

COGNOMS: NOM:

3er Control Arquitectura de Computadors

Curs 2015-2016 Q1

- Temps: 12:00 a 15:00
- Poseu clarament amb LLETRES MAJÚSCULES a cada full els cognoms i el nom

Problema 1. (3,5 puntos)

En la práctica 6 sobre políticas de escritura programamos un simulador de una memoria cache con las siguientes características: direcciones de 32 bits, accesos a byte, mapeo directo, tamaño de cache 4 Kbytes, tamaño de bloque 32 bytes. La principal rutina a programar era la siguiente:

void reference (unsigned int address, unsigned int LE)

- a) **Escribe** 4 líneas de código en C para calcular las siguientes variables a partir del parámetro **address**

```
unsigned int byte;      // byte dentro del bloque
unsigned int bloque_m;  // bloque de memoria
unsigned int linea_mc;  // linea de cache donde se mapea dicho bloque
unsigned int tag;       // etiqueta asociada a dicho bloque
```

En la implementación de la cache **Copy back + Write Allocate**, un estudiante ha declarado las siguientes estructuras de datos globales:

```
unsigned int tags[128]; vector correspondiente a la memoria de etiquetas
unsigned int v[128];    vector correspondiente a los bits de validez
unsigned int d[128];    vector correspondiente a los dirty bits
```

- b) **Rellena** la siguiente tabla para los casos descritos en las columnas etiquetadas como **Caso** en función de estas tres variables y el parámetro **LE**.

Caso				Respuesta				
tags[linea_mc]	v[linea_mc]	d[linea_mc]	LE	miss	replacement	mida lec	mida esc	d[linea_mc]
== tag	1	1	lectura					
== tag	1	0	escritura					
== tag	0	0	lectura					
!= tag	1	0	escritura					
!= tag	1	1	lectura					

Nomenclatura de las columnas **Respuesta**:

miss: variable booleana que indica si el acceso produce hit o miss

replacement: variable que indica si el acceso reemplaza un bloque de la cache

mida lec: numero de bytes que se leen del siguiente nivel de la jerarquía (**si no se lee no pongas nada**).

mida esc: numero de bytes que se escriben en el siguiente nivel de la jerarquía (**si no se escribe no pongas nada**).

d[linea_mc]: indica el valor del dirty bit una vez finalizado el acceso 1 o a 0.

Esta cache está conectada a una memoria principal formada por un único módulo DIMM estándar de 4 Gbytes. Este DIMM tiene 8 chips de memoria **DDR-SDRAM (Double Data Rate Synchronous DRAM)** de un byte de ancho cada uno. El DIMM está configurado para leer/escribir ráfagas de 32 bytes (justo el tamaño de bloque de las caches). La latencia de fila es de 4 ciclos, la latencia de columna de 3 ciclos y la latencia de precarga de 2 ciclos.

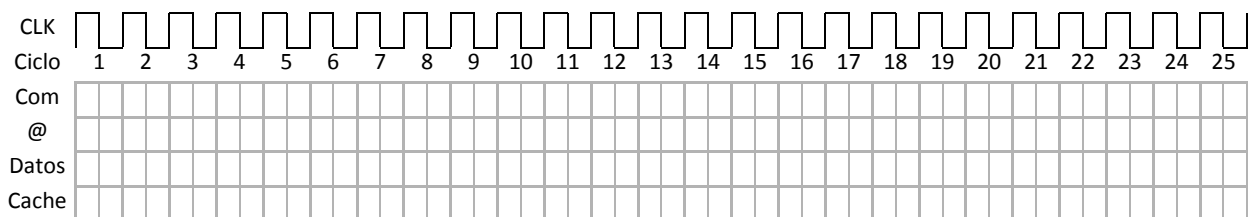
La cache no tiene buffers de escritura, por lo que siempre se escribe en memoria principal primero el bloque reemplazado (en caso que sea pertinente) antes de leer el nuevo bloque. El controlador de memoria envía los comandos necesarios a la DDR-SDRAM de forma que ambos bloques sean transferidos lo más rápidamente posible y se maximice el ancho de banda.

En los siguientes cronogramas indica la ocupación de los distintos recursos de la memoria DDR: bus de datos, bus de direcciones y bus de comandos. Indica también la ocupación de la cache (fila Cache), especifica cuando se accede a la cache si es acierto (A) o fallo (F), y cuando la CPU lee el dato de cache (LC) o escribe el dato en cache (EC). En todos los cronogramas supondremos que no hay ninguna página de DRAM abierta.

