

1. (2.5 puntos) Asumiendo la ejecución mostrada en el diagrama de instrucciones-ciclo adjuntado, en un procesador con ejecución fuera de orden y especulativo:

PC	Instruccion	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33		
_text	l.d f0,a(r28)	IF	I	AC	L1	L2	WB	C																												
salto	l.d f2,x(r1)		IF	I	AC	-	L1	L2	WB	C																										
4104	mul.d f4,f2,f0			IF	I	-	-	-	-	M1	M2	M3	M4	M5	M6	M7	WB	C																		
4108	l.d f6,y(r1)				IF	I	I	AC	L1	L2	WB	-	-	-	-	-	-	-	C																	
4112	sub.d f4,f4,f6					if	IF	I	-	-	-	-	-	-	-	-	-	A1	A2	A3	A4	WB	C													
4116	s.d f4,z(r1)						IF	I	AC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	L1	L2										
4120	dadd r1,r1,8							IF	I	E1	WB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C										
4124	bne r1,r2,salto								IF	I	-	E1	WB	-	-	-	-	-	-	-	-	-	-	-	-	-	C									
salto	l.d f2,x(r1)									IF	I	AC	L1	L2	WB	-	-	-	-	-	-	-	-	-	-	-	-	C								
4104	mul.d f4,f2,f0										IF	I	I	I	I	I	M1	M2	M3	M4	M5	M6	M7	WB	-	-	C									
4108	l.d f6,y(r1)											if	if	if	if	IF	I	AC	L1	L2	-	WB	-	-	-	-	-	C								
4112	sub.d f4,f4,f6																	IF	I	I	I	I	-	-	-	A1	A2	A3	A4	WB	C					
4116	s.d f4,z(r1)																		if	if	if	IF	I	I	I	I	I	AC	-	-	-	C	L1	L2		
4120	dadd r1,r1,8																									if	if	if	if	IF	I	E1	-	WB	-	C
4124	bne r1,r2,salto																													IF	I	-	-	E1	WB	C

Responde y **justifica detalladamente** cada una de las siguientes preguntas:

- Para cada ciclo de parada y/o conflicto estructural, indica en qué ciclo se produce y cual es su causa.
- Para la instrucción que ejecuta I en el ciclo 4, indica al final de qué ciclo se termina de calcular el resultado de la operación, al final de qué ciclo se escribe dicho resultado en el ROB y al final de qué ciclo se traslada el resultado al banco de registros.
- Suponiendo que el reorder bufer está inicialmente vacío y su primera entrada es la 0, considera el ciclo de reloj 13 y responde:
 - Indica el estado de los registros F0, F2, F4, F6 y R1 al final del ciclo. Para cada registro indica su valor y su etiqueta.
 - Muestra el contenido de la estaciones de reserva de suma y multiplicación en coma flotante que estén ocupadas al final del ciclo.
 - Muestra el contenido del ROB correspondiente a la instrucción `l.d f6,y(r1)` al final del ciclo.

Asume que el contenido inicial de los registros y la memoria es:

	r1	r2	r28	a+8192	x+8192	y+8192
valor	8192	8224	8192	5.0	1.0	-1.0

Los registros no mostrados contienen el valor 0.

Solución:

- Para cada ciclo de parada y/o conflicto estructural, indica en qué ciclo se produce y cual es su causa.
 - Ciclo 5. Conflicto por no tener un bufer de lectura disponible.
 - Ciclos 12, 13, 14, 15. Conflicto por no tener una estación de reserva disponible de multiplicación/división en coma flotante.
 - Ciclos 18, 19, 20. Conflicto por no tener estación de reserva disponible de suma/resta en coma flotante.
 - Ciclo 21. Conflicto por no tener disponible el bus común de datos (WB).
 - Ciclos 22, 23, 24, 25. Conflicto por no tener buffer de escritura disponible.
 - Ciclo 29. Conflicto por no tener disponible el bus común de datos (WB).
- Para la instrucción que se lanza en el ciclo 4, indica al final de qué ciclo se termina de calcular el resultado de la operación y en qué ciclo se escribe el resultado en el banco de registros.

