

# Parcial 2 C

---

**1. La técnica de conmutación de reducción consiste en:**

- Enviar un único elemento a un único procesador.
- Enviar varios elementos a la vez a varios procesadores.
- Enviar un único elemento pero que es recibido por varios procesadores.
- **A partir de varios elementos, utilizamos una función para convertirlos en uno solo, que recibe un único procesador**

**2. Una red Omega es:**

- Una red jerárquica.
- **Una red indirecta.**
- Una red basada en buses de contención.
- Una red directa.

**3. ¿Cuál es la unidad de transferencia en la técnica Worm-Hole?**

- El paquete.
- El phit.
- El flit.
- **El mensaje.**

**4. Un computador paralelo tiene 15 nodos conectados mediante un árbol binario. ¿Cuál es el valor del diámetro de la red?**

- 6
- 8
- 7
- 5

**5. Indica cuál de las siguientes afirmaciones es correcta:**

- La prevención de los interbloqueos es una estrategia pesimista.
- **Los bloqueos activos no se pueden dar nunca en las redes de interconexión.**
- La recuperación de los interbloqueos es una estrategia pesimista.
- Los bloqueos activos no se pueden solucionar.

**6. En el diseño de una red de interconexión influye:**

- Solamente influye la topología y el control de flujo.
- Solamente influye la topología y el encaminamiento.
- Solamente incluye la topología.

- La topología, el control de flujo y el encaminamiento.

**7. La estructura de paralelismo de "Divide y Vencerás" consiste en:**

- Distribuir diferentes partes de una tarea entre varios procesadores sin una relación de dependencia tipo master-slave.
- **Se divide una tarea en diferentes funciones que se aplican consecuentemente, de forma secuencial, sobre todos los datos.**
- "Divide y Vencerás" no tiene nada que ver con paralelismo.
- Es la misma estrategia que "master-slave"

**8. Un multiprocesador de 64 procesadores utiliza una red Omega con conmutadores de grado 2.**

**¿Cuántos conmutadores utiliza en total?**

- 60
- 32
- 64
- **192**

**9. ¿De qué tipo es la red de comunicación que se ha empleado en el laboratorio para hacer las prácticas?**

- MPP.
- Red de computadores.
- **Cluster.**
- Cluster Beowulf.

**10. En cierto momento del transcurso de la ejecución paralela de una aplicación paralela en un multicamputador de 4 nodos, todos sus nodos (N0, N1, N2, N3) envían un dato distinto al nodo N0. Dicho nodo espera a que lleguen todos los datos y realiza la siguiente operación antes de continuar con la ejecución de la aplicación:**

**Resultado = dato\_N0 + dato\_N1 + dato\_N2 + dato\_N3.**

¿De qué tipo de comunicación estamos hablando?

- Reducción.
- Difusa
- Dispersión.
- **Acumulación.**

**11. ¿Cuál es el factor de aceleración, ganancia en velocidad o speed-up que puede conseguir como máximo si tiene 2 nodos conectados formando una máquina paralela y una aplicación cuyo 75% del código ha conseguido parallelizar? Suponga que el tiempo de sobrecarga no influye.**

- **1,666**
- 4
- 1,250

- 2

**12. En un multicomputador, ¿Qué se utiliza para permitir una comunicación asíncrona entre cualesquiera de sus nodos?**

- Una red directa o estática.
- Es imposible establecer una comunicación asíncrona en un computador paralelo.
- **Buffers intermedios.**
- Enlaces con más ancho de banda.

**13. El diámetro de una malla abierta es:**

- $2^*(r/2 - 1)$ , siendo la r la raíz cuadrada del número de nodos.
- **2\*(r - 1), siendo la r la raíz cuadrada del número de nodos.**
- $2^*N/2$  siendo N el número de nodos.
- $2^*(r - 1)$ , siendo r el número de nodos.

**14. El paralelismo de datos:**

- Se encuentra implícito en funciones lógicas.
- Es un tipo de paralelismo que nunca se encuentra de forma implícita.
- Se encuentra implícito en funciones aritméticas.
- **Se encuentra implícito en operaciones con imágenes.**

**15. ¿Cuántos conmutadores tiene en la segunda columna de conmutadores de una red multi-etapa tipo Delta 25x16?**

- 4
- 8
- 16
- 5

**16. ¿Cuál es el máximo factor de aceleración, ganancia en velocidad o speed-up que puede conseguir como máximo si tiene 5 nodos conectados, el tiempo de sobrecarga no influye y dispone de una aplicación totalmente paralelizable?**

- 10
- 3
- 5
- 2,5

**17. ¿Cuántas etapas o columnas de conmutadores tiene una red multietapa tipo Delta 25x16?**

- 4
- 25
- 5

- 2

#### **18. ¿Dondé está implícito el paralelismo de datos?**

- Utilizando funciones ILP.
- **En las operaciones con estructuras de datos.**
- En algoritmos puramente secuenciales.
- Utilizando estructuras DDPP.

#### **19. Diferencia entre los multiprocesadores y multicomputadores**

- **Los multiprocesadores comparten el mismo espacio de memoria**
- Los procesadores tienen la capacidad de crear hilos y los computadores no.
- Procesar un hipercubo es solo posible en un multicomputador mientras que en el multiprocesador no.
- Los multiprocesadores están deprecados en diferencia a los multicomputadores.

#### **20. En cuanto al modo de programación SPMD**

- Hace de la caché un elemento accedido en exceso.
- Es equivalente al modo SWPT.
- Podremos secuencializar flujos de datos consecutivamente.
- **Un solo programa que pueda actuar sobre varios flujos de datos.**

#### **21. Grado de un cubo de 6 dimensiones:**

- **G = D = n = 6; Grado = 6; n = log2(N); N = número de nodos.**
- $G = D/2 = n = 6; \text{Grado} = 6; n = \log_2(N); N = \text{número de nodos.}$
- $G = D = n = 6; \text{Grado} = 6; n = 2\log_2(N); N = \text{número de nodos.}$
- $G = D^2 = n = 6; \text{Grado} = 6; n = \log_2(N/2); N = \text{número de nodos.}$

#### **22. Cálculo de aceleración:**

- $S = T_s^2 / (T_p/M)(n).$
- **S = Ts/Tp(n).**
- $S = T_s * T_p(n).$
- $S = T_s - S T_p(n).$

**Question 1**

Answer saved

Marked out of 1.00

Flag question

La técnica de comunicación de reducción consiste en:

Select one:

- a. Enviar un único elemento a un único procesador
- b. Enviar varios elementos a la vez a varios procesadores
- c. Enviar un único elemento pero que es recibido por varios procesadores
- d. A partir de varios elementos, utilizar una función para convertirlos en uno solo, que recibe un único procesador

**Question 2**

Not yet answered

Marked out of 1.00

Flag question

Una red Omega es

Select one:

- a. Una red jerárquica
- b. Una red indirecta
- c. Una red basada en buses de contención
- d. Una red directa

**Question 3**

Not yet answered

Marked out of 1.00

Flag question

¿Cuál es la unidad de transferencia en la técnica de conmutación Worm-Hole?

Select one:

- a. El paquete
- b. El phit
- c. El flit
- d. El mensaje

Un computador paralelo tiene 15 nodos conectados mediante un árbol binario. ¿Cuál es el valor del diámetro de la red?

Select one:

- a. 6
- b. 8
- c. 7
- d. 5

[Next](#)**Question 5**

Not yet answered

Marked out of 1.00

Flag question

Indica cuál de las siguientes afirmaciones es correcta:

Select one:

- a. La prevención de los interbloqueos es una estrategia pesimista
- b. Los bloqueos activos no se pueden dar nunca en las redes de interconexión directas
- c. La recuperación de los interbloqueos es una estrategia pesimista
- d. Los bloqueos activos no se pueden solucionar

**Question 10**

Not yet answered

Marked out of 1.00

Flag question

Un multiprocesador de 64 procesadores utiliza una red Omega con conmutadores de grado 2. ¿Cuántos conmutadores utiliza en total?

Select one:

- a. 60
- b. 32
- c. 64
- d. 192

**Question 7**

Answer saved

Marked out of 1.00

Flag question

En el diseño de una red de interconexión influye

Select one:

- a. Solamente influye la topología y el control de flujo
- b. Solamente influye la topología y el encaminamiento
- c. Solamente influye la topología
- d. La topología, el control de flujo y el encaminamiento

**Question 9**

Not yet answered

Marked out of 1.00

Flag question

La estructura de paralelismo "divide y vencerás" consiste en:

Select one:

- a. Distribuir diferentes partes de una tarea entre varios procesadores sin una relación de dependencia tipo master-slave
- b. Se divide una tarea en diferentes funciones que se aplican consecutivamente, de forma secuencial, sobre todos los datos
- c. "Divide y vencerás" no tiene nada que ver con el paralelismo
- d. Es la misma estrategia que "master-slave"

**Pregunta 1**No s'ha respond  
encara

Puntuat sobre 1,00

▼ Marca la  
pregunta

Indica cuál de las siguientes afirmaciones es correcta:

Trieu-ne una:

- a. Los bloqueos activos no se pueden solucionar
- b. La prevención de los interbloqueos es una estrategia pesimista
- c. Los bloqueos activos no se pueden dar nunca en las redes de interconexión directas
- d. La recuperación de los interbloqueos es una estrategia pesimista

Por descarte

**Pregunta 2**No s'ha respond  
encara

Puntuat sobre 1,00

▼ Marca la  
pregunta

¿Cuántos conmutadores tiene en la segunda columna de conmutadores de una red multi-etapa tipo Delta 25x16?

Trieu-ne una:

- a. 4
- b. 8
- c. 16
- d. 5

$$4^2 \rightarrow 16 \quad (2^{\text{a}} \text{ Columna} \rightarrow 16 \rightarrow \sqrt{16} = 4)$$

**Pregunta 3**

Resposta desada

Puntuat sobre 1,00

▼ Marca la  
pregunta

El paralelismo de datos:

Trieu-ne una:

- a. Se encuentra implícito en funciones lógicas
- b. es un tipo de paralelismo que nunca se encuentra de forma implícita.
- c. Se encuentra implícito en funciones aritméticas
- d. Se encuentra implícito en operaciones con imágenes

**Pregunta 4**No s'ha respond  
encara

Puntuat sobre 1,00

▼ Marca la  
pregunta

El diámetro de una malla abierta es:

Trieu-ne una:

- a.  $2*(r/2 - 1)$  siendo r la raíz cuadrada del número de nodos
- b.  $2*(r-1)$  siendo r la raíz cuadrada del número de nodos
- c.  $2*N/2$  siendo N el número de nodos
- d.  $2*(r-1)$  siendo r el número de nodos

$$2(r-1) \text{ donde } N = r^2 \rightarrow r = \sqrt{N}$$

**Pregunta 5**No s'ha respond  
encara

Puntuat sobre 1,00

▼ Marca la  
pregunta

En un multicomputador, ¿qué se utiliza para permitir una comunicación asíncrona entre cualesquiera de sus nodos?

Trieu-ne una:

- a. Una red directa o estática
- b. Es imposible establecer una comunicación asíncrona en un computador paralelo
- c. Buffers intermedios
- d. Enlaces con más ancho de banda

**Pregunta 6**

No s'ha respot encara

Puntuat sobre 1,00

Marca la pregunta

¿Cuál es el factor de aceleración, ganancia en velocidad o speed-up que puede conseguir como máximo si tiene 2 nodos conectados formando una máquina paralela y una aplicación cuyo 75% del código ha conseguido parallelizar? Suponga que el tiempo de sobrecarga no influye.

Trieu-ne una:

 a. 1.666 b. 4 c. 1.250 d. 2

P= nº nodos

$$\text{Ganancia} = Ts / Tp(p) = Ts / Ts * (\frac{1}{4} + \frac{3}{4} * \frac{1}{p}) = 4p / p+3$$

$$\text{Si } P=2 \rightarrow 8 / 5 = 1.6$$

**Pregunta 7**

No s'ha respot encara

Puntuat sobre 1,00

Marca la pregunta

La técnica de comunicación de reducción consiste en:

Trieu-ne una:

 a. Enviar un único elemento pero que es recibido por varios procesadores b. Enviar un único elemento a un único procesador c. Enviar varios elementos a la vez a varios procesadores d. A partir de varios elementos, utilizar una función para convertirlos en uno solo, que recibe un único procesador**Pregunta 8**

Resposta desada

Puntuat sobre 1,00

Marca la pregunta

En cierto momento del transcurso de la ejecución de una aplicación paralela en un multicomputador de 4 nodos, todos sus nodos (N0,N1,N2,N3) envían un dato distinto al nodo N0. Dicho nodo espera a que lleguen todos los datos y realiza la siguiente operación antes de continuar la ejecución de la aplicación:

$$\text{Resultado} = \text{dato\_N0} + \text{dato\_N1} + \text{dato\_N2} + \text{dato\_N3}$$

¿De qué tipo de comunicación estamos hablando?

Trieu-ne una:

 a. Reducción b. Difusión c. Dispersión d. Acumulación**Pregunta 9**

Resposta desada

Puntuat sobre 1,00

Marca la pregunta

Una red Omega es

Trieu-ne una:

 a. Una red directa b. Una red jerárquica c. Una red indirecta d. Una red basada en buses de contención**Pregunta 10**

Resposta desada

Puntuat sobre 1,00

Marca la pregunta

¿De qué tipo es la red de comunicación que ha empleado en el laboratorio para hacer las prácticas?

Trieu-ne una:

 a. MPP b. Red de computadores c. Cluster d. Cluster Beowulf

¿Cuál es la unidad de transferencia en la técnica de conmutación Worm-Hole?

Trieu-ne una:

- a. El mensaje
- b. El paquete
- c. El phit
- d. El flit

Diferencia entre los multiprocesadores y multicomputadores

**Solución:** Los multiprocesadores comparten el mismo espacio de memoria

Modo de programación SPMD

**Solución:** Un solo programa que puede actuar sobre varios flujos de datos.

De que depende el diseño de una red

**Solución:** De la topología del control de flujo y del Encaminamiento

Donde está implícito el paralelismo de datos.

**Solución:** En las operaciones con estructuras de datos

Diámetro de una red en árbol binario de 15 nodos

**Solución:** 6      ->     $2 * (k - 1)$  , k = niveles árbol, serian 4, por tanto  $2 * 3 = 6$

En que unidad se mide la red worm-hole

**Solución:** transferencia entre interfaces (unidad: paquete/mensaje) – Transf. Entre conmut. (unidad: flit)

Un multiprocesador de 64 procesadores utiliza una red Omega con conmutadores de grado 2. ¿Cuántos conmutadores utiliza en total?

Trieu-ne una:

- a. 60
- b. 192
- c. 64
- d. 32

Jorge A. b) En un pdf por internet he encontrado la siguiente fórmula

$(P/k) \log_k P$

$(64/2) * (\log 64 / \log 2) = 192$

¿Cuál es el máximo factor de aceleración, ganancia en velocidad o speed-up que puede conseguir como máximo si tiene 5 nodos conectados, el tiempo de sobrecarga no influye y dispone de una aplicación totalmente paralelizable?

Trieu-ne una:

- a. 10
- b. 3
- c. 5
- d. 2.5

Lo ideal en una aplicación totalmente paralela es lineal al número de procesadores, es decir, 5

$$\frac{T_s}{T_p(P)} = \frac{T_s}{\frac{T_s}{P}} = \frac{1}{\frac{1}{5}} = 5$$

¿Cuántas **etapas o columnas** de conmutadores tiene una red multi-etapa tipo Delta 25x16?

Trieu-ne una:

- a. 4
- b. 25
- c. 5
- d. 2

Pag103:  $5^2 * 4^2$ , 5\*4 conmutadores y 2 etapas...

La estructura de paralelismo "divide y vencerás" consiste en:

Trieu-ne una:

- a. Es la misma estrategia que "master-slave"
- b. *Distribuir diferentes partes de una tarea entre varios procesadores sin una relación de dependencia tipo master-slave*
- c. "Divide y vencerás" no tiene nada que ver con el paralelismo
- d. Se divide una tarea en diferentes funciones que se aplican consecutivamente, de forma secuencial, sobre todos los datos

La b) por descarte. A y c es seguro que NO, la d) creo que no

**Pregunta 1**No s'ha respot  
encara

Puntuat sobre 1,00

▼ Marca la  
pregunta

Indica cuál de las siguientes afirmaciones es correcta:

Trieu-ne una:

- a. Los bloqueos activos no se pueden solucionar
- b. La prevención de los interbloqueos es una estrategia pesimista
- c. Los bloqueos activos no se pueden dar nunca en las redes de interconexión directas
- d. La recuperación de los interbloqueos es una estrategia pesimista

**Pregunta 2**No s'ha respot  
encara

Puntuat sobre 1,00

▼ Marca la  
pregunta

¿Cuántos conmutadores tiene en la segunda columna de conmutadores de una red multi-etapa tipo Delta 25x16?

Trieu-ne una:

- a. 4
- b. 8
- c. 16
- d. 5

**Pregunta 3**

Resposta dessada

Puntuat sobre 1,00

▼ Marca la  
pregunta

El paralelismo de datos:

Trieu-ne una:

- a. Se encuentra implícito en funciones lógicas
- b. es un tipo de paralelismo que nunca se encuentra de forma implícita.
- c. Se encuentra implícito en funciones aritméticas
- d. Se encuentra implícito en operaciones con imágenes

**Pregunta 4**

No s'ha respot encara

Puntuat sobre 1,00

▼ Marca la pregunta

El diámetro de una malla abierta es:

Trieu-ne una:

- a.  $2*(r/2 - 1)$  siendo r la raíz cuadrada del número de nodos
- b.  $2*(r-1)$  siendo r la raíz cuadrada del número de nodos
- c.  $2*N/2$  siendo N el número de nodos
- d.  $2*(r-1)$  siendo r el número de nodos

**Pregunta 5**

No s'ha respot encara

Puntuat sobre 1,00

▼ Marca la pregunta

En un multicomputador, ¿qué se utiliza para permitir una comunicación asíncrona entre cualesquiera de sus nodos?

Trieu-ne una:

- a. Una red directa o estática
- b. Es imposible establecer una comunicación asíncrona en un computador paralelo
- c. Buffers intermedios
- d. Enlaces con más ancho de banda

**Pregunta 6**

No s'ha respost encara

Puntuat sobre 1,00

▼ Marca la pregunta

¿Cuál es el factor de aceleración, ganancia en velocidad o speed-up que puede conseguir como máximo si tiene 2 nodos conectados formando una máquina paralela y una aplicación cuyo 75% del código ha conseguido parallelizar? Suponga que el tiempo de sobrecarga no influye.

Trieu-ne una:

- a. 1.666
- b. 4
- c. 1.250
- d. 2

**Pregunta 7**

No s'ha respost encara

Puntuat sobre 1,00

▼ Marca la pregunta

La técnica de comunicación de reducción consiste en:

Trieu-ne una:

- a. Enviar un único elemento pero que es recibido por varios procesadores
- b. Enviar un único elemento a un único procesador
- c. Enviar varios elementos a la vez a varios procesadores
- d. A partir de varios elementos, utilizar una función para convertirlos en uno solo, que recibe un único procesador

**Pregunta 8**

Resposta desada

Puntuat sobre 1,00

Marca la pregunta

En cierto momento del transcurso de la ejecución de una aplicación paralela en un multicomputador de 4 nodos, todos sus nodos (N0,N1,N2,N3) envían un dato distinto al nodo N0. Dicho nodo espera a que lleguen todos los datos y realiza la siguiente operación antes de continuar la ejecución de la aplicación:

$$\text{Resultado} = \text{dato\_N0} + \text{dato\_N1} + \text{dato\_N2} + \text{Dato\_N3}$$

¿De qué tipo de comunicación estamos hablando?

Trieu-ne una:

- a. Reducción
- b. Difusión
- c. Dispersión
- d. Acumulación

**Pregunta 9**

Resposta desada

Puntuat sobre 1,00

Marca la pregunta

Una red Omega es

Trieu-ne una:

- a. Una red directa
- b. Una red jerárquica
- c. Una red indirecta
- d. Una red basada en buses de contención

**Pregunta 10**

Resposta desada

Puntuat sobre 1,00

 Marca la pregunta

¿De qué tipo es la red de comunicación que ha empleado en el laboratorio para hacer las prácticas?

Trieu-ne una:

- a. MPP
- b. Red de computadores
- c. Cluster
- d. Cluster Beowulf

¿Cuál de las siguientes opciones no es una ventaja del buffer de renombrado con acceso asociativo frente al acceso indexado?

Seleccione una:

- a. Permite varias escrituras pendientes a un mismo registro.
- b. No tiene ninguna ventaja.
- c. Tiene un elemento que indica si el valor es válido o no.
- d. La búsqueda de un registro se debe hacer comparando el registro con todas las entradas hasta encontrarlo.

El algoritmo de renombrado se utiliza en las arquitecturas superescalares para:

Seleccione una:

- a. Es un algoritmo que se utilizar para permitir la ejecución desordenada ya que después de la ejecución se encarga de reordenar las instrucciones.
- b. Para nada. En superescalares no se usa ese algoritmo.
- c. Poder ejecutar dos instrucciones al mismo tiempo
- d. Para evitar los efectos de los riesgos WAR y WAW

Una ventana de instrucciones alineada es:

Seleccione una:

- a. Una ventana de instrucciones que carga instrucciones cuando está completamente vacía
- b. Una ventana de instrucciones que carga instrucciones conforme se va vaciando (sin esperar a vaciarse del todo)
- c. Una ventana de instrucciones que tiene una linea de separación entre los códigos de instrucción
- d. Una ventana de instrucciones extraterrestre

En un esquema de predicción explícita, los bits de predicción:

Seleccione una:

- a. Se pueden almacenar de forma acoplada a una estructura existente (como la BTB) o de forma independiente en una estructura específica que almacene únicamente la historia de los saltos.
- b. Solo se pueden almacenar de forma independiente en una estructura específica que almacene únicamente la historia de los saltos.
- c. Si la predicción es explícita, no se utilizan bits de predicción
- d. Solo se pueden almacenar de forma acoplada a una estructura existente (como la BTB)

El almacenamiento implícito de la predicción de un salto consiste en:

Seleccione una:

- a. Escribir en una hoja de papel la predicción del salto
- b. No se almacena la predicción sino la dirección del salto en el buffer correspondiente y en función de este almacenamiento se decide la predicción.
- c. Almacenar la predicción en un buffer implícito
- d. Se almacena la predicción en una tabla junto a la dirección de salto.

Ante un salto incondicional se pueden utilizar los siguientes tipos de predicción:

Seleccione una:

- a. Predicción implícita y explícita
- b. La predicción implícita no se puede utilizar ya que es necesario tener información sobre el estado anterior del salto y en un salto incondicional esto no es posible
- c. La predicción explícita no se puede utilizar ya que es necesario tener información sobre el estado anterior del salto y en un salto incondicional esto no es posible
- d. La predicción no tiene sentido cuando el salto es incondicional

El acceso a memoria concurrente o tipo C es:

Seleccione una:

- a. Un tipo de acceso utilizado por arquitecturas vectoriales que permite acceder a posiciones de memoria en un mismo bloque de forma concurrente
- b. Un tipo de acceso utilizado por arquitecturas vectoriales que permite acceder a posiciones de memoria en diferentes bloques de forma simultánea
- c. Un tipo de acceso utilizado por arquitecturas vectoriales que permite acceder a posiciones de memoria diferentes en un mismo bloque de forma simultánea
- d. Un tipo de acceso utilizado por arquitecturas vectoriales que permite acceder a posiciones de memoria en diferentes bloques de forma concurrente

El paralelismo a nivel de instrucción o ILP es:

Seleccione una:

- a. Un tipo de paralelismo que incorpora un rendimiento muy bajo y por eso casi no se usa.
- b. Un paralelismo funcional que consiste en ejecutar varias instrucciones a la vez
- c. Es un tipo de paralelismo inventado. No existe.
- d. Un paralelismo de datos que consiste en trabajar con los datos de varias instrucciones a la vez

El buffer de reorden en una arquitectura superescalar permite:

Seleccione una:

- a. El buffer de reorden es una estructura que se utiliza para la predicción de saltos y no tiene nada que ver con el orden de ejecución de las instrucciones.
- b. El buffer de reorden no se utiliza en las arquitecturas superescalares.
- c. Completar (sacar del cauce) las instrucciones en un orden diferente al orden de programa.
- d. La ejecución de las instrucciones fuera de orden.

y la 8 es de benchmark, que cual era el mejor benchmark para evaluar no se se que

Decia que si la aplicación, puesto que es la manera real de saber si funciona  
depende de lo que quieras evaluar

kernet porque no se que  
y sintetico

**El buffer de reorden en una arquitectura superescalar permite:**

- Completar (sacar del cauce) las instrucciones en un orden diferente al orden de programa.(las saca en orden, pero vale para controlar los riesgos de WAW y WAR actualizando el registro, una vez que todas hayan acabado todas).
  - El buffer de reorden es una estructura que se utiliza para la predicción de saltos y no tiene nada que ver con el orden de ejecución de las instrucciones. (falsa, se usa en el renombrado ses.3)
  - La ejecución de las instrucciones fuera de orden.
  - El buffer de reorden no se utiliza en las arquitecturas superescalares. (falsa, si se usa)
- 

**El mejor tipo de benchmark para evaluar un sistema es:**

- Los kernels porque permiten evaluar aspectos concretos.
  - Los benchmarks sintéticos porque son independientes de las aplicaciones.
  - Las aplicaciones reales porque evalúan la realidad.
  - Depende de lo que se deseé evaluar.
- 

**Ante un salto incondicional se pueden utilizar los siguientes tipos de predicción:**

- La predicción no tiene sentido cuando el salto es incondicional.
  - La predicción explícita no se puede utilizar ya que es necesario tener información sobre el estado anterior del salto y en un salto incondicional esto no es posible
  - Predicción implícita y explícita
  - La predicción implícita no se puede utilizar ya que es necesario tener información sobre el estado anterior del salto y en un salto incondicional esto no es posible
- 

**Suponiendo que los ciclos de latencia de inicio para una máquina vectorial son los siguientes: Cargas = 7. Sumas = 3. Desplazamientos = 4. Y que queremos realizar una operación de carga, suma y desplazamiento con encadenamiento de cauce para un vector de 10 componentes, ¿Cuántos ciclos tardaríamos?**

-7+3+4+10. (Por descarte)

-7+3+4+5.

-7+3+4+10+3\*10.

-7+3+4+6+10.

---

### **En cuanto a los riesgos por dependencia de datos en las arquitecturas superescalares...**

-Los riesgos RAW (lectura después de escritura) son los únicos riesgos que no se pueden solucionar por renombrado.

-Los riesgos WAW y WAR son riesgos por dependencia de datos que se pueden solucionar por adelantamiento.(Se solucionan con renombrado --->FALSA)

-Los riesgos RAR son riesgos por dependencia de datos que se pueden solucionar utilizando renombrado.(Dice que se solucionan WAW y WAR---> FALSA)

-Los riesgos WAW y WAR son riesgos por dependencia de datos que **no** se pueden solucionar utilizando renombrado .(Si se pueden solucionar ---->FALSA)

---

### **El algoritmo de renombrado se utiliza en las arquitecturas superescalares para:**

-Para evitar los efectos de los riesgos WAR y WAW

-Para nada. En superescalares no se usa ese algoritmo.

-Es un algoritmo que se utilizar para permitir la ejecución desordenada ya que después de la ejecución se encarga de reordenar las instrucciones.

-Poder ejecutar dos instrucciones al mismo tiempo

---

### **El paralelismo a nivel de instrucción o ILP es:**

-Un tipo de paralelismo que incorpora un rendimiento muy bajo y por eso casi no se usa.

-Un paralelismo funcional que consiste en ejecutar varias instrucciones a la vez.

-Es un tipo de paralelismo inventado. No existe.

-Un paralelismo de datos que consiste en trabajar con los datos de varias instrucciones a la vez.

---

### **¿Cuál de las siguientes afirmaciones es correcta para la gestión de los riesgos de control?**

- El procesamiento especulativo de los saltos es la estrategia más común en los procesadores superescalares.
  - El bloqueo del procesamiento del salto es la estrategia más común en los procesadores superescalares.(no)
  - La gestión del salto retardado es la estrategia más común en los procesadores superescalares.
  - La gestión de múltiples caminos es la estrategia más común en los procesadores superescalares.
- 

### **Una ventana de instrucciones alineada es:**

- Una ventana de instrucciones que carga instrucciones conforme se va vaciando (sin esperar a vaciarse del todo).(falsa, es justo lo contrario --> no-alineada)
  - Una ventana de instrucciones que tiene una línea de separación entre los códigos de instrucción.(charla)
  - Una ventana de instrucciones que carga instrucciones cuando está completamente vacía.
  - Una ventana de instrucciones extraterrestre.(mm nop)
- 

### **¿Cuál de las siguientes afirmaciones es correcta?**

- Todas las afirmaciones son correctas.
  - Los campos de la BTB se actualizan después de ejecutar el salto.(vdd)
  - La tabla de historia de los saltos con bits desacoplados permite predecir instrucciones que no estén en la BTAC.(vdd)
  - La tabla de historia de los saltos con bits desacoplados aumenta el hardware necesario para gestionar los saltos.(vdd)
- 

### **¿Cuáles son las diferencias entre la BTAC y la BTIC?**

- La BTAC contiene las direcciones de destino de salto mientras que la BTIC contiene las instrucciones de destino del salto.
- Tanto la BTIC como la BTAC son cachés que permiten gestionar de forma adecuada los saltos aunque la BTIC es más rápida que la BTAC.
- En la BTIC las direcciones destino del salto se leen al mismo tiempo que se captan instrucciones de salto.