Para el acceso:

`tags[linea_mc] != tag && v[linea_mc] == 1 && d[linea_mc] == 0 && LE == escritura`

c) **Rellena** el siguiente cronograma.

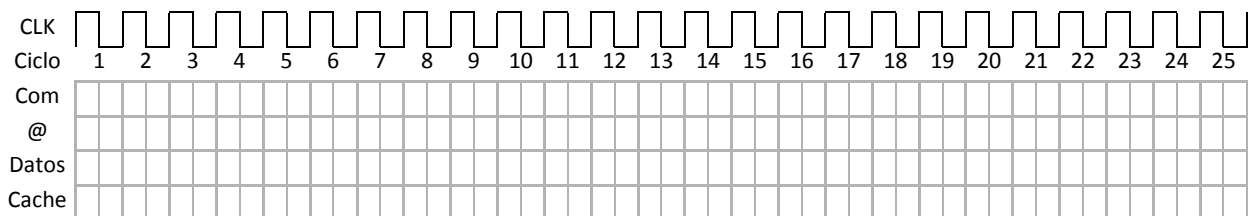


Para el acceso:

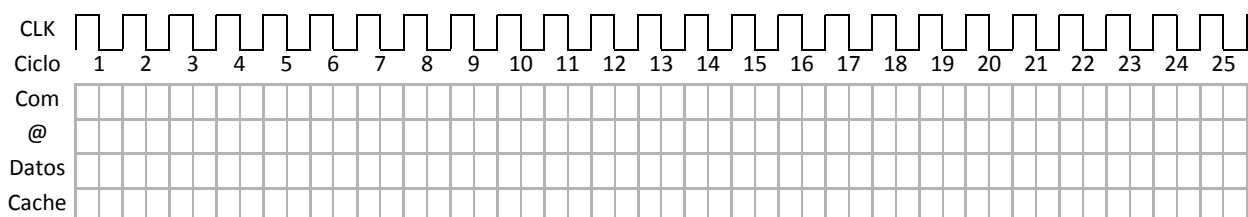
`tags[linea_mc] != tag && v[linea_mc] == 1 && d[linea_mc] == 1 && LE == lectura`

Rellena los siguientes cronogramas en función de la ubicación de los bloques involucrados.

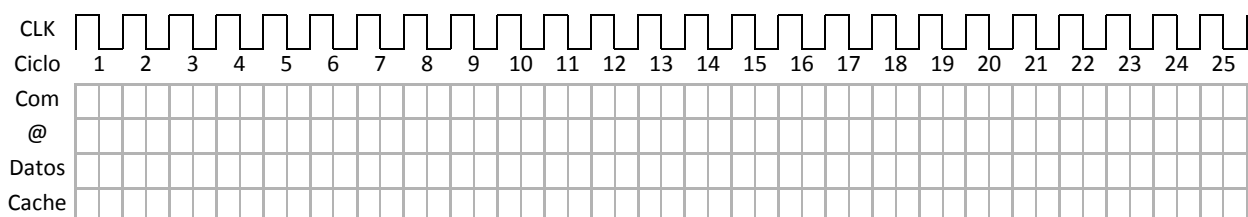
d) Ambos bloques están ubicados en el mismo banco pero en páginas distintas.



e) Ambos bloques están ubicados en la misma página.



f) Ambos bloques están ubicados en bancos distintos.



COGNOMS: NOM:

3er Control Arquitectura de Computadors

Curs 2015-2016 Q1

Problema 2. (3 puntos)

Se dispone de un procesador de tipo Load/Store conectado a un sistema de memoria con una memoria cache L1 unificada (MC) directa con política de escritura write through y write NO allocate.

El tiempo de acceso (Tsa) a memoria cache (MC) es de 15 ns. tanto en lectura como en escritura. El tiempo de acceso a memoria principal (MP) para escribir una palabra es de 70 ns. Para leer o escribir un bloque en la MP se emplean 90 ns.

- a) **Describe**, para cada uno de los siguientes casos: (1) Acierto en MC, (2) Fallo en MC, las acciones a realizar para acceder al sistema de memoria en un **acceso de escritura** e indica el tiempo que tarda **en realizarse el acceso** en cada caso, justificando cómo lo has calculado.

