

COGNOMS: NOM:

3er Control Arquitectura de Computadors

Curs 2014-2015 Q2

- Temps: 16 a 19
- Poseu clarament amb LLETRES MAJÚSCULES a cada full els cognoms i el nom

Problema 1. (3 puntos)

Dado el siguiente fragmento de código escrito en C, que compilamos para un sistema linux de 32 bits:

```
typedef struct {  
    char a;  
    short b;  
    char c;  
    double d;  
} A;  
typedef struct {  
    A x[3];  
    char ch;  
    int z;  
} B;  
int examen(int p1, short p2[4], B p3, char p4){  
    A VL1;  
    short VL2[4];  
    char VL3;  
    char VL4;  
    ...  
}
```

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

Struct A

Struct B

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

Dado el siguiente fragmento de código escrito en C, que compilamos para un sistema linux de 32 bits:

```
A x[1000]; /* A es el struct del apartado a) */  
for (i=0; i<999; i++)  
    x[i].d = x[i+1].d + 1.0;
```

- c) **Calcula** el número de aciertos y fallos de lectura y de escritura que se producen en la cache de datos al ejecutar completamente el bucle **for**. El vector x está almacenado a partir de la dirección 0x12345000. La memoria cache de datos tiene las siguientes características:
- Write through, Write NO allocate
 - Líneas de 128 bytes
 - Mapeo directo
 - Tamaño cache: 1 Kbyte.

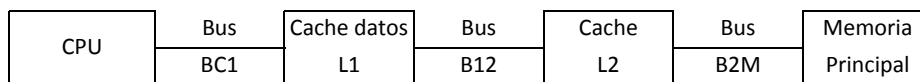
COGNOMS: NOM:

3er Control Arquitectura de Computadores

Curs 2014-2015 Q2

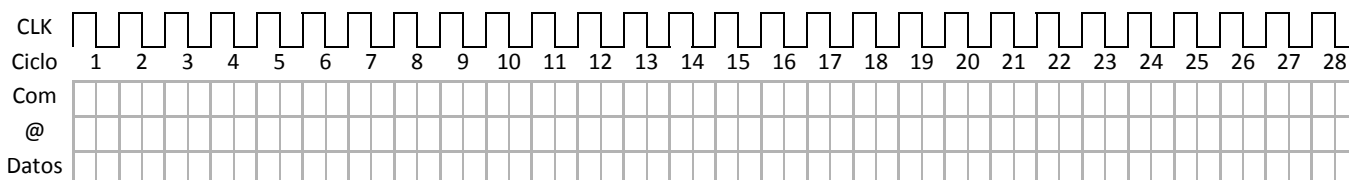
Problema 2. (3,5 puntos)

Disponemos de la jerarquía de memoria (solo datos, ignoramos las instrucciones) mostrada en la siguiente figura:



La memoria principal está 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 64 bytes (justo el tamaño de bloque de las caches). La latencia de fila es de 3 ciclos, la latencia de columna de 4 ciclos y la latencia de precarga de 2 ciclos.

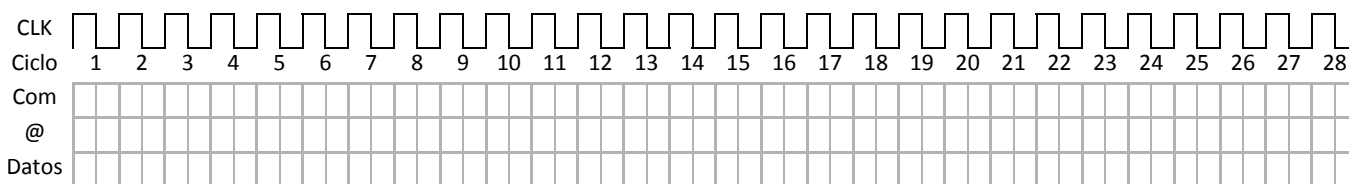
- a) **Rellena** el siguiente cronograma indicando la ocupación de los distintos recursos (bus de datos, bus de direcciones, bus de comandos) para una operación de lectura de un bloque de 64 bytes de la memoria **DDR**.



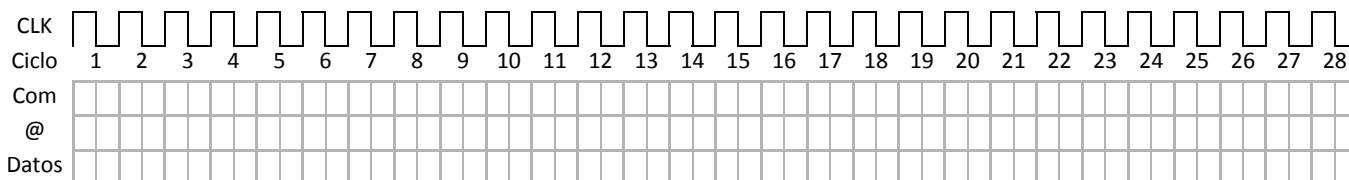
Cuando en la L2, que es copy-back, se reemplaza un bloque modificado, primero se tiene que escribir el bloque reemplazado en la memoria principal y a continuación leer el bloque que se necesita. En este caso, nuestro controlador de memoria envía los comandos necesarios a la DDR-SDRAM de forma que ambos accesos sean servidos lo más rápidamente posible. Suponiendo que inicialmente no hay ninguna página abierta. Rellena los siguientes cronogramas, indicando la ocupación de los distintos recursos, para una operación de escritura seguida de una lectura, de forma que se maximice el ancho de banda útil del bus.

