



APELLIDOS Y NOMBRE _____

GRUPO _____ N° EXPEDIENTE _____

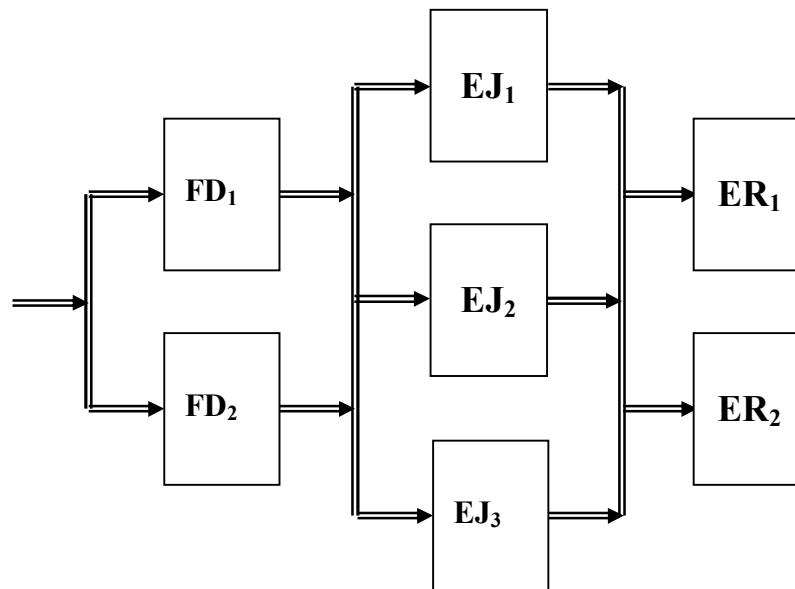
1).- Dada la estructura de unidades funcionales (FD_i : unidades de fetch y decodificación; EJ_1 y EJ_2 : unidades de ejecución de enteros, EJ_3 : unidad de ejecución en coma flotante; ER_i : unidades de escritura de resultado) de la figura que hay a continuación, y la secuencia de instrucciones $I_1 \dots I_{10}$, que tiene las siguientes particularidades:

- Todas las instrucciones tardan un ciclo de reloj en cada fase, excepto I_3 e I_5 cuyas ejecuciones duran 3 ciclos y las instrucciones I_2 , I_7 e I_8 que duran 2.
- I_3 , I_5 e I_7 operan con reales, por lo tanto, deben ejecutarse exclusivamente en la unidad EJ_3 .
- Existen las siguientes dependencias reales entre instrucciones:
 - I_5 depende de I_4
 - I_8 depende de I_7
 - I_9 depende de I_8

Se pide:

- a) Mostrar mediante una tabla la secuencia de ejecución de las instrucciones, en el caso de que no se permita **ningún tipo de desorden** (0,75 pts.)
- b) Repetir el apartado anterior para el caso de que se permita **desorden, tanto en ejecución, como en escritura de resultados.** (0,75 pts)

NOTA: En caso de coincidencia en una transición entre dos etapas por parte de dos instrucciones, tendrá prioridad la que se encuentre primero en la secuencia del código (subíndice menor).





1 cont.)-

SOLUCIÓN

Ejecución ordenada:

Ciclo	FD ₁	FD ₂	EJ ₁	Ej ₂	EJ ₃	ER ₁	ER ₂
1	I ₁	I ₂					
2	I ₃	I ₄	I ₁	I ₂			
3	I ₅	I ₆	I ₄	I ₂	I ₃	I ₁	
4					I ₃	I ₂	
5					I ₃		
6						I ₃	I ₄
7	I ₇	I ₈	I ₆		I ₅		
8					I ₅		
9					I ₅		
10	I ₉				I ₇	I ₅	I ₆
11					I ₇		
12						I ₇	
13		I ₁₀	I ₈				
14			I ₈				
15						I ₈	
16			I ₉	I ₁₀			
17						I ₉	I ₁₀

Ejecución desordenada:

Ciclo	FD ₁	FD ₂	Ventana	EJ ₁	Ej ₂	EJ ₃	ER ₁	ER ₂
1	I ₁	I ₂						
2	I ₃	I ₄	I₃ , I₂	I ₁	I ₂			
3	I ₅	I ₆	I₃ , I₄	I ₄	I ₂	I ₃	I ₁	
4	I ₇	I ₈	I ₅ , I₆	I ₆		I ₃	I ₂	I ₄
5	I ₉	I ₁₀	I ₅ , I ₇ , I ₈			I ₃	I ₆	
6			I₅ , I ₇ , I ₉ , I₁₀	I ₁₀		I ₅	I ₃	
7			I ₇ , I ₈ , I ₉			I ₅	I ₁₀	
8			I ₇ , I ₈ , I ₉			I ₅		
9			I₇ , I ₈ , I ₉			I ₇	I ₅	
10			I ₈ , I ₉			I ₇		
11			I ₈ , I ₉				I ₇	
12			I₈ , I ₉	I ₈				
13			I ₉	I ₈				
14			I ₉				I ₈	
15			I₉	I ₉				
16							I ₉	



APELLIDOS Y NOMBRE _____

GRUPO _____ N° EXPEDIENTE _____

2).-

Indique razonadamente si son correctas las siguientes afirmaciones:

- Una unidad de ejecución con estructura pipeline reduce el número de ciclos necesarios para ejecutar una instrucción. *.(0,5 ptos)*

FALSO. No reduce el número de ciclos, solapa las fases de ejecución de forma que el tiempo percibido es menor.

