

**Arquitectura e Ingeniería de Computadores - 3<sup>er</sup> Curso. Sol. del Examen recup. (2º parcial) 1/2/2023**  
*Responde cada pregunta en una hoja distinta. Tiempo disponible: 1h30m*

1. (2.5 puntos) Considera los siguientes ciclos del diagrama instrucciones–tiempo de un bucle:

PC	Instruccion	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61
4120	mul.d f4,f3,f3	C																			
4124	mul.d f4,f0,f3	WB	C																		
4128	mul.d f5,f1,f3	M3	WB	C																	
4132	add.d f5,f5,f2	-	-	A1	A2	WB	C														
4136	add.d f4,f4,f5	-	-	-	-	-	A1	A2	WB	C											
4140	s.d f4,y(r1)	-	-	-	-	-	-	-	-	-	C	L1	L2								
4144	s.d f5,x(r1)	-	-	-	-	-	-	-	-	-	-	C	L1	L2							
4148	dadd r1,r1,8	E1	-	WB	-	-	-	-	-	-	-	-	C								
4152	bne r1,r2,loop	I	-	-	E1	-	WB	-	-	-	-	-	-	C							
loop	l.d f0,a(r1)	IF	I	-	AC	L1	L2	WB	-	-	-	-	-	-	C						
4108	l.d f1,b(r1)		IF	I	-	AC	L1	L2	-	WB	-	-	-	-	-	C					
4112	l.d f2,c(r1)			IF	I	-	AC	L1	L2	-	WB	-	-	-	-	-	C				
4116	l.d f3,d(r1)				IF	I	-	AC	L1	L2	-	WB	-	-	-	-	-	C			
4120	mul.d f4,f3,f3					IF	I	-	-	-	-	-	M1	M2	M3	WB	-	-	-	C	
4124	mul.d f4,f0,f3						IF	I	-	-	-	-	-	M1	M2	M3	WB	-	-	-	C
4128	mul.d f5,f1,f3							IF	I	-	-	-	-	-	M1	M2	M3	WB	-	-	C
4132	add.d f5,f5,f2								IF	I	-	-	-	-	-	-	-	-	A1	A2	WB
4136	add.d f4,f4,f5									IF	I	-	-	-	-	-	-	-	-	-	-
4140	s.d f4,y(r1)										IF	I	I	I	AC	-	-	-	-	-	-
4144	s.d f5,x(r1)											if	if	IF	I	AC	-	-	-	-	-
4148	dadd r1,r1,8															IF	I	E1	-	WB	-
4152	bne r1,r2,loop																IF	I	-	-	E1

Responde a las siguientes preguntas (**justificando todas tus respuestas**):

- En los ciclos 44 y 45, dos instrucciones de carga retrasan un ciclo la ejecución de sus respectivas etapas AC. Detalla para cada una de ellas por qué se produce este retraso.
- El diagrama muestra que una instrucción repite su etapa Issue durante los ciclos 52, 53 y 54. ¿Qué tipo de riesgo motiva esta situación? ¿Por qué se produce? De los tres ciclos ¿son todos ciclos de parada? En caso afirmativo justifica el por qué, y en caso negativo indica cual(es) lo son y cual(es) no.
- Una instrucción de almacenamiento hace commit en el ciclo 51, ¿al final de qué ciclo recupera dicha instrucción el valor de su operando fuente, y al final de qué ciclo actualiza con dicho valor su posición de memoria destino?
- Determina el número mínimo de estaciones de reserva del multiplicador y del sumador, así como el número mínimo de buffers de lectura y escritura del operador de carga y almacenamiento, que se necesitarían para ejecutar el fragmento de código mostrado sin necesidad de insertar ciclos de parada, asumiendo un ROB con un número suficientemente grande de entradas.

**Solución:**

- (0.5 puntos) En los ciclos 44 y 45, dos instrucciones de carga retrasan un ciclo la ejecución de sus respectivas etapas AC. Detalla para cada una de ellas por qué se produce este retraso.  
 La primera instrucción de carga (*l.d f0,a(r1)*) espera a tener disponible R1 que se escribe en el ciclo 44. Por ello no puede ejecutar su etapa AC hasta el ciclo el 45. (0.25 puntos)  
 La segunda instrucción de carga (*l.d f1,b(r1)*) debe esperarse a que la anterior ejecute su etapa AC porque sólo existe un operador de carga. Por ello no ejecuta AC hasta el ciclo 46. (0.25 puntos)

- b) (0.75 puntos) El diagrama muestra que una instrucción repite su etapa Issue durante los ciclos 52, 53 y 54. ¿Qué tipo de riesgo motiva esta situación? ¿Por qué se produce? ¿Son los tres ciclos de parada?.