Observación importante: El bus de datos, a diferencia del de direcciones y el de comandos, realiza transmisiones en ambos sentidos. En las memorias reales no es posible cambiar el sentido de la transmisión de forma inmediata. Desde que se reciben los últimos datos de una escritura hasta que se envían los primeros de la siguiente lectura (y también viceversa) debe transcurrir un ciclo como mínimo. Ten en cuenta esta restricción en los siguientes cronogramas.

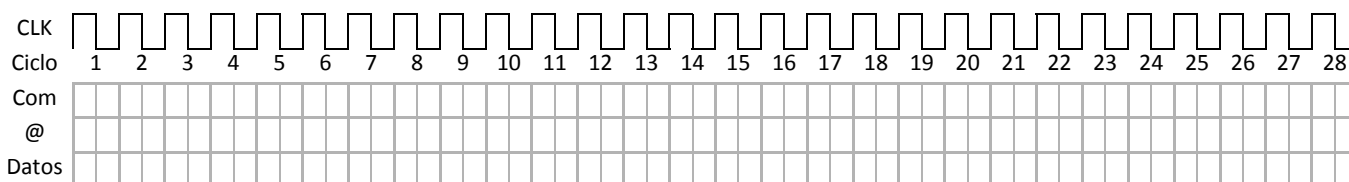
- b) Ambos bloques están ubicados en el mismo banco pero en páginas distintas.



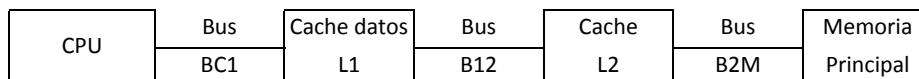
- c) Ambos bloques están ubicados en la misma página.



- d) Ambos bloques están ubicados en bancos distintos.



En nuestra jerarquía de memoria (solo datos, ignoramos las instrucciones) que reproducimos de nuevo por comodidad:



La CPU ejecuta un programa que realiza 800 millones de loads y 200 millones de stores, todos ellos de 32 bits (4bytes). La siguiente tabla muestra las principales características de las caches y su comportamiento con dicho programa.

Cache	Tamaño de bloque	Número de bloques	Asociatividad	Política de escritura	% fallos ^a lecturas	% fallos ^b escrituras	bloques ^c modificados
L1	64 bytes	256	directa	write through + write NO allocate	10%	50%	1/4
L2	64 bytes	4096	directa	copy back + write allocate	40%	5%	1/3

- a. Tasa de fallos local relativa al número de lecturas de dicha cache.
- b. Tasa de fallos local relativa al número de escrituras en dicha cache.
- c. Probabilidad de que al reemplazar un bloque este haya sido modificado desde que entró en la cache.

- e) **Calcula** el número de lecturas y escrituras que se realizan por los buses B12 y B2M y el número total de bytes que se transfieren. Justifica brevemente la respuesta con un cálculo o frase. Como ejemplo se muestran las lecturas y escrituras de la CPU en la cache L1 por el bus BC1 y el total de bytes transferidos.

Bus	Lecturas	Escrituras	Bytes
BC1	800 millones (1 lectura por load)	200 millones (1 escritura por store)	4.000 millones $(800 \times 10^6 + 200 \times 10^6) \times 4$
B12			
B2M			

En una jerarquía de memoria inclusiva el nivel superior contiene siempre los bloques del nivel inferior. Esto significa que los bloques almacenados en L1 están todos presentes en L2, y por lo tanto la L1 contiene un subconjunto de los bloques almacenados en L2. Sin embargo, nuestra jerarquía, debido a la combinación de políticas de escritura no es inclusiva.

- f) **Explica** con un ejemplo, porque en nuestra jerarquía puede darse el caso que un bloque de memoria esté presente en L1 sin estarlo en L2.

COGNOMS: NOM:

3er Control Arquitectura de Computadores

Curs 2014-2015 Q2

Problema 3. (3,5 puntos)

Disponemos del diseño de dos procesadores que son totalmente compatibles a nivel de lenguaje máquina. Estos procesadores se denominan: HP de “High Performance” y LP de “Low Power”. La siguiente tabla muestra las características más relevantes: carga capacitiva equivalente (C), voltaje de alimentación (V), intensidad de fugas (I_f), frecuencia de funcionamiento (F) y CPI de una aplicación A (obtenido con el mismo ejecutable para ambos procesadores).

