

COGNOMS: NOM:

3er Control Arquitectura de Computadors

Curs 2015-2016 Q2

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

Problema 1. (3,2 puntos)

Dado el siguiente código escrito en ensamblador del x86:

```

        movl $0, %ebx
        movl $0, %esi
for:
        cmpl $512*1000, %esi
        jge end

        (a) movl (%ebx, %esi, 4), %eax
        (b) addl 2*4*1024(%ebx, %esi, 4), %eax
        (c) movl %eax, 3*4*1024(%ebx, %esi, 4)

        incl %esi
        jmp for
end:

```

Sabemos que el código se ejecuta en un sistema con memoria cache y memoria virtual. La memoria virtual utiliza páginas de tamaño 4KB y disponemos de un TLB de 4 entradas y reemplazo LRU. La memoria cache de datos (únicos accesos a memoria que contemplaremos en este problema) es *Write Through + Write No Allocate*, de 2 vías con reemplazo LRU, tamaño 4 KB y 32 bytes por bloque. Responde a las siguientes preguntas:

- a) **Calcula**, para cada uno de los accesos etiquetados como (a, b, c), el conjunto de la memoria cache al que se accede en cada una de las siguientes iteraciones del bucle.

iteración	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
a																		
b																		
c																		

- b) **Calcula** la cantidad de aciertos y de fallos de cache, en todo el código.

- c) Para cada uno de los accesos indicados (etiquetas a, b, c), **indica** a qué página de la memoria virtual se accede en cada una de las siguientes iteraciones del bucle (recuerda que los accesos son a 4 bytes).

iteración	0	1*512	2*512	3*512	4*512	5*512	6*512	7*512	8*512	9*512
a										
b										
c										

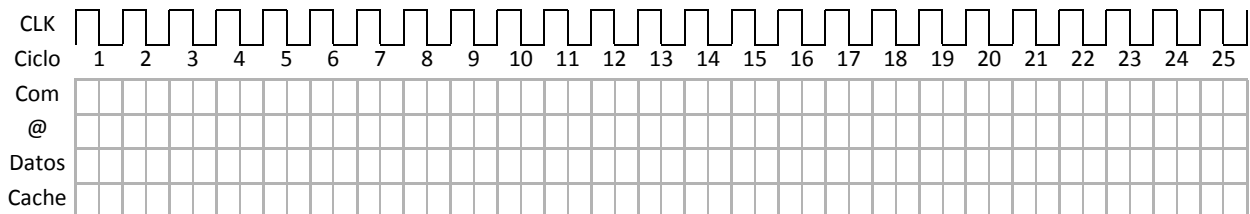
d) **Calcula** la cantidad de aciertos y de fallos de TLB, en todo el código.

--

La cache de datos 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)**. El DIMM esta configurado para leer/escribir ráfagas de 32 bytes (justo el tamaño de bloque de la cache). La latencia de fila es de 4 ciclos, la latencia de columna de 3 ciclos y la latencia de precarga de 2 ciclos.

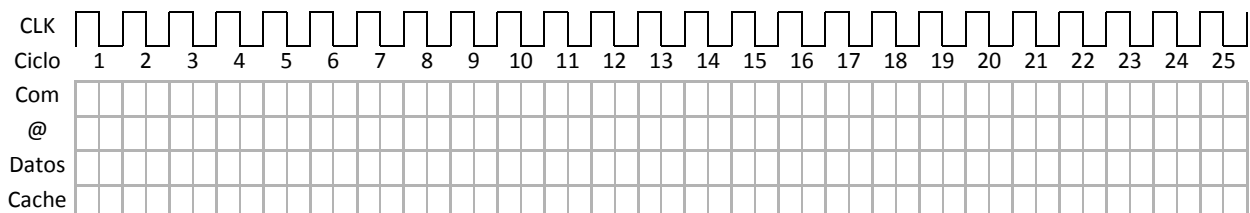
En los siguientes cronogramas, todos fallos en lectura, indica la ocupación de los distintos recursos de la memoria DDR: bus de datos, bus de direcciones y bus de comandos. En la fila etiquetada como Cache indica en el ciclo en que se accede a la cache y se produce el fallo (F), y en que ciclo la CPU recibe el dato de cache (LC). En todos los cronogramas supondremos que no hay ninguna página de DRAM abierta.

e) **Rellena** el siguiente cronograma para un fallo en lectura.

