

ESTRUCTURA Y TECNOLOGÍA DE COMPUTADORES

I.T.I. Gestión, I.T.I. Sistemas e I. Superior
3^{er} parcial, 2 de junio de 2009

Apellidos:	Nombre:
D.N.I.:	
Grupo: Sistemas I <input type="checkbox"/> Sistemas II <input type="checkbox"/> Gestión I <input type="checkbox"/> Gestión II <input type="checkbox"/> Superior <input type="checkbox"/>	

PARTE II: CUESTIONES TEÓRICO-PRÁCTICAS

- 1) (3 puntos) La máquina M posee un sistema de memoria con las siguientes características:
- Tabla de páginas de 256 entradas
 - Memoria física de 256 bytes
 - Tamaño página virtual de 16 bytes
 - TLB asociativo por conjuntos de 2 vías con una estrategia de reemplazo aleatoria y una política de postescritura.
 - El TLB posee 2 conjuntos
 - Memoria caché de correspondencia directa combinada para datos e instrucciones con capacidad para 16 palabras (palabras de 32 bits), bloques de 2 palabras y política de postescritura.

```
        .data
        ...
almacen: .space 8
        ...
#####SEGMENTO DE TEXTO#####
        .text
        ...
        li    $s0, 2
        li    $s1, 4
        la    $s2, almacen
bucle:   lb    $t0, 0($s2)
        addi  $s2, $s2, 4
        addi  $s0, $s0, -1
        bne  $s0, $zero, bucle
        ...
```

Se pide:

- (0'5 puntos) 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).
- (0'5 puntos) Dibujar un esquema detallado del TLB y de la caché. Calcular sus tamaños incluyendo todos los bits de control necesarios (justificando su inclusión).
- (1'50 puntos) Hacer un seguimiento del estado del TLB, tabla de páginas y caché en la ejecución del programa anterior.
Tener en cuenta los siguientes datos:

- Dirección de comienzo del “almacen” → $(0)_{10}$
- Dirección de comienzo instrucciones → $(80)_{10}$
- Los bits de validez de la tabla de páginas están puestos a 1 en las posiciones 0, y 5, siendo los números de página física que ocupan esas posiciones $(12)_{10}$ y $(0)_{10}$ respectivamente.
- Sabemos que durante la ejecución de este código, en memoria principal está libre la página física $(2)_{10}$.
- Inicialmente caché y TLB están vacíos.

- d) (0'5 puntos) Calcular la tasa de fallos de instrucciones y datos de la caché en la ejecución del programa anterior. Suponiendo que el procesador está dedicado en exclusiva a la ejecución de nuestra aplicación y que son 20 los ciclos necesarios para acceder a memoria principal y 2000 los necesarios para acceder a memoria secundaria, ¿cuántos ciclos estará parado el procesador sin realizar ninguna tarea debido a los distintos fallos de la caché, el TLB y la tabla de páginas? (En los 2000 ciclos de acceso a memoria secundaria se incluye el tiempo necesario para llevar la página desde memoria secundaria a memoria principal y la posterior actualización de la tabla de páginas y del TLB).

