

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

1. (4 puntos) Se dispone de un procesador MIPS superescalar de 2 vías con ejecución fuera de orden y especulación hardware basada en el algoritmo de Tomasulo. Las instrucciones atraviesan las siguientes etapas: IF (búsqueda de instrucciones), I (decodificación y lanzamiento de las instrucciones), En (ejecución en el operador multiciclo correspondiente), WB (transferencia del resultado por los buses comunes de datos) y C (confirmación de las instrucciones). El procesador dispone de un predictor de saltos del tipo *Branch Target Buffer* (BTB) de 1 bit que ofrece la predicción al final de la etapa IF.

Las características de los operadores son las siguientes:

	Nº Operadores	Latencia	Características
Carga/Almacenamiento	2	2	Segmentada; 4 buffers de lectura y 4 de escritura
Suma/Resta CF	2	2	No segmentada; 4 estaciones de reserva
Multiplicador CF.	2	3	Segmentada; 4 estaciones de reserva
Enteros/Saltos	2	1	8 estaciones de reserva

Se pretende evaluar el comportamiento del procesador ante el siguiente bucle, el cual es el resultado de aplicar loop unrolling al bucle DAXPY visto en prácticas:

$$z_i = a \cdot x_i + y_i \quad (1)$$

```

.data
x: .double 1.0,2.0,3.0, ...
y: .double 1.0,2.0,3.0, ...
z: .space 512
a: .double 1.0
.text
start: dadd r1,r0,x
       dadd r2,r0,y
       dadd r3,r0,z
       l.d f0,a(r0)
       dadd r4,r1,#512      ; r1 apunta a x
loop:  l.d f2,0(r1)
       l.d f6,8(r1)
       mul.d f2,f0,f2
       mul.d f6,f0,f6
       l.d f4,0(r2)
       l.d f8,8(r2)
       add.d f4,f2,f4
       add.d f8,f6,f8
       s.d f4, 0(r3)
       s.d f8, 8(r3)
       dadd r1,r1,#16
       dadd r2,r2,#16
       dadd r3,r3,#16
       dsub r5,r4,r1
       bnez r5,loop
       trap #0              ; Fin de programa

```

La figura siguiente muestra el diagrama incompleto instrucciones–tiempo correspondiente a una iteración intermedia del bucle:

PC	Instruccion	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
16	dadd r4,r1,#512	IF	X																	
loop	l.d f2,0(r1)	IF	I	AC	L1	L2	-	WB	-	-	-	C								
24	l.d f6,8(r1)	IF	I	AC	L1	L2	WB	-	-	-	C									
28	mul.d f2,f0,f2	COMPLETAR																		
32	mul.d f6,f0,f6	IF	I	-	-	-	M1	M2	M3	WB	C									
36	l.d f4,0(r2)	COMPLETAR																		
40	l.d f8,8(r2)	IF	I	AC	L1	L2	WB	-	-	-	C									
44	add.d f4,f2,f4	IF	I	-	-	-	-	-	-	-	A1	A2	WB	C						
48	add.d f8,f6,f8	IF	I	-	-	-	-	-	-	-	A1	A2	WB	C						
52	s.d f4, 0(r3)	COMPLETAR																		
56	s.d f8, 8(r3)	IF	I	AC	-	-	-	-	-	-	-	-	-	C	S1	S2				
60	dadd r1,r1,#16	IF	I	E1	WB	-	-	-	-	-	-	-	-	C						
64	dadd r2,r2,#16	IF	I	E1	WB	-	-	-	-	-	-	-	-	C						
68	dadd r3,r3,#16	IF	I	E1	WB	-	-	-	-	-	-	-	-	C						
72	dsub r5,r4,r1	IF	I	E1	-	WB	-	-	-	-	-	-	-	C						
76	bnez r5,loop	COMPLETAR																		
16	dadd r4,r1,#512	COMPLETAR																		
loop	l.d f2,0(r1)	IF	I	AC	L1	L2	-	WB	-	-	-	C								

