



Estructura y Tecnología de Computadores

Examen final

9 de julio de 2010

Grupo:	DNI:
Nombre:	Apellidos:

Instrucciones:

- **Sólo en una de las preguntas se podrá obtener una nota inferior a 3.** Si no se cumple esta condición, el examen estará suspenso.
- La nota obtenida en el examen, una vez se divida por 4, representará el 50 % de la nota final de la asignatura, siendo necesario obtener una calificación mayor o igual que 5 para eliminar esta parte.
- No se dará por válida ninguna respuesta que no contenga el desarrollo de la solución paso a paso.
- El peso de cada ejercicio aparece entre paréntesis.
- Responda a las preguntas en bolígrafo.

1. (10 puntos) Diseñar un circuito, con el mínimo número de componentes, que proporcione a sus 3 salidas el equivalente binario de la secuencia 5 – 2 – 4 – 1, en cuanto reciba una orden de activación por su entrada *START*. Cuando esta orden cese, el circuito quedará con el estado en que se encontrase en ese instante, esperando a que se produzca de nuevo la activación de *START* para comenzar a generar la secuencia. Utilizar biestables tipo S-R disparados por flanco ascendente. No es necesario dibujar el circuito, basta con obtener las ecuaciones simplificadas del mismo.
2. (10 puntos) Utilizando la metodología descrita en los apuntes para la inclusión de nuevas instrucciones en el esquema de implementación multiciclo, realice las fases de análisis y diseño de la siguiente instrucción:

- beqri \$a, \$b, \$c

Si los contenidos de los registros \$a y \$b coinciden, salta a la dirección de memoria contenida en la palabra de memoria apuntada por \$c.

\$a, \$b y \$c se refieren a cualquiera de los 32 registros de propósito general de MIPS.

Para que la solución del ejercicio se considere correcta, es importante que incluya claramente todos los pasos correspondientes de las fases de análisis y diseño. Las modificaciones al camino de datos y la unidad de control deberán estar explicadas por escrito. De forma complementaria, se pueden realizar modificaciones y anotaciones sobre los diagramas originales para facilitar la explicación.

3. (10 puntos) La máquina M posee un sistema de memoria virtual con las siguientes características:
 - Tabla de páginas de 512 entradas.
 - Memoria física de 64 KB.
 - Tamaño de página virtual de 256 bytes.
 - TLB asociativo con 16 conjuntos de 2 vías (2 bloques por conjunto). El TLB sigue una estrategia de reemplazo aleatoria y una política de postescritura.

Se pide:

- a) Especificar detalladamente el formato de la dirección física y de la dirección virtual. Dibujar un esquema de la tabla de páginas y calcular su tamaño incluyendo los bits de control necesarios (justificando su inclusión). Para cada entrada de la tabla de páginas, su bit de validez es 1 si el número de página virtual DIV 8 es 0. Además, el número de página física almacenado es $8 - \text{número de página virtual}$.
 - b) Dibujar un esquema detallado del TLB y calcular su tamaño incluyendo todos los bits de control necesarios (justificando su inclusión).
 - c) Mostrar de forma exacta cómo evoluciona el contenido del TLB y la tabla de páginas después de cada una de las siguientes referencias a bytes (direcciones virtuales) generadas por el procesador: L25, E904 y E75 (L = LECTURA, E = ESCRITURA), determinando en cada referencia a memoria si se produce un acierto/fallo de TLB y/o un acierto/fallo de página y obtener la dirección física resultante. Las direcciones vienen dadas en base 10.
 - d) La máquina M posee una memoria caché de datos de correspondencia directa con bloques de 4 palabras y 64 bytes de datos en total. La caché sigue una política de postescritura. Dibuja un esquema detallado de la memoria caché y calcular su tamaño incluyendo los bits de control necesarios (justificando su inclusión). Las palabras son de 16 bits.
4. (10 puntos) Disponemos de una cámara digital de 10 megapíxeles donde cada píxel está representado por 24 bits (8 bits por canal de color). Las fotos de la cámara se pueden descargar conectándola a un PC a través del bus USB 2.0 y transferirlos a la memoria del procesador con un ancho de banda efectivo de 15 MBytes/s de datos a través del bus, el cual tiene un ancho de banda total de 480 Mbits/s. Si el procesador utiliza memoria virtual paginada con un tamaño de página de 4 KB y opera a una frecuencia de 3 GHz se pide:
- a) El % de ocupación de la CPU para la E/S al descargar imágenes de la cámara de fotos si la transferencia se realiza mediante interrupciones y se necesitan 400 ciclos de reloj para transferir 4 bytes de datos.
 - b) El % de ocupación de la CPU si la transferencia se realiza mediante DMA y en cada operación DMA se transfiere una página (4 KB) completa a memoria principal. La preparación de la operación de DMA necesita 4000 ciclos de reloj y el tratamiento de la interrupción de terminación es de 8000 ciclos.
 - c) Supongamos que la descarga de imágenes no se pudiese hacer mediante interrupciones ni DMA, y que sólo se puede acceder a la cámara mediante polling. Si suponemos que cada palabra transferida (4 bytes) necesita una operación de encuesta, la cual consume 1000 ciclos de CPU, ¿cuál debería ser la frecuencia de la CPU para poder soportar la técnica de polling?
 - d) ¿Cuál sería el tiempo total en segundos, incluyendo los ciclos requeridos de preparación y terminación, necesario para transmitir una imagen completa por el método basado en DMA?

