

COGNOMS:

[illegible]

NOM:

[illegible]

DNI/NIE:

--	--	--	--	--	--	--	--	--

IMPORTANTE: Escriba apellidos, nombre y DNI/NIE 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, todo lo que haya fuera de ellos es ignorado.

Problema 1. (3,5 puntos)

Tenemos un procesador conectado a un sistema de memoria que tiene una memoria principal que contiene instrucciones y datos. Llamamos S1-MP a este sistema. En este problema tendremos en cuenta solo el acceso a datos, y supondremos que el acceso a instrucciones no penaliza el tiempo de ejecución. Todas las instrucciones tardan 1 ciclo en ejecutarse, más el tiempo que tarden en acceder a la MP para acceder a los datos (en caso de que sea necesario este acceso). La penalización por acceso a la MP es de 120 ciclos. Ejecutamos en el sistema S1-MP un código que tiene un 10% de las instrucciones que realizan 1 acceso a datos, y un 20% de las instrucciones que realizan 2 accesos a datos.

- a) **Calcula** cuantos accesos por instrucción (en media) realiza este procesador a la MP en el sistema S1-MP para acceder a los datos del programa.

- b) **Calcula** el CPI del sistema S1-MP cuando ejecuta el código descrito.

Para mejorar el rendimiento, se ha decidido incorporar una cache de datos (L1D) entre el procesador y la MP. A este sistema le llamamos S2-L1D. La latencia de acceso de la L1D es de 2 ciclos. Esto significa que todos los accesos a la cache añaden 2 ciclos al tiempo de ejecución de la instrucción. La latencia de acceso a la MP sigue siendo de 120 ciclos. La política de escritura de la L1D es write-through - write-no-allocate. Un 40% de los accesos a memoria son escrituras.

- c) **Calcula** el CPI del sistema S2-L1D en función de la tasa de fallos de la L1D (tf).

- d) **Calcula** cuál sería la mayor tasa de fallos (tf) que podría soportar el sistema S2-L1D para que el código se ejecute en la mitad de tiempo que en el sistema S1-MP.

--

Después de un largo debate entre los ingenieros, la L1D implementada en el sistema S2-L1D tiene una tasa de fallos de 0,2, tanto para lecturas como para escrituras. Todas las instrucciones tienen un consumo base de 10nJ (sin acceder a memoria). La L1D tiene un consumo de 30nJ por acceso (de escritura o lectura). Un acceso a MP (tanto a bloque como a palabra) consume 400nJ.

e) **Calcula** el ahorro de energía medio por instrucción del sistema S2-L1D respecto al sistema S1-MP.

f) **Calcula** la potencia disipada por el sistema S2-L1D (en Vatios) si la frecuencia es de 2GHz.

Debido a la aparición del ataque Rowhammer, los diseñadores quieren introducir un buffer de escrituras entre la L1D y la MP para evitar que estos ataques realicen un acceso ilícito al sistema. A este sistema le llamamos S3-BUF. El buffer funciona como una cola FIFO, y va guardando en la cola todas las escrituras que realiza el programa. Una vez el dato se escribe en el buffer, el procesador sigue ejecutando el código (no espera a que el dato se escriba en memoria). Cuando la memoria está libre, el buffer escribe en la MP. En el caso de realizar una lectura de MP, primero se mira si los datos están en el buffer. Si están, se leen desde el buffer. Si no están, se accede a la MP después de haber accedido al buffer. Sólo un 10% de las lecturas que se realizan en el buffer encuentran el dato que buscaban. Cada acceso al buffer (de lectura o escritura) consume 50nJ.

g) **Calcula** la energía media por instrucción en el sistema S3-BUF.

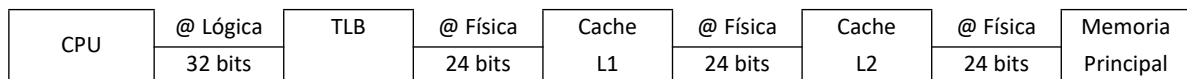
h) **Calcula** la latencia máxima de lectura (en ciclos) que debería tener el buffer en el sistema S3-BUF para que el rendimiento sea superior a 165 MIPS.