- a) Rellene las filas incompletas del diagrama utilizando el diagrama instrucciones–tiempo proporcionado.
- b) Considerando la instrucción con PC 48 (add.d f8, f6, f8) indique en qué ciclo:
- 1) Dispone del contenido del operando correspondiente al registro f6.
  - 2) Dispone del contenido del operando correspondiente al registro f8.
  - 3) Se ocupa la estación de reserva.
  - 4) Se ocupa la entrada del ROB.
  - 5) Se escribe el registro f8.
  - 6) Se libera la estación de reserva.
  - 7) Se libera la entrada del ROB.
- c) Tomando el diagrama mostrado como una iteración representativa,
- 1) ¿Cuántos ciclos consume una iteración cuando el predictor acierta?
  - 2) ¿Y cuando falla?
  - 3) Si asumimos que se ejecutan 32 iteraciones a este bucle y considerando que el procesador funciona a una frecuencia de 1 GHz, ¿Cuántos MIPS ofrecerá el procesador ejecutando el código bajo estudio? Asume que el predictor falla en las predicciones de la primera y la última iteración. Para el cálculo, considera solamente las instrucciones que componen el bucle.

### Solución:

- a) Rellene las filas incompletas del diagrama utilizando la plantilla siguiente:

PC	Instruccion	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
16	dadd r4, r1, #512	IF	X																	
loop	l.d f2, 0(r1)	IF	I	AC	L1	L2	-	WB	-	-	-	C								
24	l.d f6, 8(r1)		IF	I	AC	L1	L2	WB	-	-	-	C								
28	mul.d f2, f0, f2		IF	I	-	-	-	-	M1	M2	M3	WB	C							
32	mul.d f6, f0, f6			IF	I	-	-	-	M1	M2	M3	WB	C							
36	l.d f4, 0(r2)			IF	I	AC	L1	L2	WB	-	-	-	C							
40	l.d f8, 8(r2)				IF	I	AC	L1	L2	WB	-	-	-	C						
44	add.d f4, f2, f4				IF	I	-	-	-	-	-	-	A1	A2	WB	C				
48	add.d f8, f6, f8					IF	I	-	-	-	-	-	A1	A2	WB	C				
52	s.d f4, 0(r3)					IF	I	AC	-	-	-	-	-	-	-	-	C	S1	S2	
56	s.d f8, 8(r3)						IF	I	AC	-	-	-	-	-	-	-	C	S1	S2	
60	dadd r1, r1, #16						IF	I	E1	WB	-	-	-	-	-	-	-	C		
64	dadd r2, r2, #16							IF	I	E1	WB	-	-	-	-	-	-	C		
68	dadd r3, r3, #16							IF	I	E1	WB	-	-	-	-	-	-	C		
72	dsub r5, r4, r1								IF	I	E1	-	WB	-	-	-	-	C		
76	bnez r5, loop								IF	I	-	-	-	E1	-	WB	-	-	-	C
16	dadd r4, r1, #512									IF	X									
loop	l.d f2, 0(r1)									IF	I	AC	L1	L2	-	WB	-	-	-	C

- b) Considerando la instrucción con PC 48 (add.d f8, f6, f8) indique en que ciclo:

- 1) Dispone del contenido del operando correspondiente al registro f6. Ciclo 39
- 2) Dispone del contenido del operando correspondiente al registro f8. Ciclo 37
- 3) Se ocupa la estación de reserva. Ciclo 34
- 4) Se ocupa la entrada del ROB. Ciclo 34
- 5) Se escribe el registro f8. Ciclo 43
- 6) Se libera la estación de reserva. Ciclo 42
- 7) Se libera la entrada del ROB. Ciclo 43

