

ESTRUCTURA Y TECNOLOGÍA DE COMPUTADORES

I.T.I. Gestión, I.T.I. Sistemas e I. Superior

FINAL, 3^{er} parcial – 3 de julio 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"/>	
Me examino del (marque tantas casillas como sea necesario): 1 ^{er} parcial <input type="checkbox"/> 2 ^o parcial <input type="checkbox"/> 3 ^{er} parcial <input type="checkbox"/> Todo <input type="checkbox"/>	

PARTE II: CUESTIONES TEÓRICO-PRÁCTICAS DEL 3^{ER}- PARCIAL

1) (3 puntos) La máquina M posee un sistema de memoria con las siguientes características:

- Tabla de páginas de 512 entradas.
- Memoria física de 256 bytes.
- Tamaño de página virtual/página física de 8 bytes.
- TLB asociativo de 4 conjuntos de 2 vías (cada uno) con una estrategia de reemplazo LRU y una política de postescritura.
- Memoria caché de correspondencia directa combinada para datos e instrucciones con capacidad para 8 palabras (palabras de 32 bits), bloques de 2 palabras y política de postescritura.

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, la tabla de páginas y la caché en la ejecución del siguiente programa:

```
##### SEGMENTO DE DATOS #####
.data
almacen: .space 96
...
##### SEGMENTO DE TEXTO #####
.text
li $s0, 2
li $s1, 0
la $s2, almacen
lw $t0, 0($s2)
add $s1, $s1, $t0
```

Teniendo en cuenta que:

- La dirección virtual de comienzo del segmento de datos es 0_{10} .
- La dirección virtual de comienzo del segmento de código es 96_{10} .
- Los bits de validez de la tabla de páginas están puestos a 1 en las posiciones 0 y 12, siendo los números de página física que ocupan esas posiciones 16_{10} y 20_{10} respectivamente. La página física 21 y siguientes están todas libres.
- Inicialmente, la caché y el TLB están vacíos.

d) (0'5 puntos) 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) Un disco duro se encuentra conectado a un ordenador a través de un bus de entrada/salida síncrono que posee 32 líneas para direcciones y 32 líneas para datos. Transmitir una dirección o un dato consume un ciclo de bus. Leer o escribir en memoria supone una latencia de 3 ciclos. A partir del quinto ciclo el sistema puede leer o escribir hasta 4 palabras de 32 bits a razón de una palabra por ciclo de bus. Además para las escrituras se necesitan dos ciclos adicionales para escribir el código de corrección de errores. Si sabemos que tanto el bus como el procesador tienen una frecuencia de funcionamiento es 500 Mhz y que el 70% de operaciones son de lectura y el 30% de escritura.

Se pide:

- Porcentaje de uso de la CPU si se usan interrupciones en la gestión de la entrada/salida. Sabemos que en cada interrupción se transfieren 8 palabras y supone un gasto de 10 ciclos.
- Porcentaje de uso de la CPU si se usa DMA en la gestión de la entrada/salida. Sabemos que inicializar la DMA consume 200 ciclos y 100 ciclos el tratamiento de la interrupción. En cada operación se transfiere un bloque de 10 KB.
- Otra opción para gestionar la entrada/salida es mediante sondeo. Se establece como requisito que en esta tarea nunca se utilice más de un 49% del tiempo del procesador. Además sabemos que para realizar cada sondeo se necesitan un total de 20 ciclos. Sabiendo esto, ¿qué número mínimo de palabras se necesitará transmitir en cada sondeo para que se cumpla el requisito anteriormente dado?
- Queremos transferir a memoria un archivo de 30 GB que se encuentra en disco mediante el método de DMA. ¿Qué tiempo se necesitará para realizar esta tarea? Nota: tener en cuenta que mientras se programa y se trata la interrupción de la DMA no se transfiere ningún dato.

