

APELLIDOS: _____ NOMBRE: Antón

Hojas adicionales adjuntas: _____

Normas: Escribe los apellidos y el nombre en cada hoja de examen y en cada hoja adicional que utilices. Es posible escribir la respuesta en la misma hoja de cada ejercicio, pero si utilizas hojas adicionales no mezcles en la misma hoja respuestas de ejercicios diferentes: se entregará cada ejercicio por separado. La respuesta a los ejercicios debe estar escrita con bolígrafo. Al terminar, indica en cada hoja de examen las hojas adicionales utilizadas para cada ejercicio.

- [1p] 1. El siguiente código se ejecuta en un procesador MIPS de 5 etapas como el estudiado en clase (IF, ID, EX, MEM, WB). Este procesador cuenta con una unidad de detección de riesgos en la etapa ID y una unidad de anticipación en la etapa EX. El salto se decide en la etapa ID y se utiliza una técnica de predicción de salto.

```
1      addi $t4, $0, 10
2 loop: lw $a1, 0($a0)
3      lw $t0, 0($a1)
4      sw $t0, 0($a0)
5      addi $t4, $t4, -1
6      addi $a0, $a0, 4
7      bne $t4, $0, loop
8      sw $t0, 0($a0)
```

Responde razonadamente a las siguientes cuestiones:

- (a) [0,25p] Identifica las dependencias y su tipo dentro del bucle en el código anterior. ¿Cuáles de estas dependencias provocarán un riesgo en la ejecución?
- (b) [0,25p] Discute la veracidad de la siguiente afirmación: “Si el procesador implementase salto retardado, se puede eliminar la instrucción 4 y, tras la ejecución, el array que recorre \$a0 tendrá los mismos valores que en el resultado original pero desplazados una posición.”.
- (c) [0,25p] Calcula la tasa de acierto del predictor de salto según los siguientes modos de procesamiento de salto que podría implementar el procesador: salto fijo no efectivo, predictor de salto de 1 bit y predictor de salto de 2 bits. En los modos de predicción dinámica, la predicción inicial es de salto no efectivo.
- (d) [0,25p] Para **cada una de las etapas** del pipeline distintas de EX, razona si sería posible **mover** a dicha etapa la unidad de anticipación y en caso afirmativo determina si el rendimiento sería mejor o peor que con la configuración del enunciado.

APELLIDOS: _____ NOMBRE: _____

Hojas adicionales adjuntas: _____

- [1p] 2. Una computadora dispone de un sistema de memoria cache de dos niveles. El primero nivel tiene un tiempo de acierto de 1 ciclo, mientras que el segundo nivel tiene una penalización por fallo de 50 ciclos. El tamaño de línea, tanto a nivel de cache como de memoria principal, es de 16 bytes. El tamaño de palabra es de 4 bytes. Considere el siguiente código:

```
int A[2*N];  
for( i = 0; i < N; ++i ) {  
    r += A[2*i];  
}
```

El tamaño de un `int` es de 4 bytes, y las variables escalares se almacenan en registros del micro. Responda a las siguientes preguntas:

- (a) [0,25p] Indíquese la tasa de fallos local de la cache de nivel 1.
- (b) [0,25p] Indíquese la tasa de fallos global de la cache de nivel 2.
- (c) [0,25p] Sabiendo que el tiempo medio de acceso a memoria durante la ejecución de este programa es de 31 ciclos, indique el tiempo de acierto de la cache de segundo nivel.
- (d) [0,25p] Supóngase un sistema sin caches en el que la matriz A se almacena en una memoria principal con entrelazado de orden inferior a nivel de palabra. Se sabe que en este caso se pueden atender simultáneamente hasta un máximo de cuatro accesos consecutivos del código anterior. Indíquese cuantos módulos componen el sistema.

APELLIDOS: _____ NOMBRE: _____

Hojas adicionales adjuntas: _____

- [1p] 3. Considera un sistema de memoria virtual paginado en 2 niveles. El tamaño del espacio virtual es de 256 TiB y el del espacio físico es de 4 GiB. El tamaño de página es de 64 KiB y se dispone de una TLB totalmente asociativa de 512 entradas. Las tablas de páginas de ambos niveles tienen el mismo número de entradas y la misma estructura. Cada entrada contiene, además del bit de residencia, 7 bits de control.
- (a) [0,3p] Si sabemos que la dirección virtual 0x395B 1880 51A9 se traduce en la dirección física 0xEDDB 51A9 y que la entrada correspondiente en la tabla de segundo nivel comienza en la dirección 0x9ED3 4980, escribe un posible contenido para las entradas de todas las tablas que permitan hacer esta traducción (indicando asimismo sus campos).
 - (b) [0,2p] Escribe los contenidos de las entradas en la TLB implicadas en la anterior traducción. Indica qué cambios se producirían si, justo a continuación, el procesador emite la dirección 0x395B 1880 51E9.
 - (c) [0,2p] Calcula el tamaño conjunto de todas las tablas de páginas de un proceso. Si el tamaño de página se incrementase a 16 MiB manteniendo el tamaño de los espacios virtual y físico y la información de control, ¿aumentaría o disminuiría el espacio ocupado por las tablas? Razona tu respuesta.
 - (d) [0,3p] Si en este ordenador tenemos una caché de 8 MiB asociativa por conjuntos con 16 vías y un tamaño de línea de 64 B, ¿podría aplicarse la técnica VIPT (índice virtual, etiqueta física)? Justifica tu respuesta.