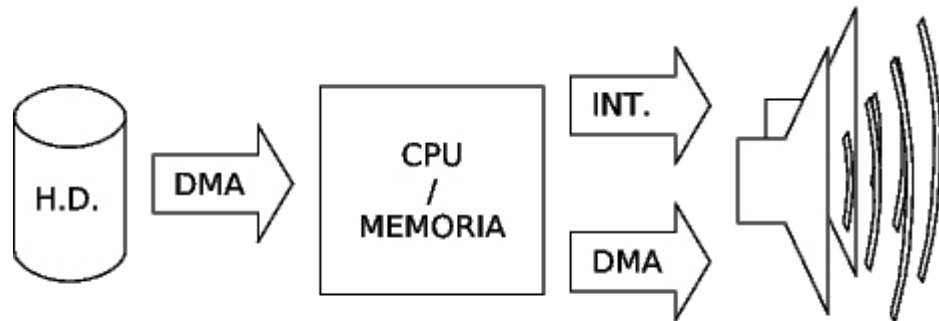
- 2) (3 puntos) Un sistema dispone de un procesador Core 2 Duo a 2,66 Ghz y de una tarjeta de sonido capaz de generar sonido con calidad de CD, es decir, 44.100 muestras por segundo, donde en cada muestra se envían 32 bits (16 bits por cada uno de los dos canales de sonido que forman una comunicación en estéreo). Se pretende usar este sistema para reproducir ficheros de sonido en formato WAV. Para ello, se leen muestras de sonido del disco duro y se envían a la tarjeta de sonido siguiendo las siguientes 3 etapas:
- Etapa 1: se leen muestras de sonido en bloques de 4 KB del disco duro. Para ello, se usa un controlador DMA que requiere 4000 ciclos de inicialización y 2000 ciclos al terminar la transferencia.
 - Etapa 2: procesamiento de los datos en la CPU. Para el formato WAV supondremos que el procesamiento requiere 0 ciclos de CPU ya que la tarjeta de sonido es capaz de reproducir directamente datos en dicho formato.
 - Etapa 3: para reproducir el sonido, el procedimiento para enviar las muestras a la tarjeta de sonido puede ser por unos de estos dos métodos diferentes:
 1. Por interrupciones (44.100 interrupciones por segundo), donde en cada interrupción, que requiere 200 ciclos de CPU, se envía a la tarjeta una muestra de 32 bits.
 2. Por DMA, donde cada transferencia de 8 KB requiere un procesamiento de 4000 ciclos de inicialización más 2000 ciclos al terminar.

Se pide:

- (1 punto) Calcular el % de CPU consumido en esta tarea (las 3 etapas), usando interrupciones en la etapa 3.
- (0'5 puntos) Calcular el % de CPU consumido en esta tarea (las 3 etapas), usando DMA en la etapa 3.

- c) (1 punto) Calcular el % de CPU consumido en esta tarea (las 3 etapas), si se reproducen ficheros MP3 (formato que comprime WAV a un 10% del tamaño original), sabiendo que el procesamiento para descomprimir los datos en la CPU de un bloque de 4 KB de MP3 requiere 40 millones de ciclos en la etapa 2. Supondremos que en la etapa 3 se sigue usando DMA.
- d) (0'5 puntos) Supongamos que en la etapa 3 los datos se envían a la tarjeta de sonido a través de un bus con líneas dedicadas para datos y direcciones, con 32 bits para datos y otros 32 bits para direcciones. Si enviar una dirección requiere un ciclo de bus, la tarjeta de sonido tiene una latencia de 2 ciclos y el bus soporta un modo ráfaga capaz de enviar 32 bytes por ráfaga, a razón de 4 bytes por ciclo, ¿con qué frecuencia mínima debe funcionar el bus para reproducir el sonido correctamente?

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



PROBLEMA 1

a)

256 páginas virtuales = $2^8 \rightarrow$ 8 bits para el número de página virtual
 Memoria física de 256 Bytes = $2^8 \rightarrow$ 8 bits para la dirección física
 Tamaño de página de 16 bytes = $2^4 \rightarrow$ 4 bits para el desplazamiento

Con todo ello tenemos:

Dirección virtual: 12 bits

Número página virtual: 8 bits	Desplazamiento: 4 bits
-------------------------------	------------------------

Dirección física: 8 bits

Número página física: 4 bits	Desplazamiento: 4 bits
------------------------------	------------------------

Tabla de páginas:

Tamaño	1			4 bits
	V	M	U	Número Página Física
0	0	0	0	-
1	0	0	0	-
2	0	0	0	-
3	0	0	0	-
4	0	0	0	-
5	0	0	0	-
6	0	0	0	-
7	0	0	0	-
8	0	0	0	-
...	0	0	0	-
2^8-2	0	0	0	-
2^8-1	0	0	0	-

Tamaño = $2^8 * (3 + 4)$ bits = **1792 Bits**

Tamaño (ajustando el tamaño a 1 byte por cada entrada) = $2^8 * 1$ byte = **256 Bytes**

b)

	V	M	Etí	M'	U'	NPF		V	M	Etí	M'	U'	NPF
0	1	1	7	1	1	4	0	1	1	7	1	1	4
1							1						

Tenemos 2 conjuntos. Por lo tanto, de los 8 bits que conforman el NPV, necesitamos para el índice 1 bit y nos quedan 7 bits para la etiqueta.