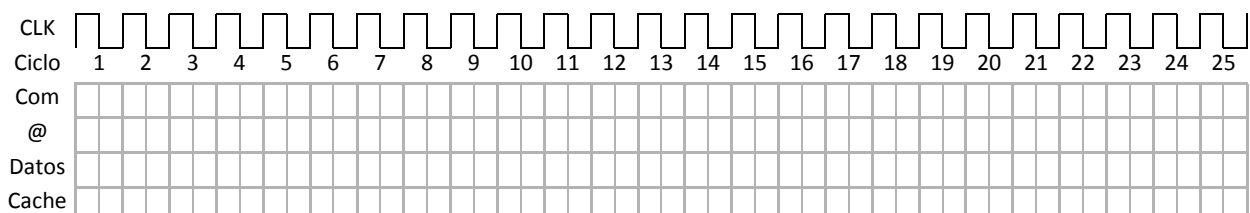


Esta cache de datos tiene un mecanismo de prefetch que, en caso de fallo en lectura, puede desencadenar un prefetch del siguiente bloque. El controlador de memoria envía los comandos necesarios a la DDR-SDRAM de forma que ambos bloques sean transferidos lo más rápidamente posible y se maximice el ancho de banda. Rellena los siguientes cronogramas para **fallo en lectura + prefetch** en función de la ubicación de los dos bloques involucrados.

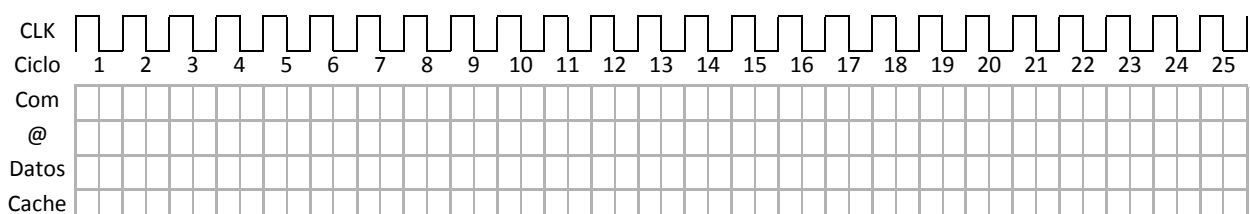
f) Ambos bloques están ubicados en el mismo banco pero en páginas distintas.



g) Ambos bloques están ubicados en la misma página.



h) Ambos bloques están ubicados en bancos distintos.



COGNOMS: NOM:

3er Control Arquitectura de Computadors

Curs 2015-2016 Q2

Problema 2. (3,2 punts)

Tenim una CPU (C1) que té un temps de cicle (T_c) de 1 ns. A l'executar un programa P (que executa 40×10^9 instruccions) en un simulador de C1 on tots els accessos a memòria fan *hit* a la Cache d'Instruccions (I\$) i Dades (D\$) tarda 50×10^9 cicles.

- a) **Calcula** el CPI ideal (CPI_{ideal}) i el temps d'execució en segons (T_{exec}) del programa P en aquest sistema de memòria ideal.

Mesurem el número mig de referències per instrucció (nr) i veiem 1.5 refs/instrucció repartides de la següent manera (1) 1.00 refs/inst a instruccions i (2) 0.5 refs/inst a dades.

Amb una I\$ i una D\$ reals tenim un *miss rate* de D\$ del 9% i de I\$ del 6%.

En cas d'encert a la I\$ i a la D\$ el temps de servei es de 1 cicle. En cas de *miss* a la I\$ o a la D\$ el temps de penalització per accedir a la memòria es de 100 cicles.

La D\$ segueix una política d'escriptura amb *Copy Back* i *Write Allocate*, tot i que en el programa P el nombre de blocs modificats es negligible.

- b) **Calcula** el temps mig d'accés a memòria en cicles pels accessos a instruccions (T_{maI})

- c) **Calcula** el temps mig d'accés a memòria en cicles pels accessos a dades (T_{maD})

- d) **Calcula** el temps mig d'accés a memòria en cicles per tots els accessos (T_{ma})

- e) **Calcula** el temps d'execució del programa P a la CPU C1 amb caches I\$ i D\$ reals (T_{exeR1})

Per a millorar el rendiment del programa P dissenyem una nova CPU (C2) a partir de la CPU C1 descrita anteriorment a la que li hem afegit una cache de segon nivell Unificada (L2\$). El temps d'execució del programa P a C2 es de 218 s i el *miss rate* local de la L2\$ pel programa P es del 30%. La mida de bloc (línia) de totes les caches es de 64 bytes. Els accessos a la I\$ son sempre de 4 bytes (la mida de les instruccions), els accessos a la D\$ son sempre de 8 bytes.

- f) **Calcula** el nombre d'accessos, el nombre de bytes que es demanen i el ample de banda (en MBytes/segon) a tots els elements de la jerarquia: I\$, D\$, L2\$ i Memòria Principal (MP), que utilitza el programa P a la CPU C2. Justifica les respostes.

