

Responde cada pregunta en una hoja distinta. Tiempo disponible: 2 horas 30 minutos

1. (3 puntos) En un procesador con ejecución fuera de orden y ROB con 120 entradas se ha obtenido el siguiente diagrama de instrucciones/ciclo.

PC	Instrucción	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
16	(i1) l.s f0,0(r1)	IF	I	AC	L1	L2	WB	C										
20	(i2) l.s f1,0(r2)		IF	I	AC		L1	L2	WB	C								
24	(i3) add.s f0,f1,f2			IF	I					A1	A2	WB	C					
28	(i4) add.s f1,f2,f3				IF	I	A1	A2		WB				C				
32	(i5) add.s f2,f4,f3					IF	I	A1	A2		WB					C		
36	(i6) mul.d f2,f2,f1						IF	I				M1	M2	WB		C		
40	(i7) add.s f3,f1,f0							IF	I	I			A1	A2	WB		C	
44	(i8) add.s f4,f5,f6								IF	IF	I	A1	A2			WB		C

Los registros enteros r1 y r2 tienen almacenados los valores 100 y 200, respectivamente; las posiciones de memoria 100 y 200 contienen los valores 7.0 y 8.0, respectivamente; y el contenido del banco de registros de coma flotante es el siguiente:

Registro	valor	ROB
f0	0.0	
f1	1.0	
f2	2.0	
f3	3.0	
f4	4.0	
f5	5.0	
f6	6.0	

Se pide, **únicamente para el ciclo 12 (final de ciclo)**:

- Indicar cuántas entradas ocupadas tendría el ROB y su contenido. Asume que la primera instrucción del diagrama tenía asignada la entrada uno.
- Contenido del banco de registros de coma flotante.
- Indica si, a partir del diagrama, se puede considerar que el número de estaciones de reserva en el sumador/restador se puede considerar ilimitado o limitado para el código mostrado. Justifica la respuesta. Indica también cuántas estarían ocupadas al final del ciclo 12 y el contenido de las mismas. Para la realización de este apartado asume que la primera que se ocupa es a1, la segunda a2, y así sucesivamente.

**Solución:**

Información al final de ciclo 12. SOLUCIÓN

- El número de entradas ocupadas en el ROB es ....**5**.....que viene delimitado por las instrucciones que ya han realizado la etapa ...**Issue**..... y tienen pendiente realizar la etapa .....**COMMIT**..... Contenido del ROB asumiendo que la primera instrucción del diagrama tenía asignada la entrada 1 y el ROB tiene 40 entradas.

Nº entrada	Busy	Instrucción	Estado	Destino	Valor	Predicción	PC
4	1	i4	WB	f1	5.0		28
5	1	i5	WB	f2	7.0		32
6	1	i6		f2			36
7	1	i7		f3			40
8	1	i8		f4			44

- Contenido del banco de registros de coma flotante.

Registro	valor	ROB
f0	10.0	
f1	8.0	4
f2	2.0	6
f3	3.0	7
f4	4.0	8
f5	5.0	
f6	6.0	

- c) El número de estaciones de reserva en sumador/restador es tres. Por este motivo la instrucción i7 debe esperar a realizar la etapa issue a que la instrucción i4 libere una de las estaciones de reserva.

Nº entrada	Busy	Operación	Qj	Vj	Qk	Vk	ROB	result
a2	1	+ (i7)		5.0		10.0	7	
a3	1	+ (i8)		5.0		6.0	8	11.0

□

## 2. (1 punto)

Responde a las siguientes preguntas teóricas. Cuando una instrucción de salto llega a la etapa commit y se detecta un fallo de predicción de salto, indica qué acciones deben realizarse en cada una de las siguientes estructuras:

- ROB.
- Banco de registros de coma flotante.
- Estaciones de reserva del sumador/restador.
- Buffer de stores (tampones de escritura).

**Solución:**

**PREGUNTAS TEÓRICAS.** Contesta las acciones realizadas en cada estructura con una **frase concisa y precisa**.

- ROB.  
Deben liberarse (busy=0) todas las entradas del ROB.
- Banco de registros de coma flotante.  
Deben eliminarse las marcas, los valores permanecen ya que son no especulativos.
- Estaciones de reserva del sumador/restador.  
Deben liberarse todas.
- Buffer de stores.  
Deben liberarse todas las entradas, excepto stores confirmadas.

□

## 3. (4 puntos)

El procesador ARM K27v es un procesador con tamaño de palabra de 64 bits y una arquitectura *load/store* similar al MIPS. Su versión más reciente funciona con una frecuencia de reloj de 3.6GHz.

Para una carga típica se obtiene un CPI en ausencia de fallos de 1.3 ciclos. Dicha carga tiene un 25 % de instrucciones de tipo *load* y un 10 % de instrucciones de tipo *store*.

El procesador dispone de dos niveles de memoria *cache*, L1 y L2, con tasas de fallos locales para la carga estudiada de 11 % y 36 % respectivamente. El tiempo de acierto es de 1 ciclo para la cache L1 y 8 ciclos para la cache L2. Ambos niveles trabajan con un tamaño de bloque de 128 bytes.

