

COGNOMS: 



NOM: 





**IMPORTANT** leer atentamente antes de empezar el examen: Escriba los apellidos y el nombre antes de empezar el examen. Escriba un solo carácter por recuadro, en mayúsculas y lo más claramente posible. Es importante que no haya tachones ni borrones y que cada carácter quede enmarcado dentro de su recuadro sin llegar a tocar los bordes. Use un único cuadro en blanco para separar los apellidos y nombres compuestos si es el caso. No escriba fuera de los recuadros.

**Problema 1. (2 puntos)**

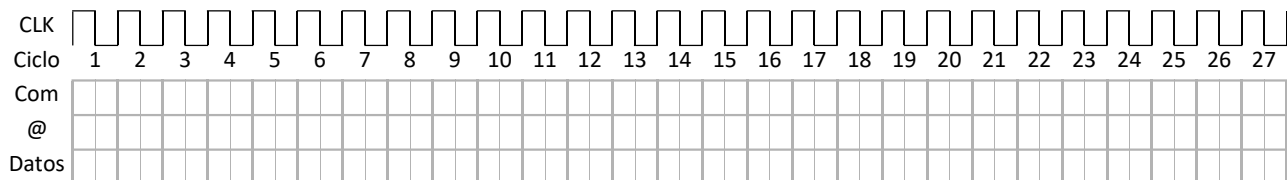
Una **CPU** está conectada a una cache y una memoria principal formada por un único módulo DIMM estándar de 16 GBytes. Este DIMM tiene 8 chips de memoria **DDR-SDRAM (Double Data Rate Synchronous DRAM)** de 1 byte de ancho cada uno. La DDR-SDRAM tiene 8 bancos. El DIMM esta configurado para leer/escribir 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. Cuando se solicitan múltiples bloques a la DDR-SDRAM, el controlador de memoria envía los comandos necesarios a la DDR-SDRAM de forma que los bloques sean transferidos lo más rápidamente posible.

La siguiente tabla muestra en qué banco y qué página de DRAM (fila) se encuentran los bloques etiquetados con las letras A B C D E.

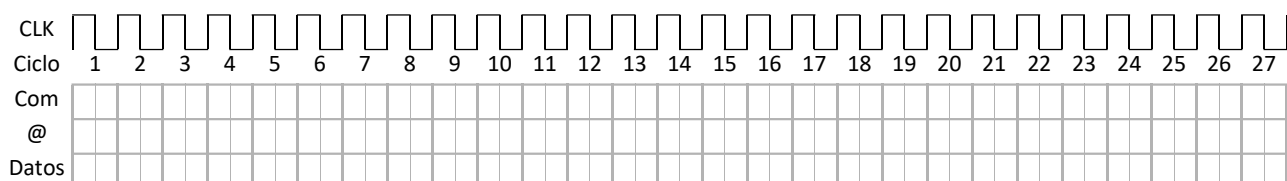
Bloque	A	B	C	D	E
Banco	0	0	1	1	5
Página	10	10	10	25	40

Rellena los siguientes cronogramas para la lectura de varios bloques de 64 bytes (**sin cambiar el orden de los accesos**), en función de la ubicación de los bloques involucrados de forma que se minimice el tiempo total. Indica la ocupación de los distintos recursos de la memoria DDR: bus de datos, bus de direcciones y bus de comandos. En todos los cronogramas supondremos que no hay ninguna página de DRAM abierta previamente y que al final deben quedar todas cerradas.

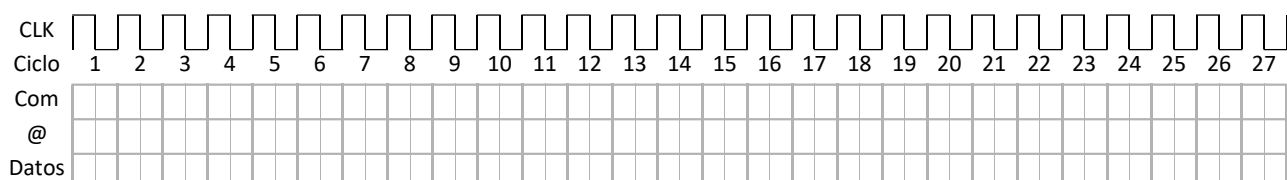
a) **Rellena** el siguiente cronograma para la lectura de los bloques AB.



