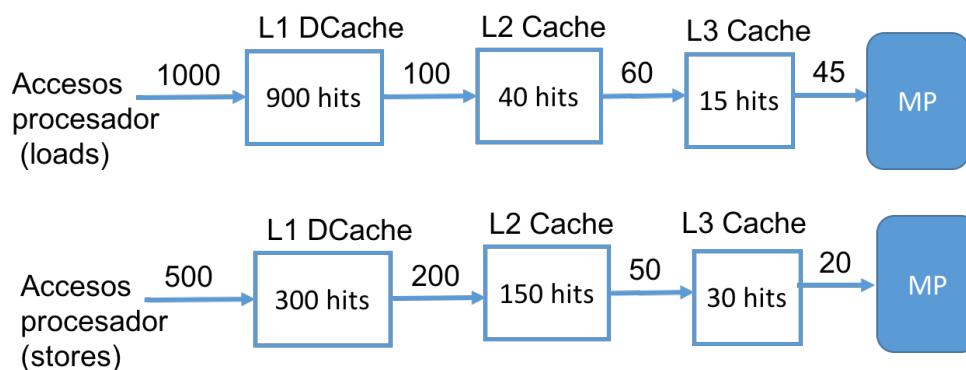


1. (2.8 puntos) Un procesador ejecuta una aplicación con las siguientes características. El 30 % de sus instrucciones son de acceso a memoria, dos tercios son *loads* y un tercio *stores*.

El procesador dispone de 3 niveles de cache. Los tiempos de acierto en ciclos del procesador son 1 ciclo para las caches de L1 (tanto de instrucciones como de datos), 6 ciclos para la L2 y 18 ciclos para la L3.

La siguiente figura muestra el número de loads y stores emitidas por el procesador que llegan a los distintos niveles de cache. La parte superior muestra el flujo de loads y la parte inferior las stores. El número dentro de la caja indica los aciertos en dicha cache y el número a la salida el número de fallos. Por ejemplo, de cada 1000 instrucciones load emitidas, 900 aciertan en la cache de L1 y 100 fallan en la L1 y van a la L2, donde aciertan 40 y fallan 60, y así sucesivamente, hasta que 45 de los accesos llegan a memoria principal (MP).

La cache de instrucciones de nivel 1 (no mostrada) tiene una tasa de aciertos del 100 % gracias a la prebúsqueda y su tiempo de acierto es 1 ciclo. El tiempo de acceso de la memoria principal es de 100 ciclos de procesador.



Se pide obtener:

- Tasas de fallo global y local para la cache L2 y la L3. SOLO para loads.
- Fórmula del tiempo de acceso para 3 niveles de cache sin distinguir el tipo de acceso. SOLO se pide la fórmula.
- Tiempo medio de acceso de las instrucciones loads y tiempo medio de acceso de las instrucciones store.
- Tiempo medio de acceso a memoria considerando los distintos tipos de acceso (loads, stores e instrucciones). DEBE deducirse una fórmula y calcularse a partir de los obtenidos en el apartado anterior.
- Razona si la política de escritura empleada en la cache L1 es write-through o write-back sabiendo que la cache L2 es inclusiva.

### **Solución:**

a) Tasas de fallo local y globales:

$$TF_{local,L2} = \frac{\#fallos\ L2}{\#accesos\ L2} = \frac{60}{100} = 0,6 \rightarrow \text{igual al 60 \%}.$$

$$TF_{local,L3} = \frac{\#fallos\ L3}{\#accesos\ L3} = \frac{45}{60} = 0,75 \rightarrow \text{igual al 75 \%}.$$

$$TF_{global,L2} = \frac{\#fallos\ L2}{\#accesos\ procesador} = \frac{60}{1000} = 0,06 \rightarrow \text{igual al 6 \%}.$$

$$TF_{global,L3} = \frac{\#fallos\ L3}{\#accesos\ procesador} = \frac{45}{1000} = 0,045 \rightarrow \text{igual al 4'5 \%}.$$