Nota: considerar $1 \text{ MB} = 2^x \text{ bytes}$; $1 \text{ KB} = 2^y \text{ bytes}$ y $1 \text{ GHz} = 10^z \text{ hercios}$.

Soluciones

Ejercicio 1

El autómata que modeliza el sistema a implementar constará de 4 estados, correspondientes con las 4 posibles salidas a mostrar. La única entrada del sistema es la señal *START* que habilita la transición entre estados. El autómata será por tanto el indicado en 1.

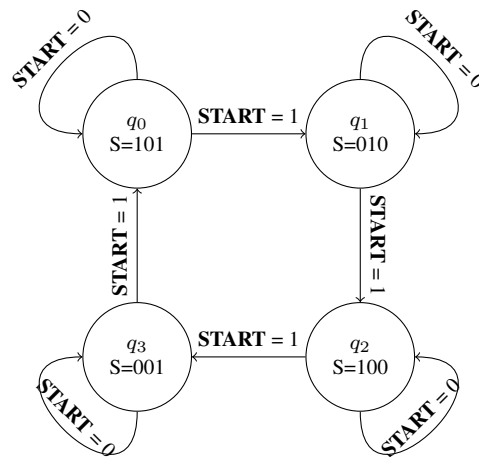


Figura 1: Autómata que modeliza el sistema secuencial a implementar.

Representado en forma de tabla tendremos:

Estado Actual	Estado Siguiente		Salida $Sal_2Sal_1Sal_0$
	$START = 0$	$START = 1$	
q_0	q_0	q_1	101
q_1	q_1	q_2	010
q_2	q_2	q_3	100
q_3	q_3	q_0	001

Cuadro 1: Tabla de transición entre estados y función salida para nuestro autómata.

Al haber 4 estados, es suficiente con dos flip-flops cuyas salidas serán Q_0 y Q_1 . Utilizaremos la codificación en binario natural ($q_0 = 00$, $q_1 = 01$, $q_2 = 10$, $q_3 = 11$). Una vez realizada la codificación de estados el siguiente paso es reescribir dicha tabla haciendo uso de la codificación elegida tal y como se muestra en la tabla 2. Obsérvese que diferenciamos el estado futuro (Q_i^*) del actual (Q_i) utilizando un asterisco.

Q_1Q_0	$Q_1^*Q_0^*$		Salida
	$START = 0$	$START = 1$	$Sal_2Sal_1Sal_0$
00	00	01	101
01	01	10	010
10	10	11	100
11	11	00	001

Cuadro 2: Tabla de transición entre estados y función de salida con los estados ya codificados.

Una vez realizada dicha codificación debemos determinar qué valores deben de tener las entradas de los biestables para que dicha transición se realice en la salida. En el caso de biestables tipo S-R usaremos la tabla 3.

$Q \rightarrow Q^*$	S	R
$0 \rightarrow 0$	0	—
$0 \rightarrow 1$	1	0
$1 \rightarrow 0$	0	1
$1 \rightarrow 1$	—	0

Cuadro 3: Determinación del valor de la entrada de un biestable S-R en función del estado actual y del estado deseado.