APELLIDOS: _____ NOMBRE: _____

Hojas adicionales adjuntas: _____

[1p] 4. Tenemos un sistema con memoria virtual de páginas de 2 kiB que conecta la memoria principal con el almacenamiento secundario mediante un bus síncrono de 128 bits a 100 MHz. Las transferencias de las páginas están gestionadas por una DMA a través de este bus en bloques del tamaño de la anchura del bus, necesitando 280 ns para la lectura del primer bloque y 100 ns para el resto en una operación de lectura de una página. Tanto el envío de 128 bits a través del bus como el de la dirección al controlador del almacenamiento secundario requieren un ciclo. Las transferencias de datos leídos más recientemente pueden solaparse con la lectura de los siguientes. Sabiendo que la DMA realiza la lectura de una página en una única transacción, calcula:

- (a) [0,4p] Latencia para la lectura de una página.
- (b) [0,2p] Ancho de banda del bus para las operaciones de lectura de página desde el almacenamiento secundario.

El almacenamiento secundario tiene una capacidad neta de 8 TiB, repartidos en cuatro discos que permiten accesos concurrentes. Queremos tener un RAID con información redundante y solicitamos precios de discos a varios proveedores que nos proporcionan la siguiente información:

- Compañía MACWEL: discos de 2 TB, serie 3NV-0001 450€, serie 4NW-0001 490€.
- Compañía TEABATE: disco de 2 TB, serie WW-0001-YN1 440€.
- Compañía EastDigi: discos de 2 TB, serie ED-A-0001-W 430€, serie ED-A-0015-Y 460€, serie ED-B-1020-C 500€.

Teniendo en cuenta que las características técnicas de todos estos discos son similares y sabiendo que la inversión no puede superar los 2500€, contesta:

- (c) [0,2p] ¿Qué tipo de RAID implementarías con el mejor rendimiento? ¿Por qué?
- (d) [0,2p] ¿Cuál sería la combinación de discos que utilizarías y cuál sería su coste? Razona la elección.

1. (a) Dentro del bucle (líneas 2 a 7) hay las siguientes dependencias:

- $2 \rightarrow 3$: Dependencia verdadera (\$a1). Provoca riesgo RAW al no poder anticipar \$a1 hasta finalizar la etapa MEM.
- $3 \rightarrow 4$: Dependencia verdadera (\$t0). Provoca riesgo RAW al no poder anticipar \$t0 hasta finalizar la etapa MEM.
- $6 \rightarrow 2, 4$: Antidependencia (\$a0).
- $5 \rightarrow 7$: Dependencia verdadera (\$t4). Provoca riesgo RAW al tener que esperar a que \$t4 se escriba en el banco de registros.

En memoria hay una antidependencia entre 4 y 2 en la dirección [\$a0 + 0] que no se ha calificado negativamente.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
addi \$t4, \$0, 2	IF	ID	EX	ME	WB									
loop: lw \$a1, 0(\$a0)		IF	ID	EX	ME	WB								
lw \$t0, 0(\$a1)			IF	ID	—	EX	ME	WB	← RAW					
sw \$t0, 0(\$a0)				IF	—	ID	—	EX	ME	WB	← RAW			
addi \$t4, \$t4, -1					IF	—	ID	EX	ME	WB				
addi \$a0, \$a0, 4						IF	ID	EX	ME	WB				
bne \$t4, \$0, loop							IF	ID	—	EX	ME	WB	← RAW	