Tamaño = $2 * 2 * (2 + 7 + 2 + 4)$ bits = **60 bits**

Caché

	V	M	TAG	DATA
C	1	1	2	64
1				
2				
3				
4				
5				
6				
7				

Si la caché tiene capacidad de 16 palabras y cada bloque tiene 2 palabras, resultan un total de 8 bloques. Además tenemos un bit de validez que indica si el contenido de un bloque de la caché es válido y un bit de modificación ya que se usa una política de postescritura. No hace falta el bit de uso ya que es una caché de correspondencia directa. En cuanto a la etiqueta, tendrá un tamaño de 2 bits, ya que los ocho bits de la dirección física se reparten de la siguiente manera:

- Los dos bits menos significativos → Byte offset
- El bit siguiente → Word offset
- Los tres bits siguientes → dirección del bloque
- Y los dos bits restantes (más significativos) → etiqueta.

00 → Tag 000 → Dirección de bloque 0 → Word offset 00 → Byte offset

Tamaño = $8 * (2 + 2 + (2 * 32))$ bits = **544 bits → 68 bytes**

C)

Para comenzar con la ejecución de la aplicación, el procesador busca la primera instrucción. Esto lo hace a través de su dirección virtual que es:

D.V. = $(80)_{10} \rightarrow 00000101\ 0000$
 NPV (8 bits más significativos) → 00000101
 Page Offset (4 bits menos significativos) → 0000

Con esta dirección, la MMU intentará traducir el número de página virtual en un número de página física usando el TLB. Como sabemos, al principio el TLB está vacío, por lo que se producirá un fallo y tendremos que acceder a la tabla de páginas.

La situación inicial de la tabla de páginas es:

	V	M	U	Número Página Física
0	1	0	0	$(12)_{10}$
1	0	0	0	-
2	0	0	0	-
3	0	0	0	-
4	0	0	0	-
5	1	0	0	$(0)_{10}$
...
255	0	0	0	-

En esta tabla de páginas tendremos que comprobar si el bit de validez de la posición apuntada por número de página virtual está puesto a 1, es decir, que la página está en memoria principal.

Como vimos en el apartado a) el NPV está formado por los 8 bits más significativos de la dirección virtual, por lo que la posición que tenemos que comprobar es:

$$\text{NPV} = 00000101 \rightarrow \text{posición 5}$$

Como se puede observar, en esa posición el bit de validez está puesto a 1, por lo que tenemos un acierto y la página se encuentra en memoria principal.

El siguiente paso es actualizar el TLB, quedando de la siguiente forma:

	V	M	Etí	M'	U'	NPF
0	0					
1	1	1	(2) ₁₀	0	1	(0) ₁₀

	V	M	Etí	M'	U'	NPF
0	0					
1	0					

Ahora ya podemos hallar la dirección física. Esta sería:

$$\text{D.F.} \rightarrow 0000\ 0000 = (0)_{10}$$

$$\text{NPF (cuatro bits más significativos)} \rightarrow 0000$$

$$\text{Page Offset (cuatro bits menos significativos)} \rightarrow 0000$$

El NPF es el valor indicado en la posición 5 de la tabla de páginas, codificado con 4 bits.

El page offset son los cuatro bits menos significativos de la dirección virtual. Recuerdo que el page offset son los mismos bits tanto en la dirección virtual como en la dirección física.

Una vez que tenemos la dirección física, accedemos a caché en busca de nuestra instrucción. Para ello tenemos que comprobar el bit de validez del bloque correspondiente.

$$\text{D.F.} \rightarrow 00\ 000\ 000, \text{ es decir, el bloque de dirección } (0)_{10}.$$

Como sabemos la caché está en un principio vacía, por lo que se producirá un fallo de caché que habrá que resolver.