Existe un riesgo estructura (0.25 puntos) motivado por una falta de buffers de escritura en el operador de almacenamiento (0.25 puntos). De los 3 ciclos, los ciclos 52 y 53 son de parada, mientras que el 54 es el de ejecución de la etapa Issue de la instrucción. (0.25 puntos)

- c) (0.5 puntos) ¿Al final de qué ciclo la instrucción de almacenamiento que hace commit en el ciclo 51 recupera el valor de su operando fuente, y al final de qué ciclo actualiza con dicho valor la posición de memoria destino?

- Recuperación del valor de f4 en ciclo: 49 (WB de la add.d anterior). (0.25 puntos)
- Actualización memoria al final del ciclo: 53 (cuando finaliza L2). (0.25 puntos)

- d) (0.75 puntos) Determina el número mínimo de estaciones de reserva del multiplicador y del sumador, así como el número mínimo de buffers de lectura y escritura del operador de carga y almacenamiento, que se necesitarían para ejecutar el fragmento de código mostrado sin necesidad de insertar ciclos de parada, asumiendo un ROB con un número suficientemente grande de entradas.

- ER del multiplicador: 3 (0.15 puntos)
- ER del sumador: 2 (0.15 puntos)
- Carga/almacenamiento, BL: 4 (0.15 puntos), BE:4 (0.3 puntos)

□

2. (2.5 puntos) Se desea optimizar la ejecución de una aplicación en un computador superescalar con ejecución fuera de orden de 64 bits funcionando a una frecuencia de reloj de 2.5GHz. Esta aplicación ejecuta un total de 24 millones de instrucciones, obteniendo un CPI medio en ausencia de fallos de cache de 0.8 ciclos. Al analizar la aplicación se ha observado que se ejecutan un 24 % de instrucciones de acceso a memoria, siendo un 15 % lecturas y un 9 % escrituras. Para las escrituras, el procesador dispone de buffers de escritura.

La jerarquía de memoria está compuesta por dos niveles de cache, siendo el tamaño de bloque para ambos niveles de 64 bytes:

**Primer nivel:** Caches separadas para instrucciones y datos. Tiempo de acierto,  $TA_{L1} = 1$  ciclo. Tasa de fallos de instrucciones,  $TF_{L1i} = 1,3\%$ . Tasa de fallos de datos, para lecturas y escrituras,  $TF_{L1lec} = 3,2\%$  y  $TF_{L1esc} = 5,8\%$  respectivamente.

**Segundo nivel:** Tiempo de acierto,  $TA_{L2} = 8$  ciclos. Tasa de fallos,  $TF_{L2} = 38\%$ .

**Memoria principal:** Tiempo de acceso medio a memoria principal,  $TA_{MP} = 70$  ciclos de cpu.

La ejecución fuera de orden permite que un 40 % de la penalización de fallos en la cache L1 de datos quede solapada con la ejecución de otras instrucciones, siendo el resto de solapamientos despreciable.

Se pide:

- a) La penalización de fallos para la cache L1.
- b) Tiempo de ejecución de la aplicación bajo estudio.
- c) Con el fin de mejorar el tiempo de ejecución se pretende cambiar la memoria principal por una DDR3 2000MHz, con un tiempo promedio de lectura de bloque de 80 ciclos de bus de memoria. ¿Es conveniente aplicar esta modificación? **Justifique cuantitativamente la respuesta.**

**Solución:**

- a) La penalización de fallos para la cache L1.

$$PF_{L1} = TA_{L2} + TF_{L2} \times PF_{L2} = 8 + 0,38 \times 70 = 34,6 \text{ ciclos de cpu.}$$

b) Tiempo de ejecución de la aplicación bajo estudio.