- (b) Es falso. El resultado del código sería totalmente diferente, porque la instrucción 8 del hueco de retardo sobrescribirá en cada iteración el valor que leerá la instrucción 2 en la siguiente iteración. Si en la instrucción 3 en \$t0 no se carga una dirección de memoria, el resultado será un error de acceso a memoria en la segunda iteración.
- (c) El salto se ejecuta 10 veces. Con salto fijo no efectivo sólo se acierta en la última ejecución: $TA = 1/10$. Con un predictor de 1 bit se fallará en la primera y la última ejecución del salto: $TA = 8/10$. Con un predictor de 2 bits se fallará en las 2 primeras y en la última ejecución: $TA = 7/10$.
- (d) IF En la etapa IF no tiene sentido colocar la unidad de anticipación porque todavía no se ha decodificado la instrucción y desconocemos los registros que utilizará.
 ID En la etapa ID, anticipando los valores a la salida del banco de registros, favorecería a la instrucción de salto (la única para la que esta opción es útil) y se elimina el riesgo entre 5 y 7. Sin embargo, en las dependencias con las instrucciones de carga de memoria (lw) es necesario esperar un ciclo adicional. El rendimiento por tanto es peor que en el código original.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
addi \$t4, \$0, 2	IF	ID	EX	ME	WB										
loop: lw \$a1, 0(\$a0)		IF	ID	EX	ME	WB									
lw \$t0, 0(\$a1)			IF	ID	—	—	EX	ME	WB						
sw \$t0, 0(\$a0)				IF	—	—	ID	—	—	EX	ME	WB			
addi \$t4, \$t4, -1							IF	—	—	ID	EX	ME	WB		
addi \$a0, \$a0, 4								IF	ID	EX	ME	WB			
										↓ \$t4					
bne \$t4, \$0, loop										IF	ID	EX	ME	WB	

MEM En esta etapa, anticipando el valor del registro a la entrada del banco de memoria, se elimina el riesgo entre 3 y 4 porque la instrucción sw podrá obtener el valor de \$t0 del registro de segmentación MEM/WB (la instrucción 4 necesita el valor de \$a0 en su etapa ID para calcular la dirección destino y \$t0 en su etapa MEM para guardarlo en memoria). El riesgo entre 2 y 3 se incrementa en un ciclo al no poder anticipar el valor de \$a1 a la ALU, que necesita la segunda lw para calcular la dirección de memoria a la que acceder. El rendimiento por tanto no varía.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
addi \$t4, \$0, 2	IF	ID	EX	ME	WB									
loop: lw \$a1, 0(\$a0)		IF	ID	EX	ME	WB								
lw \$t0, 0(\$a1)			IF	ID	—	—	EX	ME	WB					
										↓ \$t0				
sw \$t0, 0(\$a0)				IF	—	—	ID	EX	ME	WB				
addi \$t4, \$t4, -1							IF	ID	EX	ME	WB			
addi \$a0, \$a0, 4								IF	ID	EX	ME	WB		
bne \$t4, \$0, loop									IF	ID	—	EX	ME	WB

WB En la etapa WB no tiene sentido colocar la unidad de anticipación pues (entre otras cosas) no existen registros de segmentación posteriores desde los cuales se puedan anticipar datos.

2. El acceso al array es secuencial saltando un elemento de tipo entero en cada acceso (de cada 8 Bytes en el array se leen 4).
- Cada fallo en la cache L1 trae de la cache L2 16 bytes, que contienen datos útiles para esa iteración y la siguiente. La tasa local de fallos es por tanto 50%
 - Cuando ocurre un fallo en la cache L1 también ocurre en la cache L2, porque no se reutilizan datos en L2. Por tanto, la tasa local de fallos en L2 es del 100%. La tasa global de fallos es el número de fallos en L2 dividido por el número de accesos a L1. Entonces, esa tasa global es del 50% igual que en L1.
 - Para calcular el TMA del sistema caché necesitamos las tasas locales de fallos en L1 y L2, que son del 50% y 100% respectivamente. El tiempo de acceso a L1 y la penalización por fallo de L2 nos las proporciona el enunciado. Entonces tenemos que $1 + 0.5(t_{accesoL2} + 1 \times 50) = 31 \rightarrow t_{accesoL2} = 10$
 - En cada acceso a datos se cargan 4 palabras desde memoria principal a memoria cache. Esas 4 palabras son suficientes para completar 2 iteraciones del código. Como el entrelazamiento es de orden inferior a nivel de palabra, cada una de estas palabras se encontrará en un módulo diferente. Para que puedan accederse exactamente a los datos de 4 iteraciones de forma simultánea son necesarios por tanto el doble de módulos: necesitamos 8 módulos de memoria
3. (a) $|V| = 256 \text{ TiB} = 2^{48} B \Rightarrow \text{DV de 48 bits}$
 $|F| = 4 \text{ GiB} = 2^{32} B \Rightarrow \text{DF de 32 bits}$
 $|P| = 64 \text{ KiB} = 2^{16} B \Rightarrow \Delta \text{ de 16 bits}$

Si las tablas de páginas de ambos niveles tienen el mismo número de entradas, los dos campos para indexarlas tienen el mismo tamaño.

