

Cognoms: Nom:

3er Control Arquitectura de Computadors

Curs 2013-2014 Q1

Problema 1. (1,5 puntos)

Se quiere diseñar la memoria cache de datos de primer nivel para un procesador de tipo Load/Store. Se barajan dos alternativas:

- (1) write through y write NO allocate.
- (2) copy back y write allocate

Se dispone de un programa X que ejecuta 8×10^9 instrucciones dinámicas, de las cuales el 25% son instrucciones de acceso a memoria (una instrucción puede hacer un acceso a memoria como máximo), el 30% son instrucciones que realizan una operación en coma flotante, el 35% son instrucciones aritmético lógicas y el 10% restante son instrucciones de salto. Se han obtenido por simulación las siguientes medidas para el programa X:

- porcentaje de escrituras (sobre el total de accesos): 15%
- porcentaje de bloques modificados: 10%
- tasa de aciertos caso (1): 0.9
- tasa de aciertos caso (2): 0.8
- Tiempo de ejecución caso (1): 2,5 segundos
- Tiempo de ejecución caso (2): 3 segundos

El tiempo de acceso (T_{sa}) a memoria cache (MC) es de 10 ns. El tiempo de acceso a memoria principal (MP) para escribir una palabra es de 80 ns. Para leer o escribir un bloque en la MP se emplean 100 ns.

a) **Calculad** el tiempo empleado en realizar 1000 accesos consecutivos en el caso (1)..

b) **Calculad** el tiempo empleado en realizar 1000 accesos consecutivos en el caso (2)..

Cognoms: Nom:

3er Control Arquitectura de Computadors

Curs 2013-2014 Q1

Problema 2. (1,5 puntos)

Dado el siguiente código escrito en C:

```
typedef struct {  
    char a;  
    short b;  
    char c;  
    int d;  
    int e[4];  
} s1;  
  
int examen(s1 j, s2 *k, char m, short *n){  
    short u;  
    char v;  
    double w;  
    ...  
}
```

a) **Dibuja** como quedarían almacenadas en memoria las estructuras s1 y s2, indicando claramente los desplazamientos respecto al inicio y el tamaño de todos los campos.

b) **Dibuja** el bloque de activación de la función examen, indicando claramente los desplazamientos relativos al registro EBP necesarios para acceder a los parámetros y a las variables locales.

Cognoms: Nom:

3er Control Arquitectura de Computadors

Curs 2013-2014 Q1

Problema 3. (4 puntos)

Un DSP (Digital Signal Processor) con arquitectura VLIW genera direcciones físicas de 32 bits. Este procesador tiene una cache de datos que es accedida mediante direcciones físicas. Esta cache tiene bloques (líneas) de 64 bytes, tiene una capacidad de 96 Kbytes y es 3-asociativa.

- a) **Indica** el número de bloques, el número de vías y el número de conjuntos de la cache.

- b) **Dibuja** una dirección física con los campos de bits relevantes para el acceso a la cache, indicando claramente el nombre y tamaño de cada uno de ellos.

Uno de los diseñadores ha propuesto organizar esta cache en 4 bancos con entrelazado a nivel de bloque (línea) de forma que bloques consecutivos se mapeen en bancos consecutivos.

- c) **Dibuja** una dirección física con los campos de bits relevantes para el acceso a la cache organizada en bancos, indicando claramente el nombre y tamaño de cada uno de ellos.

- d) **Explica** los beneficios de una cache organizada en bancos.

El conjunto procesador-cache está conectado a un sistema de memoria principal mediante un único canal de 64 bits al que se ha conectado un DIMM de 4 Gbytes. Este DIMM tiene 8 chips de memoria DDR-SDRAM (Double Data Rate Synchronous DRAM) de un byte de ancho cada uno. Cada chip esta organizado en 16 bancos y cada banco contiene 8K (8192) páginas. El espacio de direcciones físicas se ha mapeado en el DIMM de forma que los datos estén entrelazados a nivel de página, por lo que páginas consecutivas estarán mapeadas en bancos consecutivos. El DIMM esta configurado para leer/escribir ráfagas de 64 bytes, por lo que con un solo acceso se puede leer/escribir un bloque (línea) de memoria. La latencia de fila es de 5 ciclos, la latencia de columna de 4 ciclos y la latencia de precarga de 2 ciclos.

- e) **Describe** muy brevemente el significado de “página” en este contexto.

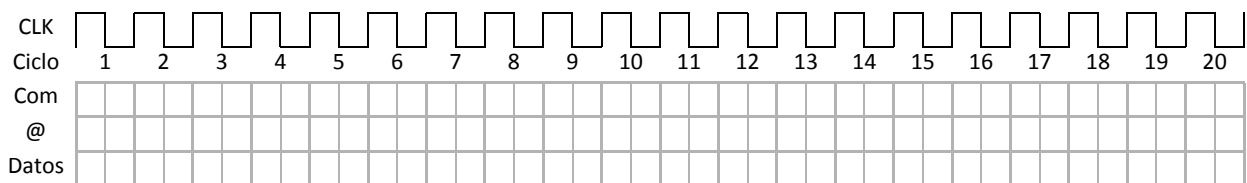
- f) **Dibuja** una dirección física con los campos de bits relevantes para el acceso a la DDR-SDRAM, indicando claramente el nombre y tamaño de cada uno de ellos.

--

- g) **Calcula** el tamaño de página y el número de bloques consecutivos que es posible acceder en una misma página.

--

- h) **Rellena** el siguiente cronograma indicando la ocupación de los distintos recursos para una operación de lectura de un bloque de 64 bytes de la memoria DDR-SDRAM.