La tabla 4 muestra el cálculo de las entradas de excitación cuando utilizamos biestables S-R.

Q_1Q_0	$Q_1^*Q_0^*$		S_1R_1		S_0R_0	
	$START = 0$	$START = 1$	$START = 0$	$START = 1$	$START = 0$	$START = 1$
00	00	01	0 –	0 –	0 –	10
01	01	10	0 –	10	– 0	01
10	10	11	– 0	– 0	0 –	10
11	11	00	– 0	01	– 0	01

Cuadro 4: Cálculo de los valores de las entradas de excitación de los dos biestables tipo S-R.

El siguiente paso consiste en la simplificación de las entradas de excitación de ambos biestables así como de las tres salidas. La figura 2 muestra la simplificación de las mismas utilizando mapas de karnaugh. Los valores obtenidos son:

$$S_1 = \overline{Q_1} \cdot Q_0 \cdot START; R_1 = Q_1 \cdot Q_0 \cdot START$$

$$S_0 = \overline{Q_0} \cdot START; R_0 = Q_0 \cdot START$$

$$Sal_2 = \overline{Q_0}; Sal_1 = \overline{Q_1} \cdot Q_0; Sal_0 = \overline{Q_1} \cdot \overline{Q_0} + Q_1 \cdot Q_0 = \overline{Q_1} \oplus \overline{Q_0}$$

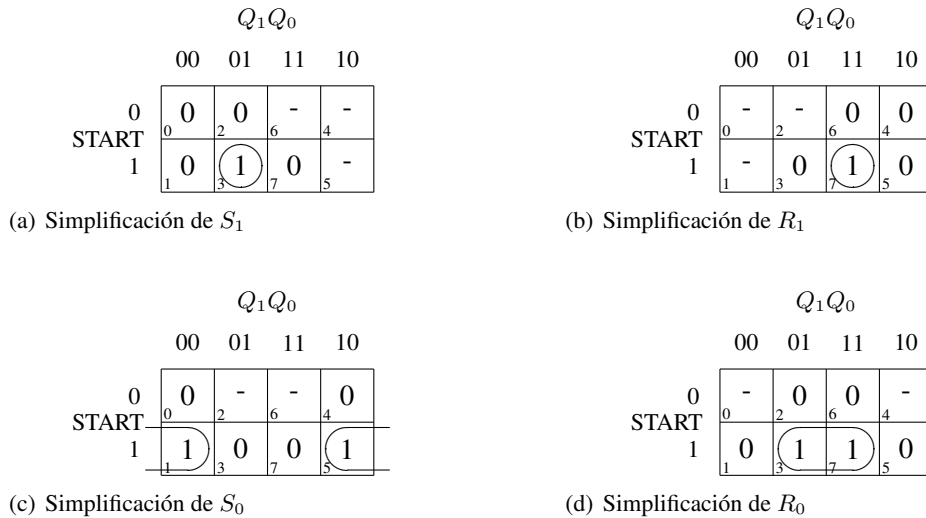


Figura 2: Simplificación de las entradas de los biestables S-R.

Ejercicio 2

■ Análisis:

1. Especificación precisa de la semántica de la instrucción:

La semántica de “beqri \$a, \$b, \$c” es:

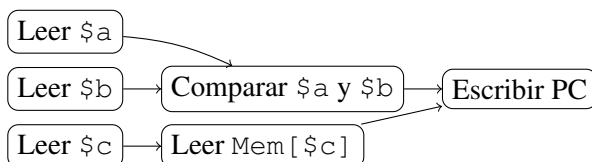
$$\text{si } \$a = \$b \Rightarrow PC \leftarrow \text{Mem}[\$c]$$

2. Identificación del trabajo a realizar por cada unidad funcional principal:

- Banco de registros:
 - Leer \$a.
 - Leer \$b.
 - Leer \$c.
- ALU:
 - Comparar \$a y \$b (restándolos).
- Memoria:
 - Leer Mem[\$c].

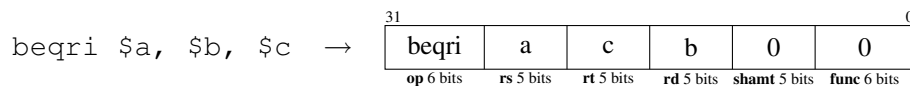
Además, se escribe en el registro PC el valor leído de memoria si el resultado de la ALU es cero.

