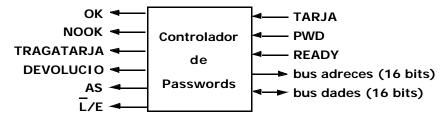
Examen Final Estructura de Computadores II

curso 2003-2004 Q1

- El problema 1 s'ha d'entregar en un full apart.
- La resta de problemes s'han d'entregar als fulls de respostes.
- Les notes sortiran el dia 30 de gener a les 12:00.
- · La revisió es farà el 4 de febrer.

Problema 1 (3 punts)

Es vol implementar un controlador de passwords de tarjes de crèdit.



El funcionament del controlador de passwords es el següent:

- Quan es posa una tarja s'activa el senyal **TARJA** durant 1 cicle i al següent cicle apareix el codi de la tarja al bus de dades $(D_{15:0})$.
- Un cop llegit el codi, quan s'activa el senyal PWD durant 1 cicle, apareix al mateix cicle el password al bus de dades (D_{12:0}). Els bits D_{15:13} valen 000.
- Llavors amb el codi de la tarja s'accedeix a una memòria d'on s'obté el password $(D_{15:3})$ i el nombre d'errors $(D_{2:0})$ asociats a la tarja.
- Si el password es correcte cal posar els errors a 0 a la memòria (mantenint el passsword) i s'activa el senyal OK durant 1 cicle. Dos cicles després d'activar OK es retorna la tarja activant DEVOLUCIO durant 2 cicles
- Sino, s'incrementa el nombre d'errors:
 - si errors < 7 es graba el nombre d'errors a la memòria (mantenint el password) i s'activa el senyal NOOK durant 1 cicle. Dos cicles després d'activar NOOK es retorna la tarja activant DEVOLUCIO durant 2 cicles.
 - sino, s'activa el senyal TRAGATARJA durant 3 cicles.
- Aquest procés es repeteix indefinidament (entre tarja i tarja disposem de temps suficient).

El accessos a memòria es fan tal com es va veure al tema 1, disposa de les senyals pertinents i té les següents característiques:

- La unitat mínima (i única) d'adreçament son 16 bits.
- El **READY** només està actiu durant 1 cicle.
- En les lectures la dada només està disponible durant el cicle que hi ha el READY actiu.

Es demana dissenyar el circuit anomenat Controlador de Passwords utilitzant la tècnica de microprogramació. En concret es demana:

- a) Diseny del camí de dades del controlador.
- b) Format de la microinstrucció.
- c) Microprograma de control.

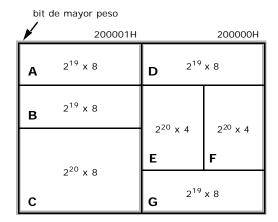
Podeu utilitzar els registres, constants i unitats funcionals que cregueu més convenients. Es valorarà la EFICIENCIA, CLARETAT i SIMPLICITAT de la solució.

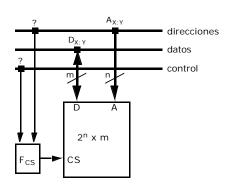
Problema 2 (2 punts)

Disponemos de un procesador de 16 bits con las siguientes señales externas:

- Control: READY, AS y L/E.
- Datos: D_{15:0}.
- Direcciones A_{23:1} y BE_{1:0}.

Queremos conectar a este procesador un sistema de memoria de 4Mbytes (2²²). La memoria está mapeada a partir de la dirección 20000H y está construida con una serie de módulos tal y como se muestra en la figura. Esta figura corresponde a la organización clásica de 2 columnas de byte que permiten accesos a byte y word.





Completad la siguiente tabla indicando para cada módulo, su dirección de inicio, su conexión a los buses de datos y direcciones, así como las funciones de decodificación (F_{CS}).

Módulo	@inicio	$D_{X:Y}$	$A_{X:Y}$	F _{CS}
Α	200001H	D15:8	A19:1	A23 · A22 · A21 · A20 ·
В		D15:8	A19:1	A23 · A22 · A21 · A20 ·
С		D15:8	A20:1	A23 · A22 · A21 ·
D	200000H			
E				
F				
G	500000H			A23 · A22 · A21 · A20 ·

Problema 3 (2 punts)

Disponemos de un procesador de 16 bits con un sistema de Memoria con las siguientes características:

- Memoria Cache (MC):
 - 2 asociativa.
 - Tamaño de línea: 16 bytes.
 - Política de escritura: WRITE THROUGH + WRITE NO ALLOCATE.
 - Tiempo de servicio en caso de acierto: 1 ciclo.
 - Continuación Anticipada.
- Memoria Principal (MP):
 - 4 módulos entrelazados de 4 bytes cada uno.
 - Tiempo de acceso a MP: 3 ciclos.
 - Ancho de Banda entre MP y MC: 4 bytes por ciclo.