Pasado un tiempo, el sistema ha accedido a memoria principal y nos ha traído el bloque de caché que buscábamos, quedando la caché de la siguiente forma:

	V	M	TAG	DATA
(1	0	00	li \$s0, 2 li \$s1, 4
1				
2				
3				
4				
5				
6				
7				

El valor del tag es 00 ya que la etiqueta está formada por los dos bits más significativos de la dirección física. El bit de modificación vale 0 ya que sólo vamos a realizar una lectura.

Una vez que la instrucción está en la caché se puede proceder a su ejecución.

INSTRUCCIÓN 2

Ahora pasamos a la ejecución de la siguiente instrucción cuya dirección virtual es (84)₁₀, es decir, D.V. \rightarrow 00000101 0100.

$$\text{NPV} \rightarrow 00000101 \rightarrow (5)_{10}.$$

Para esta instrucción tenemos acierto de TLB, por lo que se puede hacer la traducción sin tener que acceder a la tabla de páginas.

$$DF \rightarrow 0000\ 0100 \rightarrow (4)_{10}$$

Con esta dirección accedemos a la caché y se produce un acierto ya que en el bloque 0 (00000100) el bit de validez está puesto a 1 y además las etiquetas coinciden. La diferencia es que ahora accedemos a la palabra situada en la posición 1 indicada por el Word offset (00000 1 00).

INSTRUCCIÓN 3

Pasemos ahora a la tercera instrucción. La dirección de ésta es:

$$D.V. = (88)_{10} \rightarrow \text{En binario} = 00000101\ 1000$$

Como podemos observar el NPV sigue siendo el mismo que en el caso anterior, es decir, que cuando accedemos al TLB tenemos un acierto, por lo que podemos hacer la traducción de DV \rightarrow DF, quedando esta última:

$$DF \rightarrow 0000\ 1000 \rightarrow (8)_{10}$$

Una vez que tenemos la dirección física, accedemos a caché en busca de nuestra instrucción. Para ello tenemos que comprobar el bit de validez del bloque correspondiente. D.F. \rightarrow 00 001 000, es decir, el bloque de dirección (1)₁₀.

El bit de validez de este bloque está puesto a 0, por lo que tenemos un fallo de caché.

Pasado un tiempo, el sistema ha accedido a memoria principal y nos ha traído a caché el bloque que buscábamos, quedando ésta de la siguiente forma:

	V	M	TAG	DATA
c	1	0	00	li \$s0, 2 li \$s1, 4
1	1	0	00	la \$s2, almacen lb \$t0, 0(\$s2)
2				
3				
4				
5				
6				
7				

Una vez que la instrucción está en la caché, ésta puede pasar al procesador para su ejecución.

INSTRUCCIÓN 4

Ahora pasamos a la ejecución de la siguiente instrucción cuya dirección virtual es (92)₁₀, es decir, D.V. \rightarrow 000001011100.

$$NPV \rightarrow 00000101 \rightarrow (5)_{10}$$

Para esta instrucción tenemos acierto de TLB, por lo que se puede hacer la traducción sin tener que acceder a la tabla de páginas.

$$DF \rightarrow 0000\ 1100 \rightarrow (12)_{10}$$

Con esta dirección accedemos a la caché y se produce un acierto ya que en el bloque 1 el bit de validez está puesto a 1, y además las etiquetas coinciden. La diferencia es que ahora accedemos a la palabra situada en la posición 1 indicada por el Word offset (00000 1 00).

Por otro lado, esta última instrucción es una instrucción de carga, es decir, que hay que hacer un nuevo acceso a memoria, concretamente a la dirección donde se sitúa “almacen”, es decir, a la dirección virtual $(0)_{10}$,

$$DV \rightarrow 00000000\ 0000$$

$$NPV \rightarrow 00000000 \rightarrow (0)_{10}$$

Según el NPV, tenemos que acceder a la posición 0 del TLB (000000 0). Pero los dos bits de validez del conjunto 0 están puesto a 0, por lo que tendremos un fallo de TLB y se tendrá que acceder a la tabla de páginas.

En la tabla de páginas tenemos que comprobar la posición 0 ya que el NPV vale $(0)_{10}$.