3. Establecimiento del orden de precedencia entre las distintas tareas a realizar:



■ Diseño:

1. **Definición de la codificación de la instrucción:** Debido a que la instrucción tiene tres operandos en registros, debemos usar el formato R. Usaremos la siguiente distribución de campos:



Es importante colocar el registro *c* en el campo *rt* (o, alternativamente, en el *rs*) para poder leerlo en el ciclo 2 sin tener que modificar el comportamiento del procesador en dicho ciclo.

2. División del trabajo en ciclos:

Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4
IR ← Memoria[PC]	A ← \$a	A ← \$a	si A = B entonces PC ← MDR
PC ← PC + 4	B ← \$c	B ← \$b	
		MDR ← Memoria[B]	

Nótese que la comparación de A y B y la escritura condicional en PC se puede realizar en el mismo ciclo, de forma análoga a como se hace en el ciclo 3 de la instrucción *beq*. La escritura condicional se implementará usando la señal *cer0* de la ALU y la señal de control *EscrPCCond*.

3. Extensión del camino de datos:

No podemos realizar algunas de las acciones especificadas en la tabla anterior con el camino de datos visto en clase. Por tanto, tendremos que extenderlo de la siguiente manera:

- Para $B \leftarrow \$b$: Añadir un multiplexor en la entrada «Reg. de lectura 2». Tendrá dos entradas y estará controlado por una nueva señal *RegOri2* con los siguientes significados:
 - *RegOri2* = 0: El registro que se lee y se almacena en B viene dado por el campo *rt* (como en el camino no modificado).
 - *RegOri2* = 1: El registro que se lee y se almacena en B viene dado por el campo *rd*.*RegOri2* tomará el valor 0 en los estados preexistentes del autómata de control.
- Para $MDR \leftarrow Memoria[B]$: Añadir una entrada nueva al multiplexor controlado por la señal de control *IoD* que esté conectada a la salida del registro B. La señal *IoD* pasa a tener 2 bits de ancho y su significado será:
 - *IoD* = 00: El PC suministra la dirección para acceder a memoria.
 - *IoD* = 01: ALUOut proporciona la dirección para acceder a memoria.
 - *IoD* = 10: El registro B proporciona la dirección para acceder a memoria.
- Para $si\ A = B\ entonces\ PC \leftarrow MDR$: Conectar la salida del registro MDR con la entrada libre disponible en el multiplexor controlado por la señal *FuentePC*. El significado de esta señal según su valor será ahora:
 - *FuentePC* = 00: La salida de la ALU se envía para su escritura en el PC.
 - *FuentePC* = 01: El contenido de ALUOut se envía para su escritura en el PC.
 - *FuentePC* = 10: El campo *j* desplazado dos bits a la izquierda y concatenado con los 4 bits más significativos del registro PC se envía para su escritura en el PC.
 - *FuentePC* = 11: El contenido de MDR se envía para su escritura en el PC.

4. Extensión del control:

Habrá que añadir 2 nuevos estados al autómata de control original. El resultado se muestra en la figura 3.