índ TP_1	índ TP_2	Δ	núm PF	Δ
395B	1880	51A9	EDDB	51A9
16 bits	16 bits	16 bits	16 bits	16 bits

La tabla de primer nivel nos permite localizar la tabla de segundo nivel correspondiente. Por tanto, en el número de página física almacenado tendremos la parte correspondiente a la dirección de entrada de la TP_2 que leemos.

	R	control	núm PF
$TP_1[395B]$	1	—	9ED3
	1 bit	7 bits	16 bits

Suponiendo 0s en los bits de control tendríamos almacenado el contenido 0x809ED3. La tabla de segundo nivel ya nos da el número de página física resultado de la traducción. Así:

	R	control	núm PF
$TP_{2,395B}[1880]$	1	—	EDDB
	1 bit	7 bits	16 bits

Análogamente, suponiendo 0s en los bits de control, tendríamos almacenado el contenido 0x80EDDB.

- (b) La TLB contiene pares página virtual-página física de las últimas traducciones.

núm PV	núm PF
395B 1880	EDDB
32 bits	16 bits

La dirección 0x395B 1880 51E9 se encuentra en la misma página: sólo cambian los bits dentro del desplazamiento Δ . Por tanto, la TLB no cambia.

- (c) $|TP_1| = 2^{16} \text{ entradas} \times (1 + 7 + 16) \text{ bits} = 2^{16} \times 3 \text{ B} = 192 \text{ KiB}$
Cada una de las TP_2 tiene el mismo tamaño, y tenemos una tabla para cada entrada de la TP_1
 $|TP| = |TP_1| + 2^{16} \times |TP_{2,i}| = |TP_1| \times (1 + 2^{16}) \approx 12 \text{ GiB}$

Si incrementamos el tamaño de página estamos reduciendo el número de páginas del sistema, por lo que las tablas tendrán menos entradas (por haber menos páginas virtuales) y estas entradas serán de menor tamaño (ya que necesitaremos menos bits para guardar el número de página física puesto que también tendremos menos páginas físicas).

- (d) $|C| = |ZA| = 8 \text{ MiB} = 2^{23} B$
 $|l| = 64 B = 2^6 B \Rightarrow \Delta_l \text{ de } 6 \text{ bits}$
 $\# \text{líneas} = |ZA|/|l| = 2^{23-6} = 2^{17}$
 $\# \text{conjuntos} = \# \text{líneas} / 16 \text{ vías} = 2^{17-4} = 2^{13} \Rightarrow \text{índice de } 13 \text{ bits}$

núm PF	Δ	etiqueta	índice	Δ_l
16 bits	16 bits	13 bits	13 bits	6 bits

Para poder aplicar la técnica VIPT, el índice que utilizamos para acceder a la caché tiene que coincidir dentro de los bits de desplazamiento de página Δ (la parte de la DV que no se traduce). En este caso, $16 \leq (13 + 6)$ hace que no sea posible.

4. (a) El bus de 128 bits indica que se leen 16 bytes de cada vez. Así, el número de lecturas para transferir una página sería:

$$NL = \frac{2 \text{ kiB}}{2^4 \text{ B}} = \frac{2^{11}}{2^4} = 2^7 = 128$$

La frecuencia del bus es de 100 MHz, con lo cual el período es:

$$T = \frac{1}{100 \times 10^6} = 10^{-8} \text{ s} = 10 \text{ ns}$$

La transferencia de una página requiere:

- 1 ciclo para enviar la dirección.
- 280 ns para leer los primeros 16 bytes. Corresponden a 28 ciclos.
- 100 ns para leer cada uno de los 127 grupos de 16 bytes restantes. Serían $10 \times 127 = 1270$ ciclos.
- 1 ciclo para transferir los últimos 16 bytes.

En total una página necesita $1 + 28 + 1270 + 1 = 1300$ ciclos, siendo la latencia de transferencia de una página:

$$L = 1300 \times 10 = 13000 \text{ ns} = 13 \mu\text{s}$$

- (b) El ancho de banda sería:

$$AB = \frac{2 \text{ kiB}}{13 \times 10^{-6} \text{ s}} \approx 0,154 \times 10^9 \frac{\text{B}}{\text{s}} = 154 \frac{\text{MB}}{\text{s}}$$

- (c) Para accesos concurrentes se emplean los RAID 4, 5 ou 6 usando como mínimo 5 discos. Para 6 discos, con estos precios necesitaríamos un mínimo de 2580€, superior al presupuesto disponible. Por tanto, habría que escoger entre RAID 4 y RAID 5, dando mejor rendimiento el RAID 5.
- (d) Para un RAID 5 tenemos que comprar 5 discos, y para mayor fiabilidad es preferible que sean de distinta serie, así que según los precios una posible elección sería escoger un disco de cada tipo excepto el ED-B-1020-C de EastDigi que es el más caro de esta casa, con un gasto final de 2270€.