Afortunadamente, el bit de validez en esta entrada de la tabla de páginas está puesto a 1, indicando que la página está en memoria. Por lo que se produce un acierto de tabla de páginas.

A continuación pasamos a actualizar el TLB, quedando este de la siguiente manera:

	V	M	Etí	M'	U'	NPF
0	1	1	$(0)_{10}$	0	1	$(12)_{10}$
1	1	1	$(2)_{10}$	0	1	$(0)_{10}$

	V	M	Etí	M'	U'	NPF
0	0					
1	0					

Una vez actualizado el TLB, pasamos a hacer la traducción del NPV a NPF para hallar la dirección física, quedando ésta de la siguiente forma:

$$DF \rightarrow 1100\ 0000 \rightarrow (192)_{10}$$

Con esta dirección física se acceder a caché, concretamente se comprueba el bloque 0 (11 000 000). En esta posición el bit de validez está puesto a 1, por lo que ahora tenemos que comprobar las etiquetas para ver si tenemos un acierto.

Las etiquetas no coinciden, pues la etiqueta de caché es 00 y la obtenida en la nueva dirección es 11. Tenemos, pues, un fallo de caché, como el bit de modificación está puesto a 0, bastará con machacar la posición 0 con los nuevos datos y actualizar el campo etiqueta. La caché quedaría así:

	V	M	TAG	DATA
0	1	0	11	D(192) D(196)
1	1	0	00	la \$s2, almacen lb \$t0, 0(\$s2)
2				
3				
4				
5				
6				
7				

INSTRUCCIÓN 5

La dirección de la nueva instrucción es:

$$D.V. = (96)_{10} \rightarrow \text{En binario} = 00000110\ 0000$$

$$NPV \rightarrow 00000110 \rightarrow (6)_{10}$$

Con esta dirección la MMU intenta hallar el número de página física accediendo al TLB. Se comprueba el conjunto 0 del TLB, ya que el bit menos significativo del NPV vale 0.

En una de las posiciones de este conjunto el bit de validez está puesto a 1 y en la otra a 0, por lo que pasamos a comprobar las etiquetas para la posición con bit de validez puesto a 1.

Como podemos comprobar estas no coinciden $(0)_{10} \neq (3)_{10}$. Por lo que se produce un fallo de TLB y tenemos que acceder a la tabla de páginas, concretamente tenemos que comprobar la entrada $(6)_{10}$. Como vemos, esta entrada está puesto el bit de validez a 0, por lo que se produce

un fallo de tabla de páginas, y tendrá que ser el sistema operativo el que nos resuelva este problema trayendo una página desde memoria secundaria a memoria principal. Para hacer esto, usará la página física que sabemos que se encuentra disponible, es decir, la página física (2)₁₀. Quedando por tanto la tabla de páginas y el TLB de la siguiente forma:

	V	M	U	Número Página Física
0	1	0	0	(12) ₁₀
1	0	0	0	-
2	0	0	0	-
3	0	0	0	-
4	0	0	0	-
5	1	0	0	(0) ₁₀
6	1	0	0	(2)₁₀
...
255	0	0	0	-

	V	M	Etí	M'	U'	NPF
0	1	1	(0) ₁₀	0	1	(12) ₁₀
1	1	1	(2) ₁₀	0	1	(0) ₁₀

	V	M	Etí	M'	U'	NPF
0	1	1	(3)₁₀	0	1	(2)₁₀
1	0					

Ahora ya podemos pasar a hallar la dirección física. Esta sería:

$$D.F. \rightarrow 0010\ 0000 = (32)_{10}$$

Donde los 4 bits menos significativos serían el page offset que se obtiene de la D.V. y el 4 bits más significativos serían los recuperados del TLB.

Una vez que tenemos la dirección física, accedemos a caché en busca de nuestra instrucción. Para ello tenemos que comprobar el bit de validez del bloque correspondiente. D.F. \rightarrow 00 **100** 000, es decir, el bloque de dirección (4)₁₀.

Este bit se encuentra a 0, por lo que se produce un fallo de caché y habrá que acceder a memoria.