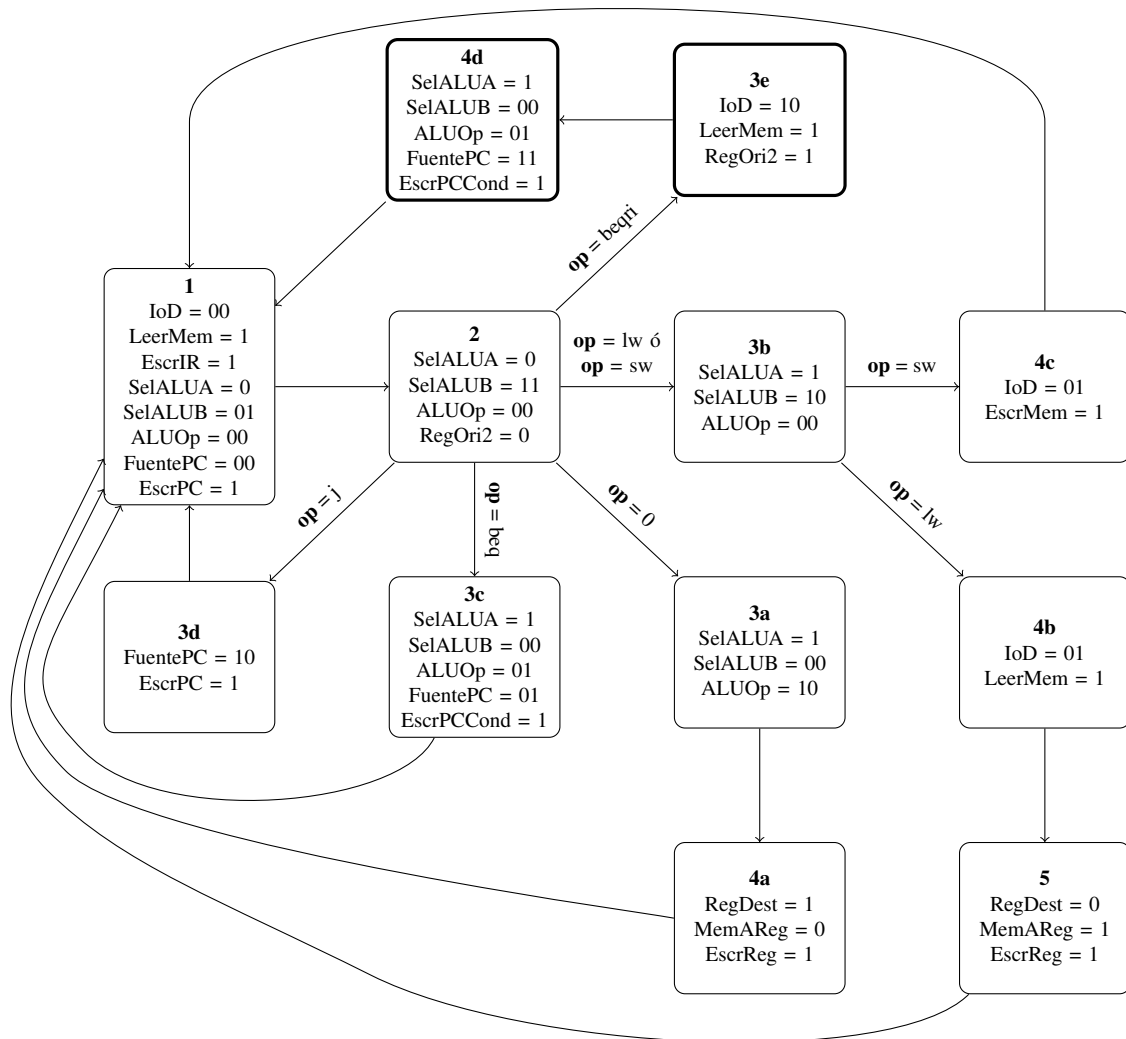
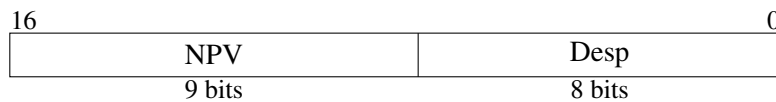


Figura 3: Autómata de control modificado para incluir la instrucción *beqri*. Los estados añadidos han sido resaltados.

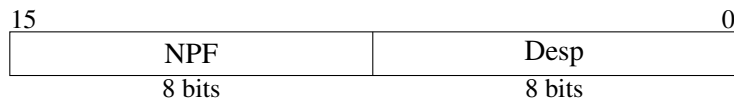
Ejercicio 3

- a) Si la tabla de páginas tiene 512 entradas, eso es porque hay 512 páginas virtuales, luego el campo NPV de la dirección virtual necesitará $\log_2 512 = 9$ bits.

Como las páginas tienen un tamaño de 256 bytes, eso significa que el campo de desplazamiento de la dirección virtual necesita $\log_2 256 = 8$ bits. La dirección virtual, por lo tanto, tiene un tamaño total de $9 + 8 = 17$ bits y queda formada de la siguiente manera:



En el caso de la dirección física, sabemos que la memoria principal tiene un tamaño total de 64 KB, por lo que la dirección física tiene un tamaño total de $\log_2 64 \times 2^{10} = 16$ bits. De esos, 8 bits constituyen el desplazamiento y el resto (otros 8 bits) el NPF. La dirección física queda, pues, de la siguiente manera:



Cada entrada de la tabla de páginas necesitará un bit de modificación (M) para permitir la política de postescritura y un bit de uso (U) para permitir la política de reemplazo LRU, además del bit de validez (V). La tabla de páginas queda así:

	V	M	U	NPF (8 bits)
0	1	0	0	8
1	1	0	0	7
2	1	0	0	6
3	1	0	0	5
4	1	0	0	4
5	1	0	0	3
6	1	0	0	2
7	1	0	0	1
8	0	–	–	–
9	0	–	–	–
10	0	–	–	–
...
511	0	–	–	–

Cada entrada de la tabla de páginas tiene (al menos) 11 bits, por lo que en total se necesitarán $512 \times 11 = 5632$ bits. Si se redondea el tamaño de cada entrada a 16 bits (que es el tamaño de palabra según otro apartado del problema) se necesitarán en total $512 \times 2 = 1024$ bytes = 1 KB.

- b) Debido a que el TLB tiene 16 conjuntos, los 4 bits de menor peso del NPV se utilizarán para elegir la posición en la caché, mientras que los 5 restantes formarán la etiqueta.

Cada conjunto tiene 2 vías. Cada vía necesitará un bit de modificación (M) para permitir la política de postescritura y un bit de validez (V) para saber cuándo la vía contiene información útil. No es necesario un bit de uso (U) porque el TLB sigue una política de reemplazo aleatorio. Además de esos bits, cada vía contendrá también los bits M y U de la entrada correspondiente de la tabla de páginas (que mostraremos como M' y U', respectivamente) y el campo NPF con el correspondiente número de página física. El TLB queda, pues, de la siguiente forma:

		Vía 0					Vía 1				
		V	M	Etiqueta (5 bits)	M'	U'	NPF (8 bits)				
0	0	-	-	-	-	-	-				
1	0	-	-	-	-	-	-				
2	0	-	-	-	-	-	-				
...				
15	0	-	-	-	-	-	-				

Por tanto, el tamaño total del TLB será de $16 \times 2 \times (1 + 1 + 5 + 1 + 1 + 8) = 544$ bits.

- c) • Lectura de la dirección $25 = 0x19$.

NPV: 0.

Desplazamiento: $0x19$.

Acierto/fallo de TLB: fallo (el TLB está inicialmente vacío).

Acierto/fallo de página: acierto (según el bit de validez de la entrada 0 de la tabla de páginas).

Conjunto del TLB: 0, y queda de la siguiente manera:

		Vía 0					Vía 1				
		V	M	Etiqueta (5 bits)	M'	U'	NPF (8 bits)				
0	1	1	1	0	0	1	8				
	0	-	-	-	-	-	-				

Ya que el bit de uso U' se pone a 1, el bit M del TLB debe ponerse también a 1.

NPV: 8.

Dirección física resultante: $0x819$.

Entrada de la tabla de páginas: no cambia.

- Escritura en la dirección $904 = 0x388$.

NPV: 3.

Desplazamiento: $0x88$.

Acierto/fallo de TLB: fallo, ya que el conjunto 3 del TLB está vacío.

Acierto/fallo de página: acierto (según el bit de validez de la entrada 3 de la tabla de páginas).

Conjunto del TLB: 3, y queda de la siguiente manera:

		Vía 0					Vía 1				
		V	M	Etiqueta (5 bits)	M'	U'	NPF (8 bits)				
3	1	1	1	0	1	1	5				
	0	-	-	-	-	-	-				

Ya que el bit de uso U' y el bit de modificación M' se ponen a 1, el bit M del TLB debe ponerse también a 1.

NPV: 5.

Dirección física resultante: $0x588$.

Entrada de la tabla de páginas: no cambia.

- Lectura de la dirección $75 = 0x4B$.

NPV: 0.

Desplazamiento: $0x4B$.

Acierto/fallo de TLB: acierto, ya que el conjunto 0 del TLB tiene la vía 0 ocupada y las etiquetas coinciden.

Conjunto del TLB: no cambia.

NPV: 8.

Dirección física resultante: 0x84B.

Entrada de la tabla de páginas: no cambia.