- Se obtiene el resultado en el ciclo 15.
- Se escribe el ROB en el ciclo 16.
- Se escribe el resultado en el banco de registros en el ciclo 17.

c) Suponiendo que el reorder buffer está inicialmente vacío y su primera entrada es la 0, considera el ciclo de reloj 13 y responde:

- 1) Indica el estado de los registros F0, F2, F4, F6 y R1 al final del ciclo. Para cada registro indica su valor y su etiqueta.

	F0	F2	F4	F6	R1
rob		#8	#4	#3	#6
valor	5.00	1.00			8192

- 2) Muestra el contenido de la estaciones de reserva de suma y multiplicación en coma flotante que estén ocupadas al final del ciclo.

Est. reserva	Ocup.	Op	Q1	V1	Q2	V2	rob	result
a1	SI	-	#2			-1.0	#4	
m1	SI	*		1.0		5.0	#2	

- 3) Muestra el contenido del ROB correspondiente a la instrucción `l.d f6,y(r1)` al final del ciclo.

#	Ocup.	instr	Comple.	dest	value	Pred.
#3	SI	<code>l.d f6,y(r1)</code>	Sí	F6	-1.0	

□

- Las características de las unidades funcionales son las siguientes:

Se pretende evaluar el comportamiento del procesador ante el siguiente bucle, el cual calcula por aproximación la raíz cuadrada del operando c , depositando el resultado en q .

- Completa la tabla adjunta con el diagrama de instrucciones/tiempo correspondiente a la ejecución del programa. Incluye desde la instrucción `dadd r1,r0,r0` hasta la instrucción `mul.d f3,f2,f1` de la segunda iteración; y desde el ciclo 1 hasta el ciclo en el que se realiza la fase C de la instrucción `mul.d f3,f2,f1` de la segunda iteración. Considera que el predictor acierta y que al inicio de la ejecución el ROB y las estaciones de reserva están vacías, y que los registros no tienen marcas.
- Indica el tiempo que consume una iteración cuando el predictor acierta.
- Sabiendo que en la última iteración el predictor falla, incurriendo en una *penalización* de 36 ciclos de reloj, y que el bucle mostrado realiza 4 iteraciones, calcula el tiempo de ejecución *total* del programa. Incluye la ejecución de las instrucciones de inicialización y de la instrucción `s.d f1,q($gp)` hasta que realiza la etapa C. Recuerda que en la primera iteración el predictor acierta.

a) Completa la tabla adjunta con el diagrama de instrucciones/tiempo correspondiente.

[illegible]

3. (2.5 puntos) Un procesador con arquitectura load/store y ejecución en orden, trabaja con un reloj a 1 GHz y dispone de dos niveles de memoria cache.

El primer nivel dispone de memorias cache separadas para instrucciones y datos, 32 KB en cada caso. Para la cache de instrucciones se observa una tasa de fallos de lectura del 1.2 %. Para la cache de datos se observan unas tasas de fallos de lectura y escritura del 3 % y del 4 % respectivamente. Ambas tienen un tiempo de acierto de 1 ciclo de procesador.

La memoria cache de segundo nivel tiene un tamaño de 128 KB, un tiempo de acierto de 10 ciclos de procesador y una tasa de fallos del 45 %.

El tiempo promedio necesario para obtener un bloque de memoria principal es de 100 ciclos de procesador y se ha observado que el CPI es de 1.9 cuando no se producen fallos de cache.

Finalmente, todas las medidas se han obtenido con una carga de prueba formada por un programa que ejecuta 100 millones de instrucciones donde un 25 % son instrucciones load y un 15 % instrucciones store.

Se plantean dos modificaciones con la intención de mejorar las prestaciones:

- 1) Introducir buffers de escritura para la cache de primer nivel.
- 2) Sustituir la memoria cache de segundo nivel por otra memoria cache que tiene una tasa de fallos del 35 % y un tiempo de acierto de 15 ciclos de procesador.