- c) Tomando el diagrama mostrado como una iteración representativa,

- 1) ¿cuantos ciclos consume una iteración cuando el predictor acierta? 8 ciclos
- 2) ¿y cuando falla? 19 ciclos
- 3) Si asumimos que se ejecutan 32 iteraciones a este bucle y considerando que el procesador funciona a una frecuencia de 1 GHz, ¿Cuántos MIPS ofrecerá el procesador ejecutando el código bajo estudio? Asume que el predictor falla en las predicciones de la primera y la última iteración.

$$MIPS = \frac{15 \text{instrucciones} \times 32 \text{iteraciones}}{(30 \text{iteraciones} \times 8 \text{ciclos} + 2 \text{iteraciones} \times 19 \text{ciclos}) \times 1 \text{ns} \times 10^6} = 1726,61$$

## 2. (1,5 puntos)

El fabricante del procesador del ejercicio 1 también vende un modelo menos potente con las mismas características excepto que sólo cuenta con un operador de cada tipo. Sobre este procesador se ejecuta el siguiente bucle:

```
loop:  dsub r1,r1,#8
      l.d f0, a(r1)
      add.d f4, f0, f2
      s.d f4, b(r1)
      s.d f4, c(r1)
      bnez r1,loop
      trap #0
```

El diagrama instrucciones–tiempo para dos iteraciones consecutivas cualesquiera del bucle es el siguiente:

PC	Instruc.	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
loop	dsub r1,r1,8	IF	I	E1	WB	-	-	-	-	-	-	-	C						
1	l.d f0,A(r1)	IF	I	-	-	AC	L1	L2	WB	-	-	-	C						
2	add.d f4,f0,f2		IF	I	-	-	-	-	-	A1	A2	WB	C						
3	s.d f4,B(r1)		IF	I	-	-	AC	-	-	-	-	-	C	L1	L2				
4	s.d f4,C(r1)			IF	I	-	-	AC	-	-	-	-	-	C	L1	L2			
5	bnez r1,loop			IF	I	E1	WB	-	-	-	-	-	-	C					
loop	dsub r1,r1,8				IF	I	E1	WB	-	-	-	-	-	-	C				
1	l.d f0,A(r1)				IF	I	-	-	AC	L1	L2	WB	-	-	C				
2	add.d f4,f0,f2					IF	I	-	-	-	-	-	A1	A2	WB	C			
3	s.d f4,B(r1)					IF	I	-	-	AC	-	-	-	-	-	C	L1	L2	
4	s.d f4,C(r1)						IF	I	-	-	AC	-	-	-	-	-	C	L1	L2
5	bnez r1,loop						IF	I	E1	WB	-	-	-	-	-	-	C		

Responde a las siguientes cuestiones, **justificando la respuesta**:

- ¿Cuántos ciclos toma lanzar (*Issue*) todas las instrucciones de la segunda iteración mostrada desde que se lanza la segunda `dsub` en el ciclo 8?
- ¿Cuántos ciclos toma retirar (*Commit*) todas las instrucciones de la segunda iteración mostrada desde que se retira la segunda `dsub` en el ciclo 17?
- Para implementar un modelo del mismo procesador con mayor frecuencia ha sido necesario incrementar la latencia del sumador de coma flotante a 5 ciclos. Esto afecta al diagrama instrucciones–tiempo, que queda como sigue. Observa que, en este caso, el número de ciclos que toma retirar todas las instrucciones de la segunda iteración desde que se retira la segunda `dsub` es 5.