-La BTAC es una caché más rápida que la BTIC aunque necesita más hardware

---

### **Una arquitectura vectorial es:**

-Todas las respuestas son correctas

-Una arquitectura donde cada operación vectorial codifica gran cantidad de cálculo, reduciendo el número de instrucciones y evitando riesgos de control (vdd ses 1)

-Una arquitectura donde el cálculo de los componentes del vector se realiza de forma independiente obteniendo buenos rendimientos. (vdd ses 1)

-Una arquitectura orientada al procesamiento de vectores (suma de vectores, productos escalares, etc.).(vdd ses 1)

---

### **El acceso a memoria concurrente o tipo C es:**

-Un tipo de acceso utilizado por arquitecturas vectoriales que permite acceder a posiciones de memoria en un mismo bloque de forma concurrente.

-Un tipo de acceso utilizado por arquitecturas vectoriales que permite acceder a posiciones de memoria en diferentes bloques de forma simultánea.

-Un tipo de acceso utilizado por arquitecturas vectoriales que permite acceder a posiciones de memoria diferentes en un mismo bloque de forma simultánea.

-Un tipo de acceso utilizado por arquitecturas vectoriales que permite acceder a posiciones de memoria en diferentes bloques de forma concurrente.

---

### **¿Cuál de las siguientes opciones no es una ventaja del buffer de renombrado con acceso asociativo frente al acceso indexado?**

(asociativo : -permite varias escrituras pendientes a un mismo registro; -se utiliza el bit último para marcar cual ha sido la más reciente

indexado: -Solo permite una escritura pendiente a un mismo registro; -Se mantiene la escritura más reciente)

-no tiene ninguna ventaja.

-Permite varias escrituras pendientes a un mismo registro.(esto si que es una ventaja frente al indexado, falsa)

-Tiene un elemento que indica si el valor es valido o no.

-La búsqueda de un registro se debe hacer comparando el registro con todas las entradas hasta encontrarlo

---

### **El almacenamiento implícito de la predicción de un salto consiste en:**

- No se almacena la predicción sino la dirección del salto en el buffer correspondiente y en función de este almacenamiento se decide la predicción.
  - Se almacena la predicción en una tabla junto a la dirección de salto.
  - Almacenar la predicción en un buffer implícito(falsa)
  - Escribir en una hoja de papel la predicción del salto(falsa)
- 

### **Qué diferencias existen entre las estaciones de reserva y la ventana de instrucciones?**

- Las instrucciones se cargan en la ventana de instrucciones una vez descodificadas y se utiliza un bit para indicar si un operando está disponible. En la estación de reserva las instrucciones se cargan una vez finalizadas(??)
  - La ventana de instrucciones almacena las instrucciones pendientes de ejecutar y las estaciones de reserva las ejecutadas pero no finalizadas.(las ejecutadas están en la unidad de ejecución no en la estación de reserva)
  - Las instrucciones que se encuentran en la estación de reserva han sido enviadas a ejecución, mientras que las instrucciones que se encuentran en la ventana de instrucciones aún no han sido enviadas a ejecución (en la ventana de instrucciones esperan a tener los operandos, no es emitida hasta que no los tiene.)
  - Las instrucciones que se encuentran en la estación de reserva han sido emitidas mientras que las instrucciones que se encuentran en la ventana de instrucciones aún no han sido emitidas(esto es verdad,)
- 

### **En un esquema de predicción explícita, los bits de predicción:**

- Se pueden almacenar de forma acoplada a una estructura existente (como la BTB) o de forma independiente en una estructura específica que almacene únicamente la historia de los saltos.
- Solo se pueden almacenar de forma acoplada a una estructura existente (como la BTB)

-Solo se pueden almacenar de forma independiente en una estructura específica que almacene únicamente la historia de los saltos.

---

**Indica cual es la Incorrecta**

en el buffer de renombrado se utiliza un puntero(...)

**AIC**

## **Tema 2: Segmentación y superescalares**

**Alejandro Perez Martínez**

**Segmentación:** División de un proceso en un conjunto de etapas especializadas.

- **Ganancia:**

$$G = \frac{Ti}{k} = \frac{K \cdot n \cdot t}{kt + (n-1)t} = \frac{k \cdot n}{k + n - 1}$$

- **Productividad:** N° de operaciones por unidad de tiempo.

$$P = \frac{n}{TLI + (n-1)t} \quad TLI = k \cdot t$$

- **Eficiencia:** Relación entre ganancia de velocidad que proporciona el cauce y el n° de etapas del mismo

$$E_k = \frac{G \cdot k}{k} = \frac{n \cdot T}{k(TLI + (n-1)t)}$$

- **Ganancia en el caso real:**

$$G_k = \frac{T \sin\_segmentar}{T_{segmentado}} = \frac{n \cdot T}{TLI + (n-1)t} = \frac{n \cdot T}{(k \cdot t) + (n-1)t}$$

## **RIESGOS**

- **Riesgo de datos:** Se producen por dependencia entre operandos y resultados de instrucciones distintas.
- **Riesgos de control:** Se originan a partir de instrucciones de salto condicional que, según su resultado determinan la secuencia de instrucciones que hay que procesar tras ellos.
- **Riesgo de colisión o estructurales:** Se producen cuando instrucciones diferentes necesitan el mismo recurso al mismo tiempo.

## **TIPOS DE RIESGOS DE DATOS**

- **RAW (Read After Write).**
- **WAR (Write After Read).**
- **WAW (Write After Write).**

## **SOLUCIONES A LOS RIESGOS DE DATOS (TIPO RAW)**

- **Reorganización de código**
  - Intercambio de instrucciones.
  - Introducción de instrucciones NOP (reduce prestaciones del cauce)
- **Interbloqueo de etapas**
  - Se introducen elementos Hardware en el cauce que detectan la existencia de dependencias y detienen el cauce el n° de ciclos necesarios.
- **Atajos o bypass.** Anticipa el dato a la etapa que lo necesita antes de ser escrito en el banco de registro.

### **RIESGOS DE CONTROL (Instrucciones de salto)**

- Abortar operaciones.
- Bloqueos de cauce o uso del NOP
- Salto retardado: El computador reorganiza el código introduciendo instrucciones anteriores a la del salto después de ella (debe comprobar que el programa se ejecuta correctamente).

### **RIESGOS ESTRUCTURALES (INSTRUCCIONES DE ACCESO A MEMORIA)**

- Modificaciones Hardware en la memoria.
- Introducción de NOP.
- Modificaciones en el cauce: introducción de una etapa.

### **DIMENSIONES DE DISEÑO DE UN PROCESADOR SEGMENTADO**

- Organización del cauce
  - Número de etapas (A más etapas, mayor puede ser el cauce).
  - Subtareas que implementa cada etapa.
  - Distribución de las secuencias de etapas.
  - Uso de caminos de bypass. Es posible adelantar los datos necesarios entre etapas para que las dependencias no tengan efecto.
  - Temporización del cauce:
    - Síncrono: Señal de reloj común.
    - Asíncrono: Líneas de sincronización entre etapas consecutivas.

### **RESOLUCIÓN DE DEPENDENCIAS**

- Estática (compilador)
- Dinámica (elementos Hardware).
- Combinada.

La utilización de varias unidades funcionales permitirá la ejecución de instrucciones fuera de orden.

Un procesador superescalar es un procesador segmentado que puede iniciar varias instrucciones (que se ejecutarán simultáneamente) y terminar varias instrucciones por ciclo.

### **FASE DEL PROCESAMIENTO SUPERESCALAR**

- Captación de instrucciones (IF): Se llevan las instrucciones de caché al procesador.
- Decodificación de Instrucciones (ID): Se identifica qué operación hay y con qué operandos.
- Emisión de instrucciones (ISS): Las instrucciones esperan hasta que las unidades funcionales donde se van a ejecutar están libres y los operandos que necesitan están disponibles.
- Ejecución de instrucciones (EX): Se realizan las operaciones codificadas en las instrucciones. Termina con la ejecución de los resultados a los registros intermedios.
- Escritura (WB): Escritura de los resultados en los registros de la arquitectura.

## **CARACTERÍSTICAS DEL PROCESAMIENTO SUPERESCALAR**

- En una secuencia de instrucciones se pueden distinguir tres tipos de ordenación:
  - El orden en el que se captan las instrucciones (orden en el código).
  - Orden en el que se ejecutan (orden de emisión).
  - Orden en el que cambian los registros y la memoria (orden de finalización).
- El procesador debe ser capaz de identificar el paralelismo entre instrucciones (ILP) y organizar la captación, decodificación y ejecución en paralelo.

## **ORDEN DE EMISIÓN Y FINALIZACIÓN**

- **Emisión ordenada con finalización ordenada.**
- **Emisión desordenada con finalización ordenada.**
- **Emisión ordenada con finalización desordenada.**
- **Emisión desordenada con finalización desordenada.**

## **VENTANA DE INSTRUCCIONES**

- Almacena las instrucciones pendientes (todas si la ventana es centralizada o las de un tipo determinado si es distribuida).
- Las instrucciones se cargan en la ventana de una vez decodificados y se utiliza un bit para indicar si un operando está disponible (se almacena el valor o se indica el registro desde donde se lee) o no (se almacena la unidad funcional desde donde llegará el operando).
- Una instrucción puede ser emitida cuando tiene todos sus operandos disponibles y la unidad funcional donde se procesará. Hay diversas posibilidades para el caso en el que varias instrucciones estén disponibles.

## **INTERRUPCIONES Y OPTIMIZACIÓN**

**Interrupciones:** Afectan negativamente al rendimiento del procesador al romper el flujo de instrucciones.

## **CLASIFICACIÓN**

	<b>Indican una condición de error y dan paso a una rutina de recuperación</b>	<b>Interrupciones críticas para comunicación con el sistema operativo</b>
<b>Fuente interna a la CPU</b>	<b>EXCEPCIONES(A)</b> <ul style="list-style-type: none"><li>• Overflow</li><li>• Anomalía en coma flotante</li></ul>	<b>MEMORIA VIRTUAL.</b> <b>INSTRUCCIONES NO IMPLEMENTADA (B)</b> <ul style="list-style-type: none"><li>• Breakpoint</li><li>• Llamada al S.O.</li><li>• Fallo página</li></ul>
<b>Fuente externa</b>	<b>FALLO DE HARDWARE (C)</b> <ul style="list-style-type: none"><li>• Error en el Hardware</li></ul>	<b>TEMPORIZACIÓN E/S (D)</b> <ul style="list-style-type: none"><li>• Petición de dispositivos E/S</li></ul>

## **POSIBILIDADES**

- **Excepciones síncronas/asíncronas:** Si el evento se produce en el mismo lugar cada vez que se ejecutan el programa es síncrono. Los asíncronos se originan en los dispositivos externos al procesador y la memoria. Se pueden atender después de que se completa la instrucción a curso.
- **Solicitadas por el usuario o sobrevenidas:** Si son solicitadas se tratan como excepciones (aunque no lo son) ya que utilizan el mismo mecanismo para almacenar y restaurar el estado. Las sobrevenidas se deben a un evento que no controla el programa.
- **Enmascarables por el usuario o no enmascarable.**
- **Dentro de una instrucción o entre instrucciones:** Según el evento impide que se termine la ejecución de la instrucción porque ocurre en mitad de la misma o bien ocurre entre instrucciones.
- **Excepciones para terminar o con continuación:** Catastróficas o permiten la continuación.

## **OPTIMIZACIÓN EN LA IMPLEMENTACIÓN DE LAS INSTRUCCIONES**

- **Precisa:** Se atiende primera a la que llega a WB (no ordenadamente) 
- **Imprecisa:** Cuando se produce se atiende. 
- **Externas-Criticas (D):** Una vez se detecta la interrupción, se deja de captar instrucciones, se terminan las instrucciones del cauce a partir de la instrucción a la que apuntaba el PC y que no se aceptó.
- **Externas-error (C):** Como el caso anterior o congelar el estado del procesador en el momento de la interrupción-error para que la rutina de gestión para analizarlo.
- **Internas-error (A):** Se suelen implementar de forma imprecisa, ya que el programa interrumpido no suele poder continuar. En el caso del debugger, si deben implementarse como precisas.
- **Internas-críticas (B):** Se deben implementar como precisas y de forma eficaz puesto que son necesarias para el funcionamiento correcto de todos los programas.

## **ANÁLISIS CAUCE SEGMENTADO**

1. Construir tabla de reserva
2. Determinar la lista de latencias prohibidas
3. Construir el vector de colisiones
4. Construir el grafo de estados

## **OPTIMIZACIÓN**

### **1. Duplicación de etapas:**

- a. El nº máximo de X en una etapa coincide con la MLM teórica.
  - b. Si MLM teorica < MLM real podemos disminuir la MLM teórica eliminando X de la tabla de reserva → duplicación de etapas.
2. Se puede demostrar que si el CLP existen latencias que son múltiplos de la MLM teórica → MLMreal > MLMTeórica
    - a. Insertar etapas sin cálculo
    - b. Disminuir la latencia que es múltiplo.

## **SCOREBOARDING**

Evita los riesgos RAW pero en WAW y WAR produce detenciones del cauce.

## **RENOMBRAMIENTO**

Evita todos los tipos de riesgos por dependencias de datos.

## **RENOMBRAMIENTO DE REGISTROS**

- Técnica para evitar el efecto de dependencias WAR, o Antidependencias (en la emisión desordenada) y WAW, o dependencias de salida (ejecución desordenada)
- Implementación Estática: Durante la compilación.
- Implementación Dinámica: Durante la ejecución (circuitería adicional y registros extras).

## **CARACTERÍSTICAS DE LOS BUFFERS DE RENOMBRADO**

- Tipos de buffers (separados o mezclados con los registros de la arquitectura)
- Número de buffers de renombrado.
- Mecanismos para acceder a los Buffers (asociativos o indexados) 
- Velocidad del renombrado: Máximo número de nombres asignados por ciclo que admite el procesador.

<b>Consistencia del procesador.</b> Consistencia en el orden en que se completan las instrucciones	<b>Débil:</b> Las instrucciones se pueden completar desordenadamente siempre que no se vean afectadas las dependencias.	Deben detectarse y resolverse las dependencias
	<b>Fuerte:</b> Las instrucciones deben completarse estrictamente en el orden en que están en el programa.	Se consigue mediante el uso del ROB
<b>Consistencia de memoria.</b> Consistencia del orden de los accesos a memoria.	<b>Débil:</b> Los accesos a memoria (load/store) pueden realizarse desordenadamente siempre que no afecten a las dependencias.	Deben detenerse y resolverse las dependencias de acceso a memoria
	<b>Fuerte:</b> Los accesos a memoria deben realizarse estrictamente en el orden en que están en el programa.	Se consigue mediante el uso del ROB.

## CONSISTENCIA DEL PROCESADOR

- El puntero de cabecera apunta a la siguiente posición libre y el puntero de cola a la siguiente instrucción a retirar.
- Las instrucciones se introducen en el ROB en orden del programa estricto y pueden estar marcadas como emitidas (issue, i), en ejecución (x), o finalizada su ejecución (f).
- Las instrucciones sólo se pueden retirar (se produce la finalización con la escritura en los registros de la arquitectura) si han finalizado, y todas las que les proceden también.
- La consistencia se mantiene porque sólo las instrucciones que se retiran del ROB se completan (escribe en los registros de la arquitectura) y se retiran en el orden estricto del programa.

## CONSISTENCIA DE LA MEMORIA

- La consistencia de memoria débil (reordenación de los accesos a memoria).

## BYPASS DE LOADS/STORES

- Los loads pueden adelantarse a los stores pendientes y viceversa (siempre que no se violen dependencias).
- Hacen posibles loads y stores especulativos.
- Cuando un load se adelanta a un store que la precede antes de que se haya determinado la dirección se habla de load especulativos. Igual para el store que se adelanta a un load o un store.
- Permite ocultar las Faltas de Caché.
  - Si se adelanta un acceso a memoria a otro que dio lugar a una falta de caché y accede a Memoria Principal.

## CLASIFICACIÓN DE LOS SALTOS

<b>INCONDICIONALES</b>	Salto incondicional (BR)
	Llamada a subrutina (Salta y salva el PC) (BSR)
	Retorno de subrutina (retorna el PC salvado)
<b>CONDICIONALES</b>	Condición de bucle (salta si se cumple, si no NOP) (BLT)
	Otro salto condicional (salta si se cumple, si no NOP) (BNE)

## GESTIÓN DE SALTOS CONDICIONALES NO RESUELTOS

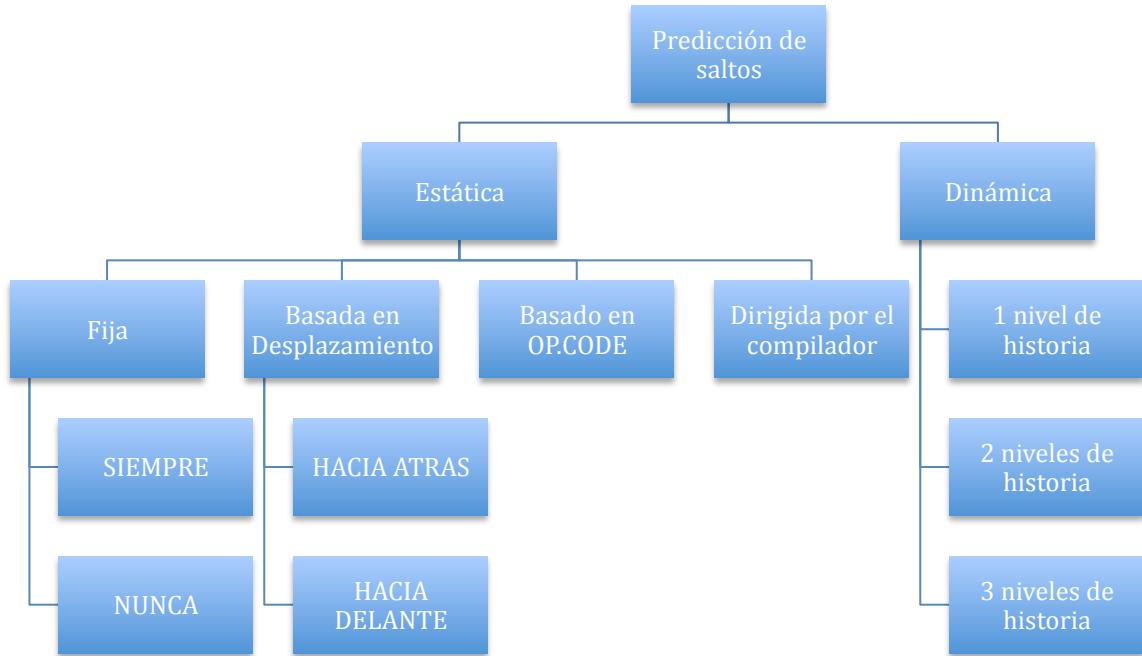
Uso de los ciclos que siguen a la instrucción de salto condicional	<b><u>Salto retardado</u></b>	Se utilizan los ciclos que siguen a la captación de una instrucción de salto para insertar instrucciones que deben ejecutarse independientemente del resultado del salto.
Gestión de saltos condicionales no resueltos (una condición de salto no se puede comprobar si no se ha terminado de evaluar)	<b><u>Bloqueo del procesamiento de salto</u></b>	Se bloquea la instrucción de salto hasta que la condición esté disponible
	<b><u>Procesamiento especulativo de los saltos</u></b>	La ejecución persigue por el camino más posible si se ha errado en la predicción hay que recuperar el camino correcto
	<b><u>Múltiples caminos</u></b>	Se ejecutan los dos caminos posibles después de un salto hasta que la condición de salto se evalúan. En ese momento se cancela el camino incorrecto.
Evitar saltos condicionales	<b><u>Ejecución vigilada</u></b>	Se evitan los saltos condicionales incluyendo en la arquitectura instrucciones con operaciones condicionales

## PREDICCIÓN DE SALTO

- **Predicción fija:** Se toma siempre la misma dirección: 'taken' o 'not taken'.

## PREDICCIÓN VERDADERA

- **Estática:** Según los atributos de la instrucción de salto (el código de operación, el desplazamiento, la decisión del compilador).
- **Dinámica:** Según el resultado de ejecuciones pasadas de las instrucciones (historia de la instrucción de salto).



## EJERCICIOS TEMA 2 AIC

- 1) Para el fragmento de código siguiente y suponiendo que tenemos 1 unidad funcional de carga que tarda 2 ciclos, 1 unidad funcional de almacenamiento que tarda 2 ciclos, 2 unidades funcionales de Suma/Resta que tardan 1 ciclo cada una y 1 unidad funcional de multiplicación que tarda 5 ciclos, se pide indicar el orden de la emisión de las instrucciones para los siguientes casos:
- Ventana centralizada. Emision ordenada y alineada
  - Ventana centralizada. Emision desordenada y alineada

Suponemos que se captan y se emiten 2 instrucciones por ciclo.

a)

	<b>IF</b>	<b>ID/ISS</b>	<b>EX</b>
<b>Lw r1, 0xclf</b>	1	2	3-4
<b>Lw r2, 0x10c</b>	1	2	5-6
<b>Add r2, r2, #7</b>	2	5	7
<b>Add r3, r1, r1</b>	2	5	7
<b>Lw r1 , 0x10c</b>	3	8	9-10
<b>Mult r4 ,r2 ,r3</b>	3	8	11-15
<b>Sw (r4), r3</b>	4	12	16-17

b)

	<b>IF</b>	<b>ID/ISS</b>	<b>EX</b>
<b>Lw r1, 0xclf</b>	1	2	3-4
<b>Lw r2, 0x10c</b>	1	2	5-6
<b>Add r2, r2, #7</b>	2	5	7
<b>Add r3, r1, r1</b>	2	5	6
<b>Lw r1 , 0x10c</b>	3	8	9-10
<b>Mult r4 ,r2 ,r3</b>	3	8	9-13
<b>Sw (r4), r3</b>	4	10	14-15

- 2) En un programa, una instrucción de salto condicional tiene la siguiente estructura:

S N N N S S S N N N S N S N S N

S = Se toma el salto

N = No se toma el salto

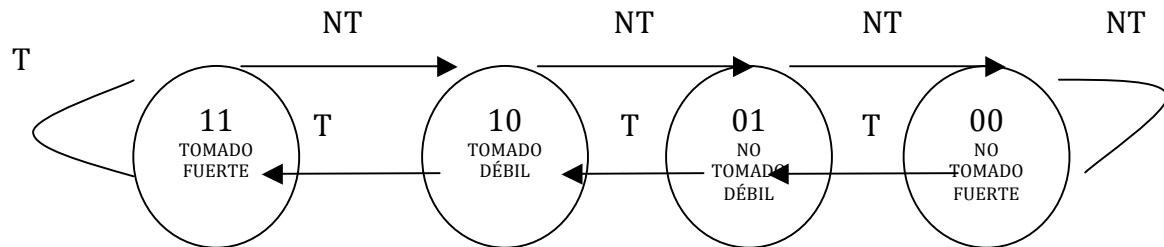
Indicar el número de penalizaciones que obtendremos si se produce:

- Predicción fija (Always not token)
- Predicción estática (si negativo se toma, si positivo no)
- Predicción dinámica con 2 bits inicializado en 11
- Predicción dinámica con 2 bits inicializado en 111

Solución:

- Existen 9 saltos no tomados y 7 saltos tomados, por lo tanto, como la predicción es NOT Token, penalizaremos los S, así pues tenemos 7 Penalizaciones
- El salto siempre es hacia atrás, por lo tanto se prevé que siempre saltará. Como no salta en 9 ocasiones, la penalización es de 9.

c)



	<b>S</b>	<b>N</b>	<b>N</b>	<b>N</b>	<b>S</b>	<b>S</b>	<b>S</b>	<b>N</b>	<b>N</b>	<b>N</b>	<b>S</b>	<b>N</b>	<b>S</b>	<b>N</b>	<b>S</b>	<b>N</b>
<b>Pred.</b>	S	S	S	N	N	N	S	S	N	N	N	N	N	N	N	N
<b>Penaliz.</b>		P	P		P	P		P	P		P		P		P	
<b>Estado Siguiiente</b>	11	10	01	00	01	10	11	10	01	00	01	00	01	00	01	00

11

9 Penalizaciones

d)

Estado	Salto	Predicción	Penalización
<b>111</b>	S	S	-
<b>111</b>	N	S	P
<b>011</b>	N	S	P
<b>001</b>	N	N	-
<b>000</b>	S	N	P
<b>100</b>	S	N	P
<b>110</b>	S	S	-
<b>111</b>	N	S	P
<b>011</b>	N	S	P
<b>001</b>	N	N	-
<b>000</b>	S	N	P
<b>100</b>	N	N	-
<b>010</b>	S	N	P
<b>001</b>	N	N	-
<b>000</b>	S	N	P
<b>100</b>	N	N	-
<b>010</b>			

## 9 PENALIZACIONES

# AIC

## Tema 1: Introducción a las arquitecturas

Álex Pérez

**Arquitectura:** Abarca las abstracciones que proporcionan las interfaces entre hardware/software y usuario/sistema, y la estructura y organización que permiten implementar dichas abstracciones, proporcionando prestaciones y costes razonables en la ejecución de los programas de computador.

### **Mejoras en la arquitectura (2 mejoras, paralelismo y localidad)**

- **Paralelismo:**
  - **Segmentación de cauce:** Cada elemento se divide en una serie de etapas por las que van pasando los operandos, instrucciones, etc,... al ser procesados. El funcionamiento independiente de cada etapa permite que se puedan ejecutar distintas operaciones, instrucciones... concurrentemente.
  - **Repetición de elementos.**
- **Localidad:** Acerca datos e instrucciones al lugar donde se necesitan porque el acceso a los mismos son los más rápido posible (jerarquía de memoria).

### **Clasificación de las arquitecturas**

	<b><u>Paralelismo Funcional</u></b>	<b><u>Paralelismo de Datos</u></b>
<b>SISD( 1 Flujo de instrucciones + 1 Flujo de Datos)</b>	<ul style="list-style-type: none"> <li>• Procesadores segmentados</li> <li>• Procesadores superescalares</li> <li>• Procesadores VLIW</li> </ul>	
<b>SIMD( 1 Flujo de instrucciones + Múltiples Flujo de Datos)</b>		<ul style="list-style-type: none"> <li>• Procesadores vectoriales</li> <li>• Procesadores Matriciales</li> <li>• Procesadores Sistólicos</li> </ul>
<b>MIMD (Múltiples Instrucciones + Múltiples Flujo de Datos)</b>	<ul style="list-style-type: none"> <li>• <b>Memoria compartida:</b></li> <li>• Acceso uniforme (SMPs)</li> <li>• Acceso no uniforme (NUMA)</li> <li>• <b>Paso de mensajes:</b></li> <li>• Multicomputadores</li> </ul>	

### Tipos de paralelismo

- **Paralelismo de Datos:** La misma función, instrucción se ejecuta en paralelo pero en cada una de esas ejecuciones se aplica sobre un conjunto de datos distintos.
- **Paralelismo funcional:** varias funciones, tareas, instrucciones se ejecutan en paralelo.
  - **Nivel de instrucción (ILP):** Se ejecutan en paralelo las instrucciones de un programa (Segmentación).
  - **Nivel de bucle o hebra (thread):** Se ejecutan en paralelo distintas iteraciones de un bucle o secuencias de instrucciones de un programa.
  - **Nivel de procedimiento (proceso):** Los distintos procedimientos que constituyen un programa se ejecutan simultáneamente.
  - **Nivel de programa:** La plataforma ejecuta en paralelo programas diferentes que pueden corresponder, o no, a una misma aplicación.

**Tiempo de ejecución:** Tiempo necesario para completar la ejecución de un programa.

- **Tiempo CPU usuario:** Tiempo de la CPU para ejecutar un programa
- **Tiempo CPU sistema:** Tiempo de la CPU para ejecutar las tareas del SO para el programa.
- **Esperas:** E/S o ejecución de otros programas.

$$T_{CPU} = \text{Ciclos del programa} \times T_{CICLO} = \frac{\text{Ciclos}}{\text{Frecuencia}}$$

$$\text{Ciclos x instrucción (CPI)} = \frac{\text{Ciclos}}{N^{\circ} \text{ Instrucciones}(NI)}$$

$$T_{CPU} = NI \times CPI \times T_{CICLO}$$

Para procesadores que pueden emitir varias instrucciones x ciclo

$$CPI = CPE \times IPE$$

CPE = Ciclos entre inicio de emisión de instrucción.

IPE = Instrucciones que pueden emitirse.

Para procesadores que pueden codificar varias operaciones en una instrucción

$$NI = N^{\circ} \text{ operaciones que realiza / operaciones que puede codificar}$$

### RENDIMIENTO

- **Ganancia de velocidad:** incremento de la velocidad que se consigue gracias al aumento de recursos de una máquina respecto a la máquina base.

$$G = \frac{T_i}{T_p}$$

- **Eficiencia:** se obtiene dividiendo la ganancia entre el número de recursos (permite evaluar en qué medida las prestaciones de un programa se acercan al máximo)

$$E = \frac{Ganancia}{N^{\circ} Procesadores}$$

- **Productividad:** número de operaciones que se ejecutan por unidad de tiempo.