Por otra parte, la memoria principal empleada es una SDRAM DDR4 2400MHz, con un tiempo promedio de lectura de bloque de 38,6 ciclos de bus de memoria.

Calcula:

- La penalización de fallo para la *cache* L1.
- El tiempo de ejecución de un programa formado por 15 millones de instrucciones.
- Existe la posibilidad de realizar una nueva implementación del procesador empleando una memoria *cache* L1 de doble tamaño del actual que reduciría la tasa de fallo a sólo 7 %. No obstante esta memoria sería más lenta que la original lo que ha condicionado al equipo de diseño a reducir la frecuencia del procesador a 2.8Ghz. Por su parte la cache L2 aumentaría su tasa de fallos a 57 % y su tiempo de acierto pasaría de 8 ciclos a 6 ciclos. Calcular si esta nueva configuración ofrece una mejora y en tal caso cuantificarla.
- En el apartado anterior la inclusión de una cache L1 de mayor tamaño ha tenido una serie de consecuencias. Intenta dar una posible explicación a cada una de las siguientes modificaciones:
  - Se ha tenido que reducir la frecuencia de reloj del procesador.
  - Se han reducido los ciclos del tiempo de acierto de la cache L2.

**Solución:**

- La penalización de fallo para la *cache* L1.

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

Sabiendo que  $PF_{L2} = T_{MP}$  (tiempo de lectura de bloque en memoria principal), que nos dan en ciclos del bus de memoria y que hemos de pasar a *ns*,

$$PF_{L2} = T_{MP} = 38,6 \cdot \frac{1}{2,4 \cdot 10^9} = 16,08 \cdot 10^{-9} s \equiv 16,08 \text{ ns}$$

la tasa de fallo de L2 nos la dan como dato,

$$TF_{L2} = 0,36$$

tiempo de acierto de la cache de segundo nivel (nos lo dan en ciclos del procesador) que pasamos a *ns*,

$$TA_{L2} = 8 \times \frac{1}{3,6 \cdot 10^9} = 2,22 \text{ ns}$$

finalmente,

$$PF_{L1} = T_{L2} = 2,22 \text{ ns} + 0,36 \times 16,08 \text{ ns} = 8 \text{ ns}$$

- El tiempo de ejecución de un programa formado por 15 millones de instrucciones.  
El tiempo de ejecución se calcularía como:

$$T_{ex} = I \times CPI \times T_{CPU} + I \times API \times TFL \times PFL$$

Siendo  $API = 1 + 0,25 + 0,10 = 1,35$  el número medio de accesos a memoria por instrucción,  $PFL = PF_{L1}$  y el resto de datos nos los dan o ya están calculados.

$$T_{ex} = 15 \cdot 10^6 \times 1,3 \times \frac{1}{3,6 \times 10^9} + 15 \cdot 10^6 \times 1,35 \times 0,11 \times 8 \cdot 10^{-9} = 0,0232 \text{ s}$$

- Existe la posibilidad de realizar una nueva implementación del procesador empleando una memoria *cache* L1 de doble tamaño del actual que reduciría la tasa de fallo a sólo 7 %. No obstante esta memoria sería más lenta que la original lo que ha condicionado al equipo de diseño a reducir la frecuencia del procesador a 2.8Ghz. Por su parte la cache L2 aumentaría su tasa de fallos a 57 % y su tiempo de acierto pasaría de 8 ciclos a 6 ciclos. Calcular si esta nueva configuración ofrece una mejora y en tal caso cuantificarla.

En este caso habrá que recalcular  $PF_{L1}$  ya que depende del  $TA_{L2}$  y del periodo del reloj del procesador que se han modificado,

$$PF_{L1} = T_{L2} = 6 \times \frac{1}{2,8 \text{ GHz}} + 0,57 \times 16,08 \text{ ns} = 11,31 \text{ ns}$$

Sustituyendo en la ecuación del tiempo de ejecución, junto con  $TF_{L1}$  y  $T_{CPU}$  que han sido también modificados,

$$T_{ex} = 15 \cdot 10^6 \times 1,3 \times \frac{1}{2,8 \times 10^9} + 15 \cdot 10^6 \times 1,35 \times 0,07 \times 11,31 \cdot 10^{-9} = 0,0229 \text{ s}$$

Comparando ambas ejecuciones,

$$A = \frac{0,0232}{0,0229} = 1,0131 \rightarrow 1,31 \%$$

comprobamos que la nueva configuración ofrecerá una mejora del 1.31 %.

d) En el apartado anterior la inclusión de una cache L1 de mayor tamaño ha tenido una serie de consecuencias. Intenta dar una posible explicación a cada una de las siguientes modificaciones:

1) Se ha tenido que reducir la frecuencia de reloj del procesador.