$$T_{ejec} = I \times CPI \times T + I \times (API_{inst} \times TF_{L1i} + API_{lec} \times TF_{L1lec} \times FNS) \times PF_{L1} \times T =$$

$$= 24 \times 10^6 \times 0,8 \times \frac{1}{2,5 \times 10^9} + 24 \times 10^6 \times (1 \times 0,013 + 0,15 \times 0,032 \times 0,6) \times 34,6 \times \frac{1}{2,5 \times 10^9} =$$

$$= 7,68ms + 5,27ms = 12,95ms$$

c) Con el fin de mejorar el tiempo de ejecución se pretende cambiar la memoria principal por una DDR3 2000MHz, con un tiempo promedio de lectura de bloque de 80 **ciclos de bus de memoria**. ¿Es conveniente aplicar esta modificación? **Justifique cuantitativamente la respuesta.**

$$PF_{L1.1} = TA_{L2} + TF_{L2} \times PF_{L2} = 8 + 0,38 \times 80 \times \frac{2500MHz}{2000MHz} = 46 \text{ ciclos de cpu.}$$

$$T_{ejec.1} = I \times CPI \times T + I \times (API_{inst} \times TF_{L1i} + API_{lec} \times TF_{L1lec} \times FNS) \times PF_{L1} \times T =$$

$$= 24 \times 10^6 \times 0,8 \times \frac{1}{2,5 \times 10^9} + 24 \times 10^6 \times (1 \times 0,013 + 0,15 \times 0,032 \times 0,6) \times 46 \times \frac{1}{2,5 \times 10^9} =$$

$$= 7,68ms + 7,012ms = 14,69ms$$

$$S_1 = \frac{T_{ejec.orig}}{T_{ejec.1}} = \frac{12,95}{14,69} = 0,88$$

La mejora propuesta pierde prestaciones obteniendo una aceleración por debajo de 1 ( $S = 0,88$ ). Concretamente se pierde un 13,43 % de prestaciones ( $SlowDown = \frac{T_{ejec.1}}{T_{ejec.orig}} = \frac{14,69}{12,95} = 1,1343$ ).

□

3. (2.5 puntos) El siguiente diagrama i-t muestra la primera iteración de un bucle que se ejecuta sobre un procesador superescalar basado en la arquitectura MIPS64.

PC		01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17
0x00	loop: L.D F1,0(R10)	IF	I	AC	L1	L2	WB	C										
0x04	L.D F3,8(R10)	IF	I		AC	L1	L2	WB	C									
0x08	ADD.D F2,F1,F0	IF	I					A1	A2	A3	WB	C						
0x0C	ADD.D F4,F1,F10	IF	I					A1	A2	A3	WB	C						
0x10	ADD.D F6,F3,F0		IF	I					A1	A2	A3	WB	C					
0x14	ADD.D F8,F3,F10		IF	I					A1	A2	A3	WB	C					
0x18	S.D F2,0(R11)		IF	I		AC							C	L1	L2			
0x1C	S.D F4,8(R11)		IF	I			AC						C		L1	L2		
0x20	S.D F6,16(R11)			IF	I			AC					C			L1	L2	
0x24	S.D F8,24(R11)			IF	I				AC				C				L1	L2
0x28	DADDI R10,R10,16			IF	I	EX	WB						C					
0x2C	DADDI R11,R11,32			IF	I		EX	WB					C					
0x30	DADDI R1,R1,-1				IF	I		EX	WB					C				
0x34	BNEZ R1,loop				IF	I				EX	WB				C			
0x38	loop_end: <loop_end+0>					IF	X											
0x3C	<loop_end+4>					IF	X											
0x00	loop: L.D F1,0(R10)						IF	I	...									

Responda a las siguientes cuestiones, **justificando cada una de las respuestas**:

- ¿Cuántas vías implementa el procesador?
- ¿Cuántos operadores de carga/almacenamiento incluye el procesador?
- ¿Cuántos operadores de suma/resta en coma flotante incluye el procesador como mínimo? ¿estarían segmentados?
- ¿Cuál es la predicción del predictor de saltos para el salto con PC 0x34?
- ¿Por qué se cancelan las instrucciones con PC 0x38 y 0x3C?
- Asumiendo que todas las iteraciones duran lo mismo que la iteración mostrada, ¿cuál es el tiempo de ejecución del bucle en ciclos si el número de iteraciones es  $n$ ?
- Completa el diagrama en la hoja adjunta para las 4 primeras instrucciones de la segunda iteración.