$$b) \text{Tiempo} - \text{acceso} = TA_{L1} + TF_{L1} \times (TA_{L2} + TF_{L2} \times (TA_{L3} + TF_{L2} \times PF_{L3}))$$

$$c) T - \text{acceso}_{ld} = TA_{L1} + TF_{L1,ld} \times (TA_{L2} + TF_{L2,ld} \times (TA_{L3} + TF_{L3,ld} \times PF_{L3})) \\ = 1 + 0,1 \times (6 + 0,6 \times (18 + 0,75 \times 100)) = 7,18 \text{ ciclos}$$

$$T - \text{acceso}_{st} = TA_{L1} + TF_{L1,st} \times (TA_{L2} + TF_{L2,st} \times (TA_{L3} + TF_{L3,st} \times PF_{L3})) \\ = 1 + 0,4 \times (6 + 0,25 \times (18 + 0,4 \times 100)) = 9,2 \text{ ciclos}$$

d) Para calcular el tiempo medio por acceso habría que considerar los 3 tipos de acceso:

- El 100 % de las instrucciones tardan 1 ciclo en el acceso a la I-Cache.
- El 20 % (loads) de las instrucciones tardan 7,18 ciclos en el acceso a los datos.
- El 10 % (stores) de las instrucciones tardan 9,2 ciclos.

Es decir, el tiempo medio de acceso por cada 130 accesos sería:

$$T.\text{acceso} = \frac{100}{130} \times 1 + \frac{20}{130} \times 7,18 + \frac{10}{130} \times 9,2 = 2,58 \text{ ciclos}.$$

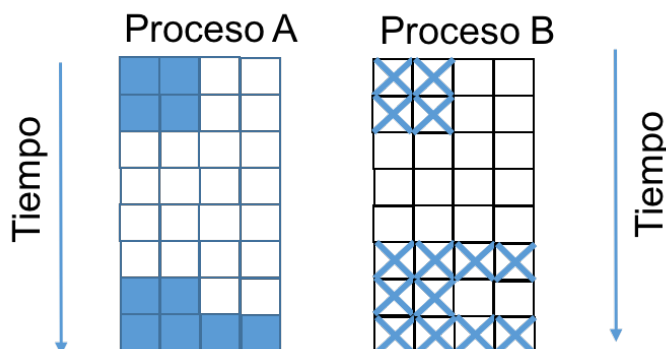
No obstante, en la pregunta también se ha considerado correcto el tiempo medio de acceso por instrucción. En este caso, todas las instrucciones realizan un acceso a la cache de instrucciones, que equivale a:

$$T.\text{acceso} = \frac{100}{100} \times 1 + \frac{20}{100} \times 7,18 + \frac{10}{100} \times 9,2 = 3,35 \text{ ciclos}.$$

e) Es write back puesto que el número de escrituras en L2 (stores) es inferior al número de L1.

□

2. (1.2 puntos) La siguiente figura muestra el diagrama de ejecución de 2 procesos en un procesador SMT. En concreto muestra para cada ciclo de **ejecución en solitario** las instrucciones lanzadas a ejecución desde la estación de reserva al operador correspondiente durante 8 ciclos. Cada fila muestra las instrucciones ejecutadas en un determinado ciclo. Asumiendo que no hay limitación de recursos (por ej. ni de ancho de banda de etapa de búsqueda, ni de E.R, ni de operadores, ...) en el procesador.