### Ley de Amdhal

$$A = \frac{1}{(1 - Fm) + \frac{Fm}{Am}} \quad A = Fa + ((1 - Fm) + \frac{Fm}{Am})$$

$$G = \frac{n}{1 + (n - 1)Fnm}$$

### **EJERCICIOS**

Dado un sistema superescalar de 5 cauces no segmentados ¿Cuál es el tiempo de ejecución de un programa de 1000 instrucciones, donde el tiempo del sistema es  $\frac{1}{2}$  del tiempo de la CPU y el tiempo de E/S es de  $\frac{1}{5}$  del tiempo de la CPU segundo?

$$\begin{aligned} TEjecucion &= T(E/S) + T(Sistema) + T(CPU) \Rightarrow \\ TEjec &= 1/5 T(CPU) + 1/2T(CPU) + T(CPU) \Rightarrow (1/5 + \frac{1}{2} + 1)T(CPU) \Rightarrow 17/10 \\ T(CPU) &\Rightarrow \end{aligned}$$

$$T(CPU) = NI \times CPI \times Tciclo$$

$$NI = 1000$$

$$CPI = CPE/IPE = CPE/5 \text{ (pq tiene 5 cauces)}$$

$$TEjec = 17/10 T(CPU) = 17/10 * (1000 * (CPE/5) * Tciclo) = 340 * CPE * TCiclo$$

Apartado b) Ganancia si se segmentara el cauce en 10 etapas

Como hemos segmentado, CPE pasa a ser  $CPE/K$ , siendo  $K$  el número de etapas en las que hemos segmentado. Por tanto:

$$\begin{aligned} TEjec &= 17/10 T(CPU) = 17/10 * (1000 * ((CPE/5)/10) * Tciclo) = 34 * CPE * TCiclo \\ \text{Ganancia} &= Tsin \text{ segmentar} / Tsegmentado = (340 * CPE * TCiclo) / (34 * CPE * TCiclo) \\ &= 10 \end{aligned}$$

Hemos conseguido una ganancia de 10 respecto al no segmentado

Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800  
addi r1,r1,#8  
addi r2,r2,#8  
ld f0,coef  
ld f2,-8(r1)  
ld f4,0(r1)  
muld f8,f2,f0  
oddd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (4c), 1 FP add (2c), 3 ALU int (1c) y 1 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción **addi r2,r2,#8?**

5 ✓

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción **ld f4,0(r1)?**

11 ✗

¿Cuántos ciclos tarda el código?

22 ✗

Escribir comentario o corregir la calificación

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 6 núcleos de un cierto tipo **T1** y 6 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 76% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 6 cores of a certain **T1** type and 6 cores of another **T2** type. Suppose that **T1** cores are 76% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

Respuesta:

- b) ¿Qué fracción del problema total debe ejecutar cualquiera de los núcleos más rápidos?

[ENGLISH]

- b) What fraction of the total problem should any of the faster cores run?

Le paso los enunciados que le tocaron (son todos distintos) y las soluciones para que las trabaje.

Para ver qué está mal necesitamos que se nos explique qué es cada cosa, qué está haciendo con cada expresión matemática y porqué. De otra forma, es imposible seguir el hilo de unas "cuentas" que parecen aleatorias. Le vuelvo a recomendar que estudie bien los contenidos y que vuelva a realizar los ejercicios desde cero, explicando bien cada paso. Seguramente llegará al resultado correcto, de otra forma me los podrá enviar más adelante y veremos dónde está el problema.

P5: Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo T1 y 4 núcleos de otro tipo T2. Suponga que los núcleos de tipo T1 son un 38% más rápidos que los de tipo T2. Suponga también que tiene una aplicación totalmente parallelizable en OpenMP y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

a) Calcule el valor de la ganancia paralela en velocidad (speed-up) usando al menos 2 decimales. Respuesta correcta 3.45

P6

b) ¿Qué fracción del problema total debe ejecutar cualquiera de los núcleos más rápidos? Respuesta correcta 0.29

P7

Dada una red multietapa tipo Delta con 797 entradas y 10038 salidas, se pide:

¿Cuántos conmutadores hay en la etapa 3 de la implementación más económica de esta topología?

Nota: Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

Respuesta correcta: 1372 (topología: 4\*\*5x7\*\*5)

Nota: Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 2 núcleos de un cierto tipo **T1** y 2 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 75% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando al menos 2 decimales.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 2 cores of a certain **T1** type and 2 cores of another **T2** type. Suppose that **T1** cores are 75% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using at least 2 decimals.

Respuesta:

- b) ¿Qué fracción del problema total debe ejecutar cualquiera de los núcleos más rápidos?

[ENGLISH]

- b) What fraction of the total problem should any of the faster cores run?

Respuesta:

Un computador SMP de 4 procesadores utiliza el protocolo MESI para controlar la coherencia de datos en las cachés. Las líneas de caché y los bloques son de 2 palabras (pueden guardar el contenido de 2 posiciones de memoria). La política de reemplazo de caché es LRU (least recently used), el tamaño de caché de cada procesador es de 2 líneas.

El contenido de las cachés y la memoria principal es el siguiente:

[ENGLISH] A 4-processor SMP computer uses the MESI protocol to control data consistency in the caches. The cache lines and blocks are 2 words long (they can hold the contents of 2 memory locations). The cache replacement policy is LRU (least recently used), the cache size of each processor is 2 lines.

The content of caches and main memory is as follows:

Cachés

	P1			P2			P3			P4		
línea	Dir	Dato	Estado									
0	0	0	S	6	5	M	6	6	I	4	9	E
	1	-6		7	6		7	4		5	10	
1	2	2	M	0	0	S	2	4	I	0	0	S
	3	3		1	-6		3	7		1	-6	

Memoria

Dirección	0	1	2	3	4	5	6	7	8	9
Dato	0	-6	4	7	9	10	-4	3	2	3

En un momento dado, los procesadores ejecutan las siguientes referencias: P2:WRITE@1(8), P3:WRITE@7(27), P4:WRITE@4(9)

Nota: PJ:WRITE@M(N) significa: el procesador J escribe el valor N en la posición M, y PJ:READ@M significa: el procesador J lee de la dirección M.

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

[ENGLISH] At any given time, the processors run the following references: P2:WRITE@1(8), P3:WRITE@7(27), P4:WRITE@4(9)

Note: PJ:WRITE@M(N) means: the J-processor writes the value N to the M-position, and PJ:READ@M means: the J-processor reads from the M-address.

Specify which of the following matches the state of the caches and/or main memory at the end of this sequence.

Seleccione una:

- a. Todos líneas de todas las cachés son inválidas. [ENGLISH] All lines in all caches are invalid.
- b. Línea 0 en P3 estado I; Línea 1 en P1 en estado S.[ENGLISH] Line 0 in P3 state I; Line 1 in P1 in state S
- c. Línea 0 en P3 estado M; Línea 0 en P2 en estado I. [ENGLISH] Line 0 in P3 state M; Line 0 in P2 in state I.
- d. Línea 0 en P3 estado M; Línea 0 en P1 en estado S. [ENGLISH] Linea 0 en P3 estado M; Línea 0 en P1 en estado S.

Pregunta 7

Sin responder  
aún

Puntúa como  
3,00

Markar  
pregunta

Sea un computador superescalar capaz de decodificar 4 inst/c, emitir 4 inst/c, escribir 4 inst/c en los registros correspondientes y retirar 4 inst/c. Dispone de buffer de reorden con número de orden de emisión. La emisión puede ser desordenada pero **no** contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 3 FP mul/div (2c), 3 FP add (2c), 3 ALU int (2c) y 3 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción **addi r2,r2,88**?

6

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción **ld f4,0(r1)**?

11

¿Cuántos ciclos tarda el código?

22

Sea un computador superescalar capaz de decodificar 4inst/c, emitir 4inst/c, escribir 4inst/c en los registros correspondientes y retirar 4 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800  
addi r1,r1,#8  
addi r2,r2,#8  
ld f0,coef  
ld f2,-8(r1)  
ld f4,0(r1)  
muld f8,f2,f0  
addd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (4c), 1 FP add (2c), 1 ALU int (1c) y 1 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?

¿Cuántos ciclos tarda el código?

En una máquina superescalar con buffer de reorden....

- a. En el ROB se almacena una copia exacta del banco de registros.
- b. En el ROB se renombran varias veces los registros del banco de registros.
- c. En el ROB se almacenan de forma temporal los valores que volverán a pasar a la etapa EX pero que no se almacenarán en la etapa WB
- d. En el ROB se almacenan de forma temporal los valores que finalmente se guardarán en el banco de registros.

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo **T1** y 4 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 43% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 4 cores of a certain **T1** type and 4 cores of another **T2** type. Suppose that **T1** cores are 43% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

Respuesta:

- b) ¿Qué fracción del problema total debe ejecutar cualquiera de los núcleos más rápidos?

[ENGLISH]

- b) What fraction of the total problem should any of the faster cores run?

Respuesta:

Pregunta 2

Sin responder  
aún

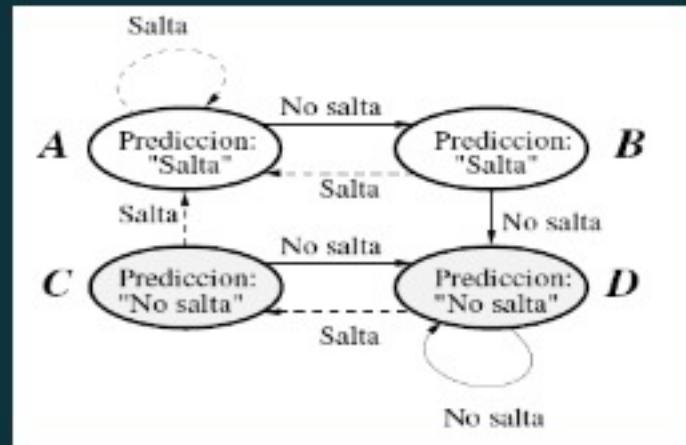
Puntúa como  
0,50

¶ Marcar  
pregunta

En una máquina superescalar con circuito de adelantamiento y con emisión no alineada...

- a. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones siempre que exista hueco.
- b. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones siempre en el siguiente ciclo de la búsqueda.
- c. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones si previamente se han ejecutado todas las instrucciones que pasaron en ciclos anteriores a la ventana y exista hueco.
- d. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones si previamente se han ejecutado todas las instrucciones que pasaron en ciclos anteriores a la ventana.

Un máquina superescalar dispone de un predictor de saltos dinámico que utiliza BTB de 4 entradas y 2 bits de predicción como el que se muestra en la figura.



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de r1=0:

```
0x000 addi r3,r1,#80  
0x001 addi r1,r1,#8  
0x002 addi r2,r2,#8  
0x003 addi r5,r1,#3  
0x004 beqz r5, 0x007  
0x005 addi r6,r5,#1  
0x006 bnez r6, 0x001  
0x007 subd f2,f1, f3
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x007	D
0x006	0x001	D

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración en el código?

00

¿Cuáles será los bits de predicción de la segunda fila de la BTB en ese caso?

01

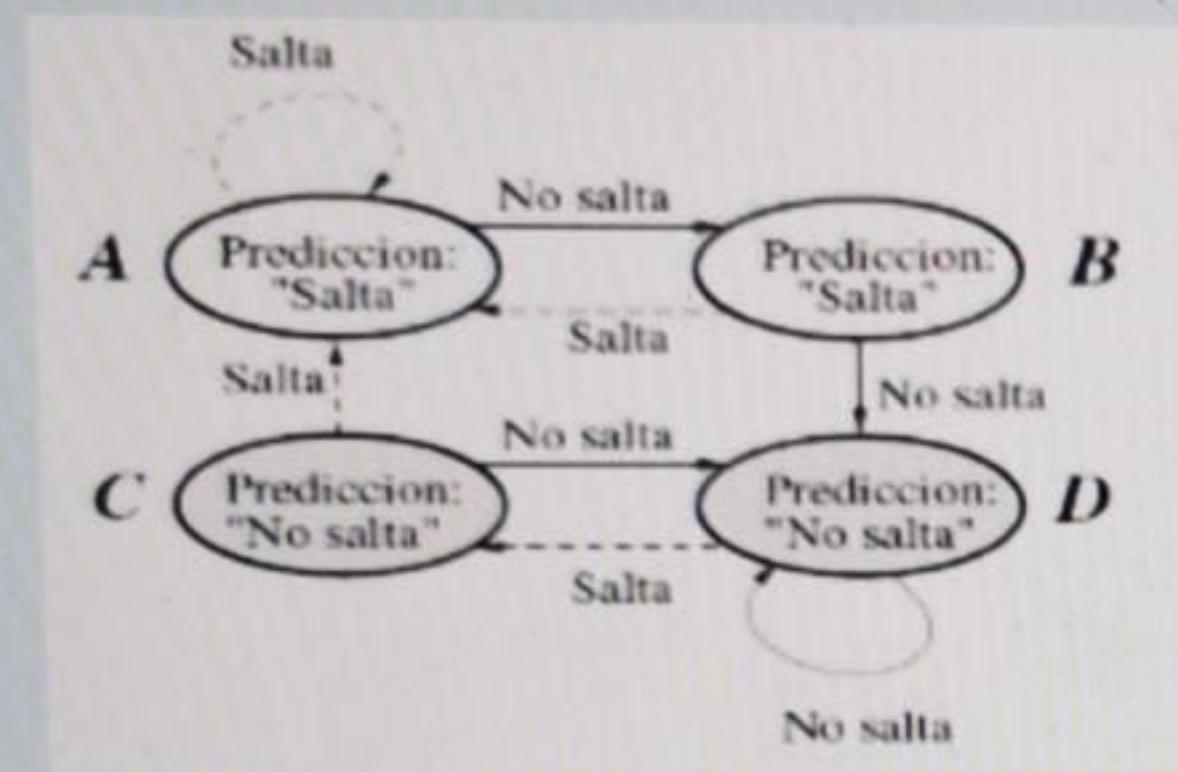
La ejecución de las instrucciones en una máquina superescalar con buffer de reorden o buffer de renombrado....

- a. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siguen un orden ascendente o descendente dependiendo de los riesgos RAW del programa almacenado en memoria.
- b. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siempre siguen un orden descendente conforme al orden del programa almacenado en memoria.
- c. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siempre siguen un orden ascendente conforme al orden del programa almacenado en memoria.
- d. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siguen un orden ascendente o descendente dependiendo de los riesgos WAR y WAW del programa almacenado en memoria.

En una máquina superescalar con buffer de reorden....

- a. En el ROB se renombran varias veces los registros del banco de registros.
- b. En el ROB se almacenan de forma temporal los valores que finalmente se guardarán en el banco de registros.
- c. En el ROB se almacenan de forma temporal los valores que volverán a pasar a la etapa EX pero que no se almacenarán en la etapa WB
- d. En el ROB se almacena una copia exacta del banco de registros.

Un máquina superescalar dispone de un predictor de saltos dinámico que utiliza BTB de 4 entradas y 2 bits de predicción como el que se muestra en la figura.



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de r1=-11 y r5=1:

```
0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x006
0x005 addi r6,r5,#1
0x006 subi r6, r5, #1
0x007 beqz r6, 0x001
0x008 subd f2,f1, f3
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x006	A
0x007	0x001	A

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada?

En una máquina superescalar con buffer de reorden o buffer de renombrado....

- a. La etapa WB sigue un orden ascendente o descendiente en ciclos dependiendo de los riesgos RAW
- b. La etapa WB siempre sigue un orden ascendente en ciclos.
- c. La etapa WB sigue un orden ascendente o descendiente en ciclos dependiendo de los riesgos WAR y WAW
- d. La etapa WB siempre sigue un orden descendiente en ciclos.

Dada una red multietapa tipo **Delta** con 785 entradas y 11062 salidas, se pide:

¿Cuántos conmutadores hay en la etapa 0 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

#### [ENGLISH]

Given a **Delta** multi-stage network with 785 inputs and 11062 outputs, calculate:

How many switches are there in the stage 0 of the most economical implementation of this topology?

**Note:** Remember that the first stage is labeled as 0, the second stage is labeled as 1, and so on.

Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer d reorden con número de entradas suficiente, la emisión puede ser desordenada pero **no** contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800  
addi r1,r1,#8  
addi r2,r2,#8  
ld f0,coef  
ld f2,-8(r1)  
ld f4,0(r1)  
muld f8,f2,f0  
addd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```



Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (5c), 1 FP add (7c), 3 ALU int (3c) y 3 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción **addi r2,r2,#8**?

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción **ld f4,0(r1)**?

¿Cuántos ciclos tarda el código?

Sea un computador superescalar capaz de decodificar 1 inst/c, emitir 1 inst/c, escribir 1 inst/c en los registros correspondientes y retirar 1 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800  
addi r1,r1,#8  
addi r2,r2,#8  
ld f0,coef  
ld f2,-8(r1)  
ld f4,0(r1)  
muld f8,f2,f0  
addd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (5c), 1 FP add (7c), 3 ALU int (3c) y 3 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?

13

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?

¿Cuántos ciclos tarda el código?

Por favor, responda a todas las partes de la pregunta.

Pregunta 1

Respuesta incompleta

Puntúa como  
2,50

Marcar pregunta

Sea un computador superescalar capaz de decodificar 4inst/c, emitir 4inst/c, escribir 4inst/c en los registros correspondientes y retirar 4 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (4c), 1 FP add (2c), 1 ALU int (1c) y 1 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?

5

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?

8

¿Cuántos ciclos tarda el código?

1  
sta  
da  
a como  
arcar  
unta

Sea un computador superescalar capaz de decodificar 4inst/c, emitir 4inst/c, escribir 4inst/c en los registros correspondientes y retirar 4 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada pero **no** contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 2 FP mul/div (8c), 2 FP add (4c), 2 ALU int (3c) y 2 load/store (5c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?



¿Cuántos ciclos tarda el código?

# 0-21\_INGENIERIA DE LOS COMPUTADORES

onal / Mis cursos / (34025\_1\_2\_3\_40\_2020-21) / Calificaciones - Marks / Examen final

En una máquina vectorial tenemos el código que se muestra a continuación:

```
lv v1,vector1;  
add v3,v1,v2;  
xor v4,v1,v2;  
sv vector3,v4;  
lv v2,vector2;
```

Suponiendo que tenemos vectores de 53 elementos y que las cargas y almacenamiento tienen una latencia de 6 ciclos. ¿Cuántos ciclos se tardaría en ejecutar el código en una máquina vectorial?

Respuesta:

La ejecución de las instrucciones en una máquina superescalar con buffer de reorden o buffer de renombrado....

- a. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siempre siguen un orden ascendente conforme al orden del programa almacenado en memoria.
- b. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siguen un orden ascendente o descendente dependiendo de los riesgos RAW del programa almacenado en memoria.
- c. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siguen un orden ascendente o descendente dependiendo de los riesgos WAR y WAW del programa almacenado en memoria.
- d. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siempre siguen un orden descendente conforme al orden del programa almacenado en memoria.

Dada una red multietapa tipo **Delta** con 960 entradas y 11464 salidas, se pide:

¿Cuántos **commutadores** hay en la etapa 1 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

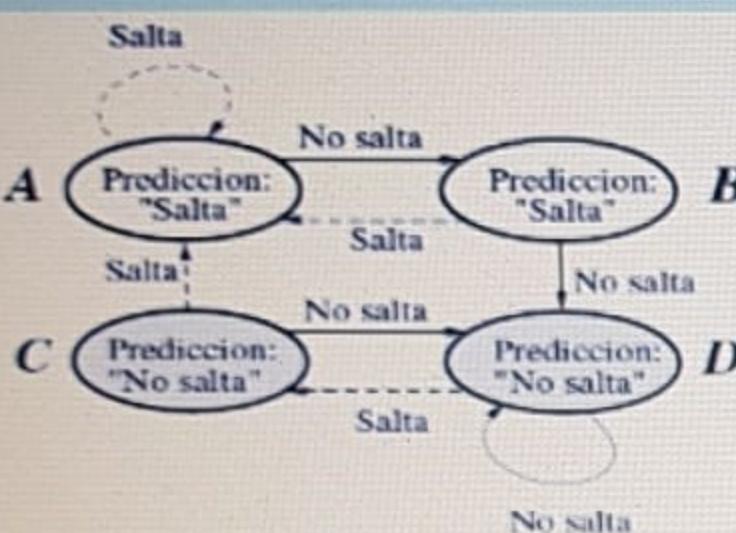
Pregunta 3

Respuesta  
guardada

Puntúa como  
1,50

Markar  
pregunta

Un máquina superescalar dispone de un predictor de saltos dinámico que utiliza BTB de 4 entradas y 2 bits de predicción como el que se muestra en la figura.



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de r1=-11 y r5=1:

```
0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x006
0x005 addi r6,r5,#1
0x006 subi r6, r5, #1
0x007 beqz r6, 0x001
0x008 subd f2,f1, f3
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x006	D
0x007	0x001	D

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración?

00

¿Cuáles será los bits de predicción de la segunda fila de la BTB en ese caso?

01

¿Cuántos ciclos tarda el código?

27

Pregunta 2

Sin responder  
aún

Puntúa como  
0,50

Marcar  
pregunta

En una máquina superescalar con buffer de reorden o buffer de renombrado....

- a. La etapa WB sigue un orden ascendente o descendiente en ciclos dependiendo de los riesgos RAW
- b. La etapa WB siempre sigue un orden descendiente en ciclos.
- c. La etapa WB sigue un orden ascendente o descendiente en ciclos dependiendo de los riesgos WAR y WAW
- d. La etapa WB siempre sigue un orden ascendente en ciclos.

En una máquina superescalar con buffer de reorden....

- a. En el ROB se almacenan de forma temporal los valores que finalmente se guardarán en el banco de registros.
- b. En el ROB se renombran varias veces los registros del banco de registros.
- c. En el ROB se almacenan de forma temporal los valores que volverán a pasar a la etapa EX pero que no se almacenarán en la etapa WB
- d. En el ROB se almacena una copia exacta del banco de registros.

Un máquina superescalar dispone de un predictor de saltos dinámico que utiliza BTB de 4 entradas y 2 bits de predicción como el que se muestra en la figura.



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de r1=-11 y r5=1:

```
0x000 addi r3,r1,#80  
0x001 addi r1,r1,#8  
0x002 addi r2,r2,#8  
0x003 addi r5,r1,#3  
0x004 beqz r5, 0x006  
0x005 addi r6,r5,#1  
0x006 subi r6, r5, #1  
0x007 beqz r6, 0x001  
0x008 subd f2,f1, f3
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x006	B
0x007	0x001	B

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración en el código?

¿Cuáles será los bits de predicción de la segunda fila de la BTB en ese caso?

Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada pero **no** contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800  
addi r1,r1,#8  
addi r2,r2,#8  
ld f0,coef  
ld f2,-8(r1)  
ld f4,0(r1)  
muld f8,f2,f0  
addd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (4c), 1 FP add (2c), 3 ALU int (1c) y 1 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción **addi r2,r2,#8**?

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción **ld f4,0(r1)**?

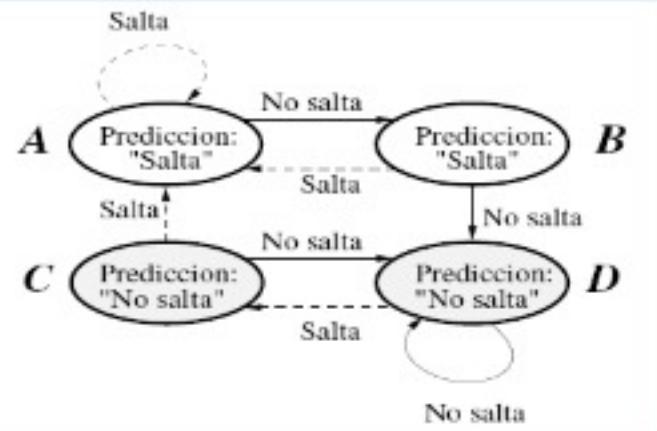
¿Cuántos ciclos tarda el código?

**Pregunta 2**Sin responder  
aúnPuntúa como  
0,50Marcar  
pregunta

En una máquina superescalar con circuito de adelantamiento y con emisión no alineada...

- a. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones siempre que exista hueco.
- b. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones si previamente se han ejecutado todas las instrucciones que pasaron en ciclos anteriores a la ventana.
- c. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones siempre en el siguiente ciclo de la búsqueda.
- d. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones si previamente se han ejecutado todas las instrucciones que pasaron en ciclos anteriores a la ventana y exista hueco.

Un máquina superescalar dispone de un predictor de saltos dinámico que utiliza BTB de 4 entradas y 2 bits de predicción como el que se muestra en la figura.



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de r1=-11 y r5=1:

```
0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x006
0x005 addi r6,r5,#1
0x006 subi r6, r5, #1
0x007 beqz r6, 0x001
0x008 subd f2,f1, f3
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x006	A
0x007	0x001	C

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración en el código?

¿Cuáles será los bits de predicción de la segunda fila de la BTB en ese caso?

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo **T1** y 4 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 38% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

In a vector computer we have the code shown below:

```
lv v1,vector1;  
add v3,v1,v2;  
xor v4,v1,v2;  
sv vector3,v4:  
lv v2,vector2;
```

Assuming we have vectors of 381 elements and that loads and stores have a latency of 12 cycles, while arithmetic-logic operations have a latency of 4 cycles. How many cycles would it take to execute the code on a vector machine with chaining of operations?

Respuesta:



Dada una red multietapa tipo **Delta** con 758 entradas y 16178 salidas, se pide:

¿Cuántos conmutadores hay en la etapa 1 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

**[ENGLISH]**

Given a **Delta** multi-stage network with 758 inputs and 16178 outputs, calculate:

How many switches are there in the stage 1 of the most economical implementation of this topology?

**Note:** Remember that the first stage is labeled as 0, the second stage is labeled as 1, and so on.

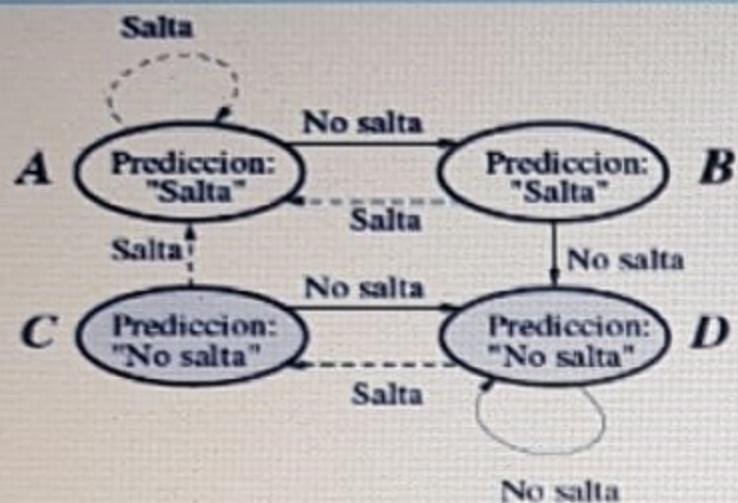
Respuesta:

guardada

Puntúa como  
1,50

Markar  
pregunta

predicción como el que se muestra en la figura.



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de  $r1=-11$  y  $r5=1$ :

```
0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x006
0x005 addi r6,r5,#1
0x006 subi r6, r5, #1
0x007 beqz r6, 0x001
0x008 subd f2,f1, f3
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x006	D
0x007	0x001	D

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración?

00

¿Cuáles será los bits de predicción de la segunda fila de la BTB en ese caso?

01

Dada una red multietapa tipo **Delta** con 896 entradas y 15429 salidas, se pide:

¿Cuántos comutadores hay en la etapa 3 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

[ENGLISH]

Given a **Delta** multi-stage network with 896 inputs and 15429 outputs, calculate:

How many switches are there in the stage 3 of the most economical implementation of this topology?

**Note:** Remember that the first stage is labeled as 0, the second stage is labeled as 1, and so on.

Respuesta:

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo **T1** y 4 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 64% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 4 cores of a certain **T1** type and 4 cores of another **T2** type. Suppose that **T1** cores are 64% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

Respuesta:



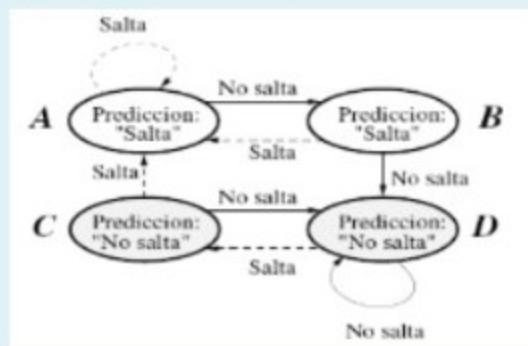
### Pregunta 3

Sin responder aún

Puntúa como 1,50

Marcar pregunta

Un máquina superescalar dispone de un predictor de saltos dinámico que utiliza BTB de 4 entradas y 2 bits de predicción como el que se muestra en la figura.



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de  $r1=-11$  y  $r5=1$ :

```

0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x006
0x005 addi r6,r5,#1
0x006 subi r6, r5, #1
0x007 beqz r6, 0x001
0x008 subd f2,f1, f3

```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x006	A
0x007	0x001	A

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración?

¿Cuáles será los bits de predicción de la segunda fila de la BTB en ese caso?

Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada pero **no** contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800  
addi r1,r1,#8  
addi r2,r2,#8  
ld f0,coef  
ld f2,-8(r1)  
ld f4,0(r1)  
muld f8,f2,f0  
addd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (4c), 1 FP add (2c), 3 ALU int (1c) y 1 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción **addi r2,r2,#8**?

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción **ld f4,0(r1)**?

¿Cuántos ciclos tarda el código?

- Mem[M] returns the byte at memory location M (in hexadecimal).

