

Ejercicio 2.

a.

	addi	x1, x0, 1000	# contador
	addi	x3, x0, 0	# primera posición de B
ETIQ	lw	x4, 0(x3)	# valor en B[x3]
	addi	x4, x4, -2	# le resta dos
	sw	x4, 0(x3)	# almacena el valor modificado
	addi	x3, x3, 4	# posición del siguiente valor de B
	addi	x1, x1, -1	# resta 1 al contador
	bnez	x1, ETIQ	# salta si el contador llegó a 0

b. Como el vector B tiene 1000 elementos, esto significa que ocupa 500 bloques (2 elementos en cada bloque) en memoria desde la dirección 0, la caché tiene un tamaño de 256 bloques, entonces para los primeros 512 elementos del vector se dan 256 fallos, que se dan cuando se leen los elementos pares pues no se ha subido el bloque, pero para el siguiente elemento el bloque ya estaría en caché. Los 488 elementos restantes del vector ocuparían 244 bloques de la caché empezando desde el primero, entonces habría 244 fallos más y como se escriben sobre los bloques que ya la caché los marcó como modificados, entonces se tienen que escribir estos en memoria, por lo que se escriben 244 bloques. En total se dan **500 fallos** y se **escriben 488 palabras**.

c. Se dan **500 fallos** igual que en el ejercicio anterior, pues cuando se accede a los elementos pares, el bloque no está en la caché pero ya para los elementos impares sí está. Como la estrategia es Write Through, las palabras que se modifican también deben modificarse en el nivel siguiente de la jerarquía, por lo que se dan **1000 escrituras**, una para cada elemento.

Ejercicio 3.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
<u>fadd</u> f1, f2, f3	IF	ID	A1	A2	A3	A4	M	WB												
<u>fsub</u> f4, f5, f1		IF	ID	A1	A1	A1	A1	A2	A3	A4	M	WB								
<u>fmul</u> f6, f7, f1			IF	ID	M1	M1	M1	M2	M3	M4	M5	M6	M7	M	WB					
<u>fsw</u> f1, 120(x0)				IF	ID	EX	M	M	WB											
<u>fadd</u> f9, f1, f6					IF	ID	ID	ID	A1	A1	A1	A1	A1	A1	A2	A3	A4	M	WB	
<u>fsw</u> f4, 80(x0)						IF	ID	EX	M	M	M	M	WB							

El CPI es $12/6 = 2$.

Ejercicio 4.

- a. El procesador posee 64 núcleos, puede manejar hasta 128 hilos. La caché L1 es de tamaño 4MB, la caché L2 de 32MB y la caché L3 de 256MB. Su uso principal está orientado a estaciones de trabajo donde se manejan cálculos a nivel técnico y científico.
- b. Los ocho canales de memoria indican que el procesador es capaz de acceder de manera simultánea a ocho módulos distintos de la memoria RAM, lo que aumenta la velocidad en que se acceden los datos que se solicitan de ese procesador a la memoria.

Ejercicio 5.

- a. La estrategia que tendría una mayor tasa de fallos de caché sería la de mapeo directo porque muchos bloques de la memoria podrían ocupar un mismo bloque en la caché, ya que el número de bloque en caché asignado a un bloque de memoria es fijo. La estrategia que produciría menos fallos sería la completamente asociativa porque cualquier bloque de memoria puede ir en cualquier bloque de la caché, y la estrategia de reemplazo usualmente favorece a que el bloque reemplazado sea uno que probablemente no se vuelva a ocupar más. Entre estas dos, la más lenta es la completamente asociativa pues en mapeo directo, con un simple cálculo se sabe cuál bloque en la caché ocuparía el bloque en memoria, sin embargo, en la estrategia completamente asociativa se requiere de recolectar mayor información para saber cuál bloque es el que va a ser reemplazado, esto sería saber cuál es el más viejo, cuál es el menos usado recientemente, etc. y esta información debe almacenarse en algún lado y luego accederse, por lo que también es más compleja.
- b.
 - I. Un núcleo superescalar es aquel que puede emitir y finalizar más de una instrucción en un ciclo de reloj.
 - II. Porque aunque no esté paralelizado, se pueden ejecutar más de una instrucción al mismo tiempo.
 - III. Un núcleo multihilo es aquel que posee la circuitería necesaria para manejar varios hilos, esto significa que debe ser capaz de manejar igualmente un contador de programa, registros, etc. para cada uno de estos hilos.
 - IV. Multihilo de grano grueso: en este caso, se empiezan a emitir instrucciones de un hilo particular, pero cuando este hilo tenga un retraso considerable, como por ejemplo un fallo de la caché, el procesador cambia de contexto al de otro programa (otro PC, valor de registros, etc.) y comienzan a ejecutarse instrucciones

con ese contexto. Multihilos de grano fino: en este caso, en cada ciclo de reloj se cambia de hilo, por lo que en un ciclo se emiten instrucciones únicamente de un hilo particular, para esto el procesador debe saber a qué hilo pertenecen las instrucciones. Multihilos simultáneos: en este caso, en cada ciclo de reloj es posible emitir instrucciones de hilos diferentes, no se podría implementar si el procesador no fuera superescalar pues se necesita que se emita más de una instrucción a la vez en caso de que sean de hilos diferentes.

- c. Utilizaría una tabla (en el procesador, como la TBL) que contenga, para cada registro, un bit 0 o 1 indicando si ya hay una instrucción ejecutándose que va a modificar ese registro (pero que aún no lo ha hecho), de manera que si una instrucción va a modificar un registro y la tabla indica un 1, entonces el registro destino de esa instrucción pasa a ser un registro temporal que posee el campo para el dato que se va almacenar, así como el dato de cuál registro del set original es el que va a modificar. Cuando la instrucción que modificó el bit en la tabla de 0 a 1 termina, se verifica en este set de registros temporales si alguno de esos modifica el registro que acaba de modificarse por la instrucción que en un principio se ejecutó de primero. Todos los registros temporales se indican como “desocupado” una vez que ya hayan cambiado el registro que originalmente iban a cambiar.



- d. La codificación de instrucciones se da para que se pueda comunicar al procesador que hay una instrucción que se quiere realizar, de manera que las instrucciones puedan pasar de unas que el programador entiende pero traducidas a unas que la máquina entiende. Es importante pues define cómo va a ser la circuitería necesaria para que el procesador decodifique instrucciones y las ejecute de manera más óptima.