COGNOMS:

NOM: DNI/NIE:

Problema 2. (3,5 puntos)

Tenemos un procesador con un espacio de direccionamiento lógico de 4 Gbytes (direcciones lógicas de 32 bits). La unidad mínima de direccionamiento es un byte. La memoria virtual esta basada en traducción de direcciones mediante paginación, con paginas de 1 Mbyte. Para acelerar la traducción, el procesador tiene un TLB que genera direcciones físicas de 24 bits. El resto de la jerarquía de memoria se accede mediante direcciones físicas y está formado por una cache L1, una cache L2 y la memoria principal.



La siguiente tabla resume las principales características de cada uno de los componentes de la jerarquía:

	TLB	L1	L2
Tamaño	2 entradas	8 Kbytes	64 Kbytes
Políticas de escritura	---	Write Through + Write NO Allocate	Copy Back + Write Allocate
Asociatividad	Totalmente asociativo	2 asociativa	Mapeo Directo
Tamaño de Bloque	---	32 bytes	64 bytes
Algoritmo de reemplazo	LRU	LRU	---

En los siguientes apartados se asume que todos los accesos son a byte. En todas las tablas se describen los campos necesarios. Si un campo no se describe es porque ya se ha descrito en tablas anteriores. Para representar un valor en hexadecimal con un numero de bits no múltiplo de 4, asume ceros en los de más peso (101000 -> 28).

- a) **Dibuja** una dirección lógica con los campos de bits relevantes para la traducción de direcciones, indicando claramente el nombre y tamaño de cada uno de ellos.

- b) **Rellena** la siguiente tabla a partir de la secuencia de referencias a memoria. Sabemos el TLB está inicialmente vacío. La tabla de la derecha muestra el contenido de las 6 primeras posiciones de la tabla de paginas.

@lógica: Dirección lógica (en hexadecimal) generada por el procesador.
 L/E: El acceso es lectura (L) o escritura (E).
 VPN: Virtual Page Number (página lógica) (en hexadecimal).
 Desp: Desplazamiento dentro de la página (en hexadecimal).
 PPN: Physical Page Number (página física) (en hexadecimal).
 @física: Dirección física (en hexadecimal)
 A/F: Acierto de TLB (A) o fallo de TLB (F).

@lógica	L/E	VPN	Desp	PPN	@física	A/F
00000000	E					
00201111	L					
00002222	L					
00503333	E					
00204444	L					
00505555	E					

VPN	PPN	P	M
000	A	1	0
001	B	1	0
002	0	1	0
003	5	1	0
004	4	1	0
005	3	1	0

- c) **Dibuja** una dirección física con los campos de bits relevantes para el acceso a la cache L1, indicando claramente el nombre y tamaño de cada uno de ellos.

- d) **Rellena** la siguiente tabla a partir de la secuencia de referencias a memoria dada:

BM: Bloque de memoria que se accede (en hexadecimal).
 byte: Byte dentro del bloque (en hexadecimal).
 TAG: Etiqueta del acceso a cache (en hexadecimal).
 CJT: Conjunto de cache en que se mapea el bloque (en hexadecimal).
 A/F: Acierto de Cache L1 (A) o Fallo de Cache L1 (F).
 LMP: Numero de bytes leídos de L2 (**vacío si no se lee**).
 EMP: Numero de bytes escritos en L2 (**vacío si no se escribe**).

@física	L/E	BM	byte	TAG	CJT	A/F	LMP	EMP
000000	L							
444001	E							
888002	L							
000013	L							
444014	L							
444015	E							

- e) **Dibuja** una dirección física con los campos de bits relevantes para el acceso a la cache L2, indicando claramente el nombre y tamaño de cada uno de ellos.

- f) **Rellena** la siguiente tabla a partir de la secuencia de referencias a memoria dada:

LC: Bloque (línea) de cache en que se mapea el bloque de memoria (en hexadecimal).
 A/F: Acierto de Cache L2 (A) o Fallo de Cache L2 (F).
 LMP: Numero de bytes leídos de memoria principal (**vacío si no se lee**).
 EMP: Numero de bytes escritos en memoria principal (**vacío si no se escribe**).

