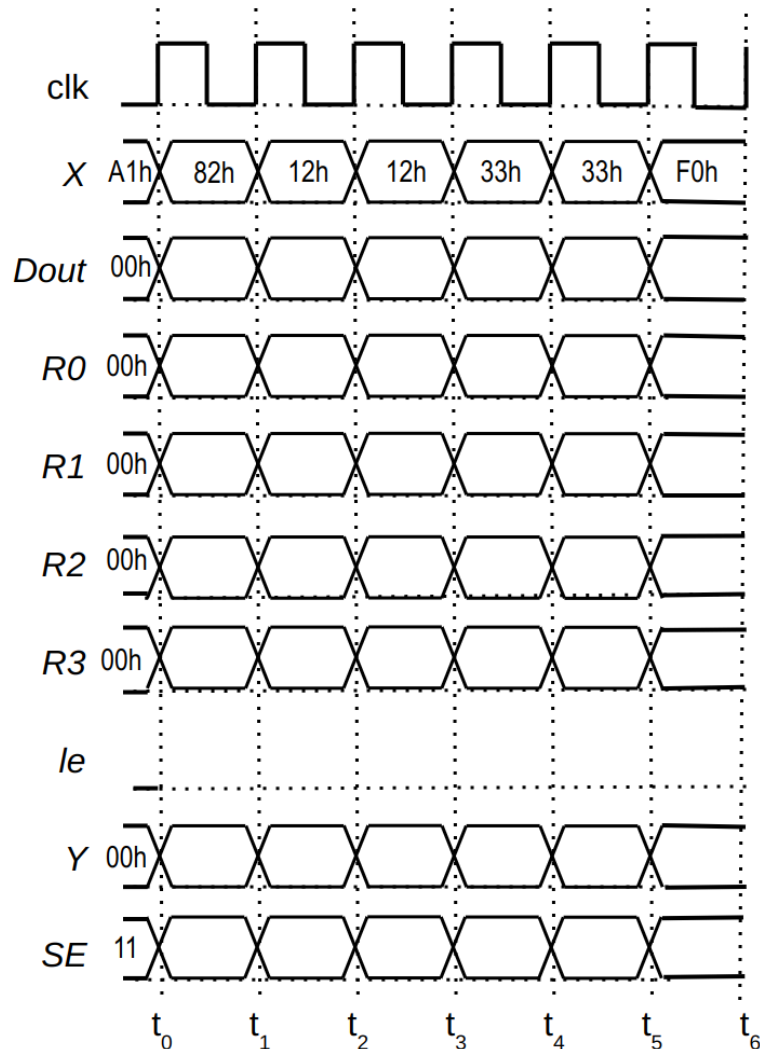
- b) **[20%]** Dado el circuito siguiente, donde BR es un banco de 4 registros que denominamos R0, R1, R2 y R3.



Completad el cronograma siguiente, poniendo los valores de los registros en hexadecimal y el valor de *SE* en binario. No hace falta que justificuéis la respuesta.

Examen 2024/25-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	22/1/2025	10:00



Para rellenar el cronograma debemos primer saber cómo calcular cada una de las señales:

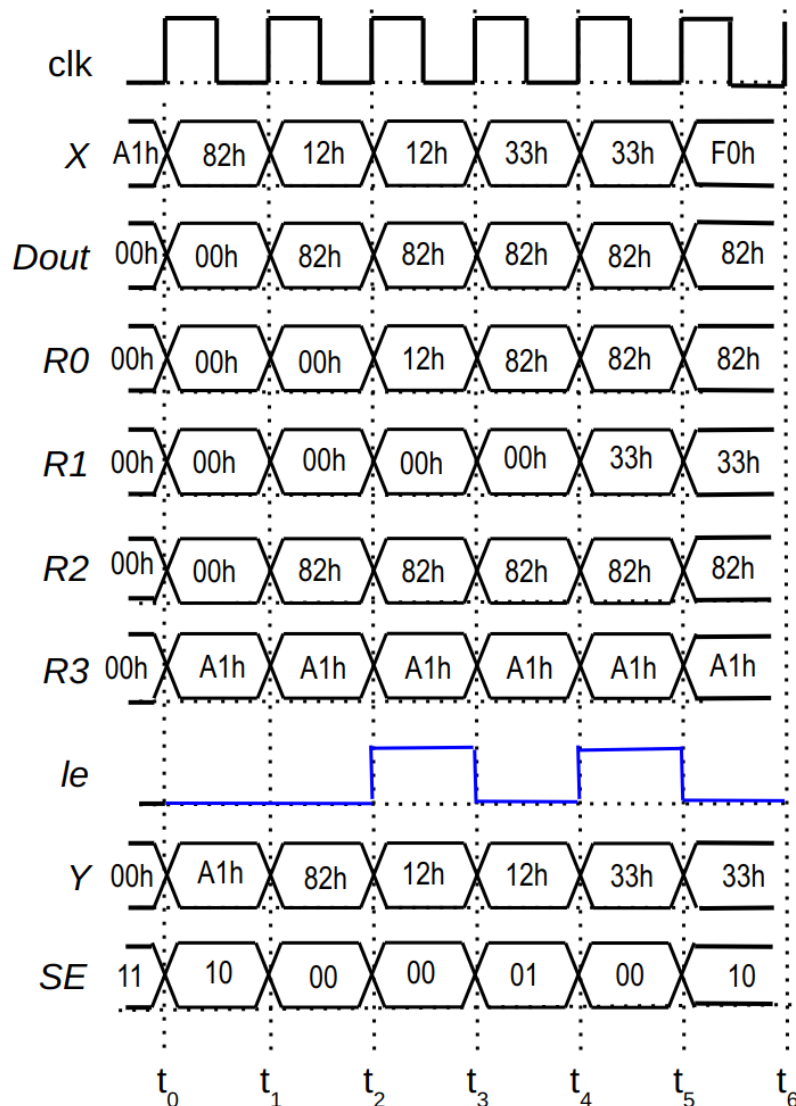
- Registro REG / Salida Y: En cada flanco ascendente del reloj, se carga la entrada X del circuito y su valor aparece en la salida Y del registro.
- SE: El valor de SE depende de la entrada de la señal de selección *le* del multiplexor. Si *le* = 0, entonces SE corresponde a los bits de mayor y menor peso de la salida del registro. En caso de *le* = 1, entonces SE = 00.
- le: La señal *le* es 1 en caso de que el valor de la señal Y y de la señal X coincidan.
- SL: El valor de SL siempre es 10. Por lo tanto, el valor de *Dout* en un ciclo concreto es el valor del registro R2 en este mismo ciclo.
- BR: En relación con el banco de registros BR, observamos que *E* = 1, por tanto, la escritura siempre está habilitada. En cuanto a la escritura, se realiza en el flanco ascendente del reloj, por lo tanto, se escribe en el registro indicado por el valor de SE. El valor que se escribe en este registro es el valor que llega a la entrada *Din*, que corresponde a la salida de un multiplexor en el momento del flanco ascendente. El valor de esta salida del multiplexor es X en caso de *le* = 0, y es *Dout* en el caso de *le* = 1.

Examen 2024/25-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	22/1/2025	10:00

Por ejemplo, al llegar al instante t_0 , el valor de SE es 11 y se escribe en R3 el valor de $X = A1h$. El valor de los registros R0-R2 no cambia. En el ciclo $t_0 - t_1$ $le = 0$ ya que $X = 82h$ y $Y = A1h$ no coinciden. Por lo tanto, $SE = 10$ y el valor que se escribirá por Din en el registro R2 del BR en el siguiente ciclo será 82h.

Otro ejemplo es al llegar al instante t_3 . Dado que $le=1$ en el ciclo $t_2 - t_3$, SE tiene el valor 00 y Din el valor de $Dout = 82h$ en el ciclo $t_2 - t_3$. Por lo tanto, R0 en el ciclo $t_3 - t_4$ obtiene este valor 82h.



Examen 2024/25-1

Asignatura	Código	Fecha	Hora inicio
Fundamentos de computadores	75.562	22/1/2025	10:00

PROBLEMA 4 [10%]

a) [5%] ¿Qué es una ALU?

Un recurso de cálculo programable.

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.