Indicad la ocupación de los recursos del sistema:

- Cache: cuándo accede el procesador y cuándo se sirve el acceso.
- Bus: cuándo se está transfiriendo información entre MP y MC (o CPU).
- Memoria: cuándo se está accediendo a los módulos de memoria.

En los siguientes casos:

a) Acceso en lectura y fallo a la dirección 9874H.

Cache									
Bus									
Memoria									

b) Acceso en escritura y fallo a la dirección 765EH.

Cache									
Bus									
Memoria									

- c) 5 Accesos consecutivos con las siguientes características:
- 1: lectura y fallo a la dirección 6708H
- 2: lectura y acierto a la dirección 3456H
- 3: escritura y acierto a la dirección 3450H
- 4: lectura y fallo a la dirección 6900H
- 5: escritura y fallo a la dirección 6908H

Cache															
Bus															
Memoria															

Problema 4 (2 punts)

Disponemos de un procesador de 16 bits con direcciones de 20 bits que tiene una memoria cache de datos con las siguientes características:

- · 3-asociativa, con algoritmo de reemplazo LRU
- 256 bytes por línea
- 12 líneas
- política de escritura: copy back + write allocate

El contenido inicial de la cache (por simplicidad hemos utilizado el número de línea de MP, en vez del tag) es el siguiente:

conjunto 0	DB	conjunto 1	DB	conjunto 2	DB	conjunto 3	DB
984	0	441	0	666	0	OA7	1
98C	1	A19	0	40A	1	45B	0
000	1	A11	0	002	0	F2F	0

El DB=1 indica que la línea correspondiente ha sido modificada. La información de reemplazo está implicita en la posición. Las posiciones inferiores corresponden a las líneas que llevan más tiempo sin utilizarse. Las posiciones superiores corresponden a las últimas líneas utilizadas. Por ejemplo, en el conjunto 3, la línea 0A7 es la última utilizada, y la línea F2F la que lleva más tiempo sin ser utilizada.

Rellenad la siguiente tabla, indicando para cada referencia, el número de línea de MP que le corresponde, a qué conjunto de MC va a parar, si es acierto o fallo, si hay lectura de MP, si hay escritura en MP y la línea reemplazada cuando proceda.

tipo acceso	dirección (hex)	línea MP	conunto MC	¿acierto o fallo?	lectura MP ¿si/no?	Escritura MP ¿si/no?	¿Línea reemplazada?
escr	45B60						
lect	00211						
escr	AE212						
lect	8FCB3						
lect	00254						
escr	00035						
lect	40A96						
escr	8FC77						
escr	98D48						
lect	66619						

¿Cuál es el contenido final de la Memoria Cache?

conjunto 0	DB	conjunto 1	DB	conjunto 2	DB	conjunto 3	DB
						45B	1
						OA7	1
\ <u></u>						F2F	0

Pregunta 5 (1 punt)

a) Dado el siguiente programa escrito en C:

```
for (i=0; i<10000; i++) /* El vector v tiene 10000 elementos */ v[i] = v[i] + 4; /* Cada entero ocupa 4 bytes */
```

Teniendo en cuenta, exclusivamente los accesos al vector v, una herramienta de evaluación nos ha dado los siguientes resultados:

Lecturas de Memoria: 10000Escrituras de Memoria: 10000Fallos en Memoria Cache: 1250

Estos datos proporcionan información sobre uno de los parámetros de la Memoria Cache (tamaño de cache, tamaño de línea, asociatividad, etc.) de datos. ¿Puedes identificarlo? Razona la respuesta.

b) Dado el siguiente programa escrito en C:

```
for (i=0; i<10000; i++) /* El vector v tiene 20000 elementos */ R0 = R0 + v[i] + v[i+4096]; /* Cada entero ocupa 4 bytes */
```

Teniendo en cuenta, exclusivamente los accesos al vector v, una herramienta de evaluación nos ha dado los siguientes resultados:

· Lecturas de Memoria: 20000

• Fallos en Memoria Cache: 20000

Estos datos proporcionan información sobre alguno de los parámetros de la Memoria Cache (tamaño de cache, tamaño de línea, asociatividad, etc.) de datos. ¿Puedes identificarlo? Razona la respuesta.

c) Disponemos de un procesador que tiene memorias caches separadas para datos e instrucciones. Algunos
de los parámetros de las dos memorias caches son los siguientes:

	Cache de instrucciones	Cache de datos
tiempo de servicio en caso de acierto (tsa)	1 ciclo	1 ciclo
tiempo de servicio en caso de fallo y reemplazo de línea no modificada (tsf_{NO})	10 ciclos	14 ciclos
tiempo de servicio en caso de fallo y reemplazo de línea modificada (tsf _{MOD})	-	26 ciclos
%líneas modificadas	-	30%

Dad alguna razón que justifique la diferencia en el tsf_{NO} en ambas caches. Razonad la respuesta.