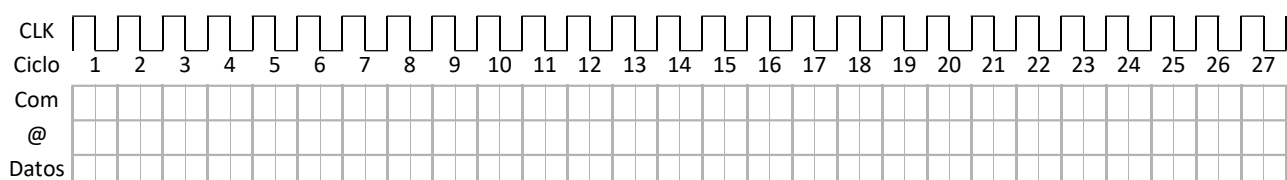
b) **Rellena** el siguiente cronograma para la lectura de los bloques AC.



c) **Rellena** el siguiente cronograma para la lectura de los bloques CD.



d) **Rellena** el siguiente cronograma para la lectura de los bloques ADBE.



COGNOMS:

NOM:

### Problema 2. (3 puntos)

Un procesador funciona a una frecuencia de 1GHz y esta conectado a una cache de datos nivel 1 (L1) 2-asociativa de 16 KBytes de capacidad con bloques de 64 bytes. La cache no tiene ninguna de las mejoras vistas en clase. En caso de acierto, los accesos a la cache tardan 1 ciclo y en caso de fallo tenemos una penalización media de 20 ciclos. Hemos determinado que la cache se encuentra en el camino crítico del procesador, por lo que la frecuencia (y por tanto el tiempo de ciclo) del procesador vienen determinados por el diseño de la cache.

En la siguiente tabla, las filas indican posibles mejoras (vistas en clase) que podríamos introducir en L1. Las columnas indican algunos parámetros de rendimiento en los que estas mejoras podrían influir, los tiempos tanto de acceso como de penalización se consideraran en ns y no en ciclos ya que el tiempo de ciclo puede cambiar. Cada mejora será considerada de forma individual, sin cambiar ningún otro parámetro de la cache. Para indicar el impacto de las mejoras en los parámetros de rendimiento rellena la tabla usando la siguiente nomenclatura: **(M)** Mejora, **(NA)** No Afecta, **(E)** Empeora.

a) **Indica** el impacto de las mejoras en los parámetros de rendimiento.

Mejora/cambio	Parámetro de rendimiento			
	Frecuencia del procesador	Tiempo (ns) de acceso a la cache L1 en caso de acierto	Tasa de fallos de la cache L1	Tiempo (ns) medio de penalización en caso de fallo
Cambiar a cache Directa de 16 KBytes				
Añadir una cache L2 de 256 KBytes				
Aumentar el tamaño de la cache a 32 KBytes				
Añadir continuación anticipada				
Segmentar la cache en 2 etapas				
Organizar la cache en 4 bancos				
Añadir un mecanismo de prefetch hardware				
Hacer la cache no bloqueante				

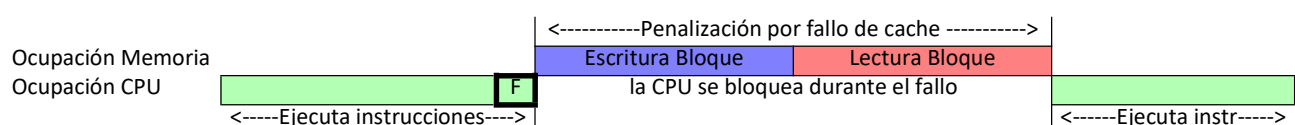
**Nota 1:** El apartado vale 1.5 puntos, cada fallo o cuadro en blanco resta 0.1 puntos en el apartado (mínimo 0).

**Nota 2:** La decisión de si un cambio mejora, empeora o no afecta a un parámetro debéis basarla en el caso general. Por ejemplo: aunque en determinadas circunstancias un predictor de vía podría empeorar el consumo de energía, en general lo mejorará.

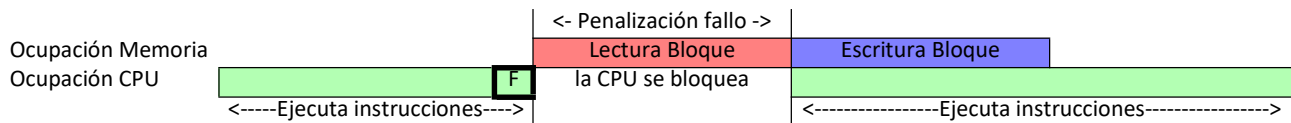
**Finalmente se ha decidido estudiar el impacto de las políticas de escritura.**

Se ha simulado la ejecución de un programa P en un procesador que denominaremos IDEAL. En este procesador IDEAL no hay ninguna penalización por fallo de cache. De esta simulación se ha obtenido que el programa P se ha ejecutado en  $5 \times 10^9$  ciclos durante los que ha ejecutado  $2 \times 10^9$  instrucciones, de las que  $500 \times 10^6$  son instrucciones de acceso a datos (Load/Store) y se han producido  $50 \times 10^6$  fallos en la cache de datos (los fallos en la cache de instrucciones son negligibles, con lo que los ignoraremos durante todo el problema).