Ayuda/Help: Estado inicial / Initial state:

	Cache Proc 1		Cache Proc 2		Cache Proc 3		C
Línea Line	Datos Data	Estado State	Datos Data	Estado State	Datos Data	Estado State	D
0	?	I	?	I	?	I	
1	?	I	?	I	?	I	

- a.
- Línea 1 de P2 en estado S, Línea 0 de P2 en estado E, Línea 0 de P3 en estado S.
  - Line 1 of P2 in state S, Line 0 of P2 in state E, Line 0 of P3 in state S.

- b.
- Línea 0 de P1 en estado M, Línea 0 de P3 en estado E, Mem[0x1]=0x7
  - Line 0 of P1 in M-state, Line 0 of P3 in E-state, Mem[0x1]=0x7

- c.
- Línea 0 de P1 en estado E, Línea 0 de P3 en estado E, Mem[0x1]=0x7
  - Line 0 of P1 in E-state, Line 0 of P3 in E-state, Mem[0x1]=0x7

Pregunta 8

Respuesta  
guardadaPuntúa como  
1,25Mark  
pregunta

Se tiene un cierto multiprocesador de **4 núcleos** conectados a un bus común que tiene 64GiB de memoria principal e implementa el protocolo **MESI** para mantener la coherencia entre sus cachés. El sistema de caché tiene **2 bytes por línea o bloque de caché**. El tamaño de cada caché es de tan sólo **2 líneas** y la política de reemplazo es LRU.

El estado inicial de TODAS las líneas es "inválido". Suponga que toda la memoria está inicializada a cero.  
Suponga que se producen los siguientes accesos a memoria (**suponga datos de 8 bits**):

**Accesos a memoria:**

- 1: P2\_Read@0x05
- 2: P2\_Read@0x03
- 3: P3\_Read@0x04
- 4: P1\_Write@(0x1, 3)

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

Recuerdo la notación (la misma que la que hemos visto en clase):

- **PJ:Write(@M,N)** significa: el procesador J escribe el valor N en la posición de memoria M (dada en hexadecimal)
- **PJ:Read@M** significa: el procesador J lee de la dirección M (dada en hexadecimal). - sadasdsad
- Mem[M] devuelve el byte en la posición M (en hexadecimal) de memoria.

**[ENGLISH]**

You have a **4-core** multiprocessor connected to a common bus that has 64GiB of main memory and implements the **MESI protocol** to maintain consistency between its caches. The cache system has **2 bytes per line (and block) of cache**. The size of each cache is only **2 lines** and the replacement policy is LRU.

The initial state of ALL lines is "invalid". Assume that all memory is initialized to zero.

Se tiene un cierto multiprocesador de **4 núcleos** conectados a un bus común que tiene 64GiB de memoria principal e implementa el protocolo **MESI** para mantener la coherencia entre sus cachés. El sistema de caché tiene **2 bytes por línea o bloque de caché**. El tamaño de cada caché es de tan sólo **2 líneas** y la política de reemplazo es LRU.

El estado inicial de TODAS las líneas es "inválido". Suponga que toda la memoria está inicializada a cero.

Suponga que se producen los siguientes accesos a memoria (**suponga datos de 8 bits**):

**Accesos a memoria:**

- 1: P1\_Write@(0x01,7)
- 2: P2\_Read@0x00
- 3: P3\_Read@0x03
- 4: P4\_Write@(0x4, 9)

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

Recuerdo la notación (la misma que la que hemos visto en clase):

- **PJ:Write(@M,N)** significa: el procesador J escribe el valor N en la posición de memoria M (dada en hexadecimal)
- **PJ:Read@M** significa: el procesador J lee de la dirección M (dada en hexadecimal). - sadasdsad
- Mem[M] devuelve el byte en la posición M (en hexadecimal) de memoria.

**[ENGLISH]**

You have a **4-core** multiprocessor connected to a common bus that has 64GiB of main memory and implements the **MESI protocol** to maintain consistency between its caches. The cache system has **2 bytes per line (and block) of cache**. The size of each cache is only **2 lines** and the replacement policy is LRU.

The initial state of ALL lines is "invalid". Assume that all memory is initialized to zero.

After that, the following memory accesses occur (**suppose 8 bits data**):

Memory accesses:

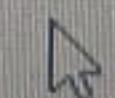
- 
- 1: P1\_Write@(0x01,7)
  - 2: P2\_Read@0x00
  - 3: P3\_Read@0x03
  - 4: P4\_Write@(0x4, 9)

En una máquina vectorial tenemos el código que se muestra a continuación:

```
lv v1,vector1;  
add v3,v1,v2;  
xor v4,v1,v2;  
sv vector3,v4:  
lv v2,vector2;
```

Suponiendo que tenemos vectores de 97 elementos y que las cargas y almacenamiento tiene una latencia de 11 ciclos, mientras que las operaciones aritmético-lógicas tienen una latencia de 3 ciclos. ¿Cuántos ciclos se tardaría en ejecutar el código en una máquina vectorial con encadenamiento de operaciones?

Respuesta:



Dada una red multietapa tipo Delta con 859 entradas y 10071 salidas, se pide:  
¿Cuántos conmutadores hay en la etapa 3 de la implementación más económica?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

- a. • Líneas 0 de P1y P2 en estado S, Línea 0 de P3 en estado E, Línea 0 de P4 en estado M.
  - Lines 0 of P1and P2 in state S, Line 0 of P3 in state E, Line 0 of P4 in state M.
- b. • Líneas 0 de P1y P2 en estado E, Línea 0 de P3 en estado E, Línea 0 de P4 en estado M.
  - Lines 0 of P1and P2 in state E, Line 0 of P3 in state E, Line 0 of P4 in state M.
- c. • Línea 0 de P1 en estado M, Línea 0 de P3 en estado E, Mem[0x1]=0x7
  - Line 0 of P1 in M-state, Line 0 of P3 in E-state, Mem[0x1]=0x7
- d. • Línea 0 de P1 en estado E, Línea 0 de P3 en estado E, Mem[0x1]=0x7
  - Line 0 of P1 in E-state, Line 0 of P3 in E-state, Mem[0x1]=0x7



c. • Línea 0 de P1 en estado M, Línea 0 de P3 en estado E, Mem[0x1]=0x7

• Line 0 of P1 in M-state, Line 0 of P3 in E-state, Mem[0x1]=0x7

Dada una red multietapa tipo **Delta** con 781 entradas y 11625 salidas, se pide:

¿Cuántos commutadores hay en la etapa 4 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

[ENGLISH]

Given a **Delta** multi-stage network with 781 inputs and 11625 outputs, calculate:

How many switches are there in the stage 4 of the most economical implementation of this topology?

**Note:** Remember that the first stage is labeled as 0, the second stage is labeled as 1, and so on.

Respuesta:

Pregunta **8**

Sin responder  
aún

Puntúa como  
1,25

◀ Marcar  
pregunta

Se tiene un cierto multiprocesador de **4 núcleos** conectados a un bus común que tiene 64GiB de memoria principal e implementa el protocolo **MESI** para mantener la coherencia entre sus cachés. El sistema de caché tiene **2 bytes por línea o bloque de caché**. El tamaño de cada caché es de tan sólo **2 líneas** y la política de reemplazo es LRU.

El estado inicial de TODAS las líneas es "inválido". Suponga que toda la memoria está inicializada a cero. Suponga que se producen los siguientes accesos a memoria (**suponga datos de 8 bits**):

**Accesos a memoria:**

- 1: P1\_Write@(0x01,7)
- 2: P2\_Read@0x00
- 3: P3\_Read@0x03
- 4: P4\_Write@(0x4, 9)

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

Recuerdo la notación (la misma que la que hemos visto en clase):

- **PJ:Write(@M,N)** significa: el procesador J escribe el valor N en la posición de memoria M (dada en hexadecimal)
- **PJ:Read@M** significa: el procesador J lee de la dirección M (dada en hexadecimal). - sadasdsad
- Mem[M] devuelve el byte en la posición M (en hexadecimal) de memoria.

**[ENGLISH]**

You have a **4-core** multiprocessor connected to a common bus that has 64GiB of main memory and

Dada una red multietapa tipo **Delta** con 1019 entradas y 12123 salidas, se pide:

¿Cuántos conmutadores hay en la etapa 0 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

[ENGLISH]

Given a **Delta** multi-stage network with 1019 inputs and 12123 outputs, calculate:

How many switches are there in the stage 0 of the most economical implementation of this topology?

**Note:** Remember that the first stage is labeled as 0, the second stage is labeled as 1, and so on.

Respuesta:

Se tiene un cierto multiprocesador de **4 núcleos** conectados a un bus común que tiene 64GiB de memoria principal e implementa el protocolo **MESI** para mantener la coherencia entre sus cachés. El sistema de caché tiene **2 bytes por línea o bloque de caché**. El tamaño de cada caché es de tan sólo **2 líneas** y la política de reemplazo es LRU.

El estado inicial de TODAS las líneas es "inválido". Suponga que toda la memoria está inicializada a cero.

Suponga que se producen los siguientes accesos a memoria (**suponga datos de 8 bits**):

#### Accesos a memoria:

- 1: P1\_Read@0x01
- 2: P1\_Read@0x00
- 3: P2\_Read@0x03
- 4: P3\_Read@0x02

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

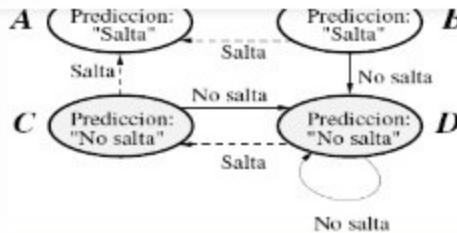
Recuerdo la notación (la misma que la que hemos visto en clase):

- **PJ:Write(@M,N)** significa: el procesador J escribe el valor N en la posición de memoria M (dada en hexadecimal)
- **PJ:Read@M** significa: el procesador J lee de la dirección M (dada en hexadecimal). - sadasdsad
- Mem[M] devuelve el byte en la posición M (en hexadecimal) de memoria.

- a. • Línea 0 de P1 en estado E, línea 0 de P2 y P3 en estado S, memoria actualizada.
  - Line 0 of P1 in state E, line 0 of P2 and P3 in state S, memory updated.
- b. • Ninguna es correcta.
  - Non of them is correct.
- c. • Líneas 0 y 1 de P1 en estado E, línea 0 de P2 y P3 en estado S, memoria actualizada.
  - Lines 0 and 1 of P1 in state E, line 0 of P2 and P3 in state S, memory updated.
- d. • Líneas 0 y 1 de P1 en estado E, línea 0 de P2 y P3 en estado S, memoria no actualizada.
  - Lines 0 and 1 of P1 in state E, line 0 of P2 and P3 in state S, memory not updated.

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 5 núcleos de un cierto tipo **T1** y 5 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 58% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

Marcar pregunta


Suponed que se está ejecutando el siguiente código, siendo el valor inicial de r1=-11 y r5=1:

```

0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x006
0x005 addi r6,r5,#1
0x006 subi r6, r5, #1
0x007 beqz r6, 0x001
0x008 subd f2,f1, f3
  
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTB	Bits
0x004	0x006	D
0x007	0x001	D

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración en el código

¿Cuáles será los bits de predicción de la segunda fila de la BTB en ese caso?

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo **T1** y 4 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 68% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 4 cores of a certain **T1** type and 4 cores of another **T2** type. Suppose that **T1** cores are 68% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

Respuesta:

Pregunta 3

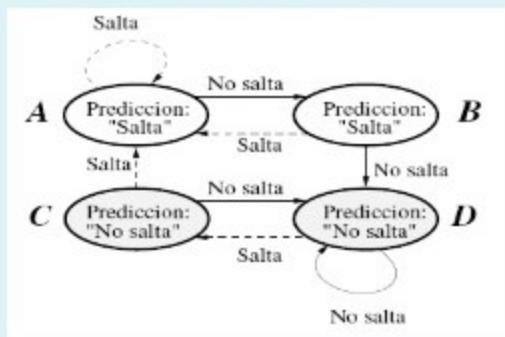
Sin responder

aún

Puntúa como  
1,50

Markar  
pregunta

Un máquina superescalar dispone de un predictor de saltos dinámico que utiliza BTB de 4 entradas y 2 bits de predicción como el que se muestra en la figura.



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de r1=-11 y r5=1:

```
0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x006
0x005 addi r6,r5,#1
0x006 subi r6, r5, #1
0x007 beqz r6, 0x001
0x008 subd f2,f1, f3
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x006	D
0x007	0x001	D

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración en el código

Dada una red multietapa tipo **Delta** con 781 entradas y 11625 salidas, se pide:

¿Cuántos conmutadores hay en la etapa 4 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

**[ENGLISH]**

Given a **Delta** multi-stage network with 781 inputs and 11625 outputs, calculate:

How many switches are there in the stage 4 of the most economical implementation of this topology?

**Note:** Remember that the first stage is labeled as 0, the second stage is labeled as 1, and so on.

Respuesta:

Pregunta 2

Sin responder  
aún

Puntaría como  
0,50

✗ Marcar  
pregunta



La ejecución de las instrucciones en una máquina superescalar con buffer de reorden o buffer de renombrado....

- a. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siempre siguen un orden ascendente conforme al orden del programa almacenado en memoria.
- b. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siguen un orden ascendente o descendente dependiendo de los riesgos RAW del programa almacenado en memoria.
- c. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siguen un orden ascendente o descendente dependiendo de los riesgos WAR y WAW del programa almacenado en memoria.
- d. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siempre siguen un orden descendente conforme al orden del programa almacenado en memoria.

Dada una red multietapa tipo Delta con 859 entradas y 10071 salidas, se pide:

¿Cuántos conmutadores hay en la etapa 3 de la implementación más económica de esta topología?

Nota: Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

La ejecución de las instrucciones en una máquina superescalar con buffer de reorden o buffer de renombrado...

- a. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siempre siguen un orden ascendente conforme al orden del programa almacenado en memoria.
- b. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siguen un orden ascendente o descendente dependiendo de los riesgos WAR y WAW del programa almacenado en memoria.
- c. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siempre siguen un orden descendente conforme al orden del programa almacenado en memoria.
- d. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siguen un orden ascendente o descendente dependiendo de los riesgos RAW del programa almacenado en memoria.

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 6 núcleos de un cierto tipo **T1** y 6 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 74% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 6 cores of a certain **T1** type and 6 cores of another **T2** type. Suppose that **T1** cores are 74% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

Respuesta:

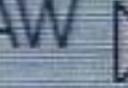
- b) ¿Qué fracción del problema total debe ejecutar cualquiera de los núcleos más rápidos?

[ENGLISH]

- b) What fraction of the total problem should any of the faster cores run?

Respuesta:

En una máquina superescalar con buffer de reorden o buffer de renombrado...

- a. La etapa WB siempre sigue un orden ascendente en ciclos.
- b. La etapa WB sigue un orden ascendente o descendiente en ciclos dependiendo de los riesgos RAW 
- c. La etapa WB siempre sigue un orden descendiente en ciclos.
- d. La etapa WB sigue un orden ascendente o descendiente en ciclos dependiendo de los riesgos WAR y WAW

Se tiene un cierto multiprocesador de **4 núcleos** conectados a un bus común que tiene 64GiB de memoria principal e implementa el protocolo **MESI** para mantener la coherencia entre sus cachés. El sistema de caché tiene **2 bytes por línea o bloque de caché**. El tamaño de cada caché es de tan sólo **2 líneas** y la política de reemplazo es LRU.

El estado inicial de TODAS las líneas es "inválido". Suponga que toda la memoria está inicializada a cero.

Suponga que se producen los siguientes accesos a memoria (**suponga datos de 8 bits**):

#### **Accesos a memoria:**

- 1: P1\_Read@0x01
- 2: P1\_Read@0x00
- 3: P2\_Read@0x03
- 4: P3\_Read@0x02

Pregunta **2**

Sin responder  
aún

Puntúa como  
0,50

☒ Marcar  
pregunta

En una máquina superescalar con buffer de reorden...

- a. En el ROB se almacenan de forma temporal los valores que volverán a pasar a la etapa EX pero que no se almacenarán en la etapa WB
- b. En el ROB se almacenan de forma temporal los valores que finalmente se guardarán en el banco de registros.
- c. En el ROB se almacena una copia exacta del banco de registros.
- d. En el ROB se renombran varias veces los registros del banco de registros.

Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entrada suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)           |
ld f4,0(r1)             |
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (4c), 1 FP add (2c), 3 ALU int (1c) y 1 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción **addi r2,r2,#8**?

6

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción **ld f4,0(r1)**?

11

¿Cuántos ciclos tarda el código?

22

Pregunta 1

Sin responder  
aún

Puntúa como  
2,50

Markar  
pregunta

Sea un computador superescalar capaz de decodificar 1 inst/c, emitir 1 inst/c, escribir 1 inst/c en los registros correspondientes y retirar 1 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (5c), 1 FP add (7c), 3 ALU int (3c) y 3 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?

9

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?

¿Cuántos ciclos tarda el código?

En una máquina vectorial tenemos el código que se muestra a continuación:

```
lv v1,vector1;  
add v3,v1,v2;  
xor v4,v1,v2;  
sv vector3,v4:  
lv v2,vector2;
```

```
lv v1,  
add v3,v1,v2;  
xor v4,v1,v2;  
sv vector3,v4:  
lv v2;
```

Suponiendo que tenemos vectores de 233 elementos y que las cargas y almacenamiento tiene una latencia de 10 ciclos, mientras que las operaciones aritmético-lógicas tienen una latencia de 4 ciclos. ¿Cuántos ciclos se tardaría en ejecutar el código en una máquina vectorial sin encadenamiento de operaciones?

Respuesta:

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 3 núcleos de un cierto tipo **T1** y 3 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 39% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 3 cores of a certain **T1** type and 3 cores of another **T2** type. Suppose that **T1** cores are 39% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

Respuesta:

Sea un computador superescalar capaz de decodificar 4inst/c, emitir 4inst/c, escribir 4inst/c en los registros correspondientes y retirar 4 inst/c. Si reorden con número de entradas suficiente, la emisión puede ser desordenada pero **no** contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 2 FP mul/div (8c), 2 FP add (4c), 2 ALU int (3c) y 2 load/store (5c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?

¿Cuántos ciclos tarda el código?

Dada una red multietapa tipo **Delta** con 960 entradas y 11464 salidas, se pide:

¿Cuántos conmutadores hay en la etapa 1 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

[ENGLISH]

Dada una red multietapa tipo **Delta** con 797 entradas y 10038 salidas, se pide:  
¿Cuántos conmutadores hay en la etapa 3 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

**[ENGLISH]**

Given a **Delta** multi-stage network with 797 inputs and 10038 outputs, calculate:

How many switches are there in the stage 3 of the most economical implementation of this topology?

**Note:** Remember that the first stage is labeled as 0, the second stage is labeled as 1, and so on.

Respuesta:

Pregunta 1

Sin responder  
aún

Puntúa como  
2,50

☒ Marcar  
pregunta

Sea un computador superescalar capaz de decodificar 1 inst/c, emitir 1 inst/c, escribir 1 inst/c en los registros correspondientes y retirar 1 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (5c), 1 FP add (7c), 3 ALU int (3c) y 3 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?

9

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?

22

¿Cuántos ciclos tarda el código?

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo **T1** y 4 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 33% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 4 cores of a certain **T1** type and 4 cores of another **T2** type. Suppose that **T1** cores are 33% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

Respuesta:

1,205

En una máquina superescalar con buffer de reorden o buffer de renombrado....

- a. La etapa WB sigue un orden ascendente o descendiente en ciclos dependiendo de los riesgos RAW
- b. La etapa WB siempre sigue un orden ascendente en ciclos.
- c. La etapa WB sigue un orden ascendente o descendiente en ciclos dependiendo de los riesgos WAR y WAW
- d. La etapa WB siempre sigue un orden descendiente en ciclos.

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo **T1** y 4 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 26% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 4 cores of a certain **T1** type and 4 cores of another **T2** type. Suppose that **T1** cores are 26% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

Respuesta:

En una máquina vectorial tenemos el código que se muestra a continuación:

```
lv v1,vector1;  
add v3,v1,v2;  
xor v4,v1,v2;  
sv vector3,v4:  
lv v2,vector2;
```

Suponiendo que tenemos vectores de 450 elementos y que las cargas y almacenamiento tiene una latencia de 11 ciclos, mientras que las operaciones aritmético-lógicas tienen una latencia de 6 ciclos. ¿Cuántos ciclos se tardaría en ejecutar el código en una máquina vectorial sin encadenamiento de operaciones?

Respuesta:

¿Cuántos ciclos tarda el código?

35

Pregunta 2

Sin responder  
aún

Puntúa como  
0,50

Marcar  
pregunta

En una máquina superescalar con buffer de reorden....

- a. En el ROB se almacenan de forma temporal los valores que volverán a pasar a la etapa EX pero que no se almacenarán en la etapa WB
- b. En el ROB se almacenan de forma temporal los valores que finalmente se guardarán en el banco de registros.
- c. En el ROB se almacena una copia exacta del banco de registros.
- d. En el ROB se renombran varias veces los registros del banco de registros.

[Quitar mi elección](#)

Pregunta 1

Sin responder  
aún

Puntúa como  
2,50

Marcar  
pregunta

Sea un computador superescalar capaz de decodificar 1 inst/c, emitir 1 inst/c, escribir 1 inst/c en los registros correspondientes y retirar 1 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (5c), 1 FP add (7c), 3 ALU int (3c) y 3 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?

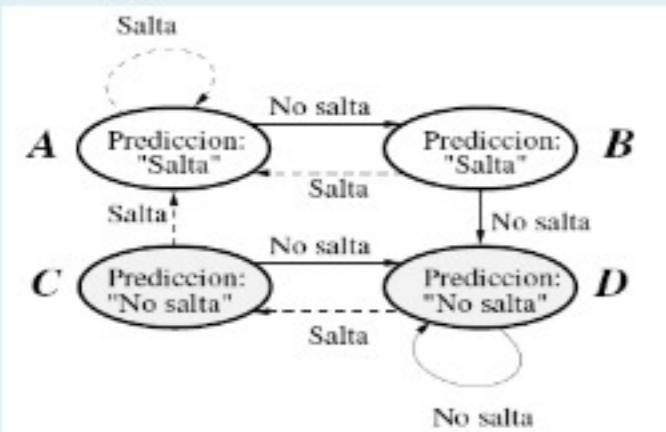
9

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?

22

¿Cuántos ciclos tarda el código?

Un máquina superescalar dispone de un predictor de saltos dinámico que utiliza BTB de 4 entradas y 2 bits de predicción como el que se muestra en la figura.



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de r1=-11 y r5=1:

```
0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x006
0x005 addi r6,r5,#1
0x006 subi r6, r5, #1
0x007 beqz r6, 0x001
0x008 subd f2,f1, f3
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x006	A
0x007	0x001	C

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración en el código?

¿Cuáles será los bits de predicción de la segunda fila de la BTB en ese caso?

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo **T1** y 4 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 64% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 4 cores of a certain **T1** type and 4 cores of another **T2** type. Suppose that **T1** cores are 64% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

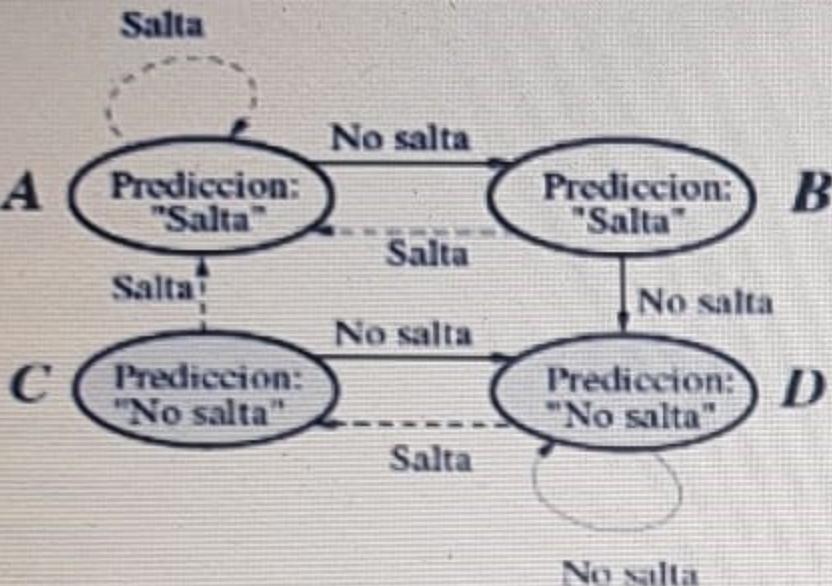
Respuesta:

guardada

Puntúa como

1,50

Marcar pregunta



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de  $r1=-11$  y  $r5=1$ :

```
0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x006
0x005 addi r6,r5,#1
0x006 subi r6, r5, #1
0x007 beqz r6, 0x001
0x008 subd f2,f1, f3
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x006	D
0x007	0x001	D

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración?

00

¿Cuáles será los bits de predicción de la segunda fila de la BTB en ese caso?

01

En una máquina vectorial tenemos el código que se muestra a continuación:

```
lv v1,vector1;  
add v3,v1,v2;  
xor v4,v1,v2;  
sv vector3,v4:  
lv v2,vector2;
```

Suponiendo que tenemos vectores de 211 elementos y que las cargas y almacenamiento tiene una latencia de 8 ciclos, mientras que las operaciones aritmético-lógicas tienen una latencia de 6 ciclos. ¿Cuántos ciclos se tardaría en ejecutar el código en una máquina vectorial con encadenamiento de operaciones?

Respuesta:

2 instrucciones  $\rightarrow$  aritmética/logic (2 ciclos)  
3 instrucciones  $\rightarrow$  load/store (11 ciclos)

$$TCV = 3 \cdot TL1(\text{store/load}) + 2 * TL1(\text{arithmetic/logic}) + \text{Long Vector}$$
$$= 3 \cdot 11 + 2 \cdot 2 + 208 = 245 \text{ cycles}$$

Se tiene un cierto multiprocesador de **4 núcleos** conectados a un bus común que tiene 64GiB de memoria principal e implementa el protocolo **MESI** para mantener la coherencia entre sus cachés. El sistema de caché tiene **2 bytes por línea o bloque de caché**. El tamaño de cada caché es de tan sólo **2 líneas** y la política de reemplazo es LRU.

El estado inicial de TODAS las líneas es "inválido". Suponga que toda la memoria está inicializada a cero.

Suponga que se producen los siguientes accesos a memoria (**suponga datos de 8 bits**):

**Accesos a memoria:**

- 1: P1\_Read@0x01
- 2: P1\_Read@0x00
- 3: P2\_Read@0x03
- 4: P3\_Read@0x02

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

Recuerdo la notación (la misma que la que hemos visto en clase):

- **PJ:Write(@M,N)** significa: el procesador J escribe el valor N en la posición de memoria M (dada en hexadecimal)
- **PJ:Read@M** significa: el procesador J lee de la dirección M (dada en hexadecimal). - sadasdsad
- Mem[M] devuelve el byte en la posición M (en hexadecimal) de memoria.

- a. • Línea 0 de P1 en estado E, línea 0 de P2 y P3 en estado S, memoria actualizada.
  - Line 0 of P1 in state E, line 0 of P2 and P3 in state S, memory updated.
- b. • Ninguna es correcta.
  - Non of them is correct.
- c. • Líneas 0 y 1 de P1 en estado E, línea 0 de P2 y P3 en estado S, memoria actualizada.
  - Lines 0 and 1 of P1 in state E, line 0 of P2 and P3 in state S, memory updated.
- d. • Líneas 0 y 1 de P1 en estado E, Línea 0 de P2 y P3 en estado S, memoria no actualizada.
  - Lines 0 and 1 of P1 in state E, line 0 of P2 and P3 in state S, memory not updated.

En una máquina vectorial tenemos el código que se muestra a continuación:

```
lv v1,vector1;  
add v3,v1,v2;  
xor v4,v1,v2;  
sv vector3,v4:  
lv v2,vector2;
```

Suponiendo que tenemos vectores de 36 elementos y que las cargas y almacenamiento tiene una latencia de 10 ciclos, mientras que las operaciones aritmético-lógicas tienen una latencia de 4 ciclos. ¿Cuántos ciclos se tardaría en ejecutar el código en una máquina vectorial con encadenamiento de operaciones?

Respuesta:

## ¿Cuántos ciclos tarda el código?

29

En una máquina superescalar con buffer de reorden o buffer de renombrado....

- a. La etapa WB siempre sigue un orden descendiente en ciclos.
- b. La etapa WB siempre sigue un orden ascendente en ciclos.
- c. La etapa WB sigue un orden ascendente o descendiente en ciclos dependiendo de los riesgos RAW
- d. La etapa WB sigue un orden ascendente o descendiente en ciclos dependiendo de los riesgos WAR y WAW

Ir a...

ta 2  
sponder

a como

arcar  
unta

**b) ¿Qué fracción del problema total debe ejecutar cualquiera de los núcleos más rápidos?**

[ENGLISH]

**b) What fraction of the total problem should any of the faster cores run?**

Respuesta:

Memory accesses:

- 1: P2\_Write@(0x02,7)
- 2: P1\_Write@(0x03,9)
- 3: P4\_Read@0x02
- 4: P3\_Read@0x1

Select which of the following items matches the state of the caches and/or main memory at the end of this sequence.

The notation is the same as we have seen in class:

- **PJ:Write(@M,N)** means: processor J writes the value N to memory location M (given in hexadecimal).
- **PJ:Read@M** means: processor J reads from address M (given in hexadecimal).
- Mem[M] returns the byte at memory location M (in hexadecimal).