- d) Si cada bloque de la caché tiene 4 palabras y las palabras son de 2 bytes (16 bits), el tamaño de cada bloques es de $4 \times 2 = 8$ bytes. Como la caché tiene 64 bytes en total, eso significa que hay $64/8 = 8$ bloques, luego son necesarios $\log_2 8 = 3$ bits para el campo del índice. Al ser las palabras de 2 bytes, hacen falta $\log_2 2 = 1$ bit para el desplazamiento de byte (DB) y, al ser cada bloque de 4 palabras, hacen falta $\log_2 4 = 2$ bits para el desplazamiento de palabra (DP).

Por otro lado, debemos tener en cuenta que la dirección física es de 16 bits (ver apartado a). Por lo tanto, hacen faltan $16 - 3 - 2 - 1 = 10$ bits para el campo de la etiqueta. Todo esto hace que una dirección física se descomponga de la siguiente manera para acceder a la caché:

15	0
Etiqueta	Índice DP DB
10 bits	3 bits 2 bits 1 bit

La caché queda de la siguiente manera:

	V	M	Etiqueta (10 bits)	Bloque (4x16=64 bits)			
0	0	–	–	–	–	–	–
1	0	–	–	–	–	–	–
2	0	–	–	–	–	–	–
3	0	–	–	–	–	–	–
4	0	–	–	–	–	–	–
5	0	–	–	–	–	–	–
6	0	–	–	–	–	–	–
7	0	–	–	–	–	–	–

El bit de validez (V) indica si la entrada contiene datos válidos o no mientras que el bit de modificación (M) se ha incluido al seguir la caché una política de postescritura para saber si una determinada entrada ha sido modificada o no.

El tamaño total de la caché es, pues, de $8 \times (1 + 1 + 10 + 64) = 608$ bits.

Ejercicio 4

- a) Como generamos una interrupción cada 4 bytes transmitidos, el número de interrupciones/seg = $15 \times 2^{20} \text{ bytes/seg} / 4 \text{ bytes/inter} = 3,932,160$ interrupciones/seg.
 Ciclos consumidos/seg = $3,932,160 \text{ interrupciones/seg} \times 400 \text{ ciclos/interrupción} = 1,572,864,000$ ciclos/seg.
 % ocupación CPU = $1,572,864,000 / 3 \times 10^9 = 52,4\%$.
- b) Número de operaciones DMA/seg = $15 \times 2^{20} \text{ bytes/seg} / 4 \times 2^{10} \text{ bytes/op.DMA} = 3,75 \times 2^{10} = 3840$ op.DMA/seg.
 Ciclos consumidos/seg = $3840 \text{ op.DMA/seg} \times 12000 \text{ ciclos/op.DMA} = 46,080,000$ ciclos/seg.
 % ocupación CPU = $46,080,000 / 3 \times 10^9 = 1,5\%$.
- c) Para soportar una transferencia de 15 MBytes/seg mediante sondeo necesitamos:
 $15 \times 2^{20} \text{ bytes/seg} / 4 \text{ bytes/encuesta} = 3,932,160$ encuestas/seg.
 Como cada encuesta consume 1000 ciclos tenemos que:

$3,932,160 \text{ encuestas/seg} \times 1000 \text{ ciclos/encuesta} = 3,932,160,000 \text{ ciclos/seg} = 3,93 \text{ GHz}$.

Ésa es la frecuencia mínima (3,93 GHz) y tendríamos un 100 % de ocupación de la CPU.

d) Tamaño 1 imagen = $10 \times 10^6 \text{ píxeles} \times 3 \text{ bytes/píxel} = 30 \times 10^6 \text{ bytes}$.

Tiempo transmisión datos = $30 \times 10^6 \text{ bytes} / 15 \times 2^{20} \text{ bytes/seg} = 1,90 \text{ segundos}$.

Cuando usamos DMA, el tiempo total debe incluir los ciclos de preparación y terminación. El número de operaciones DMA para transmitir $30 \times 10^6 \text{ bytes}$, a razón de 4KB por operación DMA es: $30 \times 10^6 / 4096 = 7324,2 \text{ op.DMA} \rightarrow$ redondeando al siguiente entero: 7325 op.DMA.

Tiempo preparación y terminación = $12000 \text{ ciclos/op.DMA} \times 7325 \text{ op.DMA} / 3 \times 10^9 \text{ ciclos/seg} = 0,0293 \text{ segundos}$.

Tiempo total = $1,90 \text{ segundos} + 0,0293 \text{ segundos} = 1,9293 \text{ segundos}$.