- En una configuración superescalar se puede conseguir que se obtengan instrucciones acabadas con un periodo menor que un ciclo de reloj. *(0,5 ptos)*

FALSO. La configuración superescalar incorpora varias líneas pipeline que, si operan de forma adecuada, conseguirán varias instrucciones finalizadas en un ciclo de reloj. Eso sólo puede ocurrir en una configuración supersegmentada.

- Un procesador supersegmentado presenta una estructura escalar con varias vías. *(0,5 ptos)*

FALSO. Se trata de una única línea pipeline en la que se realizan más de una fase por ciclo dentro de las unidades de ejecución que la componen, típicamente una operación por semiciclo.



APELLIDOS Y NOMBRE _____

GRUPO _____ N° EXPEDIENTE _____

3).-

Dado el siguiente fragmento de programa escrito en lenguaje simbólico, indicar todas las dependencias y pseudodependencias que tiene, especificando para cada una de ellas lo siguiente: entre que instrucciones se genera, con relación a que registro se produce, que tipo de dependencia o pseudodependencia es y si se puede solventar, cual es la solución. (0,8 Puntos)

$I_1: R_1 \rightarrow R_5$
 $I_2: R_5 + R_3 \rightarrow R_3$
 $I_3: R_4 \rightarrow R_5$
 $I_4: R_4 \rightarrow R_6$
 $I_5: R_2 \rightarrow R_4$

SOLUCIÓN

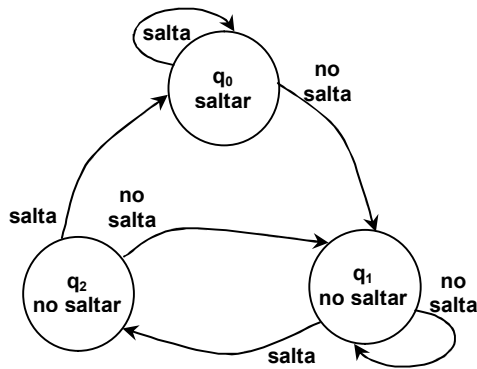
<i>Tipo de dependencia</i>	<i>Instrucc.</i>	<i>Reg.</i>	<i>¿Tiene solución?</i>
<i>Dependencia real o de escritura/lectura</i>	I_1 e I_2	R_5	<i>No. Hay que esperar</i>
<i>Pseudodependencia de salida o de escritura/escritura</i>	I_1 e I_3	R_5	<i>Sí. Renombrado de registros</i>
<i>Antidependencia o pseudodependencia de lectura/escritura</i>	I_2 e I_3	R_5	<i>Sí. Renombrado de registros</i>
	I_3 e I_5	R_4	
	I_4 e I_5	R_4	



APELLIDOS Y NOMBRE _____

GRUPO _____ N° EXPEDIENTE _____

- 1) **4).**- El autómata de una unidad dinámica de predicción de saltos sigue el esquema de la figura. Suponiendo que el criterio estático inicial sitúa al autómata en el estado q_0 . rellene la tabla de secuencia de estados e indique el número de aciertos y errores que se obtendrían en los supuestos especificados en ella. (1,2 Puntos)



Ciclo	Estado actual	Predicción	Resultado ejecución	Estado siguiente
1			salta	
2			no salta	
3			salta	
4			salta	
Errores:				

SOLUCIÓN

Atendiendo al esquema del autómata la tabla que daría de la siguiente forma:

Ciclo	Estado actual	Predicción	Resultado ejecución	Estado siguiente
1	q_0	salta	salta	q_0
2	q_0	salta	no salta	q_1
3	q_1	no saltar	salta	q_2
4	q_2	no saltar	salta	q_0
Errores:			3	



APELLIDOS Y NOMBRE _____

GRUPO _____ N° EXPEDIENTE _____

5).-

En un ordenador se ha implementado memoria entrelazada. Del bus de direcciones del procesador parten las líneas $A_3 \dots A_{35}$. En el selector de bloques de memoria se utilizan los bits A_3, A_4, A_{34} y A_{35} . Se pide:

- Espacio de direccionamiento físico máximo del procesador.
- Tamaño del bus de datos.
- Tipo de entrelazado empleado (Nivel alto (bancos), nivel bajo (módulos), mixto).
- Número de bancos de memoria y su tamaño (si hay más de un banco de memoria).
- Número de módulos dentro de cada banco y su tamaño (si hay más de uno).
- Tamaño de un módulo de memoria.

(0,7 ptos.)

Solución

- $A_0 \dots A_{35}$ implican 36 líneas de direcciones, por tanto la capacidad máxima de direccionamiento de memoria física es de 2^{36} bytes, es decir, 64 GBytes.
- Los bits de menos peso de ABus han desaparecido porque el DBus puede transferir varios bytes en un único acceso. Como faltan 3 bits de ABus: $A_0 \dots A_2$, significa que en un acceso se leen 2^3 bytes, es decir, 8 bytes o 64 bits, que es el tamaño de DBus.
- Como se usan bits de la parte alta y de la parte baja de ABus, significa que se usa entrelazado de nivel alto y de nivel bajo.
- Al utilizarse 2 bits de la parte alta de ABus, tenemos 2^2 Bancos, es decir 4 Bancos de memoria. Como se usan los bits A_{34} y A_{35} significa que tenemos toda la memoria implementada y por lo tanto cada Banco de memoria tiene $64/4 = 16$ GBytes.
- Al utilizarse 2 bits de la parte baja de ABus, tenemos 2^2 módulos, es decir 4 módulos de memoria por cada Banco.
- Cada módulo de memoria tiene $16/4 = 4$ GBytes