Ayuda/Help: Estado inicial / Initial state:

	Cache Proc 1		Cache Proc 2		Cache Proc 3		Cache Proc 4	
Línea Line	Datos Data	Estado State	Datos Data	Estado State	Datos Data	Estado State	Datos Data	Estado State
0	?	I	?	I	?	I	?	I
1	?	I	?	I	?	I	?	I

Pregunta 2

Sin responder  
aún

Puntúa como  
0,50

▼ Marcar  
pregunta

En una máquina superescalar con circuito de adelantamiento y con emisión no alineada...

- a. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones siempre que exista hueco.
- b. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones siempre en el siguiente ciclo de la búsqueda.
- c. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones si previamente se han ejecutado todas las instrucciones que pasaron en ciclos anteriores a la ventana.
- d. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones si previamente se han ejecutado todas las instrucciones que pasaron en ciclos anteriores a la ventana y exista hueco.

En una máquina superescalar con circuito de adelantamiento y con emisión no alineada...

- a. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones si previamente se han ejecutado todas las instrucciones que pasaron en ciclos anteriores a la ventana y exista hueco.
- b. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones si previamente se han ejecutado todas las instrucciones que pasaron en ciclos anteriores a la ventana.
- c. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones siempre que exista hueco.
- d. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones siempre en el siguiente ciclo de la búsqueda.

Se tiene un cierto multiprocesador de 4 núcleos conectados a un bus común que tiene 64GiB de memoria principal e implementa el protocolo MESI para mantener la coherencia entre sus cachés. El sistema de caché tiene 2 bytes por línea o bloque de caché. El tamaño de cada caché es de tan sólo 2 líneas y la política de reemplazo es LRU.

El estado inicial de TODAS las líneas es "inválido". Suponga que toda la memoria está inicializada a cero.

Suponga que se producen los siguientes accesos a memoria (suponga datos de 8 bits):

**Accesos a memoria:**

- 1: P2\_Read@0x05
- 2: P2\_Read@0x03
- 3: P3\_Read@0x04
- 4: P1\_Write@0x1, 2)

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

Recuerde la notación (la misma que la que hemos visto en clase):

- **P<sub>i</sub>:Write(N|M,M)** significa: el procesador *i* escribe el valor N en la posición de memoria M (dada en hexadecimal)
- **P<sub>i</sub>:Read(M|M)** significa: el procesador *i* lee de la dirección M (dada en hexadecimal). - sadasdsad
- **Mem(M)** denota el byte en la posición M (en hexadecimal) de memoria.

**Opciones:**

You have a 4-core multiprocessor connected to a common bus that has 64GiB of main memory and implements the **MESI** protocol to maintain consistency between its caches. The cache system has **2 bytes per line (line blocks) of cache**. The size of each cache is only **2 lines** and the replacement policy is **LRU**.

The initial state of all lines is "invalid". Assume that all memory is initialized to zero.

After that, the following memory accesses occur (suppose 8 bits data):

**Memory accesses:**

- 1: P2\_Read@0x05
- 2: P2\_Read@0x03
- 3: P3\_Read@0x04
- 4: P1\_Write@0x1, 2)

Pregunta 4

Sin responder aún

Puntúa como 0,50

Markar pregunta

En una máquina vectorial tenemos el código que se muestra a continuación:

```
lv v1,vector1;  
add v3,v1,v2;  
xor v4,v1,v2;  
sv vector3,v4:  
lv v2,vector2;
```

Suponiendo que tenemos vectores de 208 elementos y que las cargas y almacenamiento tiene una latencia de 11 ciclos, mientras que las operaciones aritmético-lógicas tienen una latencia de 2 ciclos. ¿Cuántos ciclos se tardaría en ejecutar el código en una máquina vectorial sin encadenamiento de operaciones?

Respuesta:

Sea un computador superescalar capaz de decodificar 4inst/c, emitir 4inst/c, escribir 4inst/c en los registros correspondientes y retirar 4 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada pero **no** contiene una unidad de adelantamiento. Teniendo el siguiente código:

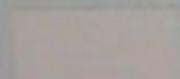
```
addi r3,r1,#800  
addi r1,r1,#8  
addi r2,r2,#8  
ld f0,coef  
ld f2,-8(r1)  
ld f4,0(r1)  
muld f8,f2,f0  
addd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 2 FP mul/div (8c), 2 FP add (4c), 2 ALU int (3c) y 2 load/store (5c).

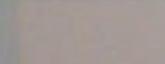
¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?



¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?



¿Cuántos ciclos tarda el código?



nta 4

sponder

a como

arcar

unta

En una máquina vectorial tenemos el código que se muestra a continuación:

```
lv v1,vector1;  
add v3,v1,v2;  
xor v4,v1,v2;  
sv vector3,v4:  
lv v2,vector2;
```

Suponiendo que tenemos vectores de 196 elementos y que las cargas y almacenamiento tiene una latencia de 11 ciclos, mientras que las operaciones aritmético-lógicas tienen una latencia de 3 ciclos. ¿Cuántos ciclos se tardaría en ejecutar el código en una máquina vectorial sin encadenamiento de operaciones?

Respuesta:

Dada una red multietapa tipo **Delta** con 785 entradas y 11062 salidas, se pide:

¿Cuántos conmutadores hay en la etapa 0 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

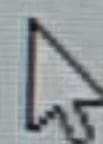
[ENGLISH]

Given a **Delta** multi-stage network with 785 inputs and 11062 outputs, calculate:

How many switches are there in the stage 0 of the most economical implementation of this topology?

**Note:** Remember that the first stage is labeled as 0, the second stage is labeled as 1, and so on.

Respuesta:



```
addd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```

The following pipelined units are available: 2 FP mul/div (8c), 2 FP add (4c), 2 ALU int (3c) and 2 load/store (5c). In which cycle does the execution stage (EX) of the first instruction **addi r2,r2,#8 finish?**

8

In which cycle does the execution stage (EX) of the instruction **ld f4,0(r1)** finish?

12

How many cycles does the code last?

29

Pregunta 5

Sin responder  
aún

Puntúa como  
1,50

🚩 Marcar  
pregunta

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 2 núcleos de un cierto tipo **T1** y 2 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 18% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 2 cores of a certain **T1** type and 2 cores of another **T2** type. Suppose that **T1** cores are 18% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

Respuesta:

Dada una red multietapa tipo **Delta** con 779 entradas y 16615 salidas, se pide:  
¿Cuántos conmutadores hay en la etapa 1 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

### [ENGLISH]

Given a **Delta** multi-stage network with 779 inputs and 16615 outputs, calculate:

How many switches are there in the stage 1 of the most economical implementation of this topology?

**Note:** Remember that the first stage is labeled as 0, the second stage is labeled as 1, and so on.

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo **T1** y 4 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 64% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 4 cores of a certain **T1** type and 4 cores of another **T2** type. Suppose that **T1** cores are 64% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overhead time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

Se tiene un cierto multiprocesador de **4 núcleos** conectados a un bus común que tiene 64GiB de memoria principal e implementa el protocolo **MESI** para mantener la coherencia entre sus cachés. El sistema de caché tiene **2 bytes por línea o bloque de caché**. El tamaño de cada caché es de tan sólo **2 líneas** y la política de reemplazo es LRU.

El estado inicial de TODAS las líneas es "inválido". Suponga que toda la memoria está inicializada a cero.

Suponga que se producen los siguientes accesos a memoria (**suponga datos de 8 bits**):

**Accesos a memoria:**

- 1: P1\_Read@0x01
- 2: P1\_Read@0x00
- 3: P2\_Read@0x03
- 4: P3\_Read@0x02

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

Recuerdo la notación (la misma que la que hemos visto en clase):

- **PJ:Write(@M,N)** significa: el procesador J escribe el valor N en la posición de memoria M (dada en hexadecimal)
- **PJ:Read@M** significa: el procesador J lee de la dirección M (dada en hexadecimal). - sadasdsad
- **Mem[M]** devuelve el byte en la posición M (en hexadecimal) de memoria.

Dada una red multietapa tipo **Delta** con 737 entradas y 11939 salidas, se pide:  
¿Cuántos conmutadores hay en la etapa 1 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

**[ENGLISH]**

Given a **Delta** multi-stage network with 737 inputs and 11939 outputs, calculate:

How many switches are there in the stage 1 of the most economical implementation of this topology?

**Note:** Remember that the first stage is labeled as 0, the second stage is labeled as 1, and so on.

Respuesta: 100

En una máquina superescalar con circuito de adelantamiento y con emisión no alineada...

- a. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones siempre en el siguiente ciclo de la búsqueda.
- b. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones siempre que exista hueco.
- c. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones si previamente se han ejecutado todas las instrucciones que pasaron en ciclos anteriores a la ventana y exista hueco.
- d. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones si previamente se han ejecutado todas las instrucciones que pasaron en ciclos anteriores a la ventana.

Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada pero **no** contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800  
addi r1,r1,#8  
addi r2,r2,#8  
ld f0,coef  
ld f2,-8(r1)  
ld f4,0(r1)  
muld f8,f2,f0  
addd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (5c), 1 FP add (7c), 3 ALU int (3c) y 3 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?

6

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?

8

¿Cuántos ciclos tarda el código?

22

En una máquina vectorial tenemos el código que se muestra a continuación:

```
lv v1,vector1;  
add v3,v1,v2;  
xor v4,v1,v2;  
sv vector3,v4:  
lv v2,vector2;
```

Suponiendo que tenemos vectores de 196 elementos y que las cargas y almacenamiento tiene una latencia de 11 ciclos, mientras que las operaciones aritmético-lógicas tienen una latencia de 3 ciclos. ¿Cuántos ciclos se tardaría en ejecutar el código en una máquina vectorial sin encadenamiento de operaciones?

Respuesta:

A superscalar computer is capable of decoding 4inst/c, issuing 4inst/c, writing 4inst/c into the corresponding registers, and completing 4 inst/c. It has a reorder buffer with a sufficient number of entries, the issue can be out of order and contains a forwarding unit. Having the following code:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

The following pipelined units are available: 2 FP mul/div (8c), 2 FP add (4c), 2 ALU int (3c) and 2 load/store (5c).

In which cycle does the execution stage (EX) of the first instruction **addi r2,r2,#8** finish?

La ejecución de las instrucciones en una máquina superescalar con buffer de reorden o buffer de renombrado....

- a. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siguen un orden ascendente o descendente dependiendo de los riesgos WAR y WAW del programa almacenado en memoria.
- b. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siempre siguen un orden ascendente conforme al orden del programa almacenado en memoria.
- c. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siguen un orden ascendente o descendente dependiendo de los riesgos RAW del programa almacenado en memoria.
- d. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siempre siguen un orden descendente conforme al orden del programa almacenado en memoria.

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo **T1** y 4 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 64% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 4 cores of a certain **T1** type and 4 cores of another **T2** type. Suppose that **T1** cores are 64% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overhead time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

# INGENIERIA DE LOS COMPUTADORES\_34025

Mis cursos / (34025\_1,2,3,40,2020-21) / Calificaciones - Marks / Examen final



Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 2 núcleos de un cierto tipo **T1** y 2 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 47% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en OpenMP y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la ganancia paralela en velocidad (speed-up) usando al menos 2 decimales.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 2 cores of a certain **T1** type and 2 cores of another **T2** type. Suppose that **T1** cores are 47% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel speed-up gain using at least 2 decimals.

Respuestas

- b) ¿Qué fracción del problema total debe ejecutar cualquiera de los núcleos más rápidos?

[ENGLISH]

b) What fraction of the total problem should any of the faster cores run?

Respuestas

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo **T1** y 4 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 38% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?

9

¿Cuántos ciclos tarda el código?

27

En una máquina superescalar con buffer de reorden o buffer de renombrado....

- a. La etapa WB sigue un orden ascendente o descendiente en ciclos dependiendo de los riesgos RAW
- b. La etapa WB siempre sigue un orden descendiente en ciclos.
- c. La etapa WB sigue un orden ascendente o descendiente en ciclos dependiendo de los riesgos WAR y WAW
- d. La etapa WB siempre sigue un orden ascendente en ciclos.

Sea un computador superescalar capaz de decodificar 4inst/c, emitir 4inst/c, escribir 4inst/c en los registros correspondientes y retirar 4 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada pero **no** contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 2 FP mul/div (4c), 2 FP add (2c), 2 ALU int (1) y 2 load/store (2).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción **addi r2,r2,#8**?

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción **ld f4,0(r1)**?

Pregunta 8

Sin responder  
aún

Puntúa como  
1,25

¶ Marcar  
pregunta

Se tiene un cierto multiprocesador de **4 núcleos** conectados a un bus común que tiene 64GiB de memoria principal e implementa el protocolo **MESI** para mantener la coherencia entre sus cachés. El sistema de caché tiene **2 bytes por línea o bloque de caché**. El tamaño de cada caché es de tan sólo **2 líneas** y la política de reemplazo es LRU.

El estado inicial de TODAS las líneas es "inválido". Suponga que toda la memoria está inicializada a cero.

Suponga que se producen los siguientes accesos a memoria (**suponga datos de 8 bits**):

**Accesos a memoria:**

- 1: P2\_Read@0x05
- 2: P2\_Read@0x03
- 3: P3\_Read@0x04
- 4: P1\_Write@(0x1, 3)



Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

Recuerdo la notación (la misma que la que hemos visto en clase):

- PJ:Write(@M,N) significa: el procesador J escribe el valor N en la posición de memoria M (dada en hexadecimal)
- PJ:Read@M significa: el procesador J lee de la dirección M (dada en hexadecimal). - sadasdsad
- Mem[M] devuelve el byte en la posición M (en hexadecimal) de memoria.

**[ENGLISH]**

You have a **4-core** multiprocessor connected to a common bus that has 64GiB of main memory and implements the **MESI protocol** to maintain consistency between its caches. The cache system has **2 bytes per line (and block) of cache**. The size of each cache is only **2 lines** and the replacement policy is LRU.

The initial state of ALL lines is "invalid". Assume that all memory is initialized to zero.  
After that, the following memory accesses occur (**suppose 8 bits data**):

**Memory accesses:**

- 1: P2\_Read@0x05
- 2: P2\_Read@0x03
- 3: P3\_Read@0x04
- 4: P1\_Write@(0x1, 3)

Sea un computador superescalar capaz de decodificar 4inst/c, emitir 4inst/c, escribir 4inst/c en los registros correspondientes y retirar 4 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada pero **no** contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800  
addi r1,r1,#8  
addi r2,r2,#8  
ld f0,coef  
ld f2,-8(r1)  
ld f4,0(r1)  
muld f8,f2,f0  
addd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 2 FP mul/div (8c), 2 FP add (4c), 2 ALU int (3c) y 2 load/store (5c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?

¿Cuántos ciclos tarda el código?

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo **T1** y 4 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 43% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 4 cores of a certain **T1** type and 4 cores of another **T2** type. Suppose that **T1** cores are 43% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

Respuesta:

- b) ¿Qué fracción del problema total debe ejecutar cualquiera de los núcleos más rápidos?

[ENGLISH]

- b) What fraction of the total problem should any of the faster cores run?

Respuesta:

¿Cuántos ciclos tarda el código?

35

En una máquina superescalar con buffer de reorden o buffer de renombrado....

- a. La etapa WB siempre sigue un orden ascendente en ciclos.
- b. La etapa WB sigue un orden ascendente o descendiente en ciclos dependiendo de los riesgos RAW
- c. La etapa WB siempre sigue un orden descendiente en ciclos.
- d. La etapa WB sigue un orden ascendente o descendiente en ciclos dependiendo de los riesgos WAR y WAW

Sigu

Dada una red multietapa tipo **Delta** con 737 entradas y 11939 salidas, se pide:

¿Cuántos conmutadores hay en la etapa 1 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

[ENGLISH]

Given a **Delta** multi-stage network with 737 inputs and 11939 outputs, calculate:

How many switches are there in the stage 1 of the most economical implementation of this topology?

**Note:** Remember that the first stage is labeled as 0, the second stage is labeled as 1, and so on.

Respuesta:

Pregunta **4**

Sin responder  
aún

Puntúa como  
0,50

► Marcar  
pregunta

En una máquina vectorial tenemos el código que se muestra a continuación:

```
lv v1,vector1;  
add v3,v1,v2;  
xor v4,v1,v2;  
sv vector3,v4:  
lv v2,vector2;
```

Suponiendo que tenemos vectores de 208 elementos y que las cargas y almacenamiento tiene una latencia de 11 ciclos, mientras que las operaciones aritmético-lógicas tienen una latencia de 2 ciclos. ¿Cuántos ciclos se tardaría en ejecutar el código en una máquina vectorial sin encadenamiento de operaciones?

Respuesta:

En una máquina superescalar con buffer de reorden....

- a. En el ROB se almacenan de forma temporal los valores que finalmente se guardarán en el banco de registros.
- b. En el ROB se almacenan de forma temporal los valores que volverán a pasar a la etapa EX pero que no se almacenarán en la etapa WB
- c. En el ROB se renombran varias veces los registros del banco de registros.
- d. En el ROB se almacena una copia exacta del banco de registros.

[Quitar mi elección](#)

Accesos a memoria:  
1: P2\_Write@(0x02,7)  
2: P1\_Write@(0x03,9)  
3: P4\_Read@0x02  
4: P3\_Read@0x1

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

Recuerdo la notación (la misma que la que hemos visto en clase):

- PJ:Write(@M,N) significa: el procesador J escribe el valor N en la posición de memoria M (dada en hexadecimal)
- PJ:Read@M significa: el procesador J lee de la dirección M (dada en hexadecimal). - sadasdsad
- Mem[M] devuelve el byte en la posición M (en hexadecimal) de memoria.

[ENGLISH]

You have a 4-core multiprocessor connected to a common bus that has 64GiB of main memory and implements the MESI protocol to maintain consistency between its caches. The cache system has 2 bytes per line (an

The initial state of ALL lines is "invalid". Assume that all memory is initialized to zero.

After that, the following memory accesses occur (suppose 8 bits data):

Memory accesses:

-----  
1: P2\_Write@(0x02,7)  
2: P1\_Write@(0x03,9)  
3: P4\_Read@0x02  
4: P3\_Read@0x1

Select which of the following items matches the state of the caches and/or main memory at the end of this sequence.

The notation is the same as we have seen in class:

- PJ:Write(@M,N) means: processor J writes the value N to memory location M (given in hexadecimal).
- PJ:Read@M means: processor J reads from address M (given in hexadecimal).
- Mem[M] returns the byte at memory location M (in hexadecimal).

Ayuda/Help: Estado inicial / Initial state:

Línea Line	Cache Proc 1		Cache Proc 2		Cache Proc 3		Cache Proc 4	
	Datos Data	Estado State	Datos Data	Estado State	Datos Data	Estado State	Datos Data	Estado State
0	?	I	?	I	?	I	?	I
1	?	I	?	I	?	I	?	I

- a. • Ninguna es correcta  
• None of them is correct
- b. • Línea 0 de P2 en estado I, Mem[0x2] = 9, Línea 0 de P3 en estado E;  
• Line 0 of P2 in state I, Mem[0x2] = 9, Line 0 of P3 in state E;
- c. • Línea 0 de P3 en estado E, Mem[0x2] = 7, Línea 0 de P2 en estado I.  
• Line 0 of P3 in state E, Mem[0x2] = 7, Line 0 of P2 in state I.
- d. • Línea 0 de P1 y P2 en estado M, Mem[0x3] = 9  
• Line 0 of P1 and P2 in state M, Mem[0x3] = 9

Pregunta 1

Respuesta  
guardada

Puntúa como  
2,50

¶ Marcar  
pregunta

Sea un computador superescalar capaz de decodificar 1 inst/c, emitir 1 inst/c, escribir 1 inst/c en los registros correspondientes y retirar 1 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (5c), 1 FP add (7c), 3 ALU int (3c) y 3 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción **addi r2,r2,#8**?

25

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción **ld f4,0(r1)**?

20

¿Cuántos ciclos tarda el código?

Pregunta **2**

Sin responder  
aún

Puntúa como  
0,50

Marca pregunta

La ejecución de las instrucciones en una máquina superescalar con buffer de reorden o buffer de renombrado....

- a. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siguen un orden ascendente o descendente dependiendo de los riesgos WAR y WAW del programa almacenado en memoria.
- b. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siempre siguen un orden ascendente conforme al orden del programa almacenado en memoria.
- c. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siguen un orden ascendente o descendente dependiendo de los riesgos RAW del programa almacenado en memoria.
- d. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siempre siguen un orden descendente conforme al orden del programa almacenado en memoria.

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo **T1** y 4 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 64% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 4 cores of a certain **T1** type and 4 cores of another **T2** type. Suppose that **T1** cores are 64% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

Respuesta: 1,92

Dada una red multietapa tipo **Delta** con 785 entradas y 11062 salidas, se pide:  
¿Cuántos conmutadores hay en la etapa 0 de la implementación más económica de e

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así su

**[ENGLISH]**

Given a **Delta** multi-stage network with 785 inputs and 11062 outputs, calculate:  
How many switches are there in the stage 0 of the most economical implementation of

**Note:** Remember that the first stage is labeled as 0, the second stage is labeled as 1, an

Respuesta:

En una máquina vectorial tenemos el código que se muestra a continuación:

```
lv v1,vector1;  
add v3,v1,v2;  
xor v4,v1,v2;  
sv vector3,v4:  
lv v2,vector2;
```

Suponiendo que tenemos vectores de 170 elementos y que las cargas y almacenamiento tiene una latencia de 10 ciclos, mientras que las operaciones aritmético-lógicas tienen una latencia de 3 ciclos. ¿Cuántos ciclos se tardaría en ejecutar el código en una máquina vectorial sin encadenamiento de operaciones?

Respuesta:

Dada una red multietapa tipo **Delta** con 797 entradas y 10038 salidas, se pide:  
¿Cuántos commutadores hay en la etapa 3 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

**[ENGLISH]**

Given a **Delta** multi-stage network with 797 inputs and 10038 outputs, calculate:  
How many switches are there in the stage 3 of the most economical implementation of this topology?

**Note:** Remember that the first stage is labeled as 0, the second stage is labeled as 1, and so on.

Respuesta:

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 5 núcleos de un cierto tipo **T1** y 5 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 80% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 5 cores of a certain **T1** type and 5 cores of another **T2** type. Suppose that **T1** cores are 80% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

Respuesta:

5  
a  
omo

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo **T1** y 4 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 43% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 4 cores of a certain **T1** type and 4 cores of another **T2** type. Suppose that **T1** cores are 43% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

Respuesta: 1,50

- b) ¿Qué fracción del problema total debe ejecutar cualquiera de los núcleos más rápidos?

[ENGLISH]

- b) What fraction of the total problem should any of the faster cores run?

Respuesta:



En una máquina superescalar con circuito de adelantamiento y con emisión no alineada...

- a. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones si previamente se han ejecutado todas las instrucciones que pasaron en ciclos ventana y exista hueco.
- b. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones siempre en el siguiente ciclo de la búsqueda.
- c. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones siempre que exista hueco.
- d. Las instrucciones pasan de la cola de instrucciones a la ventana de instrucciones si previamente se han ejecutado todas las instrucciones que pasaron en ciclos ventana.

1

?

|

?

|

?

- a. • Líneas 0 y 1 de P1 en estado E, línea 0 de P2 y P3 en estado S, memoria actualizada.
  - Lines 0 and 1 of P1 in state E, line 0 of P2 and P3 in state S, memory updated.
- b. • Ninguna es correcta.
  - Non of them is correct.
- c. • Líneas 0 y 1 de P1 en estado E, línea 0 de P2 y P3 en estado S, memoria no actualizada.
  - Lines 0 and 1 of P1 in state E, line 0 of P2 and P3 in state S, memory not updated.
- d. • Línea 0 de P1 en estado E, línea 0 de P2 y P3 en estado S, memoria actualizada.
  - Line 0 of P1 in state E, line 0 of P2 and P3 in state S, memory updated.

Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (5c), 1 FP add (7c), 3 ALU int (3c) y 3 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción **addi r2,r2,#8?**

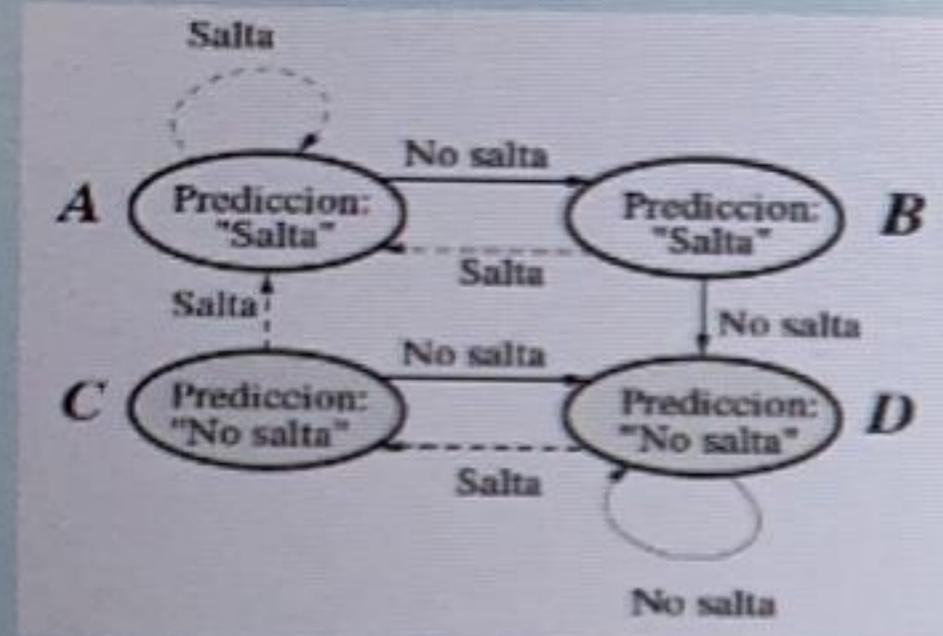
Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (5c), 1 FP add (7c), 3 ALU int (3c) y 3 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción **addi r2,r2,#8?**

Un máquina superescalar dispone de un predictor de saltos dinámico que utiliza BTB de 4 entradas y 2 bits de predicción como el que se muestra en la figura.