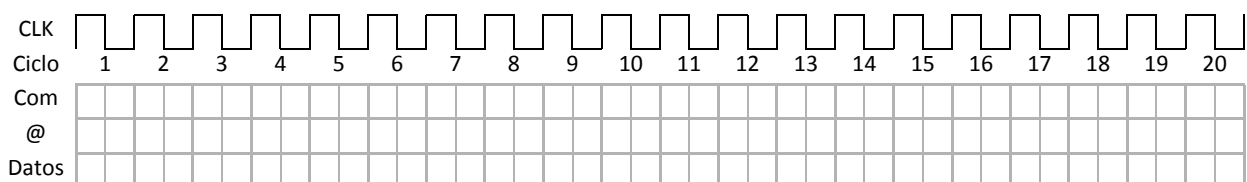
La tensión de alimentación de la DDR es de 1.5 voltios, mientras que la corriente consumida depende de la actividad:

- La memoria esta inactiva -> corriente de fugas 200 mA
 - Durante toda la operación de lectura (desde que se envía el comando ACTIVE hasta que se completa el PRECHARGE), se consumen 100 mA adicionales debidos al funcionamiento de los componentes internos (además de la corriente de fugas que sigue existiendo)
 - Durante la transferencia de datos, además de la corriente de fugas y los componentes internos, hay que alimentar los drivers de entrada salida, con lo que se consumen otros 500 mA adicionales.
- i) **Calculad** la potencia media consumida en la memoria DDR durante un intervalo de 100 ciclos suponiendo que iniciamos un acceso cada 25 ciclos.

--

Este procesador puede realizar hasta 2 accesos a datos simultaneos, por lo que es posible que se produzcan 2 fallos simultaneos en la cache de datos. En este caso, el controlador de memoria envía los comandos necesarios a la DDR-SDRAM de forma que ambos fallos sean servidos lo más rápidamente posible. Supongamos que cada fallo requiere leer un bloque de 64 bytes, que estos se encuentran en bancos distintos y que las correspondientes páginas no han sido previamente accedidas.

- j) **Rellena** el siguiente cronograma indicando la ocupación de los distintos recursos para las dos operaciones de lectura de forma que se maximice el ancho de banda útil del bus.



Cognoms: Nom:

3er Control Arquitectura de Computadors

Curs 2013-2014 Q1

Problema 4. (3 puntos)

- a) **Descriu** el funcionament d'un RAID 5 de 5 discs. Inclou a la descripció un esquema de com es distribueixen les dades, percentatge d'informació redundant, número de discs que han de fallar per a que el RAID 5 deixi de ser operatiu, etc.

- b) En cas de que falli un disc del RAID 5 de l'apartat anterior, **indica** de quants discs hi ha que llegir per a recuperar la informació i descriu clarament la forma en que la informació es recupera.

Disposem de discs físics de 3 TB de capacitat per disc, que ofereixen un ample de banda efectiu de 250 Mbytes/s de lectura i 200Mbytes/s d'escriptura per disc. Amb aquests discs físics volem muntar diferents discs lògics RAID 5. El MTTF d'un disc físic es de 100.000 hores i el MTTR d'un disc físic depèn del número de discs físics en un mateix disc lògic RAID 5 i és de 12h per cada disc físic del disc lògic.

Donades les següents definicions:

- **Capacitat:** Capacitat total del disc lògic
- **% útil:** percentatge de la capacitat total que és capacitat útil
- **Seq RD BW:** Ample de banda efectiu en cas d'accésos de lectura seqüencials
- **Seq WR BW:** Ample de banda efectiu en cas d'accésos d'escriptura seqüencials
- **RND RD BW:** Ample de banda efectiu en cas d'accésos de lectura aleatoris
- **RND WR BW:** Ample de banda efectiu en cas d'accésos d'escriptura aleatoris
- **MTTRn:** MTTR del disc lògic
- **MTTFn:** MTTF del disc lògic

c) **Emplena** la següent taula i **justifica** suficientment com has arribat al resultat quan sigui necessari.

Discs Físics	4	16	justificació
Capacitat			
% util			
Seq RD BW			
Seq WR BW			
RND RD BW			
RND WR BW			
MTTRn			
MTTFn			

Tenim una aplicació A que té dues fases d'execució (lectura y càlcul) i s'executa en un sol processador P amb un únic disc dur lògic RAID 5 de 4 discs en un temps T. La fase de lectura suposa el 30% del temps d'execució i el 80% de la fase de càlcul és paral·lelitzable. Executem la mateixa aplicació (mateix binari) en un sistema multiprocessador MP que té 4 processadors P y un RAID 5 amb 16 discs. La part paral·lela s'ha pogut paral·lelitzar de forma perfecta per 4 processadors.

d) **Calcula** el "speedup" a l'executar l'aplicació A en el sistema MP.

SS és la nova versió (millorada) del processador P. SS és una versió superescalar de P de grau 4 (pot executar fins a 4 instruccions per cycle). La nova micro-arquitectura superescalar de SS aconsegueix que el CPI de la fase de càlcul que és paral·lelitzable es divideixi per 2,3 respecte a P, a la resta de les fases no canvia el CPI.

e) **Calcula** els següents "speedups" en executar l'aplicació A en un sistema superescalar

(1) SS vs P

(2) SS vs MP

Notació: P => sistema original amb un sol processador i un RAID 5 de 4 discs

MP => sistema multiprocessador amb RAID 5 de 16 discs de l'apartat d)

SS => sistema superescalar amb un sol processador i RAID 5 de 4 discs.