**6).-**

A).- De un procesador vectorial segmentado se conocen los siguientes datos:

- Frecuencia = 500MHz.
- Número y tipo de unidades vectoriales: dos de carga/almacenamiento, una de suma, una de multiplicación y una de división.
- Tiempos de arranque:
 - Instrucciones de carga y almacenamiento vectorial: 15 ciclos de reloj.
 - Instrucciones de suma/resta vectorial: 6 ciclos de reloj.
 - Instrucciones de multiplicación vectorial: 7 ciclos de reloj.
 - Instrucciones de división vectorial: 15 ciclos de reloj.
- Tiempo de bucle vectorial: 20 ciclos de reloj.
- Longitud máxima de vector = 64 componentes.

Suponiendo que el programa a ejecutar es el siguiente:

0→F0	;	[I1]
a→F1	;	[I2]
(Rx)→V1	;	[I3]
SNESV F0, V1	;	[I4]
(Ry)→V2	;	[I5]
F1*V1→V3	;	[I6]
V2+V3→V4	;	[I7]
V4→(Ry)	;	[I8]
CVM	;	[I9]

Se pide calcular:

- Indica de forma algebraica que operaciones se realizan sobre los vectores X e Y, y si esas operaciones afectan a todos los elementos del vector. (0,3 ptos.)
- El número de convoyes que se lanzan, indicando además para cada uno de ellos las instrucciones que lo componen y su tiempo de arranque. Considere que no se admite el encadenamiento de instrucciones en la formación de los convoyes. (0,3 ptos.)
- Repita el apartado a) suponiendo que se admite el encadenamiento de instrucciones. (0,3 ptos.)
- Calcule R_{∞} para los dos supuestos anteriores. (0,3 ptos.)

B).- Un procesador vectorial ejecuta una aplicación vectorizable en un 40% a una velocidad de 800 MFLOPS. Si su rendimiento escalar nominal es de 150 MFLOPS ¿Cuál es el Speed-up (aceleración) de la máquina? (0,4 ptos.)

**Solución**

Apartado A.a El fragmento hace: $\vec{R}y = a \cdot \vec{R}x + \vec{R}y$ para todos los elementos de $\vec{R}x$ distintos de 0.

	Convoy	Instrucciones	Tiempo de arranque
Apartado A.b	1º	[I3]	15 ciclos
	2º	[I4], [I5]	MAX(6+15)=15 ciclos
	3º	[I6]	7 ciclos
	4º	[I7]	6 ciclos
	5º	[I8]	15 ciclos

	Convoy	Instrucciones	Tiempo de arranque
Apartado A.c	1º	[I3], [I4], [I5], [I6]	15+max(6,15)+ +7=37 ciclos
	2º	[I7], [I8]	6+15=21 ciclos

Apartado A.d

$$R_{\infty} = \lim_{n \rightarrow \infty} R_n = \frac{Op \cdot Freq}{\frac{T_{arr} + T_{loop}}{MVL} + T_{campanadas}} = \frac{3 \cdot 500MHz}{\frac{T_{arr} + 20}{64} + T_{campanadas}} \text{ (MFLOPS)}$$

Apartado	$T_{arranque}$	$T_{campanada}$	R_{∞} (MFLOPS)
A.b	58	5	241,21
A.c	58	2	466,02

Apartado B

$$S = \frac{R}{R_{esc}} = \frac{800}{150} = 5,33$$



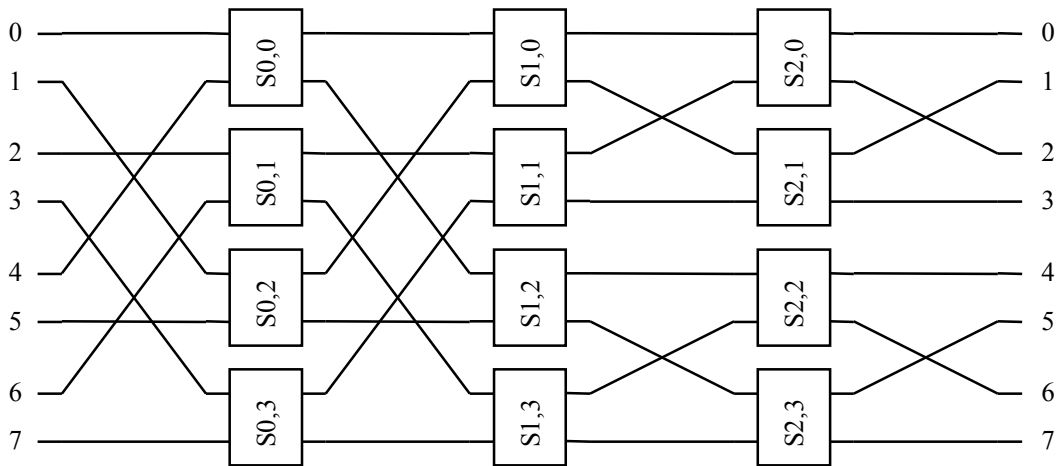
7).-

Sabiendo que en MINs la conexión mariposa β_i^k intercambia los dígitos 0 e i-ésimo:

$$\beta_i^k(x_{n-1} \dots x_{i+1} x_i x_{i-1} \dots x_0) = (x_{n-1} \dots x_{i+1} x_0 x_{i-1} \dots x_1 x_i)$$

Dibuja a continuación una MIN de 8x8 con tres etapas de conmutación y con todas las etapas de interconexión de tipo mariposa. Obviamente esta red deberá garantizar que es posible (si no hay bloqueos creados por otras conexiones en curso) conectar cualquier origen con cualquier destino. (1 pto.)