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de r1=-11 y r5=1:

```

0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x006
0x005 addi r6,r5,#1
0x006 subi r6, r5, #1
0x007 beqz r6, 0x001
0x008 subd f2,f1, f3
  
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x006	A
0x007	0x001	A

Se tiene un cierto multiprocesador de **4 núcleos** conectados a un bus común que tiene 64GiB de memoria principal e implementa el protocolo **MESI** para mantener la coherencia entre sus cachés. El sistema de caché tiene **2 bytes por línea o bloque de caché**. El tamaño de cada caché es de tan sólo **2 líneas** y la política de reemplazo es LRU.

El estado inicial de TODAS las líneas es "inválido". Suponga que toda la memoria está inicializada a cero. Suponga que se producen los siguientes accesos a memoria (**suponga datos de 8 bits**):

**Accesos a memoria:**

- 1: P1\_Write@(0x01,7)
- 2: P2\_Read@0x00
- 3: P3\_Read@0x03
- 4: P4\_Write@(0x4, 9)

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

Recuerdo la notación (la misma que la que hemos visto en clase):

- **PJ:Write(@M,N)** significa: el procesador J escribe el valor N en la posición de memoria M (dada en hexadecimal)
- **PJ:Read@M** significa: el procesador J lee de la dirección M (dada en hexadecimal). - sadasdsad

Línea Line	Datos Data	Estado State	Datos Data	Estado State	Datos Data	Estado State
0	?	I	?	I	?	
1	?	I	?	I	?	

- a. • Líneas 0 de P1y P2 en estado E, Línea 0 de P3 en estado E, Línea 0 de P4 en estado M.  
 • Lines 0 of P1and P2 in state E, Line 0 of P3 in state E, Line 0 of P4 in state M.
- b. • Líneas 0 de P1y P2 en estado S, Línea 0 de P3 en estado E, Línea 0 de P4 en estado M.  
 • Lines 0 of P1and P2 in state S, Line 0 of P3 in state E, Line 0 of P4 in state M.
- c. • Línea 0 de P1 en estado M, Línea 0 de P3 en estado E, Mem[0x1]=0x7  
 • Line 0 of P1 in M-state, Line 0 of P3 in E-state, Mem[0x1]=0x7
- d. • Línea 0 de P1 en estado E, Línea 0 de P3 en estado E, Mem[0x1]=0x7  
 • Line 0 of P1 in E-state, Line 0 of P3 in E-state, Mem[0x1]=0x7

La ejecución de las instrucciones en una máquina superescalar con buffer de reorden o buffer de renombrado....

- a. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siguen un orden ascendente o descendente dependiendo de los riesgos WAR y WAW del programa almacenado en memoria.
- b. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siempre siguen un orden ascendente conforme al orden del programa almacenado en memoria.
- c. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siguen un orden ascendente o descendente dependiendo de los riesgos RAW del programa almacenado en memoria.
- d. En la etapa EX, los ciclos cuando se ejecutan las instrucciones siempre siguen un orden descendente conforme al orden del programa almacenado en memoria.

Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800  
addi r1,r1,#8  
addi r2,r2,#8  
ld f0,coef  
ld f2,-8(r1)  
ld f4,0(r1)  
muld f8,f2,f0  
addd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 3 FP mul/div (5c), 3 FP add (7c), 2 ALU int (4c) y 3 load/store (7c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?

Suponga que tiene un procesador multi-núcleo tipo **big.LITTLE** con 4 núcleos de un cierto tipo **T1** y 4 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 45% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando al menos 2 decimales.

	Cache Proc 1		Cache Proc 2		Cache Proc 3		Cache Proc 4	
Línea	Datos	Estado	Datos	Estado	Datos	Estado	Datos	Estado
Line	Data	State	Data	State	Data	State	Data	State
0	?	I	?	I	?	I	?	I
1	?	I	?	I	?	I	?	I

- a. Líneas 0 y 1 de P1 en estado E, línea 0 de P2 y P3 en estado S, memoria actualizada.  
     • Lines 0 and 1 of P1 in state E, line 0 of P2 and P3 in state S, memory updated.
- b. Ninguna es correcta.  
     • Non of them is correct.
- c. Líneas 0 y 1 de P1 en estado E, línea 0 de P2 y P3 en estado S, memoria no actualizada.  
     • Lines 0 and 1 of P1 in state E, line 0 of P2 and P3 in state S, memory not updated.
- d. Línea 0 de P1 en estado E, línea 0 de P2 y P3 en estado S, memoria actualizada.  
     • Line 0 of P1 in state E, line 0 of P2 and P3 in state S, memory updated.

Dada una red multietapa tipo **Delta** con 985 entradas y 12785 salidas, se pide:

¿Cuántos conmutadores hay en la etapa 1 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

[ENGLISH]

Given a **Delta** multi-stage network with 985 inputs and 12785 outputs, calculate:

How many switches are there in the stage 1 of the most economical implementation of this topology?

**Note:** Remember that the first stage is labeled as 0, the second stage is labeled as 1, and so on.

Respuesta:

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo **T1** y 4 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 43% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 4 cores of a certain **T1** type and 4 cores of another **T2** type. Suppose that **T1** cores are 43% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

Respuesta: 1,50



- b) ¿Qué fracción del problema total debe ejecutar cualquiera de los núcleos más rápidos?

[ENGLISH]

- b) What fraction of the total problem should any of the faster cores run?

Respuesta:

|

Sea un computador superescalar capaz de decodificar 1 inst/c, emitir 1 inst/c, escribir 1 inst/c en los registros correspondientes y retirar 1 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800  
addi r1,r1,#8  
addi r2,r2,#8  
ld f0,coef  
ld f2,-8(r1)  
ld f4,0(r1)  
muld f8,f2,f0  
adddd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (5c), 1 FP add (7c), 3 ALU int (3c) y 3 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción **addi r2,r2,#8**?

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción **ld f4,0(r1)**?

¿Cuántos ciclos tarda el código?

Sea un computador superescalar capaz de decodificar 1 inst/c, emitir 1 inst/c, escribir 1 inst/c en los registros correspondientes y retirar 1 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800  
addi r1,r1,#8  
addi r2,r2,#8  
ld f0,coef  
ld f2,-8(r1)  
ld f4,0(r1)  
muld f8,f2,f0  
addd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```



Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (5c), 1 FP add (7c), 3 ALU int (3c) y 3 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción **addi r2,r2,#8**?

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción **ld f4,0(r1)**?

¿Cuántos ciclos tarda el código?

## Accesos a memoria:

- 1: P1\_Read@0x01
- 2: P1\_Read@0x00
- 3: P2\_Read@0x03
- 4: P3\_Read@0x02

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

Recuerdo la notación (la misma que la que hemos visto en clase):

- **PJ:Write(@M,N)** significa: el procesador J escribe el valor N en la posición de memoria M (dada en hexadecimal)
- **PJ:Read@M** significa: el procesador J lee de la dirección M (dada en hexadecimal). - sadasdsad
- Mem[M] devuelve el byte en la posición M (en hexadecimal) de memoria.

## [ENGLISH]

You have a **4-core** multiprocessor connected to a common bus that has 64GiB of main memory and implements the **MESI protocol** to maintain consistency between its caches. The cache system has **2 bytes per line (and block) of cache**. The size of each cache is only **2 lines** and the replacement policy is LRU.

The initial state of ALL lines is "invalid". Assume that all memory is initialized to zero.

After that, the following memory accesses occur (**suppose 8 bits data**):

Memory accesses:

- 
- 1: P1\_Write@(0x01,7)
  - 2: P2\_Read@0x00
  - 3: P3\_Read@0x03
  - 4: P4\_Write@(0x4, 9)

Select which of the following items matches the state of the caches and/or main memory at the end of this sequence.

## ¿Cuántos ciclos tarda el código?

Pregunta 2

Sin responder  
aún

Puntúa como  
0,50

Marcar  
pregunta

En una máquina superescalar con buffer de reorden....

- a. En el ROB se almacenan de forma temporal los valores que finalmente se guardarán en el banco de registros.
- b. En el ROB se almacena una copia exacta del banco de registros.
- c. En el ROB se almacenan de forma temporal los valores que volverán a pasar a la etapa EX pero que no se almacenarán en la etapa WB
- d. En el ROB se renombran varias veces los registros del banco de registros.

→ Second objective test (Group 4 - ARA: Wed 9-11h)

Ir a...



Final e

Pregunta **7**

Sin responder  
aún

Puntúa como  
1,25

☒ Marcar  
pregunta

Dada una red multietapa tipo **Delta** con 918 entradas y 11717 salidas, se pide:

¿Cuántos conmutadores hay en la etapa 0 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

**[ENGLISH]**

Given a **Delta** multi-stage network with 918 inputs and 11717 outputs, calculate:

How many switches are there in the stage 0 of the most economical implementation of this topology?

**Note:** Remember that the first stage is labeled as 0, the second stage is labeled as 1, and so on.

Respuesta:

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo **T1** y 4 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 64% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 4 cores of a certain **T1** type and 4 cores of another **T2** type. Suppose that **T1** cores are 64% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overhead time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

- a. • Líneas 0 y 1 de P1 en estado E, línea 0 de P2 y P3 en estado S, memoria actualizada.
  - Lines 0 and 1 of P1 in state E, line 0 of P2 and P3 in state S, memory updated.
- b. • Ninguna es correcta.
  - Non of them is correct.
- c. • Líneas 0 y 1 de P1 en estado E, línea 0 de P2 y P3 en estado S, memoria no actualizada.
  - Lines 0 and 1 of P1 in state E, line 0 of P2 and P3 in state S, memory not updated.
- d. • Línea 0 de P1 en estado E, línea 0 de P2 y P3 en estado S, memoria actualizada.
  - Line 0 of P1 in state E, line 0 of P2 and P3 in state S, memory updated.

En una máquina vectorial tenemos el código que se muestra a continuación:

```
lv v1,vector1;  
add v3,v1,v2;  
xor v4,v1,v2;  
sv vector3,v4:  
lv v2,vector2;
```

Suponiendo que tenemos vectores de 36 elementos y que las cargas y almacenamiento tiene una latencia de 10 ciclos, mientras que las operaciones aritmético-lógicas tienen una latencia de 4 ciclos. ¿Cuántos ciclos se tardaría en ejecutar el código en una máquina vectorial con encadenamiento de operaciones?

Respuesta: 50

Sea un computador superescalar capaz de decodificar 4inst/c, emitir 4inst/c, escribir 4inst/c en los registros correspondientes y retirar 4 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada pero **no** contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 2 FP mul/div (4c), 2 FP add (2c), 2 ALU int (1) y 2 load/store (2).

Dada una red multietapa tipo **Delta** con 985 entradas y 12785 salidas, se pide:  
¿Cuántos comutadores hay en la etapa 1 de la implementación más económica de esta topología?

**Nota:** Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

[ENGLISH]

Given a **Delta** multi-stage network with 985 inputs and 12785 outputs, calculate:

How many switches are there in the stage 1 of the most economical implementation of this topology?

**Note:** Remember that the first stage is labeled as 0, the second stage is labeled as 1, and so on.

Respuesta:

Un computador SMP de 4 procesadores utiliza el protocolo MESI para controlar la coherencia de datos en las cachés. Las líneas de caché y los bloques son de 2 palabras (pueden guardar el contenido de 2 posiciones de memoria). La política de reemplazo de caché es LRU (least recently used), el tamaño de caché de cada procesador es de 2 líneas.

El contenido de las cachés y la memoria principal es el siguiente:

[ENGLISH] A 4-processor SMP computer uses the MESI protocol to control data consistency in the caches. The cache lines and blocks are 2 words long (they can hold the contents of 2 memory locations). The cache replacement policy is LRU (least recently used), the cache size of each processor is 2 lines.

The content of caches and main memory is as follows:

Cachés												
	P1			P2			P3			P4		
Línea	Dir	Dato	Estado									
0	0	0	S	6	5	M	6	6	I	4	9	E
	1	-6		7	6		7	4		5	10	
1	2	2	M	0	0	S	2	4	I	0	0	S
	3	3		1	-6		3	7		1	-6	

Memoria										
Dirección	0	1	2	3	4	5	6	7	8	9
Dato	0	-6	4	7	9	10	-4	3	2	3

En un momento dado, los procesadores ejecutan las siguientes referencias: P1:READ@2, P3:READ@3, P3:WRITE@2(7)

Nota: PJ:WRITE@M(N) significa: el procesador J escribe el valor N en la posición de memoria M, y PJ:READ@M significa: el procesador J lee de la dirección M.

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

[ENGLISH] At any given time, the processors run the following references: P1:READ@2, P3:READ@3, P3:WRITE@2(7)

Note: PJ:WRITE@M(N) means: Processor J writes the value N to memory location M, and PJ:READ@M means: Processor J reads from address M.

Specify which of the following matches the state of the caches and/or main memory at the end of this sequence.

Seleccione una:

- a. Línea 1 en P1 estado I; Línea 1 en P3 en estado M. [ENGLISH] Line 1 in P1 status I; Block 1 in P3 in status M
- b. Dirección @2 en P3 estado S; dirección @3 en P1 y P3 en estado S. [ENGLISH] Address @2 in P3 state S; address @3 in P1 and P3 in state S.
- c. Línea 1 en P1 estado M; Línea 1 en P3 en estado I. [ENGLISH] Line 1 in P1 status M; Line 1 in P3 in status I
- d. Dirección @2 en P3 estado I; dirección @3 en P1 y P3 en estado S. [ENGLISH] Address @2 on P3 state I; address @3 on P1 and P3 on state S.

Un computador SMP de 4 procesadores utiliza el protocolo MESI para controlar la coherencia de datos en las cachés. Las líneas de caché y los bloques son de 2 palabras (pueden guardar el contenido de 2 posiciones de memoria). La política de reemplazo de caché es LRU (least recently used), el tamaño de caché de cada procesador es de 2 líneas.

El contenido de las cachés y la memoria principal es el siguiente:

[ENGLISH] A 4-processor SMP computer uses the MESI protocol to control data consistency in the caches. The cache lines and blocks are 2 words long (they can hold the contents of 2 memory locations). The cache replacement policy is LRU (least recently used), the cache size of each processor is 2 lines.

The content of caches and main memory is as follows:

### Cachés

	P1			P2			P3			P4		
línea	Dir	Dato	Estado									
0	0	0	S	6	5	M	6	6	I	4	9	E
	1	-6		7	6		7	4		5	10	
1	2	2	M	0	0	S	2	4	I	0	0	S
	3	3		1	-6		3	7		1	-6	

### Memoria

Dirección	0	1	2	3	4	5	6	7	8	9
Dato	0	-6	4	7	9	10	-4	3	2	3

En un momento dado, los procesadores ejecutan las siguientes referencias: P2:WRITE@1(8), P3:WRITE@7(27), P4:WRITE@4(9)

Nota: PJ:WRITE@M(N) significa: el procesador J escribe el valor N en la posición M, y PJ:READ@M significa: el procesador J lee de la dirección M.

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

[ENGLISH] At any given time, the processors run the following references: P2:WRITE@1(8), P3:WRITE@7(27), P4:WRITE@4(9)

Note: PJ:WRITE@M(N) means: the J-processor writes the value N to the M-position, and PJ:READ@M means: the J-processor reads from the M-address.

Specify which of the following matches the state of the caches and/or main memory at the end of this sequence.

Seleccione una:

- a. Todos líneas de todas las cachés son inválidas. [ENGLISH] All lines in all caches are invalid.
- b. Línea 0 en P3 estado I; Línea 1 en P1 en estado S.[ENGLISH] Line 0 in P3 state I; Line 1 in P1 in state S
- c. Línea 0 en P3 estado M; Línea 0 en P2 en estado I. [ENGLISH] Line 0 in P3 state M; Line 0 in P2 in state I.
- d. Línea 0 en P3 estado M; Línea 0 en P1 en estado S. [ENGLISH] Line 0 in P3 state M; Line 0 in P1 in state S.

Un máquina superescalar dispone de un predictor de saltos dinámico que utiliza BTB de 4 entradas y 2 bits de predicción como el que se muestra en la figura.



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de r1=0:

```
0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x007
0x005 addi r6,r5,#1
0x006 bnez r6, 0x001
0x007 subd f2,f1, f3
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x007	D
0x006	0x001	D

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración en el código?

¿Cuáles será los bits de predicción de la segunda fila de la BTB en ese caso?

Sea un computador superescalar capaz de decodificar 4inst/c, emitir 4inst/c, escribir 4inst/c en los registros correspondientes y retirar 4 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Un computador SMP de 4 procesadores utiliza el protocolo MESI para controlar la coherencia de datos en las cachés. Las líneas de caché y los bloques son de 2 palabras (pueden guardar el contenido de 2 posiciones de memoria). La política de reemplazo de caché es LRU (least recently used), el tamaño de caché de cada procesador es de 2 líneas.

El contenido de las cachés y la memoria principal es el siguiente:

[ENGLISH] A 4-processor SMP computer uses the MESI protocol to control data consistency in the caches. The cache lines and blocks are 2 words long (they can hold the contents of 2 memory locations). The cache replacement policy is LRU (least recently used), the cache size of each processor is 2 lines.

The content of caches and main memory is as follows:

Cachés

	P1			P2			P3			P4		
línea	Dir	Dato	Estado									
0	0	0	S	6	5	M	6	6	I	4	9	E
	1	-6		7	6		7	4		5	10	
1	2	2	M	0	0	S	2	4	I	0	0	S
	3	3		1	-6		3	7		1	-6	

Memoria

Dirección	0	1	2	3	4	5	6	7	8	9
Dato	0	-6	4	7	9	10	-4	3	2	3

En un momento dado, los procesadores ejecutan las siguientes referencias: P1:READ@2, P3:READ@3, P3:WRITE@2(7)

Nota: PJ:WRITE@M(N) significa: el procesador J escribe el valor N en la posición de memoria M, y PJ:READ@M significa: el procesador J lee de la dirección M.

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

[ENGLISH] At any given time, the processors run the following references: P1:READ@2, P3:READ@3, P3:WRITE@2(7)

Note: PJ:WRITE@M(N) means: Processor J writes the value N to memory location M, and PJ:READ@M means: Processor J reads from address M.

Specify which of the following matches the state of the caches and/or main memory at the end of this sequence.

Seleccione una:

- a. Línea 1 en P1 estado I; Línea 1 en P3 en estado M. [ENGLISH] Line 1 in P1 status I; Block 1 in P3 in status M
- b. Dirección @2 en P3 estado S; dirección @3 en P1 y P3 en estado S. [ENGLISH] Address @2 in P3 state S; address @3 in P1 and P3 in state S.
- c. Línea 1 en P1 estado M; Línea 1 en P3 en estado I. [ENGLISH] Line 1 in P1 status M; Line 1 in P3 in status I
- d. Dirección @2 en P3 estado I; dirección @3 en P1 y P3 en estado S. [ENGLISH] Address @2 on P3 state I; address @3 on P1 and P3 on state S.

La solución es la a

instrucción 1: la memoria no varia

Instrucción 1 → P1:Read@2 (bloque 2)	P1:acceso escritura	Rd Rd / m	M → M																																																																	
P1	P2	P3	P4																																																																	
<table border="1"> <thead> <tr> <th>Línea</th> <th>dir</th> <th>dato</th> <th>Estado</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>S</td> </tr> <tr> <td></td> <td>1</td> <td>-6</td> <td></td> </tr> <tr> <td>1</td> <td>2</td> <td>2</td> <td>M</td> </tr> <tr> <td></td> <td>3</td> <td>3</td> <td></td> </tr> </tbody> </table>	Línea	dir	dato	Estado	0	0	0	S		1	-6		1	2	2	M		3	3		<table border="1"> <thead> <tr> <th>dir</th> <th>dato</th> <th>Estado</th> </tr> </thead> <tbody> <tr> <td>6</td> <td>5</td> <td>M</td> </tr> <tr> <td>7</td> <td>6</td> <td></td> </tr> <tr> <td>0</td> <td>6</td> <td>S</td> </tr> <tr> <td>1</td> <td>-6</td> <td></td> </tr> </tbody> </table>	dir	dato	Estado	6	5	M	7	6		0	6	S	1	-6		<table border="1"> <thead> <tr> <th>dir</th> <th>dato</th> <th>Estado</th> </tr> </thead> <tbody> <tr> <td>6</td> <td>6</td> <td>I</td> </tr> <tr> <td>7</td> <td>4</td> <td></td> </tr> <tr> <td>2</td> <td>4</td> <td>I</td> </tr> <tr> <td>3</td> <td>7</td> <td></td> </tr> </tbody> </table>	dir	dato	Estado	6	6	I	7	4		2	4	I	3	7		<table border="1"> <thead> <tr> <th>dir</th> <th>dato</th> <th>Estado</th> </tr> </thead> <tbody> <tr> <td>4</td> <td>9</td> <td>E</td> </tr> <tr> <td>5</td> <td>10</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>-6</td> <td>S</td> </tr> </tbody> </table>	dir	dato	Estado	4	9	E	5	10		0	0		1	-6	S
Línea	dir	dato	Estado																																																																	
0	0	0	S																																																																	
	1	-6																																																																		
1	2	2	M																																																																	
	3	3																																																																		
dir	dato	Estado																																																																		
6	5	M																																																																		
7	6																																																																			
0	6	S																																																																		
1	-6																																																																			
dir	dato	Estado																																																																		
6	6	I																																																																		
7	4																																																																			
2	4	I																																																																		
3	7																																																																			
dir	dato	Estado																																																																		
4	9	E																																																																		
5	10																																																																			
0	0																																																																			
1	-6	S																																																																		

Instrucción 2

Ins 2 → P3: Read @ 3

P3: Fallo lectura (I) Pr Rd/BusRd(S) I → S

P1: BusRd/Flush M → S

P1

0	0	0	S
1	-6		
2	2		
3	3	3	S

P2

...

P3

0	6	6	I
1	7	4	
2	4		
3	7		S

memoria	...	2	3
dato	...	2	3

Instrucción 3:

Ins 3 → P3: write @ 2 (7)

P3: Autoescritura (S) PrWr/BusRdX S → M

P1: BusRdX/Flush S → I

0	0	0	S
1	-6		
2	2		
3	3	3	I

P2

...

0	6	6	I
1	7	4	
2	4		M
3	7		

memoria	...	2	3
dato	...	2	3

Pregunta 8  
Sin responder aún  
Puntúa como 2,00  
 Marcar pregunta

Un máquina superescalar dispone de un predictor de saltos dinámico que utiliza BTB de 4 entradas y 2 bits de predicción como el que se muestra en la figura.



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de r1=-11 y r5=1:

```

0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x006
0x005 addi r6,r5,#1
0x006 subi r6, r5, #1
0x007 beqz r6, 0x001
0x008 subd f2,f1, f3
  
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BA	Bits
0x004	0x006	B
0x007	0x001	B

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración en el código?

01

¿Cuáles será los bits de predicción de la segunda fila de la BTB en ese caso?

00

Pregunta 7  
Sin responder aún  
Puntúa como 3,00  
 Marcar pregunta

Sea un computador superescalar capaz de decodificar 4 inst/c, emitir 4 inst/c, escribir 4 inst/c en los registros correspondientes y retirar 4 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada pero **no** contiene una unidad de adelantamiento. Teniendo el siguiente código:

```

addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
  
```

Se dispone de las siguientes unidades segmentadas: 3 FP mul/div (2c), 3 FP add (2c), 3 ALU int (2c) y 3 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción **addi r2,r2,#8**?

6

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción **ld f4,0(r1)**?

11

¿Cuántos ciclos tarda el código?

22

Pregunta 7  
Sin responder aún  
Puntúa como 3,00  
 Marcar pregunta

Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```

addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
  
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (4c), 1 FP add (2c), 3 ALU int (1c) y 1 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción **addi r2,r2,#8**?

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción **ld f4,0(r1)**?

¿Cuántos ciclos tarda el código?

Pregunta 2  
Sin responder aún  
Puntúa como 1,00  
▼ Marcar pregunta

El 18,7% de una cierta aplicación no se puede parallelizar. Si el resto de condiciones y parámetros son ideales, ¿cuál es el valor de la **eficiencia paralela** si se ejecuta el programa usando 47 unidades de cómputo paralelo?

**Nota:** Introduzca **3 decimales** en el resultado. **Recuerde que la eficiencia es un número entre 0 y 1.**

[ENGLISH] The 18,7% of a certain application cannot be paralleled. If all other conditions and parameters are ideal, what is the value of the parallel efficiency if the program is run using 47 parallel computing units?

**Note:** Enter 3 decimal places in the result. Remember that the efficiency is a number between 0 and 1.

Respuesta: 0,104

Pregunta 3  
Sin responder aún  
Puntúa como 1,00  
▼ Marcar pregunta

El 27,3% de una aplicación no se puede parallelizar. Si el resto de condiciones y parámetros son ideales, ¿cuál es el valor de la **ganancia paralela** si se ejecuta el programa usando 35 unidades de cómputo en paralelo?

**Nota:** Introduzca **3 decimales** en el resultado.

[ENGLISH] The 27,3% of an application cannot be paralleled. If all other conditions and parameters are ideal, what is the value of the parallel gain if the program is run using 35 parallel units?

**Note:** Enter 3 decimal places in the result.

Respuesta: |

Pregunta 4  
Sin responder aún  
Puntúa como 1,00  
▼ Marcar pregunta

Una red de conexión multietapa tipo delta tiene 62 puertos de entrada y 124 puertos de salida. Los conmutadores tienen todos 4 puertos de entrada, ¿Cuántas etapas de conmutadores necesita?

[ENGLISH] A delta-type multi-stage network connection (MIN) has 62 input ports and 124 output ports. The switches all have 4 input ports, how many switch stages do you need?

Seleccione una:  
 a. 4  
 b. 5  
 c. 3  
 d. 2

Dada una red multietapa tipo Delta con 81 entradas y 625 salidas, y en la que los conmutadores tienen 3 entradas, se pide:  
¿Cuántos conmutadores hay en la etapa 2?

Nota: Recuerde que **la primera etapa es la 0**, la segunda la 1, y así sucesivamente.

[ENGLISH] Given a multi-stage Delta-type network (MIN) with 81 inputs and 625 outputs, and where the switches have 3 inputs:

How many switches are in the 2 stage?

Note: Remember that **the first stage is 0**, the second stage is 1, and so on.

Respuesta: |

Pregunta 1  
Sin responder aún  
Puntúa como 1,00  
▼ Marcar pregunta

Dada una red multietapa tipo Delta con 81 entradas y 625 salidas, y en la que los conmutadores tienen 3 entradas, se pide:

¿Cuántos conmutadores hay en la etapa 2?

Nota: Recuerde que **la primera etapa es la 0**, la segunda la 1, y así sucesivamente.

[ENGLISH] Given a multi-stage Delta-type network (MIN) with 81 inputs and 625 outputs, and where the switches have 3 inputs:

How many switches are in the 2 stage?

Note: Remember that **the first stage is 0**, the second stage is 1, and so on.

Respuesta: |

**Pregunta 7**

Sin responder aún

Puntúa como 3,00

Marcar pregunta

Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (4c), 1 FP add (2c), 3 ALU int (1c) y 1 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?

6

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?

11

¿Cuántos ciclos tarda el código?

22

El 14,0% de una aplicación no se puede parallelizar. Si el resto de condiciones y parámetros son ideales, ¿cuál es el valor de la **ganancia paralela** si se ejecuta el programa usando 45 unidades de cómputo en paralelo?  
**Nota:** Introduzca 3 decimales en el resultado.

[ENGLISH] The 14.0% of an application cannot be paralleled. If all other conditions and parameters are ideal, what is the value of the parallel gain if the program is run using 45 parallel units?

Note: Enter 3 decimal places in the result.

Respuesta:

**Pregunta 2**

Sin responder aún

Puntúa como 1,00

Marcar pregunta

El 18,7% de una cierta aplicación no se puede parallelizar. Si el resto de condiciones y parámetros son ideales, ¿cuál es el valor de la **eficiencia paralela** si se ejecuta el programa usando 47 unidades de cómputo paralelo?

**Nota:** Introduzca 3 decimales en el resultado. Recuerde que la eficiencia es un número entre 0 y 1.

[ENGLISH] The 18,7% of a certain application cannot be paralleled. If all other conditions and parameters are ideal, what is the value of the parallel efficiency if the program is run using 47 parallel computing units?

Note: Enter 3 decimal places in the result. Remember that the efficiency is a number between 0 and 1.

Respuesta:

**Pregunta 2**

Sin responder aún

Puntúa como 1,00

▼ Marcar pregunta

El 11,9% de una cierta aplicación no se puede parallelizar. Si el resto de condiciones y parámetros son ideales, ¿cuál es el valor de la **eficiencia paralela** si se ejecuta el programa usando 31 unidades de cómputo paralelo?

**Nota:** Introduzca **3 decimales** en el resultado. **Recuerde que la eficiencia es un número entre 0 y 1.**

[ENGLISH] The 11,9% of a certain application cannot be paralleled. If all other conditions and parameters are ideal, what is the value of the parallel efficiency if the program is run using 31 parallel computing units?

Note: Enter 3 decimal places in the result. Remember that the efficiency is a number between 0 and 1.

Respuesta:

**Pregunta 3**

Sin responder aún

Puntúa como 1,00

▼ Marcar pregunta

El 38,9% de una aplicación no se puede parallelizar. Si el resto de condiciones y parámetros son ideales, ¿cuál es el valor de la **ganancia paralela** si se ejecuta el programa usando 33 unidades de cómputo en paralelo?

**Nota:** Introduzca **3 decimales** en el resultado.

[ENGLISH] The 38,9% of an application cannot be paralleled. If all other conditions and parameters are ideal, what is the value of the parallel gain if the program is run using 33 parallel units?

Note: Enter 3 decimal places in the result.

Respuesta:

Pregunta 7

Sin responder aún

Puntúa como 3,00

▼ Marcar pregunta

Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800  
addi r1,r1,#8  
addi r2,r2,#8  
ld f0,coef  
ld f2,-8(r1)  
ld f4,0(r1)  
muld f8,f2,f0  
addd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```

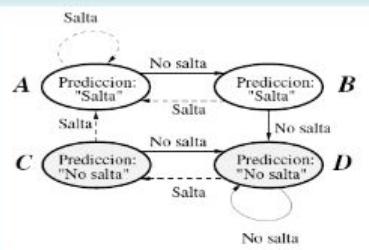
Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (4c), 1 FP add (2c), 3 ALU int (1c) y 1 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?

¿Cuántos ciclos tarda el código?

i como  
rcar  
ita



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de r1=0:

```

0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x007
0x005 addi r6,r5,#1
0x006 bnez r6, 0x001
0x007 subd f2,f1, f3
  
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x007	A
0x006	0x001	C

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración en el código?

01

¿Cuáles será los bits de predicción de la segunda fila de la BTB en ese caso?