PROBLEMA 1

a)

- 512 entradas en la tabla de páginas \rightarrow 512 páginas virtuales = $2^9 \rightarrow$ 9 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 8 bytes = $2^3 \rightarrow$ 3 bits para el desplazamiento

Con todo ello tenemos:

Dirección virtual: 12 bits

Número página virtual: 9 bits	Desplazamiento: 3 bits
-------------------------------	------------------------

Dirección física: 8 bits

Número página física: 5 bits	Desplazamiento: 3 bits
------------------------------	------------------------

Tabla de páginas:

Tamaño	1	1	1	5 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	-
2^9-2	0	0	0	-
2^9-1	0	0	0	-

$$\text{Tamaño} = 2^9 * (3 + 5) \text{ bits} = 4096 \text{ bits} = 512 \text{ bytes}$$

b)

V	M	U	Etí	M'	U'	NPF	
0	1	1	1	7	1	1	5
1							
2							
3							

V	M	U	Etí	M'	U'	NPF	
0	1	1	1	7	1	1	5
1							
2							
3							

Tenemos 4 conjuntos. Por lo tanto, de los 9 bits que conforman el NPV, necesitamos para el índice 2 bits y nos quedan 7 bits para la etiqueta.

$$\text{Tamaño} = 4 * 2 * (3+7+2+5) \text{ bits} = 136 \text{ bits}$$

Caché

	V	M	TAG	DATA
0	1	1	3	64
1				
2				
3				

Si la caché tiene capacidad para 8 palabras y cada bloque tiene 2 palabras, resultan un total de 4 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. En cuanto a la etiqueta, tendrá un tamaño de 3 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 dos bits siguientes → dirección del bloque
- Y los tres bits restantes (más significativos) → etiqueta.

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

$$\text{Tamaño} = 4 * (2 + 3 + (2*32)) \text{ bits} = \mathbf{276 \text{ bits}}$$

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:

$$\begin{aligned} \text{D.V.} &= 96)_{10} \rightarrow 000001100\ 000 \\ \text{NPV (9 bits más significativos)} &\rightarrow 000001100 \\ \text{Page Offset (3 bits menos significativos)} &\rightarrow 000 \end{aligned}$$

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 de TLB** 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	16) ₁₀

12	1	0	0	20) ₁₀

511	0	0	0	-

En esta tabla de páginas tendremos que comprobar si el bit de validez de la posición apuntada por el 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 9 bits más significativos de la dirección virtual, por lo que la posición que tenemos que comprobar es:

$$\text{NPV} = 000001100 \rightarrow \text{posición } 12$$

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

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

	V	M	U	Etí	M'	U'	NPF
0	1	1	1	3) ₁₀	0	1	20) ₁₀
1							
2							
3							

Ahora ya podemos hallar la dirección física. Ésta sería:

$$\begin{aligned} \text{D.F.} &\rightarrow 10100000 = 160)_{10} \\ \text{NPF (cinco bits más significativos)} &\rightarrow 10100 \\ \text{Page Offset (tres bits menos significativos)} &\rightarrow 000 \end{aligned}$$

El NPF es el valor indicado en la posición 12 de la tabla de páginas, codificado con 5 bits. El *page offset* son los tres bits menos significativos de la dirección virtual. Recuerda 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.

$$\text{D.F.} \rightarrow 101 \mathbf{00} 000, \text{ es decir, el bloque de dirección } 0)_{10}.$$

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
0	1	0	101	li \$s0, 2 li \$s1, 0
1				
2				
3				

El valor del tag es 101 ya que la etiqueta está formada por los tres 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 100)10, es decir, D.V. $\rightarrow 000001100 100$.

$$\text{NPV} \rightarrow 000001100 \rightarrow 12)_{10}.$$

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

$$\text{DF} \rightarrow 10100100 \rightarrow 164)_{10}$$

Con esta dirección, accedemos a la caché y se produce un **acierto de caché** ya que en el bloque 0 (101**00**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 (10100 **100**).