Solución



Esta solución es una de entre todas las posibles. Es obligado que cada uno de los bits de la dirección origen llegue una vez a una etapa de conmutación como el bit de menor peso y que salgan en el mismo orden que entraron. En esta solución se cumplen estos requisitos.

Etapa	C0	C1	C2	C3
Tipo	Mariposa orden 2	Mariposa orden 2	Mariposa orden 1	Mariposa orden 1
Bits al final de la etapa	O ₀ , O ₁ , O ₂	O ₂ , O ₁ , O ₀	O ₂ , O ₀ , O ₁	O ₂ , O ₁ , O ₀



APELLIDOS Y NOMBRE _____

GRUPO _____ N° EXPEDIENTE _____

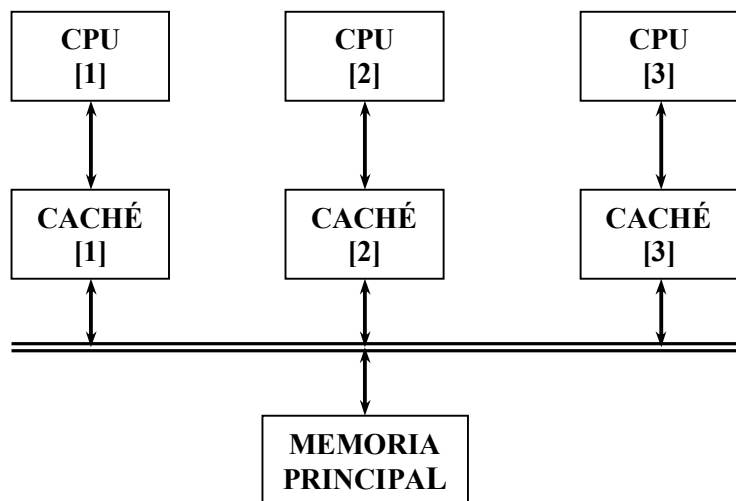
8).-

A).- En el sistema de la figura se sabe que las tres cachés siguen el protocolo de invalidación MESI, y que sobre una determinada línea de información, que inicialmente sólo está cargada y modificada en la caché 1, las CPUs realizan la secuencia de accesos indicada en la tabla.

Se pide que se rellene el resto de dicha tabla indicando:

- ¿Qué operaciones se desencadenan en el bus?. Especificar también que “snoopy” lanza cada una de ellas. (0,25 ptos.)
- ¿En qué estado queda la línea en cada caché al terminar cada acceso de CPU?. (0,25 ptos.)

B).- Repetir el apartado A suponiendo que las cachés usan el protocolo de actualización Dragón. (0,5 ptos.)





8 cont).-

SOLUCIÓN**A)**

Acceso	Operaciones sobre el bus	Estado final caché		
		[1]	[2]	[3]
Lectura CPU [3] (0.2 ptos.)	[3]BusRd(S) [1]BusWB	S	-	S
Descarga CPU [1] (0.2 ptos.)		-	-	S
Lectura CPU [2] (0.2 ptos.)	[2]BusRd(S) [3]S	-	S	S

B)

Acceso	Operaciones sobre el bus	Estado final caché		
		[1]	[2]	[3]
Lectura CPU [3] (0.2 ptos.)	[3]BusRd(S) [1]BusWB	SC	-	SC
Descarga CPU [1] (0.2 ptos.)		-	-	SC
Lectura CPU [2] (0.2 ptos.)	[2]BusRd(S) [3]S	-	SC	SC



APELLIDOS Y NOMBRE _____

GRUPO _____ N° EXPEDIENTE _____

9).-

Indica que eventos se producen en un sistema multiprocesador para mantener la coherencia de caché en un sistema que utiliza un protocolo de directorio plano con mapeado completo si:

- Hay un nodo peticionario que genera un WrMiss.
- Ese nodo no es el nodo origen.
- Existe otro nodo que es propietario de la línea.

(1 pto.)

Solución

- El nodo peticionario envía un mensaje al nodo origen solicitando la línea en la que quiere escribir para modificarla.
- El nodo origen detecta que el bit I.U. está a 1 y envía un mensaje al nodo propietario solicitando la línea para modificarla.
- El nodo propietario desactiva los bits V y D y envía la línea al nodo origen.
- El nodo origen desactiva el bit Pr. del propietario y activa el bit Pr. del peticionario. Envía la línea al peticionario.
- El peticionario recibe la línea, activa los bits V y D y modifica la línea.