En caso de acierto la memoria L1 debe poder suministrar el dato en un ciclo del procesador, si introducimos una memoria L1 más lenta esto podría ya no ser posible. Por ello tenemos que aumentar la duración de un ciclo del procesador, y por lo tanto reducir su frecuencia.

2) Se han reducido los ciclos del tiempo de acierto de la L2.

La L2 mantiene sus características, es igual de rápida, si se ha reducido el número de ciclos es porque éstos tienen ahora mayor duración.

$$8 \times \frac{1}{3,6 \cdot 10^{-9}} \sim 6 \times \frac{1}{2,8 \cdot 10^{-9}}$$

□

4. (2 puntos) Un sistema tiene instalada una memoria Samsung Hynix Micron DDR2 PC2-6400 con temporización  $CL-t_{RCD}-t_{RP}$  10-10-10 y ancho de bus de 64 bits. En cada acceso, la memoria transfiere un bloque de memoria de 64 bytes. De media, la probabilidad de que un acceso a memoria encuentre abierta la fila donde se encuentra el bloque es del 40 %.

Responde, justificando las respuestas:

a) ¿Cuál es el tiempo de lectura de un bloque en memoria principal? Calcúlalo en ciclos de bus de memoria.

b) Se han considerado tres opciones para mejorar el tiempo de lectura:

- 1) Reducir a la mitad el tiempo entre el envío de la dirección de columna por el bus de memoria y el comienzo de la recepción del bloque.
- 2) Reducir a la mitad la latencia del comienzo de la recepción de los datos cuando el acceso encuentra abierta otra fila diferente de la que contiene el bloque.
- 3) Duplicar el ancho de bus de memoria.

Para cada una de las tres opciones, calcula el tiempo de lectura en ciclos de bus.

c) Calcula el tiempo de lectura en ciclos de bus para la memoria original aplicando la técnica *Early Restart*. Asume que cada palabra de 64 bits del bloque de cache tiene la misma probabilidad de ser la requerida por el procesador.

**Solución:**

a) ¿Cuál es el tiempo de lectura de un bloque en memoria principal? Calcúlalo en ciclos de bus de memoria.

El tiempo de acceso a memoria en ciclos sigue la siguiente ecuación:

$$TA_{MM} = L_c \times (1 - ML) + Lr_c \times ML + \frac{B}{BW_c},$$

donde  $L_c = 10 + 10 + 10 = 30$ ,  $Lr_c = 10$ ,  $ML = 0,40$ ,  $B = 64 \text{ bytes}/64 \text{ bits} = 8$  y  $BW_c = 2$  (DDR), por tanto:

$$TA_{MM} = 30 \times (1 - 0,40) + 10 \times 0,40 + \frac{8}{2} = 26 \text{ ciclos.}$$

b) Para cada una de las tres opciones, calcula el tiempo de lectura en ciclos de bus.

- 1) Reducir a la mitad el tiempo entre el envío de la dirección de columna por el bus de memoria y el comienzo de la recepción del bloque.

Esta opción implica que  $Lr_c = CL = 10/2 = 5$  y  $L_c = 10/2 + 10 + 10 = 25$ , por lo tanto:

$$TA_{MM} = 25 \times (1 - 0,40) + 5 \times 0,40 + \frac{8}{2} = 21 \text{ ciclos.}$$

- 2) Reducir a la mitad la latencia del comienzo de la recepción de los datos cuando el acceso encuentra abierta otra fila diferente de la que contiene el bloque.

Esta opción implica que  $L_c = 30/2 = 15$ , por lo tanto:

$$TA_{MM} = 15 \times (1 - 0,40) + 10 \times 0,40 + \frac{8}{2} = 17 \text{ ciclos.}$$

- 3) Duplicar el ancho de bus de memoria.

Esta opción implica que  $B = 64 \text{ bytes}/128 \text{ bits} = 4$ , por lo tanto:

$$TA_{MM} = 30 \times (1 - 0,40) + 10 \times 0,40 + \frac{4}{2} = 24 \text{ ciclos.}$$

- c) Calcula el tiempo de lectura en ciclos de bus para la memoria original aplicando la técnica *Early Restart*. Asume que cada palabra de 64 bits del bloque de cache tiene la misma probabilidad de ser la requerida por el procesador.

Mediante la técnica *Early Restart*, se permite al procesador continuar inmediatamente después de que la palabra requerida se recibe por el bus de memoria.

Como la memoria es DDR, la primera palabra se recibe 1 semiciclo después de que comience la recepción del bloque, la segunda tras 2 semiciclos y así sucesivamente.

La probabilidad de que una palabra de un bloque sea la requerida por el procesador se reparte equitativamente entre las 8 palabras del bloque.

Por lo tanto, el tiempo medio de acceso a memoria queda como sigue:

$$TA_{MM} = 30 \times (1 - 0,40) + 10 \times 0,40 + \frac{1/8 + 2/8 + 3/8 + 4/8 + 5/8 + 6/8 + 7/8 + 8/8}{2} = 24,25 \text{ ciclos.}$$

□