PC	Instruc.	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
loop	dsub r1,r1,8	IF	I	E1	WB	-	-	-	-	-	-	-	-	-	-	C										
1	l.d f0,A(r1)	IF	I	-	-	AC	L1	L2	WB	-	-	-	-	-	-	C										
2	add.d f4,f0,f2		IF	I	-	-	-	-	-	-	-	A1	A2	A3	A4	A5	WB	C								
3	s.d f4,B(r1)		IF	I	-	-	AC	-	-	-	-	-	-	-	-	-	-	C	L1	L2						
4	s.d f4,C(r1)			IF	I	-	-	AC	-	-	-	-	-	-	-	-	-	-	C	L1	L2					
5	bnez r1,loop			IF	I	E1	WB	-	-	-	-	-	-	-	-	-	-	-	C							
loop	dsub r1,r1,8				IF	I	E1	WB	-	-	-	-	-	-	-	-	-	-	-	C						
1	l.d f0,A(r1)				IF	I	-	-	AC	L1	L2	WB	-	-	-	-	-	-	-	C						
2	add.d f4,f0,f2					IF	I	-	-	-	-	-	-	-	-	-	A1	A2	A3	A4	A5	WB	C			
3	s.d f4,B(r1)					IF	I	-	-	AC	-	-	-	-	-	-	-	-	-	-	-	-	C	L1	L2	
4	s.d f4,C(r1)						IF	I	-	-	AC	-	-	-	-	-	-	-	-	-	-	-	-	C	L1	L2
5	bnez r1,loop						IF	I	E1	WB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	

Si se ejecutan suficientes iteraciones como las mostradas, se observa que finalmente se producen ciclos de parada en la etapa *Issue*. Razona cuál puede ser la causa.

- Observa que la suma (etapas A1–A5) de la segunda `add.d` se inicia en el ciclo 19, lo que retrasa su retiro (*Commit*) hasta el ciclo 25 y es la causa de que retirar las instrucciones de la segunda iteración cueste 5 ciclos. Este problema se podría evitar adelantando el inicio de la segunda suma al ciclo 17, o antes, pero esto no es posible con el actual sumador, ya que existe un riesgo estructural con la primera `add.d`, que finaliza su suma en el ciclo 18.

Explica si sería posible o no solucionar este problema con un sumador segmentado.

- e) Si finalmente se utiliza un sumador no segmentado, ¿cuál debería ser su latencia máxima para evitar que se produzca el problema anterior?

**Solución:**

- a) ¿Cuántos ciclos toma lanzar (*Issue*) todas las instrucciones de la segunda iteración mostrada desde que se lanza la segunda *dsub* en el ciclo 8?

3 ciclos, el bucle contiene 6 instrucciones y cada ciclo son lanzadas dos.

- b) ¿Cuántos ciclos toma retirar (*Commit*) todas las instrucciones de la segunda iteración mostrada desde que se retira la segunda *dsub* en el ciclo 17?

3 ciclos, el bucle contiene 6 instrucciones y cada ciclo dos hacen commit.

- c) Si se ejecutan suficientes iteraciones como las mostradas, se observa que finalmente se producen ciclos de parada en la etapa *Issue*. Razona cuál puede ser la causa.

O bien el ROB o bien las estaciones de reserva del sumador/restados se van a llenar ya que las instrucciones se lanzan con más rapidez que retiran. En un caso u otro tendremos que insertar *stalls* en la etapa *issue*.

- d) Observa que la suma (etapas A1–A5) de la segunda *add.d* se inicia en el ciclo 19, lo que retrasa su retiro (*Commit*) hasta el ciclo 25 y es la causa de que retirar las instrucciones de la segunda iteración cueste 5 ciclos. Este problema se podría evitar adelantando el inicio de la segunda suma al ciclo 17, o antes, pero esto no es posible con el actual sumador, ya que existe un riesgo estructural con la primera *add.d*, que finaliza su suma en el ciclo 18.

Explica si sería posible o no solucionar este problema con un sumador segmentado.

Efectivamente con un sumador segmentado la instrucción podría ser lanzada sin esperar a que finalice la operación de suma previa y se evitaría este problema.

- e) Si finalmente se utiliza un sumador no segmentado, ¿cuál debería ser su latencia máxima para evitar que se produzca el problema anterior?

La latencia máxima sería *d3* si lo que se quiere es evitar que los commits de las instrucciones de la segunda iteración necesiten más de 3 ciclos.

□

**3. (4,5 puntos)**

El computador de bajo coste Strawberry-E One cuenta con un procesador MIPS64 funcionando a 2000 MHz. Este procesador dispone de una cache L1 que almacena instrucciones y datos con tamaño de bloque de 64 bytes. La memoria instalada es una Samsung Hynix Micron DDR2 con temporización 10-10-10 y ancho de bus de memoria de 64 bits. La frecuencia del bus de memoria es 400 MHz.