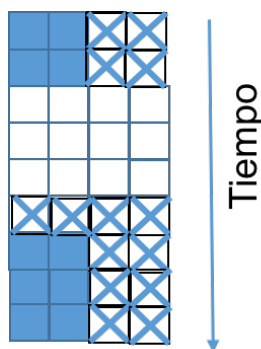


Se pide:

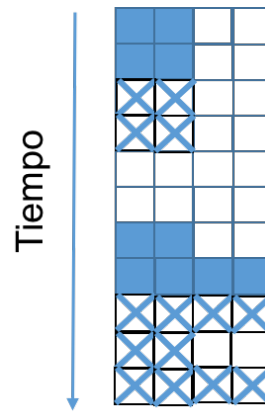
- ¿Cuál es el tiempo que tardaría cada una de las aplicaciones en ejecutarse si se ejecutasen los dos procesos simultáneamente en el mismo procesador SMT? Asume que si el procesador puede lanzar más instrucciones entre los dos procesos que el ancho de lanzamiento, se lanzan la mitad de cada proceso.
- En un procesador de grano grueso ideal que dispone de un predictor ideal en la etapa IF que realiza el cambio de contexto cuando estima que va a haber 2 ciclos durante los cuáles no se lance ninguna instrucción a ejecución, ¿en qué ciclo acabaría su ejecución la aplicación A y en qué ciclo la aplicación B asumiendo que la aplicación que primero empieza su ejecución es la A? Asume que el tiempo en realizar el cambio de contexto es nulo.
- ¿Cuánto tardarían en ejecutarse cada uno de los 2 procesos en un procesador de grano grueso (empieza el A) más realista que realiza el cambio de contexto en el instante en que han transcurrido ya 2 ciclos durante los cuales no se ha lanzado ninguna instrucción (el tiempo en realizar el cambio de contexto es nulo) ?

**Solución:**

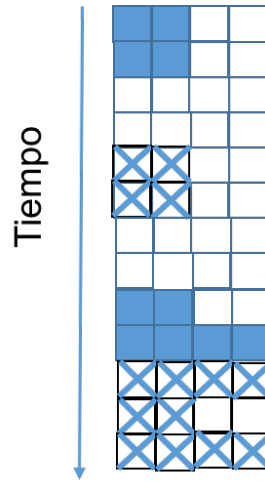
- Tardarían 9 ciclos las 2. Los 7 primeros ciclos de la ejecución SMT coincidirían con la unión de ambas gráficas. Los stalls se realizarían en paralelo. En el ciclo 8 ejecutarían 2 instrucciones cada aplicación, ya que nos dicen que no debe ejecutarse ninguna aplicación, y lo mismo en el ciclo 9.



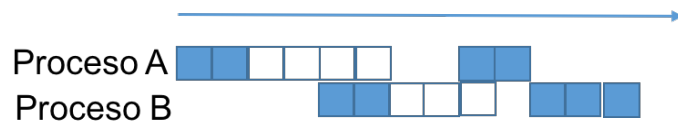
- En un procesador de grano grueso, la aplicación A acabaría en el ciclo 8 y la aplicación B en el ciclo 11.



c) La aplicación A acabaría en el ciclo 10 y la B en el ciclo 13.



Tiempo



3. (6 puntos) Se dispone de un procesador MIPS 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 (escritura en el bus común de datos) y C (confirmación de las instrucciones). El ROB tiene 32 entradas, identificándose la primera de ellas como entrada #0. 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 las unidades funcionales son las siguientes:

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

Se pretende evaluar el comportamiento del procesador ante el siguiente bucle, el cual calcula el producto escalar de dos vectores  $X$  e  $Y$ :

$$z = \sum_{i=0}^{n-1} x_i \cdot y_i \quad (1)$$

```

.data
x:    .double 1.0,2.0,3.0, ...
y:    .double 1.0,2.0,3.0, ...
z:    .double 0.0
.text
; r1 apunta a x
; r2 apunta a y
; r3 almacena el numero de componentes
loop: l.d    f0,0(r1)
      l.d    f1,0(r2)
      l.d    f2,8(r1)
      l.d    f3,8(r2)
      mul.d  f0,f0,f1
      mul.d  f2,f2,f3
      add.d  f4,f0,f4
      add.d  f5,f2,f5
      daddi  r1,r1,#16
      daddi  r2,r2,#16
      daddi  r3,r3,#-2
      sgt    r5,r3,r0
      bnez   r5,loop
      add.d  f4,f4,f5
      s.d    f4, z($gp)