	Accesos	Bytes demanats	Ample de banda
I\$			
D\$			
L2\$			
MP			

Un accés a I\$ requereix una energia dinàmica (de commutació) de 1 nJ, a D\$ de 1,5 nJ, a L2\$ de 5 nJ i a MP de 50 nJ.

- g) **Calcula** l'energia total i la potencia dinàmica mitja consumida per la jerarquia de memòria de C2 durant l'execució de P.

La cache L2\$ té un temps de servei en cas d'encert de 10 cicles i una penalització en cas de *miss* de 100 cicles.

- h) **Calcula** el mínim *hit rate* local (h) que hauria de tenir la L2\$ per a que un programa s'executi més ràpidament en la CPU C2 que en la C1

COGNOMS: NOM:

3er Control Arquitectura de Computadors

Curs 2015-2016 Q2

Problema 3. (3,6 puntos)

Tenemos que implementar un servidor de páginas web usando algunos de los siguientes componentes:

- Procesador RISC superescalar fuera de orden (RISC) que consume 60W a una frecuencia de 2GHz.
- Procesador VLIW de ancho 4 (VLIW) que consume 20W. Su frecuencia de funcionamiento es de 1 GHz. Al compilar medimos que cada instrucción del VLIW (las llamaremos I_v) equivale en media a 3 instrucciones del RISC en cualquier parte del programa.
- Discos duros de 5TB y ancho de banda de 200 MBytes/s por disco y un consumo de 10W.

El kernel del programa a ejecutar en el servidor tiene 3 fases diferenciadas:

- 1) Fase secuencial. Ejecuta $1,2 \times 10^9$ instrucciones dinámicas en el RISC.
- 2) Fase paralela de cálculo. Por simplicidad supondremos que esta fase es perfectamente paralelizable. En esta fase se ejecutan $2,4 \times 10^9$ instrucciones y realiza 5×10^9 operaciones de coma flotante en el RISC (usando instrucciones SIMD, esta es la única fase con operaciones de coma flotante del programa).
- 3) Fase de I/O. La velocidad de esta fase está siempre limitada por el rendimiento del sistema de disco, la cantidad de instrucciones que ejecuta es despreciable. La mitad de las operaciones de disco son lecturas y la otra mitad escrituras de muchos ficheros pequeños aleatoriamente distribuidos en el sistema.

Para estimar el rendimiento que podemos obtener se realiza una ejecución secuencial (en un solo procesador) del programa en un sistema (llamado SR) que dispone de un RISC y un disco. Una instancia del programa tarda 2s en ejecutarse. La fase 1 tarda el 20% de dicho tiempo, la fase 2 el 60% y la fase 3 el 20% restante.

- a) **Calcula** el IPC (instrucciones por ciclo) de las fases 1, 2 y de todo el programa en el SR.

- b) **Calcula** los MFLOPS efectivos de las fases 1 y 2 y de todo el programa en el SR.

Medimos el I_vPC (instrucciones del VLIW, I_v por ciclo) y obtenemos una medida para la fase 1 de 0,5 I_vPC y para la fase 2 una media de 1 I_vPC .

- c) **Calcula** el tiempo de cada una de las tres fases del programa en el VLIW con un disco (a esta configuración la llamaremos SV).

El sistema de disco solo consume energía durante la fase 3 (los discos están apagados durante las fases 1 y 2), mientras que al menos un procesador debe estar encendido durante todas las fases (incluida la 3).

- d) **Calcula** la ganancia en energía al ejecutar una instancia del programa del SV sobre el SR en porcentaje.

Para mejorar la fase 3 se propone usar un sistema con 10 discos. Se desea evaluar el rendimiento y la capacidad de dos alternativas: un RAID 01 (dos grupos de 5 discos) y un RAID 6.

- e) **Calcula** la capacidad efectiva de cada alternativa.

- f) **Calcula** el speedup en lectura, el speedup en escritura (ambos respecto a la configuración con un solo disco) y el tiempo de la fase 3 para cada RAID.

Decidimos evaluar la posibilidad de implementar dos sistemas multiprocesador, uno con procesadores RISC y otro con procesadores VLIW (ambos usando el RAID 01 visto anteriormente). Por razones de coste tenemos un límite de consumo instantáneo de 160 W. En cada fase se pueden apagar los componentes que no se usen, a excepción de 1 procesador que tiene que estar encendido siempre.

- g) **Calcula** el número de componentes y el tiempo total de ejecución de una instancia del programa en cada alternativa multiprocesador.

- h) **Calcula** la potencia media consumida durante la ejecución del programa usando la alternativa multiprocesador más rápida calculada en el apartado anterior.