

Responde cada pregunta en una hoja distinta. Tiempo disponible: 1 hora 45 minutos

1. (6 puntos) Considera el siguiente fragmento de código:

```
.data                ; Comienzo de los datos de memoria
n:                  .dword 80          ; Número de iteraciones * 8

x:                  .double 0.0,1.0,2.0,...,8.0,9.0 ; Vector x (10 doubles)
y:                  .space 80          ; Vector y (10 doubles)

.text 0x842000

                    ; Todos los registros valen 0 inicialmente
                    ld r4, n(r0)       ; r4 contendrá n
bucle:              ; do
                    ld f2, x(r1)      ; f2 = x[i]
                    mul.d f2, f2, f2   ; f2 = x[i]^2
                    add.d f2, f2, f2   ; f2 = 2* x[i]^2
                    s.d f2, y(r1)     ; y[i] = 2* x[i]^2
                    dadd r1, r1, #8    ; i = i+8
                    bne r1, r4, bucle  ; while i<n;
                    trap 0             ; fin
```

La ejecución de la última iteración de este código en una de las versiones del simulador del MIPS disponibles proporciona el siguiente diagrama instrucciones-tiempo:

PC	Instruc.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
.text	ld r4,n(r0)	IF	X														
bucle	ld f2,x(r1)	IF	I	AC	L1	L2	WB					C					
bucle + 04	mul.d f2,f2,f2		IF	I				M1	M2	M3	WB	C					
bucle + 08	add.d f2,f2,f2		IF	I								A1	A2	WB	C		
bucle + 12	s.d f2,y(r1)			IF	I	AC									C	L1	L2
bucle + 16	dadd r1,r1,#8			IF	I	E1	WB									C	
bucle + 20	bne r1,r4,bucle				IF	I		E1	WB							C	
bucle + 24	trap 0				IF	X											
.text	ld r4,n(r0)					IF	X										
bucle	ld f2,x(r1)					IF	I	AC	L1	L2	WB					x	
bucle + 04	mul.d f2,f2,f2						IF	I				M1	M2	M3	WB	x	
bucle + 08	add.d f2,f2,f2						IF	I	I							X	
bucle + 12	s.d f2,y(r1)							if	IF	I	AC					x	
bucle + 16	dadd r1,r1,#8							if	IF	I	E1	WB				x	
bucle + 20	bne r1,r4,bucle									IF	I		E1	WB		x	
bucle + 24	trap 0									IF	X						
.text	ld r4,n(r0)										IF	X					
bucle	ld f2,x(r1)										IF	I	AC	L1	L2	X	
bucle + 04	mul.d f2,f2,f2											IF	I			x	
bucle + 08	add.d f2,f2,f2											IF	I	I		x	
bucle + 12	s.d f2,y(r1)												if	IF	I	x	
bucle + 16	dadd r1,r1,#8												if	IF	I	X	
bucle + 20	bne r1,r4,bucle														IF	X	
bucle + 24	trap 0														IF	X	
.text	ld r4,n(r0)															X	
bucle	ld f2,x(r1)															X	
bucle + 20	bne r1,r4,bucle																IF
bucle + 24	trap 0																IF

Responde a las siguientes cuestiones razonando tus respuestas:

- ¿Por qué se buscan sistemáticamente las instrucciones *ld r4, n(r0)* y *trap 0* a pesar de no formar parte del bucle?
- ¿Por qué en el ciclo 12 se bloquea la búsqueda de nuevas instrucciones y por qué en el ciclo 13 se recupera la ejecución normal del bucle?
- ¿Por qué la etapa *EI* de una de las instrucciones *bne r1, r4, bucle* que aparecen en el diagrama se realiza en el ciclo 7 y no en el 6?
- ¿Qué ocurrirá en el ciclo 17 con cada una de las dos instrucciones (*bne* y *trap*) buscadas en el ciclo 16? Recuerda que la iteración representada en el diagrama es la última.
- ¿Se libera en algún momento algún buffer de escritura en el diagrama mostrado? En caso afirmativo indica el ciclo en el que esto ocurre y el PC de la instrucción involucrada.
- Considerando el diagrama mostrado, ¿Cuál será el valor y las marcas de los registros R1, R4 y F2 al final del ciclo en el que la instrucción *mul.d f2,f2,f2* es confirmada? Recuerda que al iniciar el bucle todos los registros se inicializan a 0. Asume que al comienzo de la última iteración representada en el diagrama, el ROB está vacío (la primera entrada disponible será la 0) y que no existen marcas en ningún registro.
- Asumiendo que el diagrama suministrado es representativo de una iteración, la última, del bucle ¿Cuál será el número de ciclos consumido por una iteración cuando el predictor de fallos falle? ¿Y cuando acierte?
- Realiza el diagrama instrucciones-tiempo de la **última iteración** del bucle cuando éste se ejecute en un procesador de 4 vías similar al anterior, pero en el que se dispone de 4 operadores no segmentados de cada tipo y de 16 estaciones de reserva para cada tipo de operador. En el caso de la unidad de carga y almacenamiento se dispondrá de 8 tampones de lectura y 8 de escritura. Asume que el procesador utiliza un predictor de tipo BTB de 2 bits. Por simplicidad, supongase también que el ROB tiene suficientes entradas para albergar todas las instrucciones a lanzar, y que tanto él como las estaciones de reserva están vacíos antes de la iteración solicitada, con lo que no hay ninguna instrucción pendiente de ejecución ni marca en los registros del procesador. No incluyas en el diagrama más instrucciones de las que te quepan en la plantilla de respuesta suministrada.

Solución:

- Por el lanzamiento alineado de instrucciones que realiza
- Porque el sumador se queda sin estaciones de reserva para lanzar a ejecución la instrucción *add.d f2, f2, f2*. En el ciclo 12 tenemos en ejecución una suma del primer bucle representado y la del segundo está en espera. Esto significa que sólo se dispone de 2 estaciones de reserva. En el ciclo 13, la suma del primer bucle representado en el diagrama termina su ejecución liberando su estación de reserva durante la etapa de WB. Esta estación es recuperada por la suma de la tercera iteración que puede ya ser lanzada a ejecución, desbloqueando el pipeline.
- Por una dependencia de datos. Hasta que la instrucción anterior no proporciona el valor de R1, la instrucción de salto no puede ser ejecutada.
- La instrucción *bne* se cancelará y la instrucción *trap* se lanzará a ejecución, tal y como se muestra a continuación:

PC	Instruccion	15	16
20	<i>bne r1, r4, bucle</i>	IF	X
24	<i>trap 0</i>	IF	I

- Sí, el buffer de escritura se libera al concluir la ejecución de la instrucción de almacenamiento con PC igual a *bucle+12* al finalizar el ciclo 16, con lo que el buffer estará disponible para uso en el ciclo 17.
- Valores y Marcas para los registros solicitados:

	R1	R4	F2
Valor	72	80	81.0
Marca	#10	-	#12

- 15 ciclos en caso de fallo y 4 ciclos en caso de acierto.
- Diagrama instrucciones-tiempo de la **última iteración** un procesador de 4 vías similar al anterior.