@física	L/E	BM	byte	TAG	LC	A/F	LMP	EMP
000000	L							
000001	E							
440002	L							
008013	E							
008014	L							
448015	E							

COGNOMS:

NOM: DNI/NIE:

Problema 3. (3 puntos)

Hemos adquirido un disco duro para nuestro equipo doméstico y nos proponemos analizar algunas de sus características. La siguiente tabla muestra los parámetros más relevantes de nuestra compra:

platos	12 platos/disco
caras por plato	2 caras/plato
pistas	100000 pistas/cara
tamaño pista	64 sectores/pista
tamaño sector	512 bytes/sector
velocidad rotación (número de vueltas por minuto)	5400 r.p.m.
coste (precio final)	24,26 euros

- a) **Calcula** la capacidad del disco si se formatea con una densidad fija de 64 sectores por pista. Escribe el resultado en GB (potencias de 10). **Calcula** también el coste por GB.

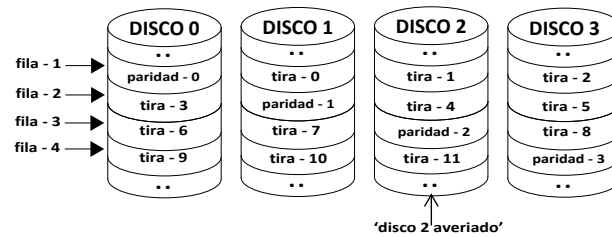
- b) **Calcula** el tiempo de transferencia de un sector. Este tiempo sólo depende de la velocidad de rotación del disco, del tamaño del sector y de la densidad de la pista. **Calcula** también el ancho de banda.

Dado el uso intensivo que damos a nuestro disco para grabación y reproducción de películas, cada vez necesitamos más capacidad de almacenamiento. En nuestra incursión en el manual descubrimos que con varios discos de características similares al que hemos comprado podemos crear nuevos sistemas de almacenamiento usando diferentes organizaciones RAID. Más concretamente, el proveedor nos propone una configuración RAID 0 con 4 discos de 300GB.

- c) **Completa** la siguiente tabla, indicando para cada una de las configuraciones RAID cuántos discos necesitaríamos para mantener la misma información útil que nos propone el proveedor (en GB), así como la cantidad de información redundante que tendremos en cada configuración (en %).

Configuración	número de discos	información útil (GB)	Información redundante (%)
RAID 0 (proveedor)	4		
RAID 10			
RAID 3			
RAID 4			
RAID 5			
RAID 6			

La siguiente figura muestra una posible configuración RAID 5 de 4 discos. En cada fila se ven las diferentes tiras de datos y de paridad.



Imagina que en la configuración RAID 5 de la figura se produce una avería del disco2 y cuándo aún no se ha recuperado el disco, llega una petición de lectura de todos los datos de una determinada fila.

- d) **Razona** si es posible realizar dicha lectura cuando es de los datos de la primera fila (fila-1). En caso afirmativo y si es necesario, indica cómo se recupera la información perdida en el disco 2. Repite tus respuestas para el caso de una lectura de la tercera fila (fila-3).

Dentro del manual del disco leemos que, cada interrupción que produce este dispositivo implica la ejecución de una rutina de servicio, que en un procesador CISC de la familia x86 requiere la ejecución de 2000 instrucciones dinámicas de lenguaje máquina x86. Nuestro procesador, que funciona a 1 GHz, es un Intel CISC x86. Sabemos que traduce internamente las instrucciones de lenguaje máquina x86 a microoperaciones (uops). Suponed que cada instrucción de la rutina se traduce en una uop y que cada 500 instrucciones CISC se necesitan 215 uops adicionales.

- e) **Calcula** el CPI y el tiempo de ejecución de la rutina de servicio de interrupciones para un valor de UPC (uops por ciclo) de 1,5.

Un ejemplo en el manual habla de un programa que tarda 2 horas en ejecutarse en un sistema con un solo procesador y 1 único disco. El programa está formado por una Fase de cálculo y una Fase de entrada/salida. El procesador tiene un rendimiento promedio, para este programa, de 500 MIPS. El manual dice que sólo la Fase de cálculo de este programa es totalmente paralelizable (sin necesidad de añadir ningún tipo de sincronización). Según el manual, el programa se acelera en 1,6x si en lugar de 1 disco se usa una configuración RAID 0 con 4 discos (suponed que todos los accesos son aleatorios y que son suficientes para saturar el ancho de banda de los 4 discos).

- f) **Calcula** a cuántos MIPS se ejecutará el programa en nuestro PC con 8 procesadores, idénticos al anterior, y una configuración RAID 0 de 4 discos como la descrita en el manual.