00

i como  
rcar  
ita



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de r1=0:

```

0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x007
0x005 addi r6,r5,#1
0x006 bnez r6, 0x001
0x007 subd f2,f1, f3
  
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x007	A
0x006	0x001	C

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración en el código?

01

¿Cuáles será los bits de predicción de la segunda fila de la BTB en ese caso?

00

1

Un computador SMP de 4 procesadores utiliza el protocolo MESI para controlar la coherencia de datos en las cachés. Las líneas de caché y los bloques son de 2 palabras (pueden guardar el contenido de 2 posiciones de memoria). La política de reemplazo de caché es LRU (least recently used), el tamaño de caché de cada procesador es de 2 líneas.

El contenido de las cachés y la memoria principal es el siguiente:

**[ENGLISH]** A 4-processor SMP computer uses the MESI protocol to control data consistency in the caches. The cache lines and blocks are 2 words long (they can hold the contents of 2 memory locations). The cache replacement policy is LRU (least recently used), the cache size of each processor is 2 lines.

The content of caches and main memory is as follows:

#### Cachés

línea	P1			P2			P3			P4		
	Dir	Dato	Estado									
0	0	0	S	6	5	M	6	6	I	4	9	E
	1	-6		7	6		7	4		5	10	
1	2	2	M	0	0	S	2	4	I	0	0	
	3	3		1	-6		3	7		1	-6	S

#### Memoria

Dirección	0	1	2	3	4	5	6	7	8	9		
Dato	0	-6	4	7	9	10	-4	3	2	3		

En un momento dado, los procesadores ejecutan las siguientes referencias: P1:READ@2, P3:READ@3, P3:WRITE@2(7)

**Nota:** PJ:WRITE@M(N) significa: el procesador J escribe el valor N en la posición de memoria M, y PJ:READ@M significa: el procesador J lee de la dirección M.

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

**[ENGLISH]** At any given time, the processors run the following references: P1:READ@2, P3:READ@3, P3:WRITE@2(7)

**Note:** PJ:WRITE@M(N) means: Processor J writes the value N to memory location M, and PJ:READ@M means: Processor J reads from address M.

Specify which of the following matches the state of the caches and/or main memory at the end of this sequence.

Seleccione una:

- a. Línea 1 en P1 estado I; Línea 1 en P3 en estado M. **[ENGLISH]** Line 1 in P1 status I; Block 1 in P3 in status M
- b. Dirección @2 en P3 estado S; dirección @3 en P1 y P3 en estado S. **[ENGLISH]** Address @2 in P3 state S; address @3 in P1 and P3 in state S.
- c. Línea 1 en P1 estado M; Línea 1 en P3 en estado I. **[ENGLISH]** Line 1 in P1 status M; Line 1 in P3 in status I
- d. Dirección @2 en P3 estado I; dirección @3 en P1 y P3 en estado S. **[ENGLISH]** Address @2 on P3 state I; address @3 on P1 and P3 on state S.

2

Un computador SMP de 4 procesadores utiliza el protocolo MESI para controlar la coherencia de datos en las cachés. Las líneas de caché y los bloques son de 2 palabras (pueden guardar el contenido de 2 posiciones de memoria). La política de reemplazo de caché es LRU (least recently used), el tamaño de caché de cada procesador es de 2 líneas.

El contenido de las cachés y la memoria principal es el siguiente:

**[ENGLISH]** A 4-processor SMP computer uses the MESI protocol to control data consistency in the caches. The cache lines and blocks are 2 words long (they can hold the contents of 2 memory locations). The cache replacement policy is LRU (least recently used), the cache size of each processor is 2 lines.

The content of caches and main memory is as follows:

#### Cachés

línea	P1			P2			P3			P4		
	Dir	Dato	Estado									
0	0	0	S	6	5	M	6	6	I	4	9	E
	1	-6		7	6		7	4		5	10	
1	2	2	M	0	0	S	2	4	I	0	0	
	3	3		1	-6		3	7		1	-6	S

#### Memoria

Dirección	0	1	2	3	4	5	6	7	8	9		
Dato	0	-6	4	7	9	10	-4	3	2	3		

En un momento dado, los procesadores ejecutan las siguientes referencias: P2:WRITE@1(8), P3:WRITE@7(27), P4:WRITE@4(9)

**Nota:** PJ:WRITE@M(N) significa: el procesador J escribe el valor N en la posición M, y PJ:READ@M significa: el procesador J lee de la dirección M.

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

**[ENGLISH]** At any given time, the processors run the following references: P2:WRITE@1(8), P3:WRITE@7(27), P4:WRITE@4(9)

**Note:** PJ:WRITE@M(N) means: the J-processor writes the value N to the M-position, and PJ:READ@M means: the J-processor reads from the M-address.

Specify which of the following matches the state of the caches and/or main memory at the end of this sequence.

Seleccione una:

- a. Todos las líneas de todas las cachés son inválidas. **[ENGLISH]** All lines in all caches are invalid.
- b. Línea 0 en P3 estado I; Línea 1 en P1 en estado S. **[ENGLISH]** Line 0 in P3 state I; Line 1 in P1 in state S
- c. Línea 0 en P3 estado M; Línea 0 en P2 en estado I. **[ENGLISH]** Line 0 in P3 state M; Line 0 in P2 in state I.
- d. Línea 0 en P3 estado M; Línea 0 en P1 en estado S. **[ENGLISH]** Line 0 in P3 state M; Line 0 in P1 in state S.

3

Sea un computador superescalar capaz de decodificar 4inst/c, emitir 4inst/c, escribir 4inst/c en los registros correspondientes y retirar 4 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

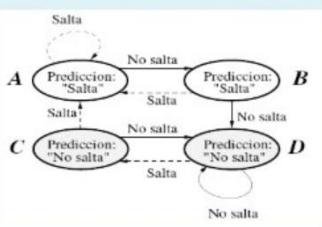
```

addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3

```

4

Un máquina superescalar dispone de un predictor de saltos dinámico que utiliza BTB de 4 entradas y 2 bits de predicción como el que se muestra en la figura.



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de r1=0:

```

0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x007
0x005 addi r6,r5,#1
0x006 bnez r6, 0x001
0x007 subd f2,f1, f3
  
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x007	D
0x006	0x001	D

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración en el código?

¿Cuáles será los bits de predicción de la segunda fila de la BTB en ese caso?

5

Un máquina superescalar dispone de un predictor de saltos dinámico que utiliza BTB de 4 entradas y 2 bits de predicción como el que se muestra en la figura.



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de r1=-11 y r5=1:

```

0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x006
0x005 addi r6,r5,#1
0x006 subi r6, r5, #1
0x007 beqz r6, 0x001
0x008 subd f2,f1, f3
  
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x006	B
0x007	0x001	B

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración en el código?

¿Cuáles será los bits de predicción de la segunda fila de la BTB en ese caso?

6

El 18,7% de una cierta aplicación no se puede parallelizar. Si el resto de condiciones y parámetros son ideales, ¿cuál es el valor de la **eficiencia paralela** si se ejecuta el programa usando 47 unidades de cómputo paralelo?

**Nota:** Introduzca **3 decimales** en el resultado. **Recuerde que la eficiencia es un número entre 0 y 1.**

[ENGLISH] The 18,7% of a certain application cannot be paralleled. If all other conditions and parameters are ideal, what is the value of the parallel efficiency if the program is run using 47 parallel computing units?

Note: Enter 3 decimal places in the result. Remember that the efficiency is a number between 0 and 1.

Respuesta:

El 27,3% de una aplicación no se puede parallelizar. Si el resto de condiciones y parámetros son ideales, ¿cuál es el valor de la **ganancia paralela** si se ejecuta el programa usando 35 unidades de cómputo en paralelo?

**Nota:** Introduzca **3 decimales** en el resultado.

[ENGLISH] The 27,3% of an application cannot be paralleled. If all other conditions and parameters are ideal, what is the value of the parallel gain if the program is run using 35 parallel units?

Note: Enter 3 decimal places in the result.

Respuesta:

7

Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (4c), 1 FP add (2c), 3 ALU int (1c) y 1 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?

¿Cuántos ciclos tarda el código?

8

Sea un computador superescalár capaz de decodificar 4 inst/c, emitir 4 inst/c, escribir 4 inst/c en los registros correspondientes y retirar 4 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada pero **no** contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 3 FP mul/div (2c), 3 FP add (2c), 3 ALU int (2c) y 3 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción **addi r2,r2,#8**?

6

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción **ld f4,0(r1)**?

11

¿Cuántos ciclos tarda el código?

22

9

Una red de conexión multietapa tipo delta tiene 62 puertos de entrada y 124 puertos de salida. Los conmutadores tienen todos 4 puertos de entrada, ¿Cuántas etapas de conmutadores necesita?

[ENGLISH] A delta-type multi-stage network connection (MIN) has 62 input ports and 124 output ports. The switches all have 4 input ports, how many switch stages do you need?

Seleccione una:

- a. 4
- b. 5
- c. 3
- d. 2

10

Dada una red multietapa tipo Delta con 81 entradas y 625 salidas, y en la que los conmutadores tienen 3 entradas, se pide:

¿Cuántos conmutadores hay en la etapa 2?

Nota: Recuerde que **la primera etapa es la 0**, la segunda la 1, y así sucesivamente.

[ENGLISH] Given a multi-stage Delta-type network (MIN) with 81 inputs and 625 outputs, and where the switches have 3 inputs:

How many switches are in the 2 stage?

Note: Remember that **the first stage is 0**, the second stage is 1, and so on.

Respuesta:

11

Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800  
addi r1,r1,#8  
addi r2,r2,#8  
ld f0,coef  
ld f2,-8(r1)  
ld f4,0(r1)  
muld f8,f2,f0  
addd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (4c), 1 FP add (2c), 3 ALU int (1c) y 1 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?

6

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?

11

¿Cuántos ciclos tarda el código?

22

12

El 14,0% de una aplicación no se puede parallelizar. Si el resto de condiciones y parámetros son ideales, ¿cuál es el valor de la **ganancia paralela** si se ejecuta el programa usando 45 unidades de cómputo en paralelo?

**Nota:** Introduzca 3 decimales en el resultado.

13

El 18,7% de una cierta aplicación no se puede parallelizar. Si el resto de condiciones y parámetros son ideales, ¿cuál es el valor de la **eficiencia paralela** si se ejecuta el programa usando 47 unidades de cómputo paralelo?

14

El 11,9% de una cierta aplicación no se puede parallelizar. Si el resto de condiciones y parámetros son ideales, ¿cuál es el valor de la **eficiencia paralela** si se ejecuta el programa usando 31 unidades de cómputo paralelo?

**Nota:** Introduzca 3 decimales en el resultado. Recuerde que la eficiencia es un número entre 0 y 1.

15

El 38,9% de una aplicación no se puede parallelizar. Si el resto de condiciones y parámetros son ideales, ¿cuál es el valor de la **ganancia paralela** si se ejecuta el programa usando 33 unidades de cómputo en paralelo?

**Nota:** Introduzca 3 decimales en el resultado.

16

Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800
addi r1,r1,#8
addi r2,r2,#8
ld f0,coef
ld f2,-8(r1)
ld f4,0(r1)
muld f8,f2,f0
addd f4,f8,f4
sd 0(r2),f4
addi r1,r1,#8
addi r2,r2,#8
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (4c), 1 FP add (2c), 3 ALU int (1c) y 1 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?

¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción `ld f4,0(r1)`?

¿Cuántos ciclos tarda el código?



Suponed que se está ejecutando el siguiente código, siendo el valor inicial de r1=0:

```

0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x007
0x005 addi r6,r5,#1
0x006 bnez r6, 0x001
0x007 subd f2,f1, f3
  
```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x007	A
0x006	0x001	C

¿Cuáles será los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración en el código?

01

¿Cuáles será los bits de predicción de la segunda fila de la BTB en ese caso?

00



Suponed que se está ejecutando el siguiente código, siendo:

```

0x000 addi r3,r1,#80
0x001 addi r1,r1,#8
0x002 addi r2,r2,#8
0x003 addi r5,r1,#3
0x004 beqz r5, 0x007
0x005 addi r6,r5,#1
0x006 bnez r6, 0x001
0x007 subd f2,f1, f3

```

Y que inicialmente la tabla BTB contiene:

Instrucción	BTA	Bits
0x004	0x007	A
0x006	0x001	C

¿Cuáles será los bits de predicción de la primera fila de la tabla BTB en la iteración en el código?

01

¿Cuáles será los bits de predicción de la segunda fila de la tabla BTB en la iteración en el código?

00

Le paso los enunciados que le tocaron (son todos distintos) y las soluciones para que las trabaje.

Para ver qué está mal necesitamos que se nos explique qué es cada cosa, qué está haciendo con cada expresión matemática y porqué. De otra forma, es imposible seguir el hilo de unas "cuentas" que parecen aleatorias. Le vuelvo a recomendar que estudie bien los contenidos y que vuelva a realizar los ejercicios desde cero, explicando bien cada paso. Seguramente llegará al resultado correcto, de otra forma me los podrá enviar más adelante y veremos dónde está el problema.

P5: Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 4 núcleos de un cierto tipo T1 y 4 núcleos de otro tipo T2. Suponga que los núcleos de tipo T1 son un 38% más rápidos que los de tipo T2. Suponga también que tiene una aplicación totalmente paralelizable en OpenMP y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

a) Calcule el valor de la ganancia paralela en velocidad (speed-up) usando al menos 2 decimales. Respuesta correcta 3.45

P6

b) ¿Qué fracción del problema total debe ejecutar cualquiera de los núcleos más rápidos? Respuesta correcta 0.29

P7

Dada una red multietapa tipo Delta con 797 entradas y 10038 salidas, se pide:

¿Cuántos conmutadores hay en la etapa 3 de la implementación más económica de esta topología?

Nota: Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

Respuesta correcta: 1372 (topología: 4\*\*5x7\*\*5)

Nota: Recuerde que la primera etapa se etiqueta como 0, la segunda como 1, y así sucesivamente.

Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el siguiente código:

```
addi r3,r1,#800  
addi r1,r1,#8  
addi r2,r2,#8  
ld f0,coef  
ld f2,-8(r1)  
ld f4,0(r1)  
muld f8,f2,f0  
adddd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```

Se dispone de las siguientes unidades segmentadas: 1 FP mul/div (4c), 1 FP add (2c), 3 ALU int (1c) y 1 load/store (2c).

¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción **addi r2,r2,#8**?

5



¿En qué ciclo finaliza la etapa de ejecución (EX) de la instrucción **ld f4,0(r1)**?

11



¿Cuántos ciclos tarda el código?

22



Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 2 núcleos de un cierto tipo **T1** y 2 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 75% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando al menos 2 decimales.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 2 cores of a certain **T1** type and 2 cores of another **T2** type. Suppose that **T1** cores are 75% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using at least 2 decimals.

Respuesta:

- b) ¿Qué fracción del problema total debe ejecutar cualquiera de los núcleos más rápidos?

[ENGLISH]

- b) What fraction of the total problem should any of the faster cores run?

Respuesta:

Un computador SMP de 4 procesadores utiliza el protocolo MESI para controlar la coherencia de datos en las cachés. Las líneas de caché y los bloques son de 2 palabras (pueden guardar el contenido de 2 posiciones de memoria). La política de reemplazo de caché es LRU (least recently used), el tamaño de caché de cada procesador es de 2 líneas.

El contenido de las cachés y la memoria principal es el siguiente:

[ENGLISH] A 4-processor SMP computer uses the MESI protocol to control data consistency in the caches. The cache lines and blocks are 2 words long (they can hold the contents of 2 memory locations). The cache replacement policy is LRU (least recently used), the cache size of each processor is 2 lines.

The content of caches and main memory is as follows:

Cachés

	P1			P2			P3			P4		
línea	Dir	Dato	Estado									
0	0	0	S	6	5	M	6	6	I	4	9	E
	1	-6		7	6		7	4		5	10	
1	2	2	M	0	0	S	2	4	I	0	0	S
	3	3		1	-6		3	7		1	-6	

Memoria

Dirección	0	1	2	3	4	5	6	7	8	9
Dato	0	-6	4	7	9	10	-4	3	2	3

En un momento dado, los procesadores ejecutan las siguientes referencias: P2:WRITE@1(8), P3:WRITE@7(27), P4:WRITE@4(9)

Nota: PJ:WRITE@M(N) significa: el procesador J escribe el valor N en la posición M, y PJ:READ@M significa: el procesador J lee de la dirección M.

Indique cuál de las siguientes opciones coincide con el estado de las cachés y/o la memoria principal al final de esta secuencia.

[ENGLISH] At any given time, the processors run the following references: P2:WRITE@1(8), P3:WRITE@7(27), P4:WRITE@4(9)

Note: PJ:WRITE@M(N) means: the J-processor writes the value N to the M-position, and PJ:READ@M means: the J-processor reads from the M-address.

Specify which of the following matches the state of the caches and/or main memory at the end of this sequence.

Seleccione una:

- a. Todos líneas de todas las cachés son inválidas. [ENGLISH] All lines in all caches are invalid.
- b. Línea 0 en P3 estado I; Línea 1 en P1 en estado S.[ENGLISH] Line 0 in P3 state I; Line 1 in P1 in state S
- c. Línea 0 en P3 estado M; Línea 0 en P2 en estado I. [ENGLISH] Line 0 in P3 state M; Line 0 in P2 in state I.
- d. Línea 0 en P3 estado M; Línea 0 en P1 en estado S. [ENGLISH] Linea 0 en P3 estado M; Línea 0 en P1 en estado S.

Suponga que tiene un procesador multi-núcleo tipo big.LITTLE con 6 núcleos de un cierto tipo **T1** y 6 núcleos de otro tipo **T2**. Suponga que los núcleos de tipo **T1** son un 76% más rápidos que los de tipo **T2**. Suponga también que tiene una aplicación totalmente paralelizable en *OpenMP* y que usa todos los núcleos para optimizar la velocidad de ejecución. Suponga también que el tiempo de sobrecarga es cero.

- a) Calcule el valor de la **ganancia paralela en velocidad (speed-up)** usando **al menos 2 decimales**.

[ENGLISH]

Suppose you have a big.LITTLE multi-core processor with 6 cores of a certain **T1** type and 6 cores of another **T2** type. Suppose that **T1** cores are 76% faster than **T2** cores. Suppose also that you have a fully parallelizable application using OpenMP and that all cores are used to optimize the execution speed. Suppose also that the overload time is zero.

- a) Calculate the value of the parallel **speed-up** gain using **at least 2 decimals**.

Respuesta:

- b) ¿Qué fracción del problema total debe ejecutar cualquiera de los núcleos más rápidos?

[ENGLISH]

- b) What fraction of the total problem should any of the faster cores run?

## IC Unidad 1: Problemas arquitecturas vectoriales

### Ejercicio 1

Entrelazado inferior con 8 módulos de memoria  $\rightarrow 2^3$  bloques  $\rightarrow$  3 bits para direccionar bloques:  $m=3$ .

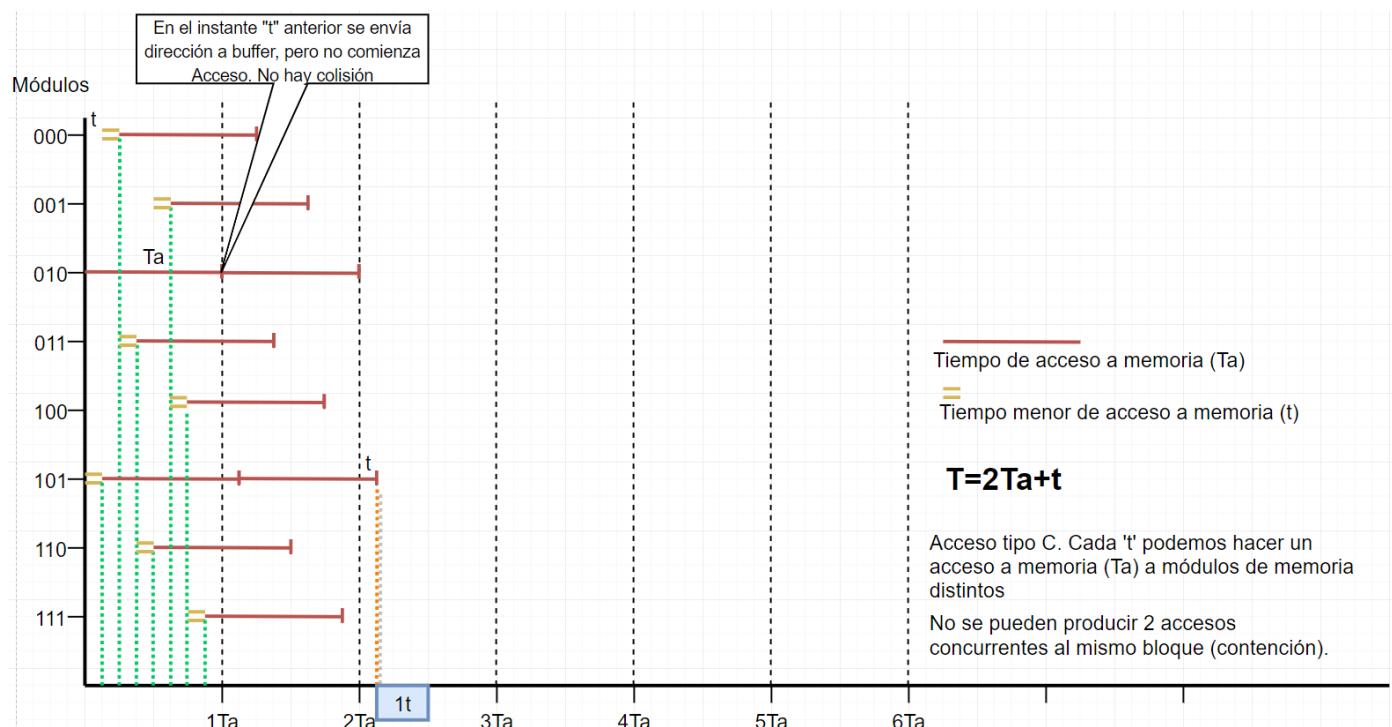
Resto de bits para direccionar posiciones de memoria dentro de los módulos:  $n=\text{resto de bits}$

Dirección inicio: 01C2h  $\rightarrow$  0000 0001 1100 0010

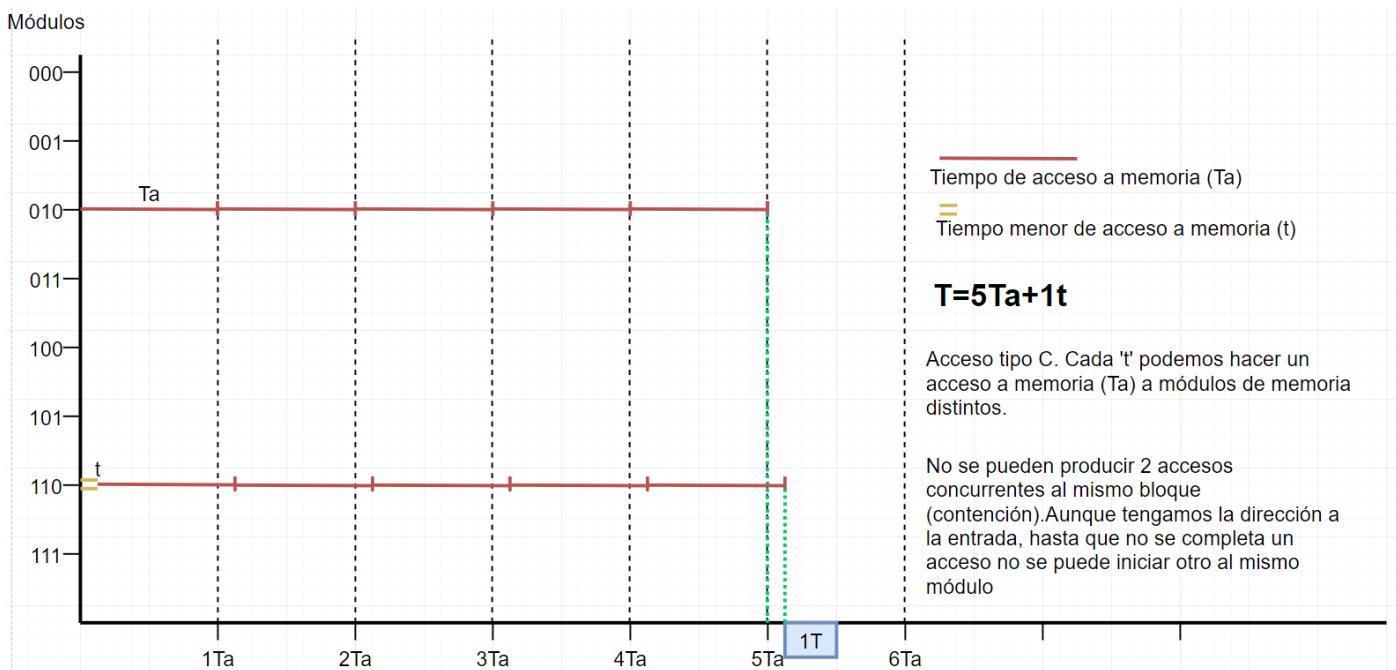
Almacenamiento del vector V en memoria a partir de la dirección de inicio con stride 3 (elementos del vector almacenados cada 3 posiciones, 2 huecos entre cada componente):

DIRECCIONES	MÓDULOS MEMORIA							
	M0 000	M1 001	M2 010	M3 011	M4 100	M5 101	M6 110	M7 111
0000 0001 1100 0				V0			V1	
0000 0001 1100 1	V2			V3			V4	
0000 0001 1101 0		V5			V6			V7
0000 0001 1101 1			V8			V9		

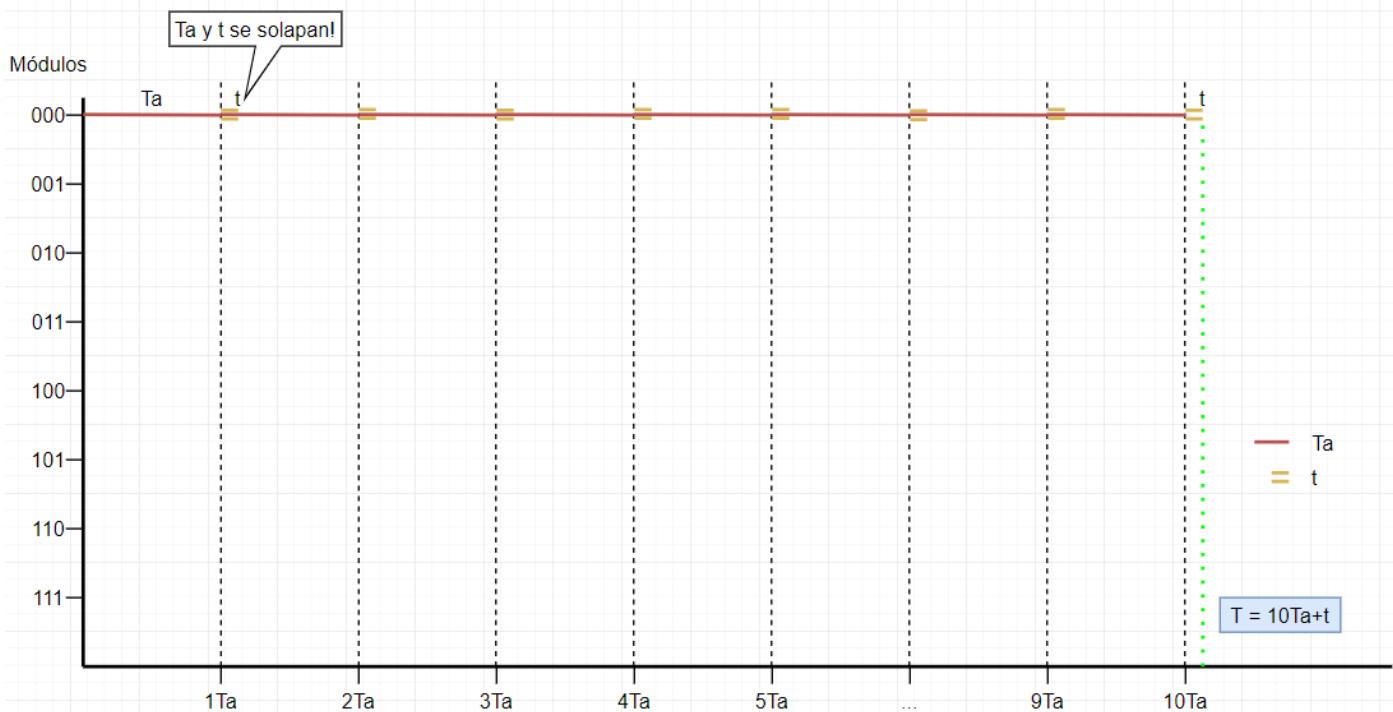
- Apartado a) Acceso simultáneo:  $T = 4Ta + 2t$
- Apartado b) Acceso concurrente:



- Apartado b), cómo sería con stride 4 (3 posiciones de memoria entre elementos del vector)



- Apartado c)



## Entrelazado superior

Posiciones consecutivas de memoria estarán ubicadas dentro del mismo módulo. Si la dirección de comienzo de almacenamiento del vector es: 01C2h = 0000 0001 1100 0010 Los 3 bits de mayor peso marcarán el módulo de memoria dónde está almacenada la 1a posición del vector. En este caso 000, por tanto el módulo 0, en la dirección 0 0001 1100 0010. A partir de esa, siguiendo el stride marcado por el enunciado, estarán almacenadas las posiciones restantes del vector.

## Acceso tipo S

Recuperamos posiciones de memoria de todos los bloques a la vez en cada acceso ( $T_a$ ), pero de cada bloque solo se puede recuperar una palabra por cada acceso. Tras cada acceso tardamos un tiempo menor ( $t$ ) en poner el dato recuperado en el bus de datos y completar la recuperación.

## Ejercicio 2

Latencias:

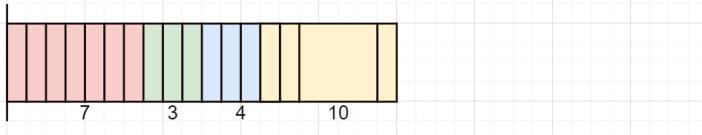
- Cargas: 7 ciclos
- Sumas: 3 ciclos
- Desplazamientos: 4 ciclos

Código ejemplo:

- $V0 = \text{load}(\text{mem})$
- $V2 = V0 + V1$
- $V3 = V2 \ll A1$

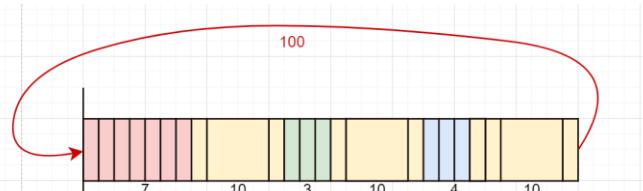
a)

$$\text{TCV} = \text{TLI}(\text{carga}) + \text{TLI}(\text{suma}) + \text{TLI}(\text{despl}) + \text{LongVector} = 7 + 3 + 4 + 10 = 24 \text{ ciclos}$$



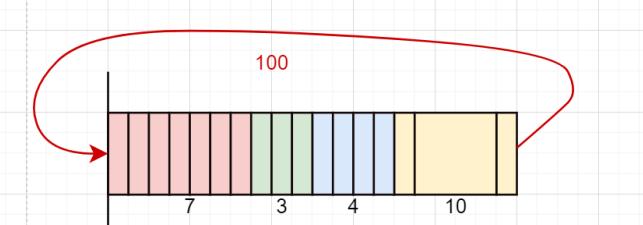
b) Nos piden la ganancia por lo que habrá que estimar el número de ciclos empleado en el caso sin encadenamiento frente al caso en el que sí disponemos de un cauce vectorial con encadenamiento. Se indica que el código anterior están dentro de un bucle que se repetirá en 100 ocasiones por lo que a partir de los TCV iniciales (como el calculado en 2.a ) obtendremos los de este caso.