Pasado un tiempo, el sistema ha accedido a memoria principal y nos ha traído el bloque de caché que buscábamos, quedando la caché de la siguiente forma:

	V	M	TAG	DATA
0	1	0	11	D(192) D(196)
1	1	0	00	la \$s2, almacen lb \$t0, 0(\$s2)
2				
3				
4	1	0	00	addi \$s2, \$s2, 4 addi \$s0, \$s0, -1
5				
6				
7				

Una vez la instrucción en la caché se puede a proceder a su ejecución.

INSTRUCCIÓN 6

Terminada la instrucción anterior, se pasa a la ejecución de la siguiente, cuya dirección virtual es $(100)_{10}$, es decir, D.V. $\rightarrow 00000110\ 0100$.

$$\text{NPV} \rightarrow 00000110 \rightarrow (6)_{10}$$

Para esta instrucción tenemos acierto de TLB ya que en el conjunto 0 tenemos una entrada donde las etiquetas coinciden (tag = 0000011), por lo que se puede hacer la traducción sin tener que acceder a la tabla de páginas.

$$\text{DF} \rightarrow 0010\ 0100 \rightarrow (36)_{10}$$

Con esta dirección accedemos a la caché y se produce un acierto ya que en el bloque 4 (00**100**100) el bit de validez está puesto a 1 y además las etiquetas coinciden. La diferencia es que ahora accedemos a la palabra situada en la posición 1 indicada por el Word offset (00100 **1**00).

INSTRUCCIÓN 7

La nueva instrucción tiene la dirección:

$$\text{D.V.} = (104)_{10} \rightarrow 00000110\ 1000$$

Como podemos observar el NPV sigue siendo el mismo que el de la instrucción anterior, por lo que al acceder al TLB tenemos un acierto.

La traducción de DV \rightarrow DF, queda de la siguiente forma:

$$\text{DF} \rightarrow 0010\ 1000 \rightarrow (40)_{10}$$

Una vez que tenemos la dirección física, accedemos a caché en busca de nuestra instrucción. Para ello tenemos que comprobar el bit de validez del bloque correspondiente.

D.F. $\rightarrow 00\ 101\ 000$, es decir, el bloque de dirección $(5)_{10}$.

El bit de validez de este bloque está puesto a 0, por lo que tenemos un fallo de caché.

Pasado un tiempo, el sistema ha accedido a memoria principal y nos ha traído el bloque de caché que buscábamos, quedando la caché de la siguiente forma:

	V	M	TAG	DATA
0	1	0	11	D(192) D(196)
1	1	0	00	la \$s2, almacen lb \$t0, 0(\$s2)
2				
3				
4	1	0	00	addi \$s2, \$s2, 4 addi \$s0, \$s0, -1
5	1	0	00	bne \$s0, \$zero, bucle
6				...
7				

Una vez la instrucción en la caché se puede proceder a su ejecución.

Al ser ésta una instrucción de salto cuya condición no se cumple, volvemos a hacer otra pasada sobre las instrucciones que componen el bucle.

SEGUNDA PASADA DEL BUCLE

Como podemos observar, la caché contiene todas las instrucciones y datos que vamos a necesitar en esta segunda y última pasada, por lo que no se producirían más cambios en la caché, en el TLB o en la tabla de páginas. (Realmente si se podrían producir cambios en el TLB ya que se puede producir actualizaciones en los bits de uso).

NOTA: en la segunda carga, la dirección virtual que se busca es la $(4)_{10}$

DV \rightarrow 00000000 0100

Habría acierto en el TLB pues en el conjunto 0 tenemos la etiqueta $(0)_{10}$, devolviéndonos éste el NPF \rightarrow $(12)_{10}$, que junto al page offset, obtendríamos la siguiente dirección física:

DF \rightarrow 1100 0100

Es decir, que tenemos que acceder al bloque $(0)_{10}$ de caché y comparar con la etiqueta 11, lo cual produce un acierto.

d)

Tasa de fallos de instrucciones = nº de fallos de instr./ nº total de instrucciones ejecutadas

$$\text{Tasa fallos} = 4 / 11 = 0,364 \rightarrow 36,4\%$$

Tasa de fallos de datos = nº de fallos peticiones de datos/ nº total de accesos a datos

$$\text{Tasa fallos} = 1 / 2 = 0,5 \rightarrow 50\%$$

Ciclos de espera del procesador = nº accesos a MP * 20 + nº accesos a MS * 2000

Ciclos de espera = (5 fallos de caché + 3 accesos a tabla de páginas) * 20 + 1 * 2000 = 2160
ciclos.