INSTRUCCIÓN 3

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

$$\begin{aligned} \text{D.V.} &= 104)_{10} \rightarrow \text{En binario} = 000001101 000 \\ \text{NPV (9 bits más significativos)} &\rightarrow 000001101 \\ \text{Page Offset (3 bits menos significativos)} &\rightarrow 000 \end{aligned}$$

Como podemos observar, el NPV ha cambiado, por lo que se producirá un **fallo de TLB** y habrá que acceder a la tabla de páginas. En este caso, la tabla de páginas nos indica que la página virtual 13 no está en memoria, por lo que se produce un **fallo de página** y habrá que traerla de disco. Como nos dicen que la página física 21 está libre, colocaremos en ella la página virtual 13. La tabla de páginas queda así:

	V	M	U	Número Página Física
0	1	0	0	16) ₁₀
	-
12	1	0	0	20) ₁₀
13	1	0	0	21)₁₀

511	0	0	0	-

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

	V	M	U	Et _i	M'	U'	NPF
0	1	1	1	3) ₁₀	0	1	20) ₁₀
1	1	1	1	3)₁₀	0	1	21)₁₀
2							
3							

Ahora ya podemos hallar la dirección física. Esta sería:

$$D.F. \rightarrow 10101000 = 168)_{10}$$

$$NPF \text{ (cinco bits más significativos)} \rightarrow 10101$$

$$\text{Page Offset (tres bits menos significativos)} \rightarrow 000$$

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.

$$D.F. \rightarrow 101 \mathbf{01} 000, \text{ es decir, el bloque de dirección } 1)_{10}.$$

Esta entrada de la caché está vacía, por lo que se produce un **fallo de caché** y habrá que traer el bloque correspondiente de la memoria principal. Tras esto, la caché queda de la siguiente forma:

	V	M	TAG	DATA
0	1	0	101	li \$s0, 2 li \$s1, 0
1	1	0	101	la \$2, almacen lw \$t0, 0(\$s2)
2				
3				

INSTRUCCIÓN 4

Ahora pasamos a la ejecución de la siguiente instrucción cuya dirección virtual es 108)₁₀, es decir, D.V. \rightarrow 000001101 100.

$$NPV \rightarrow 000001101 \rightarrow 13)_{10}.$$

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

$$DF \rightarrow 10101100 \rightarrow 172)_{10}$$

Con esta dirección, accedemos a la caché y se produce un **acuerdo de caché** ya que en el bloque 1 (10101100) el bit de validez está puesto a 1 y además las etiquetas coinciden.

La instrucción 4 genera a su vez una referencia a datos que también hay que tratar. Como el registro s2 vale 0, la dirección virtual final de datos generada por la instrucción lw es 0:

D.V. = 0_{10} → En binario = 000000000 000
 NPV (9 bits más significativos) → 000000000
 Page Offset (3 bits menos significativos) → 000

El número de página virtual 0 se corresponde con la entrada 0 del TLB. Aunque dicha entrada contiene información en uno de sus conjuntos, las etiquetas no coinciden ($0 \neq 3$), por lo que se produce un **fallo de TLB** y habrá que acceder a la tabla de páginas. Según la tabla de páginas, la página virtual 0 se encuentra en la página física 16, por lo que se produce un **acuerdo de página** y hay que actualizar el TLB, que queda de la siguiente manera:

	V	M	U	Eti	M'	U'	NPF
0	1	1	0	3_{10}	0	1	20_{10}
1	1	1	1	3_{10}	0	1	21_{10}
2							
3							

	V	M	U	Eti	M'	U'	NPF
0	1	1	1	0_{10}	0	1	16_{10}
1							
2							
3							

Ahora ya podemos hallar la dirección física. Ésta sería:

D.F. → $10000\ 000 = 128_{10}$
 NPF (cinco bits más significativos) → 10000
 Page Offset (tres bits menos significativos) → 000

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

D.F. → 100 00 000, es decir, el bloque de dirección 0_{10} .