Se lanza el acceso a la MC

(1) Acierto en MC

(2) Fallo en MC

Para mejorar el rendimiento del sistema de memoria se ha decidido añadir una Memoria Victim Cache (VC) con remplazo LRU, de 8 bloques de capacidad y acceso simultáneo a MC. Se han obtenido por simulación las siguientes medidas:

- Tasa de aciertos en MC: 90% de los accesos totales, tanto para lecturas como para escrituras
- Tasa de aciertos en VC: 60% de los accesos a VC que son fallo en MC

El tiempo de acceso (Tsa) a memoria cache (MC) o a la victim cache (VC) es de 15 ns. tanto en lectura como en escritura. El tiempo para intercambiar una línea entre MC y VC es de 10 ns., y en caso de lectura el intercambio se realiza DESPUÉS de servir el dato a la CPU.

- b) **Describe**, para cada uno de los siguientes casos: (1) Acierto en MC, (2) Fallo en MC y acierto en VC, (3) Fallo en MC y fallo en VC, las acciones a realizar para acceder al sistema de memoria en un **acceso de lectura** e indica el tiempo que tarda **en servirse el dato a la CPU** en cada caso, justificando cómo lo has calculado.

Se lanza el acceso en paralelo a la MC y a la VC

(1) Acierto en MC

(2) Fallo en MC y acierto en VC

(3) Fallo en MC y fallo en VC

- c) **Calcula** el tiempo empleado en realizar 1000 accesos de lectura consecutivos suponiendo que no hay ninguna interferencia entre ellos. Para realizar el cálculo, calcula por separado cuantos accesos hay de cada uno de los 3 casos descritos en el apartado a). No se corregirá este apartado si no se ha contestado el apartado b).

(1) Acierto en MC

(2) Fallo en MC y acierto en VC

(3) Fallo en MC y fallo en VC

Tiempo total

- d) **Describe**, para cada uno de los siguientes casos: (1) Acierto en MC, (2) Fallo en MC y acierto en VC (hay intercambio de bloques después de escribir en VC), (3) Fallo en MC y fallo en VC, las acciones a realizar para acceder al sistema de memoria en un **acceso de escritura** e indica el tiempo que tarda cada caso, justificando cómo lo has calculado.

Se lanza el acceso en paralelo a la MC y a la VC

(1) Acierto en MC

(2) Fallo en MC y acierto en VC

(3) Fallo en MC y fallo en VC

COGNOMS: NOM:

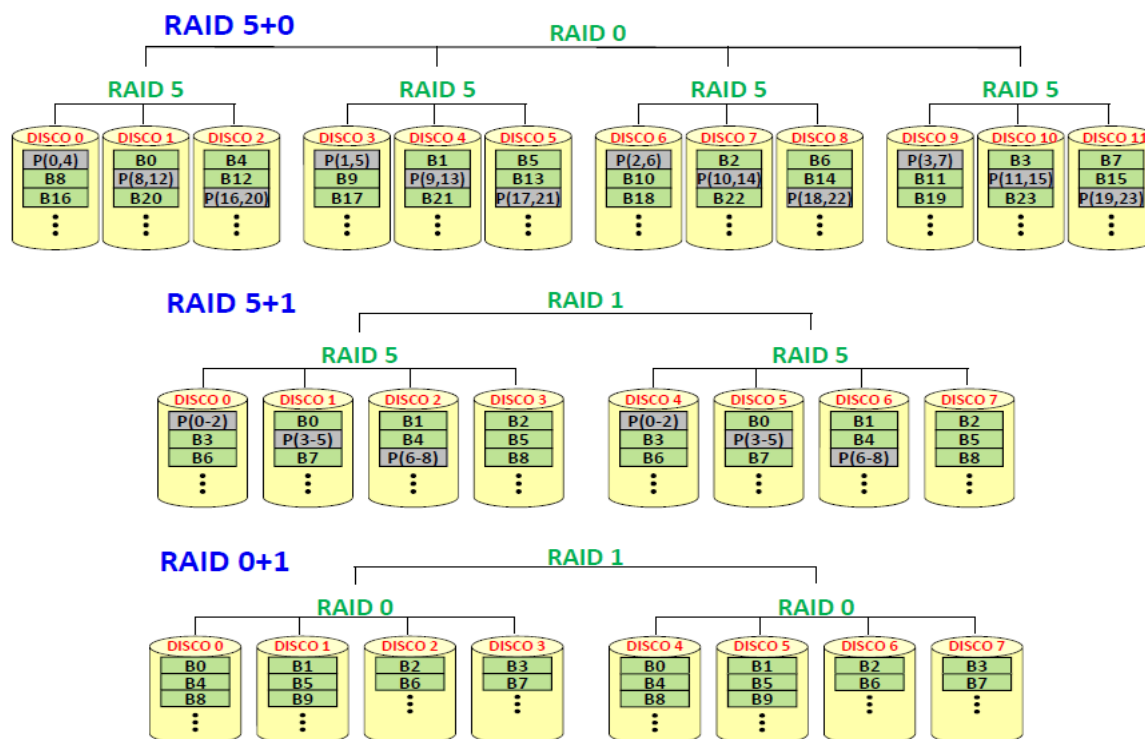
3er Control Arquitectura de Computadors

Curs 2015-2016 Q1

- Temps: 12:00 a 15:00
- Poseu clarament amb LLETRES MAJÚSCULES a cada full els cognoms i el nom

Problema 3. (3,5 puntos)

Determinadas configuraciones de RAID pueden seguir funcionando aunque algunos discos fallen. En la figura se muestran 3 configuraciones de RAID: RAID 5+0, RAID 5+1 y RAID 0+1.



- a) El RAID 5+0 de la figura puede seguir funcionando aunque fallen 4 discos. Marcad con una cruz en la tabla qué 4 discos podrían fallar y el RAID seguir funcionando.
- b) El mismo RAID 5+0 de la figura puede dejar de funcionar si fallan 2 discos. Marcad con una cruz en la tabla qué 2 discos deberían fallar para que el RAID no funcione.

	RAID 0											
	RAID 5			RAID 5			RAID 5			RAID 5		
	disco 0	disco 1	disco 2	disco 3	disco 4	disco 5	disco 6	disco 7	disco 8	disco 9	disco 10	disco 11
(a)												
(b)												

- c) El RAID 5+1 de la figura puede seguir funcionando aunque fallen 5 discos. Marcad con una cruz en la tabla qué 5 discos podrían fallar y el RAID seguir funcionando.
- d) El mismo RAID 5+1 de la figura puede dejar de funcionar si fallan 4 discos. Marcad con una cruz en la tabla qué 4 discos deberían fallar para que el RAID no funcione.

	RAID 1							
	RAID 5				RAID 5			
	disco 0	disco 1	disco 2	disco 3	disco 4	disco 5	disco 6	disco 7
(c)								
(d)								

- e) El RAID 0+1 de la figura puede seguir funcionando aunque fallen 4 discos. Marcad con una cruz en la tabla qué 4 discos podrían fallar y el RAID seguir funcionando.
- f) El mismo RAID 0+1 de la figura puede dejar de funcionar si fallan 2 discos. Marcad con una cruz en la tabla qué 2 discos deberían fallar para que el RAID no funcione.

	RAID 1							
	RAID 0				RAID 0			
	disco 0	disco 1	disco 2	disco 3	disco 4	disco 5	disco 6	disco 7
(e)								
(f)								

Queremos evaluar el rendimiento de utilizar estos RAIDs, para ello vamos a utilizar discos de 1 TB de capacidad y 500 MB/s de ancho de banda. En la evaluación vamos a utilizar una aplicación con un porcentaje muy elevado de operaciones de E/S con disco. En esta aplicación pueden identificarse 4 fases (en las fases 1, 2 y 4 sólo tendremos en cuenta el coste de la transferencia de información):

- fase 1: En donde se ha de leer de disco los datos de entrada que ocupan 10 GB, distribuidos entre todos los discos.
- fase 2: Cálculo, con un tiempo de ejecución de 3s.
- fase 3: En donde se ha de escribir un bloque de datos de 5 GB, lo que permite realizar escrituras secuenciales.
- fase 4, En donde se han de realizar escrituras aleatorias de 5 GB de datos, distribuidas uniformemente entre todos los discos.

- g) Calculad el tiempo de ejecución de nuestra aplicación si utilizamos un único disco.

- h) Calculad el tiempo de ejecución de nuestra aplicación en un RAID 5+1, organizado en 2 grupos de 10 discos cada uno.

- i) Calculad el tiempo de ejecución de nuestra aplicación en un RAID 0+1, organizado en 2 grupos de 10 discos cada uno.