Se pide:

- a) Calcula el tiempo de ejecución en segundos del programa de prueba para la configuración original del procesador.
- b) ¿Las modificaciones propuestas mejoran realmente las prestaciones del procesador? Cuantifica para cada caso la mejora, si procede.
- c) La nueva memoria cache introducida en la modificación 2), ¿será de mayor o menor tamaño que la original? Justifica tu respuesta.
- d) Si se hubieran realizado las medidas usando un programa de prueba distinto, ¿cuáles de los siguientes valores hubieran podido variar, aunque fuera sutilmente? Justifica tu respuesta.
 - Tasas de fallos en L1 y L2
 - CPI en ausencia de fallos de cache
 - Tiempo promedio acceso bloque en memoria principal
 - Número de instrucciones ejecutadas
 - Porcentaje de instrucciones load y store
 - Tiempo de acierto en cache L1 y L2

Solución:

- a) Calcula el tiempo de ejecución en segundos del programa de prueba para la configuración original del procesador.

$$T_e = T_{CPU} + T_I + T_{D_R} + T_{D_W}$$

$$T_{CPU} = I \times CPI \times T = 100 \cdot 10^6 \times 1,9 \times 10^{-9} = 0,19 \text{ segundos}$$

$$PF_{L1} = TA_{L2} + TFL2 \times PF_{L2} = 10 + 0,45 \times 100 = 55 \text{ ciclos}$$

$$PF_{L1} = PF_I = PF_R = PF_W$$

$$T_I = I \times API_I \times TFI \times PF_I \times T = 100 \cdot 10^6 \times 1 \times 0,012 \times 55 \times 10^{-9} = 0,066 \text{ segundos}$$

$$T_{D_R} = I \times API_R \times TFR \times PFR \times T = 100 \cdot 10^6 \times 0,25 \times 0,03 \times 55 \times 10^{-9} = 0,041 \text{ segundos}$$

$$T_{DW} = I \times API_W \times T_{FW} \times PF_W \times T = 100 \cdot 10^6 \times 0,15 \times 0,04 \times 55 \times 10^{-9} = 0,033 \text{ segundos}$$

$$T_e = T_{CPU} + T_I + T_{DR} + T_{DW} = 0,19 + 0,066 + 0,041 + 0,033 = 0,33 \text{ segundos}$$

- b) ¿Las modificaciones propuestas mejoran realmente las prestaciones del procesador? Cuantifica para cada caso la mejora si procede.

- 1) Al introducir buffers de escritura la $PF_W = 0$, por lo tanto el término $T_{DW} = 0$ y queda,

$$T_e = T_{CPU} + T_I + T_{DR} = 0,19 + 0,066 + 0,041 = 0,297 \text{ segundos}$$

$$S = \frac{0,33}{0,297} = 1,111 \rightarrow \text{Hay una mejora del } 11,1 \%$$

- 2) En este caso cambia,

$$PF_{L1} = T_{AL2} + T_{FL2} \times PF_{L2} = 15 + 0,35 \times 100 = 50 \text{ ciclos}$$

$$T_I = 0,06 \text{ segundos}$$

$$T_{DR} = 0,375 \text{ segundos}$$

$$T_{DW} = 0,3 \text{ segundos}$$

$$T_e = T_{CPU} + T_I + T_{DR} + T_{DW} = 0,19 + 0,06 + 0,038 + 0,03 = 0,318 \text{ segundos}$$

$$S = \frac{0,33}{0,318} = 1,038 \rightarrow \text{Hay una mejora del } 3,8 \%$$

- c) La nueva memoria cache introducida en la modificación 2), ¿será de mayor o menor tamaño que la original? Justifica tu respuesta.

Podría ser de mayor tamaño por un lado porque la tasa de fallos se reduce. Esto es típico de una memoria más grande que puede contener más datos y por tanto mejorar los principios de localidad. Y por otro lado aumenta el tiempo de acierto, y sabemos que las memorias más grandes son más lentas al necesitar más puertas para implementarse.

Otra posibilidad es que sea de igual tamaño, pero con una correspondencia mas compleja, por ejemplo asociativa por conjuntos con mayor número de vías, o completamente asociativa. En tal caso el tiempo de acierto aumenta, reduciéndose la tasa de fallos.