APELLIDOS Y NOMBRE _____

GRUPO _____ N° EXPEDIENTE _____

1).- Dada la estructura de unidades funcionales (FD_i: unidades de fetch y decodificación; EJ₁ y EJ₂: unidades de ejecución de enteros, EJ₃: unidad de ejecución en coma flotante; ER_i: unidades de escritura de resultado) de la figura que hay a continuación, y la secuencia de instrucciones I₁...I₁₀, que tiene las siguientes particularidades:

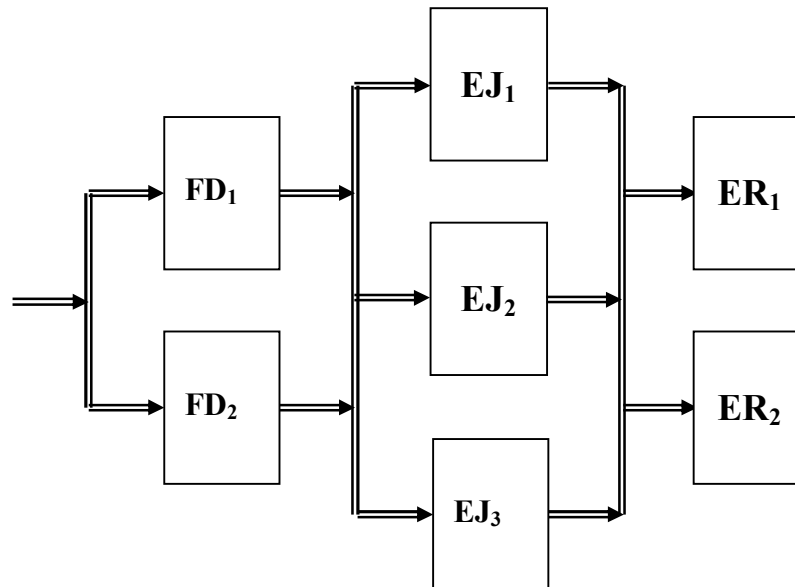
- Todas las instrucciones tardan un ciclo de reloj en cada fase, excepto I₃ e I₅ cuyas ejecuciones duran 3 ciclos y las instrucciones I₂, I₇ e I₈ que duran 2.
- I₃, I₅ e I₇ operan con reales, por lo tanto, deben ejecutarse exclusivamente en la unidad EJ₃.
- Existen las siguientes dependencias reales entre instrucciones:

I₅ depende de I₄I₈ depende de I₇I₉ depende de I₈

Se pide:

- Mostrar mediante una tabla la secuencia de ejecución de las instrucciones, en el caso de que no se permita **ningún tipo de desordenamiento** (0,75 ptos.)
- Repetir el apartado anterior para el caso de que se permita **desorden, tanto en ejecución, como en escritura de resultados.**(0,75 ptos)

NOTA: En caso de coincidencia en una transición entre dos etapas por parte de dos instrucciones, tendrá prioridad la que se encuentre primero en la secuencia del código (subíndice menor).





1 cont.)-

SOLUCIÓN

Ejecución ordenada:

Ciclo	FD ₁	FD ₂	EJ ₁	Ej ₂	EJ ₃	ER ₁	ER ₂
1	I ₁	I ₂					
2	I ₃	I ₄	I ₁	I ₂			
3	I ₅	I ₆	I ₄	I ₂	I ₃	I ₁	
4					I ₃	I ₂	
5					I ₃		
6						I ₃	I ₄
7	I ₇	I ₈	I ₆		I ₅		
8					I ₅		
9					I ₅		
10	I ₉				I ₇	I ₅	I ₆
11					I ₇		
12						I ₇	
13		I ₁₀	I ₈				
14			I ₈				
15						I ₈	
16			I ₉	I ₁₀			
17						I ₉	I ₁₀

Ejecución desordenada:

Ciclo	FD ₁	FD ₂	Ventana	EJ ₁	Ej ₂	EJ ₃	ER ₁	ER ₂
1	I ₁	I ₂						
2	I ₃	I ₄	I₁ , I₂	I ₁	I ₂			
3	I ₅	I ₆	I₃ , I₄	I ₄	I ₂	I ₃	I ₁	
4	I ₇	I ₈	I ₅ , I₆	I ₆		I ₃	I ₂	I ₄
5	I ₉	I ₁₀	I ₅ , I ₇ , I ₈			I ₃	I ₆	
6			I₅ , I ₇ , I ₉ , I₁₀	I ₁₀		I ₅	I ₃	
7			I ₇ , I ₈ , I ₉			I ₅	I ₁₀	
8			I ₇ , I ₈ , I ₉			I ₅		
9			I₇ , I ₈ , I ₉			I ₇	I ₅	
10			I ₈ , I ₉			I ₇		
11			I ₈ , I ₉				I ₇	
12			I₈ , I ₉	I ₈				
13			I ₉	I ₈				
14			I ₉				I ₈	
15			I₉	I ₉				
16							I ₉	



APELLIDOS Y NOMBRE _____

GRUPO _____ N° EXPEDIENTE _____

2).- Un microprocesador que posee un fichero de registros “ve” en cada momento los siguientes registros de enteros:

- R0 a R7 para los parámetros que le ha pasado la rutina anterior.
- R8 a R15 para las variables locales.
- R16 a R23 para pasar parámetros a las rutinas que llame.
- R24 a R63 para variables globales.

Se pide:

1. ¿Cuántos registros de enteros hay implementados en el micro, sabiendo que el fichero de registros contiene 16 niveles de anidamiento de rutinas?. Justifique brevemente la respuesta. (0,3 ptos.)
2. ¿Cuáles de los registros R2, R10, R18 y R50 seguirían siendo visibles después de ejecutar una llamada a una subrutina?. ¿En qué número de registro estarían ahora los que siguieran siendo visibles? (0,3 ptos.)
3. Partiendo de las condiciones iniciales, repetir el punto anterior suponiendo ahora que la instrucción que se ejecuta es un retorno de subrutina en vez de una llamada. (0,3 ptos.)