El mismo programa lo ejecutamos en un procesador real con las mismas características que el IDEAL con la única diferencia que en caso de fallo en la cache de datos se bloquea la ejecución de instrucciones durante un cierto numero de ciclos hasta que se resuelve el fallo. La cache de datos sigue una política de escritura COPY BACK + WRITE ALLOCATE. En caso de fallo, si la línea reemplazada ha sido modificada, la penalización es de 30 ciclos, mientras que si no lo ha sido es sólo de 15 ciclos. Se sabe que durante la ejecución de P, en media 1/3 de los bloques reemplazados han sido modificados (dirty bit = 1). La siguiente figura muestra el cronograma de un fallo en que la línea reemplazada ha sido modificada.



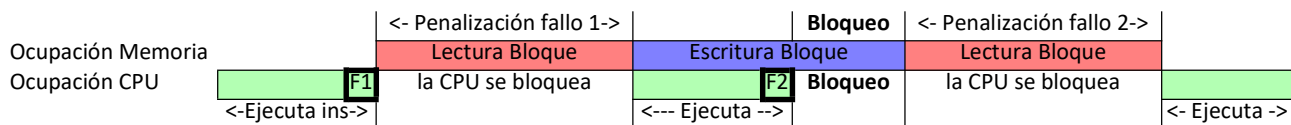
Una posible mejora (vista en clase de teoría) consiste en incorporar un buffer para almacenar el bloque reemplazado, de forma que podamos leer primero el bloque que ha provocado el fallo y a continuación escribir en memoria el bloque reemplazado. Esta implementación permite que el procesador pueda seguir ejecutando instrucciones y la cache pueda ser accedida durante la escritura del bloque reemplazado a memoria, tal como muestra la siguiente figura.



De momento supondremos la situación ideal (aunque no realista) en que nunca se produce un fallo de cache mientras se realiza la escritura del bloque reemplazado. A este procesador lo denominaremos procesador B

b) **Calculad** el numero de ciclos necesario para ejecutar P en el procesador B

En la implementación real (que denominamos procesador C) dispondremos de un buffer de una sola entrada que nos permite tener como máximo una escritura pendiente. Si durante la escritura del bloque causada por un fallo (F1) se produce un segundo fallo (F2), hay que esperar a que la jerarquía de memoria complete la escritura del bloque antes de que pueda empezar a servir el siguiente fallo, con lo que el procesador se bloqueará por unos ciclos adicionales (**Bloqueo**), tal como muestra la siguiente figura (en el ejemplo suponemos que el siguiente fallo reemplaza un bloque no modificado).



Durante la fase en que la CPU ejecuta instrucciones, todos los ciclos tienen la misma probabilidad de que se produzca un nuevo fallo y sabemos que el número medio de ciclos en que la CPU ejecuta instrucciones (sin contar los bloqueos) entre dos fallos es de 100 ciclos.

c) **Calculad** la probabilidad de que se produzca un segundo fallo durante la escritura de un bloque reemplazado

Hemos medido que, si se produce un segundo fallo durante el intervalo de escritura de un bloque anterior, en media la CPU se bloquea durante 7 ciclos (correspondientes al intervalo **Bloqueo** de la figura anterior) antes de que se pueda iniciar la lectura del bloque que ha causado el segundo fallo.

d) **Calculad** el numero de ciclos necesario para ejecutar P en el procesador C





- d) **Calcula** cuántos discos duros DD necesitaríamos para implementar cada uno de los niveles de RAID 10 (mirror doble), RAID 5 y RAID 6 si queremos almacenar al menos 24TB útiles

- e) Después de explicar a la empresa ACME las diferencias entre los tres niveles de RAID, la empresa se decanta por montar un RAID5 con 8 discos duros DD. Dibuja el esquema del RAID 5 que nos ha pedido la empresa indicando claramente el tipo de entrelazado, cómo se distribuye la paridad entre los discos, y el ancho de banda máximo que se conseguirá en el RAID5 en lectura secuencial, en escritura secuencial y en escritura aleatoria.

- f) **Calcula** el Speedup y el tiempo total que tarda la aplicación respecto a ejecutarla en el PC1 al ejecutarla en el PC6 + SIMD con un RAID 5 de 8 discos), teniendo en cuenta que todos los accesos a disco son secuenciales.