PROBLEMA 2

a)

Para reproducir el sonido con la calidad deseada, necesitamos enviar 44100 muestras por segundo y 32 bits por muestra, lo que nos da una tasa de transferencia de:

$$44100 \text{ muestras/seg} \cdot 4 \text{ bytes/muestra} = 176400 \text{ bytes/segundo}$$

Con esta tasa de transferencia, vamos a ver qué porcentaje de CPU necesitamos en cada etapa:

Etapa 1:

Necesitamos realizar $176400 / 4096 = 43,07$ transferencias de DMA por segundo.

Puesto que cada transferencia requiere $(4000 + 2000) = 6000$ ciclos de CPU, en total necesitamos $43,07 * 6000 = 258398,4$ ciclos/segundo.

Etapa 2:

Cuando se reproducen ficheros WAV, esta etapa no consume ciclos de CPU.

Etapa 3:

Se producen 44100 interrupciones por segundo, donde cada una necesita 200 ciclos de CPU.

Por tanto, en esta etapa necesitamos $44100 * 200 = 8820000$ ciclos.

Luego, el porcentaje de CPU que necesitamos en las tres etapas es:

$$\% \text{ CPU} = \frac{258398,4 + 0 + 8820000}{2,66 * 10^9} * 100 = 0,34 \%$$

b)

Este caso es similar al anterior, la diferencia está en que en la tercera etapa usamos DMA. Puesto que necesitamos seguir enviando 176400 bytes/segundo a la tarjeta de sonido, tendremos que realizar:

$$\frac{176400}{8192} = 21,53 \text{ transferencias de DMA por segundo}$$

Puesto que cada transferencia requiere $(4000 + 2000) = 6000$ ciclos de CPU, en total necesitamos $21,53 * 6000 = 129199,2$ ciclos/segundo.

Luego, el porcentaje de CPU que necesitamos en las tres etapas ahora es:

$$\% \text{ CPU} = \frac{258398,4 + 0 + 129199,2}{2,66 * 10^9} * 100 = 0,015 \%$$

c)

El utilizar ficheros MP3 cambia las etapas 1 y 2, pero no la etapa 3, ya que los datos se tienen que seguir enviando a la tarjeta de sonido a la misma velocidad y en el mismo formato.

En la etapa 1, al leer ficheros MP3, es decir, al leer la información comprimida, necesitamos leer a una velocidad 10 veces menor, lo que supone 10 veces menos transferencias de DMA y, por tanto, 10 veces menos ciclos de CPU por segundo.

Ahora necesitaremos $258398,4 / 10 = 25839,84$ ciclos/segundo.

En la etapa 2 necesitamos consumir 40 millones de ciclos de CPU para descomprimir un bloque MP3 de 4 KB. Ya hemos dicho que al leer la información comprimida, necesitamos una tasa de transferencia 10 veces menor. Por tanto, la tasa de transferencia ahora es $176400 / 10 = 17640$

bytes/segundo, lo que supone descomprimir $17640 / 4096 = 4,31$ bloques de MP3 por segundo, que en total necesitan $4,31 * 40 * 10^6 = 172,3 * 10^6$ ciclos/segundo. Luego, el porcentaje de CPU que necesitamos en las tres etapas ahora es:

$$\% \text{ CPU} = \frac{25839,84 + 172,3 * 10^6 + 129199,2}{2,66 * 10^9} * 100 = 6,5 \%$$

d)

El bus debe tener un ancho de banda efectivo de 176400 bytes/segundo. Para transferir 32 bytes, el bus necesita $1 + 2 + 8 = 11$ ciclos. La fórmula que obtenemos es:

$$\frac{32}{11} * f = 176400$$

De donde podemos despejar f que será igual a 60637,5 ciclos/segundo.