PC	Instruccion	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
.text	ld r4,n(r0)	IF	X														
bucle	l.d f2,x(r1)	IF	I	AC	L1	L2	WB	C									
8	mul.d f2,f2,f2	IF	I	-	-	-	-	M1	M2	M3	WB	C					
12	add.d f2,f2,f2	IF	I	-	-	-	-	-	-	-	-	A1	A2	WB	C		
16	s.d f2,y(r1)		IF	I	AC	-	-	-	-	-	-	-	-	-	C	L1	L2
20	dadd r1,r1,#8		IF	I	E1	WB	-	-	-	-	-	-	-	-	C		
24	bne r1,r4,bucle		IF	I	-	-	E1	WB	-	-	-	-	-	-	C		
28	trap 0		IF	X													
.text	ld r4,n(r0)				IF	X											
bucle	l.d f2,x(r1)				IF	I	-	AC	L1	L2	WB	-	-	-	-	X	
8	mul.d f2,f2,f2				IF	I	-	-	-	-	-	M1	M2	M3	WB	X	
12	add.d f2,f2,f2				IF	I	-	-	-	-	-	-	-	-	-	X	

□

2. (3 puntos) Sea un computador con los siguientes componentes:

- Un procesador con un ancho de palabra de 64 bits, frecuencia de funcionamiento de 1 GHz.
- Memoria cache L1 unificada (datos e instrucciones) de 64KB de tamaño. El tamaño de bloque es de 32 bytes y la correspondencia asociativa por conjuntos de cuatro vías. El tiempo de acceso de la cache es de 1 ciclo.
- Memoria cache L2 de 256KB de tamaño. El tamaño de bloque es de 32 bytes y la correspondencia asociativa por conjuntos de 8 vías. El tiempo de acceso de la cache es de 6 ciclos.
- La memoria principal es SDRAM DDR con temporización $CL - T_{rcd} - T_{rp} = 2 - 4 - 4$, funcionando a la misma frecuencia del procesador. Tiene un ancho de palabra de 64 bits.

En este computador se ejecuta un código de 1 millón de instrucciones (10^6), el cual experimenta un CPI de 1.5, provoca una tasa de fallos en L1 del 6 %, una tasa de fallos en L2 del 25 %, y una localidad en los accesos a memoria en fila abierta del 85 % (ML). El promedio de instrucciones que acceden a memoria por datos (lecturas y escrituras) es del 30 %.

Calcula:

- La penalización por fallo experimentado por la cache L2
- La penalización por fallo experimentado por la cache L1
- Tiempo de acceso medio al sistema de memoria (incluyendo L1, L2, y memoria)
- Tiempo de ejecución del código en ciclos y en segundos
- Indica como variaría la penalización por fallo de la cache L2 si soportara la técnica Critical Word First (CWF). Justifica tu respuesta.

Solución:

- a) La penalización por fallo experimentado por la cache L2

Latencia ante fila cerrada en memoria: $L_{fc} = CL + T_{rcd} + T_{rp} = 10 \text{ ciclos} = 10 \text{ ns}$

Latencia ante fila abierta en memoria: $L_{fa} = CL = 2 \text{ ciclos} = 2 \text{ ns}$

Tamaño de bloque de memoria: $B = 32 \text{ bytes} = 4 \text{ words}$

Ancho de banda de la memoria: $B_{wc} = 2 \text{ words/ciclo}$

Localidad fila abierta: $ML = 0,85$

Tiempo de acceso a memoria:

$$T_{mp} = ML \times L_{fa} + (1 - ML) \times L_{fc} + B/B_{wc} = 0,85 \times 2 + 0,15 \times 10 + 4/2$$

$$T_{mp} = 1,7 + 1,5 + 2 = 5,2 \text{ ciclos} = 5,2 \text{ ns}$$

Penalización por fallo de la L2:

$$PF_{L2} = T_{mp} = 5,2 \text{ ciclos} = 5,2 \text{ ns}$$

- b) La penalización por fallo experimentado por la cache L1

Tiempo de acceso L2: $TA_{L2} = 6 \text{ ciclos}$

Tasa de fallos L2: $TF_{L2} = 0,25$

Penalización por fallo en L2: $PF_{L2} = 5,2 \text{ ciclos}$

Penalización por fallo en L1: $PF_{L1} = TA_{L2} + TF_{L2} \times PF_{L2}$

$$PF_{L1} = 6 + 0,25 \times 5,2 = 7,3 \text{ ciclos} = 7,3 \text{ ns}$$

- c) Tiempo de acceso medio al sistema de memoria (incluyendo L1, L2, y memoria)