**Solución:**

- a) ¿Cuántas vías implementa el procesador?  
4. Se observa que el procesador puede realizar no más de 4 etapas IF, I y C en el mismo ciclo.
- b) ¿Cuántos operadores de carga/almacenamiento incluye el procesador?  
Sólo 1, puesto que no es capaz de iniciar más de una carga/almacenamiento al mismo tiempo.
- c) ¿Cuántos operadores de suma/resta en coma flotante incluye el procesador como mínimo? ¿estarían segmentados?  
Al menos 2, puesto que puede iniciar dos operaciones de suma en coma flotante al mismo tiempo. Estarían segmentados, porque las instrucciones de suma con PC 0x10 y 0x14 se inician un ciclo después de las sumas con los PC 0x08 y 0x0C.
- d) ¿Cuál es la predicción del predictor de saltos para el salto con PC 0x34?  
Predice que el salto se tomará, puesto que se busca la instrucción en el destino del salto (loop) en el ciclo 05.
- e) ¿Por qué se cancelan las instrucciones con PC 0x38 y 0x3C?  
Se cancelan porque son las instrucciones siguientes en orden de programa al salto y la predicción es que el salto será efectivo.
- f) Asumiendo que todas las iteraciones duran lo mismo que la iteración mostrada, ¿cuál es el tiempo de ejecución del bucle en ciclos si el número de iteraciones es  $n$ ?  
 $4n$ , puesto que la primera instrucción de la segunda iteración se busca en el ciclo 05. Es decir, cada iteración dura 4 ciclos.
- g) Completa el diagrama en la hoja adjunta para las 4 primeras instrucciones de la segunda iteración

PC		01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17
0x00	loop: L.D F1,0(R10)	IF	I	AC	L1	L2	WB	C										
0x04	L.D F3,8(R10)	IF	I		AC	L1	L2	WB	C									
0x08	ADD.D F2,F1,F0	IF	I					A1	A2	A3	WB	C						
0x0C	ADD.D F4,F1,F10	IF	I					A1	A2	A3	WB	C						
0x10	ADD.D F6,F3,F0		IF	I					A1	A2	A3	WB	C					
0x14	ADD.D F8,F3,F10		IF	I					A1	A2	A3	WB	C					
0x18	S.D F2,0(R11)		IF	I		AC							C	L1	L2			
0x1C	S.D F4,8(R11)		IF	I			AC						C		L1	L2		
0x20	S.D F6,16(R11)			IF	I			AC						C		L1	L2	
0x24	S.D F8,24(R11)			IF	I				AC					C			L1	L2
0x28	DADDI R10,R10,16			IF	I	EX	WB							C				
0x2C	DADDI R11,R11,32			IF	I		EX	WB						C				
0x30	DADDI R1,R1,-1				IF	I		EX	WB						C			
0x34	BNEZ R1,loop				IF	I				EX	WB				C			
0x38	loop_end: <loop_end+0>				IF	X												
0x3C	<loop_end+4>				IF	X												
0x00	loop: L.D F1,0(R10)					IF	I			AC	L1	L2	WB		C			
0x04	L.D F3,8(R10)					IF	I				AC	L1	L2	WB	C			
0x08	ADD.D F2,F1,F0					IF	I							A1	A2	A3	WB	C
0x0C	ADD.D F4,F1,F10					IF	I							A1	A2	A3	WB	C

4. (2 puntos) Un computador está dotado con un sistema de memoria principal compuesto por 8 DIMMs DDR4-5000 idénticos, con una temporización 16-16-16. El procesador tiene un controlador de memoria con 4 canales, habiéndose instalado 2 DIMMs por canal. El ancho del bus de memoria es de 64 bits. El último nivel de cache (LLC) utiliza un tamaño de bloque de 64B.

Los campos de la dirección física son los siguientes:

fila (15 bits)	canal (2 bits)	rango (2 bit)	banco (4 bits)	columna (9 bits)	bytes en bus (3 bits)
----------------	----------------	---------------	----------------	------------------	-----------------------

En dicho computador se pretende ejecutar la aplicación PhotoMagic (PM) para procesar una imagen que ocupa 18.75 MB (1 MB =  $2^{20}$  bytes). La aplicación accede secuencialmente a todos los píxeles de la imagen que se encuentra almacenada por filas. La configuración de la MMU indica que dichos accesos se harán a posiciones de memoria física consecutivas, estando la imagen alineada con el tamaño de una fila de memoria principal.