Procesador	C	V	I_f	F	CPI
HP	10 nF	1,2 V	10 A	4 GHz	0,5 c/i
LP	2 nF	0,9 V	1 A	1 GHz	1,5 c/i

- a) **Calcula** el speed-up del procesador HP respecto el LP al ejecutar la aplicación A.

Para calcular la disipación de potencia tendremos en cuenta la potencia debida a conmutación y la debida a fugas (la potencia debida a cortocircuito es despreciable).

- b) **Escribe** la fórmula para cada una de las fuentes de disipación de potencia y **calcula** la potencia disipada por dicha causa para ambos procesadores.

	Potencia conmutación (Pc)	Potencia fugas (Pf)	Potencia total (Pt)
Formula			
procesador HP			
procesador LP			

- c) **Calcula** la ganancia en energía del procesador LP respecto al procesador HP al ejecutar la aplicación A.

El TDP (Thermal Design Point) de un chip es la potencia máxima que el fabricante garantiza puede disipar sin que su funcionamiento se vea comprometido. Nuestros diseñadores de circuitos nos han asegurado que el área no va a ser un problema, pero nos han comentado que disponemos de un TDP máximo de 92W para los procesadores, por lo que en un chip podemos poner o bien 1 solo procesador HP (denominado chip H-1) o bien 36 procesadores LP (denominado chip L-36).

Es posible paralelizar parte de la aplicación A de forma que el trabajo se distribuya equitativamente entre los procesadores disponibles (el overhead debido a sincronización es despreciable). La parte secuencial se seguirá ejecutando en un solo procesador independientemente de los que haya disponibles en el chip. Tanto la parte secuencial como la paralela consiguen el mismo CPI que la aplicación original en los respectivos procesadores (1,5 c/i en un LP y 0,5 c/i en un HP).

A uno de los diseñadores se le ha ocurrido que, dado que el área no es un problema, podríamos poner 1 procesador HP y 36 LP en el mismo chip, de forma que la parte secuencial de la aplicación se ejecute en el procesador HP y la paralela en los 36 procesadores LP. Durante la ejecución de la parte secuencial los 36 procesadores LP estarán apagados, mientras que durante la ejecución de la parte paralela el procesador HP estará apagado, de forma que nunca se superará el TDP máximo 92W. A esta tercera opción la denominaremos chip HL-136.

- d) **Rellena** la siguiente tabla con el **speed-up** que se obtendr a con cada uno de los chips mencionados, en funci n del % de la parte paralela, **respecto a 1 procesador LP** (justifica muy brevemente cada respuesta).

Chip	0% (totalmente secuencial)	100% (totalmente paralela)	60% (el 40% es secuencial)
H-1 (1 procesador HP)			
L-36 (36 procesadores LP)			
HL-136 (1 procesador HP + 36 procesadores LP)			

Debido a mejoras tecnol gicas (y para que no la cagues, aunque te hayas equivocado), en la implementaci n final del chip HL-136, los procesadores consumen algo menos de lo previsto en el apartado b). Cuando est n activos, el procesador HP consume 60W y los procesadores LP consumen 2W cada uno. Obviamente cuando est n apagados ninguno de los procesadores consume nada. En este chip, ejecutamos una **aplicaci n B** (que tarda 2 horas) compuesta por 3 fases:

- La **Fase 1** (30 minutos) es totalmente secuencial y se ejecuta en el procesador HP (los 36 procesadores LP est n apagados).
- La **Fase 2** (30 minutos) es totalmente paralela y se ejecuta en los 36 procesadores LP (el procesador HP est  apagado).
- La **Fase 3** (1 hora) es una fase de entrada/salida en que la CPU tiene una actividad muy baja, por lo que no importa la velocidad de la CPU. Para reducir el consumo, la Fase 3 se ejecuta en uno de los procesadores LP (el procesador HP y 35 de los procesadores LP est n apagados).

- e) **Calcula** la potencia media disipada por el chip HL-136 al ejecutar la aplicaci n B

La **Fase 3** de entrada/salida se corresponde exclusivamente a accesos a disco y realiza un 60% de lecturas y un 40% de escrituras. El sistema de disco est  formado por un RAID 5 de 10 discos. Cada uno de estos discos tiene una capacidad de 1 Terabyte, un ancho de banda efectivo de 200 Mbytes/s. Tanto las escrituras como las lecturas a disco se corresponden a accesos aleatorios. El controlador del RAID ordena las peticiones de acceso para aprovechar al m ximo la concurrencia entre discos. En ambos casos consideraremos que disponemos de suficientes peticiones de lectura/escritura de forma que el controlador del RAID siempre puede aprovechar el ancho de banda de todos los discos f sicos.

- f) **Calcula** la capacidad  til y el ancho de banda efectivo del RAID 5 (durante la Fase 3).

Este RAID 5 lo sustituimos por un RAID 51 formado por 2 grupos de 10 discos (20 discos en total). El tiempo de la Fase 3 depende exclusivamente de la velocidad del sistema de discos.

- g) **Calcula** la capacidad  til y el speed-up de la fase 3 al reemplazar el RAID 5 por el nuevo RAID 51.