- d) Si se hubieran realizado las medidas usando un programa de prueba distinto, ¿cuáles de los siguientes valores hubieran podido variar, aunque fuera sutilmente? Justifica tu respuesta.

Hubieran podido modificarse las tasas de fallos y el CPI ya que son valores promedio que dependen del número de instrucciones ejecutadas y de como se comporta el procesador y la cache durante la ejecución de las mismas (ciclos de parada, instrucciones canceladas, fallos de cache, ...)

El número de instrucciones del programa de carga, así como la distribución de instrucciones load y store.

El tiempo para obtener un bloque de memoria principal podría verse modificado también si consideramos la posibilidad de que el controlador de memoria gestione las filas abiertas.

4. (2.0 puntos) Un procesador implementa un controlador de memoria con tres canales. En la configuración actual solo uno de ellos tiene conectado un DIMM DDR4, el resto de los canales están vacíos. El buffer de fila del DIMM es de 8KB y los parámetros de temporización son 18-18-18 con el bus a 3200Mhz.

El procesador dispone de una jerarquía de cache de 3 niveles, siendo la cache L3 la de último nivel. El tamaño de bloque en todas las caches es de 64B.

En dicho computador se ejecuta el *benchmark A* que accede a una matriz de 32KB (datos de coma flotante de doble precisión) que se almacena en filas de memoria consecutivas de un mismo banco.

Se pide:

- Calcula la tasa de aciertos en el buffer de fila cuando se ejecuta el benchmark A.
- Asumiendo una tasa de acierto en el buffer de fila (TA_{bf}) del 99 %, calcula el tiempo medio que tardaría en leerse un bloque de cache en ciclos de bus desde que el controlador lanza la petición al módulo hasta que finaliza la transferencia.
- Indica cuáles son las posibles razones por las que una petición puede esperar en el controlador antes de lanzarse al módulo.
- Asumiendo que la política del controlador es First Come First Served (FCFS), obtén una tabla para la siguiente secuencia de peticiones. La tabla debe tener las siguientes columnas: i) orden de planificación, ii) petición planificada, iii) acierto o fallo en buffer de fila, y iv) penalización de fallo en ciclos de bus. Asume que todas las peticiones van al mismo banco y que la fila que se encuentra abierta es la 79. Nota: la penalización media debe calcularse como la media aritmética de las 4 penalizaciones. NO utilices la fórmula.

Petición	T. llegada (ciclo bus)	Fila	columna
1	1	100	28
2	15	100	44
3	20	80	30
4	50	100	40

- Describe el funcionamiento de la política de planificación First Ready - First Come First Served (FR-FCFS). Indica si esta política mejoraría la penalización media para la secuencia de peticiones anterior. En caso afirmativo, calcula la TA_{bf} y la nueva penalización.

Solución:

- La tasa de acierto en el buffer de fila es independiente del número de filas que ocupa, hay un fallo por fila y el resto aciertos. Como en la fila caben $8KB/64B=128$ bloques, entonces la tasa de aciertos es $127/128$.
- Tiempo medio de acceso a memoria desde el controlador en ciclos de bus:

$$PF = L \cdot (1 - TA_{bf}) + L_r \cdot TA_{bf} + \frac{B}{B_w} \text{ ciclos de bus} = 54 \cdot 0,01 + 18 \cdot 0,99 + \frac{64B}{8B \cdot 2} = 22,36 \text{ ciclos}$$

- Las posibles razones son: i) tiempo de espera porque el banco destino está ocupado (contención de banco) y ii) tiempo de espera porque bus está ocupado.
- Representa i) orden de planificación, ii) petición planificada, iii) acierto o fallo en buffer de fila, y iv) penalización de fallo en ciclos de bus.
- Si mejoraría. La petición 3 y la 4, intercambiarían el orden. La nueva tasa de aciertos sería del 50 %.

$$PF = \frac{58 + 22 + 22 + 58}{4} = 40 \text{ ciclos}$$

Orden	Petición planificada	Buffer de filla ?	PF
1º	1	fallo	54 + 4
2º	2	acierto	18 + 4
3º	3	fallo	54 + 4
4º	4	fallo	54 +4
Penalización media			49 ciclos