SOLUCIÓN

Apartado 1

Registros para parámetros:	$8 \times 16 = 128$
Registros para variables locales	$8 \times 16 = 128$
Registros para variables globales	$= 40$
TOTAL	296

Apartado 2

Apartado 3

R2	No se ve	Se ve como R18
R10	No se ve	No se ve
R18	Se ve como R2	No se ve
R50	Se ve como R50	Se ve como R50

B) Supóngase que en un microprocesador con salto retardado tenemos el siguiente fragmento de un programa **fuentes** en ensamblador:

[I₁] Instrucción que no es de salto

[I₂] Salto condicional a eti1

[I₃] Salto condicional a eti2

[I₄] Instrucción que no es de salto

¿En qué orden quedarán en el programa **ejecutable**? (0,6 ptos.)

Dado que hay dos instrucciones de salto contiguas, debe insertarse una instrucción NOP entre ellas:

[I₂] [I₁] [I₃] NOP [I₄]



APELLIDOS Y NOMBRE _____

GRUPO _____ N° EXPEDIENTE _____

3).-

Dado el siguiente fragmento de programa escrito en lenguaje simbólico, indicar todas las dependencias y pseudodependencias que tiene, especificando para cada una de ellas lo siguiente: entre que instrucciones se genera, con relación a que registro se produce, que tipo de dependencia o pseudodependencia es y si se puede solventar, cual es la solución. (0,8 Puntos)

$I_1: R_1 - R_2 \rightarrow R_4$
 $I_2: R_3 + R_5 \rightarrow R_6$
 $I_3: R_4 \rightarrow R_5$
 $I_4: R_6 \rightarrow R_2$
 $I_5: R_4 * R_6 \rightarrow R_4$

SOLUCIÓN

Instrucc.	Reg.	Tipo de dependencia	¿Cuándo se produce?	¿Tiene solución?
I_1 e I_3	R_4	Dependencia real o de escritura/lectura	Basta con que haya paralelismo	No. Hay que esperar
I_1 e I_5	R_4	Dependencia real o de escritura/lectura	Basta con que haya paralelismo	No. Hay que esperar
I_2 e I_4	R_6	Dependencia real o de escritura/lectura	Basta con que haya paralelismo	No. Hay que esperar
I_2 e I_5	R_6	Dependencia real o de escritura/lectura	Basta con que haya paralelismo	No. Hay que esperar
I_1 e I_4	R_2	Antidependencia o de lectura/escritura (pseudodependencia)	Sólo si emisión de instrucciones es también desordenada	Sí. Renombrado de registros
I_2 e I_3	R_5	Antidependencia o de lectura/escritura (pseudodependencia)	Sólo si emisión de instrucciones es también desordenada	Sí. Renombrado de registros
I_3 e I_5	R_4	Antidependencia o de lectura/escritura (pseudodependencia)	Sólo si emisión de instrucciones es también desordenada	Sí. Renombrado de registros
I_1 e I_5	R_4	Dependencia de salida o de escritura/escritura (pseudodependencia)	Se produce con cualquier desorden	Sí. Renombrado de registros



APELLIDOS Y NOMBRE _____

GRUPO _____ N° EXPEDIENTE _____

4).- Indique si es cierta o falsa la siguiente afirmación, **justificando brevemente la respuesta:**

- Los multiprocesadores comparten una única memoria, común a todos ellos, y por tanto, los procesadores acceden a toda la memoria de la misma forma y con el mismo tiempo de acceso. (0,4 ptos.)

FALSO: En un primer nivel los multiprocesadores se dividen en UMA y NUMA. Los NUMA (Non Uniform Memory Access), comparten un único espacio de direccionamiento. Pero su estructura física es distribuida por lo que los bloques de memoria más próximos tienen costes de acceso menores.

- Al aumentar el número de unidades funcionales aumenta el grado de paralelismo, pero disminuye habitualmente la eficiencia. (0,4 ptos.)

CIERTO, porque aumenta el número medio de acciones simultáneas, pero debido a los conflictos, no aumenta en la misma proporción que las unidades funcionales, por lo cual la proporción entre estas dos magnitudes (la eficiencia) disminuirá.

- Los conflictos de paralelismo surgen de la ejecución desordenada de instrucciones. (0,4 ptos.)

FALSO: también se producen con ejecución ordenada; basta que se ejecute con algún grado de paralelismo.



APELLIDOS Y NOMBRE _____

GRUPO _____ N° EXPEDIENTE _____

5).-

En un ordenador se ha implementado memoria entrelazada. Del bus de direcciones del procesador parten las líneas $A_2 \dots A_{33}$. En el selector de bloques de memoria se utilizan los bits $A_2 \dots A_4, A_{33}$. Se pide:

- Espacio de direccionamiento físico máximo del procesador.
- Tamaño del bus de datos.
- Tipo de entrelazado empleado (Nivel alto (bancos), nivel bajo (módulos), mixto).
- Número de bancos de memoria y su tamaño (si hay más de un banco de memoria).
- Número de módulos dentro de cada banco y su tamaño (si hay más de uno).
- Tamaño de un módulo de memoria.

(0,7 ptos.)