Se solicita, **justificando la respuesta** con el máximo detalle:

- Tamaño máximo de la memoria instalada en GB (1 GB =  $2^{30}$  bytes).
- Número máximo de filas de memoria abiertas en la memoria principal.
- Tamaño en bytes de un buffer de fila y número de filas que necesitará abrir la aplicación PM considerando solo la lectura de la imagen.
- Tasa de aciertos en el buffer de fila,  $TAbf$ , de la aplicación PM considerando solo la lectura de la imagen.
- Suponiendo que la  $TAbf$  fuera 0.97, determinar la penalización media por fallo de la LLC en ciclos de bus y en nanosegundos,  $PF_{LLC}$ , ignorando los posibles retrasos debidos a la gestión de las peticiones en el controlador de memoria.

**Solución:**

- Tamaño máximo de la memoria instalada en GiB ( $2^{30}$  bytes).

Utilizando los bits de cada campo de la dirección física:

$$Tam_{MP} = 2^{(15+2+2+4+9+3)} = 2^{35} = 32 \text{ GiB}$$

- Número máximo de filas abiertas memoria principal.

El número máximo de filas abiertas es la multiplicación del número de bancos por rango, el número de rangos por canal por el número de canales.

$$N_{filas} = 2^{(2+2+4)} = 2^8 = 256$$

- Tamaño en bytes de un buffer de fila y número de filas que necesitará abrir la aplicación PM considerando solo la lectura de la imagen.

El tamaño de un BF se puede obtener con los bits dedicados a las columnas de una fila multiplicado por tamaño de palabra de bus.

$$Tam_{BF} = 2^{(9+3)} = 2^{12} = 4096 = 4 \text{ KiB}$$

El número de filas que abrirá/accederá la aplicación será:

$$N_{filas \text{ accedidas}} = \frac{\text{Tam imagen}}{\text{Tam buffer fila}} = \frac{18,75 \times 2^{20}}{4 \times 2^{10}} = 4800 \text{ filas}$$

- Tasa de aciertos en el buffer de fila,  $TAbf$ , de la aplicación PM considerando sólo la lectura de la imagen.

Se puede resolver de dos formas igualmente válidas:

- 1) Dado que la aplicación accede a un número exacto de filas completas, la  $TAbf$  se puede calcular utilizando solo los accesos a una fila.

Primero determinaremos cuantos bloques caben en una fila.

$$N_{\text{bloques x fila}} = \frac{\text{Tam buffer fila}}{\text{Tam bloque}} = \frac{4096}{64} = 64 \text{ bloques}$$

Si el acceso a las direcciones de un buffer de fila es secuencial, eso quiere decir que el primer acceso encontrará la fila cerrada y los otros 63 la encontrarán abierta.

$$TAbf = \frac{63}{64} = 0,984375$$

- 2) Si utilizáramos el número de accesos totales sería:

$$N_{\text{bloques accedidos}} = \frac{\text{Tam imagen}}{\text{Tam bloque}} = \frac{18,75 \times 2^{20}}{64} = 307200$$

y, dado que fallará un acceso por cada fila abierta, la  $TAbf$  sería:

$$\begin{aligned} TAbf &= \frac{\text{núm total de aciertos}}{\text{núm total de accesos}} = \frac{N_{\text{bloques accedidos}} - N_{\text{filas accedidas}}}{N_{\text{bloques accedidos}}} \\ &= \frac{307200 - 4800}{307200} = \frac{302400}{307200} = 0,984375 \end{aligned}$$

- e) Suponiendo que la  $TAbf$  fuera 0.97, determinar la penalización media por fallo de la LLC en ciclos de bus y en nanosegundos,  $PF_{LLC}$ , ignorando los posibles retrasos debidos a la gestión de las peticiones en el controlador de memoria.

La penalización por fallo de la LLC es el tiempo de acceso a memoria principal.

$$T_{\text{acceso}_{MP}} = L \times (1 - TAbf) + Lr \times TAbf + \frac{B}{Bw}$$

Donde la latencia completa es  $L = 16+16+16 = 48$  y la latencia reducida  $Lr = 16$  ciclos de bus. El tamaño de bloque  $B = 64$  bytes y el ancho de banda  $Bw = 16$  bytes/ciclo, dado que al ser una memoria de DDR es de 8 bytes por transferencia (64 bits), por 2 transferencias por ciclo. Dado que la memoria es del tipo DDR4-5000, la frecuencia del bus será 2500 MHz.

$$\begin{aligned} T_{\text{acceso}_{MP}} &= 48 \times (1 - 0,97) + 16 \times 0,97 + \frac{64}{16} \\ &= 1,44 + 15,52 + 4 = 20,96 \text{ ciclos @ 2500 MHz} = 8,384 \text{ ns} \end{aligned}$$

□