Responda a las siguientes preguntas, justificando las respuestas:

- a) Calcula el ancho de banda del bus de memoria en Mbytes/s.
- b) El computador ejecuta una aplicación para medir las prestaciones de la cache L1. Se comprueba que la penalización por fallo de L1 es de 40 nanosegundos. ¿Cuál es la probabilidad ML de que un acceso a memoria principal encuentre la fila abierta?
- c) La mencionada aplicación tarda en ejecutarse 335 segundos. Se ha medido que ejecuta aproximadamente  $10^{11}$  instrucciones, de las cuales un 40 % son de carga/almacenamiento. Por otro lado, la tasa de fallos de la cache L1 es del 5 %. ¿Cuál es el CPI de la aplicación en ausencia de fallos?

- d) Para la nueva versión de la Strawberry-E, se pretende incluir un nuevo procesador con una jerarquía de cache mejorada. Este nuevo procesador implementa una cache L2 con tiempo de acceso de 6 nanosegundos y se pretende que reduzca la penalización por fallo de L1 a aproximadamente la mitad (20 nanosegundos). ¿Cuál debería ser la tasa de fallos de la cache L2 para cumplir esta pretensión?
- e) Además de la inclusión de la cache L2, el nuevo procesador dispone de cache L1 de datos (L1D, con tasa de fallos del 4 %) e instrucciones (L1I, con tasa de fallos prácticamente nula 0 %) y un pipeline mejorado que permite que el CPI de la aplicación en ausencia de fallos sea 1. Asumiendo que se consigue que la penalización por fallo de L1 sea de 20 nanosegundos, ¿cuál sería el tiempo de ejecución de la aplicación?

**Solución:**

- a) El ancho de bus es de 64 bits (8 bytes) y puede transmitir 2 palabras de 64 bits por ciclo (DDR2), por tanto:

$$BW = 400 \times 8 \times 2 = 6400 \text{ Mbytes/s.}$$

- b) La penalización por fallo de L1 es de 40 ns. ¿Cuál es la probabilidad ML de que un acceso a memoria principal encuentre la fila abierta?

Con un sólo nivel de cache, la penalización por fallo de L1 coincide con el tiempo de acceso a memoria:

$$PF_{L1} = (L_c \times (1 - ML) + Lr_c \times ML + \frac{B}{BW_c}) \times \frac{1}{f_{bus}},$$

donde  $PF_{L1} = 40 \text{ ns}$ ,  $L_c = 10 + 10 + 10 = 30$ ,  $Lr_c = 10$ ,  $B = 64 \text{ bytes}/64 \text{ bits} = 8$ ,  $BW_c = 2$  (DDR) y  $f_{bus} = 400 \text{ MHz}$ . Así:

$$40 \text{ ns} = (30 \times (1 - ML) + 10 \times ML + \frac{8}{2}) \times \frac{1}{400 \text{ MHz}},$$

de donde:

$$ML = 0,9.$$

- c) La aplicación tarda en ejecutarse 335 segundos. Ejecuta  $10^{11}$  instrucciones, siendo un 40 % son de carga/almacenamiento. La tasa de fallos de L1 es del 5 %. ¿Cuál es el CPI de la aplicación en ausencia de fallos? Con sólo un nivel unificado de cache y sin separar lecturas de escrituras, la ecuación del tiempo de ejecución es la siguiente:

$$Teje = I \times CPI \times t_{proc} + I \times API \times TF_{L1} \times PF_{L1},$$

donde  $T = 335 \text{ segundos}$ ,  $I = 10^{11}$ ,  $t_{proc} = 1/2000 \text{ MHz}$ ,  $API = 1,40$ ,  $TF_{L1} = 0,05$  y  $PF_{L1} = 40 \text{ ns}$ . Así:

$$335 \text{ s} = 10^{11} \times CPI \times \frac{1}{2000 \text{ MHz}} + 10^{11} \times 1,40 \times 0,05 \times 40 \text{ ns},$$

de donde:

$$CPI = 1,1.$$

- d) Se mejora la jerarquía de cache. Se incluye una L2 con tiempo de acceso de 6 ns y se desea reducir a penalización por fallo de L1 a aprox. la mitad (20 ns). ¿Cuál debería ser la tasa de fallos L2?