Solución

- $A_0 \dots A_{33}$ implican 34 líneas de direcciones, por tanto la capacidad máxima de direccionamiento de memoria física es de 2^{34} bytes, es decir, 16 GBytes.
- Los bits de menos peso de ABus han desaparecido porque el DBus puede transferir varios bytes en un único acceso. Como faltan 2 bits de ABus: A_0 y A_1 , significa que en un acceso se leen 2^2 bytes, es decir, 4 bytes o 32 bits, que es el tamaño de DBus.
- Como se usan bits de la parte alta y de la parte baja de ABus, significa que se usa entrelazado de nivel alto y de nivel bajo.
- Al utilizarse 1 bit de la parte alta de ABus, tenemos 2 Bancos de memoria. Como se usa el bit A_{33} significa que tenemos toda la memoria implementada y por lo tanto cada Banco de memoria tiene $16/2 = 8$ GBytes.
- Al utilizarse 3 bits de la parte baja de ABus, tenemos 2^3 módulos, es decir 8 módulos de memoria por cada Banco.
- Cada módulo de memoria tiene $8/8 = 1$ GByte



APELLIDOS Y NOMBRE _____

GRUPO _____ N° EXPEDIENTE _____

6).-**A).-** De un procesador vectorial segmentado se conocen los siguientes datos:

- Frecuencia = 500MHz.
- Número y tipo de unidades vectoriales: dos de carga/almacenamiento, una de suma, una de multiplicación y una de división.
- Tiempos de arranque:
 - Instrucciones de carga y almacenamiento vectorial: 15 ciclos de reloj.
 - Instrucciones de suma/resta vectorial: 6 ciclos de reloj.
 - Instrucciones de multiplicación vectorial: 7 ciclos de reloj.
 - Instrucciones de división vectorial: 15 ciclos de reloj.
- Tiempo de bucle vectorial: 20 ciclos de reloj.
- Longitud máxima de vector = 64 componentes.

Suponiendo que el programa a ejecutar es el siguiente:

0→F0	:[I1]
a→F1	:[I2]
(Rx)→V1	:[I3]
(Ry)→V2	:[I4]
F1*V1→V3	:[I5]
V2+V3→V4	:[I6]
SNESV F0, V1	:[I7]
V4→(Ry)	:[I8]
CVM	:[I9]

Se pide calcular:

- a) Indica de forma algebraica que operaciones se realizan sobre los vectores X e Y, y si esas operaciones afectan a todos los elementos del vector. (0,3 pts.)
- b) El número de convoyes que se lanzan, indicando además para cada uno de ellos las instrucciones que lo componen y su tiempo de arranque. Considere que no se admite el encadenamiento de instrucciones en la formación de los convoyes. (0,3 pts.)
- c) Repita el apartado a) suponiendo que se admite el encadenamiento de instrucciones. (0,3 pts.)
- d) Calcule R_{∞} para los dos supuestos anteriores. (0,3 pts.)

B).- Un procesador vectorial ejecuta una aplicación vectorizable en un 60% a una velocidad de 900 MFLOPS. Si su rendimiento escalar nominal es de 200 MFLOPS ¿Cuál es el Speed-up (aceleración) de la máquina? (0,4 pts.)



Solución

Apartado A.a El fragmento hace: $\vec{R}y = a \cdot \vec{R}x + \vec{R}y$ almacenando todos los elementos de $\vec{R}y$ distintos de 0.

	Convoy	Instrucciones	Tiempo de arranque
Apartado A.b	1º	[I3], [I4]	MAX(15+15)=15 ciclos
	2º	[I5]	7 ciclos
	3º	[I6]	6 ciclos
	4º	[I7]	6 ciclos
	5º	[I8]	15 ciclos

	Convoy	Instrucciones	Tiempo de arranque
Apartado A.c	1º	[I3], [I4], [I5], [I6]	max(15,15)+ +7+6=28 ciclos
	2º	[I7], [I8]	6+15=21 ciclos

Apartado A.d

$$R_{\infty} = \lim_{n \rightarrow \infty} R_n = \frac{Op \cdot Freq}{\frac{T_{arr} + T_{loop}}{MVL} + T_{campanadas}} = \frac{3 \cdot 500MHz}{\frac{T_{arr} + 20}{64} + T_{campanadas}} \text{ (MFLOPS)}$$

Apartado	T _{arranque}	T _{campanada}	R _∞ (MFLOPS)
A.b	49	5	246,79
A.c	49	2	487,31

Apartado B

$$S = \frac{R}{R_{esc}} = \frac{900}{200} = 4,5$$



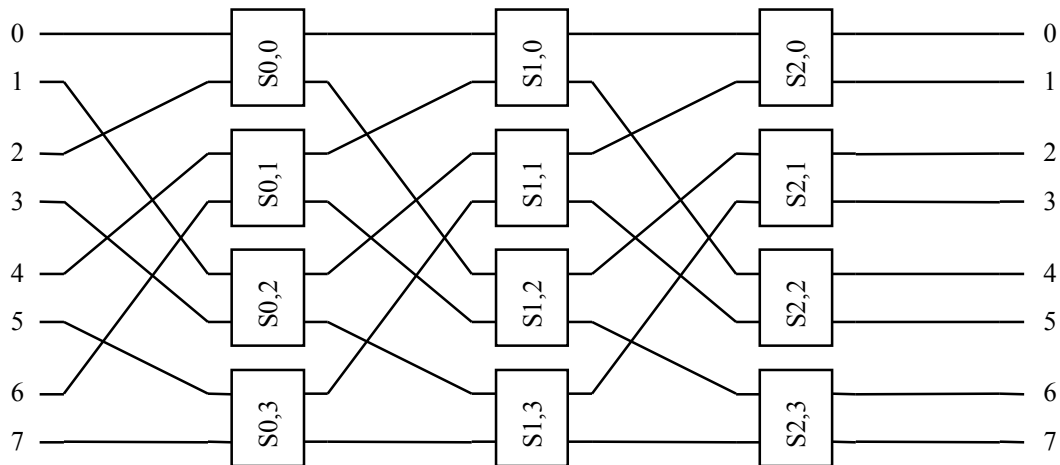
7).-

Sabiendo que en MINs la conexión en línea base δ_i^k realiza una rotación cíclica a la derecha de los $i+1$ bits menos significativos de X:

$$\delta_i^k(x_{n-1} \dots x_{i+1} x_i x_{i-1} \dots x_0) = (x_{n-1} \dots x_{i+1} x_0 x_i x_{i-1} \dots x_1)$$