Esta entrada de la caché está ocupada por otro bloque ya que las etiquetas no coincide, por lo que se produce un **fallo de caché** y habrá que traer el bloque correspondiente de la memoria principal. Tras esto, la caché queda de la siguiente forma:

	V	M	TAG	DATA
0	1	0	100	D(128) D(132)
1	1	0	101	la \$2, almacen lw \$t0, 0(\$s2)
2				
3				

INSTRUCCIÓN 5

La dirección de la nueva instrucción es 112_{10} , que se descompone de la siguiente manera:

D.V. = 112_{10} → En binario = 000001110 000
 NPV (9 bits más significativos) → 000001110
 Page Offset (3 bits menos significativos) → 000

De nuevo, el NPV ha cambiado, por lo que se producirá un **fallo de TLB** y habrá que acceder a la tabla de páginas. Al igual que en la instrucción 3, la tabla de páginas nos indica que la página virtual 14 no está en memoria, por lo que se produce un **fallo de página**. Habrá que traer la página de disco y colocarla en la página física 22 que está libre. La tabla de páginas queda así:

	V	M	U	Número Página Física
0	1	0	0	16) ₁₀
	-
12	1	0	0	20) ₁₀
13	1	0	0	21) ₁₀
14	1	0	0	22)₁₀

511	0	0	0	-

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

	V	M	U	Etí	M'	U'	NPF
0	1	1	1	3) ₁₀	0	1	20) ₁₀
1	1	1	1	3) ₁₀	0	1	21) ₁₀
2	1	1	1	3)₁₀	0	1	22)₁₀
3							

	V	M	U	Etí	M'	U'	NPF
0	1	1	1	0) ₁₀	0	1	16) ₁₀
1							
2							
3							

Ahora ya podemos hallar la dirección física. Ésta sería:

$$\begin{aligned} \text{D.F.} &\rightarrow 10110\ 000 = 176)_{10} \\ \text{NPF (cinco bits más significativos)} &\rightarrow 10110 \\ \text{Page Offset (tres bits menos significativos)} &\rightarrow 000 \end{aligned}$$

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.

$$\text{D.F.} \rightarrow 101\ \mathbf{10}\ 000, \text{ es decir, el bloque de dirección } 2)_{10}.$$

Esta entrada de la caché está vacía, por lo que se produce un **fallo de caché** y habrá que traer el bloque correspondiente de la memoria principal. Tras esto, la caché queda de la siguiente forma:

	V	M	TAG	DATA
0	1	0	100	D(128) D(132)
1	1	0	101	la \$2, almacen lw \$t0, 0(\$s2)
2	1	0	101	add \$s1, \$s1, \$t0
3				

d)

$$\text{Ciclos de espera del procesador} = n^\circ \text{ accesos a MP} * 20 + n^\circ \text{ accesos a MS} * 2000$$

$$\text{Ciclos de espera} = (4 \text{ fallos de caché} + 4 \text{ accesos a tabla de páginas}) * 20 + 2 * 2000 = 4160 \text{ ciclos.}$$

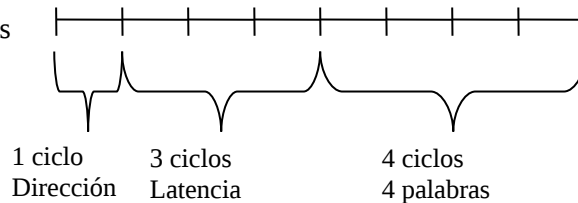
PROBLEMA 2

a)

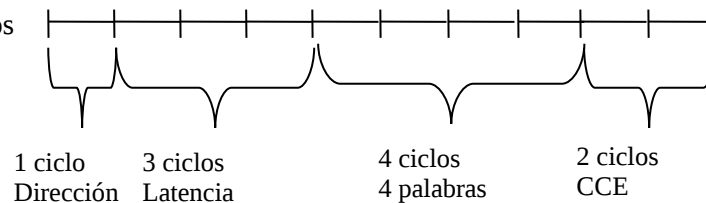
Lo primero que tenemos que hacer es calcular el ancho de banda. Para ello utilizamos la siguiente fórmula:

$$BW = \frac{\text{Bytes}}{\text{Ciclos}} * \text{frecuencia}$$

Para las lecturas tenemos



Para las escrituras tenemos



$$BW_{esc} = \frac{4\text{pal} * 4 \text{ byte/pal}}{(10 \text{ ciclos} * 0,3 + 8 \text{ ciclos} * 0,7)} * 500 * 10^6 = 930,23 \frac{\text{MB}}{\text{s}} \approx \mathbf{930 \text{ MB/s}}$$

Una vez que hemos calculado el ancho de banda, podemos pasar a calcular el porcentaje de uso de la CPU.