```

La figura siguiente muestra el diagrama instrucciones–tiempo correspondiente a la primera iteración del bucle:

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
loop	l.d f0,0(r1)	IF	I	AC	L1	L2	WB	C																				
20	l.d f1,0(r2)		IF	I	AC	-	L1	L2	WB	C																		
24	l.d f2,8(r1)			IF	I	AC	-	-	L1	L2	WB	C																
28	l.d f3,8(r2)				IF	I	AC	-	-	-	L1	L2	WB	C														
32	mul.d f0,f0,f1					IF	I	-	-	M1	M2	M3	M4	WB	C													
36	mul.d f2,f2,f3						IF	I	-	-	-	-	-	M1	M2	M3	M4	WB	C									
40	add.d f4,f0,f4							IF	I	-	-	-	-	-	A1	A2	WB	-	-	C								
44	add.d f5,f2,f5								IF	I	-	-	-	-	-	-	-	-	-	A1	A2	WB	C					
48	dadd r1,r1,#16									IF	I	E1	-	-	WB	-	-	-	-	-	-	-	-	-	-	C		
52	dadd r2,r2,#16										IF	I	E1	-	-	WB	-	-	-	-	-	-	-	-	-	C		
56	dadd r3,r3,#-2											IF	I	E1	-	-	-	-	WB	-	-	-	-	-	-	-	C	
60	sgt r5,r3,r0												IF	I	-	-	-	-	-	E1	-	WB	-	-	-	-	C	
64	bnez r5,loop													IF	I	-	-	-	-	-	-	-	E1	WB	-	-	-	C
68	add.d f4,f4,f5														IF	I	-	-	-	-	-	A1	A2	-	WB	-	-	x
72	s.d f4,z(r28)															IF	I	AC	-	-	-	-	-	-	-	-	-	x
76	trap 0																IF	I	-	-	-	-	-	-	-	-	-	x
.etext	<nop>																		if	if	if	if	if	if	if	if	if	X
loop	l.d f0,0(r1)																											IF

Se solicita:

- Considera la instrucción `add.d f4,f0,f4` e indica el ciclo de reloj en el cual:
  - se lee el valor del registro fuente `f4`.
  - se libera la estación de reserva utilizada.
  - se almacena el resultado de la ejecución en el ROB.
  - se almacena el resultado en el registro destino `f4`.
- Suponiendo, para simplificar, que el ROB, las estaciones de reserva y los operadores estaban vacíos al comenzar la ejecución del bucle, indica los valores y las marcas de los registros `f0`, `f1`, `f2` y `f4` en el ciclo de reloj en el que la instrucción `l.d f0, 0(r1)` realiza la fase C. Los registros almacenan el valor 0.0 inicialmente.
- ¿Cuál será el número de ciclos consumido por una iteración cuando el predictor acierta? ¿Y cuando falla?.
- Si asumimos que X e Y son vectores de 512 elementos y que el procesador considerado funciona a una frecuencia de 2 GHz, ¿Cuántos MFLOPS ofrecerá el procesador ejecutando el código bajo estudio? Asume que el predictor falla en las predicciones de la primera y la última iteración.
- Con el objeto de mejorar las prestaciones, se diseña la siguiente generación del procesador para que sea superescalar de **2 vías**, duplicando asimismo el número de unidades funcionales disponibles. La fase IF proporciona dos instrucciones alineadas en una dirección múltiplo de 8. Dibuja el diagrama instrucciones-tiempo de una iteración intermedia (en la que el predictor acierta). Refleja en el diagrama **sólo las instrucciones de esa iteración**.