Dibuja a continuación una MIN de 8x8 con tres etapas de conmutación y con todas las etapas de interconexión de tipo línea base del orden que estimes oportuno. Obviamente esta red deberá garantizar que es posible (si no hay bloqueos creados por otras conexiones en curso) conectar cualquier origen con cualquier destino. (1 pto.)

Solución



Esta solución es una de entre todas las posibles. Es obligado que cada uno de los bits de la dirección origen llegue una vez a una etapa de conmutación como el bit de menor peso y que salgan en el mismo orden que entraron. En esta solución se cumplen estos requisitos.

Etapa	C0	C1	C2	C3
Tipo	Línea base orden 2	Línea base orden 2	Línea base orden 2	Línea base orden 0
Bits al final de la etapa	O ₀ , O ₂ , O ₁	O ₁ , O ₀ , O ₂	O ₂ , O ₁ , O ₀	O ₂ , O ₁ , O ₀



APELLIDOS Y NOMBRE _____

GRUPO _____ N° EXPEDIENTE _____

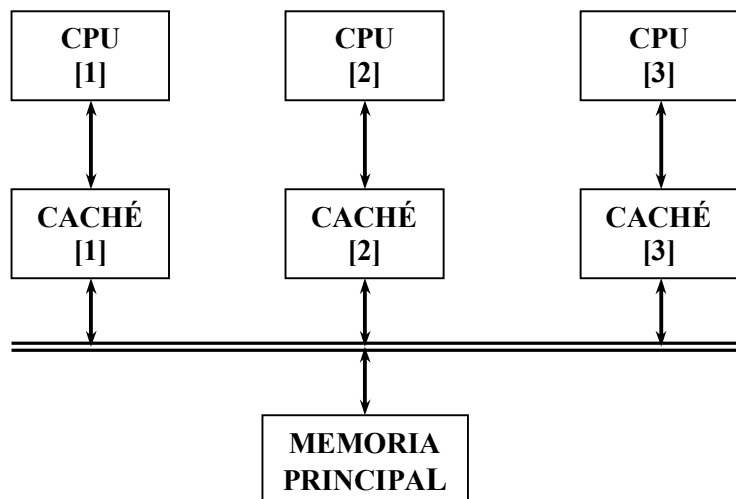
8).-

A).- En el sistema de la figura se sabe que las tres cachés siguen el protocolo de invalidación MESI, y que sobre una determinada línea de información, que inicialmente sólo está cargada y modificada en la caché 1, las CPUs realizan la secuencia de accesos indicada en la tabla.

Se pide que se rellene el resto de dicha tabla indicando:

- ¿Qué operaciones se desencadenan en el bus?. Especificar también que “snoopy” lanza cada una de ellas. (0,25 pts.)
- ¿En qué estado queda la línea en cada caché al terminar cada acceso de CPU?. (0,25 pts.)

B).- Repetir el apartado A suponiendo que las cachés usan el protocolo de actualización Dragón. (0,5 pts.)





8 cont).-

SOLUCIÓN**A)**

Acceso	Operaciones sobre el bus	Estado final caché		
		[1]	[2]	[3]
Lectura CPU [3] (0.2 ptos.)	[3]BusRd(S) [1]BusWB	S	-	S
Descarga CPU [1] (0.2 ptos.)		-	-	S
Escritura CPU [3] (0.2 ptos.)	[2]BusRdx	-		M

B)

Acceso	Operaciones sobre el bus	Estado final caché		
		[1]	[2]	[3]
Lectura CPU [3] (0.2 ptos.)	[3]BusRd(S) [1]BusWB	SC	-	SC
Descarga CPU [1] (0.2 ptos.)		-	-	SC
Escritura CPU [3] (0.2 ptos.)	[2]BusUpd(\bar{S})	-	-	M



APELLIDOS Y NOMBRE _____

GRUPO _____ N° EXPEDIENTE _____

9).-

Indica que eventos se producen en un sistema multiprocesador para mantener la coherencia de caché en un sistema que utiliza un protocolo de directorio plano con mapeado completo si:

- Hay un nodo peticionario que genera un WrHit.
- Ese nodo no es el nodo origen.
- Existen otros nodos que también disponen de una copia correcta de esa línea.

(1 pto.)

Solución

- El nodo peticionario envía un mensaje al nodo origen solicitando la propiedad de la línea en la que quiere escribir.
- El nodo origen localiza el primer nodo que también tiene esa línea y le da la orden de invalidarla.
- El nodo desactiva el bit V y envía un reconocimiento al nodo origen.
- El nodo origen desactiva el bit Pr. de dicho nodo y repite el mismo proceso con el resto de nodos que tienen esa línea.
- El nodo origen activa el bit I.U. y envía la confirmación de propiedad al peticionario.
- El peticionario recibe la confirmación, activa el bit D y modifica la línea.