En primer lugar tenemos que calcular el número de interrupciones que se producirán en un segundo.

$$\text{N}^\circ \text{ interrupciones} = \frac{930 \text{ MB/s}}{8 \text{ pal} * 4 \text{ bytes}} = \frac{930 * 10^6 \text{ Bytes/seg}}{32 \text{ bytes}} = 29062500 \text{ interrupciones/s}$$

Ahora calculamos el total de ciclos que se consumirán en la realización de las interrupciones.

$$\text{Ciclos totales en interrupciones} = 29062500 \text{ int/s} * 10 \text{ ciclos} = 290625000 \text{ ciclos/s}$$

Por último, calculamos el porcentaje de uso de la CPU.

$$\%CPU = \frac{290625000}{500 * 10^6} * 100 = \mathbf{58'12\%}$$

b)

Para este apartado se seguirán pasos parecidos a los dados en el apartado a)

En primer lugar vamos a calcular el número de operaciones DMA que se realizarán en un segundo.

$$\text{N}^\circ \text{ op. DMA} = \frac{930 \text{ MB/s}}{10 \text{ KB}} = \frac{930 * 10^3 \text{ KB/seg}}{10 \text{ KB}} = 93000 \text{ op DMA/s}$$

Ahora calculamos el total de ciclos que se consumirán en el uso de la DMA.

Ciclos totales en interrupciones = 93000 op. DMA/s * (200 + 100) ciclos = 27900000 ciclos/s

Por último, calculamos el porcentaje de uso de la CPU.

$$\%CPU = \frac{27900000}{500 * 10^6} * 100 = 5'58\%$$

c)

El primer paso es averiguar qué número de ciclos podemos dedicar como máximo a gestionar la entrada/salida mediante sondeo.

Si en un segundo el procesador es capaz de ejecutar $500 * 10^6$ ciclos, y para sondeo sólo podemos dedicar un máximo del 49% del tiempo del procesador, entonces se podrán dedicar un total de

$$500 * 10^6 * 49\% = 245 * 10^6 \text{ ciclos.}$$

Ahora vamos a calcular el número de sondeos que se pueden realizar en un segundo.

$$\text{Nº sondeos} = \frac{245 * 10^6 \text{ ciclos}}{20 \text{ ciclos}} = 12250000 \text{ sondeos}$$

Una vez que sabemos el número de sondeos, sólo nos queda calcular el número de palabras que hay que enviar en cada sondeo.

$$\text{Nº palabras} = \frac{\frac{930 * 10^6 \text{ bytes}}{4 \text{ byte/pal}}}{12250000 \text{ sondeos}} = 18'97 \rightarrow \mathbf{19 \text{ palabras}}$$

d)

En primer lugar vamos a ver lo que se tardaría en transmitir los 30GB a memoria sin tener en cuenta el tiempo usado en la programación del DMA.

$$\text{Tiempo} = \frac{30 * 10^3 \text{ MB}}{930 \text{ MB/s}} = 32'26 \text{ seg.}$$

Ahora vamos a calcular el tiempo usado en la programación del DMA. Para ello vamos a calcular cuantas veces se usa el DMA en la transferencia del fichero.

$$\text{op. DMA} = \frac{30 * 10^6 \text{ KB}}{10 \text{ KB}} = 3 * 10^6 \text{ op. DMA}$$

El siguiente paso es calcular el número de ciclos gastados en realizar todas estas operaciones.

$$\text{Ciclos DMA} = 3 * 10^6 \text{ op DMA} * (200 + 100) \text{ ciclos} = 900 * 10^6 \text{ ciclos}$$

El último paso es calcular cuánto tiempo dedica el procesador en ejecutar todos esos ciclos.

$$\text{Tiempo} = \frac{900 * 10^6}{500 * 10^6} = 1'8 \text{ seg}$$

El tiempo total para la transferencia del fichero será $32'26 \text{ seg} + 1'8 \text{ seg} = \mathbf{34'06 \text{ seg}}$