**Solución:**

- Considera la instrucción `add.d f4,f0,f4` e indica el ciclo de reloj en el cual:
  - se lee el valor del registro fuente `f4`: Ciclo 8
  - se libera la estación de reserva utilizada: Ciclo 16
  - se almacena el resultado de la ejecución en el ROB: Ciclo 16
  - se almacena el resultado en el registro destino `f4`: Ciclo 19
- Suponiendo, para simplificar, que el ROB, las estaciones de reserva y los operadores estaban vacíos al comenzar la ejecución del bucle, indica los valores y las marcas de los registros `f0`, `f1`, `f2` y `f4` en el ciclo de reloj en el que la instrucción `l.d f0, 0(r1)` realiza la fase C.  
`f0 f1 f2 f4 Valor 1.0 0.0 0.0 0.0 Marca #4 #1 #5 -`

- c) ¿Cuál será el número de ciclos consumido por una iteración cuando el predictor acierta?  
¿Y cuando falla?  
Acierto: 13 ciclos  
Fallo: 26 ciclos
- d) Si asumimos que X e Y son vectores de 512 elementos y que el procesador considerado funciona a una frecuencia de 2 GHz, ¿Cuántos MFLOPS ofrecerá el procesador ejecutando el código bajo estudio? Asume que el predictor falla en las predicciones de la primera y la última iteración.  
Operaciones FP: 4 por iteración \* 256 (el bucle está desenrollado) = 1024 ó 1025 (considerando el add.d de fuera del bucle) op FP  
Tiempo por iteración:  $254 \cdot 13 + 2 \cdot 26 = 3354$  ciclos  
 $MFLOPS = \frac{1024}{3354} = 0,31$  op/ciclo =  $0,31 \cdot 2000 \text{ MHz} = 620 \text{ MFLOPS}$
- e) Con el objeto de mejorar las prestaciones, se diseña la siguiente generación del procesador para que sea superescalar de **2 vías**, duplicando asimismo el número de unidades funcionales disponibles. La fase IF proporciona dos instrucciones alineadas en una dirección de byte múltiplo de 8. Dibuja el diagrama instrucciones-tiempo de una iteración intermedia (en la que el predictor acierta). Refleja en el diagrama **sólo las instrucciones de esa iteración**.

PC	Instruccion	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
loop	l.d f0,0(r1)	IF	I	AC	L1	L2	WB	C													
20	l.d f1,0(r2)	IF	I	AC	L1	L2	WB	C													
24	l.d f2,8(r1)		IF	I	AC	-	L1	L2	WB	C											
28	l.d f3,8(r2)		IF	I	AC	-	L1	L2	WB	C											
32	mul.d f0,f0,f1			IF	I	-	-	M1	M2	M3	M4	WB	C								
36	mul.d f2,f2,f3			IF	I	-	-	-	-	M1	M2	M3	M4	WB	C						
40	add.d f4,f0,f4				IF	I	-	-	-	-	-	A1	A2	WB	C						
44	add.d f5,f2,f5				IF	I	-	-	-	-	-	-	-	A1	A2	WB	C				
48	dadd r1,r1,#16					IF	I	E1	-	WB	-	-	-	-	-	-	-	C			
52	dadd r2,r2,#16					IF	I	E1	-	WB	-	-	-	-	-	-	-	C			
56	dadd r3,r3,#-2						IF	I	E1	-	WB	-	-	-	-	-	-	C			
60	sgt r5,r3,r0						IF	I	-	-	-	E1	WB	-	-	-	-	-	C		
64	bnez r5,loop							IF	I	-	-	-	-	E1	WB	-	-	-	-	C	
68	add.d f4,f4,f5							IF	X												
loop	l.d f0,0(r1)								IF	I	AC	L1	L2	WB	-	-	-	-	-	-	C
20	l.d f1,0(r2)								IF	I	AC	L1	L2	-	-	WB	-	-	-	-	C

□

<b>Apellidos y Nombre:</b>	
----------------------------	--

### Ejercicio 3

### Diagrama instrucciones-tiempo

[illegible]