Tiempo de acceso L1: $TA_{L1} = 1 \text{ ciclo}$

Tasa de fallos L1: $TF_{L1} = 0,06$

Penalización por fallos en L1: $PF_{L1} = 7,3 \text{ ciclos}$

Tiempo de acceso: $T_{acceso} = TA_{L1} + TF_{L1} \times PF_{L1}$

$$T_{acceso} = 1 + 0,06 \times 7,3 = 1,438 \text{ ciclos} = 1,438 \text{ ns}$$

- d) Tiempo de ejecución del código en ciclos y en segundos

Número de instrucciones: $I = 10^6 \text{ instrucciones}$

Ciclos por instrucción: $CPI = 1,5$

Número medio de accesos por instrucción: $API = 1,3$

Tiempo de ciclo: $T = 1 \text{ ns}$

Tiempo de ejecución:

$$T_{ejecucion} = I \times CPI \times T + I \times API \times TF_{L1} \times PF_{L1} \times T$$

$$T_{ejecucion} = 10^6 \times 1,5 \times 1 \text{ ns} + 10^6 \times 1,3 \times 0,06 \times 7,3 \times 1 \text{ ns}$$

$$T_{ejecucion} = 0,0015 + 0,0005694 \text{ s} = 0,0020694 \text{ s} = 2,06 \text{ ms}$$

- e) Indica como variaría la penalización por fallo de la cache L2 si soportara la técnica Critical Word First (CWF). Justifica tu respuesta.

Latencia ante fila cerrada en memoria: $L_{fc} = CL + T_{rcd} + T_{rp} = 10 \text{ ciclos} = 10 \text{ ns}$

Latencia ante fila abierta en memoria: $L_{fa} = CL = 2 \text{ ciclos} = 2 \text{ ns}$

Tamaño de bloque de memoria: $B = 32 \text{ bytes} = 4 \text{ words}$

Ancho de banda de la memoria: $B_{wc} = 2 \text{ words/ciclo}$

Localidad fila abierta: $ML = 0,85$

Tiempo de acceso a memoria:

$$T_{mp} = ML \times L_{fa} + (1 - ML) \times L_{fc} + 1 \text{ ciclo} = 0,85 \times 2 + 0,15 \times 10 + 1$$

$$T_{mp} = 1,7 + 1,5 + 1 = 4,2 \text{ ciclos}$$

$$PF_{L2} = T_{mp} = 4,2 \text{ ciclos} = 4,2 \text{ ns}$$

Se reduciría un ciclo la PF_{L2}

□

3. (1 punto) Sea un procesador cuya cache tiene una tasa de fallos del 3 % y una penalización por fallos de 40 ciclos. La cache implementa un predictor de vía que acierta en el 80 % de los casos. En caso de acierto la cache es accedida en 1 ciclo, en caso de fallo, es accedida en 2 ciclos.

- a) Explica brevemente las ventajas que ofrece la predicción de vía
b) Calcula el tiempo medio de acceso a memoria

Solución:

a) Ventajas de la predicción de vía:

- Reduce los fallos por conflicto sin aumentar el tiempo de acierto (cuando acierta la predicción) sobre una de correspondencia directa.
- Reduce el tiempo medio de acceso y la energía respecto a una asociativa.

b) Tiempo medio de acceso a memoria:

$$TA_{L1} = 0,8 \times 1 + 0,2 \times 2 = 0,8 + 0,4 = 1,2$$

$$TF_{L1} = 0,03$$

$$PF_{L1} = 40$$

$$T_{acceso} = TA_{L1} + TF_{L1} \times PF_{L1} = 1,2 + 0,03 \times 40 = 2,4 \text{ ciclos}$$

□