Con una cache de segundo nivel, la penalización de por fallo de L1 sigue la ecuación:

$$PF_{L1} = TA_{L2} + TF_{L2} \times PF_{L2},$$

donde  $PF_{L1} = 20 \text{ nanosegundos}$ ,  $TA_{L2} = 6 \text{ nanosegundos}$  y  $PF_{L2} = 40 \text{ nanosegundos}$ . Así:

$$20 \text{ ns} = 6 \text{ ns} + TF_{L2} \times 40 \text{ ns},$$

de donde:

$$TF_{L2} = 0,35.$$

- e) Además de L2, el procesador dispone L1D (con TF=4 %) y L1I (con TF=0 %) y el CPI=1 en ausencia de fallos. Si PF de L1 es 20 ns, ¿cuál es el tiempo de ejecución? Con cache de instrucciones y datos separadas, la ecuación del tiempo de ejecución es la siguiente:

$$T = I \times CPI \times t_{proc} + I \times API_{L1I} \times TF_{L1} \times PF_{L1} + I \times API_{L1D} \times TF_{L1} \times PF_{L1},$$

pero al ser la tasa de fallos de instrucciones es del 0 %, la ecuación queda como:

$$T = I \times CPI \times t_{proc} + I \times API_{L1D} \times TF_{L1} \times PF_{L1},$$

donde  $I = 10^{11}$ ,  $CPI = 1$ ,  $t_{proc} = 1/2000$  MHz,  $API_{L1D} = 0,40$ ,  $TF_{L1D} = 0,04$  y  $PF_{L1} = 20$  nanosegundos. Así:

$$T = 10^{11} \times 1 \times \frac{1}{2000 \text{ MHz}} + 10^{11} \times 0,40 \times 0,04 \times 20 \text{ ns} = 82 \text{ s}$$

□

Apellidos y Nombre:	
---------------------	--

## Ejercicio 1

a) Diagrama instrucciones-tiempo.

PC	Instruccion	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
16	dadd r4,r1,#512	IF	X																	
loop	l.d f2,0(r1)	IF	I	AC	L1	L2	-	WB	-	-	-	C								
24	l.d f6,8(r1)		IF	I	AC	L1	L2	WB	-	-	-	C								
28	mul.d f2,f0,f2																			
32	mul.d f6,f0,f6			IF	I	-	-	-	M1	M2	M3	WB	C							
36	l.d f4,0(r2)																			
40	l.d f8,8(r2)				IF	I	AC	L1	L2	WB	-	-	-	C						
44	add.d f4,f2,f4				IF	I	-	-	-	-	-	-	A1	A2	WB	C				
48	add.d f8,f6,f8					IF	I	-	-	-	-	-	A1	A2	WB	C				
52	s.d f4, 0(r3)																			
56	s.d f8, 8(r3)						IF	I	AC	-	-	-	-	-	-	-	C	S1	S2	
60	dadd r1,r1,#16						IF	I	E1	WB	-	-	-	-	-	-	C			
64	dadd r2,r2,#16							IF	I	E1	WB	-	-	-	-	-	C			
68	dadd r3,r3,#16							IF	I	E1	WB	-	-	-	-	-	C			
72	dsub r5,r4,r1								IF	I	E1	-	WB	-	-	-	C			
76	bnez r5,loop																			
16	dadd r4,r1,#512																			
loop	l.d f2,0(r1)									IF	I	AC	L1	L2	-	WB	-	-	-	C