TCV para el cauce sin encadenamiento iterando 100 veces:



$$\text{TCV} = [\text{TLI}(\text{carga}) + \text{TLI}(\text{suma}) + \text{TLI}(\text{despl}) + (\text{LongVector} * \text{NºInstrucciones})] * \text{Iteraciones} = [7 + 3 + 4 + (10 * 3)] * 100 = 4400 \text{ ciclos}$$

TCV para el cauce con encadenamiento iterando 100 veces:



$$\text{TCV} = (\text{TLI}(\text{carga}) + \text{TLI}(\text{suma}) + \text{TLI}(\text{despl}) + \text{LongVector} * \text{Iteraciones}) = (7 + 3 + 4 + 10) * 100 = 2400 \text{ ciclos}$$

Ganancia:

$$G = \frac{T_0}{T_{enc}} = \frac{4400}{2400} = 1.83$$

## IC Unidad 2: Superescalares - Problemas de emisión fuera de orden

Dado del siguiente código ejemplo:

```
lw r1,0(r2)
add r2,r1,r3
mult r3,r1,r2
sub r4,r1,r2
add r4,r1,r2
div r4,r5,r6
```

Estimar los momentos de emisión e intervalos de ejecución para cada una de las instrucciones sobre arquitecturas que cuenten con una combinación de los siguientes casos:

- Ventana centralizada y Estaciones de reserva
- Emisión ordenada y desordenada
- Unidades funcionales
  - A. 1 lw/sw, 1 add/sub, 1 mult/div
  - B. 1 lw/sw, 2 add/sub, 2 mult/div

Con latencias: lw/sw (5 ciclos), add/sub (2 ciclos), mult/div (5 ciclos)

### I. Set A de UF - Ventana centralizada - Emisión ordenada.

Nº Instr	CodOP	ISS	EXE	Comentarios
1	lw	1	2 – 6	
2	add	7	8 – 9	
3	mult	10	11 – 15	
4	sub	10	11 – 12	
5	add	13	14 – 15	Se emitirá cuando se libere la UF de add/sub tras [4]
6	div	16	17 – 21	Se emitirá cuando se libere la UF de mult/div tras [3]

### II. Set A de UF - Ventana centralizada - Emisión desordenada.

Nº Instr	CodOP	ISS	EXE	Comentarios
1	lw	1	2 – 6	
2	add	7	8 – 9	
3	mult	10	11 – 15	
4	sub	10	11 – 12	
5	add	13	14 – 15	Se emitirá cuando se libere la UF de add/sub tras [4]
6	div	1	2 – 6	Al ser emisión desordenada, puede emitirse en el ciclo 1

### III. Set A de UF - Ventana centralizada - Emisión ordenada.

Nº Instr	CodOP	ISS	EXE	Comentarios
1	lw	1	2 – 6	
2	add	7	8 – 9	
3	mult	10	11 – 15	
4	sub	10	11 – 12	

<b>5</b>	add	10	11 – 12	Hay UF suficientes por los que se puede emitir junto a [4]
<b>6</b>	div	10	11 – 15	Hay UF suficientes por los que se puede emitir junto a [3]

**IV. Set B de UF - Ventana centralizada - Emisión desordenada.**

Nº Instr	CodOP	ISS	EXE	Comentarios
<b>1</b>	lw	1	2 – 6	
<b>2</b>	add	7	8 – 9	
<b>3</b>	mult	10	11 – 15	
<b>4</b>	sub	10	11 – 12	
<b>5</b>	add	10	11 – 12	
<b>6</b>	div	1	2 – 6	Al ser emisión desordenada, puede emitirse en el ciclo 1

**V. Set A de UF – Estaciones de reserva - Emisión ordenada.**

**VI. Set A de UF - Estaciones de reserva - Emisión desordenada.**

**VII. Set B de UF - Estaciones de reserva - Emisión ordenada.**

**VIII. Set B de UF - Estaciones de reserva - Emisión desordenada.**

Para todos los casos en los que usamos estaciones de reserva los resultados son iguales que en los casos equivalentes utilizando ventanas de instrucciones (centralizadas). Esto es porque la lógica de *dispatch*, o envío de las instrucciones a las UF de ejecución una vez cumplen los requisitos, es idéntica en ambos casos.

Podemos encontrar diferencias en algunos casos por la composición del código a ejecutar, por ejemplo, por las limitaciones de tamaño (nº de líneas) en las estaciones de reserva. Esto se puede ver en los ejemplos de las diapositivas de teoría donde se utilizan ventanas de instrucciones de 4 líneas, por lo que podemos emitir todas las instrucciones del ejemplo y así, en cada ciclo, evaluar si pueden ser enviadas a ejecución o no. Sin embargo, en el ejemplo siguiente, utilizando estaciones de reserva, cada una de estas tiene 2 líneas. En total tenemos 4, como antes, sin embargo, al tener 3 instrucciones que necesitan la UF de add/sub, sólo podemos insertar 2 en la estación de reserva asociada quedando una en la cola de instrucciones a la espera de que se libere alguna posición en la estación de reserva de add/sub. Esto limita el número máximo de instrucciones de un mismo tipo que podemos enviar a ejecución en un ciclo.

## IC Unidad 2: Superescalares - Problemas de ejecución fuera de orden

1. Fragmento de código a ejecutar en procesador superescalar:
  - Capaz de captar, decodificar y emitir mediante ventana centralizada 3 instrucciones por ciclo. Asumimos ventana centralizada de 3 líneas
  - 2 UF de suma/resta (2 ciclos) y 1 UF de multiplicación (4 ciclos).
  - ROB que permite retirar hasta 2 instrucciones por ciclo.
  - (1) subd f2, f2, f1
  - (2) addd f4, f2, f3
  - (3) subd f5, f2, f3
  - (4) multd f6, f2, f5
  - (5) subd f1, f2, f5
  - (6) subd f7, f4, f6

a) Emisión ordenada (no alineada)

#Instr	Instrucción	IF	ID/ISS	Ventana*1	EX	ROB*2	WB
1	subd f2, f2, f1	1	2	2	3-4	5	6
2	addir f4, f2, f3	1	2-4	2	5-6	7	8
3	subd f5, f2, f3	1	2-4	2	5-6	7	8
4	multd f6, f2, f5	2	3-6	3	7-10	11	12
5	subd f1, f2, f5	2	3-6	5	7-8	9	12
6	subd f7, f4, f6	2	3-10	5	11-12	13	14

b) Emisión desordenada (alineada)

#Instr	Instrucción	IF	ID/ISS	Ventana*1	EX	ROB*2	WB
1	subd f2, f2, f1	1	2	2	3-4	5	6
2	addir f4, f2, f3	1	2-4	2	5-6	7	8
3	subd f5, f2, f3	1	2-4	2	5-6	7	8
4	multd f6, f2, f5	2	3-6	5	7-10	11	12
5	subd f1, f2, f5	2	3-6	5	7-8	9	12
6	subd f7, f4, f6	2	3-10	5	11-12	13	14

c) Emisión desordenada (no alineada)

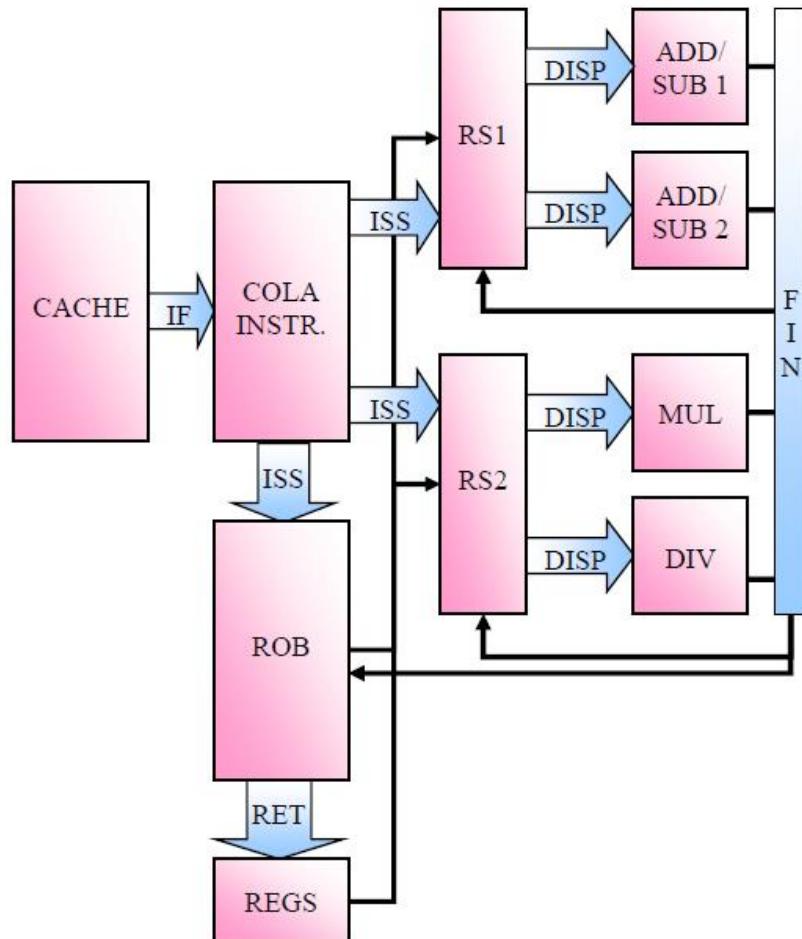
#Instr	Instrucción	IF	ID/ISS	Ventana*1	EX	ROB*2	WB
1	subd f2, f2, f1	1	2	2	3-4	5	6
2	addir f4, f2, f3	1	2-4	2	5-6	7	8
3	subd f5, f2, f3	1	2-4	2	5-6	7	8
4	multd f6, f2, f5	2	3-6	3	7-10	11	12
5	subd f1, f2, f5	2	3-6	5	7-8	9	12
6	subd f7, f4, f6	2	3-10	5	11-12	13	14

\*1 Ciclo en el que las instrucciones se insertan en la ventana

\*2 Ciclo en el que se actualizan los valores de los registros destino en el ROB y en las entradas relacionadas de la ventana de instrucciones

2. En la **cola de instrucciones** encontramos:

- (1) addd f3, f1, f2 ;  $f3 = f1 + f2$
- (2) addd f2, f3, f2 ;  $f2 = f3 + f2$
- (3) multd f4, f3, f2 ;  $f4 = f3 * f2$
- (4) divd f5, f2, f1 ;  $f5 = f2 / f1$
- (5) subd f2, f3, f1 ;  $f2 = f3 - f1$



Instrucción	ID/ISS	DISP/EX	FIN/ROB	RET/WB	Comentario
addd f3, f1, f2	1	2-3	4	5	$f1=10, f2=5 \rightarrow f3=15$
addd f2, f3, f2	1	4-5	6	7	RAW (1) $\rightarrow f2= 20$
multd f4, f3, f2	2	6-10	11	12	RAW(1)(2) $\rightarrow f4= 300$
divd f5, f2, f1	2	6-45	46	47	RAW(2) $\rightarrow f5= 2$
subd f2, f3, f1	3	4-5	6	47	RAW(1) $\rightarrow f2 5$

## IC Unidad 2: Superescalares - Problemas de riesgos de control

1. Indicar la penalización efectiva (porcentaje de casos en que falla la predicción) que se produce si se utiliza

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
S	S	N	N	N	S	S	N	S	N	S	N	S	S	S	S	S	N

- a) Predicción fija (siempre *Not Taken*)

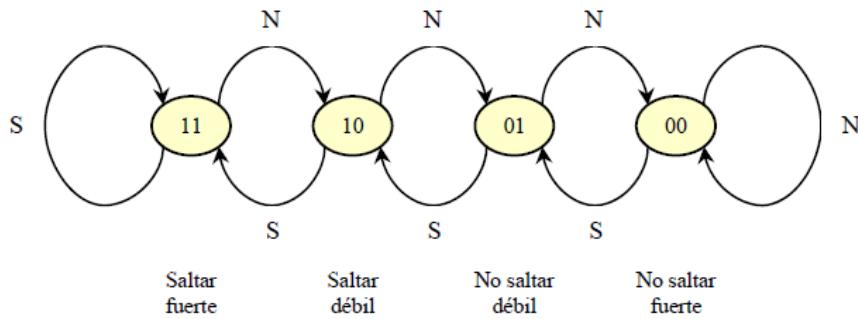
Números de fallos: 11     $P = 11/18 = 0'61 = 61\%$  penalización.

- b) Predicción estática (si salto atrás *Taken*, si hacia adelante *Not Taken*)

Número de No Tomados = 7    Número de fallos = 7

$P = 7/18 = 0.39 = 39\%$  penalización

- c) Predicción dinámica con 2 bits (inicialmente en 11)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Bits predicción	<b>11</b>	<b>11</b>	<b>10</b>	<b>01</b>	<b>00</b>	<b>01</b>	<b>10</b>	<b>01</b>	<b>10</b>	<b>01</b>	<b>10</b>	<b>01</b>	<b>10</b>	<b>11</b>	<b>11</b>	<b>11</b>	<b>10</b>	
Predicción Salto	S	S	S	S	N	N	N	S	N	S	N	S	N	S	S	S	S	
Resultado Salto	S	S	N	N	N	S	S	N	S	N	S	N	S	S	S	S	N	
Penalización	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>								

$P = 11/18 = 0'61 = 61\%$  penalización

- d) Predicción dinámica con 3 bits (inicialmente en 111)

Cada entrada guarda las 3 últimas ejecuciones del salto (0 No tomado, 1 Tomado). Se predice según el bit mayoritario. La actualización de los bits de estado se realiza en modo

FIFO, desplazando de bit de mayor peso a bit de menor peso, insertando un 0 si el último salto no ha sido efectivo o un 1 si lo ha sido. Queda la siguiente tabla:

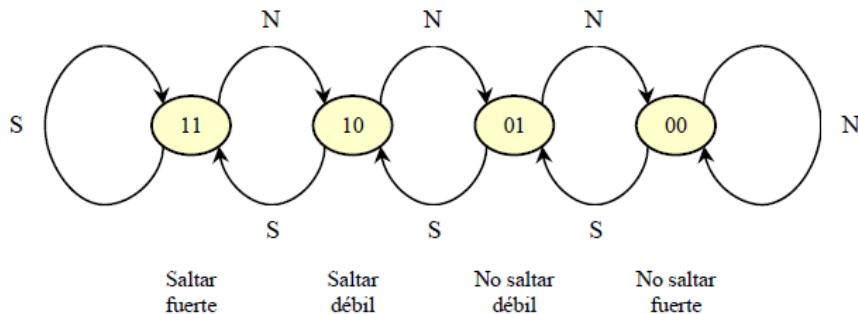
Estado	Salto
000	N
001	N
010	N
011	S
100	N
101	S
110	S
111	S

	INICIO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Bits predicción	<b>111</b>	<b>111</b>	<b>111</b>	<b>011</b>	<b>001</b>	<b>000</b>	<b>100</b>	<b>110</b>	<b>011</b>	<b>101</b>	<b>010</b>	<b>101</b>	<b>010</b>	<b>101</b>	<b>110</b>	<b>111</b>	<b>111</b>	<b>011</b>	
Predicción Salto		S	S	S	S	N	N	N	S	S	S	N	S	N	S	S	S	S	
Resultado Salto		S	S	N	N	N	S	S	N	S	N	S	N	S	S	S	S	N	
Penalización		<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>

$$P = 10/18 = 0'56 = 56\% \text{ penalización}$$

2. El programa del enunciado calcula el valor absoluto de los elementos del vector  $X$ . Va procesándolos de uno en uno y les cambia el signo si no son positivos. Consta de dos saltos, el primero de ellos, notado como S1 en el código de más abajo, sirve para detectar si los números son positivos y el segundo, notado como S2, se emplea para volver a ejecutar el código mientras queden elementos por procesar en el vector  $X$ .

- (1) `lw r1, N`
- (2) `add r2, r0, r0`
- (3) bucle: `lw r3, X(r2)`
- (4) `sgt r4, r3, r0`
- (5) **`bnz r4, mayor S1`**
- (6) `sub r3, r0, r3`
- (7) mayor: `sw X(r2), r3`
- (8) `add r2, r2, #4`
- (9) `sub r1, r1, #1`
- (10) **`bnz r1, bucle S2`**



Para cada salto, su predictor dinámico se inicializará según el predictor estático del computador, que predice como tomados los saltos hacia atrás. Por tanto, el predictor dinámico para S1 se iniciará al valor 00 (no saltar fuerte), ya que es un salto hacia delante, y el predictor dinámico para S2 se iniciará al valor 11 (saltar fuerte) porque salta hacia atrás.

Una vez inicializados los predictores dinámicos, la penalización que introduzca cada uno de ellos dependerá del comportamiento de cada salto. En el caso de S1, su comportamiento está determinado por el valor de cada elemento del vector  $X$ . S1 Saltará siempre que  $X(i)$  sea mayor que cero, y según el enunciado, el vector  $X$  es de la forma  $X = \{1, 0, 1, 0, 1, 0, \dots\}$ , por lo que S1 saltará una vez sí y otra no hasta que se termine de procesar el vector  $X$ . Por tanto, el comportamiento del predictor dinámico para S1 viene explicado según la siguiente tabla:

Iteración	1	2	3	4	6	7	8	...
Valor de $X$	1	0	1	0	1	0	1	...
Estado actual	00	01	00	01	00	01	00	...
Predicción	N	N	N	N	N	N	N	...
Ejecución	S	N	S	N	S	N	S	...
Penalización	P		P		P		P	...
Estado siguiente	01	00	01	00	01	00	01	...

Por lo que la penalización que introduce S1 es de:

$$Ps1 = 5 \times (N/2) \text{ ciclos}$$

El salto S2 saltará tantas veces como elementos tenga el vector X, es decir N veces, ya que el programa se dedica a procesar dicho vector. Teniendo esto en cuenta, el comportamiento del predictor dinámico para S2 viene explicado según la siguiente tabla:

Iteración	1	2	3	4	...	N-2	N-1	N
Estado actual	11	11	11	11	...	11	11	
Predicción	S	S	S	S	...	S	S	S
Ejecución	S	S	S	S	...	S	S	N
Penalización								P
Estado siguiente	11	11	11	11	...	11	11	10

El comportamiento de este salto es muy fácil de predecir, ya que salta siempre excepto en la última iteración del bucle, por lo que introduce una penalización de:

$$Ps2 = 5 \text{ ciclos}$$

Por tanto, teniendo en cuenta los dos predictores, la penalización total es de:

$$P_{total} = 5x(1+N/2) \text{ ciclos}$$

**Pregunta 1**No s'ha respond  
encara

Puntuat sobre 1,00

▼ Marca la  
pregunta

Indica cuál de las siguientes afirmaciones es correcta:

Trieu-ne una:

- a. Los bloqueos activos no se pueden solucionar
- b. La prevención de los interbloqueos es una estrategia pesimista
- c. Los bloqueos activos no se pueden dar nunca en las redes de interconexión directas
- d. La recuperación de los interbloqueos es una estrategia pesimista

Por descarte

**Pregunta 2**No s'ha respond  
encara

Puntuat sobre 1,00

▼ Marca la  
pregunta

¿Cuántos comutadores tiene en la segunda columna de comutadores de una red multi-etapa tipo Delta 25x16?

Trieu-ne una:

- a. 4
- b. 8
- c. 16
- d. 5

$$4^2 \rightarrow 16 \quad (2^{\text{a}} \text{ Columna} \rightarrow 16 \rightarrow \sqrt{16} = 4)$$

**Pregunta 3**

Resposta desada

Puntuat sobre 1,00

▼ Marca la  
pregunta

El paralelismo de datos:

Trieu-ne una:

- a. Se encuentra implícito en funciones lógicas
- b. es un tipo de paralelismo que nunca se encuentra de forma implícita.
- c. Se encuentra implícito en funciones aritméticas
- d. Se encuentra implícito en operaciones con imágenes

**Pregunta 4**No s'ha respond  
encara

Puntuat sobre 1,00

▼ Marca la  
pregunta

El diámetro de una malla abierta es:

Trieu-ne una:

- a.  $2*(r/2 - 1)$  siendo r la raíz cuadrada del número de nodos
- b.  $2*(r-1)$  siendo r la raíz cuadrada del número de nodos
- c.  $2*N/2$  siendo N el número de nodos
- d.  $2*(r-1)$  siendo r el número de nodos

$$2(r-1) \text{ donde } N = r^2 \rightarrow r = \sqrt{N}$$

**Pregunta 5**No s'ha respond  
encara

Puntuat sobre 1,00

▼ Marca la  
pregunta

En un multicomputador, ¿qué se utiliza para permitir una comunicación asíncrona entre cualesquiera de sus nodos?

Trieu-ne una:

- a. Una red directa o estática
- b. Es imposible establecer una comunicación asíncrona en un computador paralelo
- c. Buffers intermedios
- d. Enlaces con más ancho de banda

**Pregunta 6**

No s'ha respot encara

Puntuat sobre 1,00

Marca la pregunta

¿Cuál es el factor de aceleración, ganancia en velocidad o speed-up que puede conseguir como máximo si tiene 2 nodos conectados formando una máquina paralela y una aplicación cuyo 75% del código ha conseguido parallelizar? Suponga que el tiempo de sobrecarga no influye.

Trieu-ne una:

 a. 1.666 b. 4 c. 1.250 d. 2 $P = \text{nº nodos}$ 

$$\text{Ganancia} = Ts / Tp(p) = Ts / Ts * (\frac{1}{4} + \frac{3}{4} * \frac{1}{p}) = 4p / p+3$$

$$\text{Si } P=2 \rightarrow 8 / 5 = 1.6$$

**Pregunta 7**

No s'ha respot encara

Puntuat sobre 1,00

Marca la pregunta

La técnica de comunicación de reducción consiste en:

Trieu-ne una:

 a. Enviar un único elemento pero que es recibido por varios procesadores b. Enviar un único elemento a un único procesador c. Enviar varios elementos a la vez a varios procesadores d. A partir de varios elementos, utilizar una función para convertirlos en uno solo, que recibe un único procesador**Pregunta 8**

Resposta desada

Puntuat sobre 1,00

Marca la pregunta

En cierto momento del transcurso de la ejecución de una aplicación paralela en un multicomputador de 4 nodos, todos sus nodos (N0,N1,N2,N3) envían un dato distinto al nodo N0. Dicho nodo espera a que lleguen todos los datos y realiza la siguiente operación antes de continuar la ejecución de la aplicación:

Resultado = dato\_N0 + dato\_N1 + dato\_N2 + Dato\_N3

¿De qué tipo de comunicación estamos hablando?

Trieu-ne una:

 a. Reducción b. Difusión c. Dispersión d. Acumulación**Pregunta 9**

Resposta desada

Puntuat sobre 1,00

Marca la pregunta

Una red Omega es

Trieu-ne una:

 a. Una red directa b. Una red jerárquica c. Una red indirecta d. Una red basada en buses de contención**Pregunta 10**

Resposta desada

Puntuat sobre 1,00

Marca la pregunta

¿De qué tipo es la red de comunicación que ha empleado en el laboratorio para hacer las prácticas?

Trieu-ne una:

 a. MPP b. Red de computadores c. Cluster d. Cluster Beowulf

¿Cuál es la unidad de transferencia en la técnica de conmutación Worm-Hole?

Trieu-ne una:

- a. El mensaje
- b. El paquete
- c. El phit
- d. El flit

Diferencia entre los multiprocesadores y multicomputadores

**Solución:** Los multiprocesadores comparten el mismo espacio de memoria

Modo de programación SPMD

**Solución:** Un solo programa que puede actuar sobre varios flujos de datos.

De que depende el diseño de una red

**Solución:** De la topología del control de flujo y del Encaminamiento

Donde está implícito el paralelismo de datos.

**Solución:** En las operaciones con estructuras de datos

Diámetro de una red en árbol binario de 15 nodos

**Solución:** 6      ->     $2 * (k - 1)$  , k = niveles árbol, serian 4, por tanto  $2 * 3 = 6$

En que unidad se mide la red worm-hole

**Solución:** transferencia entre interfaces (unidad: paquete/mensaje) – Transf. Entre conmut. (unidad: flit)

Un multiprocesador de 64 procesadores utiliza una red Omega con conmutadores de grado 2. ¿Cuántos conmutadores utiliza en total?

Trieu-ne una:

- a. 60
- b. 192
- c. 64
- d. 32

Jorge A. b) En un pdf por internet he encontrado la siguiente fórmula

$(P/k) \log_k P$

$(64/2) * (\log 64 / \log 2) = 192$

¿Cuál es el máximo factor de aceleración, ganancia en velocidad o speed-up que puede conseguir como máximo si tiene 5 nodos conectados, el tiempo de sobrecarga no influye y dispone de una aplicación totalmente paralelizable?

Trieu-ne una:

- a. 10
- b. 3
- c. 5
- d. 2.5

Lo ideal en una aplicación totalmente paralela es lineal al número de procesadores, es decir, 5

$$\frac{T_s}{T_p(P)} = \frac{T_s}{\frac{T_s}{P}} = \frac{1}{\frac{1}{5}} = 5$$

¿Cuántas **etapas o columnas** de conmutadores tiene una red multi-etapa tipo Delta 25x16?

Trieu-ne una:

- a. 4
- b. 25
- c. 5
- d. 2

Pag103:  $5^2 * 4^2, 5 * 4$  conmutadores y 2 etapas...

La estructura de paralelismo "divide y vencerás" consiste en:

Trieu-ne una:

- a. Es la misma estrategia que "master-slave"
- b. *Distribuir diferentes partes de una tarea entre varios procesadores sin una relación de dependencia tipo master-slave*
- c. "Divide y vencerás" no tiene nada que ver con el paralelismo
- d. Se divide una tarea en diferentes funciones que se aplican consecutivamente, de forma secuencial, sobre todos los datos

La b) por descarte. A y c es seguro que NO, la d) creo que no

## IC Segundo parcial

1. Indica cuál de las siguientes afirmaciones es correcta:
  - La **prevención** de los **interbloqueos** es una estrategia permisista.
2. ¿Cuántos conmutadores tiene en la segunda columna de conmutadores de una red multi-etapa Delta26x16?
  - 4 ( $16 = 4^2$  donde 4 son conmutadores y 2 las etapas)
3. El paralelismo de datos:
  - En las operaciones de **estructuras de datos**.
4. El diámetro de una malla abierta es:
  - $2*(r-1)$  siendo r el número de **nodos**.
5. En un multicomputador, ¿qué se utiliza para permitir una comunicación asíncrona entre cualesquiera de sus nodos?
  - Buffers **intermedios**.
6. ¿Cuál es el factor de aceleración, ganancia en velocidad o speed-up que puede conseguir como máximo si tiene 2 nodos conectados formando una máquina paralela y una aplicación cuyo 75% del código ha conseguido parallelizar? Suponga que el tiempo de sobrecarga no influye.
  - **1.666**.
7. La técnica de comunicación de reducción consiste en:
  - A partir de varios elementos, **utilizar una función para convertirlos en uno solo**, que recibe un único procesador.
8. En cierto momento del transcurso de la ejecución de una aplicación paralela en un multicomputador de 4 nodos, todos sus nodos (N0, N1, N2, N3) envían un dato distinto al nodo N0. Dicho nodo espera a que lleguen todos los datos y realiza la siguiente operación antes de continuar la ejecución de la aplicación:  
Resultado = dato\_N0 + dato\_N1 + dato\_N2 + Dato\_N3  
¿De qué tipo de comunicación estamos hablando?
  - Reducción.
9. Una red omega es:
  - Una red **indirecta**.
10. ¿De qué tipo es la red de comunicación que ha empleado en el laboratorio para hacer prácticas?
  - Red de computadores.
11. ¿Cuál es la unidad de transferencia en la técnica de conmutación Worm-Hole?
  - Flit (comprobada Wikipedia)

12. Diferencia entre los multiprocesadores y multicomputadores.

- Los **multiprocesadores** comparten el **mismo espacio de memoria**.

13. Modo de programación SPMD.

- Un solo programa que puede actuar sobre varios flujos de datos.

14. De que depende el diseño de una red.

- De la topología del control de flujo y encaminamiento.

15. Diámetro de una red en árbol binario de 15 nodos:

- $6 \rightarrow 2*(k-1)$ , k = niveles árbol, serian 4, por tanto  $2*3 = 6$ .

16. Un procesador de 64 procesadores utiliza una red Omega con comutadores de grado

2. ¿Cuántos comutadores utiliza en total?

- $192 ((N/2) * \log_2(N) \rightarrow (64/2) * \log_2(64) = 192)$

17. ¿Cuál es el máximo factor de aceleración, ganancia en velocidad o speed-up que puede conseguir como máximo si tiene 5 nodos conectados, el tiempo de sobrecarga no influye y dispone de una aplicación totalmente paralelizable?

- 5

18. ¿Cuántas etapas o columnas de comutadores tiene una red multi-etapa tipo Delta  
25\*16?

- 2 ( $5^2+4^2 \rightarrow 5*4$  comutadores y 2 etapas).

Red delta  $a^n \times b^n$  donde n es el número de etapas, comutadores =  $a*b$

19. La estructura de paralelismo “divide y vencerás” consiste en:

Distribuir diferentes partes de una tarea entre varios procesadores sin una relación de dependencia tipo master-slave.